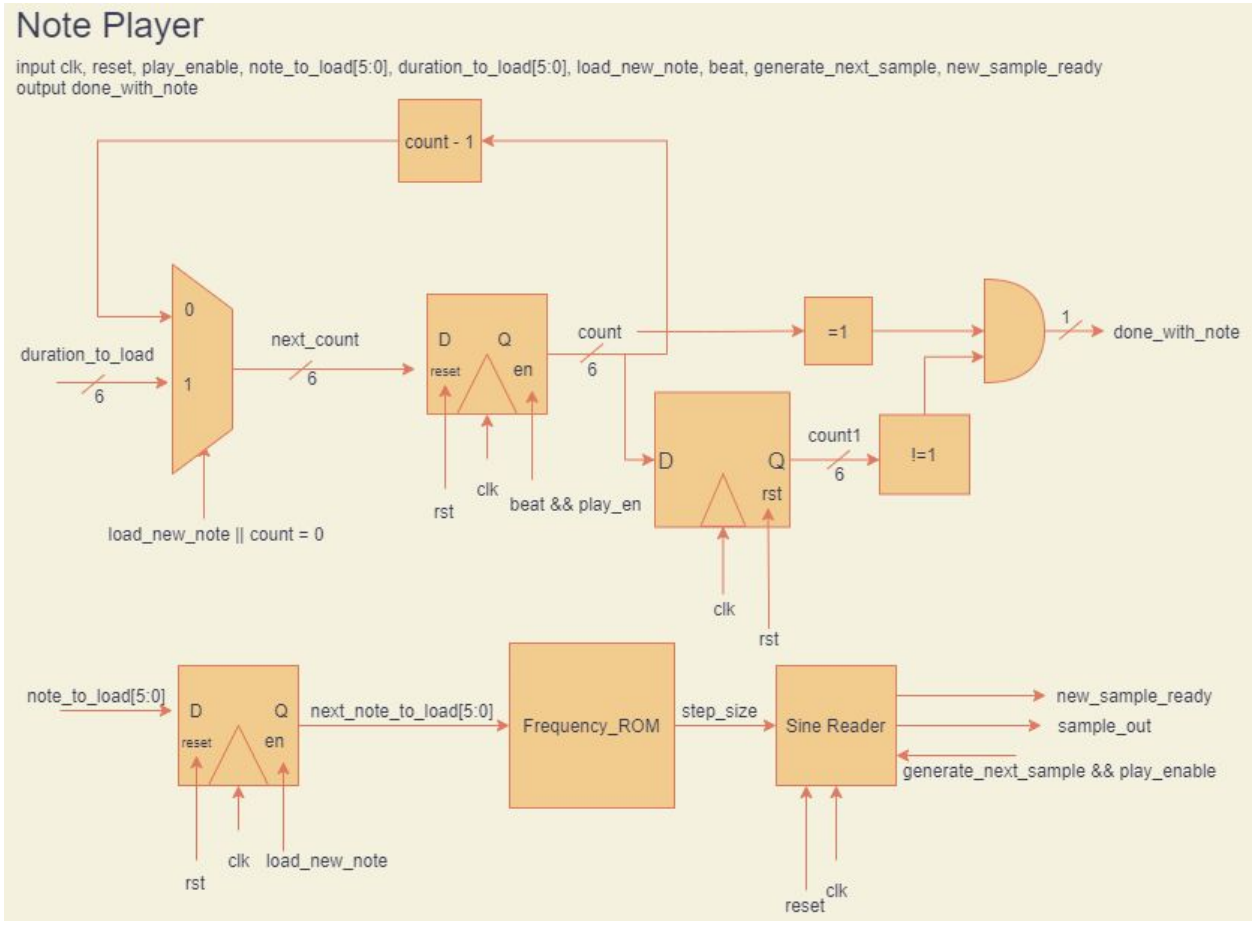
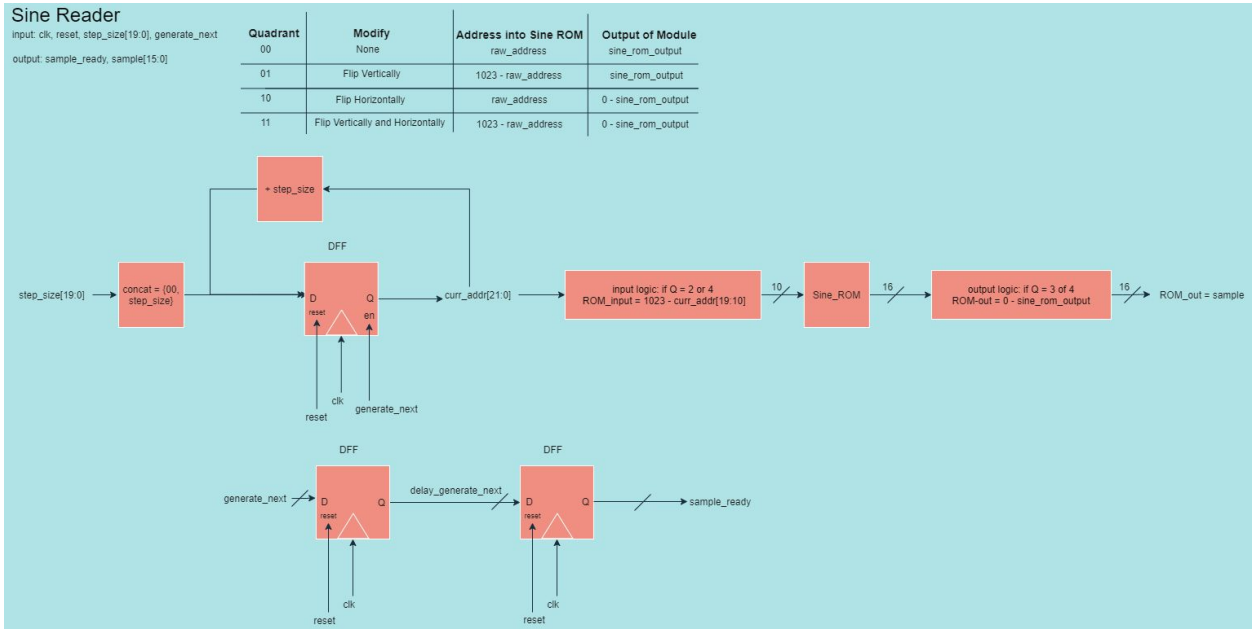
