
**Welcome to E-XFL.COM**

## Understanding Embedded - Microprocessors

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

## Applications of Embedded - Microprocessors

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

| Details | |
|---|---|
| Product Status | Obsolete |
| Core Processor | CPU32+ |
| Number of Cores/Bus Width | 1 Core, 32-Bit |
| Speed | 33MHz |
| Co-Processors/DSP | Communications; CPM |
| RAM Controllers | DRAM |
| Graphics Acceleration | No |
| Display & Interface Controllers | - |
| Ethernet | 10Mbps (1) |
| SATA | - |
| USB | - |
| Voltage - I/O | 5.0V |
| Operating Temperature | 0°C ~ 70°C (TA) |
| Security Features | - |
| Package / Case | 357-BBGA |
| Supplier Device Package | 357-PBGA (25x25) |
| Purchase URL | https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68mh360vr33lr2 |

**Home Page:**
www.freescale.com
**email:**
support@freescale.com
**USA/Europe or Locations Not Listed:**
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
(800) 521-6274
480-768-2130
support@freescale.com
**Europe, Middle East, and Africa:**
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com
**Japan:**
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku
Tokyo 153-0064, Japan
0120 191014
+81 2666 8080
support.japan@freescale.com
**Asia/Pacific:**
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate,
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com
**For Literature Requests Only:**
Freescale Semiconductor
Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
(800) 441-2447
303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor
@hibbertgroup.com



# CONTENTS

# TABLES

# Chapter 1
# Overview

This chapter gives an overview of the QMC protocol including some example applications.

## 1.1 The QMC (QUICC Multichannel Controller)

The QMC protocol emulates up to 64 logical channels within one SCC (serial communication controller) using the same time-division-multiplexed (TDM) physical interface. This multichannel protocol is implemented using the CPM ROM space and additional hardware; it is not a downloadable microcode.

The standard QUICC family members (MC68360[1], MPC860[2], etc.) work in TDM applications but can only support one logical channel per SCC. The parts currently supporting the QMC protocol are a superset to the following devices:

- MC68MH360 is a superset of the MC68EN360[3]
- MPC860MH is a superset of the MPC860EN
- MPC860DH is a superset of the MPC860DE

The QMC parts are pin-compatible with their respective family members. With minor adjustments, they can be used in identical applications such as primary rate ISDN support.

## 1.2 Introduction

Ideal for E1/T1 applications, the QMC protocol can multiplex any 64-channel combination of subgroups to one TDM interface.

Each of the channels can be separately programmed either to perform HDLC formatting/deformatting or to act as a transparent channel.

Both of the SI serial interfaces (for example, $TDM_a$ or $TDM_b$) can be dedicated to the QMC protocol. The SI transfers the whole frame to an SCC[4]. Using the CPM RISC, the SCC

[1]MC68360 is trademarked as the QUICC.
[2]MPC860 is trademarked as the PowerQUICC.
[3]On the MC68MH360, protocol support for Centronics and BISYNC have been removed to create space for the QMC microcode.
[4]This is the normal operating mode; however, it is possible to split the TDM stream over several SCCs.

