NXP USA Inc. - MC68MH360CVR25L Datasheet

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	-40°C ~ 85°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/mc68mh360cvr25l

1.3 QMC Features

- MC68MH360-specific features
  - Up to 32 independent communication channels
  - Arbitrary mapping of any of 0–31 channels to any of 0–31 TDM time slot
  - Can support arbitrary mapping of any of 0–31 channels to any of 0–63 TDM time slots in case of common Rx and Tx mapping
  - Up to three additional HDLC 64-Kbps channels at 25-MHz system clock
  - Simultaneous Ethernet support at 33-MHz system clock
  - Up to 64 DMA channels with linear buffer array
- MPC860MH/DH-specific features
  - Up to 64 independent communication channels
  - Arbitrary mapping of any of 0–63 channels to any of 0–63 TDM time slots
  - Supports arbitrary mapping of any of 0–63 channels to any of 0–127 TDM time slots in case of common Rx and Tx mapping
  - Two simultaneous 32-channel E1 links at 50-MHz system clock
  - Up to 128 DMA channels with linear buffer array
- Common features
  - Independent mapping for receive/transmit
  - Supports either transparent or HDLC protocols for each channel
  - Interrupt circular buffer with programmable size and overflow identification
  - Global loop mode
  - Individual channel loop mode through the SI
  - Programmable frame length (via SI)
- Serial interface
  - Serial-multiplexed (full duplex) input/output 2048-, 1544-, or 1536-Kbps PCM highways
  - Compatible with T1/DS1 24-channel and CEPT E1 32-channel PCM highway, ISDN basic rate, ISDN primary rate and user-defined
  - Subchanneling on each time slot
  - Allows independent transmit and receive routing, frame syncs, and clocking
  - Concatenation of any, not necessarily consecutive, time slots to channels independently for receive/transmit
  - Supports H0, H11, and H12 ISDN channels
  - Allows dynamic allocation of channels

