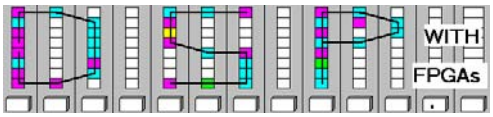
Lastname: _____

5Digit SS:_____

LABORATORY
DSP with DE2



LAB : DSP WITH DE2
(10 points)

In this lab you will be introduced to the Digital Signal processing design flow for the DE2 Cyclone II University Development board from Altera.
In the **pre-lab** you will compute with "pencil-and-paper" the results you later expect in your design implementation. In the **design part** you will complete the design of a sine wave generator and become comfortable with the DE2 environment.

Lab Objectives

After completing this lab you should be able to

- Understand the Quartus DSP on DE2 design flow
- Associate VHDL components with their functions
- Design and simulate a circuit using Quartus II

Pre-lab (3 points)

1. Download the DE2 User Manual documentation for the Cyclone II board from Altera's webpage or the course webpage and answer the following questions.

   a. What is the exact name of the FPGA on the board: _____

   b. How many logic elements (LEs) has the FPGA? _____

   c. How many embedded 18x18 multipliers are in the FPGA? _____

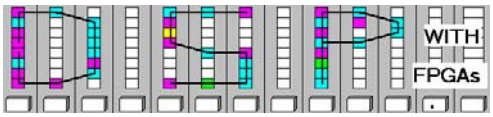
   d. How many M4K embedded RAMs does the FPGA has? _____

   e. What is the frequency of the FPGA on-board oscillators? _____

   f. Determine the pin locations for ledr[0] =_____
   Global Reset (sw[0])= _____
   and Oscillator 50 MHz = _____

2. For the following components briefly describe the function (See also PPT lecture notes on Logic Review)

Component	Brief Description
`audio_pll`	
`i2c_config`	
`audio_dac`	
`seg7_lut`	



3. A function generator uses a LUT to store a sine table. The input to the function generator is a triangular signal, and at the output an unsigned sine wave should be observed. The frequency of the sine wave should be 12 kHz . The triangle function is generated with an accumulator (increment *M*) running at 50 MHz and *B*=15 bit width. Input and output frequency are related via $F_{out}=F_{in}*M/2^B$.

   g. Determine the period length of the 12 kHz sine wave: T = _____

   h. Determine accumulator increment *M* (rounded to nearest integer) for the 15 bit accumulator with an output frequency of 12 kHz: M = _____

   i. Determine the frequency of the sine wave for M=1200 for the 15 bit accumulator

      F = _____

4. The sine frequency is in general much too high to be seen on the 7-segment LEDs. The 8 MSBs of an 32 bit accumulator are used for the LED display.

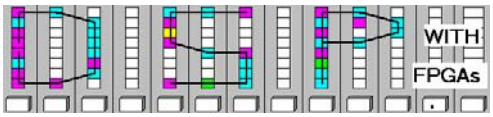
   g. Determine the 7-segment LED period length for *M*=1 and 32 bit accumulator:

      T = _____

Lastname: _____

5Digit SS:_____

LABORATORY
DSP with DE2



VHDL Design-lab

Follow the directions below to implement sine wave generator circuit.

A. Getting Started

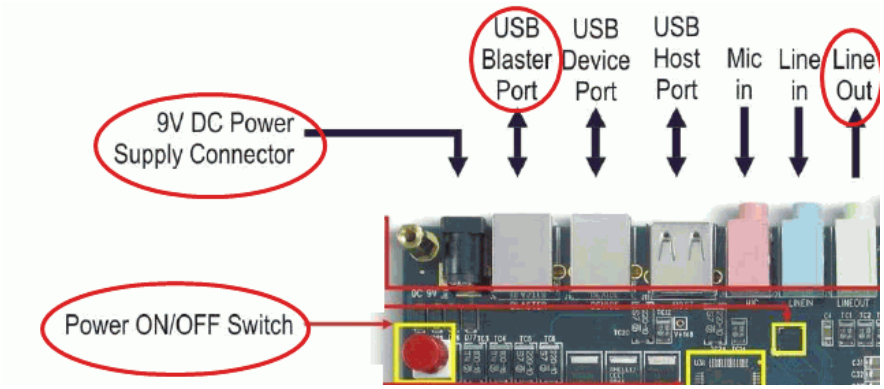
If you are in B114 or the digital logic lab:

1. On the desktop double click on **Engineering folder**.
2. From the top icon list on the Desktop click on the **Quartus II** icon 

to start **Quartus.**
3. You should not save anything on the local hard disk. You will have to use a memory stick, or your "mapped" home directory to save the files. Create a New Folder named **DSPwFPGAs** on your mapped network drive or memory stick.