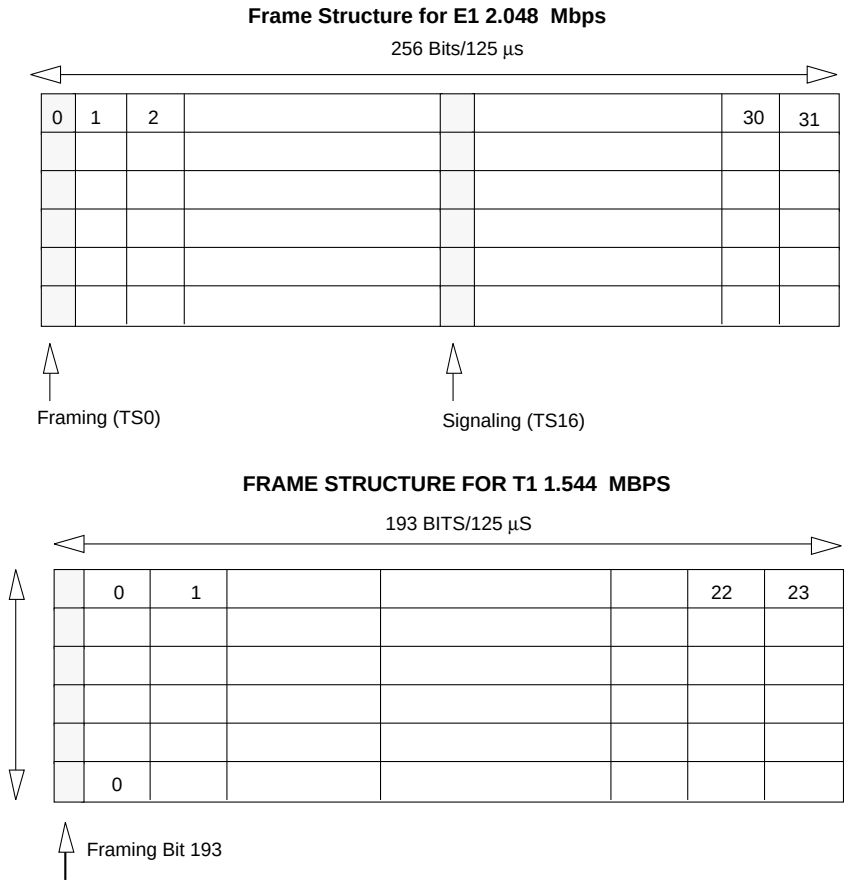



**Figure 1-8. Frame Structures for E1/CEPT and T1 TDM Interfaces**

For any station to receive and transmit on a TDM line, it is necessary for it to determine the correct time slot boundary. The service provider or the PTT provides a 4-wire interface with a continuous bit stream coming down the line. The T1 and E1 have information embedded in the data stream that delineates frames. The bit pattern in position 193 in T1 over a period of several frames establishes a synchronization pattern. A station may have the capability to search for this