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

1.4 The Time Slot Assigner and the QMC

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

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

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

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

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

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

1.5 The Serial Interface (SI)

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

2.1 QMC Memory Structure

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

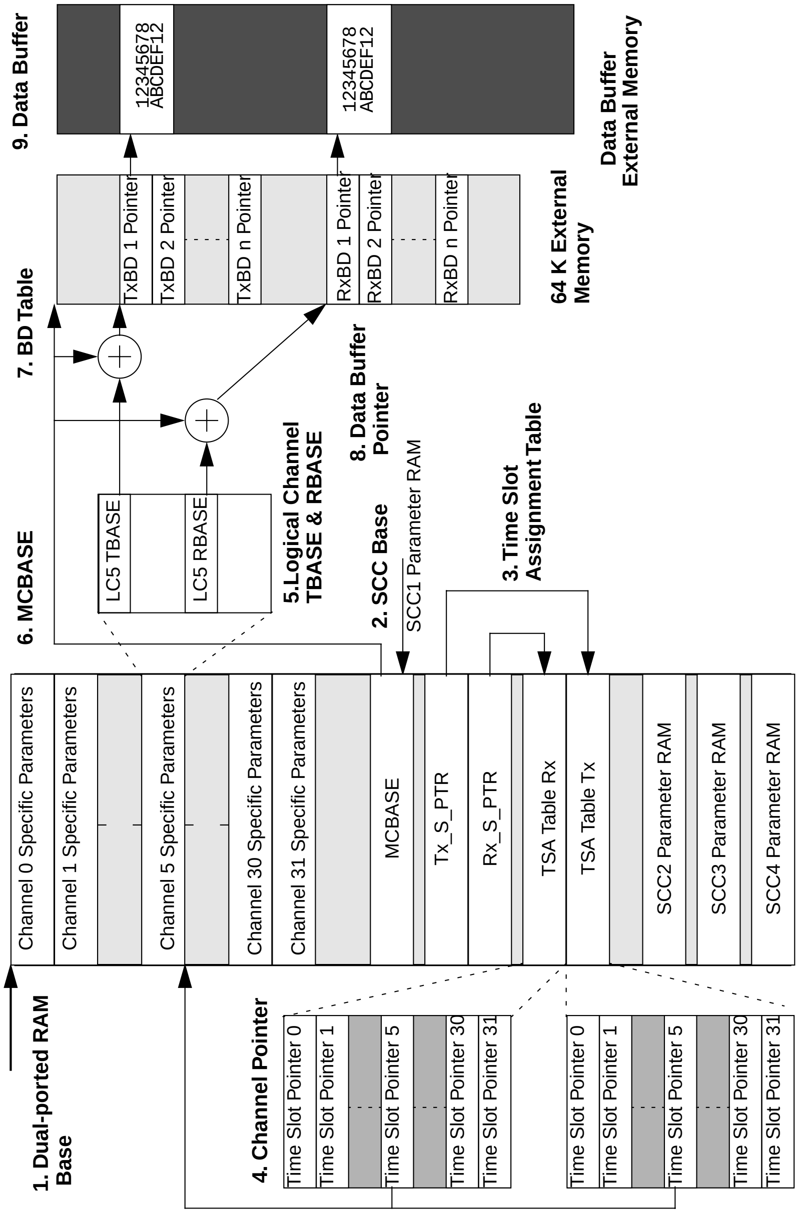
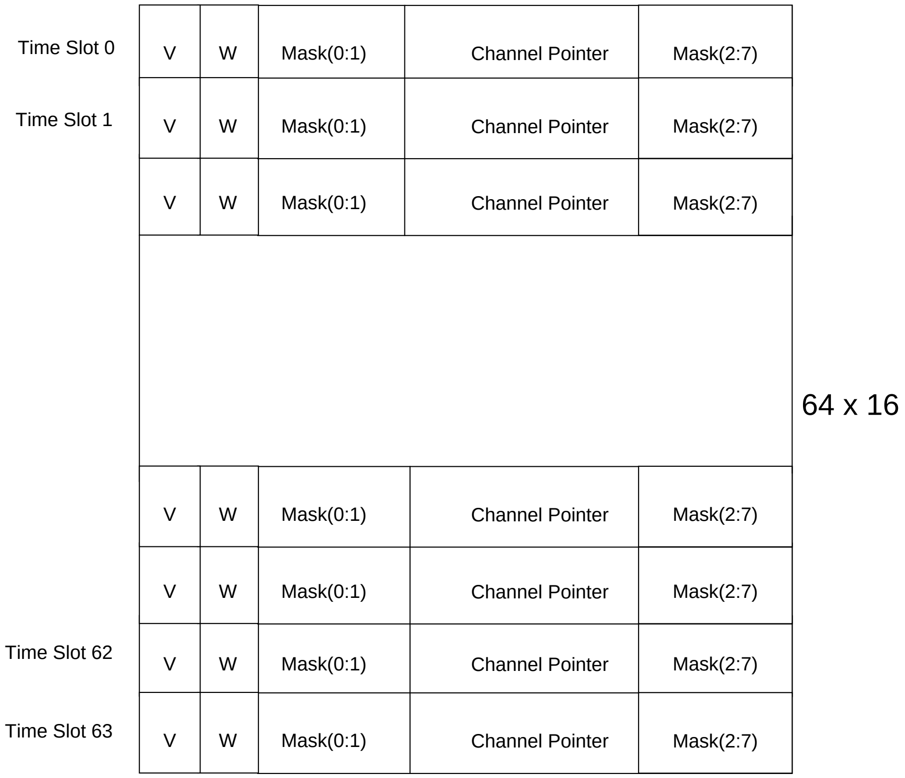


Figure 2-2. QMC Memory Structure

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



Time Slot 0	V	W	Mask(0:1)	Channel Pointer	Mask(2:7)
Time Slot 1	V	W	Mask(0:1)	Channel Pointer	Mask(2:7)
	V	W	Mask(0:1)	Channel Pointer	Mask(2:7)
	V	W	Mask(0:1)	Channel Pointer	Mask(2:7)
	V	W	Mask(0:1)	Channel Pointer	Mask(2:7)
Time Slot 62	V	W	Mask(0:1)	Channel Pointer	Mask(2:7)
Time Slot 63	V	W	Mask(0:1)	Channel Pointer	Mask(2:7)

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

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

2.3 Multiple SCC Assignment Tables

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

Table 2-9. RSTATE 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.

