Application Note **MSAN-123**

The MT8980 and the MT8981 Digital Crosspoint Switches

ISSUE 1 June 1986

Table of Contents

- **1.0 Introduction**
- **2.0 The ST-BUS**
- **3.0 Architecture of MT8980**
  - 3.1 ST-BUS Interface to the MT8980/81
  - 3.2 Microprocessor interface
  - 3.3 Internal structure
  - 3.4 Programming Examples
- **4.0 Delay through the MT8980**
  - 4.1 Switching Mode
  - 4.2 Message Mode
- **5.0 Microprocessor Accesses**
- **6.0 External Control Using CSTo**
- **7.0 High Impedance Capability**
  - 7.1 Initialization of the MT8980

1.0 Introduction

The MT8980 and the MT8981 are digital crosspoint switches. Any information in an ST-BUS channel (digitized audio, digitized video or data), can be input to the MT8980/81 and then switched to anyone of the its ST-BUS output channels. The MT8980/81 is easily controlled by a microprocessor. The microprocessor can read the contents of all of the input channels and can write information to and read information from all of the output channels.

These capabilities make the MT8980/81 an integral part of a very efficient system of information transport, the ST-BUS. The MT8980/81 provides the ST-BUS system with communication path flexibility. This is a prerequisite in today's systems, which

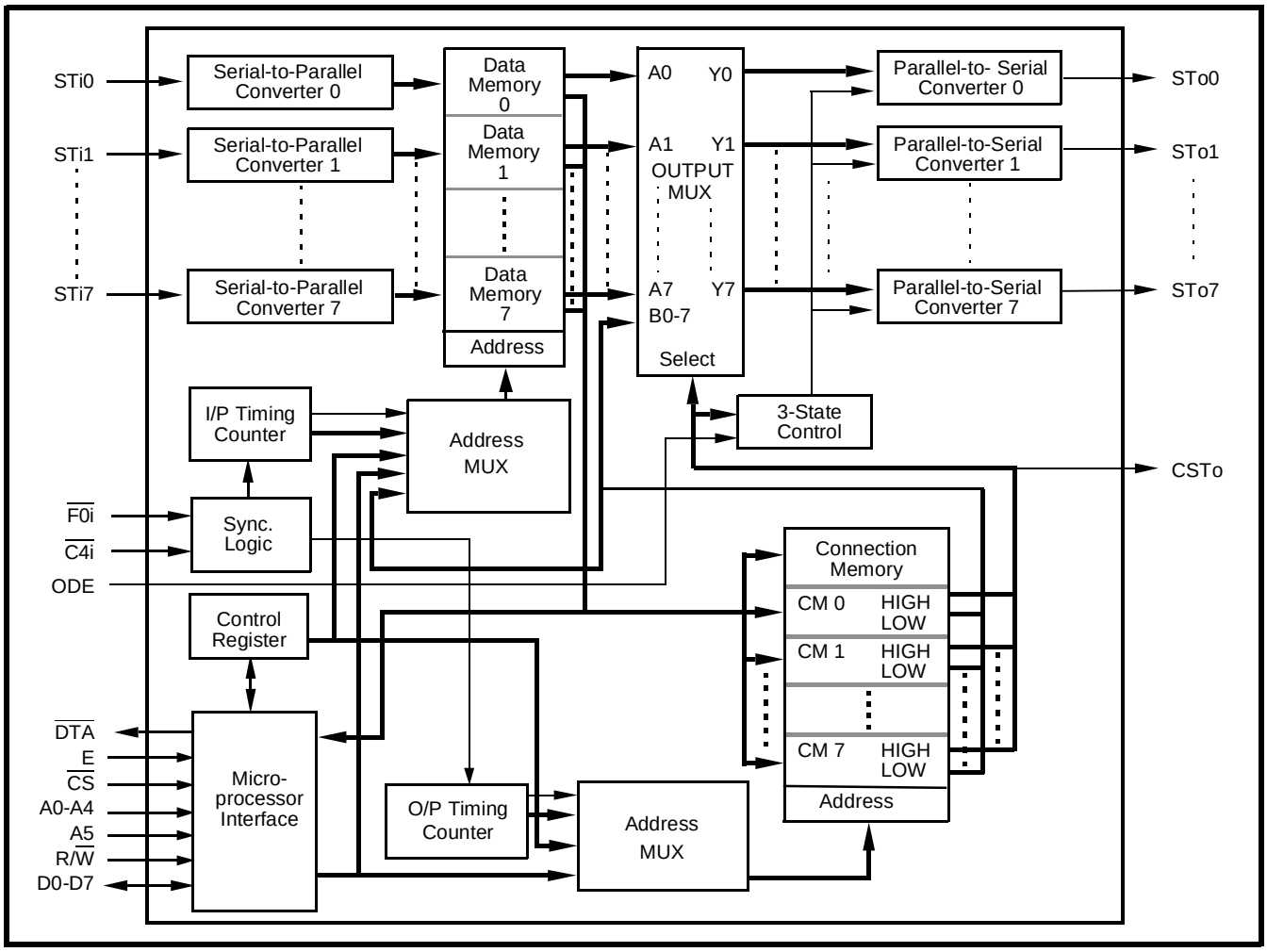

Figure 1 - Block Diagram of the MT8980

emphasize distributed architectures and the integration of voice, video and data.

The principle applications of the MT8980/81 are:

a) **Switching.** The MT8980/81 provides non-blocking switching of information channels (circuit switching). The main difference between the MT8980 and the MT8981 is that the MT8980 has a 256 channel capacity and the MT8981 has a 128 channel capacity. Each channel has a throughput capacity of 64 kbit/s. Channels may carry any form of digitized signal, from encoded voice or data to digitized video.

