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/mc68mh360vr33l

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

CONTENTS

Acronyms and Abbreviations

Table i contains acronyms and abbreviations that are used in this document.

Table i. Acronyms and Abbreviated Terms

Term	Meaning
BD	Buffer descriptor
bps	Bits per second
BRI	Basic rate interface
BRG	Baud rate generator
CPM	Communications processor module
CR	Command register
DCL	Data clock signal
FSC	Frame sync signal
GSM	Global system for mobile communications
GOV	Global receiver overrun (global error)
GUN	Global transmitting underrun (global error)
HDLC	High-level data link control
I^2C	Interprocessor-integrated controller channel
MSC	Multi-subchannel microcode
NMSI	Nonmultiplexed serial interface
QMC	QUICC multichannel controller
QUICC	QUad integrated communication controller
RCCR	RISC controller configuration register
RxBD	Receive buffer descriptor
SCC	Serial communication controller
SCCE	SCC event register
SI	Serial interface routing
SS-7	Signaling system 7
TDM	Time-division multiplexing
TSA	Time slot assigner
TSO	Time slot zero
TxBD	Transmit buffer descriptor

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.

- System interface
  - On-chip bus arbitration for serial DMAs with no performance penalty
  - Efficient bus usage (no bus usage for nonactive channels and active channels that have nothing to transmit)
  - Efficient control of the interrupts to the CPU
  - Supports external buffer descriptors table
  - Uses on-chip enlarged dual-ported RAM for parameter storage

1.4 The Time Slot Assigner and the QMC

The time slot assigner (TSA) in the MH devices is no different from the other versions. This section discusses the new possibilities when using the TSA in combination with the QMC.

The QMC protocol can be executed in nonmultiplexed serial interface (NMSI) mode, but the usual operating mode takes advantage of the programmable time slot assigner,.

A frame synchronization pulse alerts the time slot assigner to start counting clock pulses. The user programs what bits are routed to the different internal serial channels. The TSA is an intelligent multiplexer that restarts its sequence on every frame synchronization pulse.

External strobe signals allow other devices that do not have built-in time slot assigner functions to participate in the TDM interface. This is very useful when interfacing to the MC68302 or other telecommunication devices like codecs.

The time slot assigner is not limited to standard TDM lines. It is a flexible, programmable device that allows the user to route any combination of bits and bytes to any channel. For example, the user can transmit 3 bits from SCC2, skip 12 bytes, and then transmit another 17 bits from SCC1. This routing must be programmed into the TSA memory. The complexity of the routing is limited only by the number of program entries in the TSA.

Ideal for TDM bridging applications, the MC68MH360 and MPC860MH have two independent time slot assigners and physical interfaces. A complete set of independent receive and transmit clock signals, as well as independent synchronization signals, are available for each TDM.

1.5 The Serial Interface (SI)

Functions such as frame synchronization, loopback, echo, and inverted signals are performed in the serial interface and cannot be achieved in NMSI mode. It is recommended to use the serial interface even if only one SCC is used for the TDM bus.

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.

