ISSUE 1   April 1987

Contents

Applications

- ADPCM Switching
- Subrate Multiplexing On Digital Trunks

1.0   Introduction

In digital switching environments there is often a need to switch information in blocks of less than 8 bits.   These areas include applications such as submultiplexing data, and digital compression techniques, i.e., ADPCM. In submultiplexing of data, the user may need to switch 1, 2, or 4 bit nibbles, and in ADPCM the user needs to switch two 4 bit nibbles per byte. For the purposes of discussion, this document will describe the circuitry required to build a 256 by one bit switch, and will interpolate the principles used to illustrate how a 256 by two and a 256 by four bit switch can be built.

2.0   Circuit Description

The main component in a 256 by one bit switch is the MT8980, 256 by eight bit time space switch. The principle used to convert the MT8980 from a 256 by eight bit switch to a 256 by one bit switch is to dedicate each of the 256 input channels of the Data Memory to each of the 256 one bit input channels. The shift register, in Figure 1, does this by delaying each bit by one position before it is clocked into the eight ST-BUS inputs. In this manner each input bit will be placed into each of all eight possible bit positions on sequential ST-BUS streams. Each bit and the delayed version will be stored in one or two consecutive bytes of Data Memory. For example: Bit zero of channel 0 will be in position 0 in stream 0, position 1 in stream 1,....., and position 7 of stream 7. All eight bits in any input channel will follow the same type of pattern. This pattern is illustrated in Figure 2. A single output stream is then created from all eight ST-BUS output streams with the eight to one mux (74LS251) in Figure 1. During each ST-BUS channel period the mux selects one bit from each output stream, beginning with stream 0 (STo0). By only taking one bit from each of the output streams during every channel, the MT8980 is only switching one bit from the Data Memory to the output stream. Even though the MT8980 can only switch eight bits at a time from the Data Memory to an output stream, the input shift register and the output mux effectively allows the device to switch a single bit.
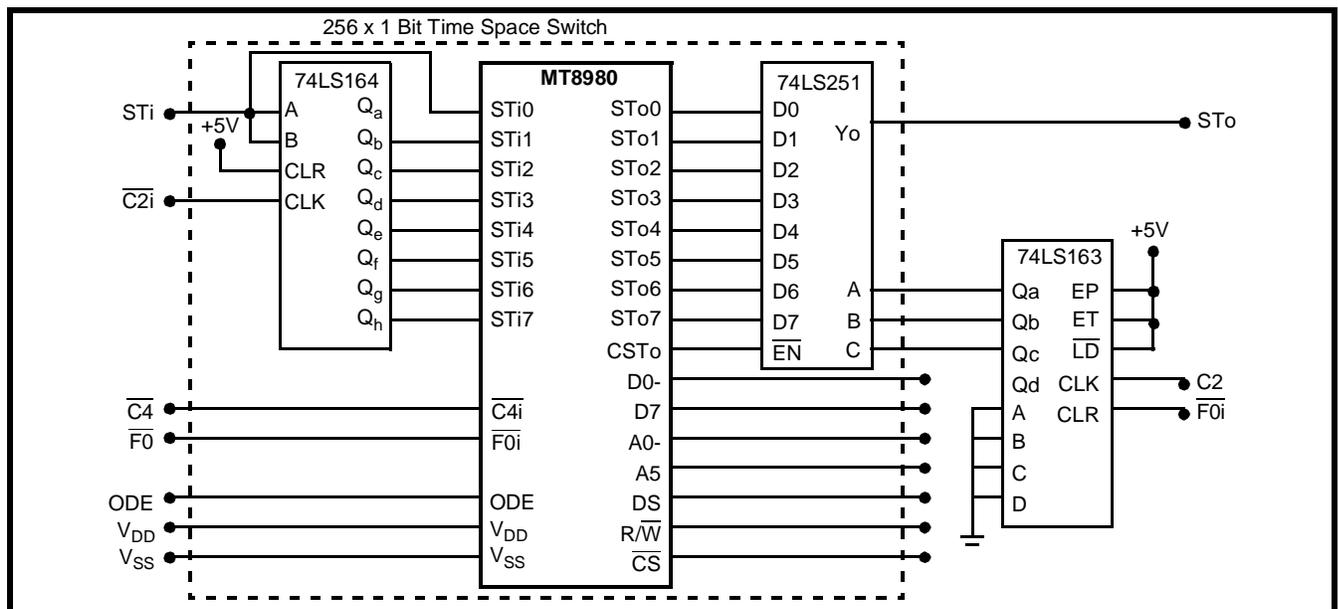

Figure 1 - One Bit Time Space Switch Circuit Schematic

2.1 Programming Algorithm

In order to program the device there are two numbering conventions to be established. The first is the numbering of the 256 input and output channels. These are numbered from 0 to 255, zero being the first channel and 255 being the last channel. The second is that once the 256 1 bit channels have been decoded to a ST-BUS channel and a bit number, the bit positions within a ST-BUS channel are numbered zero to seven, with zero being the MSB and seven being the LSB. This is opposite to the ST-BUS numbering convention, but it will simplify the programming algorithm. Listed below are the eight steps necessary to calculate the required output stream, the output channel and the source address of Connection Memory Low (CML) from the input and output channel numbers (0 to 255). When the following description is referring to conventional ST-BUS numbering schemes it will refer to it as such.