pattern and thus find the correct time for frame synchronization. In a similar way, time slots 0 and 16 are reserved not only for synchronization but also for signaling in the E1 interface.

Depending on its capability, a node can either extract this synchronization information or it can be supported by framer and time slot assigner devices.

## 2.1 QMC Memory Structure

Figure 2-2 shows how data is addressed by the QMC protocol. It discusses addressing the dual-ported RAM to access data within the buffers.

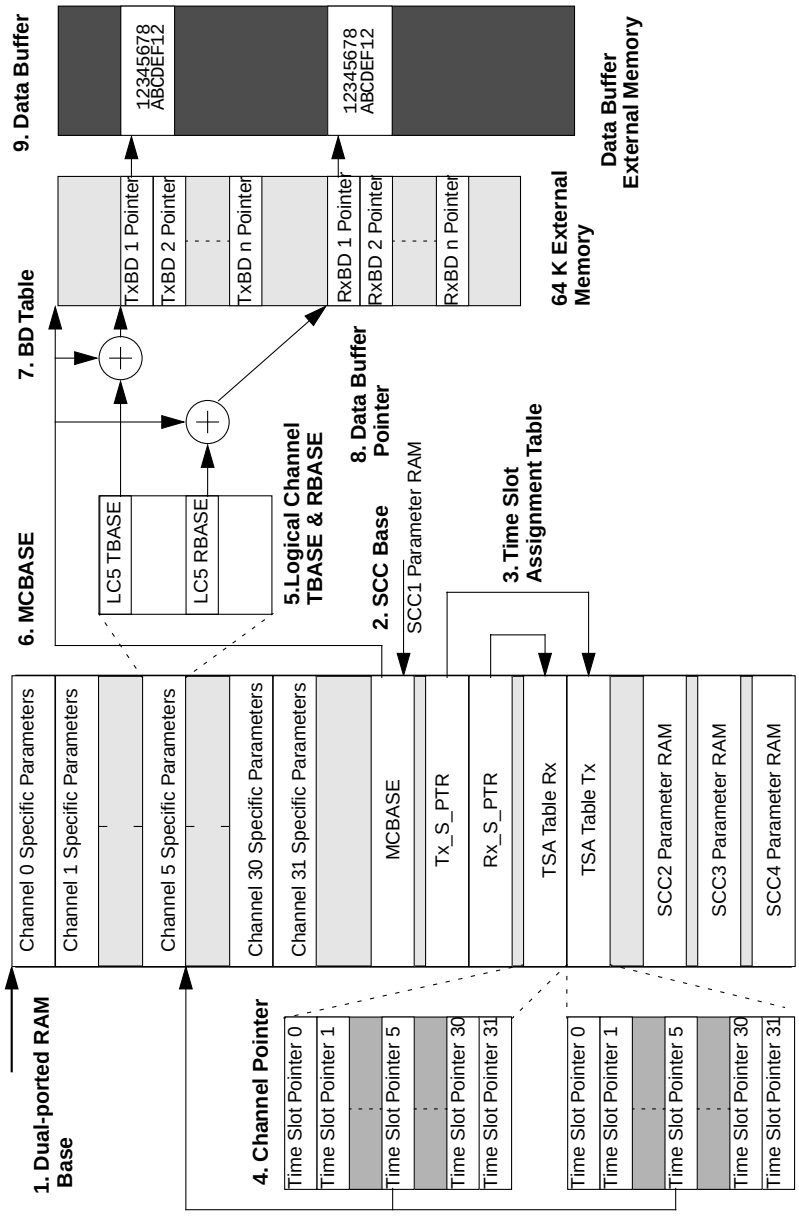
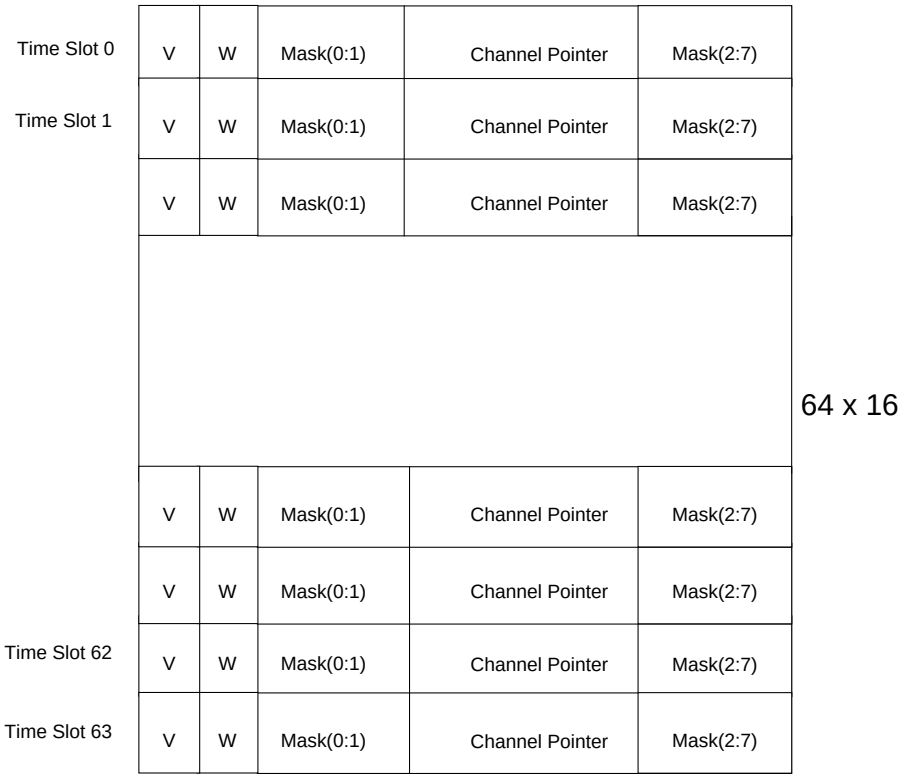