2.4.2 Channel-Specific Transparent Parameters

Table 2-10 describes the channel-specific transparent parameters. Boldfaced parameters must be initialized by the user.

Table 2-10. Channel-Specific Transparent Parameters

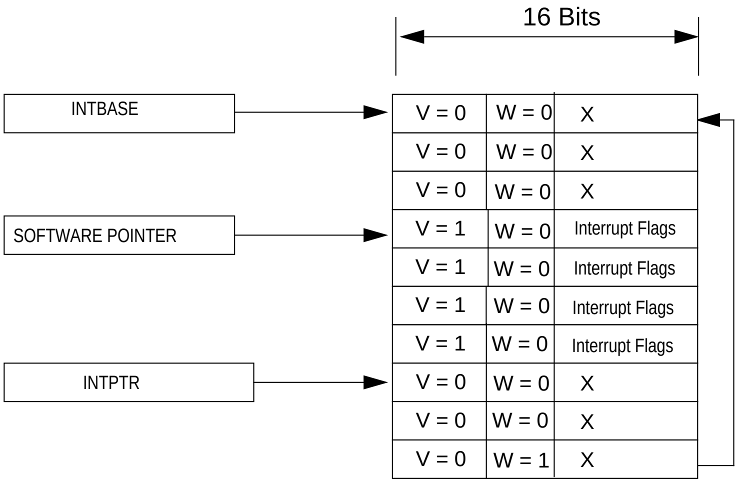
Offset	Name	Width	Description
00	**TBASE**	16	Tx buffer descriptor base address—Defines the offset of the channel's transmit BD table relative to MCBASE, host-initialized. See Figure 2-2.
02	**CHAMR**	16	Channel mode register. See Section 2.4.2.1, "CHAMR—Channel Mode Register (Transparent Mode)."
04	**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 2.4.2.2, "TSTATE—Tx Internal State (Transparent Mode)."
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)—Contains the 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 machine state (host-initialized to 0x0000_0100)—Contains the previous state of the zero-insertion state machine.
18	RES	32	
1C	**INTMSK**	16	Channel's interrupt mask flags. See Figure 2-9.
1E	BDFlags	16	Temp
20	**RBASE**	16	Receive buffer descriptor base offset—Defines the offset of the channel's 64-Kbyte receive BD table relative to MCBASE. Host-initialized. See also Figure 2-2.

Table 2-11. CHAMR Bit Settings (Transparent Mode)

Field	Name	Description
0	**MODE**	Mode—Each channel has a programmable option whether to use transparent mode or HDLC mode. 0 Transparent mode 1 HDLC mode
1	**RD**	Reverse data 0 The bit order will not be reversed, transmitting/receiving the LSB of each octet first. 1 The bit order as seen on the channels is reversed, transmitting/receiving the MSB of each octet first.
2	—	1
3	**ENT**	Enable transmit 0 Disable transmitter. If this bit is cleared and the channel's transmitter is routed to a certain time slot (within TSATTx, see Figure 2-3) the transmitter sends 1's on this time slot. 1 The transmit portion of the channel is enabled and data is sent according to protocol and to other control settings.
4	—	Reserved
5	**SYNC**	Synchronization—Controls synchronization of multichannel operation in transparent mode. 0 The first byte is put in the first available time slot or is read from the first available time slot to this logical channel. 1 The synchronization algorithm according to TRANSYNC is done.
6	RES	Reserved
7	**POL**	Enable polling—Enables the transmitter to poll the transmit BDs. 0 The CPM will not check the ready (R) bit in the transmit buffer descriptor. 1 The CPM will go check the ready (R) bit in the transmit buffer descriptor. The user can use this bit to prevent unnecessary external bus cycles when checking the ready bit (R) in the buffer descriptor. Software should always set POL at the beginning of a transmit sequence of one or more frames. The RISC processor clears POL (0) when no more buffers are ready in the transmit queue when it finds a buffer descriptor with the R bit cleared (0), that is, at the end of a frame or at the end of a multiframe transmission. To prevent deadlock, software should prepare the new BD, or multiple BDs, and set (1) the ready (R) bit in the BD before setting (1) POL. Note that the CPM automatically clears this bit; the user should never try to clear this bit in software.
8–9	—	0
10–11	—	Reserved
12–15	—	0

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.



