Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

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

Applications of Embedded - Microprocessors

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

Details	
Product Status	Obsolete
Core Processor	CPU32+
Number of Cores/Bus Width	1 Core, 32-Bit
Speed	33MHz
Co-Processors/DSP	Communications; CPM
RAM Controllers	DRAM
Graphics Acceleration	No
Display & Interface Controllers	-
Ethernet	10Mbps (1)
SATA	-
USB	-
Voltage - I/O	5.0V
Operating Temperature	0°C ~ 70°C (TA)
Security Features	-
Package / Case	357-BBGA
Supplier Device Package	357-PBGA (25x25)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68mh360zq33lr2

works transparently, not participating in any QMC protocol functions. The SCC only performs the parallel-to-serial conversion and adds elasticity through its FIFO memory. The CPM, with its special enhanced microcode and additional dedicated hardware for framing and masking support, does all of the protocol processing for each of the 64 channels. Note that it is executed without intervention from the on-board CPU. Figure 1-1 illustrates the QMC's multichannel capability. Note that each SCC can support up to 64 channels from the TDM; however, there are limitations depending on the device used. This is summarized in Section 1.3, "QMC Features."

Each SCC can work in QMC mode, either alone or together in any combination. The larger FIFO of SCC1 yields the best performance and is therefore recommended for QMC operation. One TDM connection can be routed to one or more SCCs operating in QMC mode, with each SCC operating on different time slots. It is possible to use both TDMs for QMC with combined routing to one SCC or to separate SCCs. When using two TDMs connected to one SCC, restrictions such as using common clocks and sync inputs apply; it is also important to avoid collisions by separating the serial interface (SI) routing.

