Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

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

Applications of Embedded - Microprocessors

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

Details	
Product Status	Obsolete
Core Processor	CPU32+
Number of Cores/Bus Width	1 Core, 32-Bit
Speed	25MHz
Co-Processors/DSP	Communications; CPM
RAM Controllers	DRAM
Graphics Acceleration	No
Display & Interface Controllers	-
Ethernet	10Mbps (1)
SATA	-
USB	-
Voltage - I/O	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/mc68mh360ai25l

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.

1.5.1 Synchronization

Independent receive and transmit clocks and frame synchronization signals control the data transfer. In NMSI operation, synchronization occurs only once to initiate a transfer using the CD (receive) and CTS (transmit) signals in pulse mode. If any noise corrupts either signal, the QMC will be out of synchronization until the whole protocol is restarted.

In contrast, the more robust SI performs a synchronization on each frame, limiting the damage from noise error on the clock or synchronization lines. Noisy channels can be restarted individually without interrupting other channels. For more details about possible errors in the TDM interface, see Section 1.8, “SI RAM Errors.”

1.5.2 Loopback Mode

The loopback from a transmitter to a receiver is implemented on a per channel basis for every logical channel. A common transmit and receive clock as well as a common frame synchronization pulse must be provided for loopback mode to work. The loopback is done on a fixed time slot, meaning that if one logical channel transmits on time slot 17, the loopback occurs through time slot 17 also, whether it is same logical channel or not that receives the incoming data. The reason for this restriction is that no buffering is performed after a channel is processed by the transmitter, or before it reaches the receiver.

Previously reserved, bit 15 of each entry in the SI-RAM is now the loopback bit controlling the loopback for the corresponding time slot. It is important to have each individual time slot as an entry in the SI-RAM for proper loopback on each individual channel.

1.5.3 Echo Mode

The SI can be programmed to echo incoming data. In this mode, the complete TDM link is retransmitted from the incoming L1RXDx to the L1TXDx pin on a bit-by-bit basis. The receiver section of the selected SCC can operate normally and also receive the incoming bit stream. This is also known as global echo mode on the whole link. Individual time slot echo is not possible with QMC without software intervention.

1.5.4 Inverted Signals

For each SCC, the DPLL can be used to invert the bitstream of the transmitter before the signal reaches the pin. This is not a bit-order inversion, but a logical level inversion. The DPLL can also invert the incoming data before it is forwarded to the receiver section. A logical inversion on a per channel basis is not possible in the QMC without external hardware. To invert a specific channel, the SI can be programmed to send a strobe signal at the channel’s corresponding time slot, assuming the SCC is operating in QMC mode. This strobe can then be connected to an external XOR gate to perform the inversion.

1.7 SCC Changes on the Fly

Changes can be made on the fly in the QMC routing tables, but changes made in the SI RAM require the link to be disconnected. If the connection is maintained during changes, synchronization and routing errors are likely to happen in the current frame. A workaround uses a shadow RAM routing table. The shadow table can hold alternative routing information to be switched in at the appropriate time slot boundary. The drawback to this method is that the number of entries in the SI RAM is reduced by half. But since the routing tables in the QMC protocol are being changed anyway, the recommended solution is to have all relevant time slots routed to the SCC.

The SI RAM also gives the user the capability to multiplex other channels to and from a TDM if not all time slots are used by the QMC. A third option is to have several external devices multiplexed. Use the open collector mode if several QUICCs or PowerQUICCs are connected together for subchanneling applications.

1.8 SI RAM Errors

The following three types of errors are identified:

- Data bit error
- Clock pulse error
- Synchronization pulse error

Errors in frame-based protocols are easy to detect by the protocol controller. An error in an HDLC channel is detected at the end of a frame when a buffer is closed and all status bits are reported in the buffer descriptor (BD). The error type for bit errors is normally CRC errors. For errors occurring in the SI (noise on clock or synchronization pulses), the error may also be of type frame-length-violation or non-octet-aligned. See Chapter 5, “Buffer Descriptors,” for more information. This section covers the type of errors reported through the buffer descriptors. For transparent channels, the error detection mechanism is left to the user in higher-level software. Most transparent channels, such as voice carriers, are tolerant of errors. Frame-based channels, on the other hand, require error detection since they often rely on critical control messages.

