**The American University in Cairo**

**School of Sciences and Engineering**

**CSCE 230/2301: Digital Design I**

**Spring 2016**

**Project 2 report**

**Submitted to: Dr. Mohamad Shalan**

**Submitted By: Aley Eldin Baracat, John Sourour, Fadi Zaki**

**ID No.: 900140290, 900140842, 900140212**

Project Description:

* Electronic keyboard with 5 buttons mapped to 5 different sounds.
* Our design was implemented using the Nexys 3 FPGA board (Spartan 6 - Power Unleashed) and i2s Digilent Pmod module.
* To generate specific clocks for the i2s module as well as the sounds that we want to generate (Serial Clock=1.4Mhz, Left-Right Clock=44.1KHz, Master Clock = 11.2896 Mhz), we used the ipcore clock generator.
* 2 modes of operation according to a switch on the FPGA (P and B modes).
* For the first mode, the B mode: it generates a Serial input that toggles between two levels (0 and a high number) to create a square wave which is then modified to create 5 different notes with 5 different frequencies.
* For the second mode, the P mode, we used the block memory RAM of the Nexys2 board to store the Serial Data for a Piano note. The block memoy RAM is generated using ipcoregen which gets the file width and depth and reads the data from a .COE file. The .COE file is obtained by extracting a part of a WAV file containing the note and formatting it according to the .COE syntax. Then, the memory reads the file from start to finish in a loop and outputs the Serial Data.
* In both modes, we generate a set of frequencies to create a different note for each button from the same serial input. And we use a left shifter to get the MSB of the serial input with each falling edge of the serial clock.
* To play the piano, the user can switch between the two modes using the T10 switch and then he can play up to two notes simultaneously. In case of two buttons pressed at the same time, the FPGA is designed to create a mixed frequency between the two that outputs the average frequency between the two.
* For each note, the note name is displayed on the first 2 BCD 7 segments and in case of a mix the LED is turned on and the board displays a box in the first two BCDs. The board also shows the mode in the last BCD.

Complications

* When implementing the FPGA so that the output is generated across the two channels (Left and Right), there seems to be a Setup Time violation because of the delay caused by the LSB of the previous channel; Only 1 Channel is used and therefore using headphones will result in only ne of the two to be working while the other might have some noise due to the LSB of the first channel. However, if a speaker is connected, the noise will be much higher and much noticeable.

* Although the B mode worked somewhat smoothly, P mode has had different complications in terms of memory capacity available for this specific FPA board which is 64 KBytes. That’s why only one note (62 KB) was loaded into the block RAM of the Nexys2 board and then we used the same note and created a set of notes by the different frequencies generated. Also, we had to reduce the number of samples to try and fit in the 64 Kbytes and that has resulted in a noise alongside the output data (piano note) because the sample size is not big enough for the output serial to be generated in a better quality.
* To convert WAV to COE, we used MatLab to generate raw binary data from a sound file (WAV to txt) and then we reduced the sample size and adjusted it to fit the COE syntax.
* In terms of the notes’ volume, by trial and error, we have discovered the level1 value (in B mode) to use for the best sound quality.