**Figure 2-2. QMC Memory Structure**

If the transmitter and receiver have the same mapping then it is possible to use a common time slot assignment table. This is initialized by setting both Tx_S_PTR and Rx_S_PTR to SCC Base + 20. For 64-channel support it is suggested to use common Rx and Tx parameters. The time slot assignment table will then also be common and have 64 entries starting at SCC Base + 20; see Figure 2-4.




**Note**: For the 68360, the bit numbering is reversed. See Appendix A for more information.

**Figure 2-4. Time Slot Assignment Table for 64-Channel Common Rx and Tx Mapping**

## 2.3 Multiple SCC Assignment Tables

Assume a scenario as depicted in Figure 2-5. A 2.048-Mbps TDM link is fed directly into the TSA. Within the SI RAM, the even channels (byte-wide) are muxed to SCC3 and the odd channels are muxed to SCC2. This arrangement is used to spread the load over two SCCs. This effectively doubles the FIFO depth on the QMC protocol. Time slots are switched to alternate SCCs to avoid data bursts that may stress the FIFOs. Each SCC sees a continuous bitstream without any gaps as described earlier.

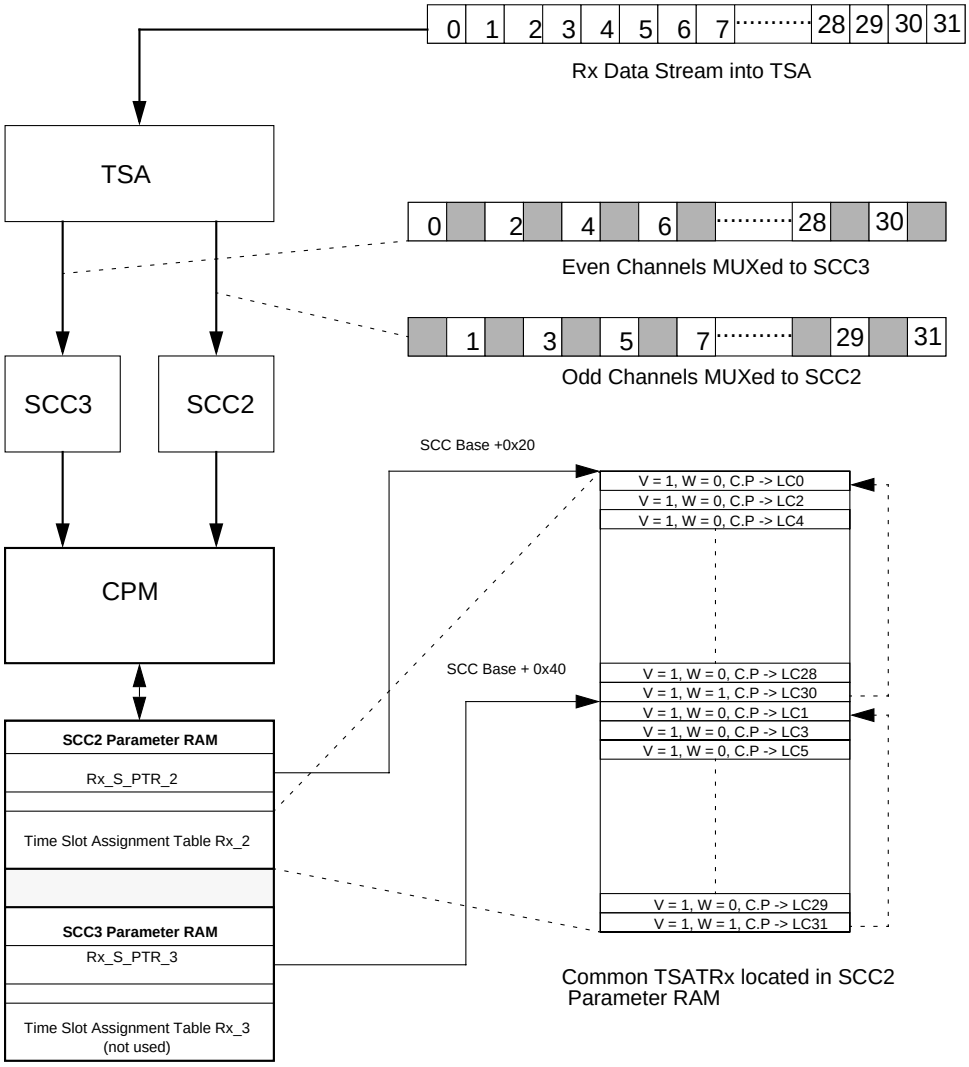



**Figure 2-5. Rx Time Slot Assignment Table for 32 Channels over Two SCCs**

### Note

It is important that multiples of bytes are routed to each SCC to delineate between time slots. Unused bits shall be routed to the SCC and be masked in the time slot assignment table.

## 2.4 Channel-Specific Parameters

The channel-specific parameters are located in the lower part of the dual-ported RAM. Each channel occupies 64 bytes of parameters. Physical time slots can be matched to logical channels in several combinations. Unused logical channels leave a hole in the channel-specific parameters that can be used for buffer descriptors for the other SCCs.

The channel-specific area determines the operating mode—HDLC or transparent. Several entries take on different meanings depending on the protocol chosen.

### 2.4.1 Channel-Specific HDLC Parameters

Table 2-4 describes the channel-specific HDLC parameters. Boldfaced parameters must be initialized by the user.

**Table 2-4. Channel-Specific HDLC Parameters**

| Offset | Name | Width (Bits) | Description |
|---|---|---|---|
| 00 | **TBASE** | 16 | Tx buffer descriptor base address—Offset of the channel's transmit buffer descriptor table relative to MCBASE, host-initialized. See Figure 2-2. |
| 02 | **CHAMR** | 16 | Channel mode register. See Section 2.4.1.1, "CHAMR—Channel Mode Register (HDLC)." |
| 04 | **TSTATE** | 32 | Tx internal state —TSTATE defines the internal Tx state.<br>Host-initialized to 0x3800_0000—FC = 8, Motorola mode for MH360.<br>Host-initialized to 0x3000_0000— AT = 0, Motorola mode for 860MH.<br>Initialize before enabling the channel. See Section 2.4.1.2, "TSTATE—Tx Internal State (HDLC)." |
| 08 | — | 32 | Tx internal data pointer—Points to current absolute address of channel. |
| 0C | **TBPTR** | 16 | Tx buffer descriptor pointer (host-initialized to TBASE before enabling the channel or after a fatal error before reinitializing the channel again)—Offset of current BD relative to MCBASE. See Table 2-1. MCBASE + TBPTR gives the address for the BD in use. |
| 0E | — | 16 | Tx internal byte count—Number of remaining bytes |
| 10 | TUPACK | 32 | (Tx Temp) Unpack 4 bytes from 1 long word |
| 14 | **ZISTATE** | 32 | Zero-insertion state (host-initialized to 0x0000_0100 for HDLC or transparent operation)—Contains the previous state of the zero-insertion state machine. |
| 18 | TCRC | 32 | Temp transmit CRC—Temp value of CRC calculation result |
| 1C | **INTMSK** | 16 | Channel's interrupt mask flags—See Section 2.4.1.3, "INTMSK—Interrupt Mask (HDLC)." |
| 1E | BDFlags | 16 | Temp |
| 20 | **RBASE** | 16 | Rx buffer descriptor offset (host-initialized)— Defines the offset of the channel's receive BD table relative to MCBASE (64-Kbyte table). See Figure 2-2. |