The number of clocks that occur between sync pulses is given in the SI RAM programming. The clock-counting state machine expects a new sync pulse after the end of each frame. The following paragraphs discuss the different error cases and describe the counter state and the frame delay before synchronization is resumed.

A clock pulse error occurs if other than exactly one clock pulse is detected by the SI RAM in a given frame. In this error case, since the SI RAM bases its routing on counting clock pulses, the now corrupted signal routing affects all channels. The SI RAM expects another sync pulse when it reaches the last entry of the frame.

Table 2-7. TSTATE Field Descriptions for 860MH (HDLC)

Field	Name	Description
0–1	—	0
2	—	1
3	**MOT**	Motorola/Intel bit 0 = The bus format is Intel format (little-endian). 1 = The system bus is considered to be organized in Motorola format (big-endian).
4	—	0
5–7	**AT[1–3]**	Address type—This field contains the address type for the transmitter DMA channel for data buffers in external memory (transmit buffers). Address types are needed by the memory controller to decode a correct memory cycle and activate the correct handshaking.

2.4.1.3 INTMSK—Interrupt Mask (HDLC)

Each event defined in the interrupt circular queue entry maps directly to a bit in INTMSK as shown in Figure 2-9. There is one mask bit for each event—NID (bit 2), IDL (bit 3), MRF (bit 10), UN (bit 11), RXF (bit 12), BSY (bit 13), TXB (bit 14) and RXB (bit 15). Bits that do not map to an event are reserved. Reserved bits must be set to zero. Refer to Chapter 4, “QMC Exceptions,” for more detail.

- 0 = No interrupt request is generated and no new entry is written in the circular interrupt table.
- 1 = Interrupts are enabled.

This register is initialized by the host prior to operation.

Interrupt Table Entry:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
V	W	NID	IDL	—	CHANNEL NUMBER					MRF	UN	RXF	BSY	TXB	RXB
Reset: 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

INTMSK:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RESERVED		INTERRUPT MASK		RESERVED						INTERRUPT MASK BITS					
RESET: 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

Figure 2-9. INTMSK and Interrupt Table Entry (HDLC)

2.4.2.2 TSTATE—Tx Internal State (Transparent Mode)

TSTATE defines the internal transmitter state. The high byte of TSTATE defines the function code/address type and the Motorola/Intel bit (bit 3) that should always be set to 1. Figure 2-12 shows the TSTATE register for transparent mode.

0	1	2	3	4	5	6	7
0	0	1	MOT	FC[3–0]/ AT[1–3]			

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

Figure 2-12. TSTATE—Tx Internal State (Transparent Mode)

For the MH360, TSTATE should be host-initialized to 0x3800_0000 before enabling the channel—function code 8. Table 2-12 describes the TSTATE fields for the MH360 with boldfaced parameters to be initialized by the user.

Table 2-12. TSTATE Field Descriptions for MH360 (Transparent Mode)

Field	Name	Description
0–1	—	0
2	—	1
3	**MOT**	Motorola/Intel bit 0 = The bus format is Intel format (little-endian). 1 = The system bus is considered to be organized in Motorola format (big-endian).
4–7	**FC[3–0]**	Function code—This field contains the function code for the transmitter DMA channel for data buffers in external memory (transmit buffers). Function codes are needed by the memory controller to decode a correct memory cycle and activate the correct handshaking.

For the 860MH, TSTATE should be host-initialized to 0x3000_0000 before enabling the channel—AT = 0. Note that for the 860MH bit 4 should always be zero as only bits 5–7 map to AT[1–3]. Table 2-13 describes the TSTATE fields for the 860MH with boldfaced parameters to be initialized by the user.

Table 2-13. TSTATE Field Descriptions for 860MH (Transparent Mode)

Field	Name	Description
0–1	—	0
2	—	1
3	**MOT**	Motorola/Intel bit 0 = The bus format is Intel format (little-endian). 1 = The system bus is considered to be organized in Motorola format (big-endian).
4	—	0
5–7	**AT[1–3]**	Address type—This field contains the address type for the transmitter DMA channel for data buffers in external memory (transmit buffers). Address types are needed by the memory controller to decode a correct memory cycle and activate the correct handshaking.

