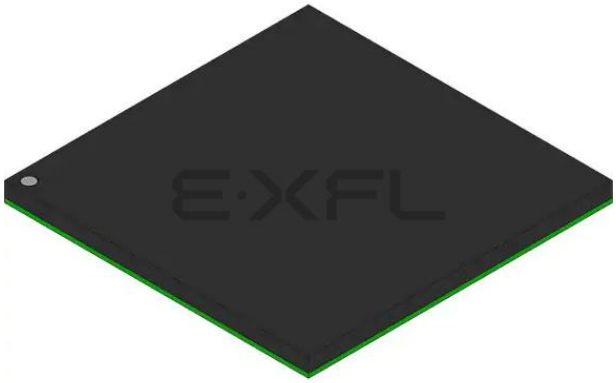


Welcome to E-XFL.COM

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	Active
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/pro/item?MUrl=&PartUrl=mc68360zq25vl

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.

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

Table 3-4. QUICC CPM Registers Memory Map

REGB + 515		8	Reserved			
REGB + 516	CMAR1	8	Channel Mask Register	00		
REGB + 517		8	Reserved			
REGB + 518	CSR1	8	IDMA1 Channel Status Register	00		
REGB + 519		24	Reserved			
REGB + 51C	SDSR	8	SDMA Status Register	00		SDMA
REGB + 51D		8	Reserved			
REGB + 51E	SDCR	16	SDMA Configuration Register	0000	H	
REGB + 520	SDAR	32	SDMA Address Register	XXXX XXXX		
REGB + 524		16	Reserved			IDMA2
REGB + 526	CMR2	16	IDMA2 Mode Register	0000		
REGB + 528	SAPR2	32	IDMA2 Source Address Pointer	0000 0000		
REGB + 52C	DAPR2	32	IDMA2 Destination Address Pointer	0000 0000		
REGB + 530	BCR2	32	IDMA2 Byte Count Register	0000 0000		
REGB + 534	FCR2	8	IDMA2 Function Code Register	00		
REGB + 535		8	Reserved			
REGB + 536	CMAR2	8	Channel Mask Register	00		
REGB + 537		8	Reserved			
REGB + 538	CSR2	8	IDMA2 Channel Status Register	00		
REGB + 539 to REGB + 53F			Reserved			
REGB + 540	CICR	24	CP Interrupt Configuration Register	xx00 0000	H	CPIC
REGB + 544	CIPR	32	CP Interrupt Pending Register	0000 0000		
REGB + 548	CIMR	32	CP Interrupt Mask Register	0000 0000		
REGB + 54C	CISR	32	CP In-Service Register	0000 0000		
REGB + 550	PADIR	16	Port A Data Direction Register	0000	H	Parallel I/O
REGB + 552	PAPAR	16	Port A Pin Assignment Register	0000	H	
REGB + 554	PAODR	16	Port A Open Drain Register	0000	H	
REGB + 556	PADAT	16	Port A Data Register	XXXX		
REGB + 558 to REGB + 55f			Reserved			
REGB + 560	PCDIR	16	Port C Data Direction Register	0000	H	
REGB + 562	PCPAR	16	Port C Pin Assignment Register	0000	H	
REGB + 564	PCSO	16	Port C Special Options	0000	H	
REGB + 566	PCDAT	16	Port C Data Register	XXXX		
REGB + 568	PCINT	16	Port C Interrupt Control Register	0000	H	
REGB + 56a to REGB + 57f			Reserved			
REGB + 580	TGCR	16	Timer Global Configuration Register	0000	H	TIMER

Freescale Semiconductor, Inc.

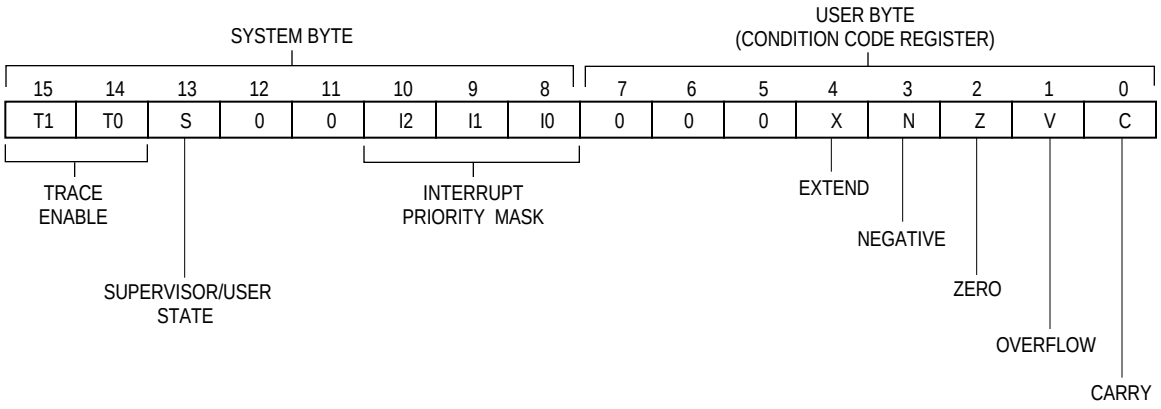


Figure 5-5. Status Register

5.3 INSTRUCTION SET

The following paragaphs describe the CPU32+ instruction set. A description of the instruction format, the operands used by the instructions, and a summary of the instructions by category are included. Complete programming information is provided in the M68000PM/AD, *M68000 Family Programmer's Reference Manual.*

The CPU32+ instructions include machine functions for all the following operations:

- Data Movement
- Arithmetic Operations
- Logical Operations
- Shifts and Rotates
- Bit Manipulation
- Conditionals and Branches
- System Control

The large instruction set encompasses a complete range of capabilities and, combined with the enhanced addressing modes, provides a flexible base for program development.

The instruction set of the CPU32+ is very similar to that of the MC68020 (see Table 5-1). The following M68020 instructions are not implemented on the CPU32+:

BFxx	—	Bit Field Instructions (BFCHG, BFCLR, BFEXTS, BFEXTU, BFFFO, BFINS, BFSET, BFTST)
CALLM, RTM	—	Call Module, Return Module
CAS, CAS2	—	Compare and Set (Read-Modify-Write Instructions)
cpxxx	—	Coprocessor Instructions (cpBcc, cpDBcc, cpGEN, cpRESTORE, cpSAVE, cpScc, cpTRAPcc)
PACK, UNPK	—	Pack, Unpack BCD Instructions

The CPU32+ traps on unimplemented instructions or illegal effective addressing modes, allowing user-supplied code to emulate unimplemented capabilities or to define special-purpose functions. However, Motorola reserves the right to use all currently unimplemented instruction operation codes for future M68000 core enhancements.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

The SIM60 has additional support of low-power modes. The clock synthesizer provides system clocks to the SIM60 and other modules. This clock scheme supports low-power modes for applications that use the baud rate generators and/or serial ports during the standby mode. The main system clock can be changed dynamically (the slow-go option) while the baud rate generators and serial ports work with a fixed frequency.