B. Power-up the DE2 Board

1) Connect the power supply to the DE2. Connect PC and board via USB cable. Both connector are in the upper left corner of the DE2 board:

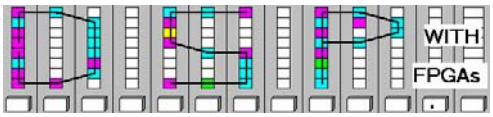


Make sure you use the **USB Blaster** and NOT the **USB Device Port**.

2) Connect board and Oscilloscope with the BNC adaptor and BNC cable. Use the audio port called **Line Out**. In case your station has no oscilloscope, connect your head set to the **Line Out** port.

3) Now turn on the power of the board by pressing the red **Power Switch** down.

4) At this point you should see the LEDs flashing, the LCD shows
line 1: **`Welcome to the`**
line 2: **`Altera DE2 Board`**



5) Use the SW(17) switch to turn on the sine wave sound on/off. Turn on the Agilent oscilloscope and press the **Auto-Scale** button. You should see the sine wave in the oscilloscope display. Now use the **Quick Meas** button to determine the frequency of the sine wave:

F= _____ KHz.

The data can also be found in the DE2 User manual (page 10). In case you have a headset you can also listen to the sine wave.

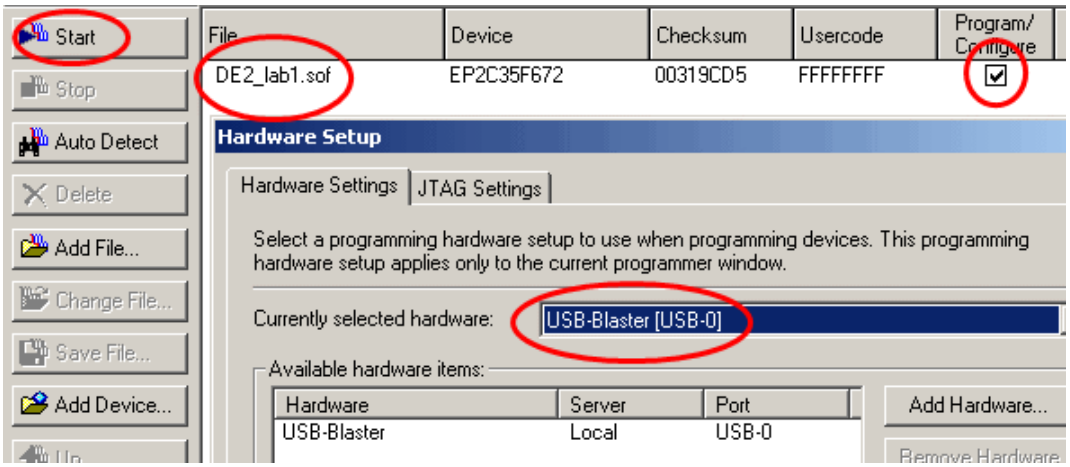
C. Compiling the "Lights ON" Existing Design

1) Download the file `DE2_lab1.zip` from the class webpage and put the file in the **DSPwFPGAs** folder. Unzip the file and you should see the following files: `DE2_lab1.vhd, DE2_lab1.qsf, DE2_lab1.qpf, audio_pll.vhd, audio_pll.qip, audio_pll.cmp, sine.mif, i2c_config.vhd, i2c_controller.vhd, audio_dac.vhd, and seg7_LUT.vhd.`

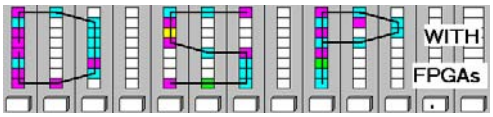
2) Use **File->OpenProject->**`DE2_lab1.qpf` to open the project file in **Quartus II**. Load the top level file `DE2_lab1.vhd` in **Quartus II**. (i.e., **File-> Open->**`DEC_lab1.vhd`). You should see after a moment the "light on" design. Note that as is the DE2_light example from the lecture notes only LEDs, switches and push button are used. Complete the Header with your name and date.

