Spectrum Analyzer

Chan Min
Matthew A. Smith
Ronald Koss

Fall 2000

Chapter 1

Objective:

The spectrum analyzer circuit takes a stereo input and displays a volume by frequency display on a standard PC monitor. Finding the amplitude of a waveform at each frequency and then mapping this to the screen using a discrete number of levels achieve this. To complete this process, four stages are needed. The first stage is the loopback circuit which routes the serial input to the serial output. This serial input can also be extracted from the circuit for our own uses. The second stage takes the inputs from the loopback circuit and finds the amplitude and frequency of the waveforms. The third stage writes the volume levels into the RAM according to frequency. The fourth stage reads these volume levels and displays them onto the screen.

Realized:

The Volume Bar circuit takes a stereo input and displays the Left or Right Channel volume to the LED bars (Only one may be displayed due to number of bars). Finding the amplitude of a waveform and then mapping it to the LED bars achieve this. To complete this process, three stages are needed. The first stage is the loopback circuit which routes the serial input to the serial output. This serial input can also be routed to other circuits. The second stage takes the loopback input and finds the amplitude of both the left and right channels. The third stage maps the volume level to the bars by breaking it into 8 discrete levels.

The RAM to VGA circuit takes a discrete volume level and displays it to the screen. This is achieved by writing all the volume levels to the RAM and then reading this data (color) and prints it to the screen. This process requires two stages. The first stage is to take the volume and write the color to a specific RAM address based on this volume level. The second stage reads these color values from the RAM and displays them on the screen. This cannot be achieved due to the timing between the reading of the color values and the maximum refresh rate of a standard monitor.

Chapter 2

Volume Bar Circuit:

I/O:

Clk: This is the main clocking frequency from the XESS Board set at 12.5 Mhz

Rst: This is the global reset signal that places the entire circuit in a known state.

Mclk: This is the main clk sent to the Codec. It is set to the same value as Clk.

Lrck: This is the left/right channel select which has a duration of 128 clk cycles for each channel.

Sclk: This the shift clk which is 4 times slower than the master clk. This clk is used to shift the 20 bit serial data in from the ADC and out to the DAC.

Sdout: This is the serial data coming out of the ADC.

Sdin: This is the serial data going in to the DAC.

Bars: This signal is a string of binary bits that specify which Bars on the XESS Board should light. They are active low and therefore 0 is lit and 1 is off.

Unfortunately, Codec requires the duration of 128,192, or 256 clock cycles for each channel. So, the loopback circuit takes twenty cycles of shift-clock, which is eighty cycles of master clock, to load one twenty bits serial data. When shifting in these serial data, most significant digit is shifted first. In the loopback circuit, the channel duration is set to 128. So, the loopback circuit wastes 48 master clock cycles after loading a twenty bits data for left or right channel. This 20 bit value is parallel loaded into the outgoing register and is shifted into the DAC in 128 master clock cycles with 48 master clock cycles wasted. Shifting in and out is done in parallel.

The MinMax circuit takes each twenty bits of data. As soon as it reads the data value, it will do a two’s complement comparison with the previous data value read. When the current data is greater than the previous value, the state in the control unit will go into the increasing loop until it reads the first decreasing value. As soon as the state sees the first decreasing value, it will load the previous data into the latch called “max”. Similarly, when the compare-state is in the decreasing loop, it will load the minimum value as soon as it sees the first increasing value. After a minimum value is found the control unit goes to the increasing loop and as soon as a maximum value is found the control unit goes to the decreasing loop. This process goes on as long as the circuit is in operation. GetDifference simply takes the maximum value minus the minimum value in two's complement and outputs the difference as the amplitude or volume.

The Bars circuit splits the volume read in into 8 different discrete ranges and then lights the corresponding bars. These ranges are adjusted to compensate for background noise and unreachable volume levels.

RAM to VGA circuit

I/O:

Clk: This is the main clocking frequency from the XESS Board set at 12.5 Mhz

Rst: This is the global reset signal that places the entire circuit in a known state.

Hsyncb, Vsyncb: outputs for the horizontal and vertical sync pulses. When they are zero, the screen is blank in the area and when they are one, the rgb is displayed.

RGB: Controls the red, green, blue guns of the monitor. Different values give different color mixes.

Address: address to be read from or written to in RAM

Data: data to be read from or to be written to RAM

Ceb,oeb,web: Control lines to RAM, Chip select, Ouput Enable, and Write Enable. They are active low.

Dis8951: Disables the 8051 microcontroller so that it does not interfere with the RAM

The RAM Writer waits for a request from the RAM Reader. When this is received, the RAM begins at address 0000 and writes to RAM address 0C00. The current volume levels given by GetDifference determine the data written into the RAM. Red is written up to the volume level and the rest is written black. Both Bars are written by one run of the circuit and they are divided by a bar of all black of equal thickness. The complete signal is then sent to the RAM Reader as a request.

The RAM Reader waits for a request from the RAM Writer. When this is received, the screen is blanked out while within a certain vertical and horizontal range. When outside the blank range the RAM is read starting at address 0000 until 0C00. The address is not incremented continuously throughout the entire range; this is due to intermittent blank stages. The complete signal is then sent to the RAM Writer as a request.