V = 0	W = 0	X
V = 0	W = 0	X
V = 0	W = 0	X
V = 1	W = 0	Interrupt Flags
V = 1	W = 0	Interrupt Flags
V = 1	W = 0	Interrupt Flags
V = 1	W = 0	Interrupt Flags
V = 0	W = 0	X
V = 0	W = 0	X
V = 0	W = 1	X

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.

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

Field	Name	Description
11	NO	Rx non-octet-aligned frame (HDLC mode only)—A frame that contained a number of bits not exactly divisible by eight was received. NO = 1 for any type of nonalignment regardless of frame length. The shortest frame that can be detected is of type Flag-Bit-Flag. This causes the buffer to be closed with the NO error indicated. Figure 5-2 shows how the non-octet alignment is reported and where data can be found.
12	AB	Rx abort sequence—A minimum of seven consecutive ones was received during frame reception. Abort is not detected between frames. The sequence ...closing-flag, data, CRC, flag, AB, flag, data, opening-flag... does not cause an abort error. If the abort is long enough to be an idle, an idle line interrupt may be generated. An abort within the frame is not reported by a unique interrupt but rather with an RXF interrupt; the user has to examine the buffer descriptor.
13	CR	Rx CRC error—This frame contains a CRC error. The received CRC bytes are always written to the receive buffer.
14–15	—	—
16–31	DL	Data length—the number of octets written by the CPM into this buffer descriptor's data buffer. It is written by the CPM once when the buffer descriptor is closed. When this buffer descriptor is the last buffer descriptor of a frame (L = 1), the data length equals the total number of octets in the frame (including the two- or four-byte CRC). Note: The amount of memory allocated for this buffer should be greater than or equal to the contents of the maximum receive buffer length register (MRBLR + 4).
32–63	**RxBP**	Rx buffer pointer—The receive buffer pointer, which always points to the first location of the associated data buffer, may reside in either internal or external memory. The Rx buffer pointer must be divisible by 4.

Figure 5-2 shows how non-octet alignment is reported and how data is stored. The two diagrams on the left show the reception of a single-buffer, 12-byte frame including the CRC. In the top case, the reception is correctly octet-aligned and the frame length indicates 12 bytes.