3) Compile this design in **Quartus II** by either pressing the  icon, hitting **Ctrl+L**, or select **Processing->Start Compilation**.

4) To program the device, start the programmer via the  icon or elect **Tools->Programmer**. The Programmer window pops up. Select **Hardware Setup->USB-Blaster**. In the programmer window make sure the **Program/Configure** option is selected and the `DE2_lab1.sof` file is selected:

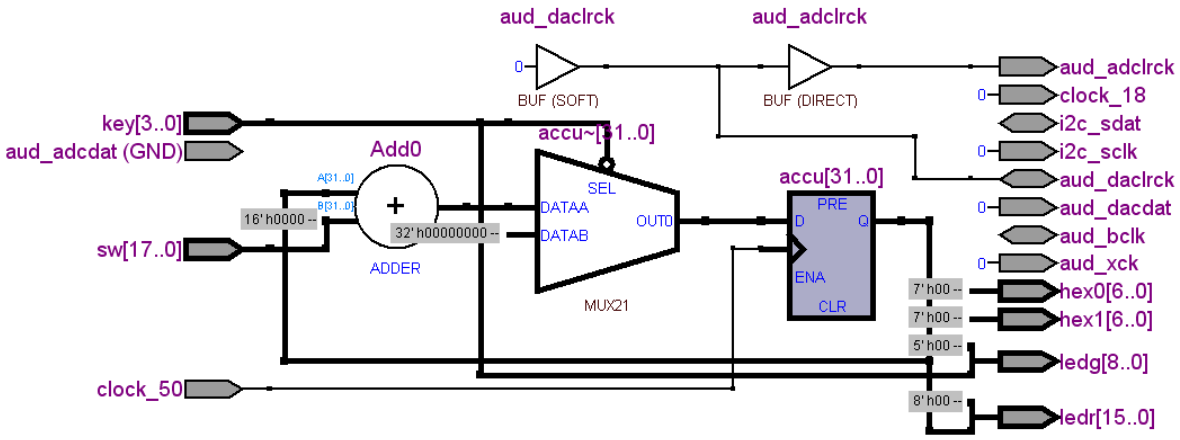


5) Now press the **Start** button in the programmer window and after a moment you should be able to try the switches and button to light up the LEDs.



D. Completing the Accumulator Design

1) Open the `DE2_lab1.vhd` file in **Quartus II**. Develop the code for the accumulator (see lecture notes "VHDL review" for help). `Key(0)` should be used as synchronous reset (active low) and the sw[17..0] should be used as accumulator increment. The register clock is the 50 MHz signal `clock_50`. Select the 8 bits in the range (14..7) as `accu_msbs` signal to be connected to the red led bits [7..0]. Use **Ctrl+K**, for the syntax check and **Tools->Netlist Viewers->RTL Viewer** to display the following figure.



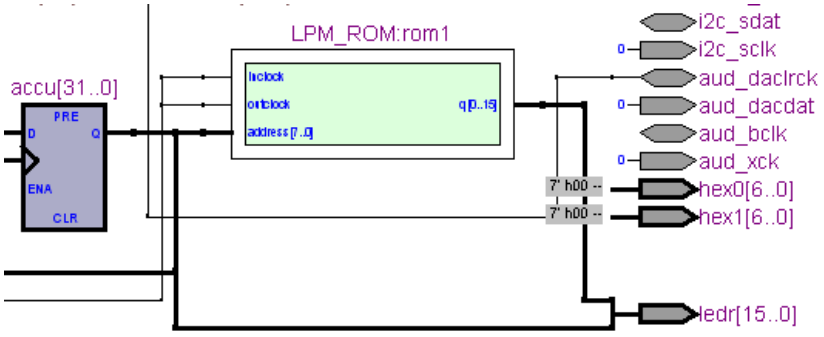
2) Now add the sine look-up table. Instantiate the `lpm_rom` component and use the following **GENERIC MAP** data:

LPM_WIDTH => 16	LPM_WIDTHAD => 8	LPM_FILE => "sine.mif"

Then implement the **PORT MAP** as follows

address => accu_msbs	inclock => clock_50	outclock => clock_50	q => lut

Finally connect the 8 MSBs of `lut[15..8]` to the red led bits [15..8]. Use **Ctrl+K**, for the syntax check and **Tools->Netlist Viewers->RTL Viewer** to display the following figure.