The breakpoint logic provides an internal breakpoint address register that allows hardware breakpoints in a QUICC system. This function is especially useful during in-field debugging activity when it is difficult to connect an in-circuit emulator or logic analyzer to the target board.

The QUICC supports the slave mode. In this mode, the CPU32+ core on the QUICC is disabled, and the QUICC functions as an intelligent peripheral. For instance, if the application requires more serial channels than the QUICC provides, multiple QUICCs may be configured onto the same system bus, one with its CPU enabled and the rest in slave mode. Alternatively, if the application needs additional CPU performance, the QUICC may function as a companion chip to an MC68EC040 (or other M68040 family member). This is called MC68040 companion mode. In this mode, the QUICC's glueless interface to the MC68EC040 provides a two-chip MC68EC040 system solution. The MC68EC040 can also control multiple QUICCs in slave mode. Finally, the QUICC slave mode may also support an external MC68EC030 or other M68030 family member.

The EBI handles the transfer of information between the internal CPU32+ core and memory, peripherals, or other processing elements in the external address space, or between an external master and the QUICC RAM and registers. Section 4 Bus Operation describes the bus operation, but the configuration control of the EBI is contained in this section.

The following functions are physically part of the SIM60, but are described in other places in this manual.

The memory controller module provides glueless interfaces to many types of memory and peripherals. It contains up to 8 general-purpose chip selects with up to 15 wait states each and a full DRAM controller that controls up to 8 DRAM banks. See 6.10 Memory Controller for further information.

The QUICC dynamically interprets the bus port size of an addressed device during each bus cycle, allowing operand transfers to/from 8-, 16-, and 32-bit ports. The $\overline{\text{DSACK}}$ signals are used to signify the data port size. Dynamic bus sizing can result in reduced system cost. For instance, an 8-bit boot EPROM may be used with 16-bit peripherals and 32-bit DRAM. Dynamic bus sizing also allows a programmer to write code that is not bus-width specific. For a discussion on dynamic bus sizing see Section 4 Bus Operation.

The QUICC is designed to allow external bus masters the opportunity to access the intermodule bus (IMB). This design has two main purposes. First, the RAM and peripherals on the QUICC can be directly accessed, if desired, by an external master. Second, the external master can use QUICC resources, such as the chip-select generation logic and DRAM controller. See Section 4 Bus Operation for further discussion.

Freescale Semiconductor, Inc.

Freescale Semiconductor, Inc.

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.

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

ENb—Enable Channel b

0 = Channel b is disabled. The SI RAMs and TDM routing are in a state of reset, but all other SI functions still operate.
1 = The SI is enabled.

ENa—Enable Channel a

0 = Channel a is disabled. The SI RAMs and TDM routing are in a state of reset, but all other SI functions still operate.
1 = The SI is enabled.

RDM1–RDM0—RAM Division Mode

These bits define the RAM division mode and the number of multiplexed channels supported in the SI.

00 = The SI supports one TDM channel with 64 entries for receive routing and 64 entries for transmit routing.
01 = The SI supports one TDM channel with 32 entries for receive routing and 32 entries for transmit routing. There are an additional 32 shadow entries for the receive routing and 32 shadow entries for transmit routing that may be used to dynamically change the routing.
10 = The SI supports two TDM channels with 32 entries for the receive routing and 32 entries for transmit routing for each of the two TDMs.
11 = The SI supports two TDM channels with 16 entries for receive routing and 16 entries for transmit routing for each channel. There are an additional 16 shadow entries for receive routing and 16 shadow entries for transmit routing that may be used to dynamically change the channel routing.

NOTE

TSAa must be used in RDM1—0 if 00 or 01 setting is desired.

7.8.5.2 SI MODE REGISTER (SIMODE). The 32-bit SIMODE defines the SI operation modes. This register allows the user (in conjunction with the SI RAM) to support any or all of the ISDN channels independently when in IDL or GCI (IOM-2) mode. Any extra SCC channel can then be used for other purposes in NMSI mode. SIMODE appears to the user as a memory-mapped, read-write register and is cleared at reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SMC2	SMC2CS			SDMb		RFSDb		DSCb	CRTb	STZb	CEb	FEb	GMb	TFSDb	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMC1	SMC1CS			SDMa		RFSDa		DSCa	CRTa	STZa	CEa	FEa	GMa	TFSDa	

SMCx—SMCx Connection

0 = NMSI mode. The clock source is determined by the SMCxCS bit, and the data comes from a dedicated pin (SMTXD1 and SMRXD1 for SMC1 or SMTXD2 and SMRXD2 for SMC2) in the NMSI.
1 = SMCx is connected to the multiplexed SI (TDM channel).

Freescale Semiconductor, Inc.

RFSDx—Receive Frame Sync Delay for TDM A or B

These two bits determine the number of clock delays between the receive sync and the first bit of the receive frame. Even if the CRTx bit is set, these bits do not control the delay for the transmit frame.

00 = No bit delay (The first bit of the frame is transmitted/received on the same clock as the sync; use for GCI.)
01 = 1-bit delay (Use for IDL.)
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.

DSCx—Double-Speed Clock for TDM A or B

Some TDMs such as GCI define the input clock to be 2× faster than the data rate. This bit controls this option.

0 = The channel clock (L1RCLKx and/or L1TCLKx) is equal to the data clock. (Use for IDL and most TDM formats.)
1 = The channel clock rate is twice the data rate. (Use for GCI.)

CRTx—Common Receive and Transmit Pins for TDM A or B

This bit is useful when the transmit and receive sections of a given TDM use the same clock and sync signals. In this mode, L1TCLKx and L1TSYNCx pins can be used as general-purpose I/O pins.

0 = Separate pins. The receive section of this TDM uses L1RCLKx and L1RSYNCx pins for framing, and the transmit section uses L1TCLKx and L1TSYNCx for framing.
1 = Common pins. The receive and transmit sections of this TDM use L1RCLKx as clock pin of channel x and L1RSYNCx as the receive and transmit sync pin. (Use for IDL and GCI.)

STZx—Set L1TXDx to Zero for TDM A or B

0 = Normal operation.
1 = L1TXDx is set to zero until serial clocks are available, which is useful for GCI activation. Refer to 7.8.7.1 SI GCI Activation/Deactivation Procedure.

CEx—Clock Edge for TDM A or B

When DSCx =0
0 = The data is transmitted on the rising edge of the clock and received on the falling edge. (Use for IDL and GCI.)
1 = The data is transmitted on the falling edge of the clock and received on the rising edge.

When DSCx = 1
0 = The data is transmitted on the rising edge of the clock and received on the rising edge. (Use for IDL and GCI.)
1 = The data is transmitted on the falling edge of the clock and received on the falling edge.

Freescale Semiconductor, Inc.

