NXP USA Inc. - KMC68360AI33L Datasheet

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	240-BFQFP
Supplier Device Package	240-FQFP (32x32)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/kmc68360ai33l

BCLRO—This active-low open-drain output indicates that one of the QUICC internal bus masters is requesting the external bus master to release the bus.

CONFIG1—See 2.1.13 Initial Configuration Pins (CONFIG) for the description.

RAS2—See 2.1.5.1 Chip Select/Row Address Select (CS6–CS0/RAS6–RAS0) for the description.

2.1.9 System Control Signals

The QUICC uses these signals to recover from an exception. Refer to Section 4 Bus Operation for more information on these signals.

2.1.9.1 SOFT RESET (RESETS). This active-low, open-drain, bidirectional signal is used to initiate reset. An external reset signal (as well as a reset from the SIM60) resets the QUICC as well as all external devices. A reset signal from the CPU32+ (asserted as part of the RESET instruction) resets external devices only—the internal state of the CPU32+ is not affected; other on-chip modules are reset, but the configuration is not altered. When asserted by the QUICC, this signal is guaranteed to be asserted for a minimum of 512 clock cycles. For more information see 4.7 Reset Operation.

2.1.9.2 HARD RESET (RESETH). This active-low, open-drain, bidirectional signal is used to initiate reset. An external hard reset signal (as well as an hard reset from the SIM60) resets the QUICC as well as all external devices and the internal state of the CPU32+; other on-chip modules are reset as well as the QUICC configuration. When asserted by the QUICC, this signal is guaranteed to be asserted for a minimum of 512 clock cycles. For more information see 4.7 Reset Operation.

During a hard reset, the address, data, and bus control pins are all three-stated. The BG pin output is the same as that on the BR input. The general-purpose I/O pins are all configured as inputs. The NC4–NC1 pins are undefined outputs. The XTAL, CLKO1, and CLKO2 pins are active outputs, except for CLKO1 which does not oscillate while the on-chip PLL is attaining a lock. The RESETS pin is an output.

2.1.9.3 HALT (HALT). This active-low, open-drain, bidirectional signal is asserted to suspend external bus activity, to request a retry when used with BERR, or to perform a single-step operation. As an output, HALT indicates a double bus fault by the CPU32+.

2.1.9.4 BUS ERROR (BERR). This active-low, open-drain, bidirectional signal indicates that an invalid bus operation is being attempted or, when used with HALT, that the bus master should retry the current cycle.

2.1.10 Clock Signals

These signals are used by the QUICC for controlling or generating the system clocks. Refer to Section 6 System Integration Module (SIM60) for more information on these clock signals.

2.1.10.1 SYSTEM CLOCK OUTPUTS (CLKO2–CLKO1). These output signals reflect the general system clock and are used as the bus timing reference by external devices. CLKO1

When the CPU32+ acknowledges hardware breakpoint ($\overline{\text{BKPT}}$ pin assertion or internal breakpoint logic) with background mode disabled, the CPU32+ performs a word read from CPU space, type 0, at an address corresponding to all ones on A4–A2 (BKPT#7), and the T-bit (A1) is set. If this bus cycle is terminated by $\overline{\text{BERR}}$, the QUICC performs hardware breakpoint exception processing. If this bus cycle is terminated by $\overline{\text{DSACKx}}$, the QUICC ignores data on the data bus and continues execution of the next instruction.

NOTE

> The $\overline{\text{BKPT}}$ pin is sampled on the same clock phase as data and is latched with data as it enters the CPU32+ pipeline. If $\overline{\text{BKPT}}$ is asserted for only one bus cycle and a pipeline flush occurs before $\overline{\text{BKPT}}$ is detected by the CPU32+, $\overline{\text{BKPT}}$ is ignored. To ensure detection of $\overline{\text{BKPT}}$ by the CPU32+, $\overline{\text{BKPT}}$ can be asserted until a breakpoint acknowledge cycle is recognized.
>
> When the QUICC is configured for a 32-bit bus, the CPU32+ can fetch two instructions simultaneously. Since there is only one $\overline{\text{BKPT}}$ pin, the external user cannot break individually on those instructions, but rather must break on both, causing the BKPT exception to be taken after the first instruction and before the second instruction. The internal breakpoint logic, however, can individually assert a breakpoint for either instruction. (See the BKAR and BKCR discussion in Section 6 System Integration Module (SIM60) for details).

The breakpoint operation flowchart is shown in Figure 4-23. Figure 4-24 and Figure 4-25 show the timing diagrams for the breakpoint acknowledge cycle with instruction opcodes supplied on the cycle and with an exception signaled, respectively.

aligned on their natural boundaries. All instruction words and extension words must reside on word boundaries. Attempting to prefetch an instruction word at an odd address causes an address error exception.

The CPU32+ has four bits (SZ1, SZ0 and SZC1, SCZ0) in the software status word (SSW) that are new or have changed definitions.

The CPU32+ offers low power consumption. The CPU32+ is implemented in high-speed complementary metal-oxide semiconductor (HCMOS) technology, providing low power use during normal operation. During periods of inactivity, the low-power stop (LPSTOP) instruction can be executed, shutting down the CPU32+ and other IMB modules, greatly reducing power consumption.

Ease of programming is an important consideration when using an integrated processor. The CPU32+ instruction format reflects a predominant register-memory interaction philosophy. All data resources are available to operations that require them. The programming model includes eight multifunction data registers and seven general-purpose addressing registers. The data registers support 8-bit (byte), 16-bit (word), and 32-bit (long-word) operand lengths for all operations. Address manipulation is supported by word and long-word operations. Although the program counter (PC) and stack pointers (SP) are special-purpose registers, they are also available for most data addressing activities. Ease of program checking and diagnosis is enhanced by trace and trap capabilities at the instruction level.

As processor applications become more complex and programs become larger, high-level languages (HLLs) become the system designer's choice in programming languages. HLLs aid in the rapid development of complex algorithms with less error and are readily portable. The CPU32+ instruction set efficiently support HLLs.