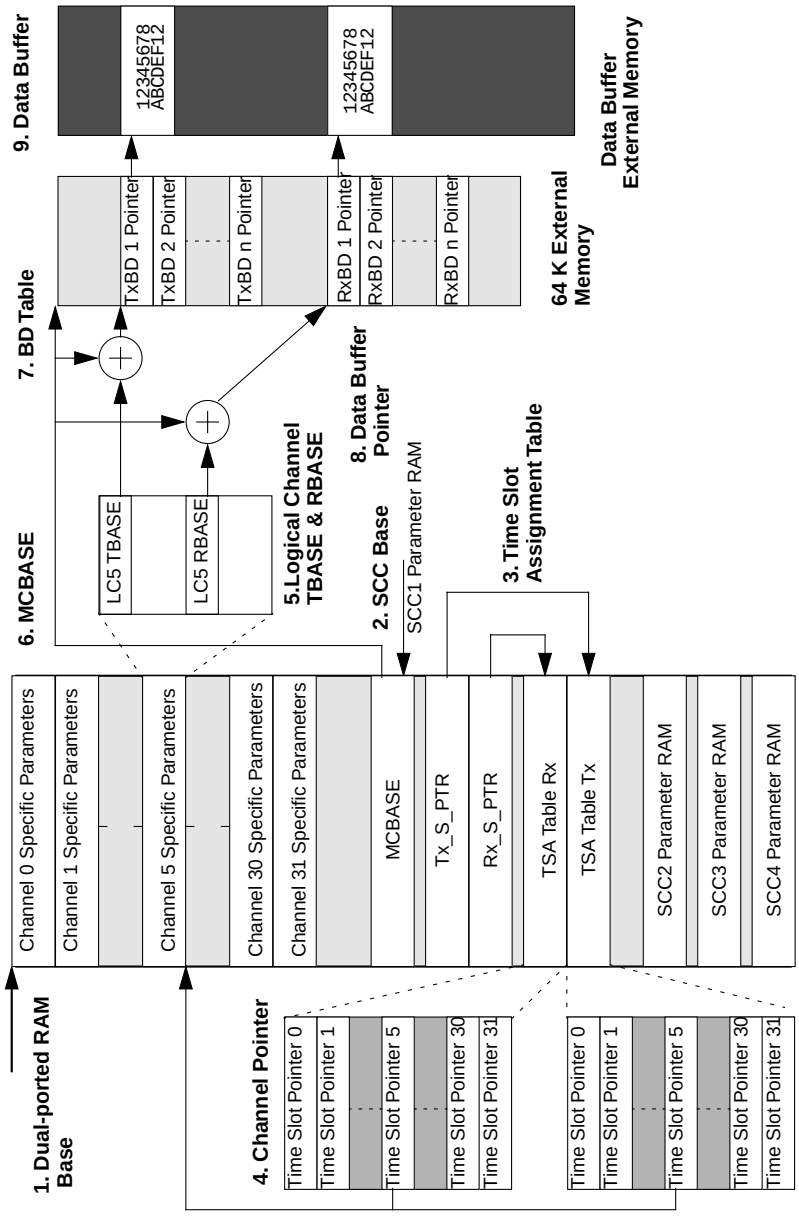


Figure 2-2. QMC Memory Structure

In Figure 2-5, each SCC has its own pointer, Rx_S_PTR_2 and Rx_S_PTR_3, addressing SCC2's time slot assignment table. This table only needs to be present once in one of the SCC2's global parameter area. Rx_S_PTR_2 points to the start of the table, address SCC base + 20. The 16 logical channels from SCC2 are located in the first 16 entries of the table. The entry for logical channel 30 has the wrap bit (W) set, causing the CPM to wrap back to logical channel 0 on reception of the next byte routed to SCC2. Rx_S_PTR_3 addresses SCC base + 40, the start of the 16 entries for SCC3. The entry for logical channel 31 has the wrap bit (W) set, causing the CPM to wrap back to logical channel 1 on reception of the next byte routed to SCC3. Each entry within the table has a channel pointer to a logical channel. It is important that different SCCs do not point to the same logical channel.

The TSATTx is also located in SCC2's parameter RAM. This means that the area reserved for the TSA tables in SCC3's parameter RAM is free for alternative use.

A second scenario is depicted in Figure 2-6. A 4.096-Mbps TDM link is fed directly into the TSA. Again, 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. Another reason this method may be used is to facilitate separate routing for the Rx and Tx logical channels. This requires two 64-entry tables that require 256 bytes, but only 128 bytes are allocated in the parameter RAM of an SCC for time slot assignment tables. In this case, the Rx table is located in SCC2's parameter RAM, and the TX table is located in SCC3's parameter RAM, making most efficient use of memory.

Changes on the fly are easily accomplished by setting or clearing the valid bit for each time slot. Changes to the mask bits can also be made on the fly. This does not cause any problems to the QMC microcode itself, but may cause protocol errors on the channel in question depending on when this change is done.

It is possible to have a time slot assignment table for every SCC in its corresponding RAM page and have all of the TDM routed to the different SCCs. This gives the user a very flexible system that can be changed on the fly without disconnecting the TDM interface. In this case the user must ensure that no collisions occur on the transmit lines from several SCCs.