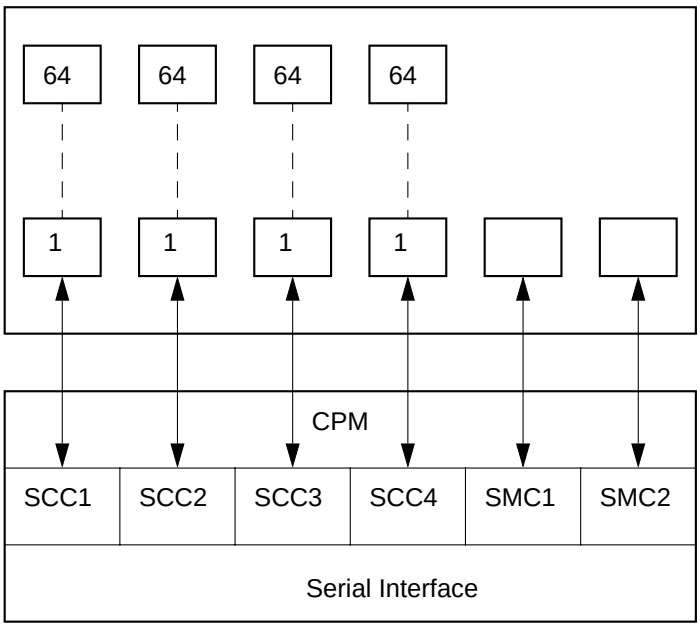


Figure 1-1. QMC Channel Addressing Capability

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

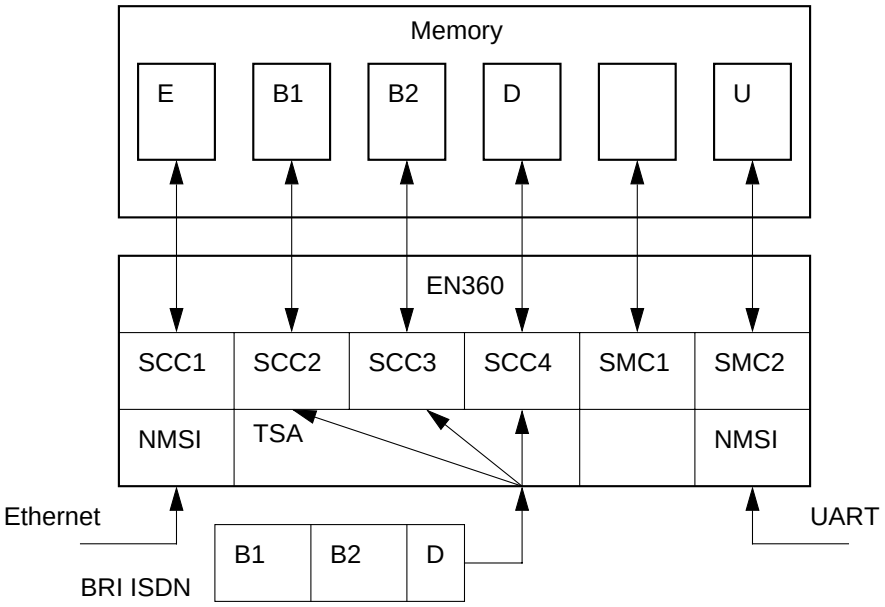


Figure 1-3. Internal Routing for Ethernet-to-BRI Bridge Using MC68EN360

The following example shows how an MC68MH360 can implement the BRI using only one SCC, leaving SCC3 and SCC4 available to run other protocols such as frame relay over HDLC and another Ethernet link, on SCC1, to the LAN. The QMC protocol allows all three channels B1, B2, and D to be routed to SCC2 using the TSA. The first byte (B1) is routed to logical channel 1, the second byte (B2) to logical channel 2, and the third byte to logical channel 3, of which only the first 2 bits represent the D channel as illustrated in Figure 1-4 and Figure 1-5. This routing is defined in the QMC time slot assignment tables. The first advantage of the QMC protocol is that it releases SCCs to run other protocols. The second advantage is highlighted in the next example.

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

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

2.4.1.3 INTMSK—Interrupt Mask (HDLC)

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

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

This register is initialized by the host prior to operation.

Interrupt Table Entry:

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

INTMSK:

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

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

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

Chapter 4
QMC Exceptions

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

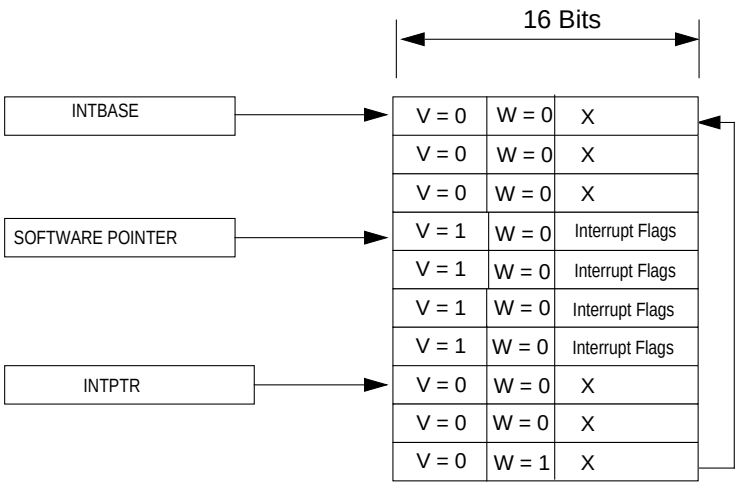


Figure 4-1. Circular Interrupt Table in External Memory

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

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

5.3 Placement of Buffer Descriptors

The internal dual-ported RAM is used to store the buffer descriptors for all non-QMC operation. This solution causes minimum loading of the external bus. When starting any operation or switching between buffers during operations, several accesses must be made by the CPM to find the actual data buffers and to read and write control and status information. This process is unseen by the user for internal accesses, and any external bus master or memory refresh control can occur uninterrupted.

5.3.1 MC68MH360 Internal Memory Structure

To support 32 channels on the MC68MH360, the entire 2-Kbyte dual-ported RAM is needed for channel-specific parameters. Each logical channel occupies 64 bytes; thus 32 channels require 2 Kbytes. No conflicts arise for QMC operation since it uses external memory for the buffer descriptors; however, buffer descriptors for other protocols must be in internal memory.

If the QMC uses all 32 channels, no space is left in the lower 2-Kbyte area; the only free areas are in the RAM pages, each 192-bytes large. Depending on the functions, channels and protocols used, some areas remain free for buffer descriptors. If a function is not enabled, its parameter RAM area may be used.

If fewer than 32 logical channels are used or if physical channels are concatenated to super channels, space is freed in the dual-ported RAM. Each logical channel creates a 64-byte hole in the dual-ported RAM that an SCC can use for buffer descriptors. QMC channels can use this area rather than external memory for buffer descriptors, reducing the load on the external bus.