5.1.1 Features

Features of the CPU32+ are as follows:

- Fully Upward Object-Code Compatible with M68000 Family
- Loop Mode of Instruction Execution
- Fast Multiply, Divide, and Shift Instructions
- Fast Bus Interface with Dynamic Bus Port Sizing
- Improved Exception Handling
- Additional Addressing Modes

  —Scaled Index
  —Address Register Indirect with Base Displacement and Index
  —Expanded PC Relative Modes
  —32-Bit Branch Displacements

- Instruction Set Additions

  —High-Precision Multiply and Divide
  —Trap on Condition Codes
  —Upper and Lower Bounds Checking

5.3.4.2 TABLE EXAMPLE 2: COMPRESSED TABLE. In Example 2 (see Figure 5-8), the data from Example 1 has been compressed by limiting the maximum value of the independent variable. Instead of the range 0 ≤ X = 65535, X is limited to 0 ≤ X ≤ 1023. The table has been compressed to only five entries, but up to 256 levels of interpolation are allowed between entries.

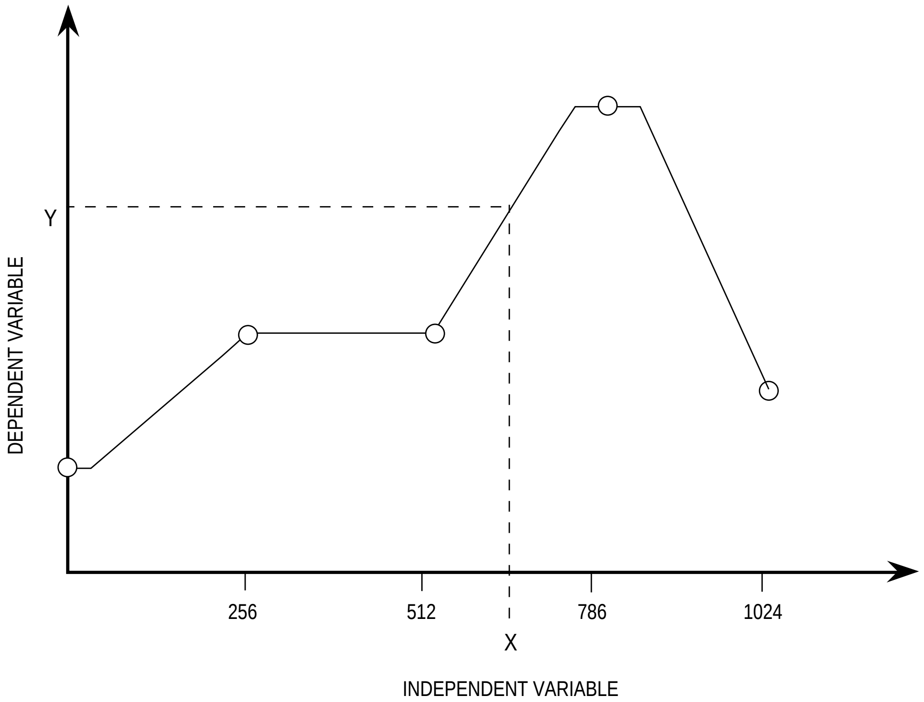


Figure 5-8. Table Example 2

NOTE

Extreme table compression with many levels of interpolation is possible only with highly linear functions. The table entries within the range of interest are listed in Table 5-14.

Table 5-14. Compressed Table Entries

Entry Number	X-Value	Y-Value
2	512	1311
3	786	1966

Since the table is reduced from 257 to 5 entries, independent variable X must be scaled appropriately. In this case the scaling factor is 64, and the scaling is done by a single instruction:

```
LSR.W #6,Dx
```

Bit 5—PWW

This read-only bit is used to indicate if the $\overline{\text{WE}}$/ADDR and the PRTY lines have been programmed by the user or are still in the three-state condition because the PEPAR register has not been written.

0 = PEPAR has not been written. The $\overline{\text{WE}}$/ADDR and the PRTY lines are still being three-stated.

1 = PEPAR was written. The $\overline{\text{WE}}$/ADDR and the PRTY lines have been programmed in the PEPAR, so the configuration choices of these pins in the PEPAR are valid.

Bit 4—$\overline{\text{CAS2}}$, $\overline{\text{CAS3}}$/$\overline{\text{IACK3}}$, $\overline{\text{IACK6}}$

0 = The $\overline{\text{CAS2}}$ and $\overline{\text{CAS3}}$ output functions are selected.

1 = The $\overline{\text{IACK3}}$ and $\overline{\text{IACK6}}$ output functions are selected.

Bit 2—$\overline{\text{CAS0}}$, $\overline{\text{CAS1}}$/$\overline{\text{IACK1}}$, $\overline{\text{IACK2}}$

0 = The $\overline{\text{CAS0}}$ and $\overline{\text{CAS1}}$ output functions are selected.

1 = The $\overline{\text{IACK1}}$ and $\overline{\text{IACK2}}$ output functions are selected.

Bit 1—$\overline{\text{CS7}}$/$\overline{\text{IACK7}}$

0 = The $\overline{\text{CS7}}$ output function is selected.

1 = The $\overline{\text{IACK7}}$ output function is selected.

Bit 0—AVEC ($\overline{\text{AVECO}}$)/$\overline{\text{IACK5}}$

0 = The $\overline{\text{AVEC}}$ input function is selected in normal operation, or $\overline{\text{AVECO}}$ is selected in slave mode.

1 = The $\overline{\text{IACK5}}$ output function is selected.

6.10 MEMORY CONTROLLER

The memory controller is a sub-block of the SIM60 that is responsible for up to eight general-purpose chip-select lines and the DRAM controller. The DRAM controller itself can control up to eight memory banks.

