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	357-BBGA
Supplier Device Package	357-PBGA (25x25)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68mh360vr25lr2

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

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.

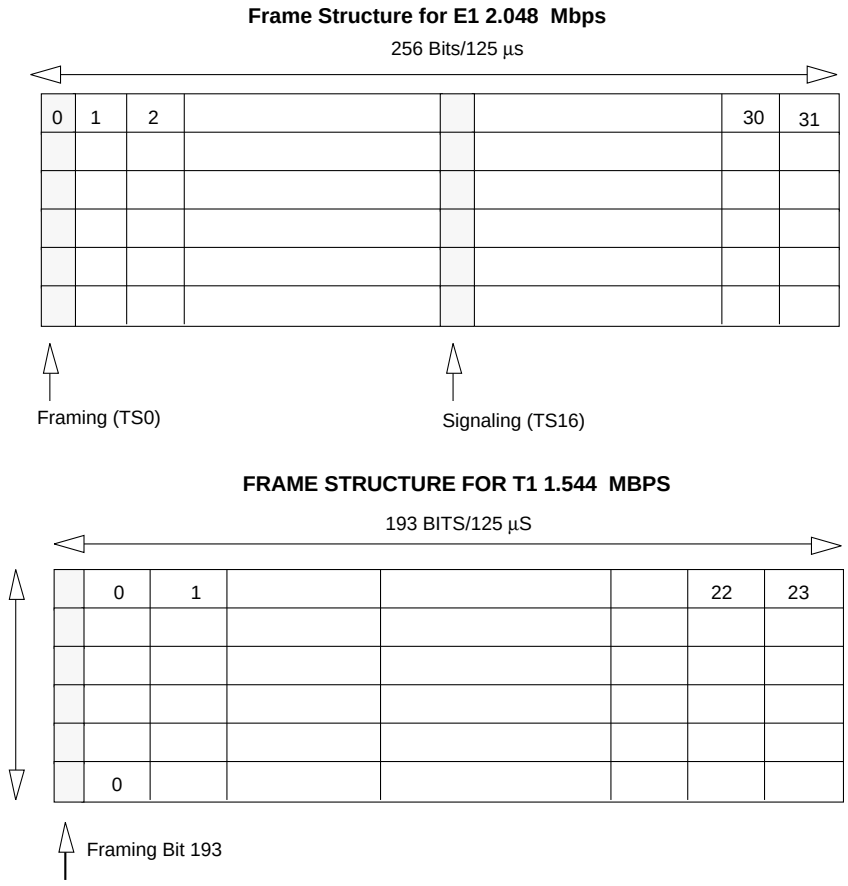


Figure 1-8. Frame Structures for E1/CEPT and T1 TDM Interfaces

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.

2.1.1 Dual-Ported RAM Base

The MC68MH360's internal memory is mapped into an 8-Kbyte block of memory, and the starting address is dictated by the DPRBASE programmed in the MBAR register. For more detail on the QUICC internal memory structure, see Section 3 of *MC68360 Quad Integrated Communications Controller User's Manual*. The MPC860MH has its internal memory mapped into a 16-Kbyte block of memory. The ISB programmed in the IMMR register determines the starting address of this memory block. For more information on the PowerQUICC internal memory structure, see Section 3 of *MPC860 PowerQUICC User's Manual*. All internal registers are addressed as offsets within the dual-ported RAM; therefore, all pointers are relative to this base address.

2.1.2 SCC Base and Global Multichannel Parameters

The SCC base points to the start of the parameter RAM for each of the SCCs at 256-byte intervals. On the MC68MH360, each SCC has 192 bytes of parameter RAM; each SCC on the MPC860MH has 256 bytes. When the QMC protocol is enabled on an SCC, its parameter RAM is used to store the global multichannel parameters for all the logical channels. This area contains parameters and pointers that are common to all channels.

NOTE

As the QMC requires 0xAF bytes of parameter RAM for its global multichannel parameters, this may cause conflict with other CPM functionality. For example, when using the MPC860MH with SCC1 in QMC mode, I^2C is unavailable.