The MT8980/81 can provide a communication path between any pair of nodes connected to the MT8980/81 through ST-BUS streams. As well, the MT8980/81 can place its ST-BUS outputs into a high impedance state on a per channel basis, so that larger matrices may be constructed.

b) **ST-BUS Device Control.** When the MT8980/81 is in Message mode, each ST-BUS channel input or output looks like a parallel register to the controlling microprocessor. Each of these input channels or output channels can map directly into the status register, or the control register of a “remote” ST-BUS device connected to the MT8980/81. An individual MT8980 can read up to 256 status registers and write up to 256 control registers.

c) **Interprocessor Communication.** The ability to provide communication paths between microprocessors is a subset of the previous application. In this case, the remote ST-BUS device is another MT8980/81 which is controlled by another microprocessor. This microprocessor is also sending and receiving information on ST-BUS channels in Message mode. The channel that the remote microprocessor is receiving on appears to the “local” microprocessor as a control register of an ST-BUS device appeared in the previous application. The channel that the remote microprocessor is sending on appears to the local microprocessor as a status register of an ST-BUS device appeared in the previous application.

The above applications are discussed in more detail in a subsequent application note.

This application note deals with the relationship between the ST-BUS and the MT8980/81. Other MT8980/81 topics covered are: microprocessor interfacing, the two major modes of the device (Switching mode and Message mode), the delay that the MT8980/81 inserts into end-to-end communication, and miscellaneous details about MT8980/81 operation.

2.0 The ST-BUS

The ST-BUS is a component interface. It minimizes the pin requirements on devices and backplanes by using time division multiplexing to transfer a number of logical data channels through a single wire or pin.

A circuit’s ST-BUS interface consists of two control signals and as many input and output streams as are needed to provide the bandwidth required by the circuit to perform its function (usually one input stream and one output stream for information to be processed, and one input and output stream for control and status information).

The standard bit rate of the ST-BUS is 2.048 Mbit/s. The overall bit stream is divided into intervals called frames, yielding a frame rate of approximately 8000 frames/s. Each frame is divided into 256 intervals, each containing one bit of digital information. For most communications applications, the bit stream is also divided into 32 intervals per frame called channels. Each channel is allotted eight bits, for a per channel data rate of:

8000 frames /sX8 bits/frame=64 kbit/s.

Channels are referenced to the start of the frame and are numbered from zero to 31.

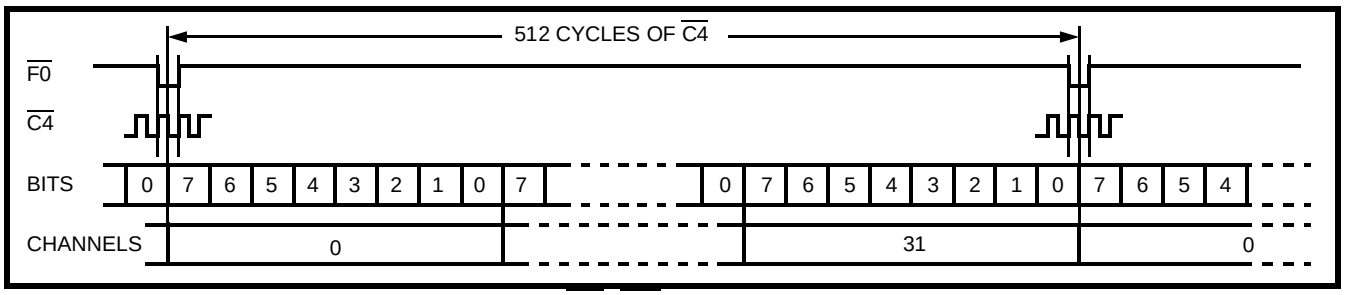

Figure 2 - F0i, C4i and ST-BUS Alignment

Information must be placed on the ST-BUS stream and removed from it synchronously. In most applications, information is placed onto the ST-BUS or received from it in one particular channel timeslot (more than one channel may be used for more bandwidth). Any device interfaced to the ST-BUS must accept a clock signal for bit timing and a framing signal for synchronization with the frame boundaries. There are several types of clock signals and frame synchronization signals (framing pulses) defined for the ST-BUS. These signals are described in the ST-BUS specification in the Mitel Semiconductor Data Book.

3.0 Architecture of MT8980

The MT8980 architecture is illustrated in Figure 1.

3.1 ST-BUS Interface to the MT8980/81

The clock input of the MT8980/81 is called $\overline{C4i}$ and its frequency (4.096 MHz) is twice the data rate. The frame pulse is the $\overline{F0i}$ signal. See Figure 2 for the timing relationship between $\overline{C4i}$, $\overline{F0i}$ and the bits and channels of the ST-BUS.

The ST-BUS outputs of the MT8980/81 derive their output information from two sources: from the ST-BUS streams input to the MT8980/81 and/or from the microprocessor controlling the MT8980/81. The MT8980/81 is also designed to allow the controlling microprocessor to read the information on the input ST-BUS channels.

The MT8980/81 has two modes of operation to accommodate the choice of information sources: Switching mode and Message mode. Switching mode allows the information contained in each output channel to be chosen from any of the input channels in a non-blocking fashion. An advantage of having the output specify the source is that more than one output, all outputs in fact, can have the same source. This is advantageous for broadcasting messages or generating resource channels (e.g., dial tone can be input in one channel but output in many). Message mode allows information to be written through the microprocessor port onto the output channels. This information will not change until rewritten. The information on the input channels can be read by the microprocessor no matter what mode the device is in.

The major difference between the MT8980 and the MT8981 is the number of 32 channel streams which each can handle. The MT8980 has eight ST-BUS inputs and eight ST-BUS outputs (256 channels in, 256 channels out). The MT8981 has four ST-BUS inputs and four ST-BUS outputs (128 channels in, 128 channels out). Essentially, the MT8980 is a 256 crosspoint digital switch, and the MT8981 is a 128 crosspoint digital switch. Also, the MT8981 does not have a CSTo output (see Section 6.0). The basic similarities between the two devices allow both to be referred to as the MT8980, unless explicit reference to the MT8981 is needed.