If the external 64-Kbyte area overlaps the internal dual-ported RAM, external and internal buffer descriptors can be combined for the QMC. This is controlled by the show cycles setting. If split buses are used with SHEN1/SHEN0 = 00, a memory access is always made to the dual-ported RAM if the source is an internal master. For more information, refer to the description of the module configuration register in the *MC68360 Quad Integrated Communications Controller User's Manual*.

Figure 5-5 shows the internal memory map. Figure 5-6 shows the SCC parameter RAM overlap example. Figure 5-7 through Figure 5-10 give a detailed memory map for each SCC, showing parameter RAM usage for different functions.

- RAM page one is dedicated to SCC1 and for miscellaneous storage.
- RAM page two is dedicated for SCC2, RISC timers, and the SPI channel.
- RAM page three is dedicated for SCC3, SMC1, and IDMA1 operations.
- RAM page four is dedicated for SCC4, SMC2, and IDMA2 operations.

If the QMC operates with a full 64 channels, no space is left in the lower 4-Kbyte area. In this case, the only free areas are in the RAM pages, each 256-bytes large. Depending on the functions, channels and protocols used, some areas remain free for buffer descriptors. Also, if a particular function is not enabled, its parameter RAM area may also be used.

If fewer than 64 logical channels are used or if physical channels are concatenated to super channels, space is freed in the dual-ported RAM. Each unused logical channel creates a 64-byte hole in the dual-ported RAM. This area is free for buffer descriptors for any SCC. QMC channels can also use this space instead of external memory for buffer descriptors, reducing the load on the external bus.

Figure 5-11 shows the internal memory map for the MPC860MH. Figure 5-12 to Figure 5-15 show a more detailed memory map for each SCC, showing the parameter RAM usage for different functions.

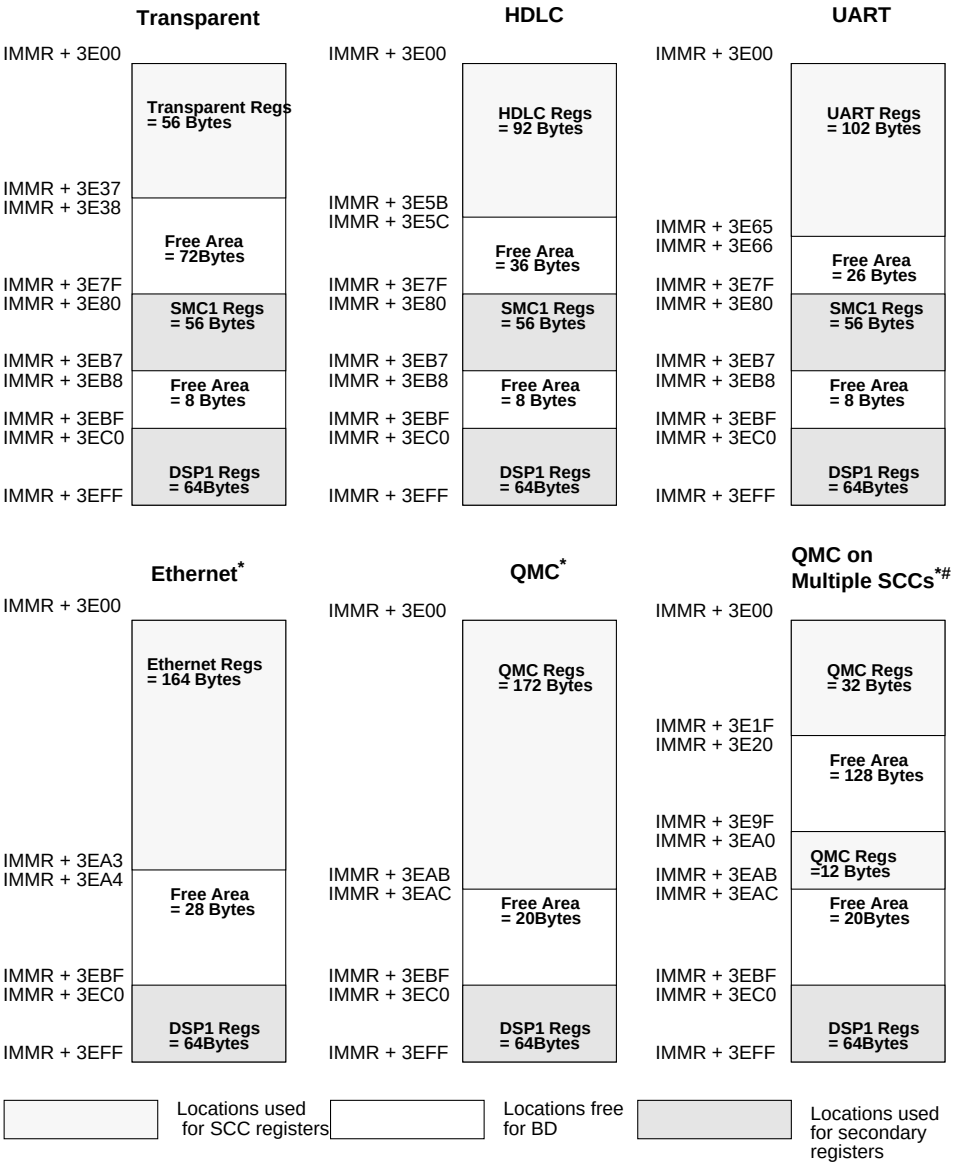
- RAM page one is dedicated to SCC1, I^2C, IDMA1 and for miscellaneous storage.
- RAM page two is dedicated for SCC2, SPI, IDMA2 and RISC timers
- RAM page three is dedicated for SCC3, SMC1 and DSP1 operations.
- RAM page four is dedicated for SCC4, SMC2 and DSP2 operations.