2.1.3 TSATRx/TSATTx Pointers and Time Slot Assignment Table

The time slot assignment table pointers are within the global multichannel parameters. There are two pointers—Tx_S_PTR for transmit and Rx_S_PTR for receive. The Rx_S_PTR is normally set to SCC Base + 20; this is the normal location of the receive time slot assignment table. The Tx_S_PTR is normally set to SCC Base + 60; this is the normal location of the transmit time slot assignment table. However, if the receiver and the transmitter have the same mapping for the logical channels, Tx_S_PTR can point to SCC base + 20 so that Rx and Tx have a common time slot assignment table. Note that if a single TDM channel is routed to more than one SCC, they may also use just one time slot assignment table for all SCCs. See Section 2.3, "Multiple SCC Assignment Tables," for more information. The time slot assignment table holds one 32-bit entry for each time slot. It has options for subchanneling, a valid bit, and a logical channel pointer. For 64-channel support there is only space for one table; therefore, common Rx and Tx parameters will need to be used unless one of the TSA tables can be accommodated elsewhere in memory, such as in the parameter RAM area of another SCC. Associated with the Rx/Tx_S_PTR are the Rx/TxPTR pointers that are maintained by the CPM and point to the current time slot.

2.1.4 TSATRx/TSATTx Channel Pointers

The channel pointers are 12-bit pointers to the channel-specific parameters in the internal dual-ported RAM. These should not be confused with TSATRx/TSATTx pointers as described in Section 2.1.3, “TSATRx/TSATTx Pointers and Time Slot Assignment Table.” The 6 most-significant bits of the address are taken from the time slot assignment table. For the MH360, the most-significant bit must be zero as the addressing range is only 2 Kbytes. The 6 least-significant bits are zero, mapping out a 64-byte area for each of the channel-specific parameters. The channel-specific parameters are common for Rx and Tx. For 32-channel support, 2 Kbytes of dual-ported RAM is required (32 * 64), and for 64-channel support, 4 Kbytes of dual-ported RAM is required (64 * 64). In most cases, time slot 0 channel pointer will address the base of dual-ported RAM for logical channel 0, and time slot 1 channel pointer would address the base of dual-ported RAM + 4 for logical channel 1. In Figure 2-2, time slot 5 channel pointer addresses logical channel 5, requiring the channel pointer being set to 0b000101.

NOTE

It is possible to concatenate multiple time slots to one logical channel. This is achieved by setting the channel pointers of the grouped time slots to the same logical channel.

2.1.5 Logical Channel TBASE and RBASE

TBASE and RBASE are within the channel-specific parameters. TBASE is the Tx buffer descriptor base address, and RBASE is the Rx buffer descriptor base address. These 16-bit offsets from MCBASE point to individual logical channel’s buffer descriptors located within the buffer descriptor table. Note that there are individual TBASE and RBASE values for each logical channel.

2.1.6 MCBASE

MCBASE is located in the global multichannel parameters. Each SCC has a unique MCBASE value pointing to the base of the SCC’s buffer descriptor table in external memory. For example, the address of logical channel five’s Tx buffer descriptor table is MCBASE + logical channel five TBASE.

2.1.7 Buffer Descriptor Table

A buffer descriptor table for each SCC is located in a 64-Kbyte area of external memory. This block size is determined by the TBASE and RBASE addressing range. The memory segment must be long-word-aligned but can start anywhere in memory. Each SCC has a maximum of 16,384 (64 Kbytes memory ÷ 4-byte pointers) buffers. For a 32-channel implementation, each logical channel has a maximum of 256 (16,384 / (32 * 2)) buffers for receive and 256 buffers for transmit. For each logical channel, there is a circular queue with programmable start address and length.

Table 2-1. Global Multichannel Parameters (Continued)

