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/mc68mh360zq25l

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

Additional Reading

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

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

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

Conventions

This document uses the following notational conventions:

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

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

2.4 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 2-5. CHAMR Field Descriptions (HDLC) (Continued)

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

2.4.1.2 TSTATE—Tx Internal State (HDLC)

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

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

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

Figure 2-8. TSTATE—Tx Internal State (HDLC)

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

Table 2-6. TSTATE Field Descriptions for MH360 (HDLC)

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

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

2.4.2.3 INTMSK—Interrupt Mask (Transparent Mode)

Each event defined in the interrupt circular queue entry maps directly to a bit in INTMSK as shown in Figure 2-13. There is one mask bit for each event—UN (bit 11), 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.

- 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 before operation.

INTERRUPT TABLE ENTRY:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
V	W	RES	RES	–	CHANNEL NUMBER					RES	UN	RES	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		RESERVED		RESERVED						RES	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 2-13. INTMSK and Interrupt Table Entry (Transparent Mode)

2.4.2.4 TRNSYNC—Transparent Synchronization

In transparent mode, the TRNSYNC register controls the synchronization for single time slots or superchannel applications.

Note

This register has no meaning if the SYNC bit in the channel mode register (CHAMR) is cleared (0).

When sending a transparent message over several time slots, it is necessary to know in which time slot the first byte of data appears.

The TRNSYNC word-length register is divided into two parts with the high byte controlling the first received time slot and the low byte controlling the transmitter synchronization.

