ECE3829 Lab 4: Embedded Processors Lab

Required deliverables:

- Functionality demonstrated and signed-off.
- PDF of C code at time of sign-off.
- Exported Vitis project at time of sign-off.
- Lab report submitted to canvas by the deadline.
- Late reports must be in by 6:00 pm on Friday 03/04 as grading must be completed for final grades to be submitted on time.
- You must get signed-off to receive credit for the C code and report .

Important Material:

Additional Hardware for this Lab

- Digilent AMP2 PMOD
    - AMP2 PMOD should be installed in the top row of connector JB.

  https://reference.digilentinc.com/reference/pmod/pmodamp2/reference-manual
- Plug in a set of headphones or a speaker to the 3.5mm audio connector of the AMP2. AMP2 output is very load so if you are using a speaker, volume control is useful. If using headphones, do not put them in your ears.

Reminder:

- Remember to create a comment header to your C code files similar to that used for your Verilog for other labs. .
- The specific implementation details defined in the lab and commenting guidelines listed in the grading rubric at the end of the lab will factor into credit received. Follow the instructions.
- You must get your lab signed-off in order to receive credit for the C code and report portions of the lab. If you run out of time, you can still sign-off with what you have working for partial credit. The C you submit to canvas must be the code used during your sign-off.

All code should be written in C for this lab.

Overview

As a starting point for this lab a Vivado project file that includes a MicroBlaze soft processor is provided. In addition, a C-code application is provided as a starting point. Using the Vitis IDE an application is developed that outputs a frequency control digital signal to the AMP2 PMOD to create different musical notes. The application code performs the following functions.

1. Read the push buttons and dip switches on the Basys3 board.
2. Set the LED outputs and 7 segment display value.
3. Create a frequency controlled square wave output to the AMP2 Audio Amplifier to generated different musical notes.
4. Generate a short tune on start up.

Part 1: Get Starting Files working.

Two files have been provided that should be used as a starting point for your own work.

1. lab4_start.c : Starting point for the C application code. Feel free to modify any of the existing functions and code as you see fit.
2. lab4_start.zip : Starting point for the Vivado project files.
3. Lab4_SettingUpVitis.pdf gives instructions on how to use these files.