Offset to SCC Base	Name	Width (Bits)	Description
18	**Rx_S_PTR**	16	Rx time-slot assignment table pointer (default = SCC base + 20 in normal mode)—This global QMC parameter defines the start value of the TSATRx table, which must be present only once per SCC global area. Other SCCs may access this location.
1A	**TxPTR**	16	TxPTR (initialize to SCC Base + 60)—This global parameter is a RISC variable that points to the current transmitter time slot. The host must initialize it to the starting location of TSATTx. The RISC processor increments this pointer whenever it completes the processing of a transmitter time slot.
1C	**C_MASK32**	32	CRC constant (0xDEBB20E3)—Required to calculate 32-bit CRC-CCITT. C_MASK32 is written by the host during QMC initialization. It is used for 32-bit CRC-CCITT calculation if HDLC mode of operation is chosen for a selected channel. (This is a programmable option. For each HDLC channel, one of two CRCs can be chosen, as programmed in CHAMR.) For more information, see Section 2.4.1, “Channel-Specific HDLC Parameters,” and Table 2-5. This entry must have a correct value if at least one HDLC channel is used; otherwise, it can be cleared (0).
20	**TSATRx**	32 Entries x 16	Time slot assignment table Rx—Host-initialized, 16-bit-wide table with 32 entries that define mapping of logical channels to time slots for the QMC receiver. The QMC protocol looks at chunks of 8 bits regardless of whether they come from one physical time slot of the TDM or whatever other combination of bits the TSA transfers to the SCC. These 8 bits are referred to as a time slot in the assignment table. It is recommended but not required to route all bits from the TDM to the SCC and to do all enabling and masking in the time-slot assignment table. See Figure 2-3.
60	**TSATTx**	32 Entries x 16	Time slot assignment table Tx—Maps a specific logical channel to each physical time slot. Time slot assignment table Tx is a host-initialized, 16-bit table with 32 entries that define the mapping of channels to time slots for the QMC transmitter. The QMC protocol looks at chunks of 8 bits regardless if they go to one physical time slot of the TDM or whatever other combination of bits are transferred from the SCC to the TDM through the TSA. These 8 bits are referred to as a time slot in the assignment table. It is recommended but not required to route all bits from the TDM to the SCC and to do all enabling and masking in the time slot assignment table. See Figure 2-3.
A0	**C_MASK16**	16	CRC constant (0xF0B8)—Required to calculate 16-bit CRC-CCITT. This constant is written by the host during QMC initialization. It is used for 16-bit CRC-CCITT calculation if HDLC mode of operation is chosen for a selected channel. (This is a programmable option. For each HDLC channel, one of two CRCs can be chosen, as programmed in CHAMR.) For more information, see Section 2.4.1, “Channel-Specific HDLC Parameters,” and Table 2-5. This entry must have a correct value if at least one HDLC channel is used; otherwise, it can be cleared (0).
A4	TEMP_RBA	32	Temporary receive buffer address
A8	TEMP_CRC	32	Temporary cyclic redundancy check

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. Host-initialized to 0x3800_0000—FC = 8, Motorola mode for MH360. Host-initialized to 0x3000_0000— AT = 0, Motorola mode for 860MH. 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.

Table 2-4. Channel-Specific HDLC Parameters (Continued)

Offset	Name	Width (Bits)	Description
22	**MFLR**	16	Maximum frame length register (host-initialized)—Defines the longest expectable frame for this channel. Its maximum value is 64 Kbytes. The remainder of a frame which is larger than MFLR is discarded and a flag in the last frame's BD is set (LG). An interrupt request (RXF and RXB) might be generated depending on the interrupt mask. The frame length is considered to be everything between flags, including CRC. MFLR is checked every long word, but the content may be on any number of bytes. If MFLR is set to 5 for example, checking is done when 8 bytes have been received. At this point, the SDMA transfers the long word to memory, and all 8 bytes will be in the receive buffer. Also at this point the MFLR violation (>5) is detected and the interrupt may be generated. No more data will be written into this buffer when the MFLR violation is detected.
24	**RSTATE**	32	Rx internal state —Initialize to 0x3900_0000 FC=9, Motorola mode for MH360 or initialize to 0x3100_0000 AT=1, Motorola mode for 860MH. See Section 2.4.1.4, "RSTATE—Rx Internal State (HDLC)," for more information.
28	—	32	Rx internal data pointer—Points to current address of specific channel.
2C	**RBPTR**	16	Rx buffer descriptor pointer (host-initialized to RBASE prior to operation or due to a fatal error)—Contains the offset from MCBASE to the current receive buffer. See Table 2-1. MCBASE + RBPTR gives the address for the BD in use.
2E	—	16	Rx internal byte count—Per Channel: Number of remaining bytes in buffer
30	RPACK	32	(Rx Temp) Packs 4 bytes to 1 long word before writing to buffer.
34	**ZDSTATE**	32	Zero deletion machine state—(Host-initialized to 0x0000_0080 in HDLC mode, 0x1800_0080 in transparent mode, prior to operation and after a fatal Rx error (global overrun, busy) before channel initialization.)—Contains the previous state of zero deletion state machine. The middle 2 bytes, represented by zeros in the initialization value above, hold the received pattern during reception. A window of 16 bits shows the history of what is received on this logical channel. More information is given in the application note section.
38	RCRC	32	Temp receive CRC—Temp value of CRC calculation result
3C	MAX_cnt	16	Max_length counter—Count length remaining
3E	TMP_MB	16	Temp—Holds MIN(MAX_cnt, Rx internal byte count)

2.4.1.1 CHAMR—Channel Mode Register (HDLC)

The channel mode register is a word-length, host-initialized register. Figure 2-7 shows the channel mode register for HDLC operation.

