Pavitar Kalra
86229853

CPEN 311: Lab 2 Report

1. Lab can be found in zipped file >>lab2/template_de1soc/. In that file you will find all my modules and testbenches, as well as the .sof file. (simple_ipod_solution.sof)
2. All parts of the lab are working (a-f). No bonus portions were completed.
3. you will find my sketches of my state machine logic below, and below that you will find annotated simulations of all state machines.
    a. You can also find testbenches for other modules such as clk_sync and address_controller, in the zipped file.

State Machine Sketch

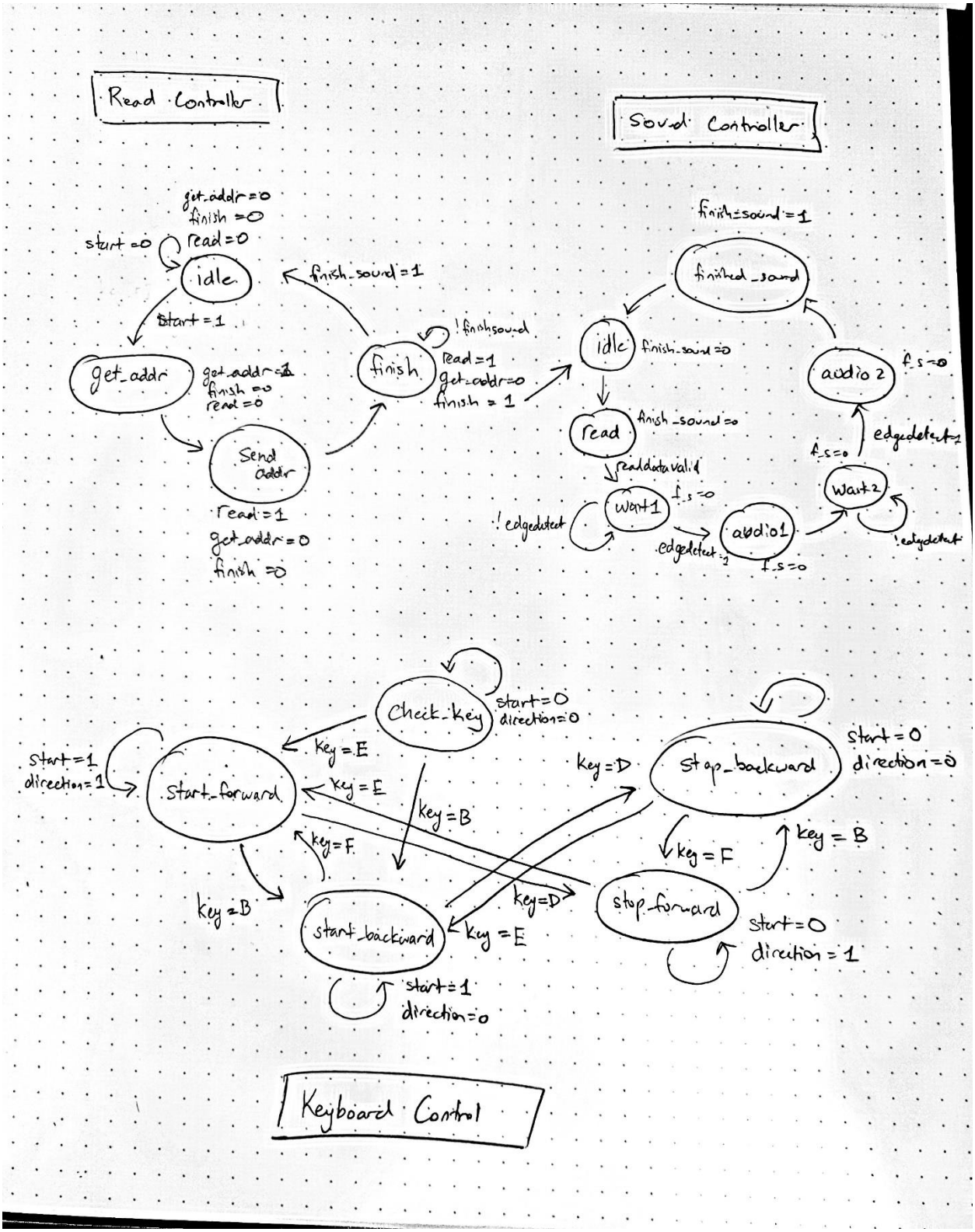
State Machine 1: Read Controller controls when an address is incremented and sent to the flash to be read.

State Machine 2: Sound Controller, controls the output of the audio data, and ensures it is at the right frequency and in sync with the 50Mhz clock.

State Machine 3: Keyboard Reader, manages inputs from the keyboard and updates start and direction flags appropriately.