C4 is a 2-byte pattern TS8, TS7, so TSn = 8

Rx Byte: (8+1) * 2 = 16
As x = 1, TSn + x = 7, so
TX Byte: (7 + 1) * 2 = 16

C5 is a 2-byte pattern TS19, TS23, so TSn = 19

Rx Byte: (19 + 1) * 2 = 40
As x = 1, TSn + x = 23, so
TX Byte: (23 + 1) * 2 = 0

C6 is a 2-byte pattern TS23, TS19, so TSn = 23

Rx Byte: (23 + 1) * 2 = 0
As x = 1, TSn + x = 19, so;
TX Byte: (19 + 1) * 2 = 40

C7 is a 4-byte pattern TS20, TS23, TS8, TS9 & TS19, so TSn = 20

Rx Byte: (20 + 1) * 2 = 42
As x = 5, TSn + x = 19, so
TX Byte: (19 + 1) * 2 = 40

C8 is a 4-byte pattern TS8, TS9, TS19, TS20 & TS23, so TSn = 8

Rx Byte: (8 + 1) * 2 = 18
As x = 5, TSn + x = 23, so
TX Byte: (23 + 1) * 2 = 0

C9 is a 4-byte pattern TS19, TS20, TS23, TS8 & TS9, so TSn = 19

Rx Byte: (19 + 1) * 2 = 40
As x = 5, TSn + x = 9, so
TX Byte: (9 + 1) * 2 = 20

**NOTE**

Case C1 and C2 can be used as described above. A more elegant solution for single time slot applications is to have the SYNC bit cleared (0) in the channel mode register. Both scenarios produce the same result.

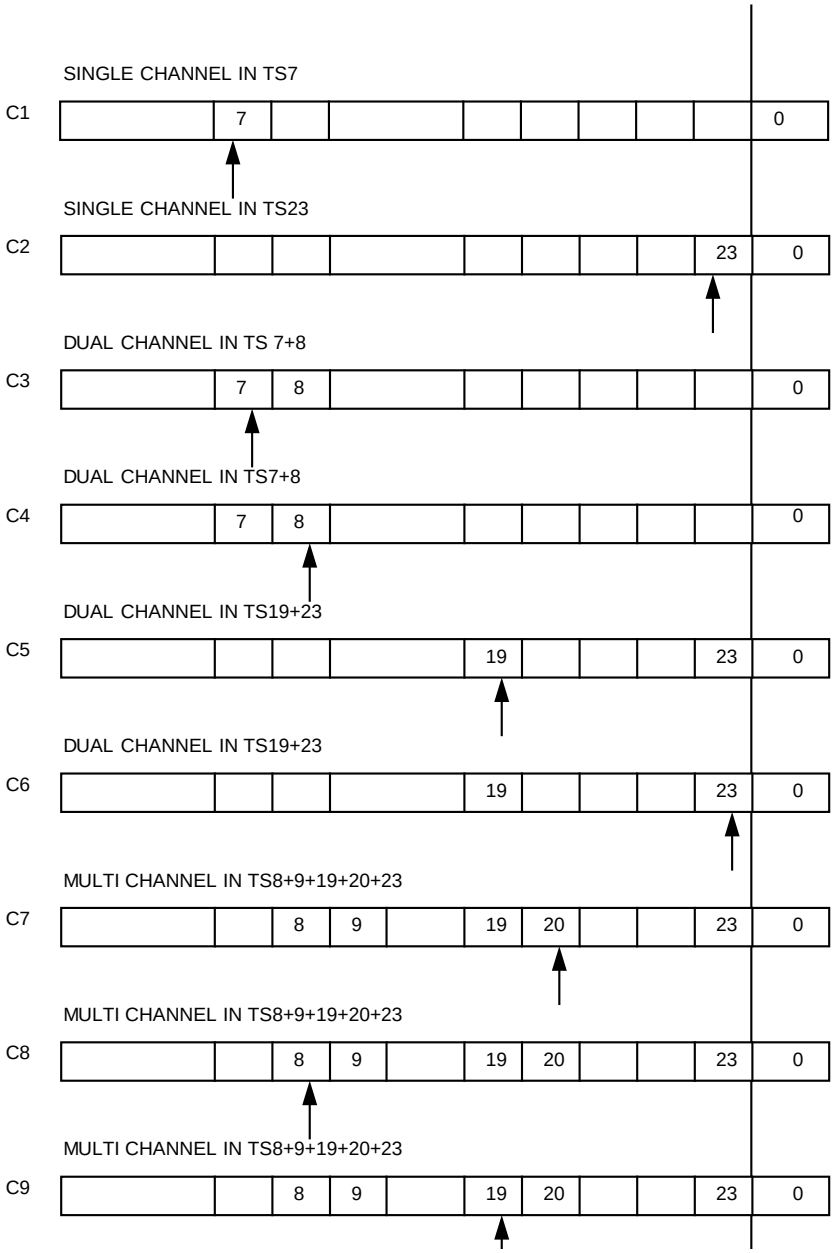
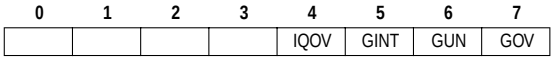