5. PAPAR bits 6, 7, and 8 = 1. Configures L1TXDa, L1RXDa, and L1RCLKa.
6. PADIR bits 6 and 7 = 1. PADIR bit 8 = 0. Configures L1TXDa, L1RXDa, and L1RCLKa.
7. PCPAR bits 3, 10, and 11 = 1. Configures L1RQa, L1TSYNCa, and L1RSYNCa.
8. PCDIR bit 3 = 0. L1RQa is an input. L1TSYNCa will perform the L1GRa function and is therefore an output, but it does not need to be configured with a PCDIR bit. L1RSYNCa is an input, but it does not need to be configured with a PCDIR bit.
9. SIGMR = $04. Enable TDMa (one static TDM).
10. 1SICMR is not used.
11. 1SISTR and SIRP do not need to be read, but can be used for debugging information once the channels are enabled.
12. 1Enable the SCC1 for HDLC operation (to handle the LAPD protocol of the D channel), and set SCC2 and SCC4 as desired.

7.8.7 SI GCI Support

The normal mode of the GCI, also known as the ISDN-oriented modular rev 2.2 (IOM-2), and the SCIT are fully supported by the QUICC. The QUICC also supports the D channel access control in S/T interface terminals by using the command/indication (C/I) channel for that function.

The GCI bus consists of four lines: two data lines, a clock, and a frame synchronization line. Usually, an 8-kHz frame structure defines the various channels within the 256-kbps data rate. The QUICC can support two independent GCI buses and has independent receive and transmit sections for each one. The interface can also be used in a multiplexed frame structure on which up to eight physical layer devices multiplex their GCI channels. In this mode, the data rate would be 2048 kbps.

In the GCI bus, the clock rate is twice the data rate. The SI divides the input clock by two to produce the data clock.

The QUICC also has data strobe lines, and the 1× data rate clock L1CLKOx output pins. These signals are used for interfacing devices to GCI that do not support the GCI bus.

The GCI signals for each transmit and receive channel are as follows:

L1RSYNCx—Used as GCI sync signal; input to the QUICC. This signal indicates that the clock periods following the pulse designate the GCI frame.

L1RCLKx—Used as GCI clock; input to the QUICC. The L1RCLKx signal is twice the data clock.

L1RXDx—Used as GCI receive data; input to the QUICC.

L1TXDx—Used as GCI transmit data; open-drain output. Valid only for the bits that are supported by the IDL; three-stated otherwise.

L1CLKOx—Optional signal; output from QUICC. This 1× clock output can be used to clock devices that do not interface directly to GCI. If the double-speed clock

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

FLC—Flow Control

0 = Normal operation. The GSMR and port C registers determine the mode of the $\overline{\text{CTS}}$ pin.

1 = Asynchronous flow control. When the $\overline{\text{CTS}}$ pin is negated, the transmitter will stop transmitting at the end of the current character. (If $\overline{\text{CTS}}$ is negated past the middle of the current character, the next full character may be sent, and then transmission will be stopped.) When $\overline{\text{CTS}}$ is asserted once more, transmission will continue where it left off. No $\overline{\text{CTS}}$ lost error will be reported. No characters except idles will be transmitted while $\overline{\text{CTS}}$ is negated.

SL—Stop Length

The SL bit selects the number of the stop bits transmitted by the UART. This bit may be modified on the fly. The receiver is always enabled for one stop bit unless the UART is in synchronous mode and the RZS bit is set. Fractional stop bits are configured in the DSR.

0 = One Stop Bit
1 = Two Stop Bits

CL—Character Length

The CL bits determine the number of data bits in the character, not including the optional parity or multidrop address bits. When less than an 8-bit character is used, the MSBs in memory are written as zeros, and on transmission the MSBs in memory are a don’t care. These bits may be modified on the fly.

00 = 5 Data Bits
01 = 6 Data Bits
10 = 7 Data Bits
11 = 8 Data Bits

UM—UART Mode

The UART mode bits select the protocol that is implemented over the ASYNC channel. These bits may be modified on the fly.

00 = Normal UART operation. Multidrop mode is disabled, and an idle-line wake-up is selected. In the idle-line wake-up mode, the UART receiver is reenabled by receiving one character of all ones.

01 = Multidrop non-automatic mode. In the multidrop mode, an additional address/data bit is transmitted with each character. The multidrop asynchronous modes are compatible with the MC68681 DUART, the MC68HC11 SCI, the DSP56000 SCI, and the Intel 8051 serial interface. The UART receiver is reenabled when the last data bit received in the character (i.e., the address bit) is a one. This means that the received character is an address that has to be processed by all inactive processors. The UART receives the address character and writes it to a new buffer. The CPU32+ core then compares the written address with its own address to decide whether to ignore or process the following characters.

10 = Reserved

11 = Multidrop automatic mode. In this mode, the CP automatically checks the address of the incoming address character using the UADDR1 and UADDR2 parameter

Freescale Semiconductor, Inc.

When the SPI is working as a master, SPICLK is the clock output signal that shifts in the received data from the SPIMISO pin and shifts out the transmitted data to the SPIMOSI pin. Additionally, an SPI master device must provide a slave select signal output to enable the SPI slave devices. This may be implemented using one of the QUICC's general-purpose I/O pins. The $\overline{\text{SPISEL}}$ pin should not be asserted while the SPI is working as a master, or the SPI will indicate an error.

When the SPI is working as a slave, SPICLK is the clock input signal that shifts in the received data from the SPIMOSI pin and shifts out the transmitted data to the SPIMISO pin. The $\overline{\text{SPISEL}}$ pin provided by the QUICC is the enable input to the SPI slave.

When the SPI is working in a multi-master environment, the $\overline{\text{SPISEL}}$ pin is still an input and is used to detect an error condition when more then one master is operating.

SPICLK is a gated clock (i.e., the clock only toggles while data is being transferred). The user can select any of four combinations of SPICLK phase and polarity using two bits in the SPI mode register (SPMODE).

The SPI pins can also be configured as open-drain pins to support a multi-master configuration where the same SPI pin can be driven by the QUICC or an external SPI device.

7.12.4 SPI Transmit/Receive Process

The following paragraphs discuss SPI master, slave, and multi-master operation.

7.12.4.1 SPI MASTER MODE. When the SPI functions in master mode, the SPI transmits a message to the peripheral (SPI slave), which in turn sends back a simultaneous reply. When the QUICC works with more than one slave, it can use the general-purpose parallel I/O pins to selectively enable different slaves.

To begin the data exchange, the CPU32+ core writes the data to be transmitted into a data buffer, configures a Tx BD with its R-bit set and configures one or more Rx BDs. The CPU32+ core should then set the STR bit in the SPCOM to start transmission of data. The data will begin transmission once the SDMA channel has loaded the transmit FIFO with data.