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

Notes: 1. All bits that are defined as reserved should be cleared (0).
2. For the 68360, the bit numbering is reversed. See Appendix A for more information.

Figure 2-7. CHAMR—Channel Mode Register (HDLC)

4.1.1 Global Underrun (GUN)

The QMC performs the following actions when it detects a GUN event:

- Transmits an abort sequence of minimum sixteen 1's in each time slot.
- Generates an interrupt request to the host (if enabled) and sets the GUN bit in the SCCE register.
- Stops reading data from buffer.
- Sends IDLEs or FLAGs in all time slots depending on channel mode settings until the host does the following:

  Host initializes all transmitting channels and time slots by preparing all buffer descriptors for transmission (R bits are set) and setting the POL bit. No other re-initialization is needed.

4.1.2 Global Overrun (GOV) in the FIFO

A global overrun affects all channels operating from an SCC. Following GOV, the QMC performs the following:

- Updates the RSTATE register to prevent further reception on this channel. Bit 20 in the RSTATE register indicates that the receiver is stopped.
- Generates an interrupt request to the host (if enabled) and sets the GOV bit in the SCCE.
- Stops writing data to all channel's buffers.
- Waits for host to initialize all the receiving channels by setting first the ZDSTATE followed by the RSTATE to their initial values.

4.1.3 Restart from a Global Error

The last two bullets in the above two sections describe the only steps necessary for re-initialization. The transmit and receive sections must be restarted individually for each separate logical channel.

For details about initialization, see Chapter 6, "QMC Initialization."

4.2 SCC Event Register (SCCE)

The QMC's SCCE is a word-length register used to report events and generate interrupt requests. See Figure 4-2 and Table 4-1 for SCCE field descriptions. For each of its flags, a corresponding programmable mask/enable bit in the SCCM determines whether an interrupt request is generated. If a bit in the SCCM register is zero, the corresponding interrupt flag does not survive, and the CPM does not proceed with its usual interrupt handling. If a bit in the SCCM is set, the corresponding interrupt flag in the SCCE survives, and the SCC event bit is set in the CPM interrupt-pending register. See Figure 4-3 for SCCM assignments.