Chapter 3
QMC Commands

The host issues commands to the QMC by writing to the command register (CR). The QMC commands are similar to those of standard QUICC HDLC protocol. The CR format for QMC is shown in Figure 3-1.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RST	QMC OPCODE			1	1	1	0	0	CHANNEL NUMBER					–	FLG

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

Figure 3-1. Command Register (CR)

3.1 Transmit Commands

STOP TRANSMIT <channel> (QMC opcode = 001)

The stop transmit command disables the transmission of data on the selected channel and clears the POL bit in the CHAMR register. Upon asserting this command in the middle of a frame, the RISC processor sends an ABORT indication (7F) followed by IDLEs or FLAGs, depending on the mode, on the selected channel. If this command is issued between frames, the RISC processor continues sending IDLEs or FLAGs (depending on the IDLM mode bit in the CHAMR register) in this channel.

The Tx buffer descriptor pointer (TBPTR) is not advanced to the next buffer; see Table 2-4 and Section 2.2, “Global Multichannel Parameters.”

Set (1) the POL bit to start transmission or to continue after a stop command.

Only after transmission start for a deactivated channel, which is identified by a cleared (0) V bit in the time slot assignment table or a cleared (0) ENT bit, is it necessary to initialize ZISTATE and TSTATE before setting (1) the POL bit.

To deactivate a channel, clear the V bit in the TSA table and the ENT bit in the channel mode register (CHAMR).

Chapter 4
QMC Exceptions

QMC interrupt handling involves two principle data structures—the SCC event register (SCCE) and the circular interrupt table. Figure 4-1 illustrates the circular interrupt table.

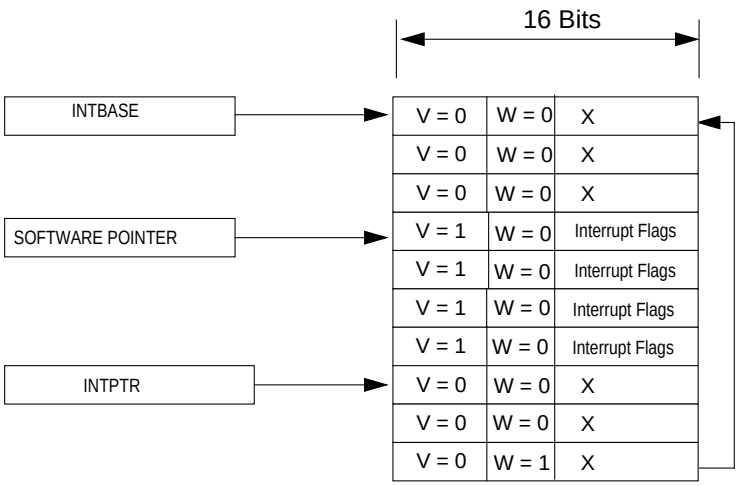
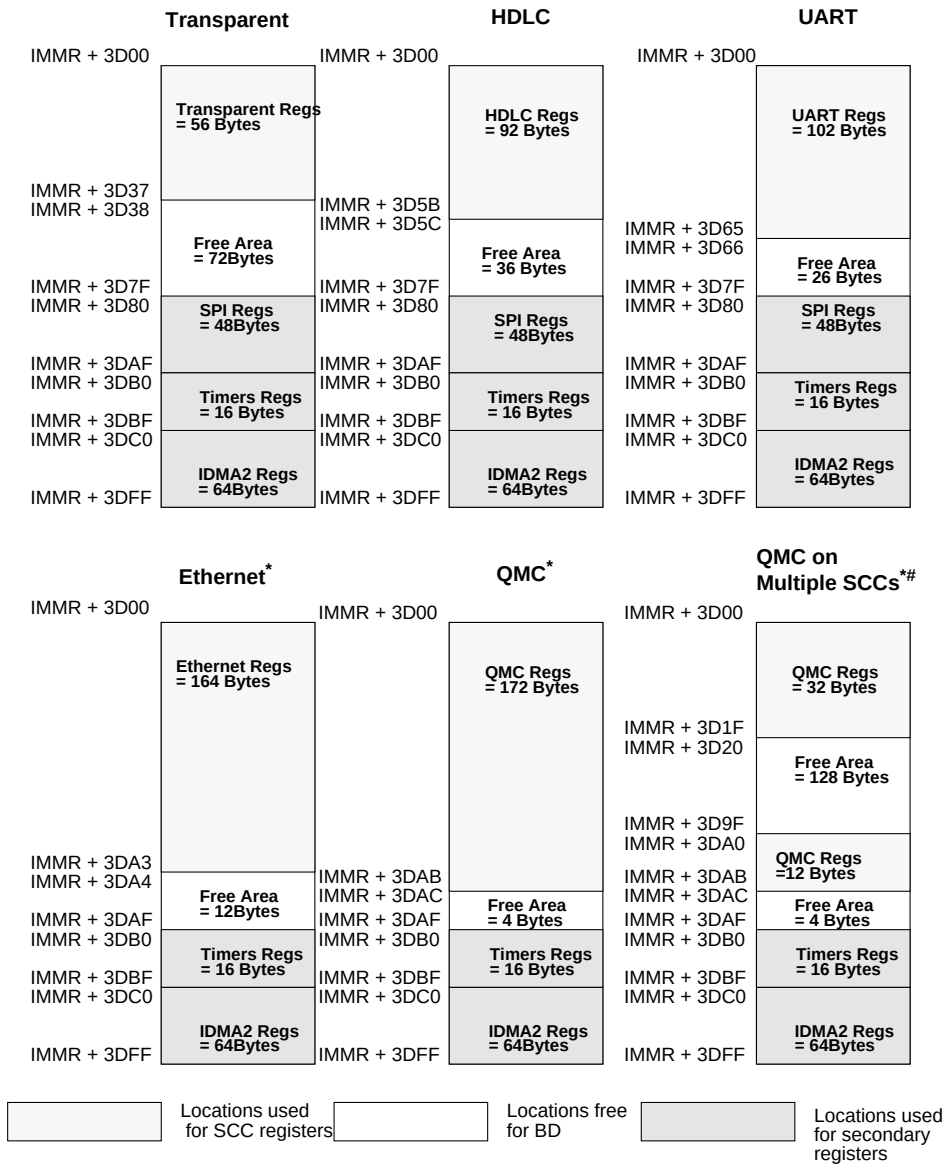