The SPI controller then generates programmable clock pulses on the SPICLK pin for each character and shifts the data out on the SPIMOSI pin. At the same time, the SPI shifts receive data in from the SPIMISO pin. This receive data is written into a receive buffer using the next available Rx BD. The SPI will continue transmitting and receiving characters until the transmit buffer has been completely transmitted or an error has occurred ($\overline{\text{SPISEL}}$ pin unexpectedly asserted). The CP then clears the R and E bits in the Tx BD and Rx BD, and issues a maskable interrupt to the CPM interrupt controller.

When multiple Tx BDs are ready for transmission, the Tx BD L-bit determines whether the SPI continues to transmit without waiting for the STR bit to be set again. If the L-bit is cleared, the data from the next Tx BD will begin its transmission following the transmission of data from the first Tx BD. In most cases, the user should see no delay on the SPIMOSI pin between buffers. If the L-bit is set, transmission will cease after data from this Tx BD has

Freescale Semiconductor, Inc.

7.13.6 Transparent Data Transfers

In the transparent handshake mode, the PIP may be configured as a transmitter or a receiver. This configuration has only one handshake pin.

The transparent mode is controlled only by the RISC. Operation using the RISC requires BDs and parameter RAM initialization very similar to the other serial channels. Data is then stored in the buffers using one of the SDMA channels (one of the available channels from SMC2).

NOTE

At the time of writing, this operation of the PIP has not been fully defined. This PIP operation may be implemented by the CPU32+ core, using the port B parallel I/O registers and any port C interrupt pin.

In this mode, the B17 pin falling edge generates the request to the RISC, which causes the RISC to receive/transmit data. The direction of the pins is controlled by the port B data direction register (PBDIR). The transparent handshake mode is shown in Figure 7-93.

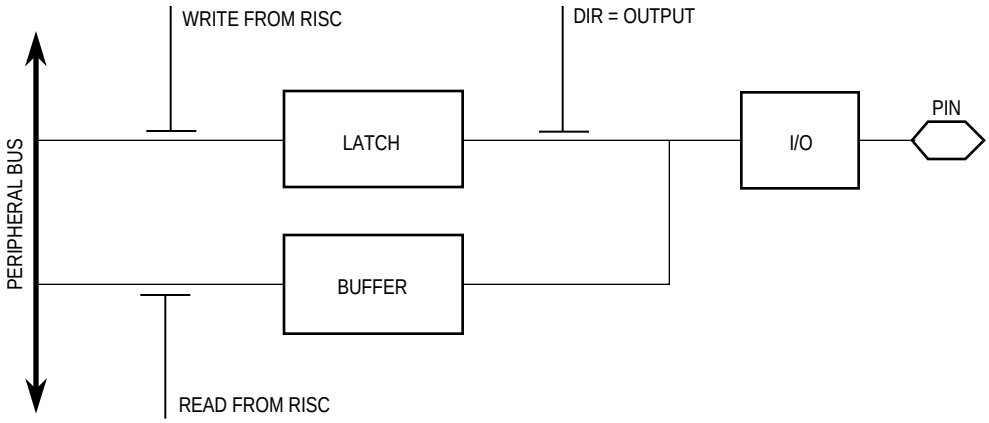


Figure 7-93. PIP Transparent Handshake Mode

7.13.7 Programming Model

The following paragraphs describe the PIP registers and parameter RAM.

7.13.7.1 PARAMETER RAM. At the time of writing, RISC operation on the PIP has not been fully defined. The user should use the CPU32+ core operation mode until the RISC microcode becomes available or the full PIP microcode becomes available in the RISC internal ROM. Please contact the local Motorola sales representative to obtain the current status of the PIP RISC microcode.

Freescale Semiconductor, Inc.

The following list gives an example of how to find the beginning of the interrupt handler from the interrupt vector. SCC1 is used as an example.

1. Formulate the 8-bit vector. The three MSBs come from VBA2–VBA0 in the CICR. Assume these are programmed to 101b. The five LSBs have a fixed value of 11110b (see Table 7-22). Thus, the 8-bit vector is 10111110b. This is the value presented on the bus during an interrupt acknowledge cycle.
2. Multiply by 4 to get the offset address of the vector in the vector table. Thus, the offset address is 1011111000b = $2F8.
3. Determine the full vector address. In a CPU32+ system, the offset is added to the vector base register in the CPU32+. Assuming that the vector base register = $80000000, the final vector address is $800002F8.
4. Determine the location of the interrupt handler. At location $800002F8, the address of the interrupt handler is stored. If the long word at location $800002F8 contains $80001000, then the first instruction of the SCC1 interrupt handler will be found at $80001000.

7.15.5 CPIC Programming Model

The user interfaces with the CPIC via four registers. The CICR defines the overall CPM interrupt attributes. The CIPR indicates which CPM interrupt sources require interrupt service. The CIMR allows the user to prevent any CPM interrupt source from generating an interrupt request. The CISR allows a fully nested environment capability for interrupt requests within the CPM interrupt level.

7.15.5.1 CPM INTERRUPT CONFIGURATION REGISTER (CICR). The 24-bit read-write CICR defines the request level for the CPM interrupts, the priority between the SCCs, the highest priority interrupt, and the vector base address. The CICR, which can be dynamically changed by the user, is cleared at reset.

23	22	21	20	19	18	17	16	15	14	13	12
SCdP		SCcP		SCbP		SCaP		IRL2	IRL1	IRL0	HP4
11	10	9	8	7	6	5	4	3	2	1	0
IHP3	HP2	HP1	HP0	VBA2	VBA1	VBA0	—				SPS

SCdP—SCCd Priority Order

These two bits define which SCC will assert its request in the SCCd priority position. The user should not program the same SCC to more than one priority position (a, b, c, or d). These bits may be changed dynamically.

00 = SCC1 will assert its request in the SCCd position.
01 = SCC2 will assert its request in the SCCd position.
10 = SCC3 will assert its request in the SCCd position.
11 = SCC4 will assert its request in the SCCd position.

Freescale Semiconductor, Inc.

NOTE

The exception vector table may have been located in ROM, in which case the PIT vector location should be initialized before power-on reset.

Also, before the interrupt can be processed, the CPU32+ status register (SR) must be programmed to change the interrupt mask bits from a value of $7 to a value of $0 (or at least a value lower than the interrupt level desired).

Next, write the PITR bits in the PITR to start the timer and wait for the interrupt.

Step 17: Test the CPM

After the SIM60 is programmed, it is time to begin testing the CPM. In order of increasing initialization complexity, the following sub-blocks may be tested. See the initialization examples included in this manual where the blocks are described.

- Dual-Port RAM
- Parallel I/O Ports A, B, and C
- Baud Rate Generators
- Four General-Purpose Timers
- RISC Timer Tables
- IDMA (without buffer chaining)
- SPI (loopback mode test)
- IDMA (with buffer chaining)
- SMCs (loopback mode test)
- SCCs (loopback mode without the time slot assigner)
- SCCs (loopback mode with the time slot assigner)

The SPI, SMCs, and SCCs should be tested in loopback mode before attempting to send and receive data externally.