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

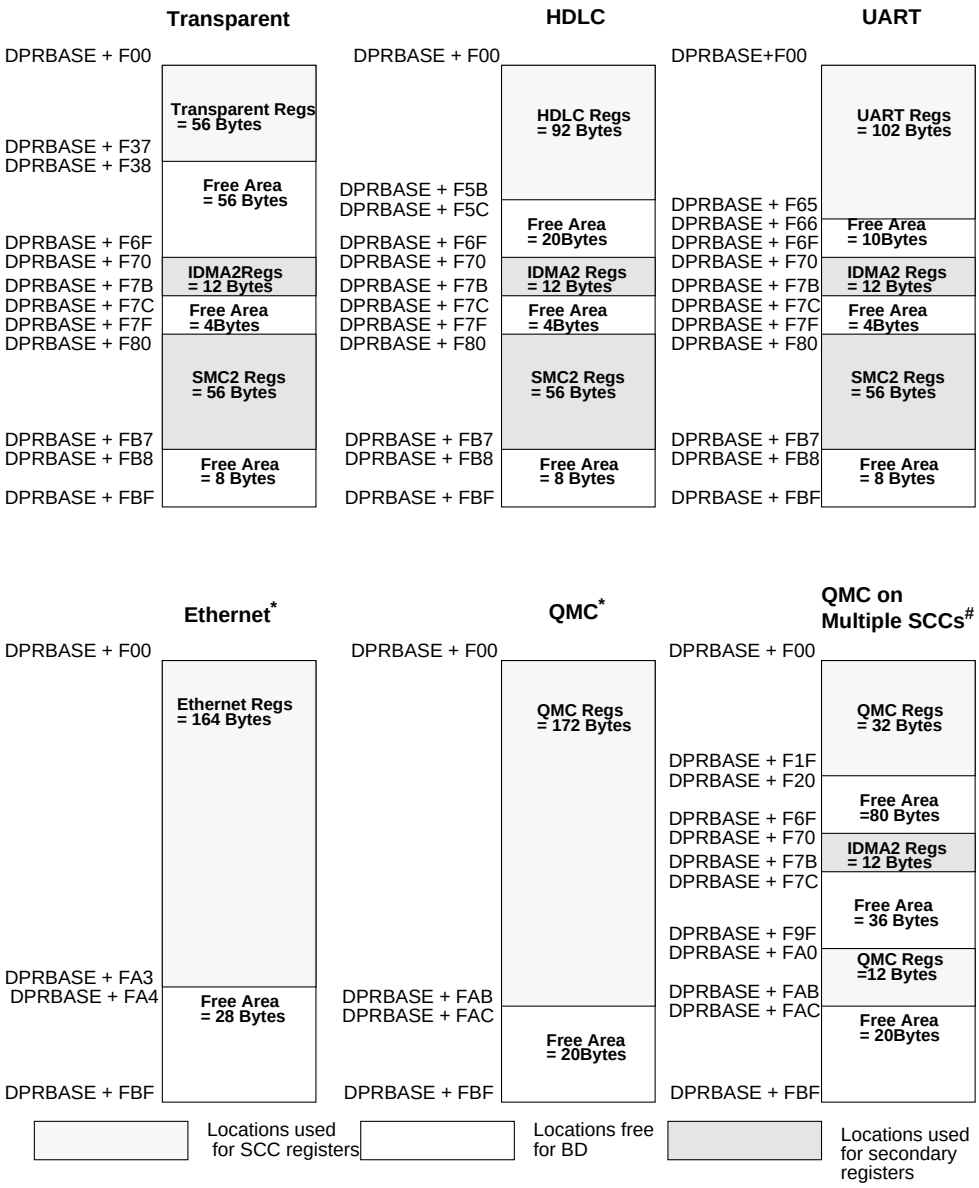
Field	Name	Description
14	TXB	Tx buffer 0 = No TXB event has occurred. 1 = A buffer has been completely transmitted. This bit is set (and an interrupt request is generated) as soon as the programmed number of PAD characters (or the closing flag, for PAD = 0) is written to the SCC's transmit FIFO. The number of PAD characters determines the timing of the TXB interrupt in relation to the closing flag sent out at TXD. See Chapter 5, “Buffer Descriptors,” for an explanation of PAD characters and their use.
15	RXB	Rx buffer 0 = No RXB event has occurred. 1 = A buffer has been received on this channel.

4.4 Channel Interrupt Processing Flow

Figure 4-5 illustrates the flow of a channel interrupt. Note that this does not describe the processing of the global interrupts GUN and GOV.

Table 5-1. Receive Buffer Descriptor (RxBD) Field Descriptions (Continued)