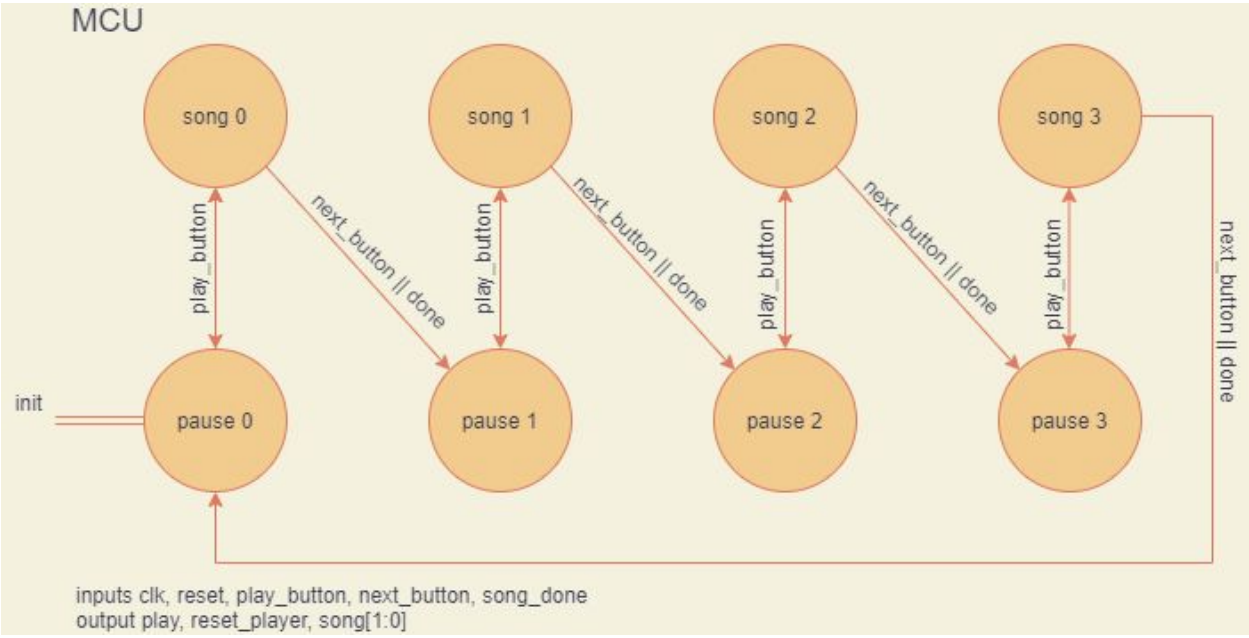
Lab 4 Diagrams