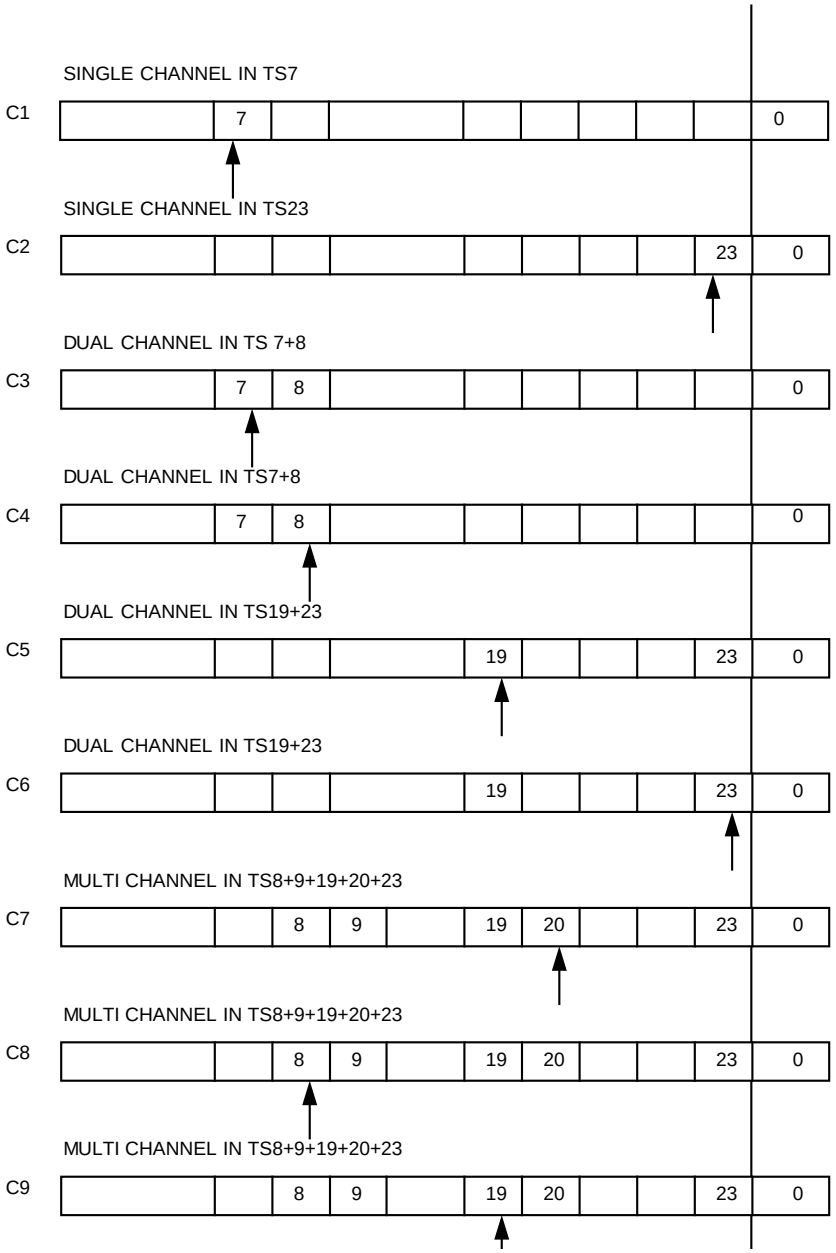

Figure 2-14. Examples of Different T1 Time Slot Allocation

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.

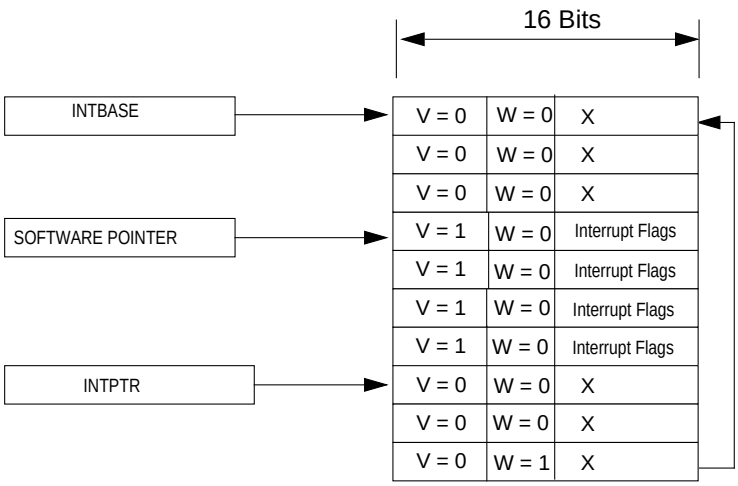

Figure 4-1. Circular Interrupt Table in External Memory

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.

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.

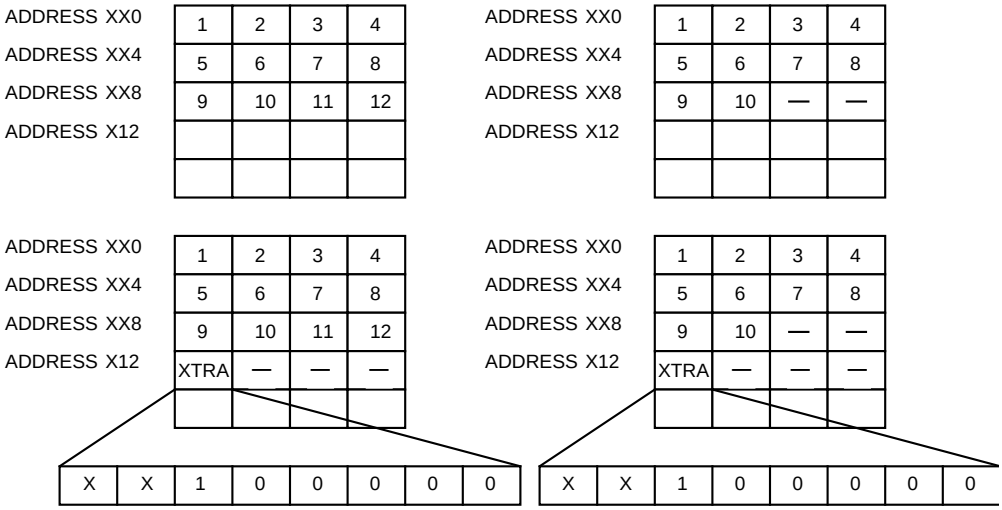

Figure 5-2. Nonoctet Alignment Data

In the bottom case, two more bits are received. The frame length is now 13 bytes, and the address positions X13 through X15 point to invalid data. Address position X12 contains information about the non-octet alignment. Valid information is written starting at the MSB position, shown as ‘x’ in the diagram. Starting from the LSB position, zeros are filled in followed by a ‘1’ immediately preceding the valid data.