Step 18: Generate Interrupts with the CPM

When testing interrupts on the CPM, the user should implement this gradually. First, generate an interrupt with a timer or parallel I/O pin. Then proceed to more complicated interrupt structures like the serial channels.

Step 19: Enable External Interrupts

The next step is to allow external devices (if any) to interrupt the QUICC. These interrupts can enter the QUICC through the SIM60 ($\overline{\text{IRQx}}$ pins) or the CPM (parallel I/O pins with interrupt capability).

Freescale Semiconductor, Inc.

QUICC supports internally (e.g., the level of the SIM60 and the level of the CPM). If such a condition occurs, $\overline{\text{BERR}}$ will be asserted by the QUICC.

9.8.1.9 SOFTWARE WATCHDOG. If desired, the MC68EC030 can program the QUICC software watchdog to generate a level 7 interrupt or a system reset. In this application, the software watchdog is configured in software to generate a reset so that the breakpoint logic can use level 7 interrupts. No additional hardware is required because the connection between the reset pins of the QUICC and the MC68EC030 is already made.

9.8.1.10 PERIODIC INTERVAL TIMER. If desired, the MC68EC030 can use the periodic interval timer on the QUICC to generate a system interrupt, such as for a real-time kernel. No additional hardware is required for this function.

9.8.1.11 MC68EC030 CACHING CONFIGURATION. The MC68EC030 can cache or not cache data and program memory as desired. However, it is strongly advisable not to cache the data that is accessed by the QUICC serial channels because of the overhead incurred every time the cached data area is written.

9.8.1.12 DOUBLE BUS FAULT. In slave mode, the QUICC double bus fault monitor is not operational.

9.8.1.13 JTAG AND THREE-STATE. The QUICC provides JTAG ports, commonly known as JTAG. This interface uses five pins: TMS, TDI, TDO, TCK, and $\overline{\text{TRST}}$. TMS and TDI are left unconnected because they have internal pullups. The JTAG ports of both parts are disabled in this application; however, the capability could be easily added.

When the QUICC is in master mode, it provides a $\overline{\text{TRIS}}$ pin that allows all outputs on the device to be three-stated. In slave mode, this feature is not available since the QUICC is a peripheral of the system.

9.8.1.14 QUICC SERIAL PORTS. The functions on QUICC parallel I/O ports A, B, and C may be used as desired in this application and have no bearing on the MC68EC030 interface. However, any unused parallel I/O pins should be configured as outputs, so they are not left floating.

9.8.2 Memory Interfaces

In this application, a number of memory arrays have been developed for EPROM, flash EPROM, EEPROM, SRAM, and DRAM. Each memory interface can be attached to the system bus as desired.

One issue not discussed is the decision of whether external buffers are needed on the system bus. This issue depends on the number of memory arrays used in the design and possibly the layout (i.e., capacitance) of the system bus.

Another issue left to the user is the number of wait states used with each memory system. This depends on the memory speed, whether external buffers are used, and the loading on the system bus pins. (The QUICC provides capacitance de-rating figures to calculate the effect of more or less capacitance on the AC Timing Specifications.)

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

The AM27–AM11 bits should be programmed to determine the block size of the chip select or $\overline{\text{RASx}}$ line. This should be the total number of bytes in each memory array except for the EEPROM, which should be 32 Kbytes, rather than 8 Kbytes.

FCM3–FCM0 may be set to all ones to allow the chip select or $\overline{\text{RASx}}$ line to assert on all function codes except CPU space (interrupt acknowledge). It is advisable to program FCM3–FCM0 to ones, at least during the initial stages of debugging.

BCYC1–BCYC0 is not applicable.

PGME should be set to enable page mode and cleared otherwise.

SPS1–SPS0 should be cleared (32-bit SRAM port).

DSSEL should be set only if this is a DRAM bank.

9.8.4 Interfacing Multiple QUICCs to an MC68EC030

It is possible to interface multiple QUICCs to an MC68EC030. The first QUICC can be configured as previously shown in this subsection. Additional QUICCs should be configured as noted in the following list:

- The additional QUICCs should have their CONFIG2–CONFIG0 pins configured for slave mode, global chip select *disabled*, and MBAR at $003FF04.
- The MBAR of the additional QUICCs should be programmed using the $\overline{\text{MBARE}}$ pin and MBARE register as described in the Section 6 System Integration Module (SIM60).
- An external bus arbiter is required to take the bus request of the additional QUICC (which is an output because of the CONFIG2–CONFIG0 pins) and prioritize it with the other QUICCs, present it to the MC68EC030, and issue a bus grant to the appropriate QUICC.
- An external interrupt prioritizer is required to determine which QUICC $\overline{\text{IOUT2}}$–$\overline{\text{IOUT0}}$ pins are currently routed to the MC68EC030. Alternatively, the additional QUICC should have its interrupts brought out on a single $\overline{\text{RQOUT}}$ pin, which is routed to one of the original QUICC interrupt inputs. This would eliminate the external logic.

9.8.5 Using a Higher Speed MC68EC030 Master with the QUICC

It is possible to interface an MC68EC030 and QUICC through an asynchronous bus. This should allow an external master to operate at higher frequencies than those of the QUICC with minimal effort. As of this writing, the QUICC top frequency is 25 MHz; whereas, MC68EC030s are available up to 40 MHz. One potentially attractive option for a designer would be to consider disabling the CPU32+ core and increasing system performance by adding a 40-MHz MC68EC030 asynchronously. While this option is available, it is important for the designer to consider what effects a higher speed MC68EC030 would ultimately have on system cost and performance over using the QUICC CPU32+ at a lower frequency.

For the designer to take full advantage of a high-speed MC68EC030, it will be necessary to add additional glue to that shown in Figure 9-27. The additional circuitry takes the form of a DRAM controller, which is used instead of using the QUICC memory controller. The need for the additional logic is twofold. First, if the QUICC memory controller capabilities are used, all memory accesses would be at the clock rate of 25 MHz. In addition, since the

Freescale Semiconductor, Inc.