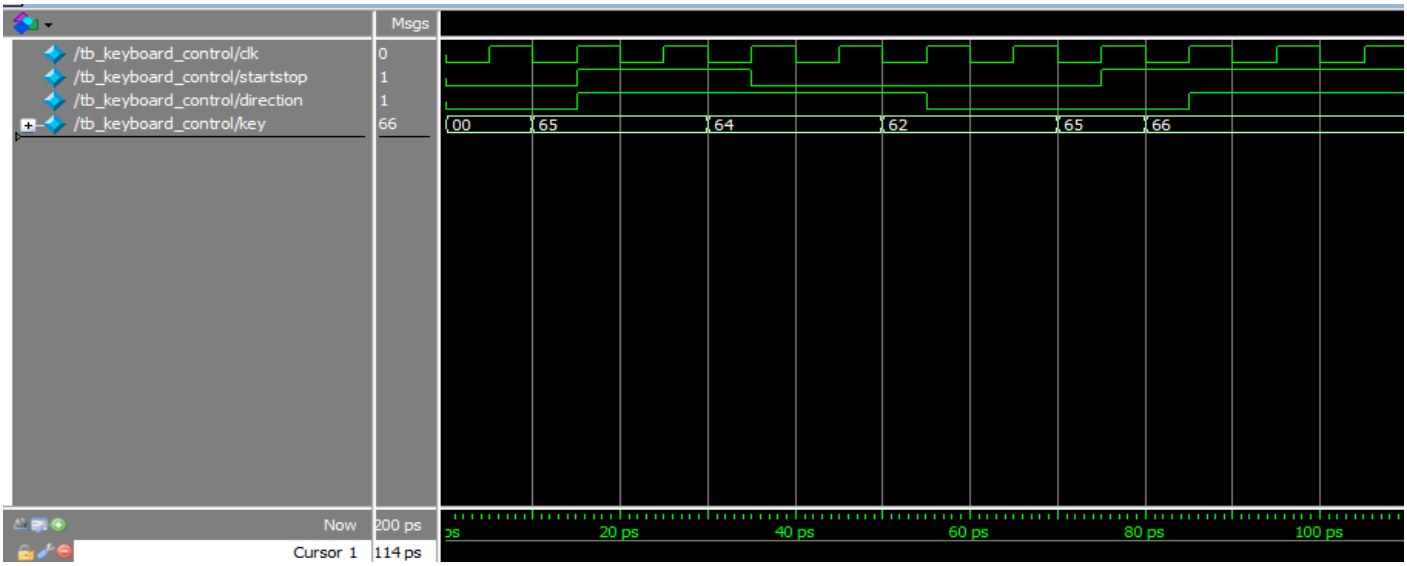
How they connect: Read controller starting in idle gets start signal from keyboard controller. Read controller then gets and address and sends it, then notifies sound controller. Sound controller then outputs the sound and sends a finish flag to the read controller which then returns to idle. The sketch of this process is on the next page:

Pavitar Kalra
86229853



Pavitar Kalra
86229853

Testbench Keyboard Control FSM: This test bench can be found in keyboard_control.sv



Annotations:

Section 1 (0ps - 10ps): No input start and direction are both low

Section 2(10ps - 30ps): start key is pressed and direction and start go high.

Section 3 (30ps -50ps): Stop key is pressed direction stays high but start goes low.

Section 4 (50ps - 70ps): Backward key is pressed, direction goes low and start stays low.

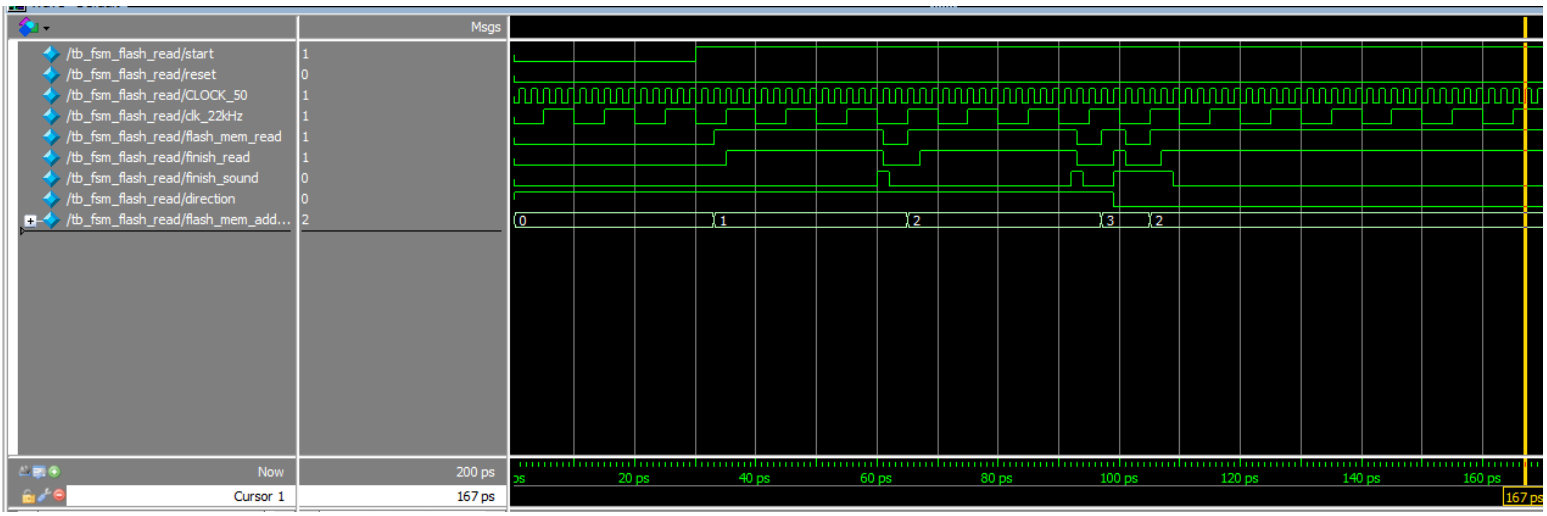
Section 5 (70ps - 80ps): Start is pressed and direction stays low while start goes high.