Table 5-4 shows the functions available for various protocols on each SCC for the MPC860MH.

Table 5-4. MPC860MH Functions Available

	Function Available?	Transparent	HDLC	UART	Ethernet	QMC	Shared QMC
SCC1	Yes	Misc, I^2C, IDMA1	Misc, I^2C, IDMA1	Misc, I^2C, IDMA1	Misc, IDMA1	Misc, IDMA1	Misc, IDMA1
	No				I^2C	I^2C	I^2C
SCC2	Yes	SPI, Timer, IDMA2	SPI, Timer, IDMA2	SPI, Timer, IDMA2	Timer, IDMA2	Timer, IDMA2	Timer, IDMA2
	No				SPI	SPI	SPI
SCC3	Yes	SMC1, DSP1	SMC1, DSP1	SMC1, DSP1	DSP1	DSP1	DSP1
	No				SMC1	SMC1	SMC1
SCC4	Yes	SMC2, DSP2	SMC2, DSP2	SMC2, DSP2	DSP2	DSP2	DSP2
	No				SMC2	SMC2	SMC2

Figure 5-12 to Figure 5-15 show that not all the functions available on each SCC can be used simultaneously due to overlaps of the register locations stored in the parameter RAM.



#—TSATRx & TSATTx are mapped to an alternative SCC's parameter RAM.
***—SMC1 is not available in these configurations due to memory conflict.**

Figure 5-14. MPC860MH SCC3 Parameter RAM Usage

Since the SIMODE register defaults to 0x0000_0000, a typical application may set these bits as follows:

```
SIMODE.CRTa = 1;            /* common syncs & clocks */
SIMODE.RFSDa = 1;           /* receive frame sync = 1 clock delay */
SIMODE.TFSDa = 1;           /* transmit frame sync = 1 clock delay */
```

Step 2: Initialize the SICR (SI clock route) register. The SICR register is defined on page 7-86 of the MC68360 User's Manual and page 16-121 of the MPC860 User's Manual. Table 6-2 shows the SICR bit settings.

Table 6-2. SICR Bit Settings

Name	No. of Bits	Description	Setting
GRx	1	Support SCCx grant mode	0
SCx	1	Connect SCCx to TDM or NMSI	1
RxCSx	3	Connect SCCx receive to a clock	X - as SCx = 1
TxCSx	3	Connect SCCx transmit to a clock	X - as SCx = 1

As the SICR register defaults to 0x0000_0000, a typical application may set these bits as follows:

```
SICR.SC1 = 1;               /* SCC1 is TDM */
```

Step 3: Configure port A for TDMa and/or TDMb signals, L1TXDx, L1RXDx, L1TCLKx, and L1RCLKx. For more information on port A, see page 7-358 of the MC68360 User's Manual and page 16-455 of the MPC860 User's Manual.

