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	25MHz
Co-Processors/DSP	Communications; CPM
RAM Controllers	DRAM
Graphics Acceleration	No
Display & Interface Controllers	-
Ethernet	10Mbps (1)
SATA	-
USB	-
Voltage - I/O	3.3V
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/mc68360vr25vlr2

—Glueless Interface to DRAM Single In-Line Memory Modules (SIMMs), Static Random-Access Memory (SRAM), Electrically Programmable Read-Only Memory (EPROM), Flash EPROM, etc.
—Four CAS lines, Four WE lines, One OE line
—Boot Chip Select Available at Reset (Options for 8-, 16-, or 32-Bit Memory)
—Special Features for MC68040 Including Burst Mode Support

- Four General-Purpose Timers
  - —Superset of MC68302 Timers
  - —Four 16-Bit Timers or Two 32-Bit Timers
  - —Gate Mode Can Enable/Disable Counting
- Two Independent DMAs (IDMAs)
  - —Single Address Mode for Fastest Transfers
  - —Buffer Chaining and Auto Buffer Modes
  - —Automatically Performs Efficient Packing
  - —32-Bit Internal and External Transfers
- System Integration Module (SIM60)
  - —Bus Monitor
  - —Double Bus Fault Monitor
  - —Spurious Interrupt Monitor
  - —Software Watchdog
  - —Periodic Interrupt Timer
  - —Low Power Stop Mode
  - —Clock Synthesizer
  - —Breakpoint Logic Provides On-Chip Hardware Breakpoints
  - —External Masters May Use On-Chip Features Such As Chip Selects
  - —On-Chip Bus Arbitration with No Overhead for Internal Masters
  - —IJTAG Test Access Port
- Interrupts
  - —Seven External IRQ Lines
  - —12 Port Pins with Interrupt Capability
  - —16 Internal Interrupt Sources
  - —Programmable Priority Between SCCs
  - —Programmable Highest Priority Request
- Communications Processor Module (CPM)
  - —RISC Controller
  - —Many New Commands (e.g., Graceful Stop Transmit, Close RxBD)
  - —224 Buffer Descriptors
  - —Supports Continuous Mode Transmission and Reception on All Serial Channels
  - —2.5 Kbytes of Dual-Port RAM
  - —14 Serial DMA (SDMA) Channels
  - —Three Parallel I/O Registers with Open-Drain Capability
  - —Each Serial Channel Can Have Its Own Pins (NMSI Mode)
- Four Baud Rate Generators

- Parallel Interface Port
  —Centronics[4] Interface Support
  —Supports Fast Connection Between QUICCs
- 240 Pins Defined: 241-Lead Pin Grid Array (PGA) and 240-Lead Plastic Quad Flat Pack (PQFP)

1.2 QUICC ARCHITECTURE OVERVIEW

The QUICC is 32-bit controller that is an extension of other members of the Motorola M68300 family. Like other members of the M68300 family, the QUICC incorporates the inter-module bus (IMB). (The MC68302 is an exception, having an M68000 bus on chip.) The IMB provides a common interface for all modules of the M68300 family, which allows Motorola to develop new devices more quickly by using the library of existing modules. Although the IMB definition always included an option for an on-chip 32-bit bus, the QUICC is the first device to implement this option.

The QUICC is comprised of three modules: the CPU32+ core, the SIM60, and the CPM. Each module utilizes the 32-bit IMB. The MC68360 QUICC block diagram is shown in Figure 1-1.

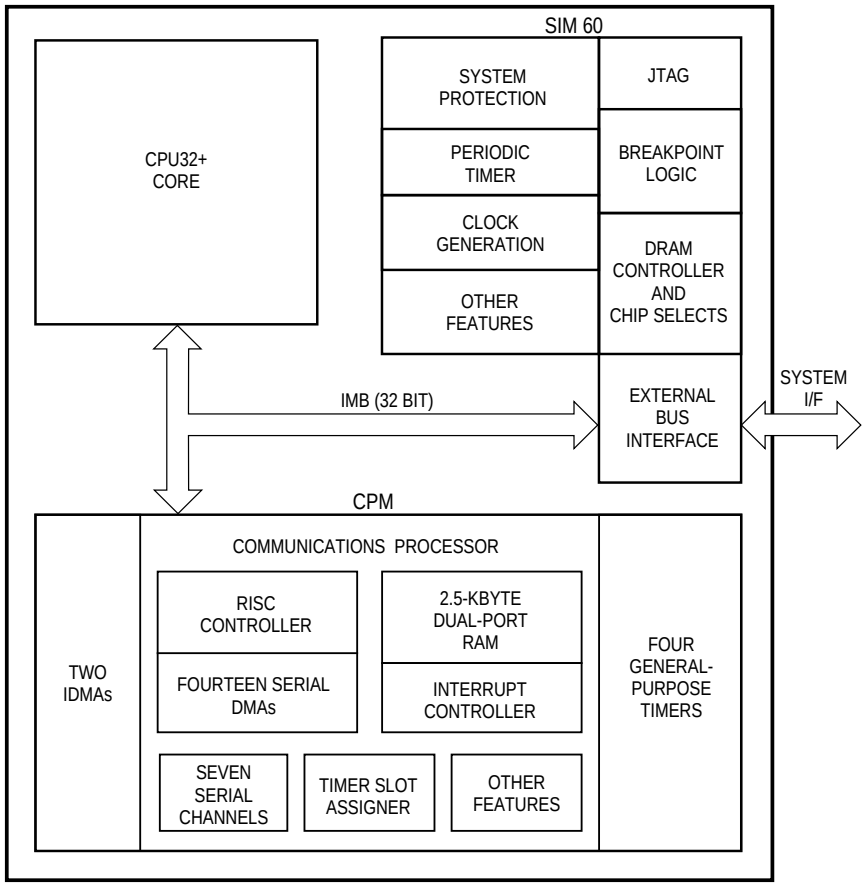


Figure 1-1. QUICC Block Diagram

4. Centronics is a trademark of Centronics, Inc.

NOTE

Write enable does not have the capability to follow dynamic bus sizing with external assertion of DSACK. Write enable will always follow the port size that is programed in GMR and the OR. For more information see 6.10 Memory Controller.