3.2 Microprocessor Interface

The microprocessor port consists of a data bus for information transfer, an address bus, a chip enable, two signals for synchronizing microprocessor bus transfers, and a transfer direction control signal. The data bus is eight bits wide and carries control information for the MT8980 from the microprocessor. The six address bits, A0-A5, help determine which of the individual locations within the MT8980 are accessed.

The Select signal ($\overline{CS}$) is the chip enable. If $\overline{CS}$ is high, no access to the MT8980 is possible. Normally, when DS is brought low, most systems remove the $\overline{CS}$ signal. This is not mandatory, as the MT8980 will not drive the data bus or receive information from the data bus unless DS is high. The Data Strobe signal (DS) and the Data Acknowledge signal ($\overline{DTA}$) perform transfer synchronization. At the rising edge of DS, all other control information must be valid. On the falling edge of DS, data from the microprocessor or data from the MT8980 is valid. DS is usually not brought low until after the MT8980 brings the $\overline{DTA}$ signal low, which occurs when the MT8980 is ready to accept or provide data. Once DS is low, the MT8980 brings $\overline{DTA}$ high, terminating the bus cycle.

The Read/Write signal (R/$\overline{W}$) determines the direction of information flow. When the signal is high, information can be read from the MT8980 by the microprocessor. When R/$\overline{W}$ is low, data from the microprocessor may be written to the MT8980.

3.3 Internal structure

There are four major functional blocks inside the MT8980. These blocks are the Data Memory, the Connection Memory High, the Connection Memory Low and the Control Register. Data Memory is related to the input ST-BUS streams; Connection Memory High and Connection Memory Low are related to the output ST-BUS streams. The Control Register performs block addressing functions and overall mode determination.

The Data Memory is where information is stored as it comes in from the ST-BUS inputs. The loading of information into Data Memory from the inputs is automatic (Data Memory may not be written into

from the microprocessor port). Channel information is converted from the serial ST-BUS format into a parallel format and stored in a corresponding byte in the Data Memory in an unchanging sequence. The size of the MT8980's Data Memory is 256 bytes and the size of the MT8981's Data Memory is 128 bytes (one byte of Data Memory per input channel).

Information may be read from the Data Memory in two ways: through the microprocessor port, or by a sequence that places Data Memory contents into an ST-BUS output channel. For a microprocessor port access, the address of a location in Data Memory is formed by Control Register bits 2, 1 and 0 (CR_b2, CR_b1 and CR_b0) and external address bits A4, A3, A2, A1 and A0. When information in Data Memory is destined for an ST-BUS output channel, the address for Data Memory is formed by the contents of the Connection Memory Low byte associated with the output channel. This latter type of Data Memory access can only occur when the output channel has been placed into Switching mode.

Connection Memory Low is a block of memory the same size as the Data Memory for the relative device (MT8980 or MT8981). Each byte in Connection Memory Low is associated with an output channel. Information can be placed into Connection Memory Low locations only through the microprocessor port, but it may exit in several directions: the information may be read by the microprocessor, form the address of a location in Data Memory (Switching