6.10.1 Memory Controller Key Features

The key features of the memory controller are as follows:

- All Eight Memory Banks Support the Following:

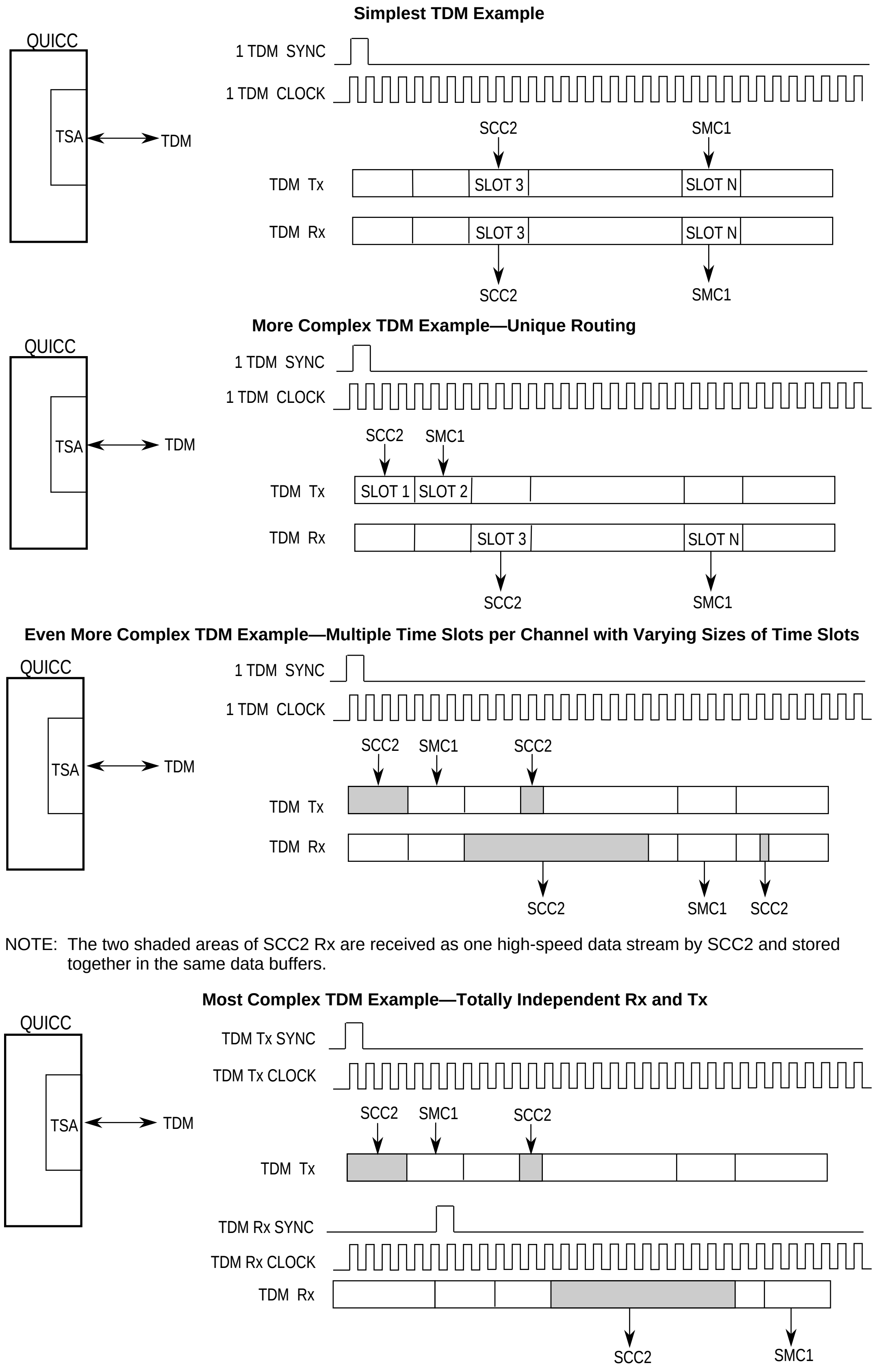
—32-Bit Address Decode with 17 Bits of Address Masking
—Various Block Sizes—2 Kbytes up to 256 Mbytes
—From 0 to 15 Wait States Programmable with $\overline{\text{DSACK}}$ Generation
—Memory Bank Can Be Used by an External Master
—Supports Burst Accesses of the MC68040
—Byte Parity Generation/Checking
—Write-Protect Capability
—Four Byte-Write Enable ($\overline{\text{WE}}$) Signals
—Output Enable ($\overline{\text{OE}}$) Signal
—Special Options for Interfacing to Slow Peripherals
—Function Code Match with Mask Can Qualify Memory Bank Accesses

7.8.2 TSA Overview

The TSA implements both the internal route selection and time-division multiplexing for multiplexed serial channels. The TSA supports the serial bus rate and format for most standard TDM buses, including the T1 and CEPT highways, the PCM highway, and the ISDN buses in both basic and primary rates. The two popular ISDN basic rate buses, IDL and GCI (also known as IOM-2), are supported. An additional level of flexibility is provided by the TSA in that it supports two TDMs. It is therefore possible to simultaneously support one T1 line and one CEPT line, one basic rate and one primary rate ISDN channel, etc.

