Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

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

Applications of Embedded - Microprocessors

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

Details	
Product Status	Obsolete
Core Processor	CPU32+
Number of Cores/Bus Width	1 Core, 32-Bit
Speed	33MHz
Co-Processors/DSP	Communications; CPM
RAM Controllers	DRAM
Graphics Acceleration	No
Display & Interface Controllers	-
Ethernet	10Mbps (1)
SATA	-
USB	-
Voltage - I/O	5.0V
Operating Temperature	0°C ~ 70°C (TA)
Security Features	-
Package / Case	240-BFQFP
Supplier Device Package	240-FQFP (32x32)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68mh360ai33l

TABLES

Table Number	Title	Page Number
i	Acronyms and Abbreviated Terms	xiii
2-1	Global Multichannel Parameters	2-5
2-2	Time Slot Assignment Table Entry Fields for Receive Section	2-9
2-3	Time Slot Assignment Table Entry Fields for Transmit Section	2-9
2-4	Channel-Specific HDLC Parameters	2-14
2-5	CHAMR Field Descriptions (HDLC)	2-16
2-6	TSTATE Field Descriptions for MH360 (HDLC)	2-17
2-7	TSTATE Field Descriptions for 860MH (HDLC)	2-18
2-8	RSTATE Field Descriptions for MH360 (HDLC)	2-19
2-9	RSTATE Field Descriptions for 860MH (HDLC)	2-20
2-10	Channel-Specific Transparent Parameters	2-20
2-11	CHAMR Bit Settings (Transparent Mode)	2-22
2-12	TSTATE Field Descriptions for MH360 (Transparent Mode)	2-23
2-13	TSTATE Field Descriptions for 860MH (Transparent Mode)	2-23
2-14	RSTATE Field Descriptions for MH360 (Transparent Mode)	2-28
2-15	RSTATE Field Descriptions for 860MH (Transparent Mode)	2-29
4-1	SCC Event Register Field Descriptions	4-4
4-2	Interrupt Table Entry Field Descriptions	4-5
5-1	Receive Buffer Descriptor (RxBD) Field Descriptions	5-1
5-2	Transmit Buffer Descriptor (TxBD) Field Descriptions	5-5
5-3	MC68360 Functions Available	5-9
5-4	MPC860MH Functions Available	5-15
6-1	Transmit Buffer Descriptor Field Descriptions	6-1
6-2	SICR Bit Settings	6-2
6-3	SIGMR Bit Settings	6-4
6-4	GSMR_H Bit Settings	6-4
6-5	GSMR_L Bit Settings	6-5
6-6	CHAMR Bit Settings	6-9
6-7	Pointer Registers	6-18
6-8	State Registers	6-18
8-1	Common QMC Configurations	8-1
8-2	CPM Performance Table	8-2
8-3	QMC Actions in Tx Buffer Switch	8-5
8-4	Simulated Latencies	8-6

- Chapter 9, “Multi-Subchannel (MSC) Microcode,” provides the MSC microcode features and operation, and discusses how to program the MSC protocol.
- Appendix A, “68360 Bit Numbering,”shows the bit numbering used for the 68360.
- Appendix B, “Frequently-Asked Questions,” provides a list of common questions and solutions for the MH360 and 860MH.
- Appendix C, “Connecting S/T or U Interfaces to QUICC32,” shows how multiple MC145574 (S/T interface) or MC145572 (U interface) can be connected to a QUICC32. It describes the level-1 connections and explains the data flow through the devices.
- This manual also includes an index.

Additional Reading

This section provides a brief list of additional reading that supplements the information in this manual.

The following materials are available from the Motorola Literature Distribution Centers listed on the back cover of this manual; the document order numbers are included in parentheses for ease in ordering:

- *MPC860 PowerQUICC User’s Manual* (MPC860UM/AD)
- *MC68360 Quad Integrated Communications Controller User’s Manual*, Rev. 1 *(M68360UM/AD)*
- *M68000 Family Programmer’s Reference Manual*, Rev. 1 (M68000PM/AD)

Conventions

This document uses the following notational conventions:

ACTIVE_HIGH	Names for signals that are active high are shown in uppercase text without an overbar. Active-high signals are referred to as asserted when they are high and negated when they are low.
$\overline{\text{ACTIVE\_LOW}}$	A bar over a signal name indicates that the signal is active low. Active-low signals are referred to as asserted (active) when they are low and negated when they are high.
0x0F	Hexadecimal numbers
0b0011	Binary numbers
REG[FIELD]	Abbreviations or acronyms for registers are shown in uppercase text. Specific bit fields or ranges are shown in brackets.