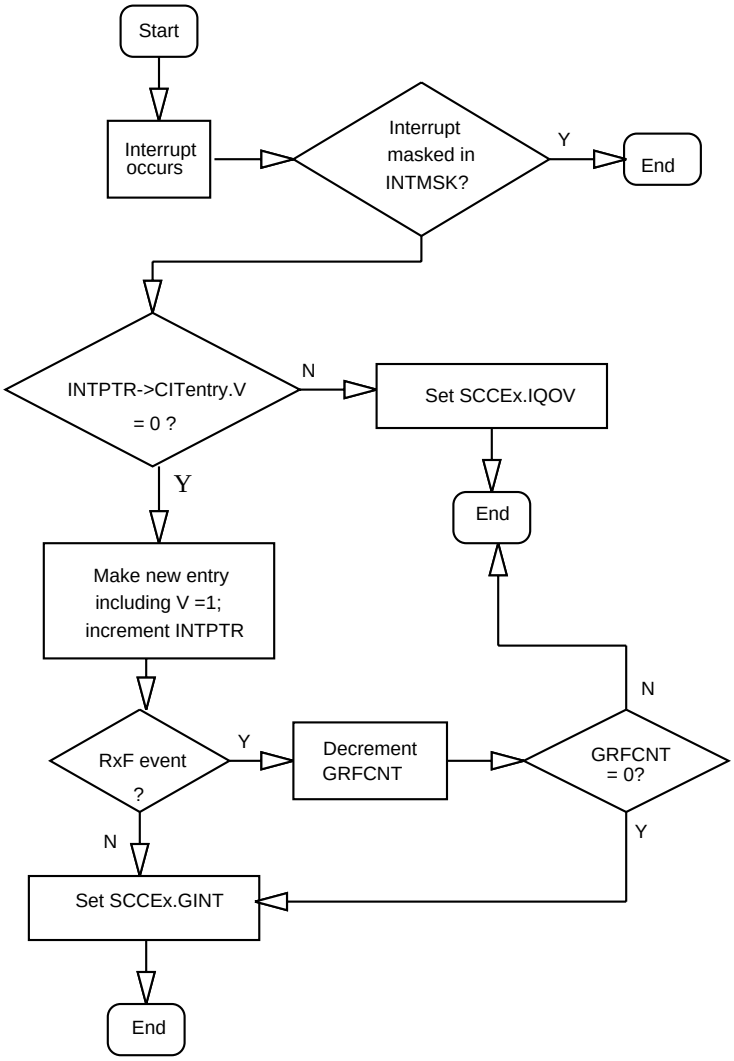
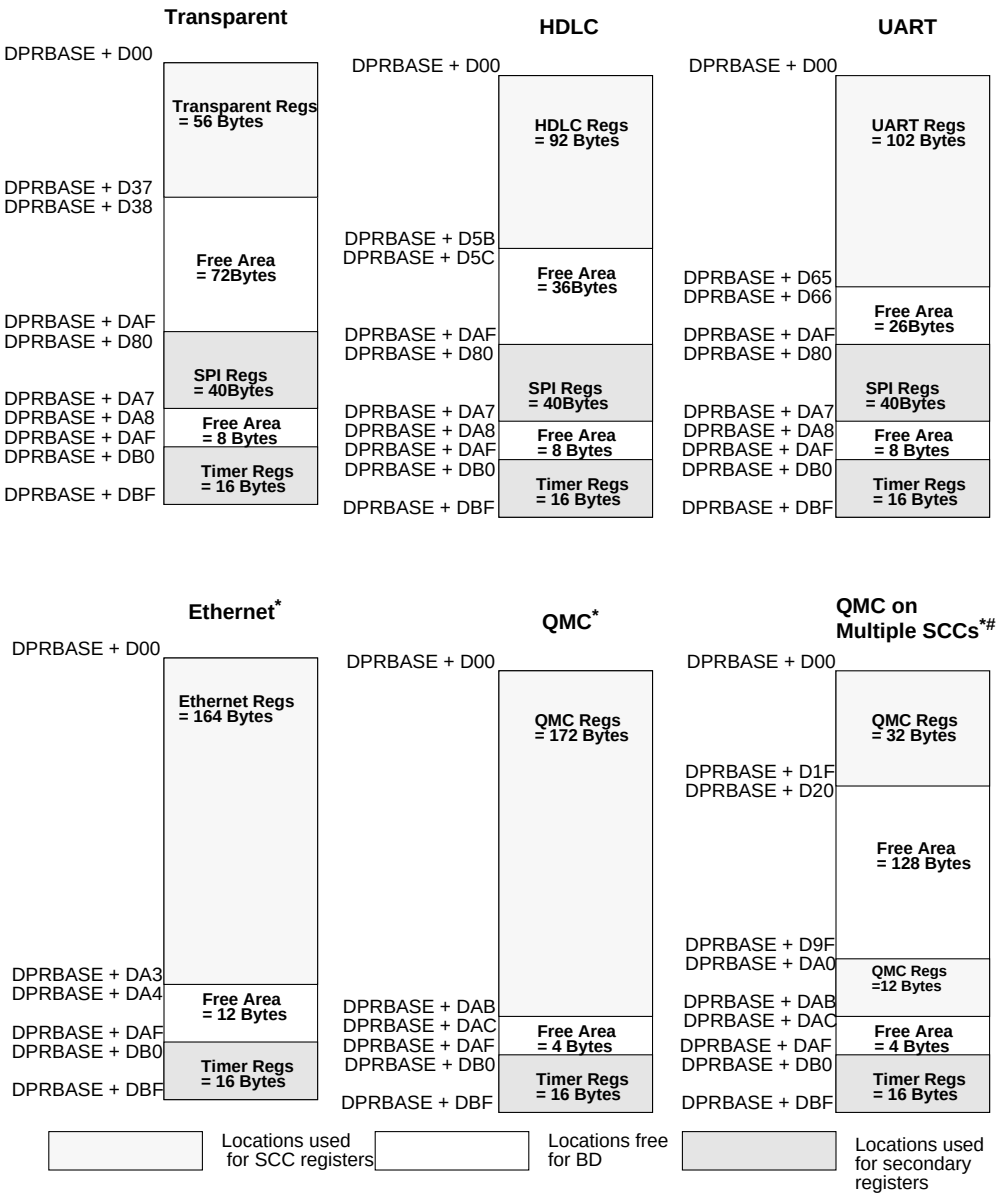