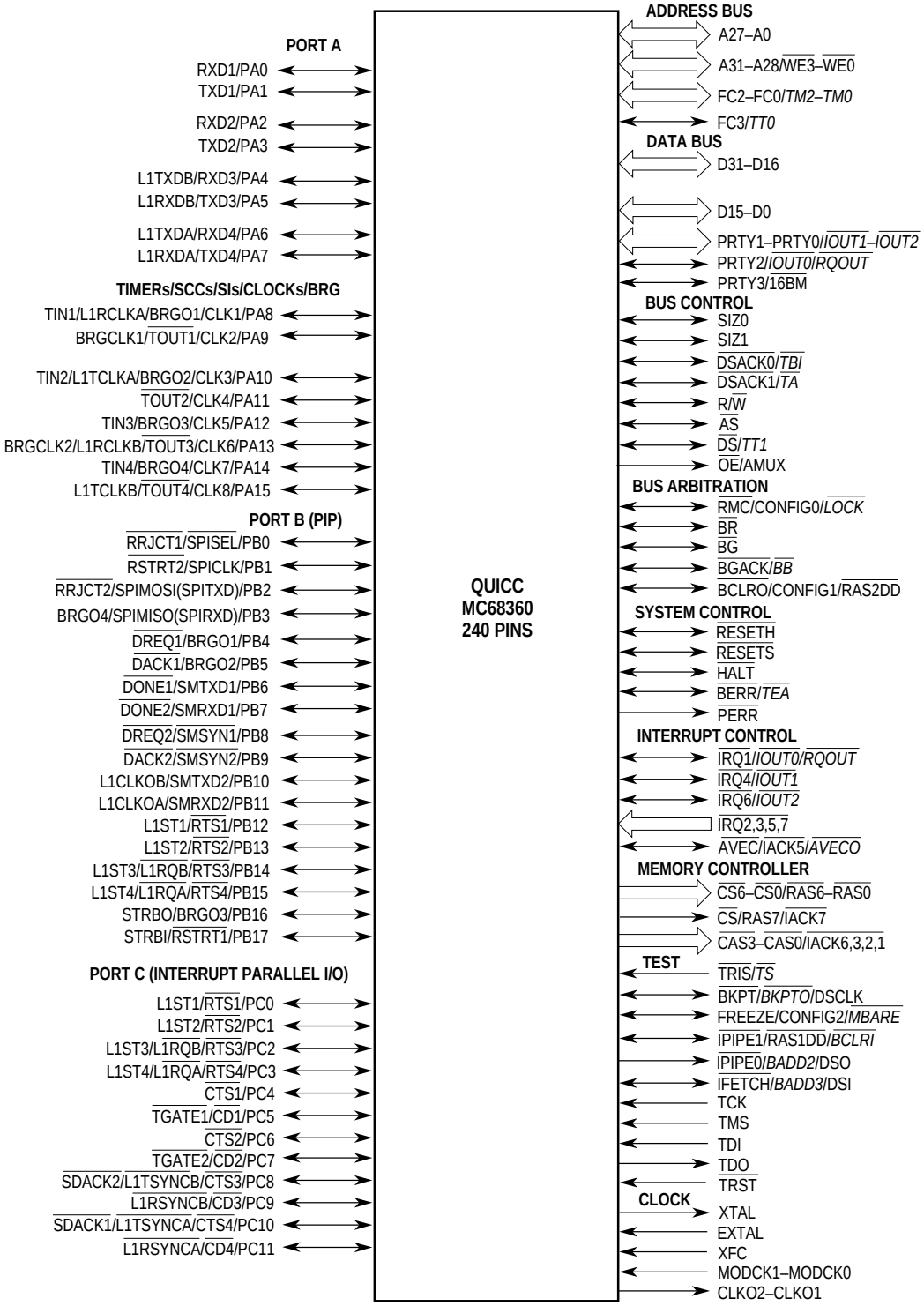


Figure 2-1. QUICC Functional Signal Groups

The UNLK instruction removes a stack frame from the end of the list by loading an address into the SP and pulling the value at that address from the stack. When the instruction operand is the address of the link address at the bottom of a stack frame, the effect is to remove the stack frame from both the stack and the linked list.

5.3.6 Pipeline Synchronization with the NOP Instruction

Although the no operation (NOP) instruction performs no visible operation, it does force synchronization of the instruction pipeline, since all previous instructions must complete execution before the NOP begins.

5.4 PROCESSING STATES

This section describes the processing states of the CPU32+. It includes a functional description of the bits in the supervisor portion of the SR and an overview of actions taken by the processor in response to exception conditions.

5.4.1 State Transitions

The processor is always in one of four processing states: normal, background, exception, or halted.

When the processor fetches instructions and operands or executes instructions, it is in the normal processing state. The stopped condition, which the processor enters when a STOP or LPSTOP instruction is executed, is a variation of the normal state in which no further bus cycles are generated.

Background state is an alternate operational mode used for system debugging. Refer to 5.6 Development Support for more information.

Exception processing refers specifically to the transition from normal processing of a program to normal processing of system routines, interrupt routines, and other exception handlers. Exception processing includes the stack operations, the exception vector fetch, and the filling of the instruction pipeline caused by an exception. Exception processing ends when execution of an exception handler routine begins. Refer to 5.5 Exception Processing for comprehensive information.

A catastrophic system failure occurs if the processor detects a bus error or generates an address error while in the exception processing state. This type of failure halts the processor. For example, if a bus error occurs during exception processing caused by another bus error, the CPU32+ assumes that the system is not operational and halts.

The halted condition should not be confused with the stopped condition. After the processor executes a STOP or LPSTOP instruction, execution of instructions can resume when a trace, interrupt, or reset exception occurs.

5.4.2 Privilege Levels

To protect system resources, the processor can operate with either of two levels of access—user or supervisor. Supervisor level is more privileged than user level. All instructions are

5.6.2.2.1 External $\overline{\text{BKPT}}$ Signal. Once enabled, BDM is initiated whenever assertion of $\overline{\text{BKPT}}$ is acknowledged. If BDM is disabled, a breakpoint exception (vector $0C) is acknowledged. The $\overline{\text{BKPT}}$ input has the same timing relationship to the data strobe trailing edge as read cycle data. There is no breakpoint acknowledge bus cycle when BDM is entered.

5.6.2.2.2 BGND Instruction. An illegal instruction, $4AFA, is reserved for use by development tools. The CPU32+ defines $4AFA (BGND) to be a BDM entry point when BDM is enabled. If BDM is disabled, an illegal instruction trap is acknowledged. Illegal instruction traps are discussed in 5.5.2.8 Illegal or Unimplemented Instructions.

5.6.2.2.3 Double Bus Fault. The CPU32+ normally treats a double bus fault (two bus faults in succession) as a catastrophic system error and halts. When this condition occurs during initial system debug (a fault in the reset logic), further debugging is impossible until the problem is corrected. In BDM, the fault can be temporarily bypassed so that its origin can be isolated and eliminated.

5.6.2.3 ENTERING BDM. When the processor detects a $\overline{\text{BKPT}}$ or a double bus fault or decodes a BGND instruction, it suspends instruction execution and asserts the FREEZE output. FREEZE assertion is the first indication that the processor has entered BDM. Once FREEZE has been asserted, the CPU enables the serial communication hardware and awaits a command.

The CPU writes a unique value indicating the source of BDM transition into temporary register A (ATEMP) as part of the process of entering BDM. A user can poll ATEMP and determine the source (see Table 5-20) by issuing a read system register command (RSREG). ATEMP is used in most debugger commands for temporary storage—it is imperative that the RSREG command be the first command issued after transition into BDM.

Table 5-20. Polling the BDM Entry Source

Source	ATEMP 31–16	ATEMP 15–0
Double Bus Fault	SSW*	$FFFF
BGND Instruction	$0000	$0001
Hardware Breakpoint	$0000	$0000

*SSW is described in detail in 5.5.3 Fault Recovery.

A double bus fault during initial SP/PC fetch sequence is distinguished by a value of $FFFFFFFF in the current instruction PC. At no other time will the processor write an odd value into this register.