1) Obtain input and output channel numbers. (0 to 255).

2) The five MSBs of each channel number form the ST-BUS channel number (0 to 31).

3) The three LSBs of each channel number form the bit number (0 to 7, MSB to LSB).

4) Output bit number=ST-BUS output stream number, i.e., CML block .

5) Output ST-BUS channel number=CML address.

6) Output bit number-Input bit number=shift .

7) If shift=positive then shift=ST-BUS source stream number, i.e., three MSBs of CML   Input ST-BUS channel number=source channel number, i.e., five LSBs of CML.

8) If shift=negative then shift+8=ST-BUS source stream number. i.e., three MSBs of CML   Input ST-BUS channel number+1= source channel number, i.e., five LSBs of CML.

Example 1:
Input channel number=74 (01001|010)
Output channel number=151 (10010|111)
Input ST-BUS channel number=9 (01001)
Output ST-BUS channel number=18 (10010)
Input bit number=2 (010)
Output bit number=7 (111)
Output bit number=Output ST-BUS stream=7
Output ST-BUS channel number=CML address=18
Shift=7-2=+5 therefore ST-BUS input stream=5 (101)
Input ST-BUS channel=Source ST-BUS channel=9
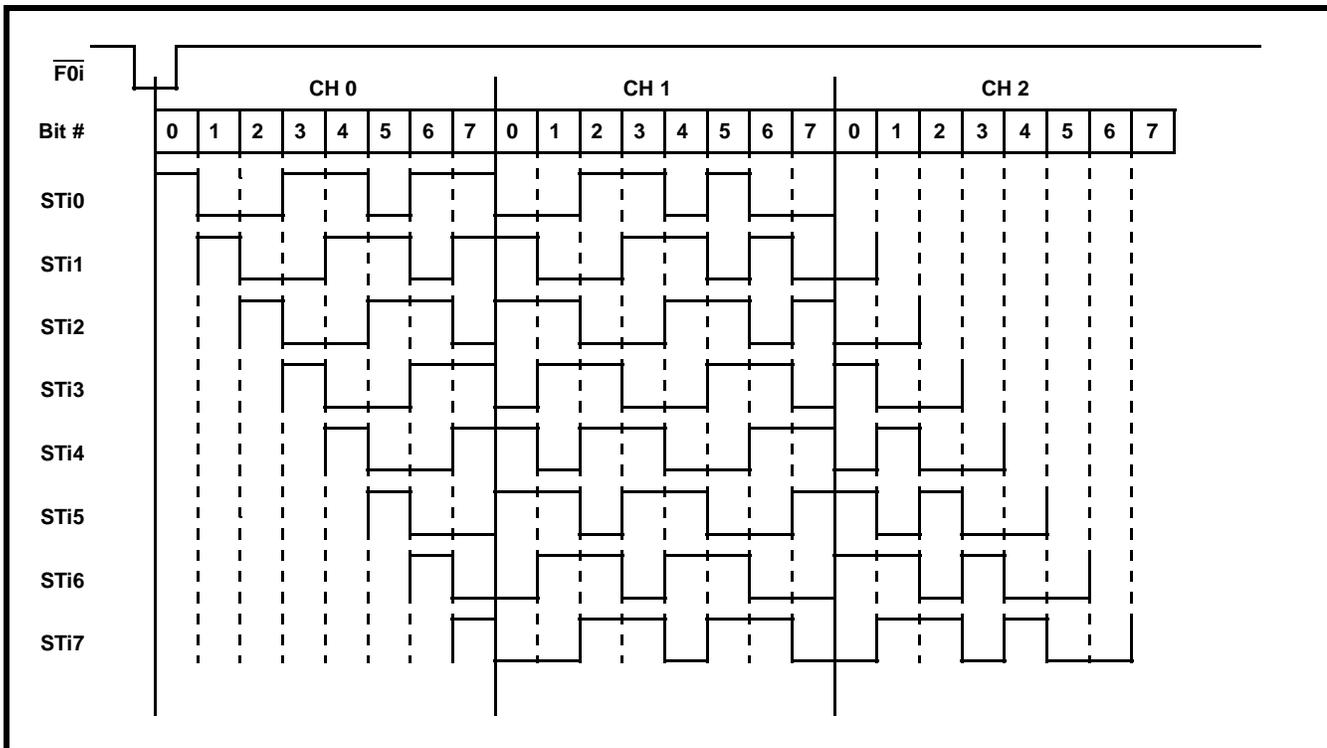CML source address(contents)=stream 5, channel 9 (101|01001)

Figure 2 - Contents of the Data Memory

Conclusion: To switch input channel 74 to output channel 151 write "10101001" to byte 18 of the CML of stream 7

Example 2:
Input channel number=99 (01100|011)
Output channel number=57 (00111|001)
Input ST-BUS channel number=12 (01100)
Output ST-BUS channel number=7 (00111)
Input bit number=3 (011)
Output bit number=1 (001)
Output bit number=ST-BUS output stream=1
Output ST-BUS channel number=CML address=7
Shift=1-3=-2 therefore ST-BUS input stream=
-2+8=6 (110)
Input ST-BUS channel+1=Source ST-BUS channel=12+1=13
CML source adress (contents) = stream 6, channel 13 (110|01101)

Conclusion: To switch input channel 99 to output channel 57 write "11001101" to byte 7 of the CML of stream 1.