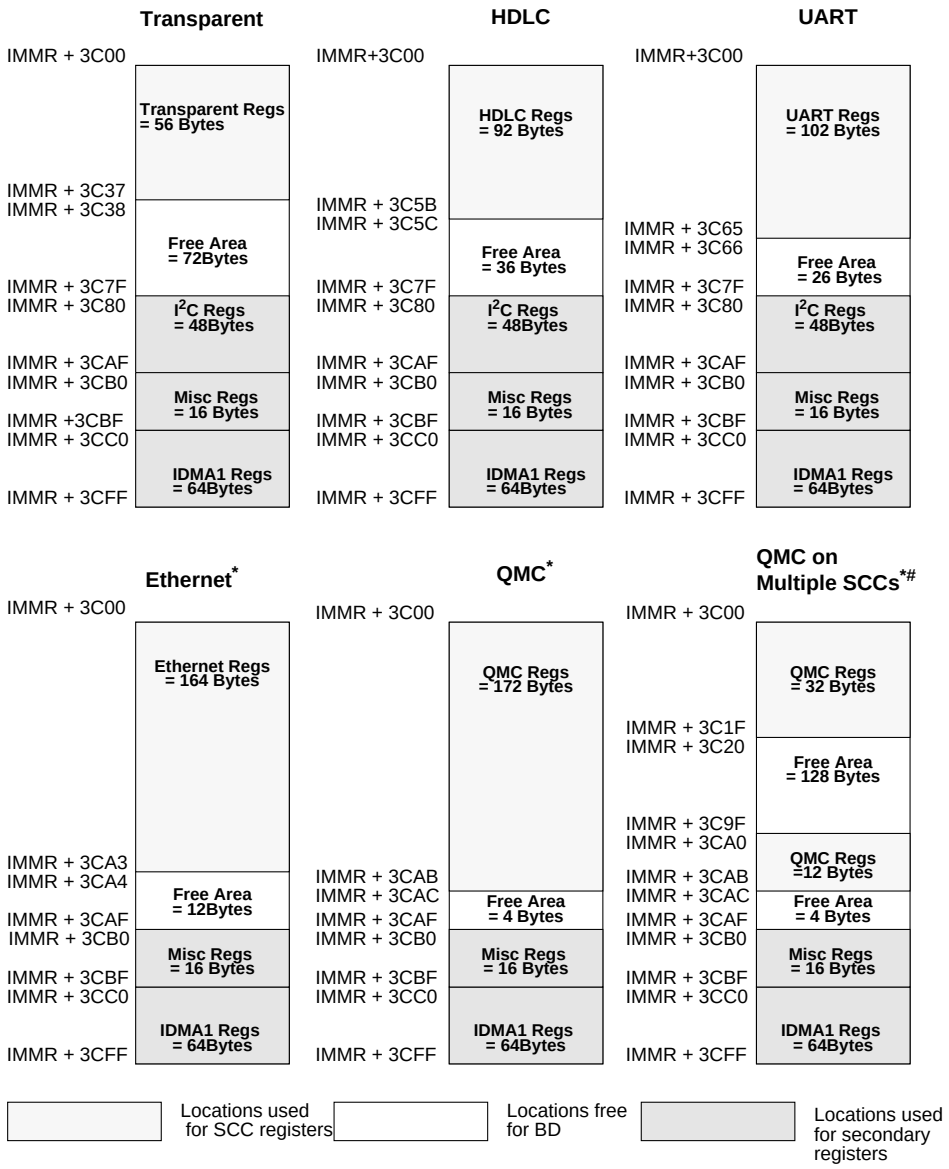
Field	Name	Description
1	—	—
2	**W**	Wrap (final buffer descriptor in table) 0 This is not the last buffer descriptor in the RxBD table. 1 This is the last buffer descriptor in the RxBD table. After this buffer is used, the CPM receives incoming data into the first buffer descriptor in the table (the buffer descriptor pointed to by RBASE). The number of RxBDs in this table is programmable and is determined only by the wrap bit and by the space constraints of the dual-ported RAM.
3	**I**	Interrupt 0 The RXB bit will not be set after this buffer has been used, but RXF operation remains unaffected. 1 The RXB bit (and/or the RXF bit in HDLC mode) of the interrupt table entry will be set when this buffer has been used by the HDLC controller. These two bits may cause interrupts (if enabled).
4	L	Last-in-frame (HDLC mode only)—The HDLC controller sets L when this buffer is the last in a frame. This implies the receipt of a closing flag or reception of an error, in which case one or more of the CD, OV, AB, and LG bits are set. The HDLC controller writes the number of frame octets to the data length field. 0 This buffer is not the last in a frame. 1 This buffer is the last in a frame.
5	F	First-in-frame—The controller sets this bit when this buffer is the first in a frame. 0 The buffer is not the first in a frame. 1 The buffer is the first in a frame.
6	**CM**	Continuous Mode 0 Normal operation. 1 The empty bit is not cleared by the CPM after this buffer descriptor is closed, allowing the associated data buffer to be overwritten automatically when the CPM next accesses this buffer descriptor. The empty bit is not cleared if an error occurs during reception. The user must terminate continuous mode by clearing this bit.
7	—	—
8	UB	User bit—The CPM never touches, sets, or clears this user-defined bit. The user determines how this bit is used. For example, it can be used to signal between higher level protocols whether a buffer has been processed by the CPU.
9	—	—
10	LG	Rx frame length violation (HDLC mode only)—A frame length greater than the maximum value was received in this channel. Only the maximum-allowed number of bytes, MFLR rounded to the nearest higher longword alignment, are written to the data buffer. This event is recognized as soon as the MFLR value is exceeded when data is long-word-aligned. When data is not long-word-aligned, this interrupt occurs when the SDMA writes 32 bits to memory. The worst-case latency from MFLR violation until detected is 3-byte timings for this channel. When MFLR violation is detected, the receiver is still receiving even though the data is discarded. The buffer is closed when a flag is detected, and this is considered to be the closing flag for this buffer. At this point, LG = 1 and an interrupt may be generated. The length field for this buffer includes everything between the opening flag and this last identified flag.