5.6.2.4 COMMAND EXECUTION. Figure 5-21 summarizes BDM command execution. Commands consist of one 16-bit operation word and can include one or more 16-bit extension words. Each incoming word is read as it is assembled by the serial interface. The microcode routine corresponding to a command is executed as soon as the command is complete. Result operands are loaded into the output shift register to be shifted out as the next command is read. This process is repeated for each command until the CPU returns to normal operating mode.

The keyed write uses the MBAR enable (MBARE) register at address $0003FF08 and the $\overline{\text{MBARE}}$ pin. Both the newly located MBAR and the MBARE are located in the CPU address space FC = 111.

With multiple QUICCs configured in slave mode, the following keyed write is used to enable the MBAR programming: the user writes the MBAR select bit of MBARE with a 1 while the $\overline{\text{MBARE}}$ pin is a logic zero. Once this is accomplished, the MBAR may be written at its new location (using the standard MBAR writing techniques). Once MBAR is written (in particular, the low-order byte of MBAR), then the MBAR select bit in MBARE is cleared, and further accesses to MBAR are impossible until the keyed write technique is used again. There is no time limit imposed between the keyed write and the MBAR write; however, once the keyed write for a particular QUICC slave has occurred, the MBAR of that slave should be written before performing another keyed write to another QUICC slave.

The keyed write may be performed gluelessly to multiple QUICC slaves in the following way. Connect (in hardware) the $\overline{\text{MBARE}}$ pin of QUICC slave #1 to bit zero of the data bus(D0). Connect the $\overline{\text{MBARE}}$ pin of QUICC slave #2 to D1, etc. Then perform the following operations in software:

1. Write the MBARE of QUICC slave #1 at $0003FF08 with value $FFFFFFFE. This sets the MBAR select bit (bit 31) and places a low voltage on only the $\overline{\text{MBARE}}$ pin of QUICC slave #1.
2. Now the MBAR of QUICC slave #1 can be accessed at $0003FF04 and written using normal MBAR writing procedures.
3. Write the MBARE of QUICC slave #2 with the value $FFFFFFFD.
4. Now the MBAR of QUICC slave #2 can be written.

This technique will work for up to 31 QUICC slaves in the system, using no glue or parallel I/O pins.

6.8.2 Global Chip Select ($\overline{\text{CS0}}$) in Slave Mode

When the QUICC is in slave mode, the user may choose whether to enable the global chip-select operation of $\overline{\text{CS0}}$. (The global chip select is used for boot ROMs and is described in 6.10 Memory Controller.) The global chip select can be either enabled or disabled by the configuration on the CONFIG pins during power-up reset and system reset ($\overline{\text{RESETH}}$ asserted). When the global chip select function is disabled, $\overline{\text{CS0}}$ can still be enabled later and used as a normal chip select, if desired.

6.8.3 Bus Clear in Slave Mode

The bus clear out ($\overline{\text{BCLRO}}$) pin can be selected to signify to the external logic that the DRAM refresh controller, IDMA channels, or SDMA channels are requesting the bus. However, in slave mode, the $\overline{\text{BCLRO}}$ pin may become the $\overline{\text{RAS2DD}}$ double drive pin, and a new pin called bus clear in ($\overline{\text{BCLRI}}$) is defined (at another location in the pinout). $\overline{\text{BCLRI}}$ indicates to that QUICC that a request is being made for the QUICC to release the system bus. The EBI 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.

EMWS—External Master Wait State (SRAM Bank Only)

This attribute should be set if an additional wait state is necessary when an asynchronous external MC68030-type device or external QUICC is accessing SRAM banks (see Table 6-11). This bit is only used if SYNC = 0.

0 = Normal operation.
1 = Insert one additional wait state for external QUICC/MC68030-type masters on their accesses to all SRAM banks.)

Table 6-11. External MC68030-Type Cycle Length (SRAM Bank in Asynchronous Operation

	External QUICC/MC68030-Type Bus Cycle Length			
	Synchronous Bus Timing (BSTM = 1)		Asynchronous Bus Timing (BSTM = 0)	
TCYC =	EMWS = 0	EMWS = 1	EMWS = 0	EMWS = 1
0	3	3	3	3
1	3	4	3	5
2	4	5	5	6
3	5	6	6	7
4	6	7	7	8
5	7	8	8	9
6	8	9	9	10
...	...	...	...	...
15	17	18	18	19

NOTE: The BSTM bit is located in the MCR of the SI60.

The following bits are used for both DRAM and SRAM memory:

SYNC—Synchronous External Access MC68030-Type

This attribute applies only to an external MC68030-type device or external QUICC that uses the on-chip memory controller. It determines how the memory controller will assert its signals in response to what it sees from the external master.

0 = Asynchronous operation of the memory controller (external MC68030-type master only).

When the SRAM controller is used, $\overline{CS}$ and $\overline{DSACK}$ assertion and negation timings are asynchronous. They are asserted and negated in relation to the external master's $\overline{AS}$ line. The CSNTQ and the TRLXQ attributes are ignored. When EMWS is set, one wait state is added to the programmed TCYC.

When the DRAM controller is used, $\overline{CAS}$ and $\overline{DSACK}$ are negated asynchronously with the negation of the external master's $\overline{AS}$.

NOTE

The DRAM controller's assertion of $\overline{RAS}$ and $\overline{CAS}$ is always synchronous to the QUICC clock. When asynchronous external masters are using the DRAM controller, the BSTM bit in the MCR should be cleared.

7.4.8 RISC Timer Interrupt Handling

The following sequence describes what would normally occur within an interrupt handler for the RISC timer tables:

1. Once an interrupt occurs, read the RISC timer event register to see which timer or timers have caused interrupts. The RISC timer event bits would normally be cleared at this time.
2. Issue additional SET TIMER commands at this time or later, as desired. Nothing need be done if the timer is being restarted automatically for a repetitive interrupt.
3. Clear the R-TT bit in the CPM interrupt status register.
4. Execute the RTE instruction.

7.4.9 RISC Timer Table Algorithm

The RISC scans the timer table once every tick. For each valid timer in the timer table, the RISC decrements the count and checks for a timeout. If no timeout occurs, it moves to the next timer. If a timeout occurs, the RISC sets the corresponding event bit in the RISC timer event register. It checks to see if the timer is to be restarted. If so, it leaves the timer valid bit set in the R_TMV location and resets the current count to the initial count; otherwise, it clears the R_TMV bit. Once the timer table is scanned, the RISC updates the TM_cnt value in the RISC timer table parameter RAM and ceases working on the timer tables until the next tick.

If a SET TIMER command is issued, the RISC controller makes the appropriate modifications to the timer table and parameter RAM, but does not scan the timer table until the next tick of the internal timer. It is important to use the SET TIMER command to properly synchronize the timer table alterations to the execution of the RISC.

7.4.10 RISC Timer Table Application: Track the RISC Loading

The RISC timers can be used to track the loading of the RISC controller. The following sequence gives a method for using the 16 RISC timers to determine if the RISC controller ever exceeds the 96% utilization level during any tick interval. Removing the timers then adds a 4% margin to the RISC utilization level. The aggressive user can use this technique to push the RISC performance to its limit in an application.