Bold font (field name)Entries in boldface must be initialized by the user.

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.

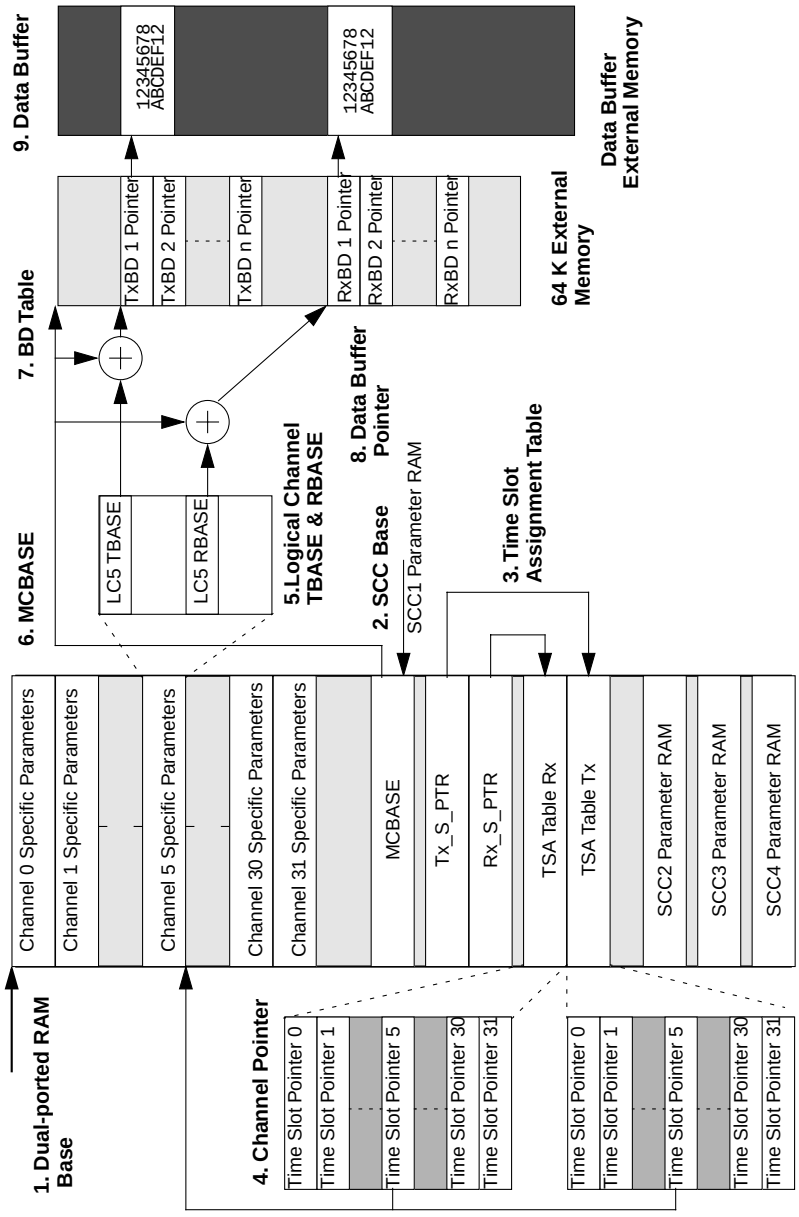

Figure 2-2. QMC Memory Structure

Table 2-5. CHAMR Field Descriptions (HDLC) (Continued)

Field	Name	Description
10–11	—	Reserved
12–15	**NOF**	Number of flags—Defines the minimum number of flags before frames. However, even if NOF = 0, at least one flag is transmitted before the first frame. See the description of the IDLM bit for more information.

2.4.1.2 TSTATE—Tx Internal State (HDLC)

TSTATE defines the internal transmitter state. The high byte of TSTATE defines the function code/address type and the Motorola/Intel bit. Bit 3 (or bit 28 for the 68360) should always be set to 1. Figure 2-8 shows the TSTATE register for HDLC operation.

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 2-8. TSTATE—Tx Internal State (HDLC)

For the MH360, TSTATE should be host-initialized to 0x3800_0000 before enabling the channel—function code 8. Table 2-6 describes the TSTATE fields for the MH360 with boldfaced parameters to be initialized by the user.