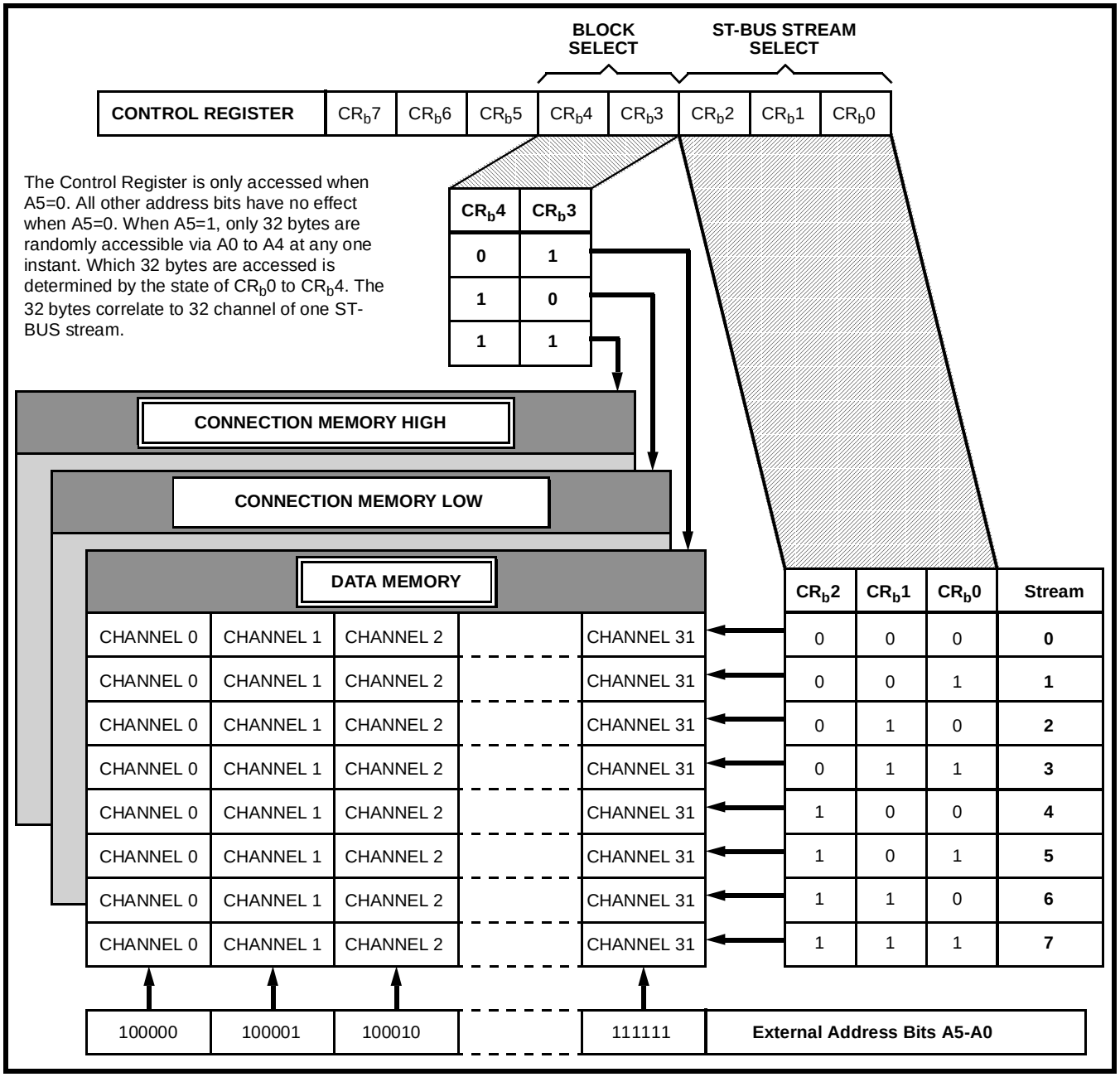

Figure 3 - Relationship Between CR_b4-0, A4-0 and Channel Access for the MT8980

mode) or be output in the associated ST-BUS output channel (Message mode). When the microprocessor is writing to or reading from Connection Memory Low, the address of the location being written to is formed by CR_b2, CR_b1, CR_b0, A4, A3, A2, A1, and A0. Either the Message mode or Switching mode function of the Connection Memory contents is performed automatically once per frame.

Connection Memory High appears to the microprocessor like Connection Memory Low did, in terms of size of the memory array, method of addressing and being able to be both read from and written to. Each byte of the Connection Memory High is associated with the same output channel as the related byte in the Connection Memory Low.

Only three bits of the Connection Memory High are useful, bits CMH_b2, CMH_b1 and CMH_b0. The other five bits are read as zeroes, and are not changeable. When CMH_b0 is a logical ‘0’ the output channel associated with the Connection Memory High location is placed into a high impedance state. When CMH_b0 is set to a logical ‘1’, the output channel driver is turned on and information from either Data Memory (switched mode) or Connection Memory Low (Message mode) is placed on the output. CMH_b1 controls the state of the bit on the CSTo output that is associated with that particular Connection Memory High location and channel. This bit is not used in the MT8981. CMH_b2 is the channel’s mode control bit. When CMH_b2 is a logical ‘0’, the associated output channel is placed into Switching mode and the contents of the Data Memory location addressed by Connection Memory Low are output on the channel. When CMH_b2 is a logical ‘1’, the output channel is in Message mode and the contents of Connection Memory Low are placed on the output channel.

The Control Register is an eight bit register which may be written to or read from the microprocessor port. This register is accessed when A5 is low. When A5 is high, one of the other memory blocks, Data Memory, Connection Memory Low or Connection Memory High, is visible to the microprocessor port. CR_b4 and CR_b3 determine which of the memory blocks is being accessed and CR_b2, CR_b1 and CR_b0 determine which portion of the selected memory block is visible. Each memory block portion selected by CR_b2-0 is 32 bytes long, corresponding to the 32 channels of one of the ST-BUS streams. The state of A4-A0 determines which byte in a 32 byte segment is accessed by the microprocessor at any one time. Figure 3 shows the relationship between CR_b4-0, A5-A0, and channel accesses in the MT8980.

CR_b6 is an overall Message mode bit. This bit can override the setting of the CMH_b2 and the CMH_b0 of an output channel. If CR_b6 is set to a logical ‘1’, all output channels are in Message mode and the CMH_b0 and the CMH_b2 of each channel has no effect. If CR_b6 is low, the Connection Memory High bits determine the mode of their respective channels (Switching or Message) and whether the channel is in a high impedance state or not.

CR_b7 is called the Split Mode bit and is an override of CR_b4 and CR_b3. If CR_b7 is set to a logical ‘1’, the device is in split mode. All microprocessor writes to the MT8980 in Split mode are to the Connection Memory Low. All microprocessor reads of the MT8980 in split mode are from Data Memory. If CR_b7 is a logical ‘0’, memory block selection is determined by CR_b4 and CR_b3.

The last bit in the Control Register, CR_b5, is unused. Also, when using the MT8981, CR_b2 is unused. It is also important to note that CRb4 and CRb3 must never be set to ‘00’.

3.4 Programming Examples

The way the control register, the memory blocks and the external address bits work together is clarified with several examples.

In the first example, the operation of reading information contained in Channel 5 of Input Stream 2 (STi2) through the microprocessor port is examined. To read ST-BUS input information, the microprocessor must read the Data Memory. Data Memory may be specified in the Control Register by either setting the Split Mode bit, CR_b7, to a logical ‘1’, or by setting CR_b4 and CR_b3 to the binary state ‘01’. To access the portion of Data Memory corresponding to STi2, CR_b2, CR_b1 and CR_b0 must be set to ‘010’. CR_b6 and CR_b5 have no effect on this example. Once the Control Register bits are set up in this manner, the microprocessor can read the information contained in any of the channels of STi2 by keeping A5 high and using A4-A0 to determine which channel is to be read. To access Channel 5, A4-A0 must be set to ‘00101’.