The user should use the standard initialization sequence, with the following differences:

1. Program the tick of the RISC timers to be 1024 x 16 = 16384.
2. Disable RISC timer interrupts, if desired.
3. Using the SET TIMER command, initialize all 16 RISC timers to have a timer period of $0000, which equates to 65536.
4. Program one of the four general-purpose timers to increment once every tick. The general-purpose timer should be free-running and should have a timeout of 65536.
5. After hours of operation, compare the general-purpose timer to the current count of RISC timer 15. If RISC timer 15 is more than two ticks different from the general-purpose timer, the RISC controller has, during some tick interval, exceeded the 96% uti-

The IBPTR entry points to the next BD that the IDMA will transfer data to when it is in IDLE state or points to the current BD during transfer processing. After a reset or when the end of an IDMA BD table is reached, the CP initializes this pointer to the value programmed in the IBASE entry.

ISTATE and ITEMP are for RISC use only.

7.6.4.2.2 IDMA Buffer Descriptors (BDs). Source addresses, destination addresses, and byte counts are presented to the RISC controller using special IDMA BDs. The RISC controller reads the BDs, programs the IDMA channel, and notifies the CPU32+ about the completion of a buffer transfer using the IDMA BDs. This concept is like that used for the serial channels on the QUICC, except that the BD is larger to contain additional information.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET + 0	V	—	W	I	L	—	CM	—	—	—	—	—	—	SE	DE	DA
OFFSET + 2																
OFFSET + 4	**DATA LENGTH**															
OFFSET + 6																
OFFSET + 8	**SOURCE DATA BUFFER POINTER**															
OFFSET + A																
OFFSET + C	**DESTINATION DATA BUFFER POINTER**															
OFFSET + E																

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

The following bits are prepared by the user before transfer and are set by the RISC controller after the buffer has been transferred.

V—Valid

0 = The data buffers associated with this BD are not currently ready for transfer. The user is free to manipulate this BD or its associated data buffer. When it is not in auto buffer mode, the RISC controller clears this bit after the buffer has been transferred (or after an error condition is encountered).

1 = The data buffers have been prepared for transfer by the user. (Note that only one data buffer needs to be prepared if the source/destination is a peripheral device.) It may be only the source data buffer when the destination is a device or the destination data buffer when the source is a device. No fields of this BD may be written by the user once this bit is set.

NOTE

The only difference between auto buffer mode and buffer chaining mode is that the V-bit is not cleared by the RISC controller in the auto buffer mode. Auto buffer mode is enabled by the CM bit.

If the IDMA has obtained the IMB and is also waiting to obtain the external bus, but the external bus master performs an access to a location internal to the QUICC, the IDMA will relinquish the IMB and retry the cycle once it has obtained the IMB.

7.6.4.8 ENDING THE IDMA TRANSFER. If no bus exceptions occur, the IDMA eventually finishes the transfer of a block of data. These paragraphs describe normal termination in more detail. (Termination by error is discussed in 7.6.4.7.2 Bus Error.)

The IDMA channel operation experiences normal termination when the BCR is decremented to zero or the external device signals a termination of the transfer using $\overline{\text{DONE}}$x. These terminations are independent of how requests are generated to the IDMA.

Additionally, the user may stop the IDMA channel by clearing STR. However, this is considered a suspension of activity, rather than normal termination, since the transfer resumes when STR is set once again.

The user may also terminate the transfer by setting the RST bit in the CMR; however, this is not a normal termination of IDMA activity.

Further description of normal termination depends on the mode of the IDMA: single buffer mode, auto buffer mode, and buffer chaining. These modes are described in the following paragraphs.

7.6.4.8.1 Single Buffer Mode Termination. The following methods may be used to terminate an IDMA transfer in the single buffer mode. They may also be used to terminate a current BD transfer in the auto buffer and buffer chaining modes.

Transfer Count Exhausted. When the channel performs an operand transfer, it decrements the BCR for each byte transferred successfully. When the BCR is decremented to zero, the transfer is terminated. When the last bus cycle of the transfer occurs (either a byte, word, or long-word access), $\overline{\text{DONE}}$x is asserted during that bus cycle. If the device is the source, further destination accesses will take place after $\overline{\text{DONE}}$x is asserted. If the device is the destination, $\overline{\text{DONE}}$x will be asserted on the final bus cycle of the destination write.

NOTE

This behavior of $\overline{\text{DONE}}$x also applies to memory-to-memory transfers. $\overline{\text{DONE}}$x is asserted on either the last source or destination bus cycle, as determined by the ECO bit in the CMR.

When the operand transfer has completed and the BCR has been decremented to zero, the channel operation is terminated, STR is cleared, and a DONE bit interrupt is generated if the corresponding CMAR bit is set. The SAPR and/or DAPR are also incremented in the normal fashion.

NOTE

If the channel was started with the BCR value set to zero, the channel will transfer 4 Gbytes before the transfer count is exhausted.

For synchronous communication, the internal clock is identical to the baud rate output. To get the desired rate, the user can select the appropriate system clock according to the following equation:

sync baud rate = (BRGCLK or CLK2 or CLK6) ÷ (clock divider + 1) ÷ (1 or 16 according to the DIV16 bit)

For example, to get the rate of 64 kbps, the system clock can be 24.96 MHz, DIV16 = 0, and the clock divider = 389.

7.10 SERIAL COMMUNICATION CONTROLLERS (SCCS)

The SCC key features are as follows:

- Implements HDLC/SDLC, HDLC Bus, BISYNC, Synchronous Start/Stop, Asynchronous Start/Stop (UART), AppleTalk (LocalTalk), and Totally Transparent Protocols
- Ethernet Version of QUICC Supports Full 10 Mbps Ethernet/IEEE 802.3 on SCC1
- Additional Protocols Supported Through Motorola-Supplied RAM Microcodes: Profibus, Signaling System#7 (SS7), Async HDLC, DDCMP, V.14, and X.21 (see Appendix C RISC Microcode from RAM).
- 2 Mbps HDLC, HDLC Bus, and/or Transparent Data Rates Supported on All Four SCCs Simultaneously (Full Duplex).
- 10 Mbps Ethernet (Half Duplex) on SCC1 and 2 Mbps on the Other SCCs Supported Simultaneously (Full Duplex)
- A Single HDLC or Transparent Channel Can Be Supported at 8 Mbps (Full Duplex)
- SCC Clocking Rates up to 12.5 MHz at 25 MHz.
- DPLL Circuitry for Clock Recovery with NRZ, NRZI, FM0, FM1, Manchester, and Differential Manchester (Also Known as Differential Biphase-L)
- SCC Clocks May Be Derived from a Baud Rate Generator, an External Pin, or DPLL. Data Clock May Be as High as 3.125 MHz with a 25-MHz Clock
- Supports Automatic Control of the $\overline{\text{RTS}}$, $\overline{\text{CTS}}$, and $\overline{\text{CD}}$ Modem Signals
- Multibuffer Data Structure for Receive and Transmit (up to 224 BDs May Be Partitioned in Any Way Desired)
- Deep FIFOs (SCC1 Has 32-Byte Rx and Tx FIFOs; SCC2, SCC3, and SCC4 Have 16-Byte Rx and Tx FIFOs)
- Transmit-On-Demand Feature Decreases Time to Frame Transmission
- Low FIFO Latency Option for Transmit and Receive in Character-Oriented and Totally Transparent Protocols
- Frame Preamble Options
- Full-Duplex Operation
- Fully Transparent Option for Receiver/Transmitter While Another Protocol Executes on the Transmitter/Receiver
- Echo and Local Loopback Modes for Testing

7.10.18.1 HDLC BUS KEY FEATURES. The HDLC bus controller contains the following key features:

- Superset of the HDLC Controller Features
- Automatic HDLC Bus Access
- Automatic Retransmission in Case of a Collision
- May Be Used with the NMSI Mode or a TDM Bus
- Delayed $\overline{\text{RTS}}$ Mode

7.10.18.2 HDLC BUS OPERATION. The following paragraphs detail the operation of the HDLC bus Controller.

7.10.18.2.1 Accessing the HDLC Bus. HDLC bus ensures an orderly access to the bus when two or more transmitters attempt to access the bus simultaneously. In such a case, one transmitter will always be successful in completing its transmission. This procedure relies upon the use of HDLC flags consisting of the binary pattern 01111110 ($7E) and the use of the zero bit insertion to prevent flag imitation.