For More Information On This Product,
Go to: www.freescale.com

10.9 BUS OPERATION AC TIMING SPECIFICATIONS (CONTINUED)

Num.	Characteristic	Symbol	3.3 V/5.0 V 25.0 MHz		5.0V 33.34MHz		Unit
			Min	Max	Min	Max	
48[5,7]	DSACK Asserted to BERR , HALT Asserted	t_{DABA}	—	30	—	22.5	ns
49	CLKO1 high to BERR, RESETS, RESETH output Low	t_{CHRO}	—	—	—	—	ns
53	Data-Out, Parity-Out Hold from CLKO1 High	t_{DOCH}	0	—	0	—	ns
54	CLKO1 High to Data-Out, Parity-Out High Impedance	t_{CHDH}	—	20	—	15	ns
55	R/W Asserted to Data Bus Impedance Change	t_{RADC}	25	—	19	—	ns
56	RESET Pulse Width (Reset Instruction)	t_{HRPW}	512	—	512	—	CLKO1
56A	RESET Pulse Width (Input from External Device)	t_{RPWI}	20	—	20	—	CLKO1
57	BERR Negated to HALT Negated (Rerun)	t_{BNHN}	0	—	0	—	ns
58	CLKO1 High to BERR, RESETS, RESETH Driven Low	t_{CHBRL}	—	30		26	ns
58A	CLKO1 Low to RESETS Driven Low (upon Reset Instruction execution only)	t_{CLRL}	—	30		26	ns
60	CLKO1 High to BCLR0 Asserted	t_{CHBCA}	—	20	—	15	ns
61	CLKO1 High to BCLR0 Negated	t_{CHBCN}	—	20	—	15	ns
62[9]	BR Synchronous Setup Time	t_{BRSU}	5	—	3.75	—	ns
63[9]	BR Synchronous Hold Time	t_{BRH}	10	—	7.5	—	ns
64[9]	BGACK Synchronous Setup Time	t_{BGSU}	5	—	3.75	—	ns
65[9]	BGACK Synchronous Hold Time	t_{BGH}	10	—	7.5	—	ns
66	BR low to CLKO1 Rising Edge (040 comp. mode)	t_{BRCH}	5	—	5	—	ns
70	CLKO1 Low to Data Bus Driven (Show Cycle)	t_{SCLDD}	0	30	0	22.5	ns
71	Data Setup Time to CLKO1 Low (Show Cycle)	t_{SCLDS}	10	—	7.5	—	ns
72	Data Hold from CLKO1 Low (Show Cycle)	t_{SCLDH}	6	—	3.75	—	ns
73	BKPT Input Setup Time	t_{BKST}	10	—	7.5	—	ns
74	BKPT Input Hold Time	t_{BKHT}	6	—	3.75	—	ns
75	RESETH Low to Config2–0, MOD1–0, B16M valid	t_{MST}	—	500	—	500	CLKO1
76	Config2–0	t_{MSH}	0	—	0	—	ns
77	MOD1–0 Hold Time, B16M Hold Time	t_{MSH}	10	—	10	—	CLKO1
80	DSI Input Setup Time	t_{DSISU}	10	—	7.5	—	ns
81	DSI Input Hold Time	t_{DSIH}	6	—	3.75	—	ns
82	DSCLK Setup Time	t_{DSCSU}	10	—	7.5	—	ns
83	DSCLK Hold Time	t_{DSCH}	6	—	3.75	—	ns
84	DSO Delay Time	t_{DSOD}	—	t_{cyc}+ 20	—	tcyc+ 20	ns
85	DSCLK Cycle	t_{DSCCYC}	2	—	2	—	CLKO1
86	CLKO1 High to Freeze Asserted	t_{FRZA}	0	35	0	26.25	ns
87	CLKO1 High to Freeze Negated	t_{FRZN}	0	35	0	26.25	ns

Freescale Semiconductor, Inc.