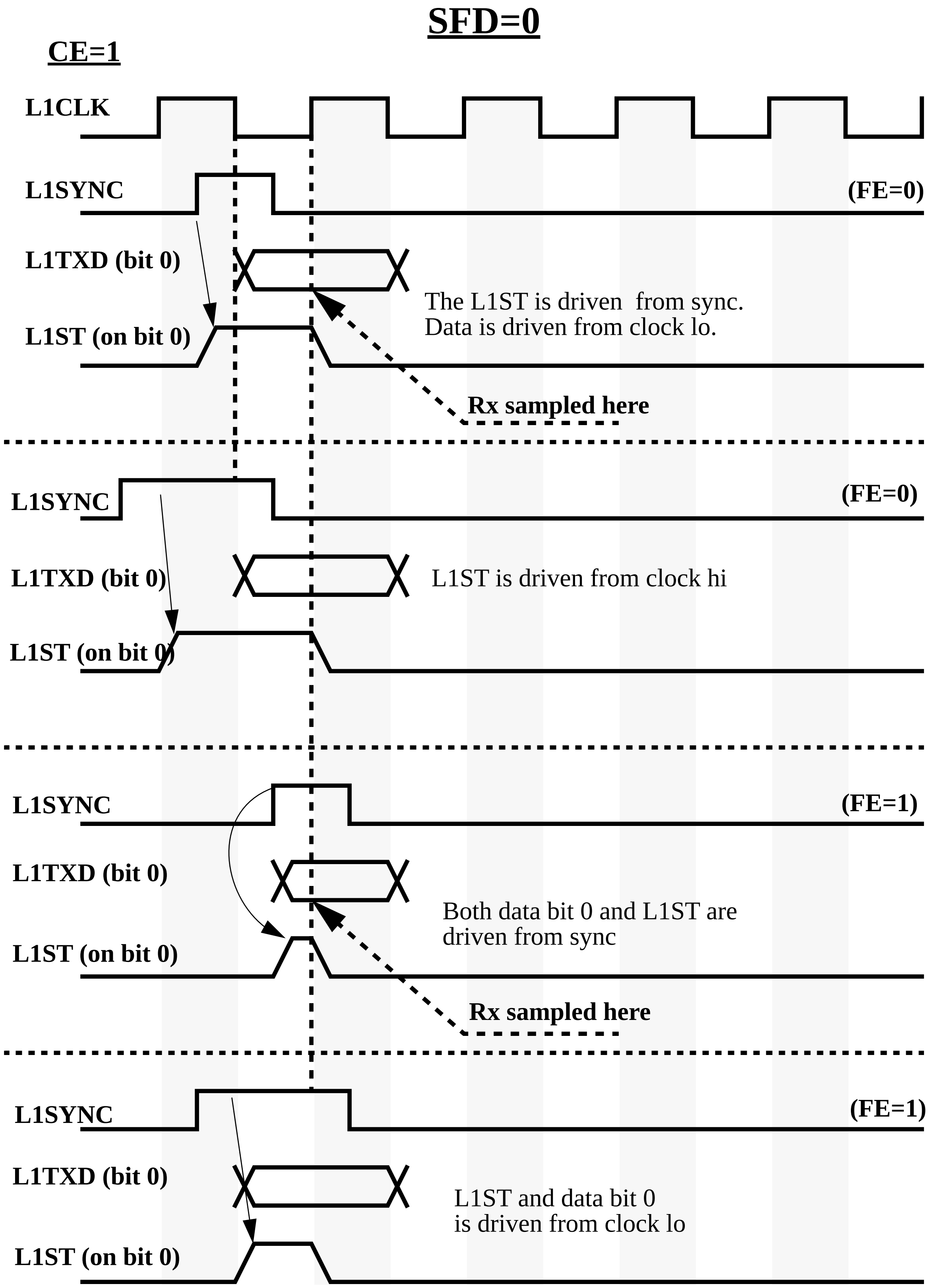
TSA programming is completely independent of the protocol used by the SCC or SMC. For instance, the fact that SCC2 may programmed for the HDLC protocol has no impact on the programming of the TSA. The purpose of the TSA is to route the data from the specified pins to the desired SCC or SMC at the correct time. It is the job of the SCC or SMC to handle the data it receives.

In its simplest mode, the TSA identifies the frame using one sync pulse and one clock signal provided externally by the user. This can be enhanced to allow independent routing of the receive and transmit data on the TDM. Additionally, the definition of a time slot need not be limited to 8 bits or even limited to a single contiguous position within the frame. Finally, the user may provide separate receive and transmit syncs as well as receive and transmit clocks. These various configurations are illustrated in Figure 7-20.



NOTE: The two shaded areas of SCC2 Rx are received as one high-speed data stream by SCC2 and stored together in the same data buffers.

Figure 7-20. Various Configurations of a Single TDM Channel


SFD=0
CE=1
L1CLK
L1SYNC
(FE=0)
L1TXD (bit 0)
L1ST (on bit 0)
The L1ST is driven from sync.
Data is driven from clock lo.
Rx sampled here
L1SYNC
(FE=0)
L1TXD (bit 0)
L1ST is driven from clock hi
L1ST (on bit 0)
L1SYNC
(FE=1)
L1TXD (bit 0)
Both data bit 0 and L1ST are
driven from sync
L1ST (on bit 0)
Rx sampled here
L1SYNC
(FE=1)
L1TXD (bit 0)
L1ST and data bit 0
is driven from clock lo
L1ST (on bit 0)

clock for multiple BRGs. The clock signals on the CLK2 and CLK6 pins are not synchronized internally prior to being used by the BRG.

Next, the BRG provides a divide-by-16 option before the clock reaches the prescaler. This option is chosen by the DIV16 bit.

The clock is then divided in the prescaler by up to 4096. This input clock divide ratio can be programmed on the fly. Two on-the-fly BRG changes should not occur within a time shorter than the period of at least two BRG input clocks.

The output of the prescaler is sent internally to the bank of clocks and may also be output externally on the BRGOx pins of either the port A or port B parallel I/O. One BRGOx pin (BRGO4–BRGO1) is an output from the corresponding BRG. If the BRG divides the clock by an even value, the transitions of the BRGO pin will always occur on the falling edge of the input clock to the BRG. If the BRG is programmed to an odd value, the transitions will alternate between the falling and rising edges of the input clock.

Additionally, the output of the BRG may be sent to the autobaud control block described in the following paragraphs.

7.9.1 Autobaud Support

In the autobaud process, a UART deduces the baud rate of its received character stream by looking at the pattern received as well as the timing information of that pattern. The QUICC BRGs have a built-in autobaud control function that automatically measures the length of a start bit and modifies the baud rate accordingly. (This capability was only available on the MC68302 with a special microcode option.)

If the ATB bit in the BRG is set, the autobaud control block starts to search for a low level on the corresponding RXDx input line (RXD4–RXD1). When it finds a low level on the RXDx line, it assumes that this is the beginning of a start bit and begins counting the start bit length. During this time, the BRG output clock toggles for 16 BRG clock cycles at the BRG input clock rate, and then stops with the BRGO output clock in the low state.

After the RXDx line changes back to the high level, the autobaud control block rewrites the CD and DIV16 bits in the BRG configuration register to the divide ratio it found. Due to measurement error that can occur at high baud rates, this divide rate written by the autobaud controller may not be the precise, final baud rate desired by the user (e.g., 56600 could be the resulting baud rate, rather than 57600). Thus, an interrupt is provided to the user in the UART SCC event register to signify that the BRG configuration register was rewritten by the autobaud controller. On recognition of this interrupt, the user should rewrite the BRG configuration register with the desired value. The user is encouraged to do this as quickly as possible, even prior to the first character being fully received, to ensure that all characters are recognized correctly by the UART. The first data must have a transition from 1 to 0, then look for a one again. Thus the first cahracter must be an odd character.

Once a full character is received, the user may check in software to see if the received character matches a predefined value (such as "a" or "A"; it must be an odd character). Software should then check for other characters (such at "t" or "T") and program the SCC to the receiver begins receiving data. This behavior is similar to the MC68302 totally transparent mode behavior when the EXSYN bit in its SCC mode register is set.