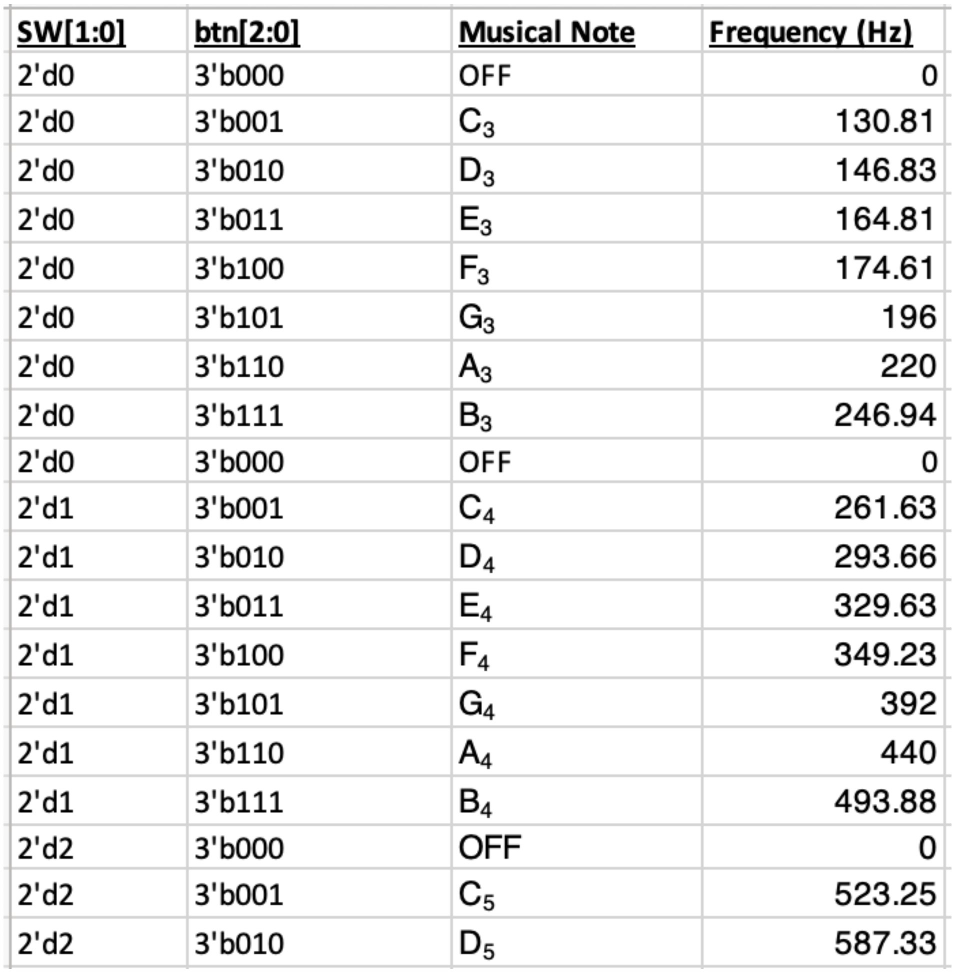
To use these starter files, you must have Vitis and Vivado 2021.1 Installed on your computer.
Refer to the Lab_GettingStarted.pdf file posted in Lab 1 if you need instructions on how to add Vitis or install Vitis and Vivado.
The starting files provide the below functionality

- Light the LED corresponding to its switch input.
- Output a continual tone at Middle C.

If these functions do not just work, seek assistance before going further.
As part of the sign-off the LED functionality must be demonstrated. So, leave the LED functionality in place.

Part 2: Create 16 different musical notes

For this part of the lab, read the value of the buttons and slider switches and change the toggle rate of the AMP2's Analog Input to create a musical note corresponding to the settings defined in the table below. The lab4_start.c shows how to use the Hardware Timer to generate a single middle C note. The target_count value needs to be calculated for each of the 16 notes.



SW[1:0]	btn[2:0]	Musical Note	Frequency (Hz)
2'd0	3'b000	OFF	0
2'd0	3'b001	C_3	130.81
2'd0	3'b010	D_3	146.83
2'd0	3'b011	E_3	164.81
2'd0	3'b100	F_3	174.61
2'd0	3'b101	G_3	196
2'd0	3'b110	A_3	220
2'd0	3'b111	B_3	246.94
2'd0	3'b000	OFF	0
2'd1	3'b001	C_4	261.63
2'd1	3'b010	D_4	293.66
2'd1	3'b011	E_4	329.63
2'd1	3'b100	F_4	349.23
2'd1	3'b101	G_4	392
2'd1	3'b110	A_4	440
2'd1	3'b111	B_4	493.88
2'd2	3'b000	OFF	0
2'd2	3'b001	C_5	523.25
2'd2	3'b010	D_5	587.33

Part 3: Show the note information on the 7-segment display

The 7-segment display is used to show the musical note information. On display D (right most display) show the letter of the corresponding musical note. For example, for btn = 3’111, the display would read b.

Part 4: Power-up play a little tune

When the Basys3 board first powers up or after a reset, it should play a little wake up tune with at least 4 notes and long enough for each note to be heard.

Extra Credit

Up to 5 points lab bonus points for any good improvements, enhancements to your design. You must demo on board and describe the additional functionality well in your report. For you could use buttons to cause distortion affects or play additional tunes. You could expand the usage of the unused 7-segment displays (taking care not to interfere with the audio quality).