Figure 4-1. Circular Interrupt Table in External Memory

INTBASE (interrupt base) points to the starting location of the queue in external memory, and INTPTR (interrupt pointer) marks the current empty position available to the RISC processor. Both pointers are host-initialized global QMC parameters; see Table 2-1. The entry whose W (wrap) bit is set to 1 marks the end of the queue. When one of the QMC channels generates an interrupt request, the RISC processor writes a new entry to the queue. In addition to the channel's number, this entry contains a description of the exception. The V (valid) bit is then set and INTPTR is incremented. When INTPTR reaches the entry with W = 1, INTPTR is reset to INTBASE.

An interrupt is written to the interrupt table only if it survives a mask with the INTMASK (interrupt mask) register. Following a write to the queue, the QMC protocol updates the SCC event register (SCCE) according to the type of exception.



#—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

5.3.4 MC68MH360 Configured for QMC and Ethernet

Certain difficulties may arise when QMC and Ethernet are used together.

In a 25-MHz system, Ethernet can work together with 16 QMC channels. In this case, a careful use of logical channels can free a 1-Kbyte space in the parameter area for up to 128 buffer descriptors. For more information, see Section 5.3.2, “Parameter RAM Usage for QMC over Several SCCs.”

For 32-channel QMC operation coupled with Ethernet, a 33-MHz part is required. In addition, Ethernet must operate on SCC1, and the QMC protocol must be divided over SCC3 and SCC4 to use the combined FIFO size. This leaves SCC4’s parameter RAM page unused as the largest single open space for the SCC1 Ethernet, resulting in 24 buffer descriptors for transmission and reception in a single block. See Chapter 2, “QMC Memory Organization,” for more information.

Chapter 6
QMC Initialization

This section describes the essential steps to initialize QMC after a hard reset. Section 6.1, “Initialization Steps,” discusses the steps required to initialize the QMC protocol, and Section 6.2, “68MH360 T1 Example,” provides example code.

6.1 Initialization Steps

This section describes the steps required to initialize the QMC protocol.

Step 1: Initialize the SIMODE (serial interface mode) register. The SIMODE register is defined on page 7-78 of the MC68360 User’s Manual, and page 16-114 of the MPC860 user’s manual. Table 6-1 shows the transmit buffer descriptor field descriptions.