**Figure 2-14. Examples of Different T1 Time Slot Allocation**

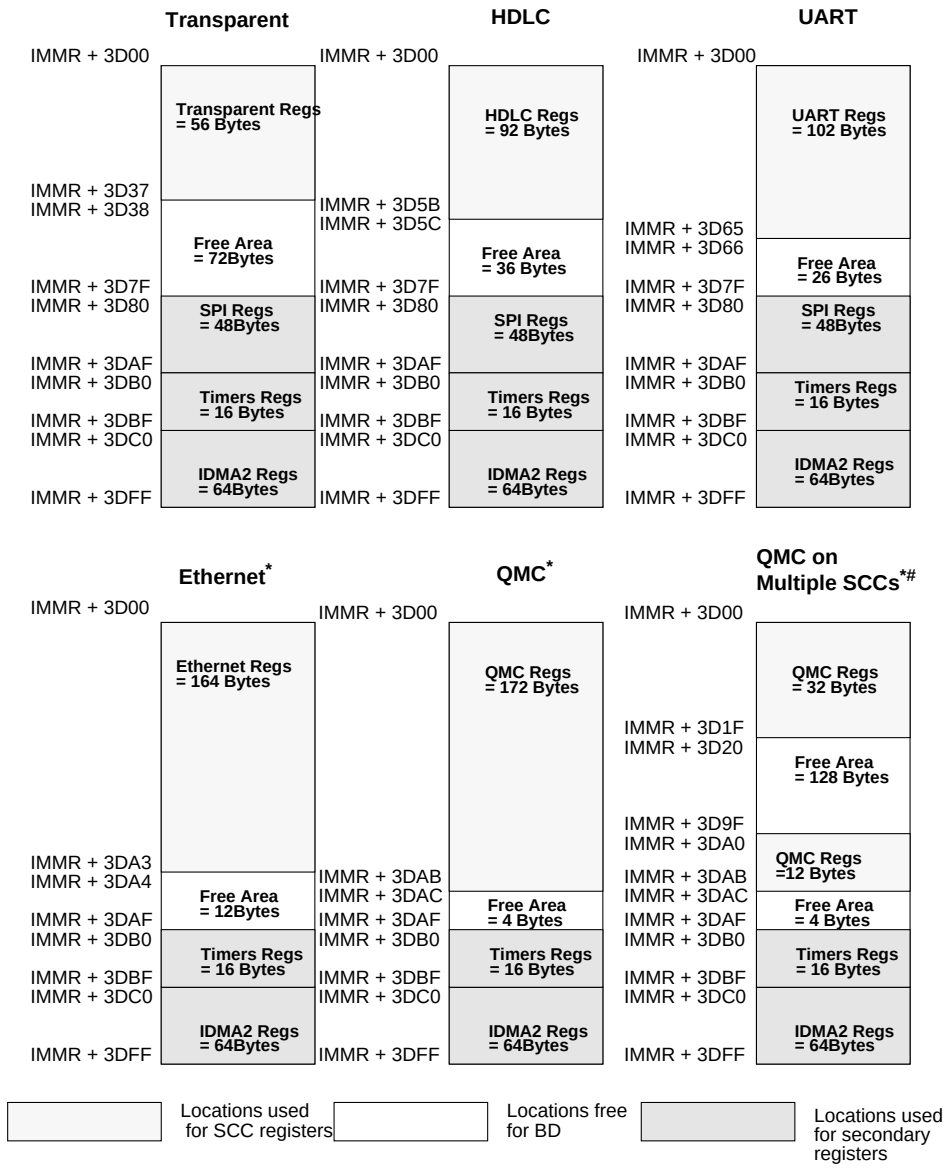



**Note**: For the 68360, the bit numbering is reversed. See Appendix A for more information.

**Figure 4-2. SCC Event Register**

**Table 4-1. SCC Event Register Field Descriptions**

| Field | Name | Description |
|---|---|---|
| 0–3 | — | Reserved |
| 4 | IQOV | Interrupt table (interrupt queue) overflow<br>0 No interrupt table overflow has occurred.<br>1 An overflow condition in the circular interrupt table occurs (and an interrupt request is generated). This condition occurs if the RISC processor attempts to write a new interrupt entry into an entry that was not handled by the host. Such an entry is identified by V = 1.<br>This entry is cleared by the software immediately after entering the interrupt routine. When this occurs, the last interrupt is lost and not overwritten on the first entry. |
| 5 | GINT | Global interrupt<br>0 No global interrupt has occurred.<br>1 This flag indicates that at least one new entry in the circular interrupt table has been generated by the QMC. The host clears GINT by writing a 1 to its location in SCCE. After clearing it, the host reads the next entry from the circular interrupt table, and starts processing a specific channel's exception.<br>The user must make sure that no more valid interrupts are pending in the interrupt table after clearing the GINT bit, before performing the RTE to avoid deadlock. This procedure ensures that no pending interrupts exist in the queue. |
| 6 | GUN | Global transmitter underrun<br>0 No global transmitter underrun has occurred.<br>1 This flag indicates that an underrun occurred in the SCC's transmitter FIFO. This error is fatal since it is unknown which channel(s) are affected. Following the assertion of the GUN bit in the SCCE, the QMC stops transmitting data on all channels. The TDM Tx line goes into idle mode. This error affects only the transmitter; the receiver continues to work.<br>After initializing all the individual channels, the host may resume transmitting. If enabled in the SCCM, an interrupt request is generated when GUN is set. The host may clear GUN by writing 1 to its location in the SCCE. |
| 7 | GOV | Global receiver overrun<br>0 No global receiver overrun has occurred.<br>1 This flag indicates that an overrun occurred in the SCC's transmitter FIFO. This error is fatal since it is unknown which channel(s) are affected. Following the assertion of the GOV bit in the SCCE, the QMC stops receiving data on all channels. Data is no longer written to memory. This error affects only the receiver; the transmitter continues to work.<br>After initializing all the individual channels, the host may resume receiving. If enabled in SCCM, an interrupt request is generated when GOV is set. The host may clear GOV by writing 1 to its location in the SCCE. |