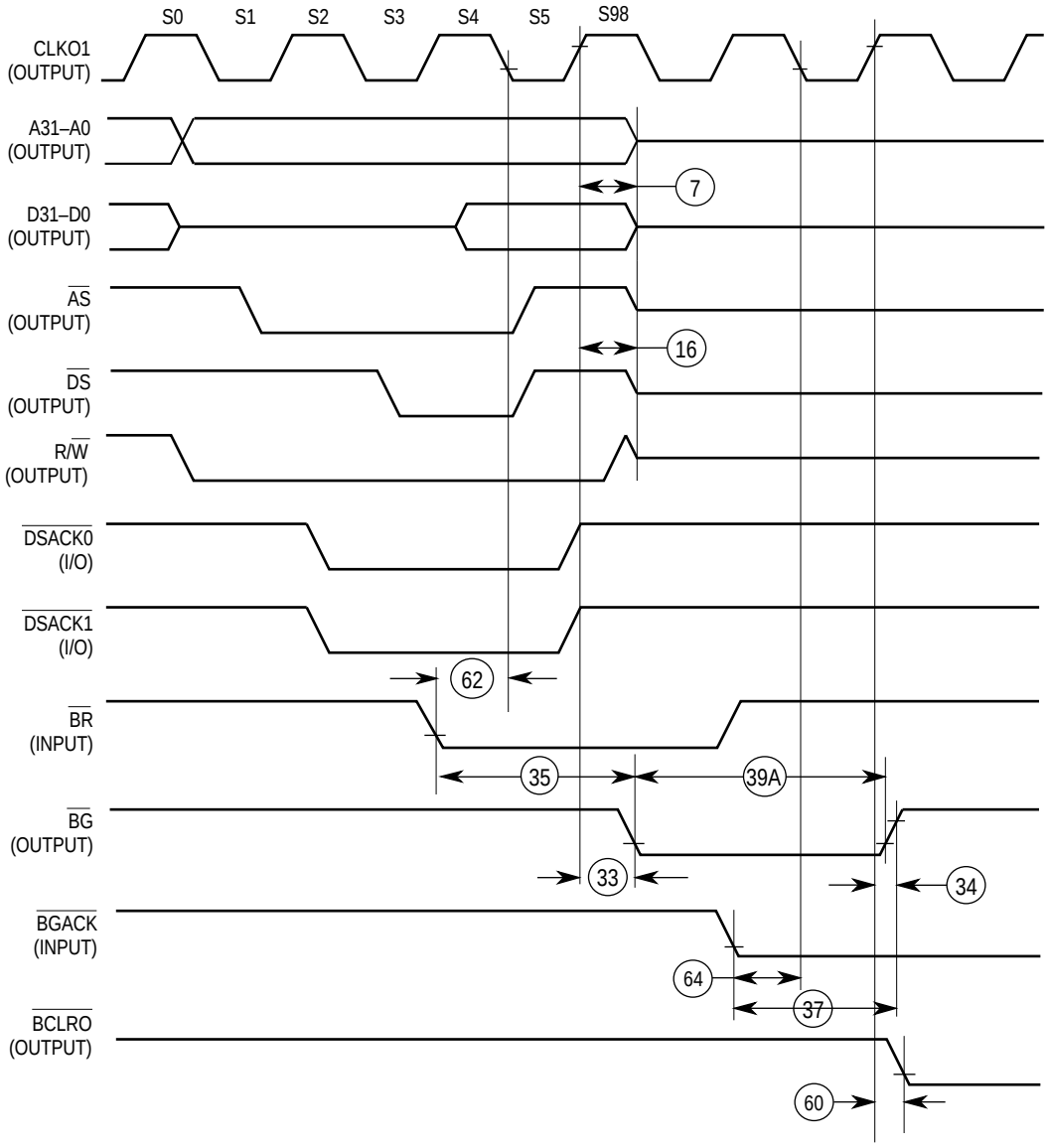


Figure 10-15. SYNC Bus Arbitration – Active Bus Case

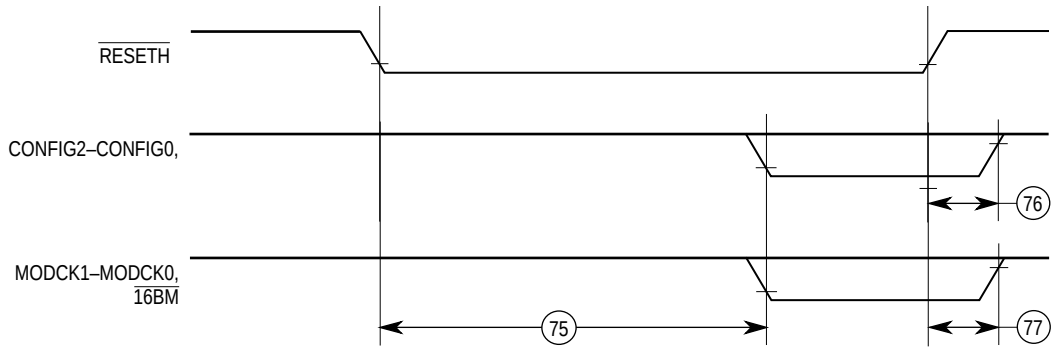


Figure 10-16. Configuration And Clock Mode Select Timing

Freescale Semiconductor, Inc.

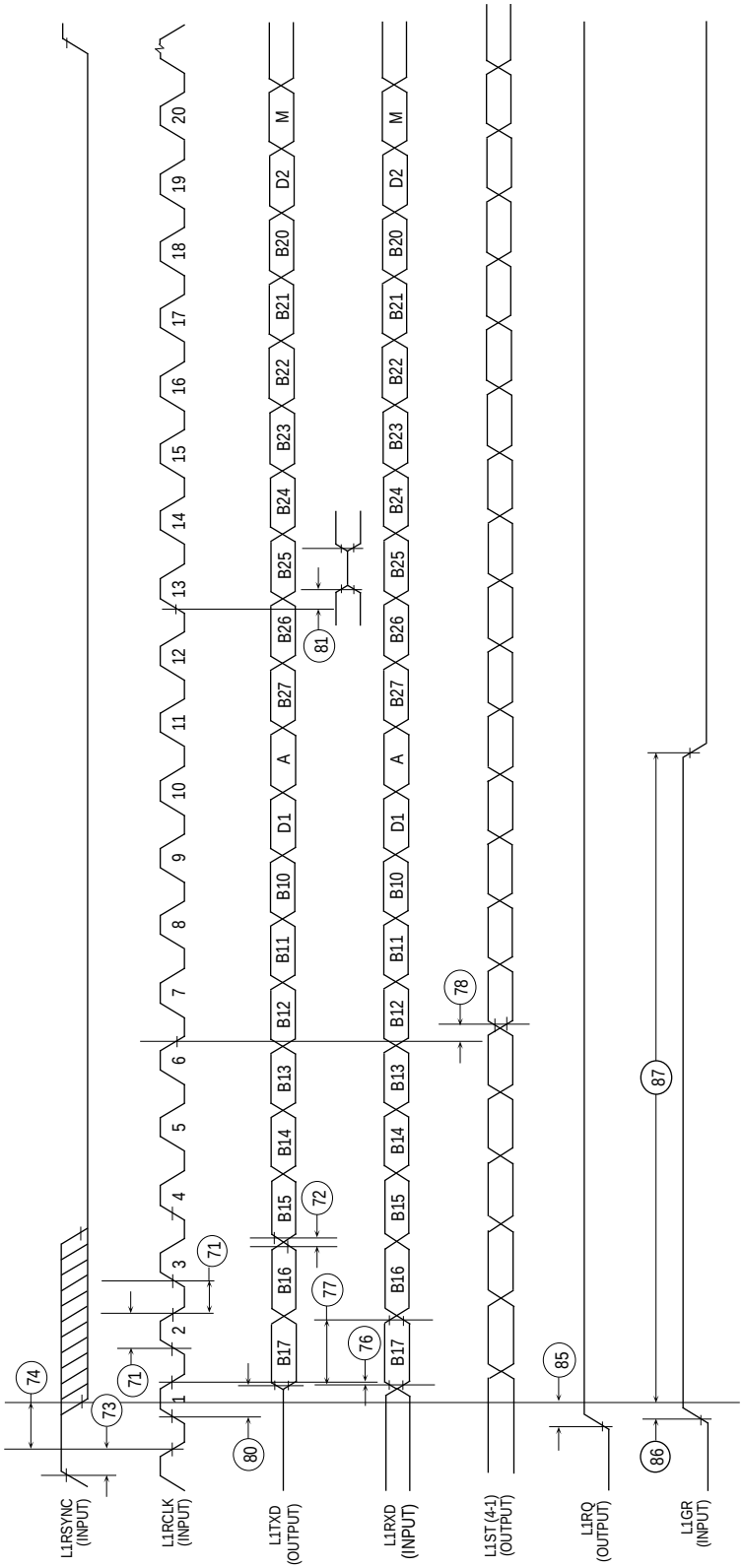
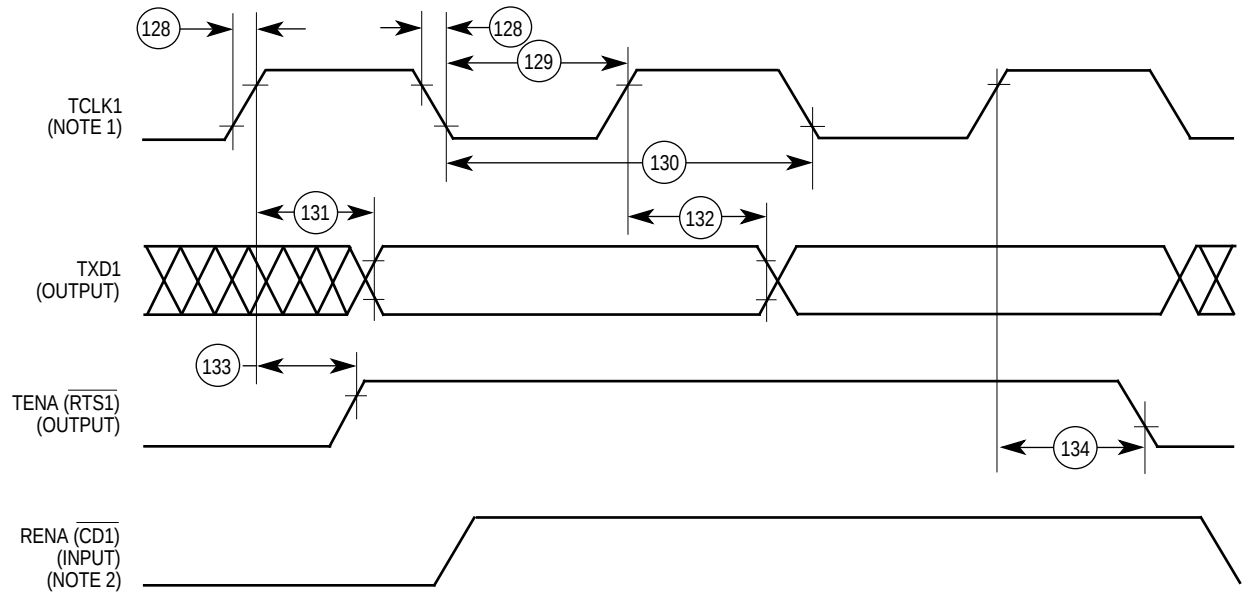