Table 6-1. Transmit Buffer Descriptor Field Descriptions

Name	No. of Bits	Description	Setting
SMCx	1	Connect to TDM or NMSI	X
SMCxCS	3	Specify clock source	X
SDMx	2	Normal, echo, or loopback mode	00
RFSDx	2	Receive frame sync delay	System-specific
DSCx	1	Double-speed clock (GCI)	System-specific
CRTx	1	Common transmit and receive sync & clk	System-specific
STZx	1	Set L1TXDx to until serial clks	0
CEx	1	Clock edge for xmit	System-specific
FEx	1	Frame sync edge	System-specific
GMx	1	Grant mode support	0
TFSDx	2	Transmit frame sync delay	System-specific

Clear the ENR and ENT bits at the end of the initialization. The MODE setting for QMC mode is 0b1010.

A typical setting would be:

```
GSMR_L = 0x0000_000A;          /* enable QMC */
```

Step 11. Initialize basic global multichannel parameters as follows. See Chapter 2, “QMC Memory Organization,” for more information.

- MCBASE: (multichannel base pointer) is a pointer to a 64-Kbyte buffer descriptor table in external memory. For example:

  ```
  SCC1.MCBASE = 0x1_0000;        /* BD base located 0x1_0000 */
  ```

- INTBASE: (interrupt table base pointer) - points to the interrupt table in external memory. For example:

  ```
  SCC1.INTBASE = 0xF000;         /* interrupt table base 0xF000 */
  ```

- MRBLR: (maximum receive buffer length) - should be large (> 30) for better performance and should be a multiple of 4 bytes. This is valid for HDLC only. For example:

  ```
  SCC1.MRBLR = 60;               /* set receive buffer length to 60 */
  ```

- GRFTHR: (global receive frame threshold) - normally set to 1. For example:

  ```
  SCC1.GRFTHR = 1;               /* 1 receive frame to interrupt */
  ```

- GRFCNT: (global receive frame count) - should be initialized to the same value as GRFTHR. For example:

  ```
  SCC1.GRFCNT = 1;               /* 1 receive frame to interrupt */
  C_MASK32:CRC constant, 32-bit =0xDEBB20E3
  SCC1.C_MASK32 = 0xDEBB20E3;    /* init 32-bit CRC const */
  C_MASK16:CRC constant,16-bit=0xFOB8
  SCC1.C_MASK16 = 0xF0B8;        /* init 16-bit CRC const */
  ```

Step 12. Copy INTBASE to INTPTR (multichannel interrupt pointer). See Chapter 4, “QMC Exceptions,” for more information.

```
SCC1.INTPTR = SCC1.INTBASE;          /* init intptr */
```

Chapter 8
Performance

Calculating performance is key to choosing the clock frequency required for a given system. For the 860MH and MH360, the large number of possible channel combinations complicates performance estimation.

This chapter addresses the problem by first providing a performance table for common configurations supported by the 860MH and/or the MH360 in Section 8.1, “Common Channel Combinations.” For configurations not covered in the first section, Section 8.2, “CPM Loading,” covers general guidelines and examples for determining the serial bit rate and CPM loading on a given system. The final section, Section 8.3, “Bus Latency and Peak Load,” addresses system bus utilization and arbitration.

For more definitive answers to performance questions, benchmark the desired configuration on a development board.

8.1 Common Channel Combinations

Table 8-1 provides some common channel combinations configured on the MH devices along with suggested operating frequencies. Note that the MH360 is available only at 25 and 33 MHz,; the 860MH is currently available at 25, 40, and 50 MHz.

Table 8-1. Common QMC Configurations

Protocols Selected	Frequency Supported			
	25 MHz	33 MHz	40 MHz	50 MHz
SCC1: 24-channel QMC. Serial bit rate 1.544 Mbps (T1)	Yes	Yes	Yes	Yes
SCC1: 32-channel QMC. Serial bit rate 2.048 Mbps (E1/CEPT)	Yes	Yes	Yes	Yes
SCC1: 10-Mbps Ethernet; SCC2: 12 x 64-Kbps QMC; SCC3: 12 x 64-Kbps QMC. TDM bit rate = 1.544 Mbps	Yes	Yes	Yes	Yes
SCC1: 10-Mbps Ethernet; SCC2: 16 x 64-Kbps QMC; TDM bit rate = 1.544 Mbps	Yes	Yes	Yes	Yes
SCC1: 32 x 64-Kbps QMC; SCC2: 64 Kbps; SCC3:64 Kbps; SCC4: 64 Kbps; all HDLC.	Yes	Yes	Yes	Yes

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.