SYNL—Sync Length (BISYNC and Transparent Mode Only)

These bits determine the operation of an SCC receiver that is configured for BISYNC or totally transparent operation only. See the data synchronization register definition in the BISYNC and totally transparent descriptions for more information.

00 = The sync pattern in the DSR is not used. An external sync signal is used instead ($\overline{\text{CD}}$ pin asserted).

01 = 4-bit sync. The receiver will synchronize on a 4-bit sync pattern stored in DSR. This character and additional syncs can be programmed to be stripped using the SYNC character in the parameter RAM. The transmitter will transmit the entire contents of the DSR prior to each frame.

10 = 8-bit sync. This option should be chosen along with the BISYNC protocol to implement mono-sync. The receiver will synchronize on an 8-bit sync pattern stored in DSR. The transmitter will transmit the entire contents of the DSR prior to each frame.

11 = 16-bit sync. Also called BISYNC. The receiver will synchronize on a 16-bit sync pattern stored in DSR. The transmitter will transmit the DSR prior to each frame.

RTSM—$\overline{\text{RTS}}$ Mode

This bit may be changed on the fly.

0 = Send idles between frames as defined by the protocol and the Tend bit. $\overline{\text{RTS}}$ is negated between frames (default).

1 = Send flags/syncs between frames according to the protocol. $\overline{\text{RTS}}$ is always asserted whenever the SCC is enabled.

RSYN—Receive Synchronization Timing (Valid for a Totally Transparent Channel Only)

0 = Normal operation.

1 = If CDS = 1, then the $\overline{\text{CD}}$ pin should be asserted on the second bit of the receive frame, rather than the first. This configuration matches the behavior of the MC68302 totally transparent receiver when its EXSYN bit is set; it is included on the QUICC for compatibility.

EDGE—Clock Edge

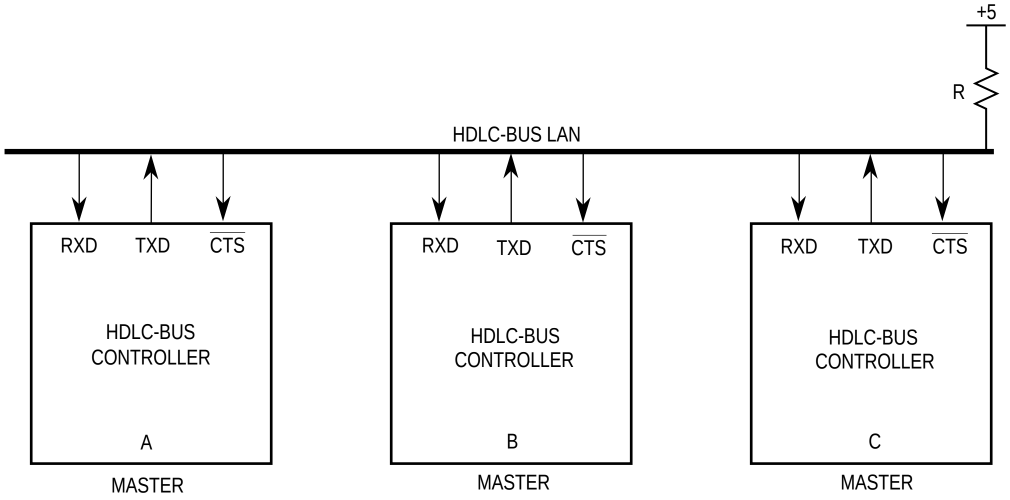
The EDGE bits determine the clock edge used by the DPLL for adjusting the receive sample point due to jitter in the received signal. The selection of the EDGE bits is ignored in the UART protocol or the x1 mode of the RDCR bits.

00 = Both the positive and negative edges are use for changing the sample point (default).

01 = Positive edge. Only the positive edge of the received signal is used for changing the sample point.

10 = Negative edge. Only the negative edge of the received signal is used for changing the sample point.

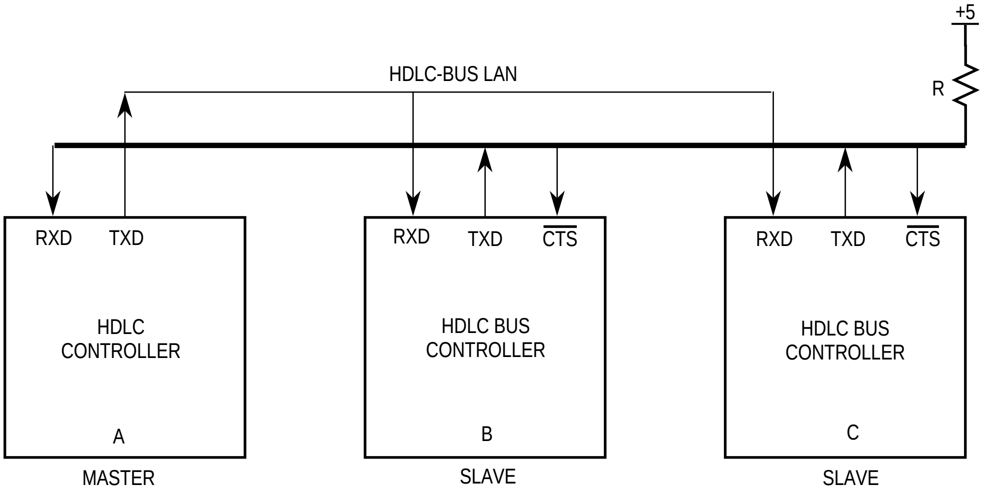
11 = No adjustment is made to the sample points.



NOTES:
1. Transceivers may be used to extend the LAN size, if necessary.
2. The TXD pins should be configured to open-drain in the port C parallel I/O port.

Figure 7-54. HDLC Bus Multi-Master Configuration

Figure 7-55 shows the other LAN-type configuration of HDLC bus. In this configuration, a master station transmits to any slave station, with no collisions possible. The slaves communicate only with the master, but may experience collisions in their access over the bus. In this configuration, a slave that must communicate with another slave must first transmit its data to the master, where the data is buffered in RAM and then retransmitted to the other slave. The benefit of this configuration, however, is that full-duplex operation may be obtained. This configuration is preferred in a point-to-multipoint environment.