While in the active condition (desiring to transmit), the HDLC bus controller will monitor the bus through the $\overline{\text{CTS}}$ pin. It counts the number of one bits using the $\overline{\text{CTS}}$ pin, and if a zero is detected, the internal counter is cleared.

Once 8 consecutive ones have been received, the HDLC bus controller will begin transmission on the line. While it is transmitting information on the bus, the transmitted data is continuously compared with the data actually on the bus. The $\overline{\text{CTS}}$ pin is used to sample the external bus.

Figure 7-56 shows how the $\overline{\text{CTS}}$ pin is used. The $\overline{\text{CTS}}$ sample is taken halfway through the bit time, using the rising edge of the transmit clock. If the transmitted bit is the same as the received $\overline{\text{CTS}}$ sample, the HDLC bus controller continues its transmission. If, however, the received CTS bit is zero, but the transmitted bit was 1, the HDLC controller ceases transmission following that bit and returns to the active condition. Since the HDLC bus uses a wired-OR scheme, a transmitted zero has priority over a transmitted one.

If the source address is included in the HDLC frame in addition to the destination address, a predefined priority of nodes will result. In addition, the inclusion of a source address will allow collisions to be detected no later than the end of the source address.

NOTE

HDLC bus can be used with many different HDLC-based frame formats. HDLC bus does not specify the type of HDLC protocol used.

However, the R-bit will be cleared if an error occurs during transmission, regardless of the CM bit.

BR—BCS Reset

0 = The transmitter BCS accumulation is not reset.
1 = The transmitter BCS accumulation is reset (used for STX or SOH) before sending the data buffer.

TD—Transmit DLE

0 = No automatic DLE transmission is to occur before the data buffer.
1 = The transmitter will transmit a DLE character before sending the data buffer, which saves writing the first DLE to a separate data buffer when working in transparent mode. See the TR bit for information on control characters.

TR—Transparent Mode

0 = The transmitter will enter (remain in) the normal mode after sending the data buffer. In this mode, the transmitter will automatically insert SYNCs in an underrun condition.
1 = The transmitter enters or remains in transparent mode after sending the data buffer. In this mode, the transmitter automatically inserts DLE-SYNC pairs in the underrun condition. Underrun occurs when the BISYNC controller finishes a buffer with the L-bit set to zero and the next BD is not available. The transmitter also checks all characters before sending them; if a DLE is detected, another DLE is automatically sent. The user must insert a DLE or program the BISYNC controller to insert it (using TD) before each control character required. The transmitter will calculate the CRC16 BCS even if the BCS bit in the BISYNC mode register is programmed to LRC. The PTCRC should be initialized to CRC16 before setting this bit.

B—BCS Enable

0 = Buffer consists of characters to be excluded from the BCS accumulation.
1 = Buffer consists of characters to be included in the BCS accumulation.

The following status bits are written by the CP after it has finished transmitting the associated data buffer.

UN—Underrun

The BISYNC controller encountered a transmitter underrun condition while transmitting the associated data buffer.

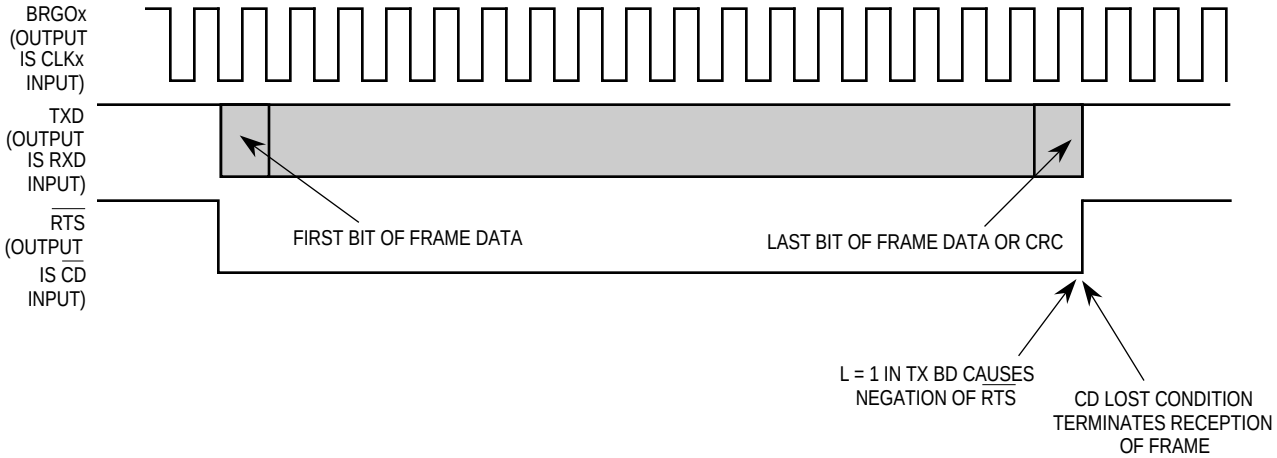
CT—CTS Lost

CTS was lost during message transmission.

Data Length

The data length is the number of octets that the CP should transmit from this BD's data buffer. It is never modified by the CP. The data length should be greater than zero.

NOTES:
1. Each QUICC generates its own transmit clocks. If the transmit and receive clocks are the same, it is possible for one QUICC to generate transmit and receive clocks for the other QUICC (for example, CLKx on QUICC 2 could be used to clock the transmitter and receiver).



NOTES:
1. CTS should be configured as always asserted in the port C parallel I/O or else connected to ground externally.
2. The required GSMR configurations are: DIAG = 00, CTSS = 1, CTSP is a don't care, CDS = 1, CDP = 0, TTX = 1, and TRX = 1. REVD and TCRC are application dependent.
3. The transparent frame will contain a CRC if the TC bit is set in the Tx BD.

Figure 7-64. Sending Transparent Frames Between QUICCs

QUICC1 and QUICC2 exchange transparent frames and synchronize each other using the RTS and CD pins. The CTS pin is not required since transmission may begin at any time. Thus, the RTS signal is directly connected to the other QUICC's CD pin. The RSYN option in GSMR is not used, and transmission and reception from each QUICC are independent.