**#—TSATRx & TSATTx are mapped to an alternative SCC's parameter RAM.**
***—SPI is not available in these configurations due to memory conflict.**

**Figure 5-13. MPC860MH SCC2 Parameter RAM Usage**

## 9.2 MSC Microcode Operation

In normal operation (without the MSC microcode), the QMC protocol allows specific bits in an 8-bit time slot to be masked to create a single subchannel per SCC. A problem arises when multiple subchannels are multiplexed within a single time slot as in GSM (global system for mobile communications) where four 16-Kbps subchannels are multiplexed into a single 64-Kbps channel over a 2.048-Mbps A bis link. A brute-force solution routes the separate subchannels to different SCCs, consuming all four SCCs for the single TDM link as shown in Figure 9-1. Each SCC filters out one of the four 2-bit subchannels in time slot 2 (TS2) using a unique mask located in its time slot assignment tables (TSATRx/TSATTx). With the MSC microcode, subchannels can be regenerated using only one SCC.

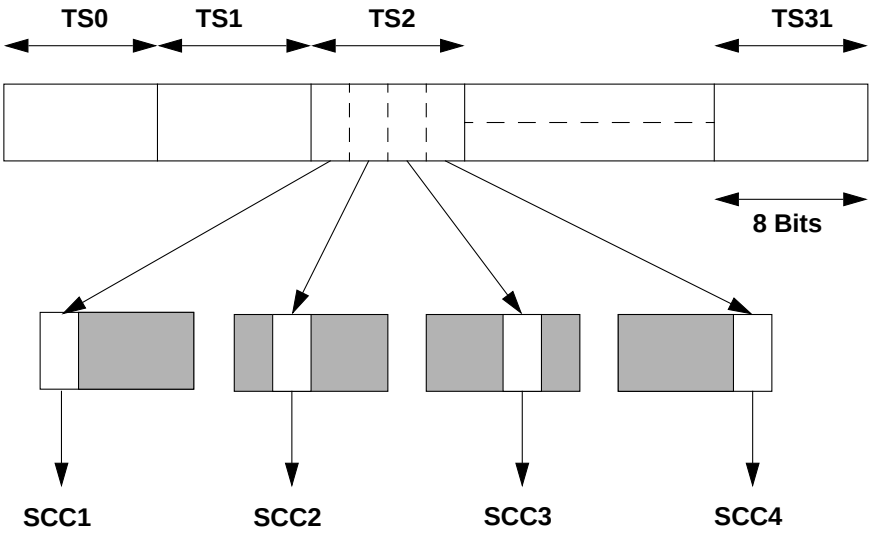
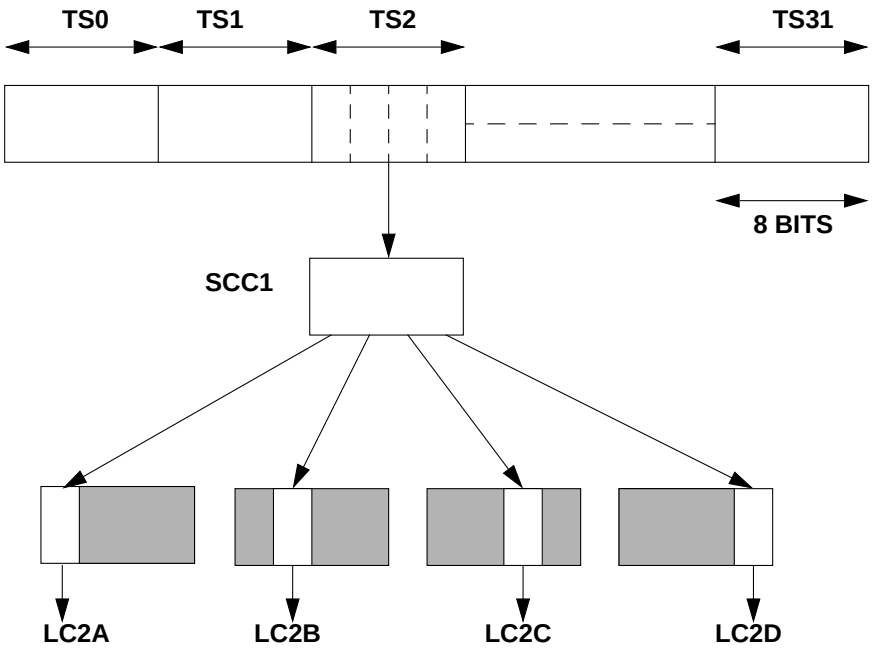



**Figure 9-1. Two-Bit Subchannel Implementation without MSC Microcode**

The MSC microcode enables an 8-bit time slot to be split into multiple, bit-resolution subchannels. The microcode applies user-defined masks in a time slot assignment table entry to subdivide a given channel. Bit 11 of a table entry is now called the L bit to mark the last subchannel of a given time slot. Figure 9-2 shows the MSC microcode solution to the above GSM problem. Again in this example, time slot 2 contains four 2-bit channels, but now the full time slot can be routed to a single SCC and split into subchannels within the time slot assignment tables.