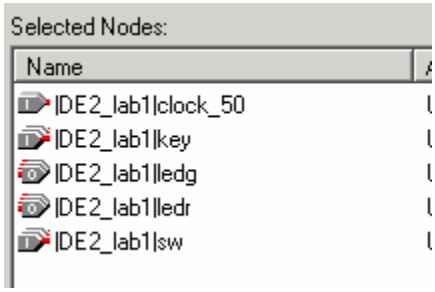


E. Simulating your Accumulator design in Quartus

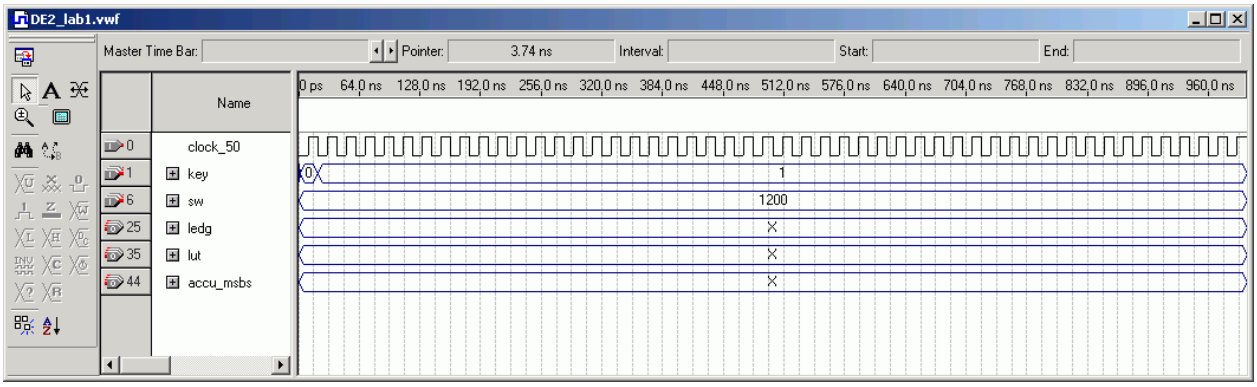
1) To run a simulation follow the usual procedure you used in the FPLD Quartus labs to do a simulation. As a reminder this step are explained in the following.

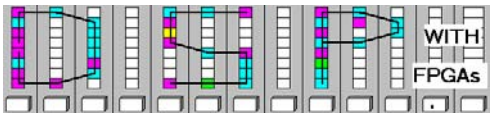
2) To open the simulator got to **Processing->Simulator Tool**. The **Simulator Too**l window will pop up. Now select open 

in the lower right corner. A window called `Waveform1.vwf` will pop up. Now double click in the white section under Name to open the **Insert Node or Bus** window. Click on the **Node Finder...** button. In the **Node Finder** use as **Filter: Pins: all** then press the **List** button. Now select the following nodes.

