Basic Structure:

Serial Terminal

Audio CODEC

picoBlaze

micro-controller

Storage

Memory

LCD

Display

User Controls

Figure 1: High-level block diagram of audio recorder/player

Microphone

& Speaker

Things to do:

\*\*\*Reset is active LOW, so use SW for reset. DON”T FORGET TO DO THIS!!!

1. Create a menu system in Verilog and output it to the terminal.
   1. Menu options shown below:
      1. Play a message
      2. Record a message
      3. Delete a message
      4. Delete all messages
      5. Volume control

Instead of that, put as many backspace characters as the previous message length. This leads to waaaaay less lines.

* 1. Figuring this out first will make creating menu options when in other sections easier. Most of the submenus will require a picoblaze subroutine.
  2. Setup logic for transitioning to other modules.
  3. Figure out a way to scroll through the menu.
     1. Find a way to transition between subroutines based on a BTN input.
     2. When the top of the menu is reached, scroll to “volume control”, and for the bottom scroll to “Play a message”
     3. Separate for each scroll (The entire menu will be CTRL+V CTRL+C with a dash added).

1. Create an FSM to handle where the user is in the menu options.
   1. FSM will use BTN’s as control signals for changing state (debounced of course)
   2. Two BTN’s used for scrolling through a menu, one for selecting a menu.
   3. One BTN for selecting an option, another for going back to the previous menu.
2. Create a memory interfacing module to use the already written modules given for interfacing the ddr2 RAM.
3. Create a microphone and speaker interfacing module to use the already written modules given for interfacing between mic’s and speakers.
4. Create a submenu module to output memory for options 1 and 3 of the menu.
   1. First figure out how to output memory
   2. Then figure out how to scroll and display current position in memory correctly
      1. Separate logic (in assembly?) for reaching the end of memory, and current position.
      2. Scrolling will involve button pushes, so debouncers are necessary.
5. Implement a “MEMORY FULL” message.
   1. Separate subroutines for scrolling through the menu with the mem-full message.
   2. If memory is full, do not allow user to record.
   3. Message should be implemented in menu system, simple check if memory is full then output message after the welcome message.
6. Implement pause/play and skip for messages.
   1. Switch to control pausing and playing
      1. When switch is up, pause.
      2. When switch is down, play
   2. Button for skipping (?)
   3. While a message is playing, user can interact with menu.
      1. Lost on this one, but we can figure it out later.
7. Implement recording of messages.
   1. Messages can only be 4 minutes long
      1. Since we’re working in nanoseconds, likely very big number to count seconds alone.
      2. Output message if max recording time is close to being reached(?)
      3. Stop recording once max time is reached.
   2. Can record a max of 6 messages
      1. Once this max is reached, go back to the menu, where the “MEMORY FULL” message will be output.
8. FINITO!!!!
9. LGSB