Group Assignment Lab 8

ECE FPGA Design using VHDL

BRAM Music Generator

Since this is a group assignment, the goal of this assignment will be different from previous individual lab assignments. The participation and discussion within the group discussion boards will be part of the overall lab assignment grade for each student. The lab report will be another part of the lab grade and the final part will be the behavior of the assignment and meeting the assignment specifications.

This lab will be an extension to lab 7 by utilizing the same mono audio output and the PWM generator. The goal of this group assignment is to work together to design and implement Block RAM (BRAM) in the FPGA to store and play music.

Lab 7 implemented a PWM generator which utilized a 10-bit input to generate the PWM output. For this lab, that means the audio datafile input should be 10-bits in resolution and, to properly work from standard audio clips, should be processed at 44.1kHz. **NOTE**: Since Lab 7 is not due until a week after this project is assigned, only use a blackbox version of the PWM generator to avoid sharing solutions with other students. Work out a general entity to use for the PWM generator and assume a working component by week 2!

This lab will include two audio clips that are provided to play back through the audio jack, There will be two methods to load the files into a BRAM for playback. The first method is a default initialization of the BRAM utilizing a COE file, which is provided to the BRAM IP generator as the default state information for the BRAM. This will allow the BRAM to play back an audio clip right away from power-up and can be used to verify your playback.

The second method will be to load an audio clip through the USB-to-serial interface on the Nexy's 4 DDR. This lab will come with an entity to use for receiving a clip from the serial interface and a separate instruction on how to load the clip through the serial interface using a Windows application provided to you. This method will require writing an interface from the serial loader entity provided to the BRAM device.

The function of the board will be as follows:

- Switches(1:0): Define the number of playback cycles to make when center pushbutton is pressed.
- Center Pushbutton: This will start the playback of the contents in BRAM at 44.1kHz and assume 10-bit segments from the BRAM. The number of complete cycles through the BRAM will be indicated by the Switches(1:0)
- Up Pushbutton: This will start loading a new sequence of values into the BRAM from the serial interface. LED(0) will light up to indicate a loading operation is in progress and should automatically turn off after loading enough 10-bit segments to properly fill the BRAM. At that point LED(1) should be turned on to indicate completion of the loading.
- Switch(15): This will function as the AUD_SD pin as in Lab 7, if this switch is set to ‘0’ the PWM amplifier will be turned off and no audio will be heard.

Task 1

As a group, work out a top level block diagram that shows how a BRAM interface can be constructed to input into the PWM generator. This block diagram can be used within the group to break down tasks and determine who works each part.

Decide on an entity structure for the PWM component, which will be lifted from a working Lab 7, but **do not send anyone in the group this component until after the submission of Lab 7 for each student.** For the first week just reference this component as an entity with inputs and outputs, but no architecture.

Task 2

Design and build the BRAM component to support the music length requirement. This component can be generated using similar techniques as the DDS generation in Lab 7. Make sure to read the BRAM datasheet for information on the component and settings that should be configured.

Add an initial COE file that is attached with this lab assignment to have a working point for playing audio. This initial COE contains 10-bit values that should be played back at 44.1kHz. Work out the surrounding process blocks needed to read the data out of the BRAM and into the PWM component.

Test this design to verify operation of the audio player and that you can hear the initial audio clip.

Task 3

The usb_music_serial entity has the following declaration:

```
    entity usb_music_serial is
      Port (  clk : in std_logic;
              reset : in std_logic;
              UART_TXD_IN : in std_logic;
              UART_RXD_OUT : out std_logic;
              load_music_sample : in std_logic;
              new_music_data : out std_logic;
              music_data : out std_logic_vector(9 downto 0)
            );
  end usb_music_serial;
```

Signal Definitions:

- clk: 100MHz master clock
- reset: Active-High system reset
- UART_TXD_IN: Bring out to top-level, this is data transmitted from the computer
- UART_RXD_OUT: Bring out to top-level, this is data received by the computer
- load_music_sample: A single clock cycle pulse will start detection for a single music_data sample
- new_music_data: A single clock cycle pulse will indicate new music_data sample is captured

- music_data: When new_music_data='1' this will contain valid new music_data sample received from the computer

The USB-to-Serial component should be used as follows:

1. Once BTN_U is pressed, turn on LED(0) and turn off LED(1). Send a single clock cycle pulse to load_music_sample to start looking for a 2-byte message from the serial port to start the music sampling.

2. Monitor the new_music_data, if the new_music_data signal pulses high for one clock cycle, load the music_data output into a temporary register, which is then loaded into BRAM.

3. Repeat step 2 by sending another load_music_sample request until BRAM is filled up for the desired music length of 264600 BRAM entries. After music length is reached, turn off LED(0) and turn on LED(1).

Using the knowledge of the usb_music_serial entity control signals, design an interface between this entity and the BRAM to load the incoming data onto the BRAM. This will allow playback of the music file when the center button is pressed after loading. Once loading is complete, turn off LED(0) and turn on LED(1) to indicate the new music file has been loaded. Keep this LED light on unless another load operation is performed.

To load a file, use the included application, 'ECE_Serial_Loader.exe' along with the new music file, 'serial_music.file.mser' which should be in the same directory as the executable. Launch the program and select the COM port for the Nexy's 4 DDR. Make sure the board is powered up and programed and the BTN_U has been pressed to light up LED(0). Press 'Send New Music File' and wait until all 529200 bytes (or 264600 music data samples) have been sent. This can take a while to fully send, so please be patient.

To Submit (ONE Submission per Group!)

Files to Submit (minimum), zipped together as Lab8_Group<groupnumber>.zip:

- VHD, XDC, BIT files for Lab 8
  - Make sure to include the BRAM Compiler files (VHD, DCP)
- Complete Synthesis Results
- Report summarizing the overall design principles and overall block diagram
  - Include a breakdown of work performed by each student and any comments on student performance (good or bad)
  - Note: If you want to leave anonymous student feedback, please send me an email to discuss anything related to other group members as soon as possible

Note: The Grading sheet on the next page provides a good overview of the key points of the design and what the instructor will look for when grading. Please note that this is a guide and isn't comprehensive. Please read over the lab assignment carefully to verify all the proper documents are submitted.

Grading Sheet

Lab 8/Module 12: BRAM Music Player

Lab	8
Student:	

Grade	Out of	Notes
	100	
	20	Quality of project files (VHDL, XDC) – Comments, clarity are important here
	40	Group Lab report submission and participation grade
	40	DEMO: Test of Group bit file submission on instructor's board (initial audio playback and audio loading through remote file)