NOTES:
1. Transceivers may be used to extend the LAN size, if necessary.
2. The TXD pins of slave devices should be configured to open-drain in the port C parallel I/O port.

Figure 7-55. HDLC Bus Single-Master Configuration

7.11.7.15 SMC UART MASK REGISTER (SMCM). The SMCM is referred to as the SMC UART mask register when the SMC is operating as a UART. It is an 8-bit read-write register with the same bit format as the SMC UART event register. If a bit in the SMC UART mask register is a one, the corresponding interrupt in the event register will be enabled. If the bit is zero, the corresponding interrupt in the event register will be masked. This register is cleared upon reset.

7.11.8 SMC UART Example

The following list is an initialization sequence for 9600 baud, 8 data bits, no parity, and 1 stop bit operation of an SMC UART assuming a 25-MHz system frequency. BRG1 and SMC1 are used.

1. The SDCR (SDMA configuration register) should be initialized to $0740, rather than being left at its default value of $0000.
2. Configure the port B pins to enable the SMTXD1 and SMRXD1. Write PBPAR bits 6 and 7 with ones. Write PBDIR bits 6 and 7 with zeros. Write PBODR bits 6 and 7 with zeros.
3. Configure the BRG1. Write BRGC1 with $010144. The DIV16 bit is not used, and the divider is 162 (decimal). The resulting BRG1 clock is 16x the desired bit rate of the UART.
4. Connect the BRG1 clock to SMC1 using the SI. Write the SMC1 bit in SIMODE with a 0. Write the SMC1CS bits in SIMODE with 000.
5. Write RBASE and TBASE in the SMC parameter RAM to point to the Rx BD and Tx BD in the dual-port RAM. Assuming one Rx BD at the beginning of dual-port RAM and one Tx BD following that Rx BD, write RBASE with $0000 and TBASE with $0008.
6. Program the CR to execute the INIT RX & TX PARAMS command for this channel. For instance, to execute this command for SCC1, write $0001 to the CR. This command causes the RBPTR and TBPTR parameters of the serial channel to be updated with the new values just programmed into RBASE and TBASE.
7. Write RFCR with $18 and TFCR with $18 for normal operation.
8. Write MRBLR with the maximum number of bytes per receive buffer. For this case, assume 16 bytes, so MRBLR = $0010.
9. Write MAX_IDL with $0000 in the SMC UART-specific parameter RAM to disable the MAX_IDL functionality for this example.
10. Clear BRKLN and BRKEC in the SMC UART-specific parameter RAM for the sake of clarity.
11. Set BRKCR to $0001, so that if a STOP TRANSMIT command is issued, one break character will be sent.
12. Initialize the Rx BD. Assume the Rx data buffer is at $00001000 in main memory. Write $B000 to Rx_BD_Status. Write $0000 to Rx_BD_Length (not required—done for instructional purposes only). Write $00001000 to Rx_BD_Pointer.
13. Initialize the Tx BD. Assume the Tx data buffer is at $00002000 in main memory

contains five 8-bit characters. Write $B000 to Tx_BD_Status. Write $0005 to Tx_BD_Length. Write $00002000 to Tx_BD_Pointer.

7. Write $FF to the SMCE to clear any previous events.
8. Write $13 to the SMCM to enable all possible SMC interrupts.
9. Write $00000010 to the CIMR to allow SMC1 to generate a system interrupt. (The CICR should also be initialized.)
10. Write $3830 to SMCMR to configure 8-bit characters, non-reversed data, and normal operation (not loopback). Notice that the transmitter and receiver have not been enabled yet.
11. Write $3833 to SMCMR to enable the SMC transmitter and receiver. This additional write ensures that the TEN and REN bits will be enabled last.

NOTE

After 5 bytes have been transmitted, the Tx BD is closed. Additionally, the receive buffer is closed after 16 bytes have been received. Any additional receive data beyond 16 bytes will cause a busy (out-of-buffers) condition since only one Rx BD was prepared.

7.11.13 SMC Interrupt Handling

The following list describes what would normally occur within an interrupt handler for the SMC.

1. Once an interrupt occurs, read the SMCE to see which sources have caused interrupts. The SMCE bits would normally be cleared at this time.
2. Process the Tx BD to reuse it if the TX bit was set in SMCE. Extract data from the Rx BD if the RX bit was set in SMCE. To transmit another buffer, simply set the Tx BD R-bit.
3. Clear the SMC1 bit in the CISR.
4. Execute the RTE instruction.

7.11.14 SMC as a GCI Controller

The SMC can be used to control the C/I and to monitor channels of the GCI frame. When using the SCIT configuration of GCI, one SMC can handle SCIT channel 0, and the other SMC can handle SCIT channel 1. The main features are as follows:

- Each SMC Channel Supports the C/I and Monitor Channels of the GCI (IOM-2) in ISDN Applications
- Two SMCs Support the Two Sets of C/I and Monitor Channels in SCIT Channel 0 and Channel 1
- Full-Duplex Operation
- Local Loopback and Echo Capability for Testing

W—Wrap (Final BD in Table)

0 = This is not the last buffer descriptor in the Tx BD Table.

1 = This is the last buffer descriptor in the Tx 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 TBASE). The number of Tx BDs in this table is programmable, and is determined only by the wrap bit and the overall space constraints of the dual-port RAM.

I—Interrupt

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

1 = The TX bit in the PIP event register will be set when this buffer has been serviced by the CP, which can cause an interrupt.

L—Last

0 = This buffer is not the last buffer of the frame.

1 = This buffer is the last buffer of the frame.

CM—Continuous Mode

0 = Normal Operation.

1 = The R-bit is not cleared by the CP after this buffer is closed, allowing the associated data buffer to be retransmitted automatically when the CP next accesses this BD. However, the R bit will be cleared if an error occurs during transmission

F—Fault

0 = The Fault status remained negated during transmission

1 = The Fault status was asserted during transmission

PE—Printer Error

0 = The PError status remained negated during transmission

1 = The PError status was asserted during transmission

S—Select Error

0 = The Select status remained asserted during transmission

1 = The Select status was negated during transmission

7.13.8.11 CENTRONICS TRANSMITTER EVENT REGISTER (PIPE) . When the Centronics Transmitter protocol is selected, the SMC2 event register is called the Centronics Transmitter event register. It is an 8-bit register which is used to report events recognized by the Centronics channel and generate interrupts. On recognition of an event, the Centronics controller will set its corresponding bit in the Centronics event register.