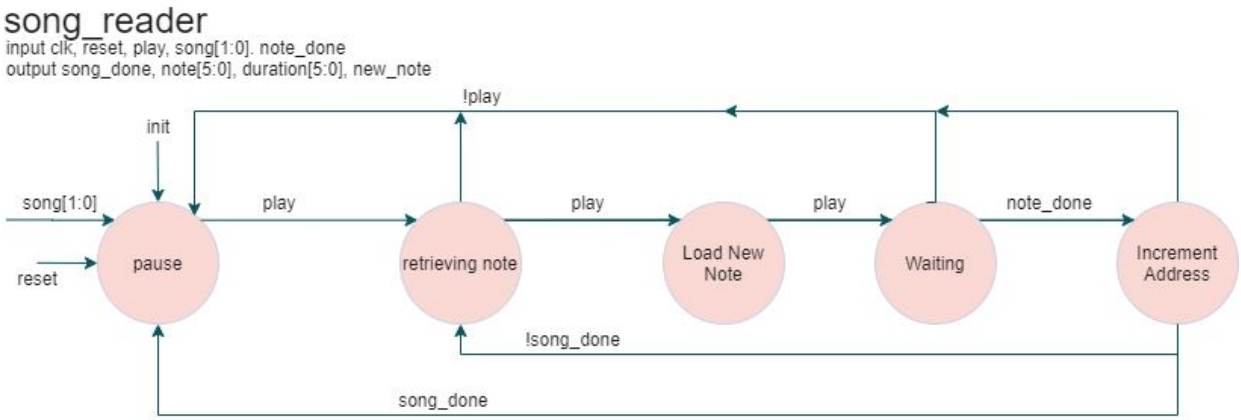
Block Diagrams





Finite State Machine Diagrams





Timing Diagrams

Song_ROM								
Cycle	**Play**	**State**	**note_done**	**song_done**	**next_address**	**current address**	**ROM output**	**new_note**
0	1	Pause	0	0	0, 0, 00000	XXXXXX	XXXXXX	0
1	1	retreiving_note	0	0	0, 0, 00000	0, 0, 000000	XXXXXX	0
2	1	load new note	0	0	0, 0, 00000	0, 0, 00000	{6,d49, 6'd12}	1
3	1	waiting	0	0	0, 0, 00000	0, 0, 00000	{6,d49, 6'd12}	0
4	1	waiting	1	0	0, 0, 00000	0, 0, 00000	{6,d49, 6'd12}	0
5	1	increment_address	0	0	0, 0, 00001	0, 0, 00000	{6,d49, 6'd12}	0
6	1	retreiving_note	0	0	0, 0, 00001	0, 0, 00001	{6,d49, 6'd12}	0
7	1	load new note	0	0	0, 0, 00001	0, 0, 00001	{6,d1, 6'd12}	1

*assume we are playing first note of first song

*assume the note just takes 1 cycle to finish playing

Note_Player							
Cycle	**play_enable**	**Load_new_note**	**Duration_to_load**	**Next_count**	**Count**	**Count1**	**done_with_note**
0	1	1	6'd6	6	0	0	0
1	1	0	6'd6	5	6	0	0
2	1	0	6'd6	4	5	6	0
3	1	0	6'd6	3	4	5	0
4	1	0	6'd6	2	3	4	0
5	1	0	6'd6	1	2	3	0
6	1	0	6'd6	0	1	2	1
7	1	0**	6'd6	6	0	1	0
8	1	1	6'd12	12	0	0	0
9	1	0	6'd12	11	12	0	0
10	0***	0	6'd12	11	12	0	0
11	1	0	6'd12	10	11	12	0
12	1	0	6'd12	9	10	11	1
13	1	0	6'd12	8	9	10	0

*first 13 cycles show the duration counter of the note_player module

**load_new_note flips 2 cycles after done_with_note because it takes 1 cycle to increment the address

and 1 cycle to load the new note
***when we turn the enable off, all the values in the counter should stay the same and stop decrementing

Cycle	State	Load_new_note	Note to load	Next_note to load	Freq_rom out	sample_out	Sample ready	Generate_next	Notes
0	Load note	1	6'd1	0	XXXX	0	0	1	Load a new note
1	Load note	0	6'd1	6’d1	XXXX	0	0	0	Note is latched
2	Freq_ROM	0	6'd1	6’d1	{10’d009, 10’d395}	0	0	0	Step size is outputted
3	Sine_ROM	0	6'd1	6’d1	10’d009, 10’d395}	0	0	0	
4	Sine_Reader Out	0	6'd1	6’d1	10’d009, 10’d395}	1	16’d00050	0	Sine reader generates a sample
5	Codec	0	6'd1	6’d1	10’d009, 10’d395}	0	16’d00050	1	Codec signals for a new sample
6	Sine_ROM	0	6'd1	6’d1	10’d009, 10’d395}	0	16’d00050	0	
7	Sine_Reader Out	0	6'd1	6’d1	10’d009, 10’d395}	1	16’d1870	0	Add step size to get new sample
8	Load note	1	6'd6	6’d1	10’d009, 10’d395}	0	16’d1870	0	
9	Load note	0	6'd6	6’d6	10’d009, 10’d395}	0	16’d1870	0	Notice that sample does not change until sample ready goes high
10	Freq_ROM	0	6’d6	6’d6	{10'd012, 10'd557}	0	16’d1870	0	
11	Sine_ROM	0	6'd6	6’d6	{10'd012, 10'd557}	0	16’d1870	0	
12	Sine_Reader Out	0	6'd6	6’d6	{10'd012, 10'd557}	1	16’d24713	0	
13	Codec	0	6'd6	6’d6	{10'd012, 10'd557}	0	16’d24713	1	

Sine_Reader								
Cycle	**Play**	**State**	**Generate_next**	**next_addr**	**current addr**	**Sine_ROM output**	**sample_ready**	**sample**
0	1	Signal to generate next sample	1	0	XXXXXX	XXXXXX	0	0
1	1	Next_addr pushed to curr_addr	0	0	0	XXXXXX	0	0
2	1	Curr_addr passed into sine_ROM	0	0	0	16'd00000	1	16'd0000
3	1	sine_ROM output	1	0	0	16'd00000	0	0
4	1	Process repeats						