Table 2-6. TSTATE Field Descriptions for MH360 (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).
4–7	**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 2-7 describes the TSTATE fields for the 860MH with boldfaced parameters to be initialized by the user.

2.4.1.4 RSTATE—Rx Internal State (HDLC)

Host-initialized to 0x3900_0000 before enabling the channel or after a fatal error (that is, global overrun, busy) or after a STOP Rx command.

The high byte of RSTATE defines the function code/address type and the Motorola/Intel bit. Bit 3 (or bit 28 for the 68360) should always be set to 1. Figure 2-10 shows the RSTATE register for HDLC operation.

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 2-10. RSTATE—Rx Internal State (HDLC)

For the MH360, RSTATE should be host-initialized to 0x3900_0000 before enabling the channel—function code 9. Table 2-8 describes the RSTATE fields for the MH360 with boldfaced parameters to be initialized by the user.

Table 2-8. RSTATE Field Descriptions for MH360 (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).
4–7	**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, RSTATE should be host-initialized to 0x3100_0000 before enabling the channel—AT = 1. Note that for the 860MH, bit 4 should always be zero as only bits 5–7 map to AT[1–3]. Table 2-9 describes the RSTATE fields for the 860MH with boldfaced parameters to be initialized by the user.

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.

Table 4-2. Interrupt Table Entry Field Descriptions (Continued)

Field	Name	Description
3	IDL	Idle 0 = No IDL event has occurred. 1 = The channel's receiver has identified the first occurrence of HDLC idle (FFFE) after any non-idle pattern. IDL interrupts are not generated in transparent mode.
4		Reserved
5–9	Channel Number	This 5-bit field identifies the requesting logical channel index (0–31).
10	MRF	Maximum receive frame length violation—This interrupt occurs in HDLC mode when more than MFLR number bytes are received. As soon as MFLR is exceeded, this interrupt is generated and the remainder of the frame is discarded. At this point the receive buffer is not closed and the reception process continues. The receive buffer is closed upon detecting a flag. The length field written to this buffer descriptor is the entire number of bytes received between the two flags. MRF interrupts are not generated in transparent mode. **Note**: The MRF interrupt is generated directly when the MFLR value is a multiple of 4 bytes. The checking of this is done on a long-word boundary whenever the SDMA transfers 32 bits to memory. If MFLR is not aligned to 4x bytes, this interrupt may be 1- to 3-byte timings late for this channel. In any case, the violation can be checked to any number of bytes. The last entry in the data buffer is always a full long word.
11	UN	Tx no data 0 = No UN event has occurred. 1 = There is no valid data to send to the transmitter. The transmitter sends an abort indication consisting of 16 consecutive 1's and then sends idles or flags according to the protocol and the channel mode register setting. This error occurs when a transmit channel has no data buffer ready for a multibuffer transmission already in progress. Transmission of a frame is a continuous bitstream without gaps or interruption. When a buffer is not ready in the middle of this sequence, it is an error situation. This can be viewed as channel underrun. The transmitter for this channel is stopped. See Section 6.3, "Restarting the Transmitter," for recovery information.
12	RXF	Rx frame 0 = No RXF event has occurred. 1 = A complete HDLC frame is received. Data is stored in external memory and the buffer descriptor is updated. If during frame reception an abort sequence of at least seven 1's is detected, the buffer is closed and both RXB and RXF are reported along with the AB in the buffer descriptor. As a result of end-of-frame, the global frame counter GRFCNT is decremented for interrupt generation. This counter is decremented on RXF only, regardless if the RXF was caused by correct closing or due to an error. RXF interrupts are not generated in transparent mode.
13	BSY	Busy 0 = No BSY event has occurred. 1 = A frame was received but was discarded due to lack of buffers. This can be viewed as channel overrun. After a busy condition, the receiver for this channel is disabled and no more data is transferred to memory. Receiver restart is described in Section 6.4, "Restarting the Receiver."

Table 6-4. GSMR_H Bit Settings (Continued)

Name	No. of Bits	Description	Setting
CDS	1	CD sampling	1
CTSS	1	$\overline{\text{CTS}}$ sampling	1
TFL	1	Transmit FIFO length	0
RFW	1	Receive FIFO width	0
TXSY	1	Transmitter synchronized to the receiver	0
SYNL	2	Sync length	00b
RTSM	1	RTS mode	0
RSYN	1	Receive synchronization timing	0

A typical setting would be:

```
GSMR_H = 0x0000_0780;          /* enable pulse mode and sampling */
```

Step 10. Initialize general SCCx mode reg low, GSMR_L (see Table 6-5). For more information on GSMR programming, see page 7-111 of the MC68360 User's Manual and page 16-153 of the MPC860 User's Manual.