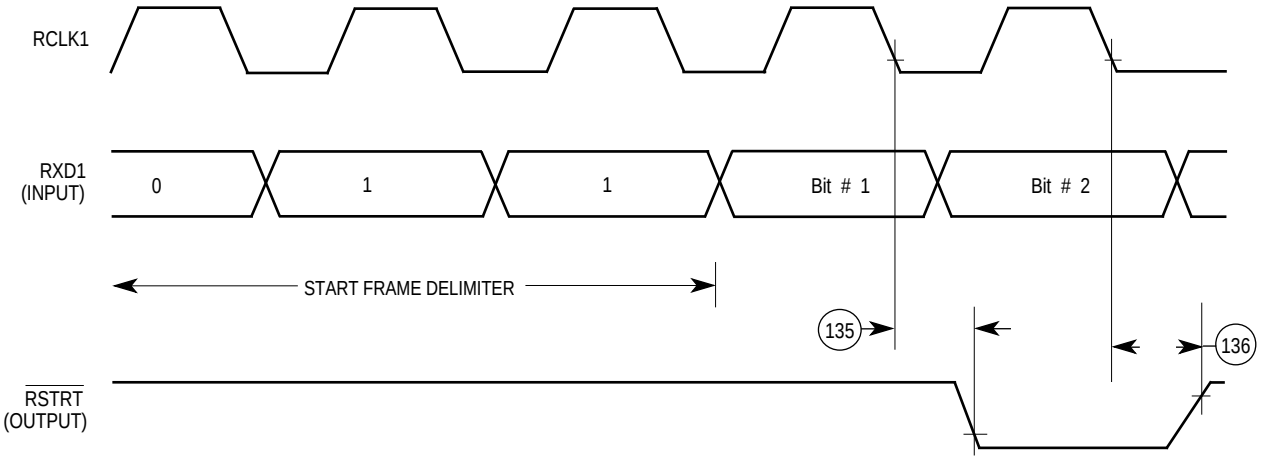
Figure 10-64. IDL Timing

Freescale Semiconductor, Inc.



NOTES:
1. Transmit clock invert (TCI) bit in GSMR is set.
2. If RENA is deasserted before TENA, or RENA is not asserted at all during transit, then CSL bit is set in the buffer descriptor at the end of frame transmission.

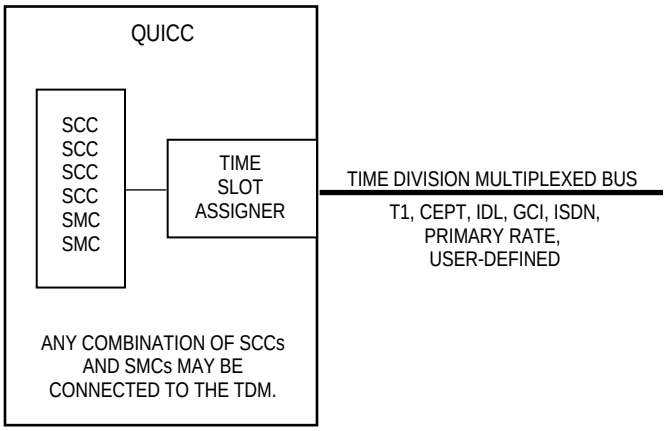
Figure 10-70. Ethernet Transmit Timing



NOTE: Valid for the ethernet protocol only.

Figure 10-71. CAM Interface Receive Start Timing

Freescale Semiconductor, Inc.

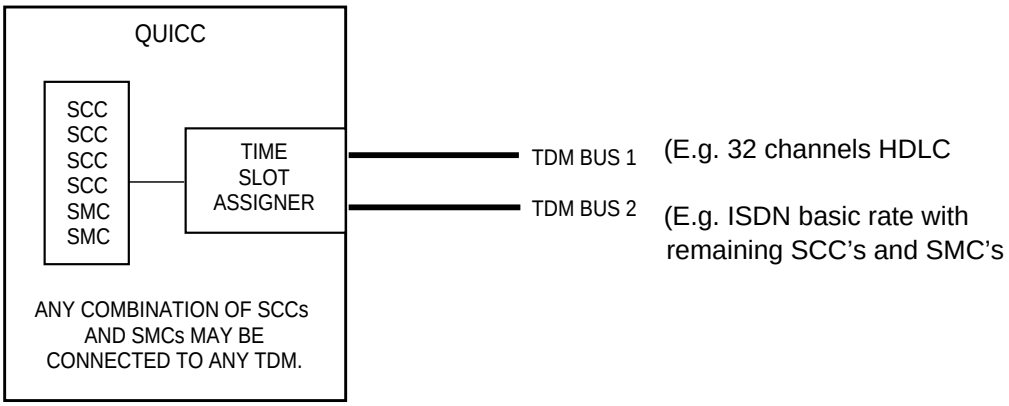


NOTE: Independent receive and transmit clocking, routing, and syncs are supported.

Figure D-3. Serial Channel to TDM Bus Implementation

Figure D-4 shows that the QUICC32 time-slot assigner can support two TDM buses. Each TDM bus can be of a different format—for example, one TDM can be a T1 line, and one can be a CEPT line. Also this technique could be used to bridge frames from basic rate ISDN to a T1/CEPT line, etc.

The QUICC32 can route channels to and from the QMC protocol to the two different TDM buses in any combination.



NOTE: Two TDM buses may be simultaneously supported with the time slot assigner.

Figure D-4. Dual TDM Bus Implementation

Figure D-5 shows a TDM application having one line termination device that extracts clocks and frame sychronization pulses. For T1/E1 line termination, devices exist that perform this function. Alternatively, this can be achieved by a DSP from the Motorola 56K family.

For line termination the QUICC32 is capable of handling up to 32 channels and it may be sufficient to omit the block labeled “Other PCM line devices”. If it is desired to incorporate other PCM line devices in the system as shown in Figure D-5, the QUICC32 can provide strobe signals to other devices that do not have a built-in time slot assigner.

Freescale Semiconductor, Inc.