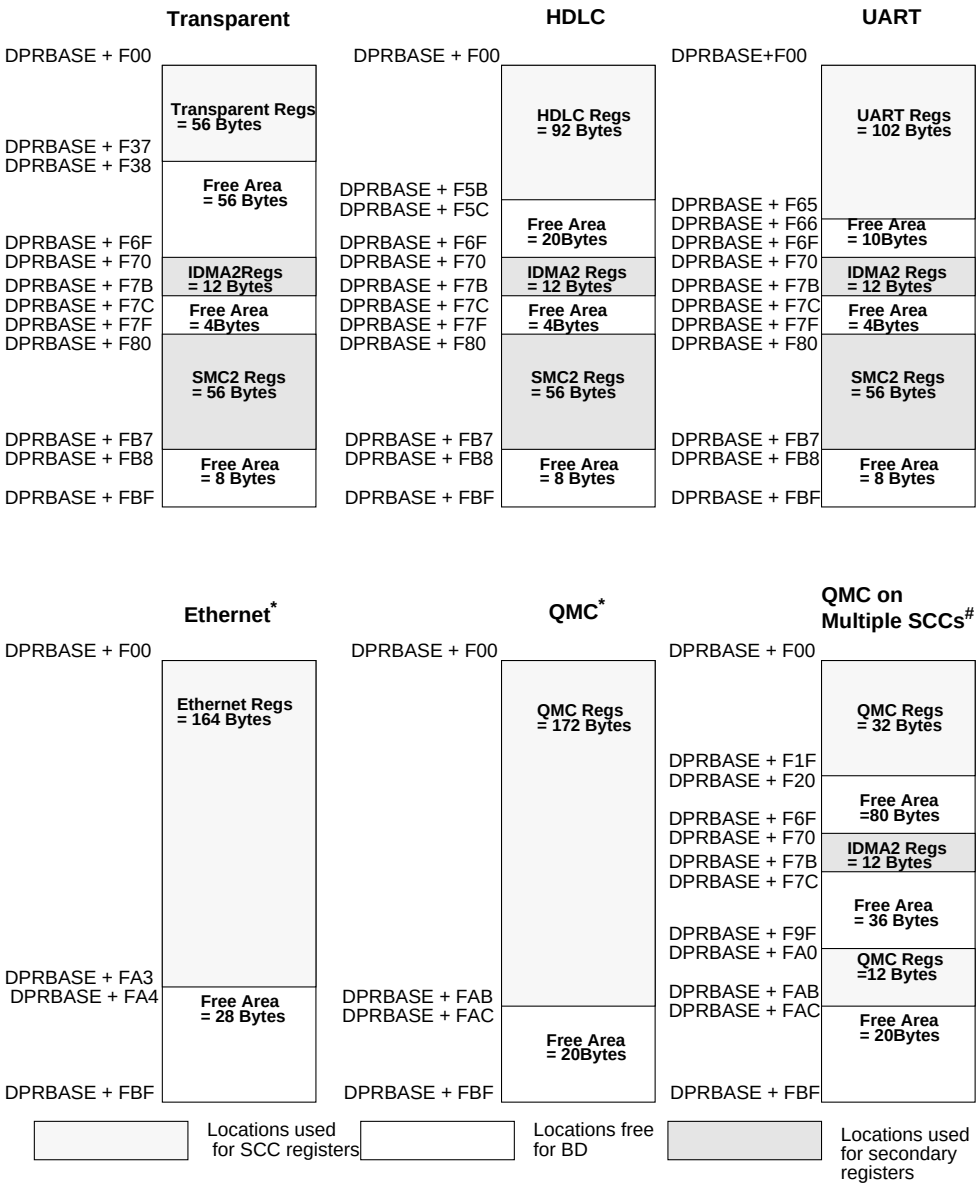
Figure 4-5. Channel Interrupt Flow



#—TSATRx & TSATTx are mapped to an alternative SCC's parameter RAM.

***—SPI not available in these configurations due to memory conflict.**

Figure 5-8. MC68MH360 SCC2 Parameter RAM Usage



#—TSATRx & TSATTx are mapped to an alternative SCC's parameter RAM, SMC2 is not available in this configuration due to memory conflict.

***—SMC2 and IDMA2 are not available in these configurations due to memory conflict.**

Figure 5-10. MC68MH360 SCC4 Parameter RAM Usage

5.3.3 MPC860MH Internal Memory Structure

Figure 5-11 shows the internal memory structure of the MPC860MH. To support 32 channels on the MP860, only 2-Kbyte dual-ported RAM is needed for channel-specific parameters. Each logical channel occupies 