Example two consists of switching Channel 31 on STi0 to Channel 1 on Output Stream 7 (STo7). In this case, the locations corresponding to Channel 1 on STo7 in the two Connection Memories have to be accessed. Connection Memory Low must contain the Data Memory address of the source channel. Connection Memory High specifies the mode of Channel 1 on STo7 and whether it is in a high impedance state or not. To access Connection

Memory Low, CR_b4 and CR_b3 must be set to ‘10’. To access STo7s’ 32 byte segment, CR_b2, CR_b1 and CR_b0 must be set to ‘111’. CR_b7 may be either a ‘1‘ or a ‘0’ for writes (if a ‘0’ then CR_b4 and CR_b3 must have been set properly). For reads, CR_b7 must be a ‘0’. If it was a ‘1’, then CR_b4 and CR_b3 would have been inconsequential and the read would have been from Data Memory. CR_b6 and CR_b5 do not affect the access of the Connection Memory Low byte. To access Channel 1, A4-A0 must be set to ‘00001’.

The information that must be written to the Connection Memory Low byte to indicate the source of the information output in STo7 Channel 1 is ‘00011111’. The three most significant bits of this byte are selecting the input stream and the five least significant bits are accessing the input channel on that stream.

To access Connection Memory High for the same output channel, CR_b4 and CR_b3 must be set to ‘11’ and CR_b7 must be ‘0’. CR_b2, CR_b1 and CR_b0 need not be changed, and the same setting of A4-A0 must be used. CMH_b0 must be set to a ‘1’ to ensure that the output channel is not in a high impedance state and CMH_b2 must be ‘0’ so the channel is in Switching mode. For the same reason, CR_b6 must be a ‘0’.

The last example involves writing information from the microprocessor port to output Channel 16 on STo4. The associated bytes for this channel in the Connection Memories may be accessed in the manner described in the last example. The only difference in the procedure is the setting of CR_b2, CR_b1 and CR_b0 (‘100’), and the setting of A4-A0 (‘10000’). If CR_b6 is set to a ‘1’ (overall Message mode) then the contents of the Connection Memory high are unimportant. If CR_b6 is a logical ‘0’, CMH_b2 must be set to a ‘1’ to specify Message mode, and CMH_b0 must be set to enable the output drivers for STo4 Channel 16. Information desired in the channel must be written to the associated Connection Memory Low byte.

4.0 Delay through the MT8980

There are standards which dictate the allowable delay that can be tolerated in an end-to-end communication link. When designing a system, the applicable standard or standards must be selected and met. The delay through the MT8980 is described and explained here, so the user may understand how the MT8980 may be used to comply successfully with a particular standard.

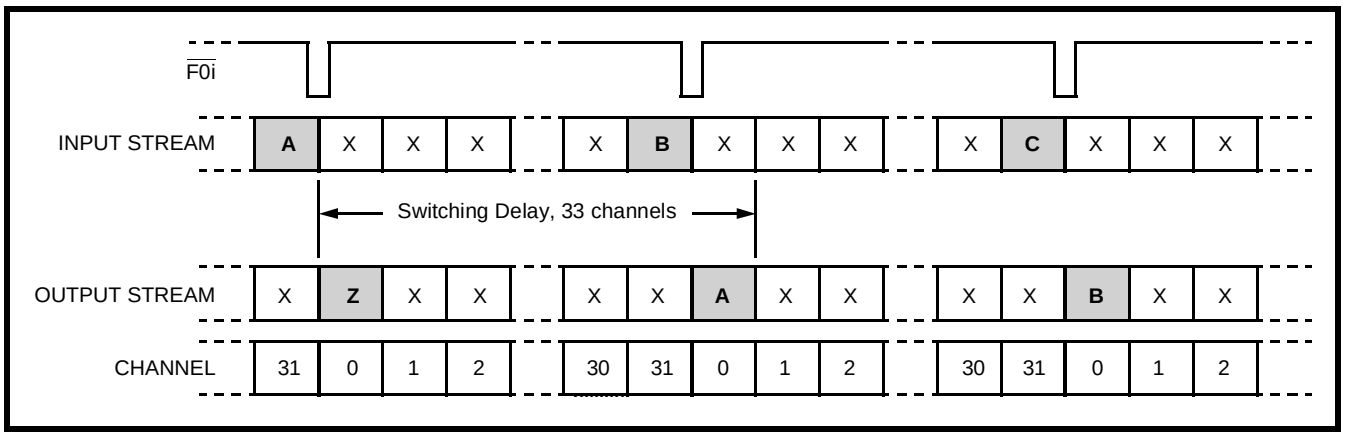

Figure 4 - Delay for Information Switched One Channel Forward, from Channel 31 to Channel Zero

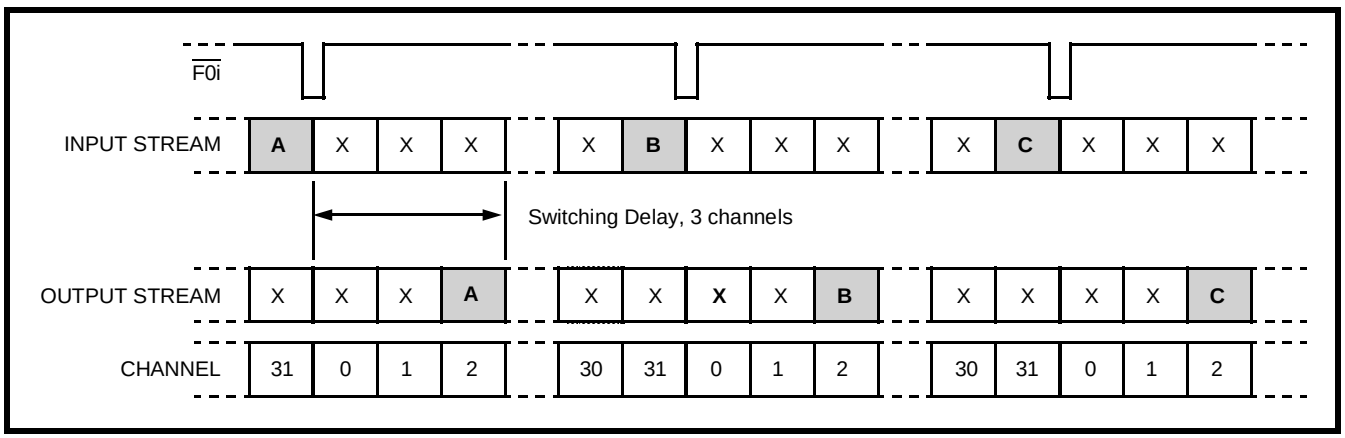

Figure 5 - Delay for Information Switched Three Channels Forward, from Channel 31 to Channel Two