Table 6-5. GSMR_L Bit Settings

Name	No. of Bits	Description	Setting
SIR	1	Infrared encoding, only on 860MH	X
EDGE	2	Clock edge	00
TCI	1	Transmit clock invert	0
TSNC	2	Transmit sense	00b
RINV	1	DPLL receive input invert data	0
TINV	1	DPLL transmit input invert data	0
TPL	3	Tx preamble length	0b000
TPP	2	Tx preamble pattern	0b00
Tend	1	Transmitter frame ending	0
TDCR	2	Transmit divide clock rate	00
RDCR	2	Receive divide clock rate	00
RENC	2	Receive decoding	00
TENC	2	Transmitter decoding	00
DIAG	2	Diagnostic mode	system-specific
ENR	1	Enable receive	0
ENT	1	Enable transmit	0
MODE	4	Channel protocol mode	0b1010

- 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

Note the ENT bit is initially cleared, but then must be set when the channel is ready to start transmitting. Similarly, the POL bit is initially cleared, but then must be set each time a buffer descriptor is enabled to transmit. Example settings are as follows:

```
ch[x].CHAMR.MODE = 1;          /* select HDLC */
ch[x].CHAMR.IDLM = 0;          /* no idles between frames */
ch[x].CHAMR.ENT = 1;           /* enable channel xmit */
ch[x].CHAMR.CRC = 1;           /* select 32-bit CRC */
ch[x].CHAMR.NOF = 7;           /* 7 flags between frames */
ch[x].CHAMR.POL = 1;           /* enable polling by RISC */
```

Step 21. Initialize the SCCE register. From reset, SCCEx will be zero requiring no initialization. However, if required, it can be cleared by writing a 1 in each of the status bits. See Section 4.1, “Global Error Events,” for more information.

```
SCCE1 = 0xF;                   /* clear all interrupts */
```

Step 22. Initialize the mask register, SCCMx. Any interrupts which are not used should be masked in the SCCM register. SCC interrupts should be enabled using the CIMR register, if required. The CIMR register is defined on page 7-381 of the MC68360 User’s Manual and page 16-483 of the MPC860 User’s Manual.

```
SCCM1 = 0xF;                   /* enable all interrupts */
CIMR.SCC1 = 1;                 /* SCC1 interrupts enabled */
```

Step 23. Enable the transmitter (ENT bit) and the receiver (ENR bit) in the general SCC mode register (GSMR).

```
GSMR_L1.ENR = 1;               /* enable receiver */
GSMR_L1.ENT = 1;               /* enable transmit */
```

6.2 68MH360 T1 Example

```
/* This is an example of transmitting and receiving on four  */
/* HDLC channels in loopback mode. */
/* Equipment : SBC360 Evaluation Board with QUICC32 */
/* (T1MH.C) */

void *const stdout = 0;            /* standard output device */
#include <string.h>               /* string functions */
#include <stdio.h>                /* I/O functions */
#define qmc1                      /* SCC1 is multichannel comm */
#include "68360.h"                /* dual-ported RAM equates */
struct dprbase *pdpr;             /* pointer to dual-ported RAM */
```

```
      poem[3] = "Couldn't put Humpty together again\n\r";
      poem[4] = "";
      poem[5] = "";
      linecntr = 0;                 /* init line counter */
      for (linecntr = 0; linecntr < 4; linecntr++)
      {
            chbd[linecntr].xmitbd0.txbdptr = poem[linecntr];/*    init    xmit
      pointer */
            chbd[linecntr].xmitbd0.txbdcnt = strlen(poem[linecntr]) + 1;/*
      init xmit cnt */
            chbd[linecntr].xmitbd0.txbdsac.R = 1;/* set xmit in BD */
      }
      for (v1 = 0; v1 < 3; v1++)
      {
            pdpr->ch[v1].CHAMR.MODE = 1;        /* select HDLC */
            pdpr->ch[v1].CHAMR.IDLM = 0;        /* no idles between frames */
            pdpr->ch[v1].CHAMR.ENT = 1;         /* enable channel xmit */
            pdpr->ch[v1].CHAMR.CRC = 1;         /* select 32-bit CRC */
            pdpr->ch[v1].CHAMR.NOF = 7;         /* 7 flags between frames */
            pdpr->ch[v1].CHAMR.POL = 1;         /* enable polling by RISC */
      }
 /* SCCE1 is cleared from reset */
      pdpr->SCCM1 = 0xF;            /* enable all intrpts */
      pdpr->CIMR.SCC1 = 1;          /* SCC1 interrupts enabled */
      pdpr->GSMR_L1.ENR = 1;        /* enable receiver */
      pdpr->GSMR_L1.ENT = 1;        /* enable transmit */
      while (pdpr->GSMR_L1.ENR == 1 | pdpr->GSMR_L1.ENT == 1)
      asm (" stop #$2000");         /* stop for next interrupt */
}

#pragma interrupt()                 /* make function an exception sr */
void SCC1esr()                      /* SCC1 exception service rtn */
{
      short er;                     /* event register scratchpad loc */
      asm(" move.w #$2300,sr");     /* decrement interrupt mask level */
      er = pdpr->SCCE1;             /* init event register scratchpad */
      pdpr->SCCE1 = er;             /* clear event register */
      if ((er & 8) == 8)            /* if interrupt table overflow */
```