The two diagrams on the right show how the data and the extra information is stored for a frame length that is not a multiple of 4 bytes. The additional information is always on a long-word boundary. In the top case the frame length is 10 bytes and in the bottom case the length is 11 bytes.

For a minimum frame consisting of ‘flag, 1 bit, flag’ the frame length is 1. The only valid entry is at address XX0 with content of x1000000.

To accommodate the extra long word that may be written at the end of a frame, it is recommended to reserve MRBLR + 4 bytes for each buffer area.

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.

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

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

A typical setting would be:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
        {};
        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 */
        }
}
```

9.2 MSC Microcode Operation

In normal operation (without the MSC microcode), the QMC protocol allows specific bits in an 8-bit time slot to be masked to create a single subchannel per SCC. A problem arises when multiple subchannels are multiplexed within a single time slot as in GSM (global system for mobile communications) where four 16-Kbps subchannels are multiplexed into a single 64-Kbps channel over a 2.048-Mbps A bis link. A brute-force solution routes the separate subchannels to different SCCs, consuming all four SCCs for the single TDM link as shown in Figure 9-1. Each SCC filters out one of the four 2-bit subchannels in time slot 2 (TS2) using a unique mask located in its time slot assignment tables (TSATRx/TSATTx). With the MSC microcode, subchannels can be regenerated using only one SCC.

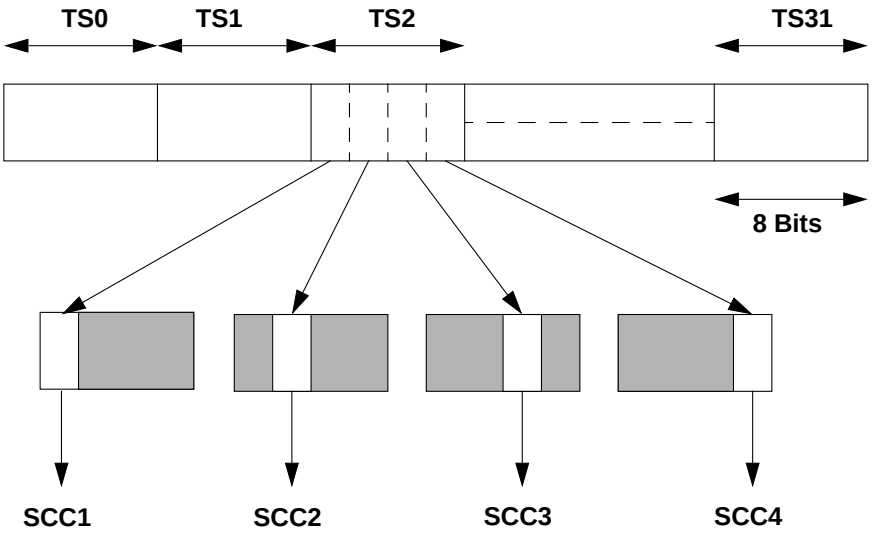

Figure 9-1. Two-Bit Subchannel Implementation without MSC Microcode