The Centronics event register is a memory-mapped register that may be read at any time. A bit is cleared by writing a one (writing a zero does not affect a bit’s value). More than one bit may be cleared at a time. All unmasked bits must be cleared before the CP will clear the internal interrupt request. This register is cleared at reset.

7.14.4 Port A Registers

Port A has four memory-mapped, read-write, 16-bit control registers.

7.14.4.1 PORT A OPEN-DRAIN REGISTER (PAODR). The PAODR indicates a normal or wired-OR configuration of the port pins. Six of the PAODR bits can be open-drain to correspond to those pins that have serial channel output capability. The other bits are always zero. PAODR is cleared at system reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	OD7	OD6	OD5	OD4	OD3	0	OD1	0

For each ODx bit, the definition is as follows:

0 = The I/O pin is actively driven as an output.

1 = The I/O pin is an open-drain driver. As an output, the pin is actively driven low, but in three-stated otherwise.

7.14.4.2 PORT A DATA REGISTER (PADAT). A read of PADAT returns the data at the pin, independent of whether the pin is defined as an input or an output. This allows detection of output conflicts at the pin by comparing the written data with the data on the pin. A write to the PADIR is latched, and if that bit in the PADIR is configured as an output, the value latched for that bit will be driven onto its respective pin. PADAT can be read or written at any time. PADAT is not initialized and is undefined at reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

7.14.4.3 PORT A DATA DIRECTION REGISTER (PADIR). PADIR is cleared at system reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DR15	DR14	DR13	DR12	DR11	DR10	DR9	DR8	DR7	DR6	DR5	DR4	DR3	DR2	DR1	DR0

For each DRx bit, the definition is as follows:

0 = The corresponding pin is an input.

1 = The corresponding pin is an output.

7.14.4.4 PORT A PIN ASSIGNMENT REGISTER (PAPAR). PAPAR is cleared at system reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DD15	DD14	DD13	DD12	DD11	DD10	DD9	DD8	DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0

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

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

WATCHDOG TIMER

The MC68302 timer 3 is called the watchdog timer. This software watchdog timer must be serviced periodically or can be used to generate a system reset. The software watchdog timer on the QUICC exists in the SIM60. It corresponds to the software watchdog available on other members of the MC68300 family.

The WRR and WCN on the MC68302 do not have direct counterparts on the QUICC. To program the software watchdog on the QUICC, the user should initially program the SYPCR, the SWIV (optional), and service the software watchdog using the SWSR. Additionally, an indication of a software watchdog reset may be found in the RSR.

9.3.4.4 INTERNAL REGISTERS (COMMUNICATION PROCESSOR). The following paragraphs detail the registers associated with the communications processor, according to their ascending order in the MC68302 memory map. Note that the address offsets of the QUICC registers are different than the offsets on the MC68302 registers.

The 8-bit command register (CR) also exists on the QUICC, but it is expanded to 16 bits.

The FLG bit still exists in the same location of the QUICC CR.

The two CH NUM bits have been shifted to bit locations 7 and 6 and expanded to four bits rather than two (bits 6 and 5 also), since more peripherals may now receive commands. If bits 6 and 5 are cleared, the SCC channel number encodings are compatible from the MC68302 to the QUICC.

The two OPCODE bits have been increased to four bits, shifted left in the register, and redefined to include many more commands. However, the original commands available on the MC68302 are still available.

The RST bit is shifted to become bit 15 of the QUICC CR. To reset the MC68302, the value $81 was issued to the CR. To reset the QUICC, the value $8001 is issued to the CR.

SCCs

The MC68302 SCC registers are mapped to the QUICC as follows.

The SCONx register controls the clocking scheme and baud rates of the MC68302 SCCs. On the QUICC, four independent baud rate generators that are not associated with specific SCCs are available. Therefore, this register is implemented in the QUICC baud rate generator and is called the BRGC. Note that the BRGC contains new bits, including a reset bit and an enable bit.

The DIV4 bit is now a divide-by-16 option and is implemented in the DIV16 bit of the QUICC BRGC.

The CD10–CD0 bits become CD11–CD0 and are located in the QUICC BRGC.

The RCS bit is implemented in the bank of clocks control in the three RxCS bits of the if DRAM is used elsewhere in the system. The QUICC does not support bursting by the MC68EC030.