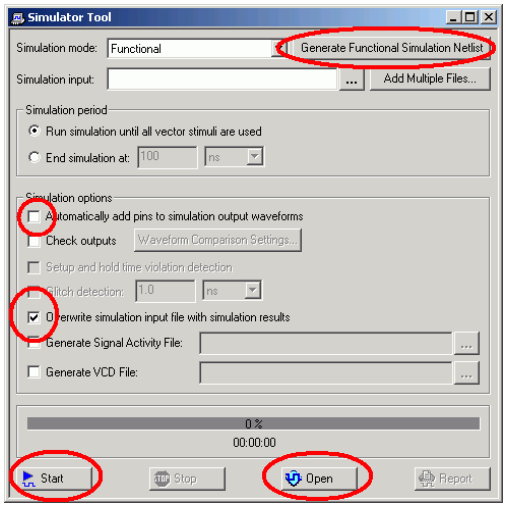


Using **CTRL+Mouse** left click to select in the **Nodes Found** section these signal and then use the  button to transfer these nodes to the **Selected Nodes** window. Then click ok and these signal should now be seen in the `Waveform1.vwf` window. Select `clock _50` with the mouse and then use **Overwrite Clock** button to set a 20 ns **Time Period.** Next select node `key` and set it using the **Arbitrary Value** icon  to value 1. Then `key` for the first 20 ns to zero. To do so select the first 20 ns of the clock_50 wave with the mouse and then use the **Forcing Low (0)** icon  . Using the **Arbitrary Value** icon  set signal `sw` to 1200. Now right click on the `ledr` signal and use **Grouping->Ungroup.** Delete `ledr[17]` and `ledr[16]`. Select `ledr[15:8]` with the mouse and then right click this area and select **Grouping-> Group**. As name choose `lut`. Do the same with `ledr[7:0]` and call this bus `accu_msbs.`
Set the **Radix** for `clock_50` to **Binary**; `key`, `sw`, and `ledg` to **Unsigned Decimal** and the node `lut` to **Signed Decimal**. Double click the signal name to chance the **Radix**.
Save the waveform file in the project directory under the name `DE2_lab1.vwf`. Now use the **View->Fit in Window** to see the full 1000 ns simulation. You wave form file should looks as follows:

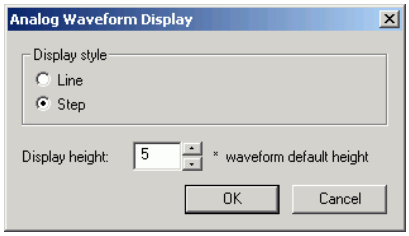




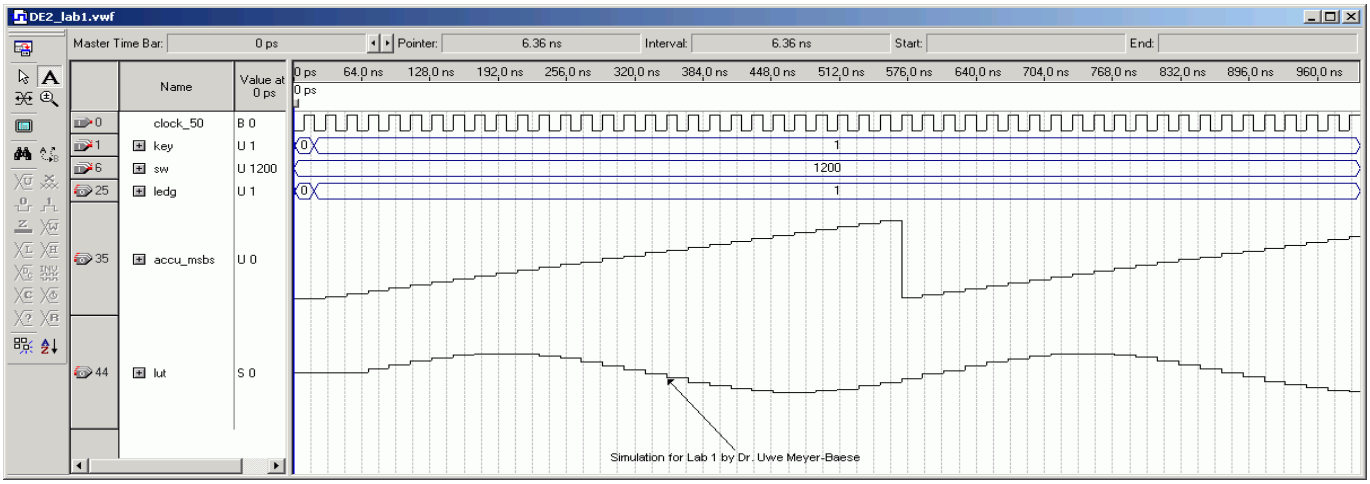
3) Now go back to the **Simulator Tool** window. Run the **Generate Functional Simulation Netlist**, select **Overwrite simulation input file with simulation results** and deselect **Automatically add pins to simulation output waveforms** as shown below.



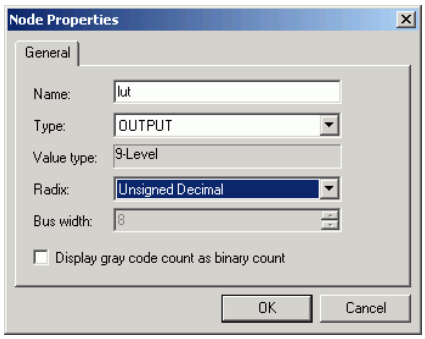
4) Finally hit the **Start** button to run the simulation. Answer with **Yes** to overwrite the current simulation results. To see the waveform in analog form right click on `accu_msbs` and select **Display Format->Analog Waveform**. Use the **Step** style and as height enter 5.