6.3 Restarting the Transmitter

A global underrun may require the SCC transmitter to be restarted. However, for channel-specific errors, only the affected channel need be restarted. The following steps are required to restart each channel:

- Prepare buffer descriptors.
- Set the POL bit in the channel mode register.

A stopped, but not deactivated channel is started as described above. A deactivated channel must first have the ZISTATE and TSTATE reinitialized to their correct values, followed by setting TSATTx[V] and CHAMR[ENT]. Lastly, set CHAMR[POL] if the buffers are ready.

6.4 Restarting the Receiver

A global receiver overrun may require the SCC receiver to be restarted. However, for channel-specific errors, only the affected channel need be restarted. The following steps are required to restart each channel:

- Prepare buffer descriptors.
- Initialize the ZDSTATE to either 0x080 (HDLC) or 0x1800_0080 (transparent).
- Initialize the RSTATE to 0x3900_0000 for MH360 and 860MH.

6.5 Disabling Receiver and Transmitter

A transmit channel can be stopped from sending any more data to the line with the STOP command described in Section 3.1, “Transmit Commands.” The transmitter will continue to send IDLEs or FLAGs according to the channel mode register setting. To deactivate a channel, the V bit has to be cleared in the time slot assignment table and the ENT bit has to be cleared in the channel mode register.

To stop a channel while receiving, use the STOP command as described in Section 3.2, “Receive Commands,” then perform a restart as described above.

6.6 Debugging Hints

Note that the following guidelines are subject to change; code should not rely on this information. The hints are for debugging purposes only.

6.6.1 Pointer Registers

Table 6-7 discusses the debugging hints for pointer registers. See Section 2.4.1, “Channel-Specific HDLC Parameters,” and See Section 2.4.2, “Channel-Specific Transparent Parameters,” for more information.

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

A.2 Registers in HDLC Mode

Figure A-3 to Figure A-6 show the 68360 bit numbering for registers in HDLC mode.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODE	0	IDLM	ENT	RESERVED			POL	CRC	0	RESERVED		NOF			
RESET:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure A-3. CHAMR—Channel Mode Register (HDLC)

Interrupt Table Entry:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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:

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

Figure A-4. Interrupt Table Entry and INTMSK (HDLC)

31	30	29	28	27	26	25	24
0	0	1	MOT	FC3–FC0			

Figure A-5. TSTATE (HDLC)

31	30	29	28	27	26	25	24
0	0	1	MOT	FC[3–0]			

Figure A-6. RSTATE (HDLC)

Appendix B
Frequently-Asked Questions

This appendix provides a list of frequently-asked questions and solutions for the MH360 and 860MH.

B.1 Questions Common to MH360 and 860MH

Q: What are the performance differences between the 68MH360 and 860MH?

A: Since the 860 and 360 have the same CPM, performance scales linearly with frequency as shown in Table B-1.

Table B-1. CPU Performance

Device	Frequency	Performance
MH360	25 MHz	6.3 Dhrystone MIPS
	33 MHz	8.4 Dhrystone MIPS
PowerQUICC	25 MHz	33 Dhrystone MIPS
	40 MHz	52 Dhrystone MIPS
	50 MHz	66 Dhrystone MIPS

Q: Comparing the MH360 and 860MH, what is the best way to support two T1 TDM channels?

A: There are two bottlenecks in running the QMC protocol on the 360 and 860. One is CPM performance, and the other is parameter RAM space.