4.1 Switching Mode

A delay through the MT8980 results when transferring channel information from an ST-BUS input stream to an output stream. The input information must first be transferred to the Data Memory, where it waits until the next available timeslot allocated to the output channel. Recalling the definition of the ST-BUS, 32 channels arrive at an input and 32 channels leave an output. Each channel is allocated one timeslot per frame, so there are 31 timeslots intervening between each timeslot allocated to a particular channel. If the information has just missed an appropriate timeslot for the output channel, it may have to wait a number of timeslots for the next allocated channel time.

The last bit of a channel is received before the channel information is said to have entered the MT8980 and is sent to the Data Memory. Consequently, a channel's information is sent to Data Memory during the timeslot immediately following a timeslot allocated to the channel. Conversely, channel information is queued for an output channel during the timeslot immediately preceding a timeslot allocated to the channel.

The implication of this ordering of events is that information entering the MT8980 cannot leave it in the same timeslot, or in the timeslot immediately following. Therefore, information that is to be output in the same channel position as the information is input, relative to frame pulse, will be output in the following frame. As well, information switched to the channel immediately following the input channel will not be output in the timeslot immediately following but in the next timeslot allocated to the output channel, one frame later. Examples of channels being switched to the channels immediately following are: switching from Channel 1 to Channel 2, Channel 4 to Channel 5, Channel 30 to Channel 31, Channel 31 to Channel 0 etc. All of the channel changes mentioned above are independent of stream changes.

Whether the information can be output during the second timeslot after the information entered the MT8980 depends on which ST-BUS stream the channel information enters on and which ST-BUS stream the information leaves on. This situation is caused by the order in which input stream information is placed into Data Memory and the order in which stream information is queued for output. Table 1 shows the input/output streamcombinations that would allow information to leave on the second timeslot after the information was received.

Input Stream	Output Stream
0	1,2,3,4,5,6,7
1	3,4,5,6,7
2	5,6,7
3	7
4	1,2,3,4,5,6,7
5	3,4,5,6,7
6	5,6,7
7	7

Table 1. Input Stream to Output Stream Combinations that can Provide the Minimum 2 Channel Delay

Output Stream	Minimum Setup time
0	$t_{CLK}\times19$
1	$t_{CLK}\times19$
2	$t_{CLK}\times15$
3	$t_{CLK}\times15$
4	$t_{CLK}\times11$
5	$t_{CLK}\times11$
6	$t_{CLK}\times7$
7	$t_{CLK}\times7$

Table 2. Minimum Setup Times for Writing to a Channel in Message Mode (Relative to the Start of the Channel)

Input Stream	Minimum Delays
0	$t_{CLK}\times1$
1	$t_{CLK}\times5$
2	$t_{CLK}\times9$
3	$t_{CLK}\times13$
4	$t_{CLK}\times1$
5	$t_{CLK}\times5$
6	$t_{CLK}\times9$
7	$t_{CLK}\times13$

Table 3. Minimum Delays for Reading Input Channels from Microprocessor Port. Delay is Relative to End of Channel

Information can always leave the MT8980 by the third timeslot after entering the MT8980, independent of which stream it is input on and which

stream it is output on. Figure 4 shows the delay of data through the MT8980 for information switched one channel forward. Figure 5 shows the delay of data through the MT8980 for information switched three channels forward. Note that in Figure 4 the input data is not output until one frame later, but in Figure 5 the delay is minimal.

The maximum delay is one frame period (approximately 125 microseconds or 512 $\overline{C4i}$ clock cycles) plus two channels. This is the delay resulting if a switch two channels forward does not meet the input/output stream requirements of Table 1. The minimum delay achievable is two channels. This is the delay resulting if the requirements of Table 1 are met.

4.2 Message Mode

In Message mode, there are two delays to contend with. The first delay is the delay between receiving information on the ST-BUS and reading it through the microprocessor port (this delay is not actually constrained to Message mode, as an input channel may be read by the microprocessor independent of any channels being in Switching mode or Message Mode. The other delay is the delay between writing information into Connection Memory Low and transmitting the information on the ST-BUS output. As with Switching mode, information destined for a particular timeslot on the ST-BUS is sent to the output stream queue during the preceding timeslot. The time during the preceding channel that this occurs is dependent on the stream. The sequence in which the information destined for an output channel is sent to each stream is: STo0, STo1, STo2, STo3, STo4, STo5, STo6, STo7.

To use this information about output sequencing and internal information movement, the microprocessor must synchronize transfers with ST-BUS timing. If the microprocessor transfer is not made before a channel's information is sent to the queue, the newly transferred information will not be output until the next frame.

To ensure that a particular channel on a particular stream may be written to without a one frame delay, a minimum set up time for the write in question must be respected. This set up time is relative to the start of the output channel's timeslot. Table 2 shows the minimum set up times for the different ST-BUS streams (set up times are expressed in multiples of t_{CLK}, the period of the $\overline{C4i}$ clock). There can also be a maximum set up time, if writing information one frame early to a particular channel on a particular stream is undesirable. The maximum set up time is merely the minimum set up time plus one frame.