#—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



#—TSATRx & TSATTx are mapped to an alternative SCC's parameter RAM.
***—I²C is not available in these configurations due to memory conflict.**

Figure 5-12. MPC860MH SCC1 Parameter RAM Usage

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

Step 16. Initialize channel-specific parameters for HDLC or transparent mode as follows. For more information on HDLC, see Section 2.4.1, “Channel-Specific HDLC Parameters,” and for transparent mode, see Section 2.4.2, “Channel-Specific Transparent Parameters.” Repeat for each of the enabled channels.

- TBASE: TxBD descriptors base address. Initialize to location of first TxBD = MCBASE+TBASE.
  RBASE: RxBD descriptors base address. Initialize to location of first RxBD = MCBASE+RBASE.

```
ch[0].RBASE = 0;              /* locate CH0 RxBDS at 0 */
ch[0].TBASE = 8;              /* locate CH0 TxBDS at 8 */
ch[1].RBASE = 0x10;           /* locate CH1 RxBDS 0x10 */
ch[1].TBASE = 0x18;           /* locate CH1 TxBDS 0x18 */
ch[2].RBASE = 0x20;           /* locate CH2 RxBDS 0x20 */
ch[2].TBASE = 0x28;           /* locate CH2 TxBDS 0x28 */
ch[3].RBASE = 0x30;           /* locate CH3 RxBDS 0x30 */
ch[3].TBASE = 0x38;           /* locate CH3 TXBDS 0x38 */
```

  Copy RBASE to RBPTR (Rx buffer descriptor pointer) and TBASE to TBPTR (Tx buffer descriptor pointer).

```
ch[x].TBPTR = ch[x].TBASE;
ch[x].RBPTR = ch[x].RBASE;
```

- TSTATE: Tx internal state. For the MH360, this is typically 0x3800_0000. For the 860MH, this is typically 0x3000_0000.

```
ch[x].TSTATE = 0x3800_0000;   /* setting for MH360 */
ch[x].TSTATE = 0x3000_0000;   /* setting for 860MH */
```

- RSTATE: Rx internal state. For the MH360, this is typically 0x3900_0000. For the 860MH, this is typically 0x3100_0000.

```
ch[x].RSTATE = 0x3900_0000;   /* setting for MH360 */
ch[x].RSTATE = 0x3100_0000;   /* setting for 860MH */
```

- ZISTATE: zero insertion should be initialized to 0x0000_0100.

```
ch[x].ZISTATE = 0x100;
```

- ZDSTATE: zero deletion machine state should be initialized to 0x1800_0080 for transparent mode or 0x0000_0080 for HDLC.

```
ch[x].ZDSTATE = 0x80;         /* set ZDZTATE for HDLC */
```

- INTMSK: channel’s interrupt mask flags. Bits should be set to enable the corresponding interrupts.

```
ch[x].INTMSK = 0xA;
```

```
        {};
        if ((er & 4) == 4)            /* if global interrupt occurred */
        {};
        if ((er & 2) == 2)            /* if global underrun */
        {};
        if ((er & 1) == 1)            /* if global overrun */
        {};
        pdpr->CISR = 0x4000_0000;     /* clear in-service bit */
}
getmbar()
{
        asm(" move.w #7,d0");         /* CPU space func code to d0 */
        asm(" movec.l d0,sfc");       /* load SFC for CPU space */
        asm(" lea.l $3ff00,a0");      /* A0 points to MBAR */
        asm(" moves.l (a0),d0");      /* get MBAR */
}
getvbr()
{
        asm(" movec.l vbr,d0");       /* get vector base reg value */
}
inittsatr(maxts)
short maxts;
{
        short curts;
        for (curts = 0; curts < maxts; curts++)
        {
                pdpr->SCC1.TSATR[curts].W = 0;      /* not last time slot */
                pdpr->SCC1.TSATR[curts].CP = curts;/* mark chan nmbr */
                pdpr->SCC1.TSATR[curts].mask7_6 = 3;/* no subchaneling */
                pdpr->SCC1.TSATR[curts].mask5_0 = 0x3F;/* no subchnlng */
                pdpr->SCC1.TSATR[curts].V = 1;      /* mark time slot valid */
        }
}
```