Running the QMC protocol consumes nearly all of the CPM bandwidth at 25 MHz. As the CPM is the same for the 360 and 860, the same limitation applies to both when running at 25 MHz.

In addition, the QMC protocol consumes nearly all of the dual-ported RAM of a 360 preventing more than one QMC from running at the same time (even if the 360's speed could be increased).

However, since the 860 has twice as much dual-ported RAM as the 360, it does have enough space to run two QMCs. Also, the CPM bandwidth is doubled at 50 MHz.

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.

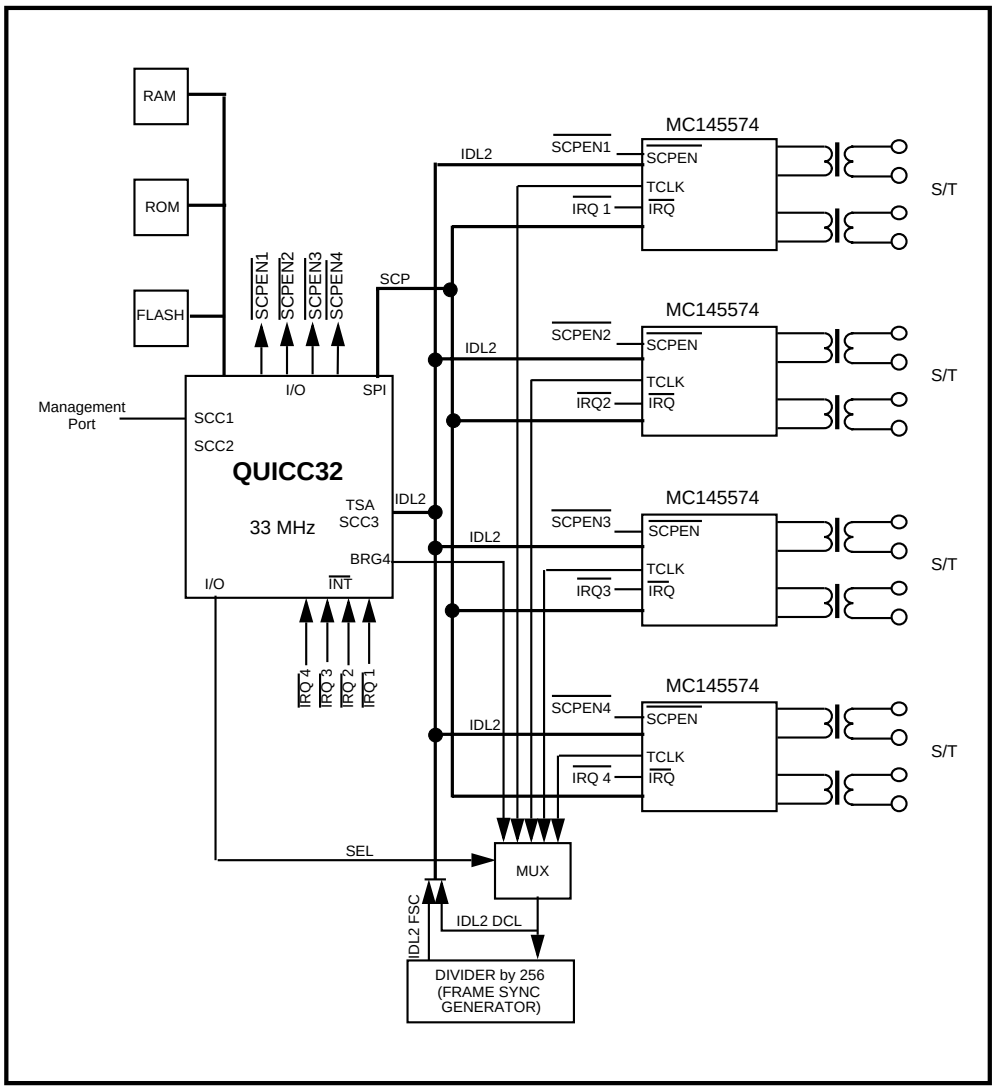

Figure C-6. Connection between Four S/T Interfaces and the QUICC32

Overview 1

QMC Memory Organization 2

QMC Commands 3

QMC Exceptions 4

Buffer Descriptors 5

QMC Initialization 6

Features Deleted in MC68MH360 7

Performance 8

Multi-Subchannel (MSC) Microcode 9

68360 Bit Numbering A

Frequently-Asked Questions B

Connecting ISDN Interfaces to QUICC32 C

Index IND