Do the same for the lut signal. Finally use the text button to add a signature with a text like "Simulation for lab1 by YOUR_NAME" You should finally see the following simulation results.



5) In case the waveform do not looks as expected make sure use have **Signed Decimal** for `lut` under **Node Properties** and **Unsigned Decimal** for the `accu_msbs`. Double click the signals to chance the **Radix**



F. Completing the lab1 Design

6) Open the `DE2_lab1.vhd` file in **Quartus II**. Now instantiate the remaining components C1-C5 as follows:

```
C1: audio_pll PORT MAP(areset=> NOT key(0), inclk0 => clock_50,
                       c0 => clock_18, locked => locked);

C2: i2c_config PORT MAP(iclk => clock_50, irst_n => key(0), --  Host Side
       i2c_sclk => i2c_sclk, i2c_sdat => i2c_sdat); --  I2C Side

C3: audio_dac PORT MAP( --  Audio control incoming
                  idata16 => dac, iclk_18_4 => clock_18, irst_n => key(0),
    --  Audio side outgoing
    oaud_bck => aud_bclk, oaud_data => aud_dacdat, oaud_lrck => daclrck);

C4: seg7_lut PORT MAP(idig => accu(31 DOWNTO 28), oseg => hex1);
C5: seg7_lut PORT MAP(idig => accu(27 DOWNTO 24), oseg => hex0);
```

The clock and the left/right channel select signals should also be connected as follows:

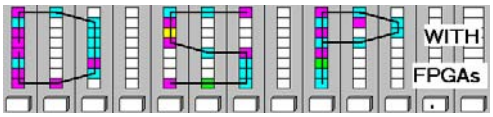
```
aud_xck     <= clock_18;     -- 18 MHz clock for audio CODEC
aud_daclrck <= daclrck;      -- Connect left/right signal to CODEC
aud_adclrck <= aud_daclrck; -- Use same left/right signal for DAC and ADC
```

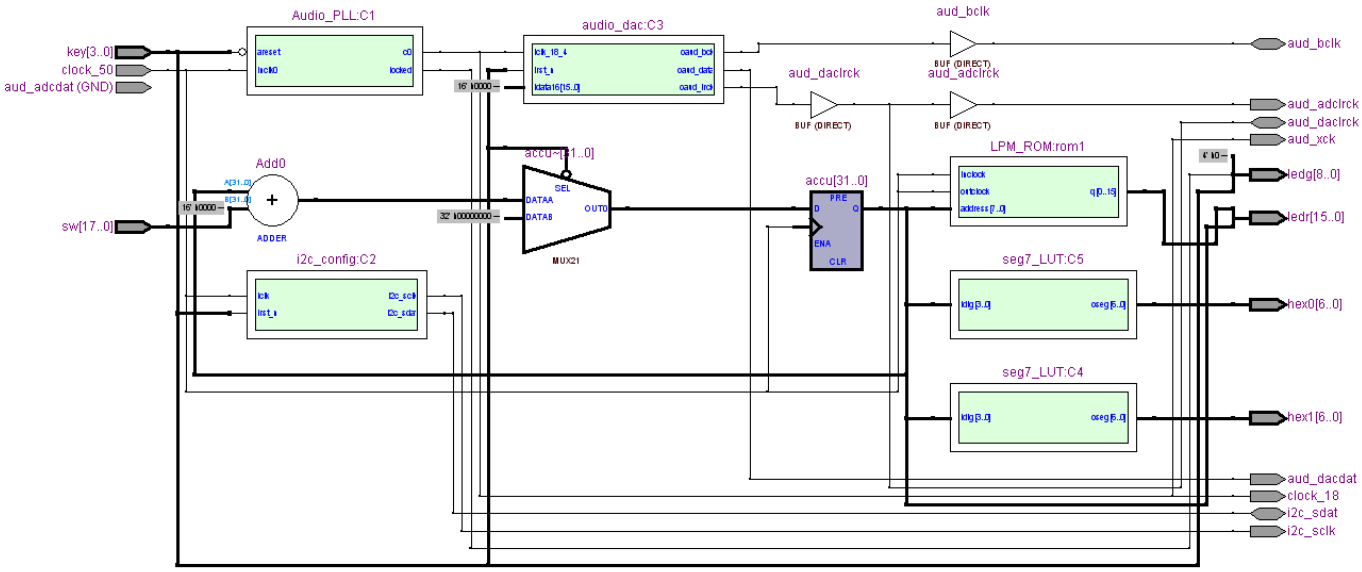
Use **Ctrl+K**, f for the syntax check and **Tools->Netlist Viewers->RTL Viewer** to display the following figure