64 bytes; thus 32 channels require 2 Kbytes, leaving 2 Kbytes free in the dual-ported RAM for buffer descriptors for other protocols.

To support 64 channels on the MPC860, 4-Kbyte dual-ported RAM is required for channel-specific parameters. Each logical channel occupies 64 bytes, requiring 4 Kbytes for 64 channels.

Non-QMC protocol implementations may be constrained by these memory requirements since their buffer descriptors need to use internal memory space.

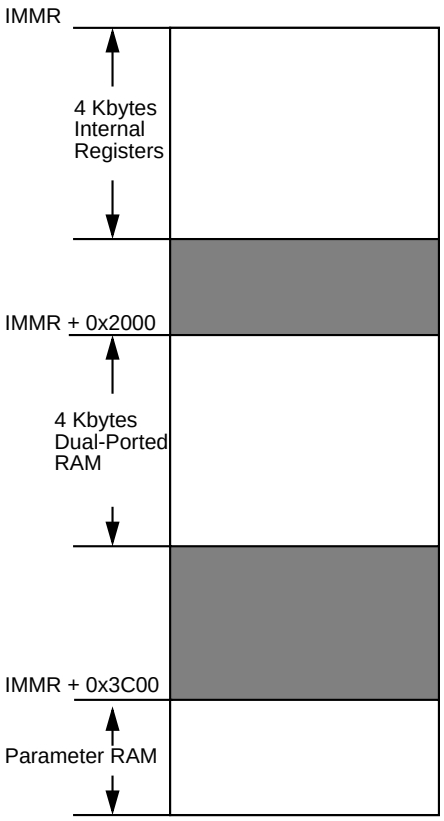


Figure 5-11. MPC860MH Internal Memory

- MFLR/MRBLR: MFLR (HDLC mode)—application-dependent.
  MRBLR (transparent mode)—must be divisible by 4 and large (>30) for better performance.

  ```
  ch[x].MFLR = 60;
  ```

- TRNSYNC: transparent synchronization, system-specific.