**NOTE**

As the L bit (bit 11 of a time slot assignment table entry) is taken from the original channel pointer field, the addressing capability of the QMC is reduced from 6 bits (64 channels) to 5 bits (32 channels) for receive and transmit.




**Masking performed by a single SCC to create four 2-bit channels**

**Figure 9-2. Two-Bit Subchannel Implementation with MSC Microcode**

## 9.3 Programming the MSC Protocol

The MSC protocol configuration is identical to the QMC configuration with the exception of the time slot assignment tables. Figure 9-3 shows the addition of the L bit needed for the MSC configuration. Note that it is possible to have up to 32 channels coming from any permutation of the time slots.

transmit time slot 1, but virtual channel 5 could be assigned to receive time slot 1. Therefore, global loopback would cause the data transmitted on channel 1 to be received on channel 5.

- For loopback on an individual time slot, set bit 15 of the corresponding entry in the SIRAM.

For the last two methods, use a common transmit/receive clock and transmit/receive sync pulses. Also in these last two methods, if the loopbacked data is to be received on the same virtual channel, make sure the transmit and receive portions of your SI RAM entries match.

Q: Does the MH360 support signalling system 7 (SS-7) on all 32 channels?

A: No. The MH360 does not support SS-7. Run the SS-7 microcode on a normal QUICC to get 4 channels of SS-7 support.

Q: Is any extra support logic required to attach an MH360 to an ISDN primary rate line?

A: Yes, a T1 transceiver and framer unit will be needed. Manufacturers include Dallas Semiconductor, Mitel, and Level One.

Q: Can the MH360 be used as an ISDN basic rate terminator?

A: Yes, send multiple Motorola U Interface outputs into a single SCC using the QMC controller and the Motorola IDL2.

Q: 360 documentation says that the time slot assigner can be used with all of the SCCs and the SMCs. However, the MH360 reference manual states that the time slot assigner works with the SCCs. Can the time slot assigner work with the SMCs?

A: Yes.

Q: What is the maximum bus latency at 25 MHz when using 32 64-Kbps channels?

A: Nine clocks. It is very important. At 25 MHz, the CPM is just able to empty the FIFO quickly enough to prevent underrun. If the part is run at 33 MHz or if only 24 channels are needed, the margin increases allowing for a better worst case.

Q: Can $TDM_b$ be used for the QMC on the MH360?

A: Yes.

Q: Can the transparent channels of the MH360 (QMC mode) synchronize on an inline data pattern?

A: No.

However, this configuration works only if the single master S/T or U interface is guaranteed to be active, constantly generating synchronized DCL and FSC signals for the network. Since the designated master transceiver cannot be guaranteed active, all transceivers must be configured in slave mode, and the problem of synchronization remains.

DCL and FSC must be generated from within the network and cannot be derived from an independent source (e.g., crystal or oscillator). However, the S/T and U interfaces, even in slave mode, provide a way to generate these signals as explained in the following sections.

### C.3.1 MC145574 S/T Interface

To facilitate the generation of timing signals required by the slave IDL interface, TCLK is provided in the S/T interface. The TCLK signal will output a clock synchronized to the received data transmitted by the NT. That clock is output only when the S/T interface is active and when the DCL and FSC signals are present (both conditions are required simultaneously). This TCLK signal can be used to provide network timing. Its frequency is selectable via the SCP.

The TCLK signal of the MC145574 is enabled by setting OR7[5]. BR13[5] and BR7[2] determine the TCLK frequency as shown in Table C-1.

**Table C-1. TCLK Frequencies Selected by BR13[5] and BR7[2]**

| BR13[5] | BR7[2] | TCLK |
|---|---|---|
| 0 | 0 | 2.56 MHz |
| 0 | 1 | 2.048 MHz |
| 1 | 0 | 1.536 MHz |
| 1 | 1 | 512 KHz |

A TCLK signal configured as a 2.048-MHz clock can be used as the DCL for the IDL bus.

Dividing the TCLK signal by 256 provides the 8-KHz frame sync FSC. Since the FSC must have a pulse between one DCL and eight DCLs in width, additional logic may be needed after the divider to generate a signal with a correct duty cycle.

Figure C-4 and Figure C-5 show a schematic and a timing diagram, respectively, for a logic design used to generate a 8-KHz FSC with a 1-DCL width pulse from a 2.048-MHz clock (TCLK).

# INDEX