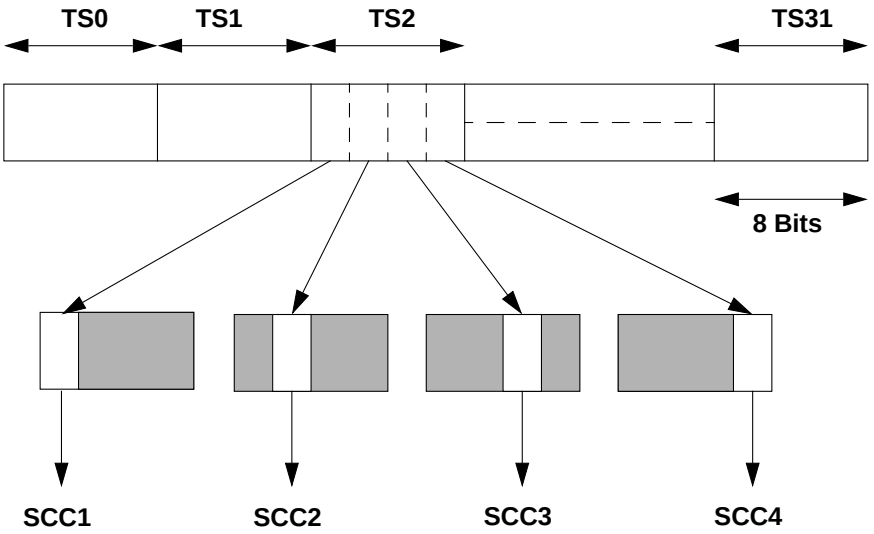


Figure 9-1. Two-Bit Subchannel Implementation without MSC Microcode

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.

9.6 Multi-Subchannel Initialization

The initialization of the MSC microcode is the same as the standard QMC initialization with the following additions:

- Prepare TSATTx and TSATRx to include the L (last) bit
- The RISC controller trap registers for the MPC860 rev A must be set as follows:
  - Address SCC base + 9CC, RCTR1 = 0x8074
  - Address SCC base + 9CE, RCTR2 = 0x8054
  - Address SCC base + 9D0, RCTR3 = 0x92F2
  - Address SCC base + 9D2, RCTR4 = 0x9097
- Load microcode S records into dual-ported RAM
- Set the RCCR (address SCC base + 9C4) = 0x0002
- Set the GSMR_H = 0x07A0 (RFW = 1)

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.3.1.1 Activation Procedure

If no S/T transceiver is active, no TCLK clock is generated. Once the first transceiver is activated, it will generate a TCLK signal only if DCL and FSC signals are present as well.

Pseudo DCL and FSC signals generated from one of the baud rate generators (BRG) of the QUICC32 can be used to generate the TCLK signal. The BRG can generate a clock based on the QUICC32's system clock. A divider factor should be chosen so that the BRG frequency is close to 2.048 MHz. This clock then feeds into the 256-divider circuitry of Figure C-4 to generate a pseudo DCL and a pseudo FSC.

A multiplexer commanded by the QUICC32 is required to select either the BRG signal or the TCLK signals of the transceivers to be the clock master generating the DCL and FSC signals.

When no transceiver is activated, the QUICC32 selects the BRG to be the clock master, and the S/T interface receives the pseudo DCL and FSC signals. (These two signals are not synchronized to the network but are not used to sample data.)

As the first MC145574 is activated, it will be able to generate the TCLK signal; see Figure C-7. This transceiver will then send an interrupt to the QUICC32 (IRQ3—register NR3[3]—meaning Info 2 has been received) indicating that the activation process has begun. The QUICC32 then uses the multiplexer to select the TCLK signal of that MC145574 to be the clock master.

As shown in Figure C-7 and Figure C-8, the TCLK signal is present before the interruption, with at least 750 µs between the IRQ and the received Info 4. The QUICC32 therefore has 750 µs to react to the IRQ and to select the new clock master.