Lastname: _____

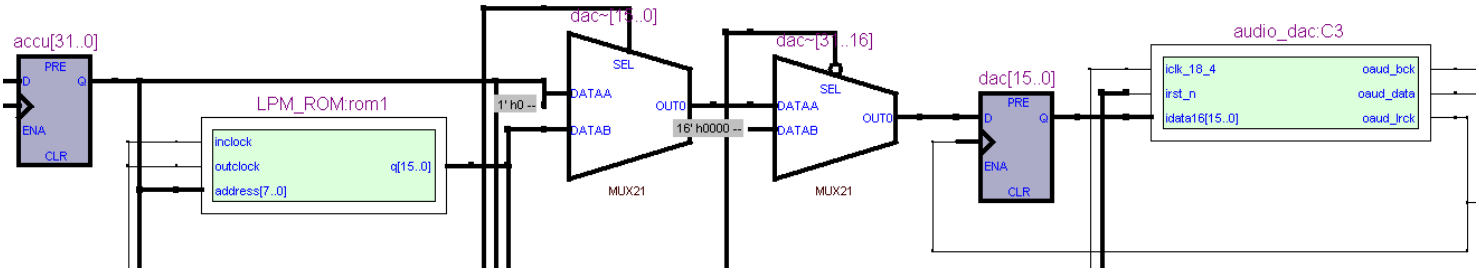
5Digit SS:_____

LABORATORY
DSP with DE2





7) The last element to be coded is the output multiplexer that allows displaying the sine and the accumulator output using the same DAC. Place this circuit between the sine ROM/accumulator and the DAC. Use a register running at the DAC frequency (DACLRCK) and controlled by `key(1)`. By default, i.e. key(1)='1', the DAC should display the `lut` value. When pressed, i.e. key(1)='0' then accumulator bits [14..0] should be displayed. Augment these 15 bits by a '0' to the left. Use **Ctrl+K**, for the syntax check and **Tools-> Netlist Viewers->RTL Viewer** to display the following figure.



6) Now run a full compilation, by either pressing the  icon, selecting **Processing->Start Compilation,** or hitting **Ctrl+L**,. From the report file and the **Classic Timing Analyzer Tool** (**Processing** menu) determine

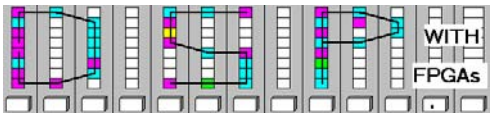
Total logic elements = _____

Embedded Multiplier 9-bit elements = _____

Total memory bits = _____

Registered Performance = _____ MHz

Enter these resource and performance data also in your VHDL header file.



G. Download your design to the FPGA board

Once you have completed and compiling your model, you can now download it to the board. Follow the board programming step as in section C on Compiling the "Lights ON" Existing Design.

H. Observing the Board

1. The switches form an eight bit input, which corresponds to the value *M* calculated for the prelab. Observe how changing the value of the input affects the speed of the seven segment display.

2. Turn on the oscilloscope. One channel will display the sine wave through the digital to analog converter. Pressing key(1) will display the output of the accumulator sliced to it's most significant bits and put through the digital to analog converter.

3. Determine the period length and frequency ($F=M/2^{15}*F_{in}$) of the sine wave for:

   SW=1 T = _____µsec F=_____KHz

   SW=3 T=_____µsec F=_____KHz

   SW=7 T=_____µsec F=_____KHz

4. Using the oscilloscope determine the maximum SW value that still a good quality sine waveform is shown on the oscilloscope:

   SW= _____ F=_____KHz

5. Using your head set determine the maximum SW value that you can hear the sine wave:

   SW= _____ F=_____KHz

6. Show your working FPGA board design to the lab instructor.

I. Deliverables

1) Solve the problems of the pre-lab. (3 points).

2) Complete this report, print the completed VHDL file `DE2_lab1.vhd`, including the resource data used and the Quartus II simulation plot with your electronic signature (7 points).

Make sure your name and SS is on all pages you turn in!