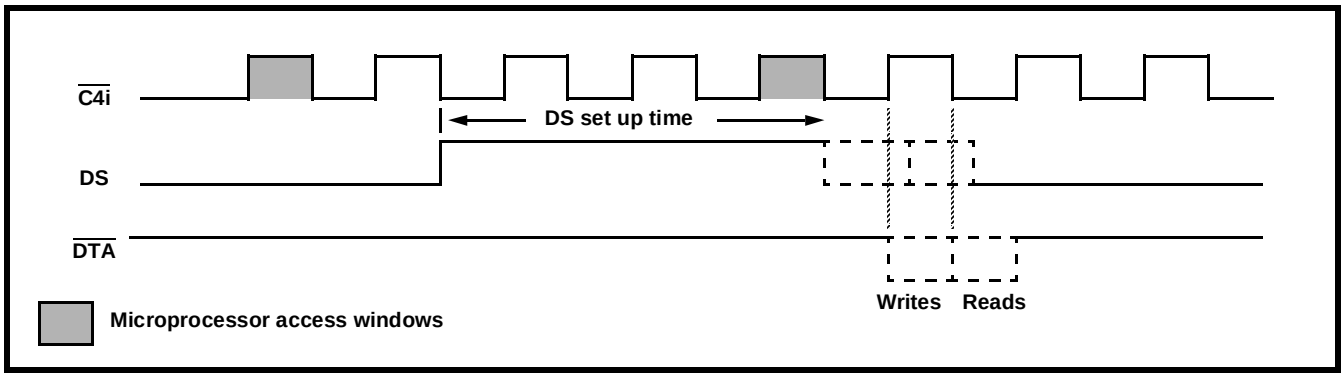

Figure 7 - DS Setup Time and $\overline{DTA}$ Return Relative to $\overline{C4i}$ and Microprocessor Access Window

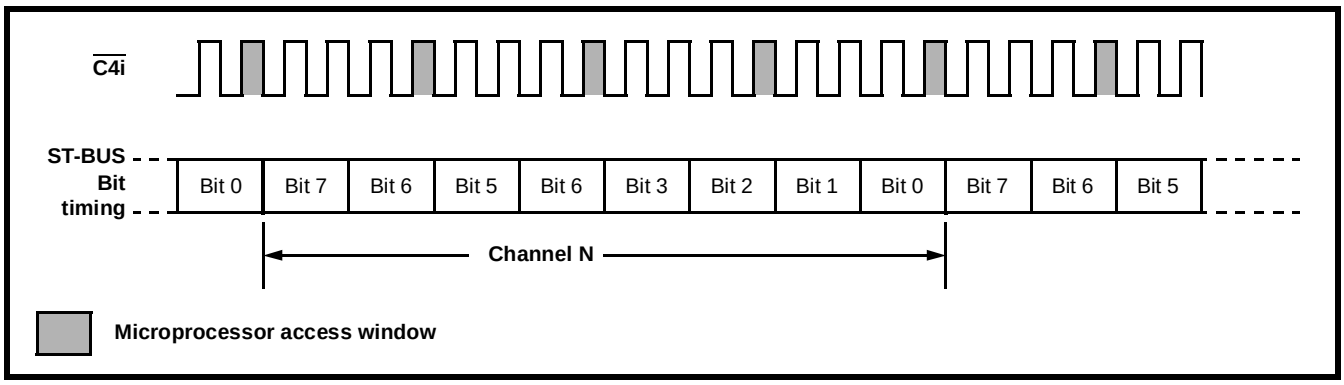

Figure 6 - Position of Microprocessor Access Windows Relative to ST-BUS Channel and $\overline{C4i}$ Timing

The minimum delays from the time a channel enters the MT8980 on an ST-BUS input to the time that the information in the channel can be extracted from the Data Memory are given in Table 3. They are measured from the end of the input channel's timeslot to the rising edge of DS. The rising edge of DS corresponds to the earliest point DS may be activated to access the first available microprocessor window from which the information will be visible (see SECTION 5.0).

If the byte of Data Memory that corresponds to the desired channel is accessed earlier than the minimum delay dictates, then the contents of the byte read by the microprocessor will be the information that the channel contained on the previous frame. Conversely, if slightly less than one frame of delay is added to the minimum delay (508 $\overline{C4i}$ clock cycles rather the 512 that make up a frame) the contents of the byte will be the information contained in the channel in the frame following the desired frame.

5.0 Microprocessor Accesses

In the MT8980 data sheet, a parameter is specified (t_{AKD}) for the maximum time taken to return Data Acknowledge after DS goes high. There are two values for this time, one labelled fast, and one labelled slow. The fast t_{AKD} applies to writes to the Control Register. The Control Register can accept data very quickly, and will not cause wait states for most microprocessors. Reading the Control Register and reading or writing to any other part of the MT8980 will receive a slower response (t_{AKD} slow).

Slow microprocessor bus transfers occur because the MT8980 only allocates discrete access windows for the slow transfers. Microprocessor access windows occur every four $\overline{C4i}$ clock cycles during a frame, relative to the frame boundary. Figure 6 and Figure 7 show the relationship between microprocessor access windows, $\overline{C4i}$, DS and the return of $\overline{DTA}$.

$\overline{DTA}$ goes low one half $\overline{C4i}$ cycle after a microprocessor access window in which the set up requirements for DS are met and the operation is a write. If the operation is a read and DS setup is met, $\overline{DTA}$ goes low one $\overline{C4i}$ clock cycle after the access window. DS must be set up three $\overline{C4i}$ cycles ahead of the falling edge of every microprocessor window cycle to meet minimum set up requirements for accessing a window.

If the controlling microprocessor can determine where the falling edge of every fourth $\overline{C4i}$ cycle in a frame occurs, it may use this as a signal to continue on to the next bus cycle rather than wait for $\overline{DTA}$ to go low. $\overline{DTA}$ is a signal provided only to tell the microprocessor that the MT8980 is ready to finish the current bus cycle. DS could be removed between the end of the microprocessor access window and the point where $\overline{DTA}$ would normally go low and the transfer of information would still complete successfully. In such a situation, $\overline{DTA}$ would not go low for that particular bus cycle.