Step 17. Initialize RxBDs. Prepare an adequate number of receive buffers at the location addressed by RBASE. In the status word, set the E bit, set the I bit if interrupts are required and set the W bit for the last buffer descriptor. The data length is normally cleared, and the buffer pointer is set to a location in external memory. See Section 5.1, “Receive Buffer Descriptor,” for more information. Repeat for each enabled channel.

Step 18. Initialize TxBDs. Prepare an adequate number of transmit buffers at the location addressed by TBASE. In the status word, set the R bit, set the I bit if interrupts are required, and set the W bit for the last buffer descriptor. Other options are available and may be set or cleared depending on the application. The data length is written with the number of bytes to transmit, and the buffer pointer is set to a location in external memory. See Section 5.2, “Transmit Buffer Descriptor,” for more information. Repeat for each enabled channel.

Step 19. Initialize the circular interrupt table. If interrupts are required, initialize the interrupt table as explained in Chapter 4, “QMC Exceptions.” Clear the V and W bits, but make sure to set the last entry’s W bit.

Step 20. Initialize the channel mode register CHAMR (see Table 6-6). For more information see Section 2.4.1.1, “CHAMR—Channel Mode Register (HDLC),” for HDLC mode and Section 2.4.2.1, “CHAMR—Channel Mode Register (Transparent Mode),” for transparent mode.

Table 6-6. CHAMR Bit Settings

Name	Number of Bits	Description	Setting
MODE	1	0—transparent; 1—HDLC	X
RD/0	1	Transparent only: reverse data	X
1/IDLM	1	HDLC only: idle mode	X
ENT	1	Enable transmit	0
SYNC	1	Transparent only: synchronization	X
POL	1	Enable polling	0
CRC	1	HDLC only: CRC	X
NOF	4	Minimum number of flags	X

```
struct  descs {
      rxbdq recvbd0;               /* receive buffer 0 */
      txbdq xmitbd0;               /* transmit buffer 0 */
      } chbd[4];                   /* 4 sets of chan descriptors */
static char *poem[6];              /* poem area */
short linecntr;                    /* line counter */
struct intrpten {
      unsigned V:1;                /* entry valid bit */
      unsigned W:1;                /* entry wrap bit */
      unsigned NID:1;              /* not-an-idle has occurred */
      unsigned IDL:1;              /* an idle has occurred */
      unsigned :1;
      unsigned CHNMBR:5;           /* channel number */
      unsigned MRF:1;              /* maximum frame length violation */
      unsigned UN:1;               /* Tx underrun */
      unsigned RXF:1;              /* receive frame */
      unsigned BSY:1;              /* frame discarded, no buffers */
      unsigned TXB:1;              /* buffer transmitted */
      unsigned RXB:1;              /* receive buffer closed */
      }  intrpt[10];               /* interrupt table */
short recvcnt,xmitcnt,othrcnt = 0;  /* interrupt counters */

main()
{
      void SCC1esr();              /* exception service rtn */
      int *pvec;                   /* exception vector pntr */
      char vecblk = 3;             /* vector number block */
      char intlvl = 4;             /* interrupt level */
      pdpr = (struct dprbase *) (getmbar() & 0xFFFFE000);/*  init  dual-ported
      RAM ptr */
      pdpr->CICR.VBA2_VBA0 = (unsigned) (vecblk);/* vecs at vec num 0x60-7F */
      pdpr->CICR.IRL2_IRL0 = (unsigned) (intlvl);/* CPM interrupts level 4 */
      pdpr->CICR.HP4_HP0 = 0x1F;    /* no int priority change */
/* SCdP is zero from reset */
      pdpr->CICR.SCcP = 1;         /* SCC2 to SCCC position */
      pdpr->CICR.SCbP = 2;         /* SCC3 to SCCB position */
      pdpr->CICR.SCaP = 3;         /* SCC4 to SCCA position */
```

Chapter 7
Features Deleted in MC68MH360

An MC68MH360 operating in normal mode without the QMC protocol can perform the same functions as the MC68360 and MC68EN360 with two exceptions in protocol support. In order to create space in the CPM ROM for the QMC protocol, the support for Centronics and BiSync have been removed from the MH360. In all other respects, the MC68MH360 is compatible with its predecessors.