Section 6 (80ps - end): Forward key is pressed and direction goes high while start stays high.

All behaviour is as expected and as designed. And all signal changes occur on posedge of the clock as expected.

Pavitar Kalra
86229853

Testbench Flash Read FSM: This test bench can be found in fsm_flash_read.sv



Annotations:

Section 1 (0ps - 32ps): No input start address is 0 and doesn't change

Section 2(32ps - 60ps): start is initialized, and FSM starts read cycle. Address increments on posedge clock. Read signal is sent high until finish_sound is triggered high.

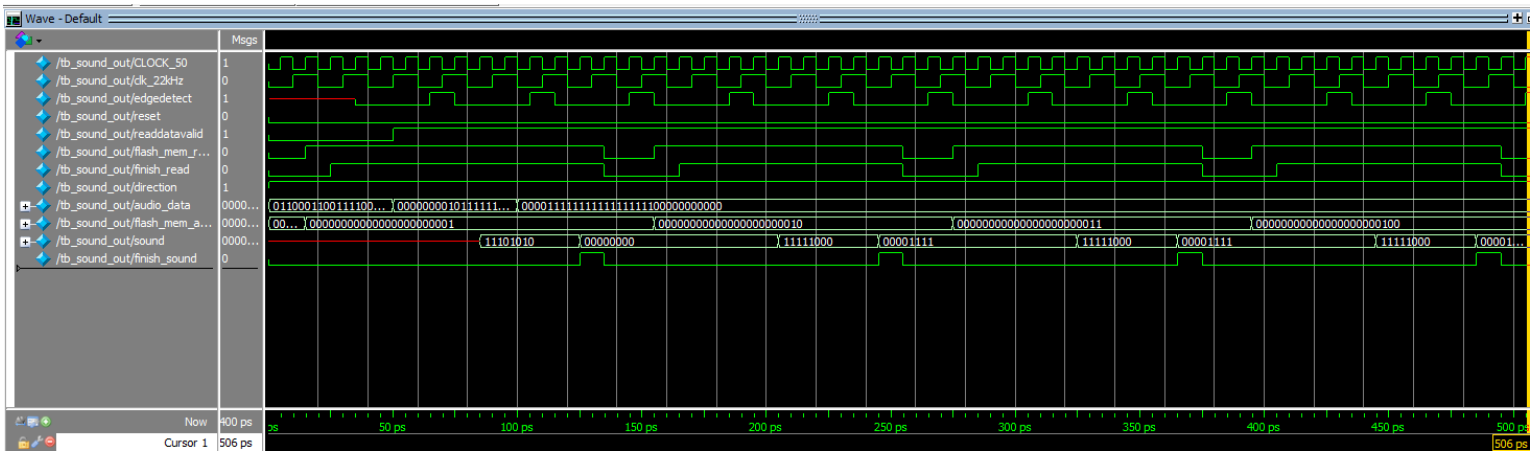
Section 3 (60ps -90ps): FSM continues as start is still high and finish_sound went low.

Section 4 (90ps - 110ps): FSM again continues as expected to next address.

Section 5 (110ps - end): Direction flag goes low so addresses start decreasing as FSM continues as normal.

Pavitar Kalra
86229853

Testbench Sound Out FSM: This test bench can be found in sound_out.sv. I am testing that the FSM works and have it connected to my read flash FSM to make sure their interface and timing works.



Annotations:

Section 1 (0ps - 50ps): Nothing happening, and sound is null.

Section 2(50ps - 100ps): Read data valid is raised at 50ps. This initiates this FSM as finish read high and start is initialized and sound out process begins. On edgedetect we see there is one clk cycle delay and then the audio start outputting. It outputs the appropriate bits as according to the direction.

Section 3 (100ps -125ps): Sound out process continues and then the sound finish flag goes up telling FSM read that it may finish.

Section 4 (125ps - 220ps): A set of audio data is sent and the FSM starts its process again. This time I allowed enough time for both notes of the read data to be outputted. Then the finish sound flag goes up.

Section 5 (220 - end): Again, sent new audio data and can observe the FSM working as usual.