Four pixels can be stored in a single byte of RAM. Since the RAM Reader prints out 256 pixels per line, 64 bytes of RAM are needed to store pixels for a single line. RAM Reader prints out 16 lines of pixels for a left channel volume bar, 16 lines of pixels for the blank between left and right channel volume bars, and 16 lines for the right channel volume bar. So, RAM Writers writes 64 x 48 bytes in the RAM, which is from 0000 to 0C00.

The RAM to VGA circuit is not working properly on the monitor as expected because of the difference in refresh rates of the monitor and the rate of the RAM Reader. The rate of RAM Reader cannot be decreased down to 60Hz to 120Hz due to Board limitation. The minimum board clock speed is still too fast for the Monitor to refresh. The codec also requires a clock of 12.5 Mhz to function properly and therefore a slower clock speed would not allow us to use the Monitor and Codec simultaneously.

The Board also does not have a sufficient number of CLBs to contain our entire design. This includes the codec analysis and graphics analysis sections.

All relevant schematics and implementation reports are attached at the end of this chapter.

Chapter 3

Volume Bar Circuit:

A simple counter can be added to the MinMax circuit to determine the frequency. This is unique to our circuit because if the amplitude were determined in another manner, this would not be possible. A counter is reset and stored in a latch every time a maximum is found, and the counter counts up every time a new data value is found. The stored value will be the corresponding wavelength to the previous volume found. This data can then be output to a modified RAM Writer which has up to 32 separate bars each corresponding to the volume of a specific wavelength range.

Thirty-two separate bars of 16 line can be displayed to the screen a once because there is a total of 512 lines in a single frame of the screen.

RAM to VGA

The RAM to VGA circuit uses a unique handshaking procedure that we designed. Since we want the circuit to write first we have the Read circuit show complete in both WaitReq and Done states. This forces the writer to go first because it shows complete only in the Done state. Therefore the process continues without interruption once started because the writer's complete signal is the same as the reader's request signal and vice versa. As soon as the first cycle of the writer completes, the reader begins and then the reader sends a complete signal in the Done state and so on.

Chapter 4

We could have minimized the number of states used within our control units if we would have approached them with a Mealy or Moore design philosophy. This would have reduced the number of CLBs needed by our design. However it may not have been enough to reduce the design down to a size that would fit on a 4010XLPC84. Had the design been scaled up to a full Spectrum Analyzer, it would require a far greater number of CLBs especially for the RAM Writer. A full spectrum analyzer could have been realized by using a number of Bar Graph displays instead of the monitor.

The Monitor would require an interrupt routine to be written so that it could sync up with the output from the RAM Reader circuit. This would require the use of the 8051 microcontroller.

We believe that all other design issues have been dealt with and do not require any further modification.

Viable Final Question:

1. Why didn't the monitor display the correct data even though the RAM was written and read correctly?
2. How could a spectrum analyzer design be implemented with a minimum modification to original design?
3. How did the MinMax circuit determine the waveform amplitude?
4. How did the RAM to VGA circuit know to write first?

Appendix A

To set the clock frequency

Follow the following instructions exactly:

1. Remove all cables from the XESS board
2. Move J12 on the XS40 board in line with other three jumpers
3. Insert the provided cd
4. Switch working directory to <cd drive>:\clock
5. reattach power and parallel cables only
6. type setclk <divisor> divisor is 1 to 2052
7. follow instructions given
8. detach all cables and move jumper to original position
9. proceed with design implementation

To duplicate the Volume Bar:

Load all of the VHD files in the barvol directory off the CD provided into Xilinx. Run through the implementation process. Select bars.ucf from the CD for the ucf file. Top Level is barvol.vhd. Using the xsload and xsport utilities provided in the Util directory on the CD type at the DOS prompt where d is the cd drive:

Before performing these steps ensure that the parallel and power cable are correctly connected to the board.

Set the clock with divisor 8

```
C:\<directory>\copy d:\util\xsload.exe
C:\<directory>\copy d:\util\xsport.exe
C:\<directory>\xsload barvol.bit
C:\<directory>\xsport 0
```

Now attach a stereo source to the Stereo In jack and a set of speakers to the Stereo Out jack.

Note: This design only functions properly on the Xstend Board V1.3 due to the Codec Difference

To duplicate the RAM to VGA

Load all of the VHD files in the ramvga directory off the CD provided into Xilinx. Run through the implementation process. Select graphictest.ucf from the CD for the ucf file. Top Level is graphictest.vhd. Using the xsload and xsport utilities provided in the Util directory on the CD type at the DOS prompt where d is the cd drive:

Before performing these steps ensure that the parallel and power cable are correctly connected to the board.

```
C:\<directory>\copy d:\util\xsload.exe
C:\<directory>\copy d:\util\xsport.exe
C:\<directory>\xsload ramvga.bit
C:\<directory>\xsport 0
```

Do not attach a monitor to the monitor output simply watch the Bar graph and 7-seg displays. Slowing down the clock allows easier viewing.