Table 8-2. CPM Performance Table (Continued)

Protocol	SCC Rate: Clock Frequency Mbps: MHz	Maximum Serial Throughput			
		25 MHz Mbps	33 MHz Mbps	40 MHz Mbps	50 MHz Mbps
Ethernet	1 : 1.136 HD	22	29	35	44
SMC transparent	1 : 16.67 FD	1.5	1.98	2.4	3
SMC UART	1 : 113.636 FD	0.220	0.290	0.352	0.440
QMC	1 : 11.90 FD	2.1	2.8	3.36	4.2
Bisync	1: 16.67 FD	1.5	1.98	2.4	3

NOTE

FD = Full duplex, HD = Half duplex

Further examples are given in Appendix A of the MPC860 user's manual.

Using Table 8-2, estimations of bandwidth utilization may be made. To calculate the total system load, add the CPM utilization from every channel together. Assuming approximately linear performance versus frequency[1], the general problem reduces to taking simple ratios:

$$\text{CPM Utiilization} = \left(\frac{\text{serial rate}_1}{\text{max serial rate}_1}\right) + \left(\frac{\text{serial rate}_2}{\text{max serial rate}_2}\right) \cdots$$

For example, since a 25-MHz Ethernet running at 22 Mbps consumes approximately 100% of the bandwidth, what bandwidth does a 10-Mbps channel require?

$$\text{CPM Utilization} = \frac{\text{serial rate}}{\text{max serial rate}} = \frac{10}{22} = 0.45$$

The above equation shows the 10-Mbps channel requiring 45% of the CPM bandwidth. More examples follow.

[1]Most protocols' performance is scalable linearly to frequency with the exception of Ethernet which has nonlinear behavior. However, for these calculations we assume linear scaling with frequency.

9.4 MSC Subchanneling Example

Figure 9-4 shows an example for eight 20-bit subchannels.

Time Slot 0	V	W	00	**0**	Channel #0A	000011
	V	W	00	**0**	Channel #0B	001100
	V	W	00	**0**	Channel #0C	110000
	V	W	11	**1**	Channel #0D	000000
Time Slot 31	V	W	00	**0**	Channel #31A	000011
	V	W	00	**0**	Channel #31B	001100
	V	W	00	**0**	Channel #31C	110000
	V	W	11	**1**	Channel #31D	000000

Figure 9-4. Example for Eight 2-Bit Subchannels

The example in Figure 9-4 uses two time slots to handle eight 2-bit subchannels. Time slot 0 is subdivided into four 2-bit subchannels. Note that time slot 0 is processed four times for the channels labeled 0A, 0B, 0C and 0D, each with different masks. It is only in the fourth entry that the L-bit (last bit) is set, instructing the CPM to process the next time slot. The same is true for time slot 31. It again is subdivided into four 2-bit subchannels. Note that the maximum number of channels is 32 due to the 5-bit channel pointer.

Therefore, a 50-MHz MPC860MH will be needed to run 64 channels of HDLC on one device.

B.2 860MH-Related Questions

Q: Is Ethernet only available on SCC1 for both 860EN and 860MH?

A: Ethernet is available on any channel. We recommend it on SCC1 due to its larger FIFO.

Q: How is 64-channel QMC implemented on the 50-MHz 860MH? What is the serial speed of the TDM channels?

A: Use two SCCs running 32-channel QMC protocol. Each channel is assumed to be 64-Kbps, like a normal time slot on a T1/E1 line, giving an aggregate rate of 4 Mbps (that is, twice the E1 rate).

Q: Does running transparent-mode processing on the QMC channels decrease the load on the CPM?

A: CPM loading in transparent mode is not significantly different from the loading in HDLC mode; therefore, performance will be the same.

Q: How many channelized T1/E1 ports does the 860MH support? (where E1 is thirty-two 64-Kbps channels and T1 is 24 channels)

A: With respect to running multiple channels of HDLC, the major limitation of the current 860MH is clock frequency. A 25-MHz part can run only 32 HDLC channels, while a 50-MHz part can run 64 channels. At this point, however, the size of the dual-ported RAM limits the number of HDLC channels to 64.

The MPC860 also has just two time slot assigners. Therefore, it can directly terminate at most two T1s or E1s.