5.3.3 MPC860MH Internal Memory Structure

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

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

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

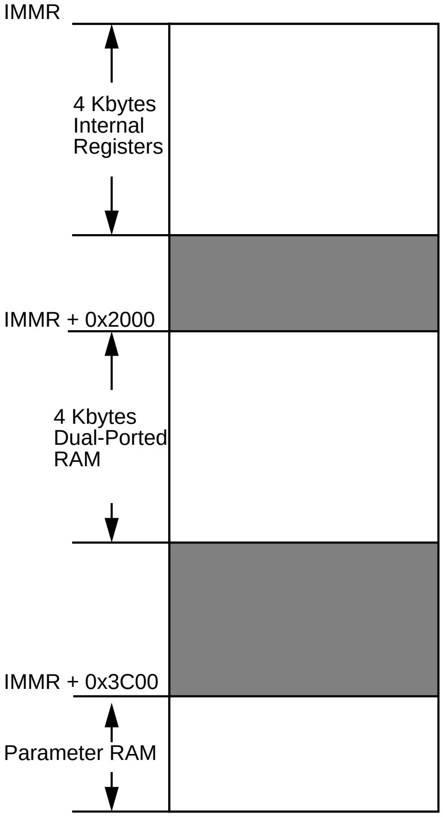
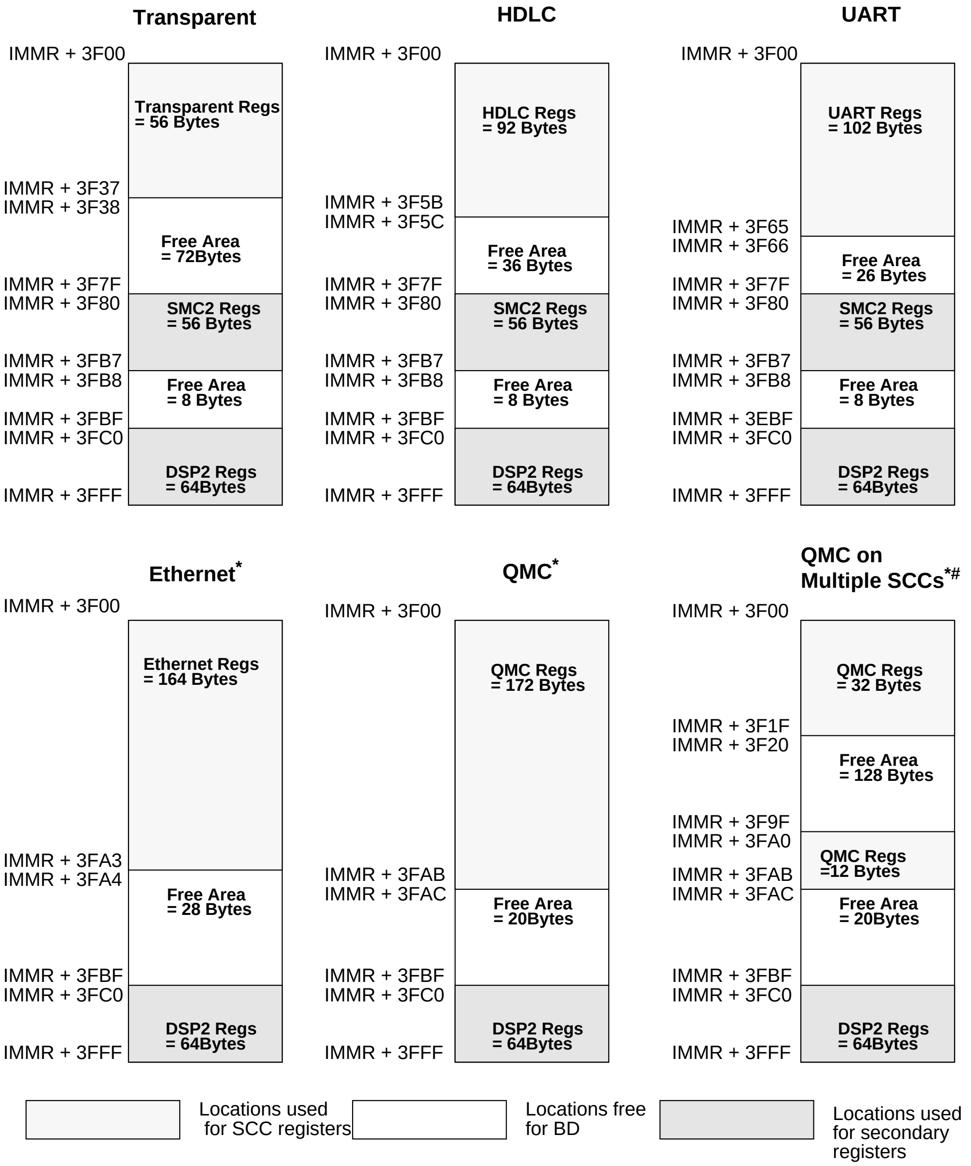


Figure 5-11. MPC860MH Internal Memory



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

Figure 5-15. MPC860MH SCC4 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.

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

Step 13. Initialize the time slot assignment tables, TSATTx and TSATRx. Each valid entry should have the V bit set. Clear the W bit in all entries except the last entry in the table. The 'mask' bits determine which bits of the time slot are processed by the CPM–normally set to 0xFF to process all 8 bits. The 6-bit CP field holds the most-significant bits of the starting address of the channel-specific parameter area. For the MH360, the most-significant bit must be zero. The 6 least-significant bits are always cleared. See Section 2.1.3, "TSATRx/ TSATTx Pointers and Time Slot Assignment Table," for more information. The following is example pseudocode for TSA table programming:

```
for (x = 0; x < time slots; x++)
{
      SCC1.TSATR[x].W = 0;          /* not last time slot */
      SCC1.TSATR[x].CP = x;         /* mark channel number */
      SCC1.TSATR[x].mask0_1 = 3;    /* no subchanneling */
      SCC1.TSATR[x].mask2_7 = 0x3F; /* no subchanneling */
      SCC1.TSATR[x].V = 1;          /* mark time slot valid */
}
      SCC1.TSATR[last].W = 1;       /* last time slot wrap */
```