The following setting enables TDMa and TDMb, and selects both L1TCLKx and L1RCLKx pins. Note that only L1RCLKx is required if common clocking is selected by the CRTx bit in the SIMODE register.

```
PAPAR = 0xA5F0;             /* init port A pin assignment register */
PADIR = 0x00F0;             /* init port A data direction register */
```

Step 4: Configure port B for TDMa and/or TDMb signals, L1CLKOx and L1ST1, 2, 3, and/or 4. For more information on port B, see page 7-363 of the MC68360 User's Manual and page 16-460 of the MPC860 User's Manual.

The following setting enables both L1CLKOx and all L1STx strobes. Note that the L1STx functions are repeated on port C and should only be configured on one port.

```
PBPAR = 0xFC00;             /* init port B pin assignment register */
PBDIR = 0x0C00;             /* init port B data direction register */
```

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 */
```

Chapter 7
Features Deleted in MC68MH360

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

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.

A.5 SCC Event Register

Figure A-12 shows the 68360 bit numbering for the SCC event register.

7	6	5	4	3	2	1	0
				IQOV	GINT	GUN	GOV

Figure A-12. SCC Event (SCCE) Register

A.6 SCCM Register

Figure A-13 shows the 68360 bit numbering for the SCCM register.

7	6	5	4	3	2	1	0
				IQOV	GINT	GUN	GOV

Figure A-13. SCCM Register

A.7 Receive and Transmit Buffer Descriptors

Figure A-14 and Figure A-15 show the 68360 bit numbering for the receive and transmit buffer descriptors.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET + 0	**E**	—	**W**	**I**	L	F	**CM**	—	UB	—	LG	NO	AB	CR	—	—
OFFSET + 2	DATA LENGTH															
OFFSET + 4	RX DATA BUFFER POINTER															
OFFSET + 6																

Figure A-14. Receive Buffer Descriptor (RxBD)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET + 0	**R**	—	**W**	**I**	**L**	**TC**	**CM**	**—**	**UB**	—	—	—	**PAD**			
OFFSET + 2	DATA LENGTH															
OFFSET + 4	TX DATA BUFFER POINTER															
OFFSET + 6																

Figure A-15. Transmit Buffer Descriptor (TxBD)

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.

Using on-chip time slot assigners, the S/T and U interfaces in IDL2 mode can match the QMC bus structure—both interfaces can be connected to a 2.048-MHz IDL2 bus and route the B1 channel, B2 channel and D channel to any time slot.