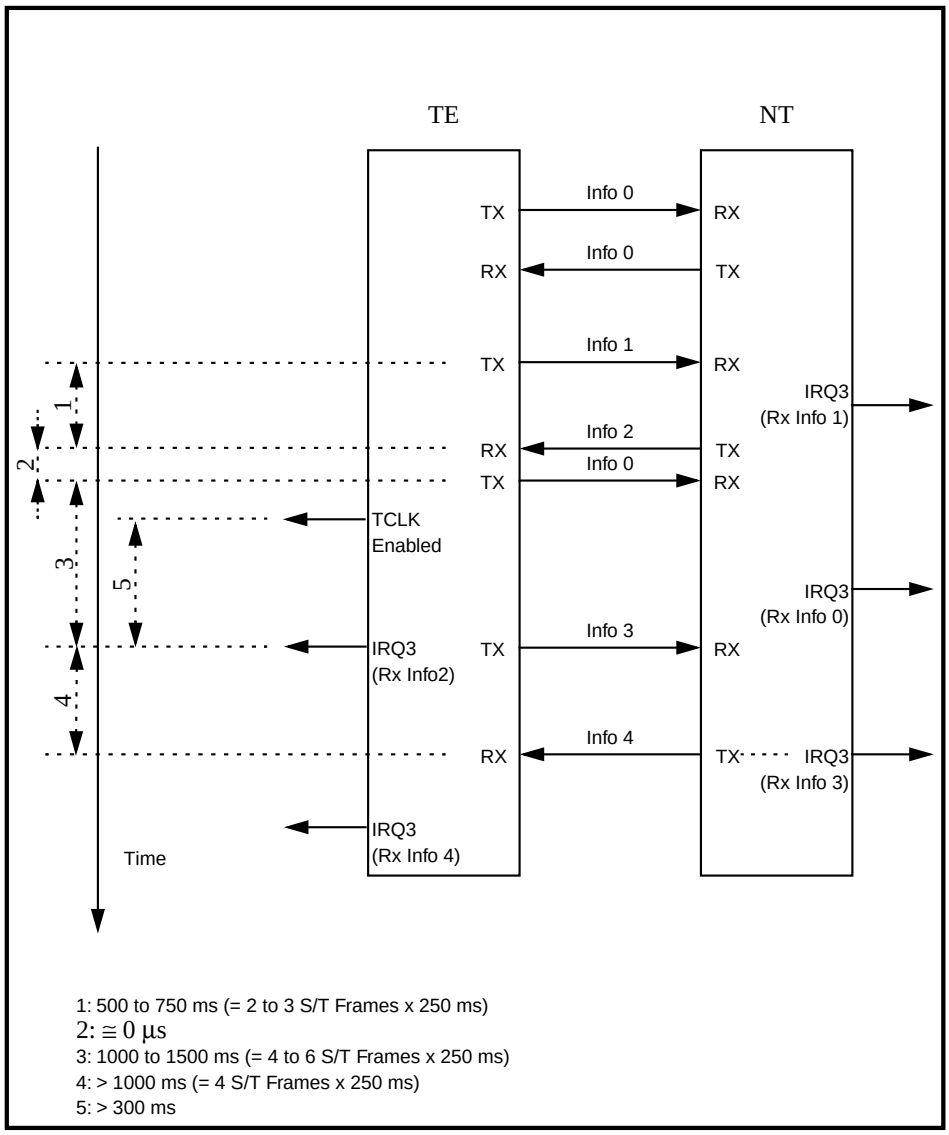


Figure C-8. Timing Diagram for an Activation Initiated by the TE

C.3.2 MC145572 U Interface

The FREQREF signal of the MC145572 provides a clock synchronized to the network timing for the U interface. This frequency reference is a fixed 2.048-MHz clock enabled by setting OR8[4] in the MC145572 register set.

The U interface's FREQREF differs from the TCLK of the S/T interface. When enabled, the FREQREF signal generates the 2.048-MHz clock regardless of the activation status of the U interface (but that clock is synchronized to the network only when the U interface is activated). Also, FREQREF does not require the DCL and FSC signals to enable clock generation.

Like the S/T interface, on the other hand, the FREQREF signal can be used as the DCL for the IDL bus. When divided by 256, FREQREF can also be used to generate the 8-KHz frame sync FSC. The same logic design used for the S/T interface must be added to insure a correct FSC duty cycle; see Figure C-4.

Also like the S/T interface, elastic buffers are included to allow the U interface to operate with any phase relationship between the IDL frame sync and the network.

Figure C-10 shows a block diagram of the connection between four U interfaces and the QUICC32. The diagram would be the same for up to 10 U interfaces.