

## Signoffs and Grade:

Name: \_\_\_\_\_

Component	Signoff	Date	Time
Mode 1 Microphone to Speaker ONLY			
Mode 2 Microphone -> LPF -> Speaker			
Mode 3 Microphone -> HPF -> Speaker			
Mode 4 Microphone -> Storage Key1 -> Speaker			
Mode 5 Microphone -> Student Concept -> Speaker			

=====

Component	Received	Possible
Signoffs		100
Penalties No signoffs after last day to turn in labs 5/5	-	
Total		100

**EDUCATIONAL OBJECTIVE:**

The educational objective of this lab is to experiment with the manipulation of real time audio data received from a microphone connected to the input of the audio CODEC on the DE1-SoC board. In this lab the Altera University audio core component will be used to input/output audio samples to/from the NIOS II.

**INTRODUCTION:**

The nios\_system developed in Lab 9, including the audio filter custom component, will be the hardware used for this lab. You may need to add a PIO for KEY1 and the LEDs. This system includes the audio core provided by Altera. Using the audio core datasheet, provided on MyCourses, you will write a program that operates as follows:

- It has five modes of operation, selectable with SW2-SW0
- In the first mode, input data samples from the CODEC are read and immediately output back to the CODEC
- In the second mode, input data samples from the CODEC are read and put through the low pass filter prior to being output to the CODEC
- In the third mode, input data samples from the CODEC are read and put through the high pass filter prior to being output to the CODEC
- In the fourth mode of operation, data samples from the CODEC are stored and not played back until KEY1 is pressed.
  - In this mode, the red LEDs should be on while recording is going on. When the record limit has been reached, the LEDs should go off, indicating to the user that they can press the KEY1 button for a playback
- The fifth mode of operation is left up to the student. Some examples are a delayed playback game, an echo, or a software filter.

Detailed instructions on how to write the program will not be provided. It is up to the designer to choose how to structure the program. You can choose to use the function calls provided on the datasheet or you can write your own. While writing the program, remember that the CODEC has been configured for a 48kHz sample rate and 16 bit data. If you use the SDRAM to store the recorded data, keep track of how much data you are storing so that you know when to quit recording (also so you know how much to play back). For testing purposes, you should set the storage limit fairly low.