Figure C-1 shows the IDL2 bus configured to match the QMC protocol.

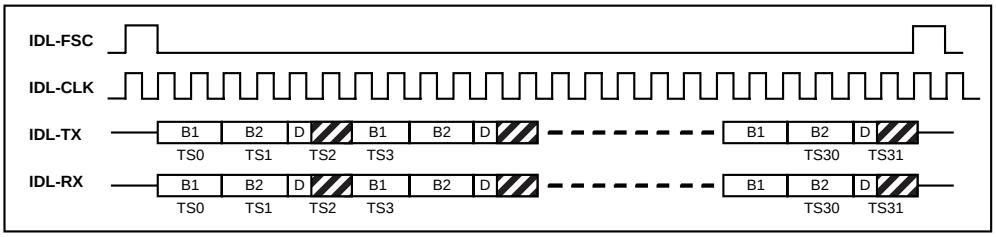


Figure C-1. IDL2 Bus Structure for a Connection to the QMC Bus

C.2 Control and Status Information

Using the SPI port, the QUICC32 and the ISDN interfaces exchange control and status information via out-of-band signaling. Optionally, the MC145572s could use an 8-bit parallel port for control and status transfer, allowing the U interfaces to be connected to the processor bus.

Figure C-2 shows the connection between the QUICC32 and an S/T interface.

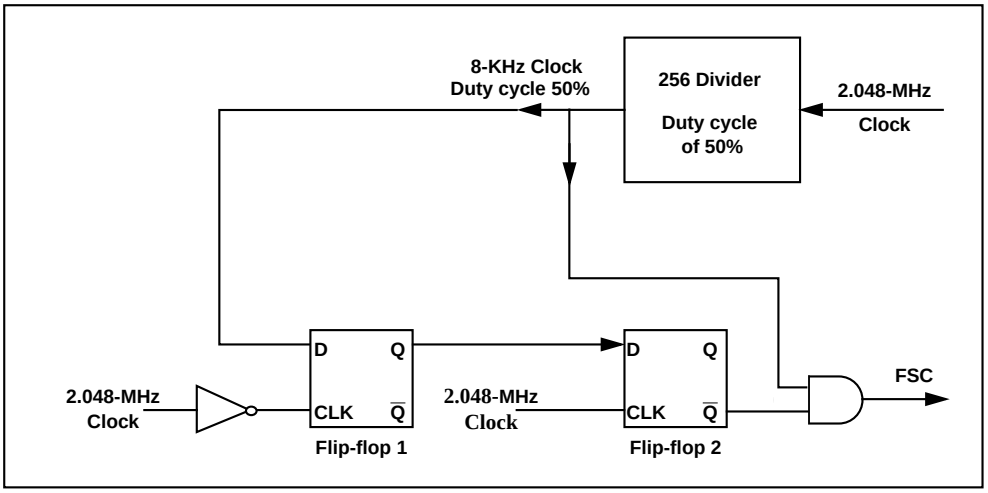


Figure C-4. FSC Generation from a 2.048-MHz Clock—Block Diagram

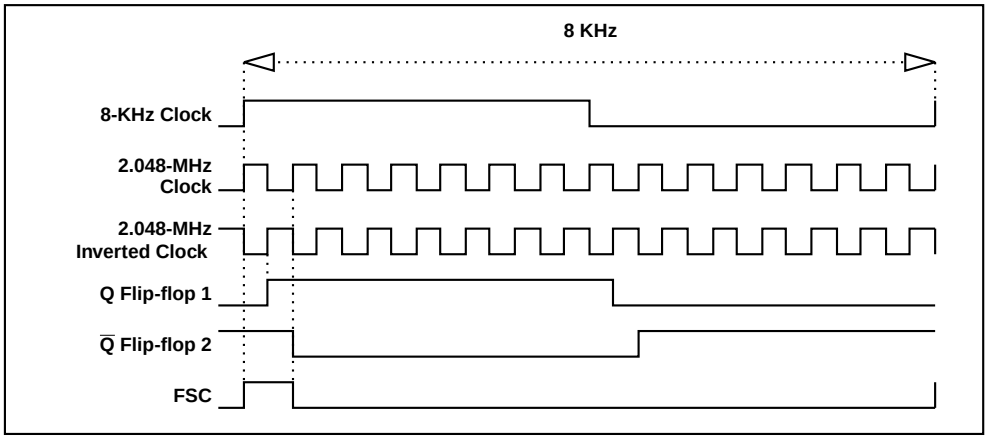


Figure C-5. FSC Generation from a 2.048-MHz Clock—Timing

The S/T interface includes elastic buffers allowing continued operation under any phase relationship between the IDL frame sync and the network. These buffers allow the frame sync to wander with respect to the network up to 60 µs peak-to-peak, exceeding the Q.502 requirements of 18 µs peak-to-peak over a 24-hour period.

Figure C-6 shows a block diagram of the connection between four S/T interfaces and the QUICC32. The diagram would be the same for up to 10 S/T interfaces.

C.3.3.2 U-Interface Configuration

Do the following for U-interface configuration:

- IDL2 with time slot assigner (TSA enabled in reg. OR6[5–7]; TSA selection in reg. OR0 to OR5)
- Slave mode (DCL & FSC are input) - (pin M/$\overline{S}$ to GND)
- FREQREF enabled at 2.048 MHz (reg. OR8[4] = 1)

C.3.3.3 QUICC32 Configuration

Do the following for QUICC32 configuration:

- SCC3 using the QMC protocol for handling the different channels of the multiplexed IDL2 bus. (D channels are HDLC encoded/decoded and B-channels can be configured for transparent or HDLC framing)
- SCC1 can be configured for Ethernet, HDLC, transparent, or UART.
- SCC2 and SCC4 can be configured for HDLC, transparent, or UART.
- The SPI is connected to the SCP port of each S/T or U interface for handling configuration and control information.
- For the U interface, the SPI/SCP connection can be replaced by a connection of the 8-bit parallel port of the U transceiver to the processor bus of the QUICC.
- One I/O signal can be dedicated for handling the $\overline{SCPEN}$ signal of each S/T or U interface.
- One interrupt signal can be dedicated for handling the $\overline{IRQ}$ signal of each S/T or U interface.

INDEX