7.10.21.5 TRANSPARENT MEMORY MAP. When configured to operate in transparent mode, the QUICC overlays the structure listed in Table 7-5 onto the protocol-specific area of the SCC parameter RAM listed in Table 7-10.

Table 7-10. Transparent-Specific Parameters

Address	Name	Width	Description
SCC Base + 30	**CRC_P**	Long	CRC Preset for Totally Transparent
SCC Base + 34	**CRC_C**	Long	CRC Constant for Totally Transparent Receiver

NOTE: The boldfaced items should be initialized by the user.

BA23–BA13 that was set in A23–A13, and clear BA31–BA24 and BA12–BA11.

The three function code (FC2–FC0) bits become the four function code (FC3–FC0) bits of the QUICC ORx.

The MC68302 ORx register most closely corresponds to the QUICC ORx.

The CFC bit is implemented as the FCM3–FCM0 bits on the QUICC ORx. This gives more flexibility in determining the function codes that cause the chip select to activate. If the MC68302 CFC bit was cleared, then clear FCM3–FCM0 on the QUICC. Also, to match MC68302 behavior, clear the NCS bit in the QUICC GMR.

The MRW bit in the ORx and the RW bit in the BRx of the MC68302 simply become the WP bit on the QUICC BRx. On the QUICC, the choice exists for asserting the chip select for reads and writes (WP = 0) or just reads (WP = 1).

The MC68302 base address mask bits A23–A13 become the AM27–AM11 bits in the QUICC ORx. Note that this allows both larger and smaller block sizes than what the MC68302 provides. To transfer a block range, take bits A23–A13 of the MC68302 and write them to AM23–AM13. Then set AM32–AM24 and clear AM12–AM11.

The three MC68302 DTACK bits become the four TCYC3–TCYC0 bits of the QUICC ORx. Note that the maximum number of wait states is increased from 6 to 15 on the QUICC.

TIMERS

The MC68302 contains two general-purpose timers. The QUICC contains four general-purpose timers, which are the same as the MC68302 timers, but slightly enhanced. The QUICC also contains a timer global configuration register (TGCR), which allows all four timers to be enabled simultaneously and allows the timers to be internally cascaded into 32-bit timers.

The MC68302 TMRx register most closely corresponds to the QUICC TMRx.

The RST bit is now located in the QUICC TGCR.

The ICLK bits are still in the same location of the QUICC TMR. The bit encodings are the same except for 00 combination, which is now implemented by the EN bit in the QUICC TGCR.

The FRR bit is still in the same location of the QUICC TMR.

The ORI bit is still in the same location of the QUICC TMR.

The OM bit is still in the same location of the QUICC TMR, but the meaning of OM = 0 mode can be different. The active-low pulse can be longer than on the MC68302, depending on the frequency of the input clock source.

The CE bits are still in the same location of the QUICC TMR.

The PS bits are still in the same location of the QUICC TMR.

The MC68302 TRRx register is the same as the QUICC TRRx.

The MC68302 TCRx register is the same as the QUICC TCRx.

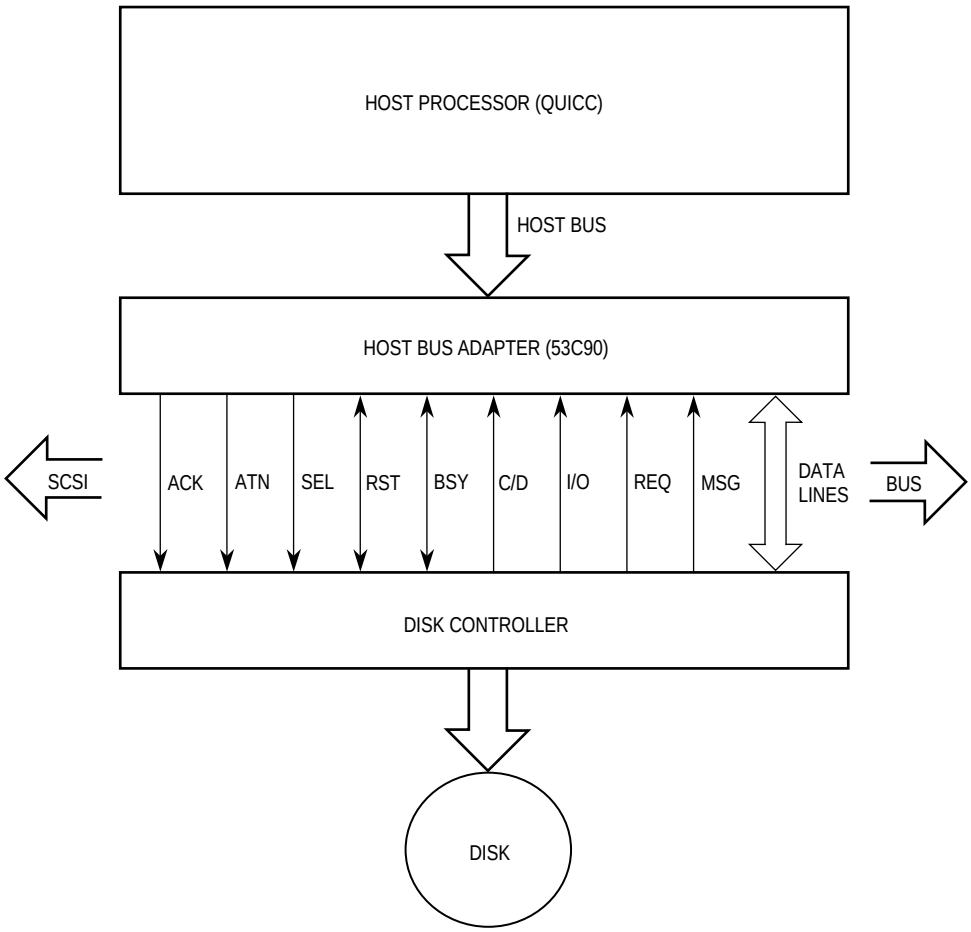


Figure 9-17. SCSI Bus Signals

Table 9-1. SCSI Bus Signals

Signal	Driven By	Signal Explanation
DB0–DB7	Initiator/Target	8-Bit Bidirectional Data Bus.
DBP	Initiator/Target	Data-Bus Parity Line. Optional.
ATN	Initiator	Attention. Used to send a message to the target when it controls the bus.
BSY	Initiator/Target	Busy. Indicates that the bus is unavailable for use.
ACK	Initiator	Acknowledge. Used by the initiator for handshaking.
RST	Any Device	Reset. Used to initiate a bus-free phase.
MSG	Target	Driven by the target to indicate that the current transfer is a message.
SEL	Initiator	Select. Used by the initiator to select a target before command execution. Also used by the target to reconnect when the reselection phase is implemented.
C/D	Target	Control/Data. Used during the information transfer phases to transfer commands, status, data or messages over the bus.
REQ	Target	Request. Used by the target during information transfer phases.
I/O	Target	Input/Output. Determines the direction of the transfer.

