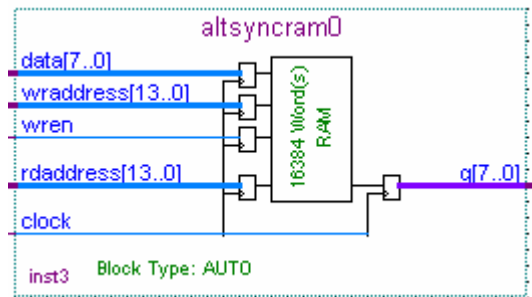


FPGA Lab 9 – Music (.wav) Player

Purpose: In this lab you will use your RS-232 serial interface to make a simple music player. You will also get experience using random access memory (RAM), which is a widely-used component in digital systems.

- 1) **Music Player.** The music player has two basic components: the download section that stores the incoming RS-232 data into RAM, and a play section that sends this RAM data to the codec. These two sections can be conveniently separated using a dual port RAM, whose module is given at the right. Note that the write and read sections operate independently. Also note that the maximum size for the RAM is about 16K 8-bit words, as determined by the internal resources of the FPGA. Unfortunately, 16K of memory only corresponds to about 1 second of music.



- 2) First code the download section, which takes data from the RS-232 interface and loads it sequentially into memory. The output of the RS-232 interface will have the 8-bit data and a “data_ready” line that pulses for one clock cycle when new data is available. Use this “data_ready” line to enable the RAM write enable line “wren”, storing the data. Also use this line to increment a counter that is connected to the address of the RAM “wraddress[13..0]”.
- 3) The play portion of the counter will use a counter that increments its output at a rate of 16 KHz, the frequency set in the .wav file. This counter is then connected to “rdaddress[13..0]”. The output of the RAM “q[7..0]” is connected to the codec.
- 4) Since this is a complicated circuit, you might want to have the push-button switches clear the counter. You might display the output of the counters on the 7-segment LED’s to indicate that the loading and counting operations are performed correctly.
- 5) Included in this lab is sound encoded as an 8-bit .wav file that has an output frequency of 16 kHz, given as the “Chimes_16s.txt” file. You can create other sound files by using the “sound recorder” program in the windows accessories directory. Load a .wav file, and convert it to the proper format using the “Save As” command. Under the change button, use the “PCM” format, along with 8-bit mono and 16 kHz options.
- 6) This data can be downloaded by using the RS-232 hyperterminal program. Use the “Transfer/Send Text” command to send your .wav file across the interface. Note that the first 20 bytes or so in this file is text describing the file type, which will look like a few milliseconds of noise at your output. For purists, you can strip off these bytes by loading your wave file in “notepad”, then saving the file, or cutting and pasting the rest of the file onto the hyperterminal program.

Optional external RAM interface:

A music player with about 1 minute of playing time can be constructed using the external RAM memory that is connected to the FPGA chip. This memory is single port, and thus has a bidirectional data bus. The circuit for the interface is given to the right, where the input and output lines at the left and righthand side correspond to that used in the FPGA dual port memory. The SRAM data bus uses a tristate output, as discussed in class, and is shown as a gated triangle symbol. The multiplexer chooses the address to be sent to the RAM using the write enable signal. The four connections on the bottom are various byte- and chip-enable lines that must be set to ground. The file “Barracuda8” has music encoded for an 8 kHz output frequency.