Q: How is the 860MH configured to support more than 32 channels.

A: The QMC protocol for the 860MH can be used to support more than 32 HDLC channels in three ways:

- In one method, use shared transmit/receive channel routing on one SCC to run the QMC protocol linking the maximum of 64 time slots of a single multiplexed line to 64 separate logical channels.
- In another method, run the QMC protocol on two separate SCCs, each with its own set of parameters. With this method, two separate E1s can be routed to the two separate SCCs. It is not possible, however, to share channels from both E1s at random between the SCCs. (One E1 will map to the 32 logical channels of one SCC.)

C.3.1.2 Deactivation Procedure

When the clock-master S/T interface is deactivated, the QUICC32 receives an interrupt indicating the deactivation status (IRQ3 —register NR3 bit 3— meaning Info 0 of Figure C-9 has been received). Then, if another S/T interface is active, its TCLK signal must be selected to become the clock master; otherwise, the QUICC32 can select the BRG to be the clock master.

As shown in Figure C-9, the TCLK signal is disabled about 72.8 µs after the interruption. Therefore, the QUICC32 has 72.8 µs to react to the IRQ and to select another clock master.

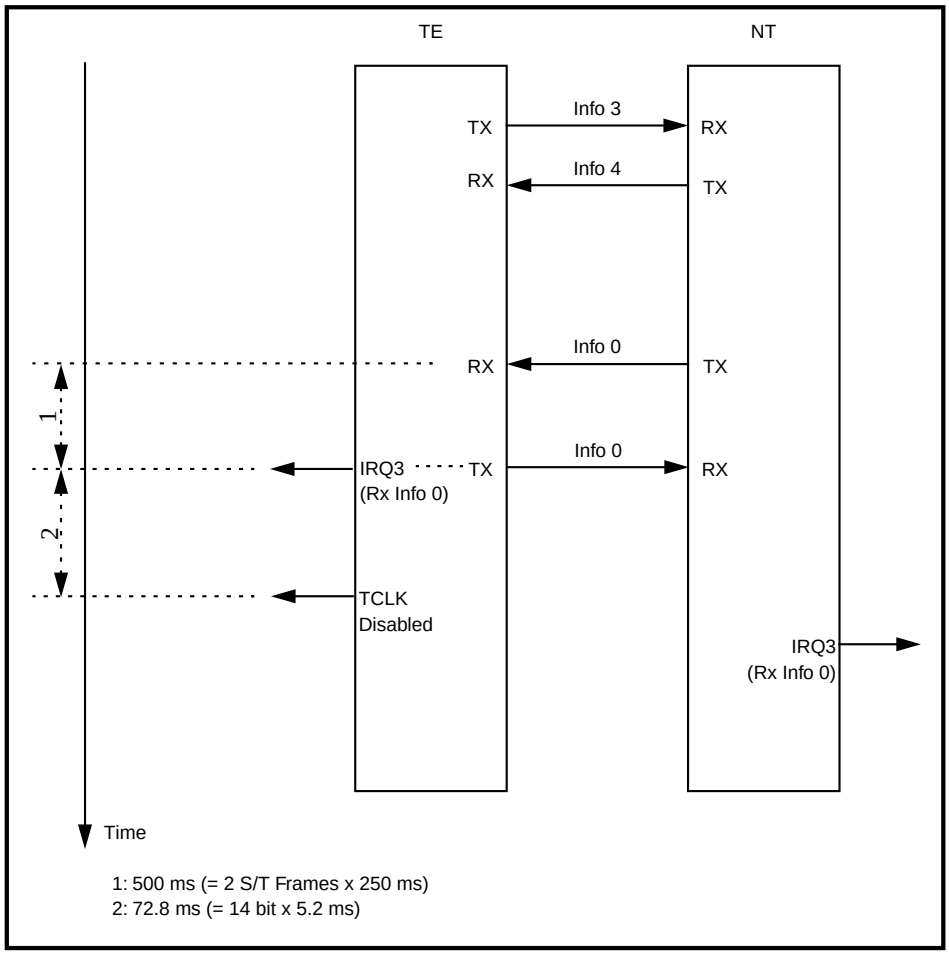


Figure C-9. Timing Diagram for a Deactivation (Always Initiated by the NT)