The SCSI bus has a total of 18 signals—9 are used for control, and 9 are used for data (one parity bit). The bidirectional data lines transfer data, commands, status, and message infor-

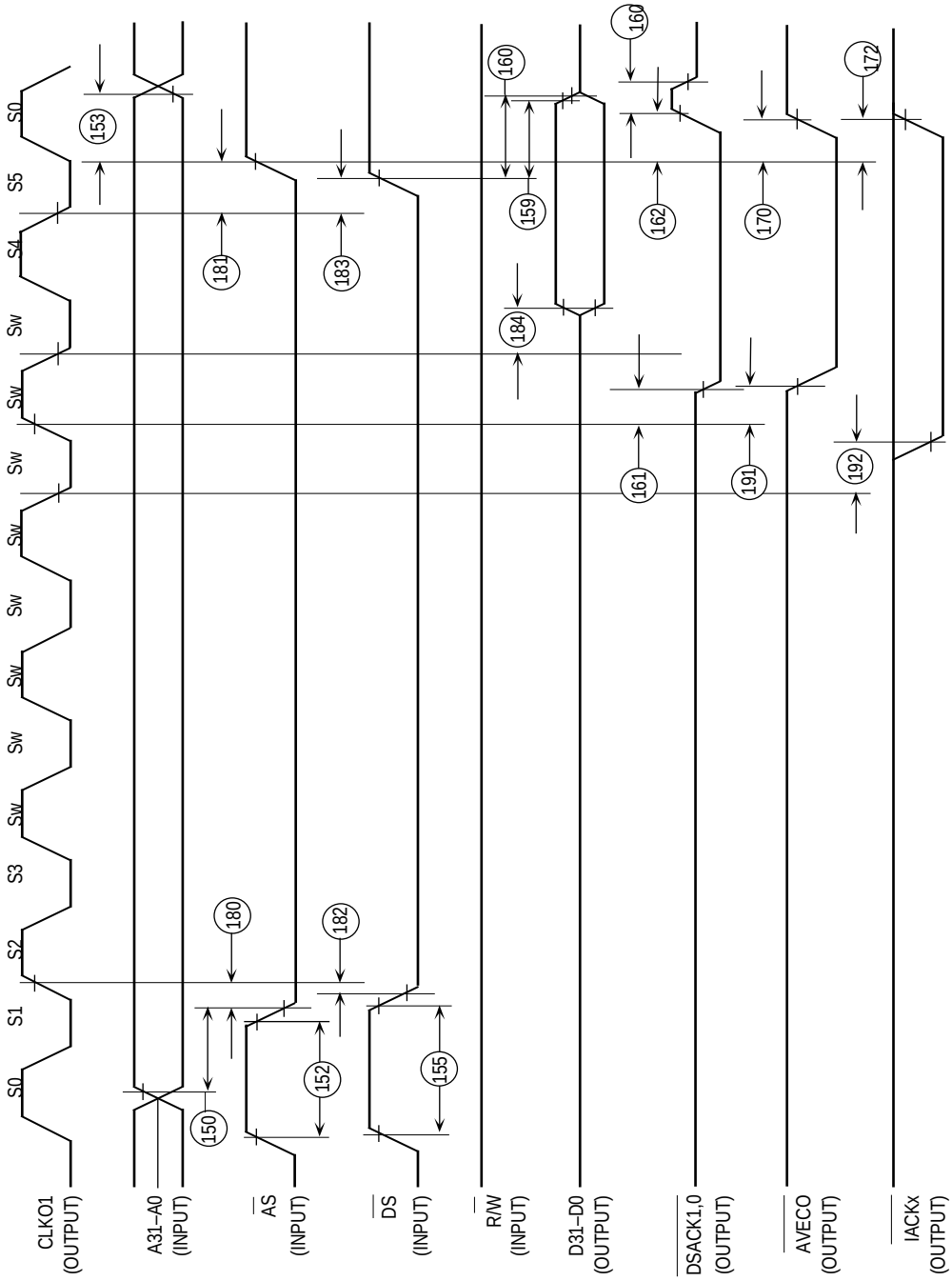


Figure 10-32. External MC68030/MC68360 Synchronous IACK Cycle Timing Diagram

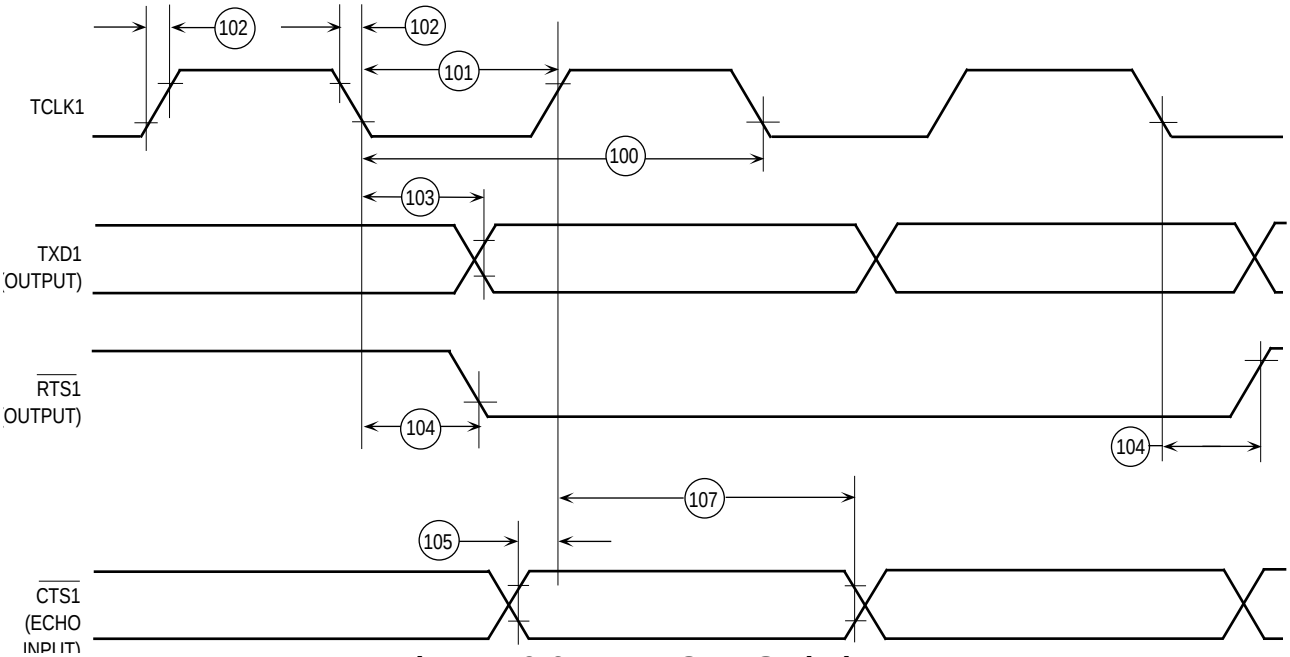


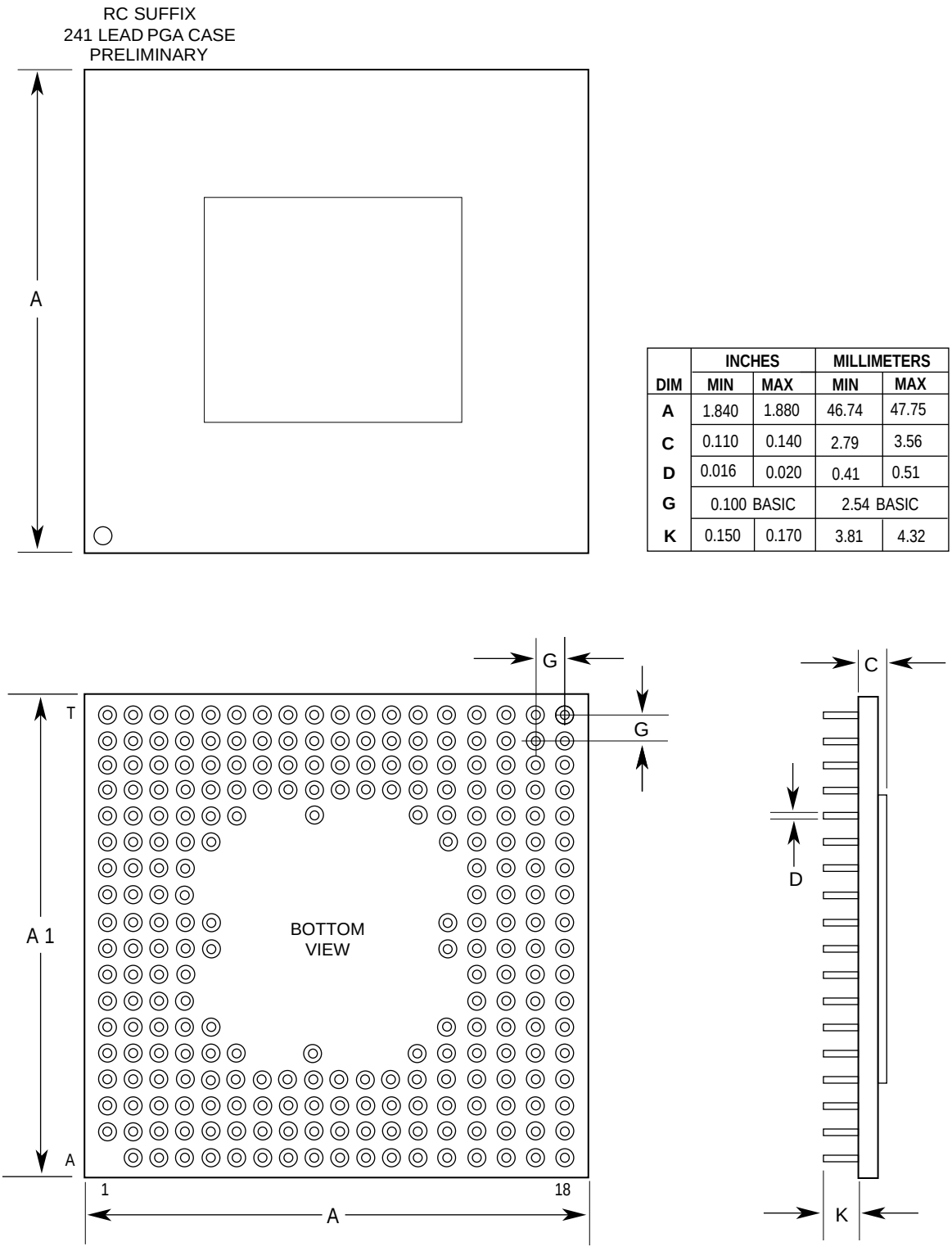
Figure 10-67. HDLC BUS Timing

10.26 ETHERNET ELECTRICAL SPECIFICATIONS

(The electrical specifications in this document are preliminary. See Figure 10-68–Figure 10-73)

Num.	Characteristic	3.3 V or 5.0 V 25.0 MHz		5.0 V 33.34 MHz		Unit
		Min	Max	Min	Max	
120	CLSN Width High	40	—	40	—	ns
121	RCLK1 Rise/Fall Time	—	15	—	15	ns
122	RCLK1 Width Low	CLKO1+ 5ns	—	CLKO1+ 5ns	—	
123[1]	RCLK1 Width high	CLKO1	—	CLKO1	—	
124	RXD1 Setup Time	20	—	20	—	ns
125	RXD1 Hold Time	5	—	5	—	ns
126	RENA Active Delay (from RCLK1 rising edge of the last data bit)	10	—	10	—	ns
127	RENA Width Low	100	—	100	—	ns
128	TCLK1 Rise/Fall Time	—	15	—	15	ns
129	TCLK1 Width Low	CLKO1+ 5ns	—	CLKO1+ 5ns	—	
130[1]	TCLK1 Width high	CLKO1	—	CLKO1	—	
131	TXD1 Active Delay (from TCLK1 rising edge)	10	50	10	50	ns
132	TXD1 Inactive Delay (from TCLK1 rising edge)	10	50	10	50	ns
133	TENA Active Delay (from TCLK1 rising edge)	10	50	10	50	ns
134	TENA Inactive Delay (from TCLK1 rising edge)	10	50	10	50	ns
135	RSTRT Active Delay (from TCLK1 falling edge)	10	50	10	50	ns
136	RSTRT Inactive Delay (from TCLK1 falling edge)	10	50	10	50	ns
137	RRJCT Width Low	1	—	1	—	CLKO1

11.6 PACKAGE DIMENSIONS—PGA (RC SUFFIX)



DIM	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	1.840	1.880	46.74	47.75
C	0.110	0.140	2.79	3.56