The MSC microcode enables an 8-bit time slot to be split into multiple, bit-resolution subchannels. The microcode applies user-defined masks in a time slot assignment table entry to subdivide a given channel. Bit 11 of a table entry is now called the L bit to mark the last subchannel of a given time slot. Figure 9-2 shows the MSC microcode solution to the above GSM problem. Again in this example, time slot 2 contains four 2-bit channels, but now the full time slot can be routed to a single SCC and split into subchannels within the time slot assignment tables.

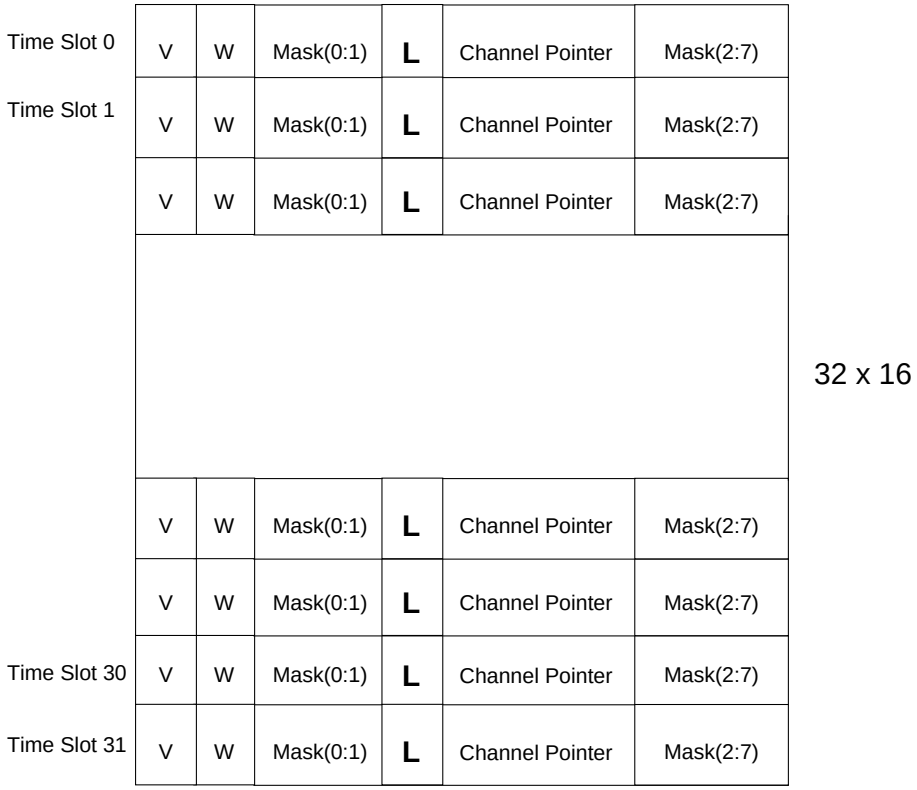

Figure 9-3. Time Slot Assignment Table Showing MSC Configuration

Table 9-1 describes the fields in the time slot assignment table for receive (TSATRx) when the MSC microcode is enabled.

Table 9-1. Time Slot Assignment Table Entry Fields for Receive (MSC)

Field	Description
V	Valid bit—The valid bit indicates whether this time slot is valid. 0 The data in this 8-bit time slot is totally ignored and not written to any buffer. 1 The data in this 8-bit time slot is valid and written to the current buffer, pointed to by the channel pointer entry, after protocol processing (for example, zero deletion in HDLC).
W	Wrap bit—Identifies the last entry in TSATTx. 0 This is not the last time slot in the frame. 1 The RISC processor wraps around and handles time slot 0 or the first 8 bits transferred from the TSA in the next request. The next request is identified by a frame synchronization pulse.

Figure A-2 shows the 68360 bit numbering for a time slot assignment table for 64-channel common Rx and Tx mapping.

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

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.

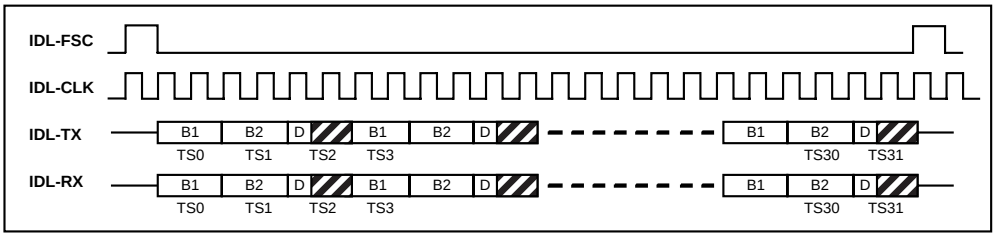

Figure C-1. IDL2 Bus Structure for a Connection to the QMC Bus

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.

C.3.1.1 Activation Procedure

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

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

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

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

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

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

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

INDEX