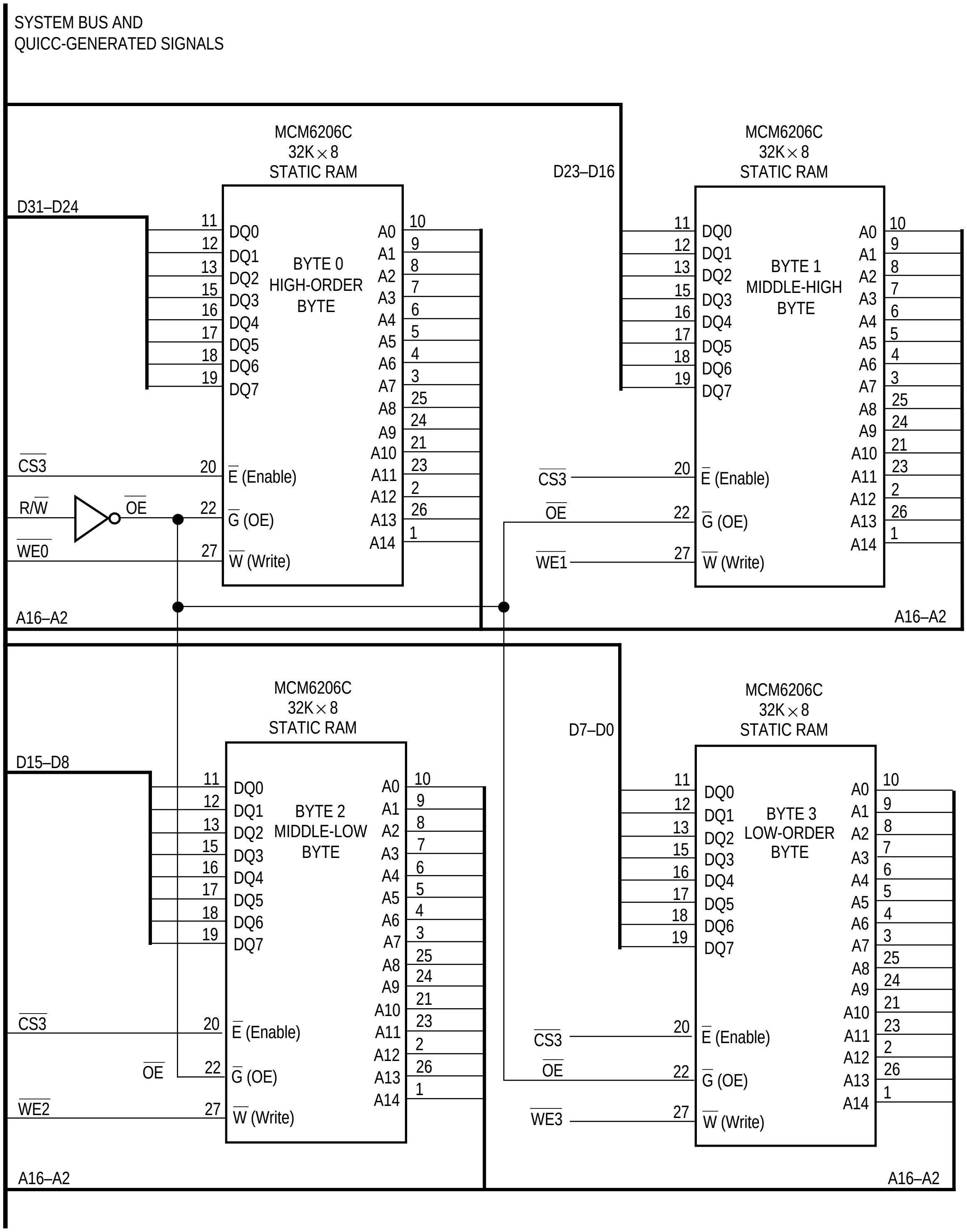


Figure 9-30. 128-Kbyte Static RAM Bank—32 Bits Wide

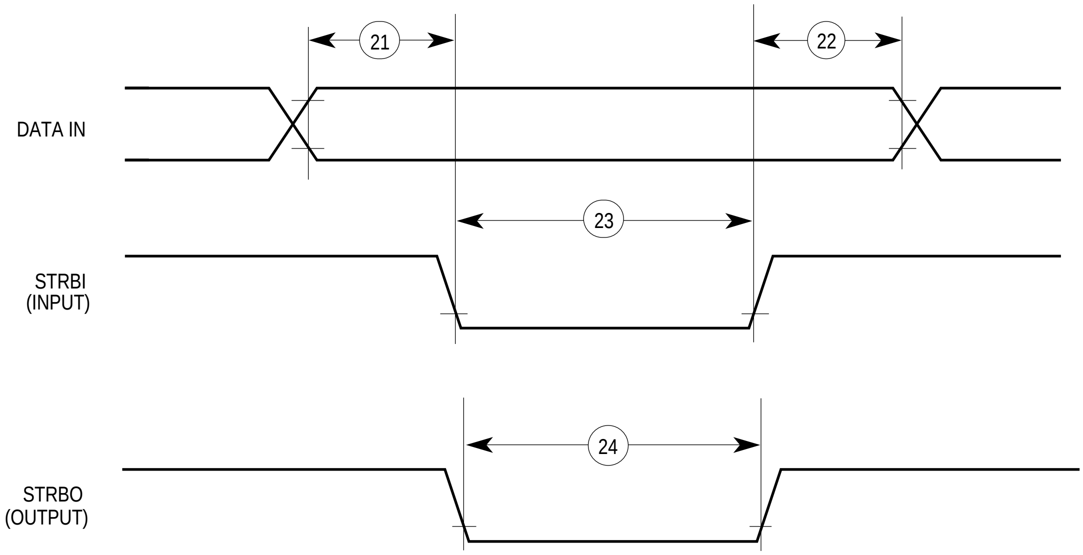


Figure 10-52. PIP Tx (Interlock Mode)

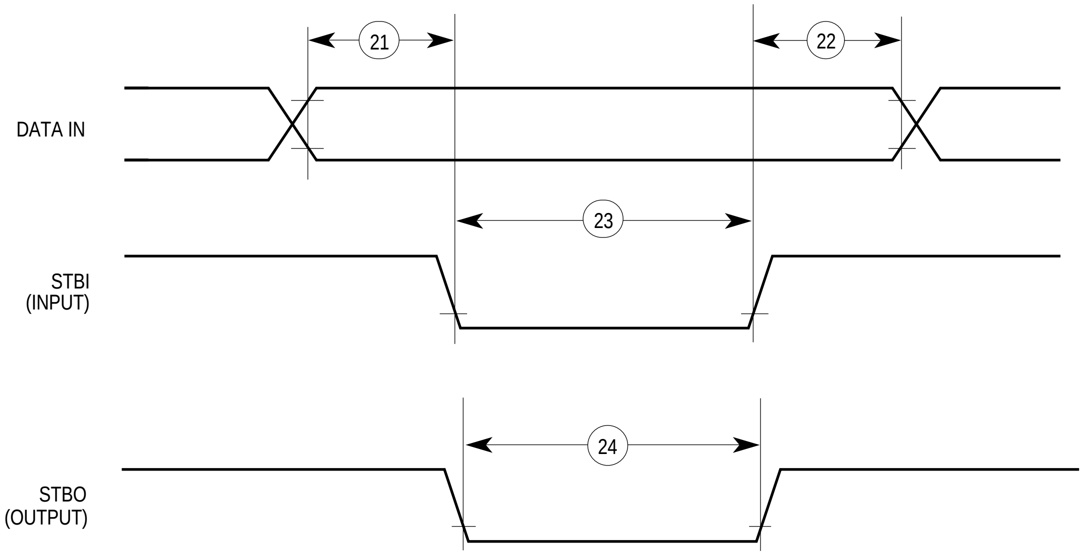


Figure 10-53. PIP Tx (Pulse Mode)

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