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

Example #1

A device is operating at 25 MHz. SCC1 runs 1x10-Mbps Ethernet in half duplex, SCC2 runs 1 x 2-Mbps HDLC, SCC3 runs 1 x 64-Kbps HDLC, SCC4 runs 1 x 9.6-Kbps UART and SMC1 runs 1 x 38-Kbps SMC UART. The following equation applies:

$$\left(\frac{10}{22}\right)+\left(\frac{2}{8}\right)+\left(\frac{0.064}{2.4}\right)+\left(\frac{0.0096}{2.4}\right)+\left(\frac{0.038}{0.22}\right) = 0.96 \qquad \textbf{(<1)}$$

This yields a percentage CPM utilization of 96% meaning the device can handle these protocols at this frequency. Note the 9.6-Kbps UART link only requires 0.4% of the CPM bandwidth, implying that in any configuration where there is free bandwidth that it will be possible to run a low-rate UART link.

Example #2

A device operating at 25 MHz is required to run 24 channels at 64 Kbps in QMC mode with an additional 2 HDLC channels operating at 128 Kbps each. The following equation applies:

$$\left(\frac{2\times 0.128}{8}\right)+\left(\frac{24\times 0.064}{2.1}\right) = 0.76 \qquad \textbf{(<1)}$$

Example #3

The last example shows an application with 32 QMC channels and one additional 2-Mbps HDLC channel. The following equation applies:

$$\left(\frac{2}{8}\right)+\left(\frac{32\times 0.064}{2.1}\right) = 1.22 \qquad \textbf{(will not work)}$$

Since the result above is greater than 1, this configuration may not work at 25 MHz. If a 33-MHz operation is used, CPM utilization will drop below 1, allowing the combination to be supported. The following equation applies:

$$1.22\times\left(\frac{25}{33}\right) = 0.92 \qquad \textbf{(<1)}$$

In general, a channel combination will work if the combined load is less than 1. The equations are scalable to frequency with the exception of the nonlinear Ethernet protocol. Taking Ethernet into account is difficult. Designers will need to benchmark performance for results near 1.

C.3.2.1 Activation Procedure

During the initialization, the FREQREF signal of each U interface is enabled.

A multiplexer commanded by the QUICC32 is used to select the U interface clock master.

When the first MC145572 is activated, its FREQREF signal synchronizes to the network. The MC145572 then sends an interrupt to the QUICC32 (IRQ1— register NR3[1]— meaning uao = 1 has been received) indicating that the activation process has begun. Before responding to LT with act = 1 (which will enable the data transfer), the QUICC32 can select, through the multiplexer, this particular FREQREF signal to be the clock master.

Since the QUICC32 has the initiative to enable the data transfer, there is no timing constraint to react to the interrupt.

C.3.2.2 Deactivation Procedure

According to the ANSI specification T1.601-1988, prior to deactivating, the LT should notify the NT of the pending deactivation by clearing the M4 channel dea bit towards the NT for at least three superframes. Then, the NT can be deactivated by sending a deactivation request.

The MC145572 not only has the ability to generate an interrupt after the reception of the third dea bit = 0, but also after the reception of the second dea bit = 0.

When the clock-master U interface is deactivated, the QUICC32 receives an interrupt indicating that the second dea bit = 0 has been received. The QUICC32 has then the ability to select another activated U interface (if there is one), to be the clock master. The QUICC32 has 12 ms (1 superframe) until receiving the next dea bit = 0, indicating the pending deactivation, and therefore 12 ms to react to the interrupt.

If none of the U interfaces are activated, no change in the multiplex selection is required.

C.3.3 System Configuration

The following sections provide a checklist of the main features that need to be configured for each device.

C.3.3.1 S/T-Interface Configuration

Do the following for an S/T-interface configuration:

- IDL2 with time slot assigner (TSA enabled in reg. OR6[5–7]; TSA selection in OR0 to OR5)
- Slave mode (DCL & FSC are input) - (pin M/$\overline{S}$ to GND)
- TCLK enabled at 2.048 MHz (OR7[5] = 1; BR13[5] = 0; BR7[2] = 1)
- D channel contention procedure disabled (BR7[6] = 1)