Sign-Off Procedure

Signoff can be done remotely via zoom or in-person during lab hours. If via zoom, you will need to have the ability to share your screen with the TA and also have video and audio capability to display and hear the Basys-3 board functions clearly.

You may repeat steps 1-4 as many times as needed. But you may only submit your exported Vitis project and pdf of your C code once during your final demonstration. No changes to these submissions are allowed after your final demonstration.

1. Have project fully compiled and be ready to show to the TA.
2. Using File->Export select System Projects to generate an archive of your C project and have it ready to submit.
3. Have all your C code in a single pdf file ready to submit.
4. Demonstrate the functionality in the Functionality Check List. TA will record results and assign points for all functionality that is correct.
    a. You can demo as many times as needed.
    b. You will always be rewarded points for the functions that are working.
5. Once you are happy with your demonstration results.
    a. You will submit the single pdf version in canvas immediately.
    b. You will submit the exported zip in canvas immediately.
    c. TA will verify they are submitted and complete the sign-off process.

Sign-Off Check List (60 points)

Project Status (6 points)

1. Expected files are in place in the Vitis software project. (2 points)
2. FPGA Hardware Programs successfully (2 points)
3. Elf file compiles and runs. (2 points)

Functionality Check List (50 points)

1. LED Operation (10 points)
   - When a switch is pushed up, the corresponding LED turns on. When switch is pushed down LED turns off.
2. If no buttons are pushed, no sound come out and displays are off. (5 points)
   - When no buttons are pushed, no sound comes out and not lights are on.
3. Creates 7 musical notes corresponding to button values in table. (10 points)
   - Test each of the 7 notes that correspond to button values 1 through 7.
4. Show the letter of the note on display D (right most display). (10 points)
   - Test each of the 7 notes that correspond to button values 1 through 7.
5. Switches[1:0] changes the note octave. 0 = octave 3, 1 = octave 4, 2 = octave 5. (5 points)
   - Test sw[1:0] values 0 through 2. Verify notes increase in octave. 0 = octave 3, 1 = octave 4, 2 = octave 5.
6. Play a startup tune. (10 points)
   - When first powered on, play a tune with at least 4 notes.

Extra Credit Demonstrated (up to 5 points)

- Extra credit must be demonstrated at time of sign-off and documented in report to receive credit.

Canvas Submissions Completed (4 points)

1. Exported System Project zip file submitted in canvas at time of sign-off
2. PDF of C code files submitted in canvas at time of sign-off

Rubric for C Code in PDF (20 points)

1. Code implements the requested functions. (10 points)

- Code is composed of initialization functions executed once at the beginning of the programmed followed by an infinite polling loop.
- The code in the polling loop is non-blocking and the polling loop delay is minimized.
- The count value of the timer module is used to determine the toggle rate of the AMP2 Audio input as well as the refresh rates of the 7-segment display.

2. Code is well commented (5 points)

3. Code is well formatted, organized and written. (5 points)

- The code should be well formatted with proper indentations and reasonable line lengths.
- Code is readable and organized so it is clear what functionality is being implemented.
- Functions are used when appropriate.
- Variable names are short but descriptive.
- Define statements are used to define constants.

Rubric for Report

Brief Introduction / Problem Statement (5 points)

General Overview of solution (5 points)

- Write a brief description of your C code including a flow diagram.
- If you implemented extra credit, include a description of the extra features.

Detailed Implementation Description (5 points)

- Write a detailed description of how each piece of required functionality is implemented and which hardware blocks were used.
- Detail any efforts taken to minimize the loop delay of the infinite polling loop.

Conclusion (5 points)

- For functions like the 7-segment display, reading switches, writing LED values, compare your designer experience.
  - Was implementation in C or Verilog easier?
  - Where there advantages or disadvantages to the different approaches?
- What problems or challenges were faced in implementation?
- What solutions were used to solve them?
- What lessons were learned from the project?
- Suggestions for further improvements and extensions?
- Where there particular things you liked about the lab?