Figure 7 shows the two ways in which $\overline{DTA}$ can be returned in response to a microprocessor access. The possible transitions of $\overline{DTA}$ and DS are dotted lines, not solid, because there are several options. If DS goes low at the first transition point, $\overline{DTA}$ does not go low. If DS is held high, the two places that a transition on $\overline{DTA}$ can occur are shown.

6.0 External Control Using CSTo

CSTo is a 2.048 Mbit/s output which is, like the ST-BUS streams, divided into frames that are 256 bits long. Each bit is controlled by one of the 256 CMH_b1's. If a CMH_b1 is a logical '1', the corresponding bit on CSTo is a high. If the CMH_b1 is a logical '0', the corresponding bit on CSTo is a low. Unlike the other ST-BUS outputs, though, CSTo cannot be placed into a high impedance state.

The CMH_b1's of locations that are related by channel timeslot are output sequentially. As an example, there are eight CMH_b1's corresponding to channel zero, one bit for each stream. These bits are output sequentially on CSTo in the following order:

a) CMH_b1 for STo0 Channel 0
b) CMH_b1 for STo1 Channel 0
c) CMH_b1 for STo2 Channel 0
d) CMH_b1 for STo3 Channel 0
e) CMH_b1 for STo4 Channel 0
f) CMH_b1 for STo5 Channel 0
g) CMH_b1 for STo6 Channel 0
h) CMH_b1 for STo7 Channel 0

The eight CSTo bits that correspond to a channel position are output in the timeslot preceding the actual channel position. The reason for this is that CSTo bits are designed to perform external control functions on the individual channels they correspond to.

One control function these bits may perform is to control loop back circuitry for individual channels. If a CSTo bit is set, the loopback circuitry could drive an MT8980 input with the information from the corresponding output channel. This function would

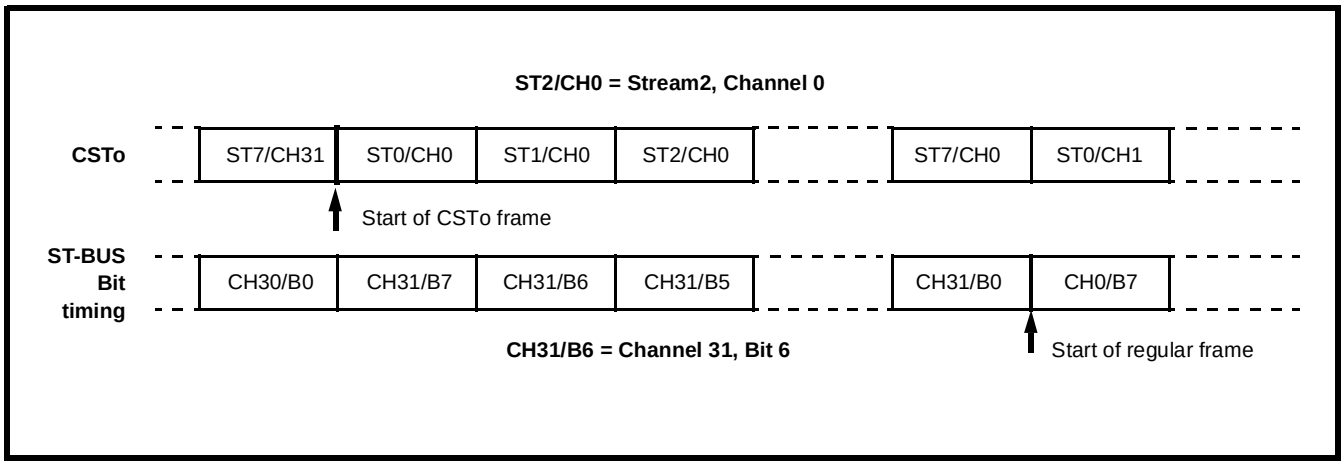

Figure 8 - CSTo Bits Compared to ST-BUS Timing

be useful for performing system level diagnostics on matrices of MT8980s.

Another use for the CSTo bits is in the astronavigation of the microprocessor with ST-BUS timing. CSTo could notify the microprocessor that a predetermined position in the ST-BUS timing has occurred. For example, to mark Channel 7 entering the MT8980, clearing all CSTo bits except the CMH_b1 of STo0 Channel 9 would produce a waveform that had a recurring, high-going, bit-wide pulse immediately following the last bit of Channel 7.

Figure 8 shows the relationship between CSTo bits, ST-BUS timing and the Connection Memory High bytes producing the bits.

7.0 High Impedance Capability

The MT8980 has the capability to put individual channels into a high impedance state, so large switching matrices may be constructed with multiple MT8980's. The ODE input, when it is low, forces all ST-BUS channels into the high impedance condition. When it is high, each channel is controlled by its individual CMH_b0. When the CMH_b0 is logical '1', the channel is driving the ST-BUS. When the CMH_b0 is logical low, the ST-BUS is in the high impedance state during the channel's timeslot.

7.1 Initialization of the MT8980

On initialization or power up, the contents of the Connection Memory High can be in any state. This is a potentially hazardous condition when multiple MT8980 ST-BUS outputs are tied together to form matrices, as these outputs may conflict. The ODE pin should be held low on power up to keep all outputs in the high impedance condition.

During the microprocessor initialization routine, the microprocessor should program the desired active paths through the matrices, and put all other channels into the high impedance state. Care should be taken that no two connected ST-BUS outputs drive the bus simultaneously. When this process is complete, the microprocessor controlling the matrices can bring the ODE signal high to relinquish high impedance state control to the CMH_b0s.