D	0.016	0.020	0.41	0.51
G	0.100 BASIC		2.54 BASIC	
K	0.150	0.170	3.81	4.32

D.2.8 Development Support

No new development tools will be needed for the QUICC32.

Only the replacement of actual silicon is needed. All Motorola and third party support tools will work as before. Motorola will provide simple device drivers for developers to have an easy start on the software.

D.2.9 Ordering Information

The following table identifies the packages and operating frequencies available for the MC68MH360.

Table D-1. MC68MH360 Package/Frequency Availability

Package Type	V_{CC}	Frequency (MHz)	Temperature	Order Number
Quad Flat Pack (FE Suffix) Quad Flat Pack (CFE Suffix)	5 V	0–25 0–25	0°C to 70°C –40°C to +85°C	MC68MH360FE25 MC68MH360CFE25
Pin Grid Array (RC Suffix) Pin Grid Array (CRC Suffix)	5 V	0–25 0–25	0°C to 70°C –40°C to +85°C	MC68MH360RC25 MC68MH360CRC25
Quad Flat Pack (FE Suffix	5 V	0–33	0°C to 70°C	MC68MH360FE33
Pin Grid Array (RC Suffix)	5 V	0–33	0°C to 70°C	MC68MH360RC33
Quad Flat Pack (FE Suffix	3.3 V	0–25	0°C to 70°C	MC68MH360FE25V
Pin Grid Array (RC Suffix)	3.3 V	0–25	0°C to 70°C	MC68MH360RC25V

The documents listed in the following table contain detailed information on the MC68360. These documents may be obtained from the Literature Distribution Centers at the addresses listed at the bottom of this page.

Table D-2. Documentation

Document Title	Order Number	Contents
MC68MH360 Specification Addendum	Contact local field sales office	Preliminary design information
M68000 Family Programmer's Reference Manual	M68000PM/AD	M68000 Family Instruction Set
The 68K Source	BR729/D	Independent vendor listing supporting software and development tools