Step 14. Initialize TSAT pointers (Tx_S_PTR and Rx_S_PTR), and the current time slot entry pointers, (RxPTR and TxPTR). Initialize both Tx_S_PTR and TxPTR to the first entry of the TSATx. Also initialize both Rx_S_PTR and RxPTR to the first entry of the TSARx. For common Rx and Tx time slot assignment tables, they all should point to SCC base + 20; however, they may be located anywhere within the dual-ported RAM. See Section 2.1.3, "TSATRx/TSATTx Pointers and Time Slot Assignment Table," for more information. The following is an example configuration:

```
      SCC1.Tx_S_PTR = SCC1.MCBASE+0x20;/* init pointer to TSATTx table */
      SCC1.TxPTR = SCC1.Tx_S_PTR;
      SCC1.Rx_S_PTR = SCC1.MCBASE+0x20;/* init pointer to TSATRx table */
      SCC1.RxPTR = SCC1.Rx_S_PTR;
```

Step 15. Initialize multichannel controller state QMC-STATE to 0x8000.

```
      pdpr->SCC1.QMC_STATE = 0x8000;
```

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 */
```

In the worst-case scenario, all channels open and close a buffer during the same TDM frame resulting in the peak load all performance calculations are based on. This peak load is far from the norm and can be controlled by the transmitter spreading the starting point of transmit buffers over several TDM frames.

In multimaster systems, bus latency may affect the performance of the device. The maximum external bus latency figures shown in Table 8-4 are measured from the assertion of the BR (bus request) to the assertion of the BGACK (bus grant acknowledge); that is, from start of bus request output being active until the cycle is completed. For multimaster systems, bus arbitration overhead is included. Latencies of up to 40 clocks were simulated; for values over 40, the acceptable latency may be larger.

Table 8-4 shows average maximum acceptable bus latencies, meaning the device can tolerate longer bus delays if they are infrequent. For lengthy delays, a larger FIFO can pick up the slack, continuing emptying or filling depending on the data flow direction. Therefore, the larger the FIFO the more tolerant the system is to infrequent peaks in bus delays. However, the average acceptable bus latency still depends on the overall data rate and frame length and not on the FIFO size.

Table 8-4. Simulated Latencies

Maximum Acceptable Latency (Bus Cycles)		Channel Combinations
25 MHz	33 MHZ	
Not supported	12	SCC1: Ethernet; SCC2: 16 x 64 Kbps; SCC3: 16 x 64 Kbps
Not supported	11	SCC1: Ethernet; SCC2: 16 x 64 Kbps; SCC3: 16 x 64 Kbps; SCC4: 64 Kbps HDLC
9 clocks	>40	SCC1: 32 x 64 Kbps. Serial bit rate 2.048 Mbps (E1/CEPT)
8 clocks	35	SCC1: 32 x 64 Kbps; SCC2: 64 Kbps; SCC3: 64 Kbps; SCC4: 64 Kbps; all HDLC
40 clocks	>40	QMC with 24 channels. Serial bit rate 1.544 Mbps (T1)
33 clocks	>40	SCC1: 24 x 64 Kbps; SCC2: 64 Kbps; SCC3: 64 Kbps; SCC4: 64 Kbps; all HDLC
8 clocks	24	SCC1: Ethernet; SCC2: 12 x 64 Kbps; SCC3: 12 x 64 Kbps. TDM bit rate = 1.544 Mbps
Not supported	23	SCC1: Ethernet; SCC2: 12 x 64 Kbps; SCC3: 12 x 64 Kbps; SCC4: 64-Kbps HDLC. TDM bit rate = 1.544 Mbps
40 clocks	>40	SCC1: 16 x 128 Kbps. TDM bit rate = 2.048 Mbps

9.4 MSC Subchanneling Example

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

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

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

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

INDEX