3.0 Building Single Bit Switch Matrices

In order to build a switching array that is larger than 256 channels, several of the single bit building blocks in Figure 1 must be connected together. In order to connect several outputs together each channel must be under individual tri-state control. This can be accomplished with the CSTo bit in the Connection Memory High (CMH) bytes and the CSTo pin. Each CMH byte contains the control bit for one of the 256 bit positions on the CSTo pin. The only anomaly to take into account is that the bit controlled by the CMH byte is in channel N-1 on CSTo. If N represents the byte number, and M

represents the stream number then the bit position that an individual CMH byte controls is byte N-1 bit position M. The CSTo pin in Figure 1 is connected to the enable input of the 74LS251 (pin 7) so that the 256 output channels can be put into tri-state by programming the CMH in the appropriate channel. To enable the output for channel 57, in example two, a "0" would be written to ST-BUS bit of 1 of CMH stream 1 channel 8, i.e., ST-BUS output channel number+1.

4.0 Two and Four Bit Switching

The principles of the single bit switch can easily be adapted to handle any number of bits. If the number of bits per channel increases then the delay between ST-BUS inputs increases by the same amount. If the number of bits per channel increases then the number of bits selected from each ST-BUS output stream increases by the same amount. For example, to build a two bit switch the delay between each of four ST-BUS inputs would be two bits, and the output mux would select two bits from each of four ST-BUS outputs. This would occupy four ST-BUS links, for a total of 128, two bit channels. The same circuit can be repeated on the four remaining ST-BUS links of the MT8980 for a combined total of 256 channels. A four bit switch could be built with a delay of four bits between two ST-BUS inputs and an output mux that selects four bits from each of two ST-BUS outputs, see Figure 3. If the same circuit is then repeated four times on a single MT8980 it will have a total capacity of 256, four bit channels.

5.0 Timing

The 256 channels from the output mux may contain glitches, depending on the data pattern. These glitches will always be at the leading edge of a bit period and therefore will not have any affect on other ST-BUS components. If it is desirable to remove the
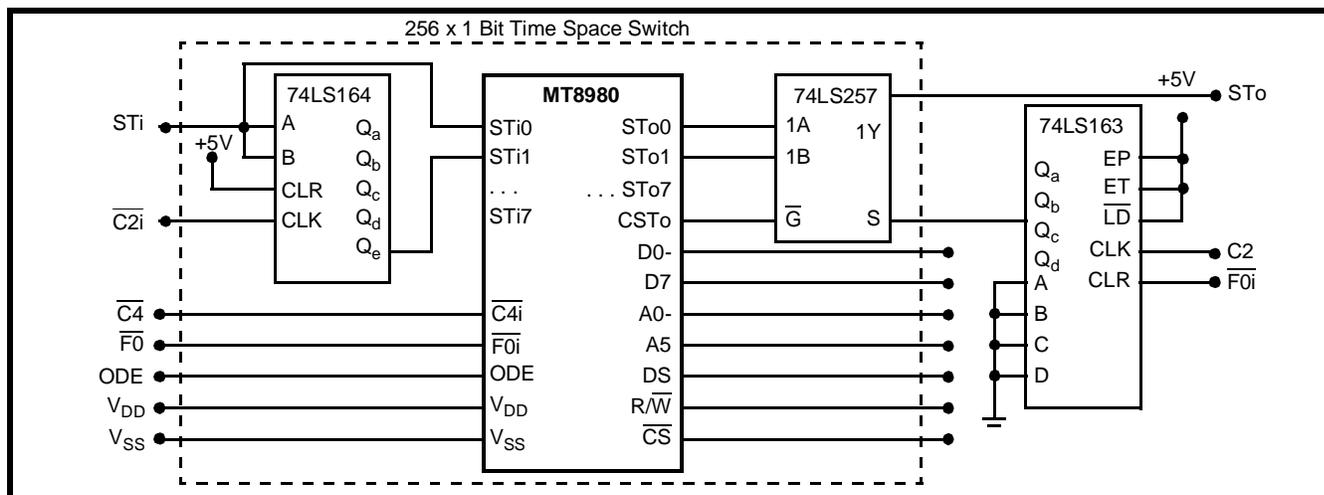

Figure 3 - Four Bit Time Space Switch Circuit Schematic

glitches it can be done very easily with D-type flip flops. It should be recognized that this creates a one bit delay that must be taken into account when determining the switch pattern to program.   It is easily taken care of by subtracting one from the output channel number before the calculation in Section 2 is performed.