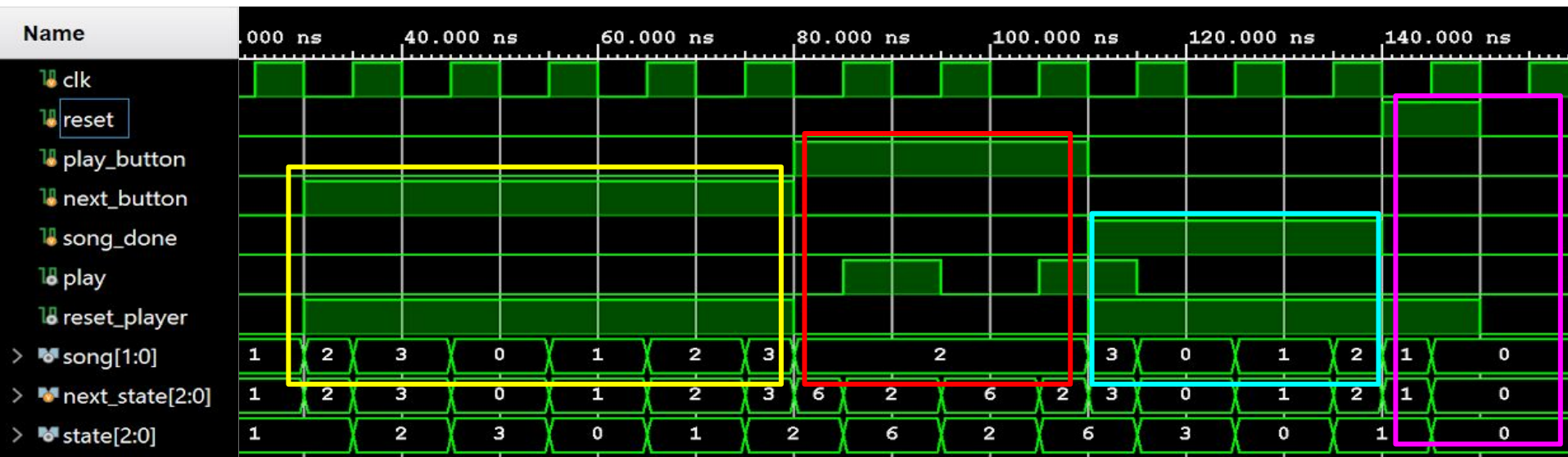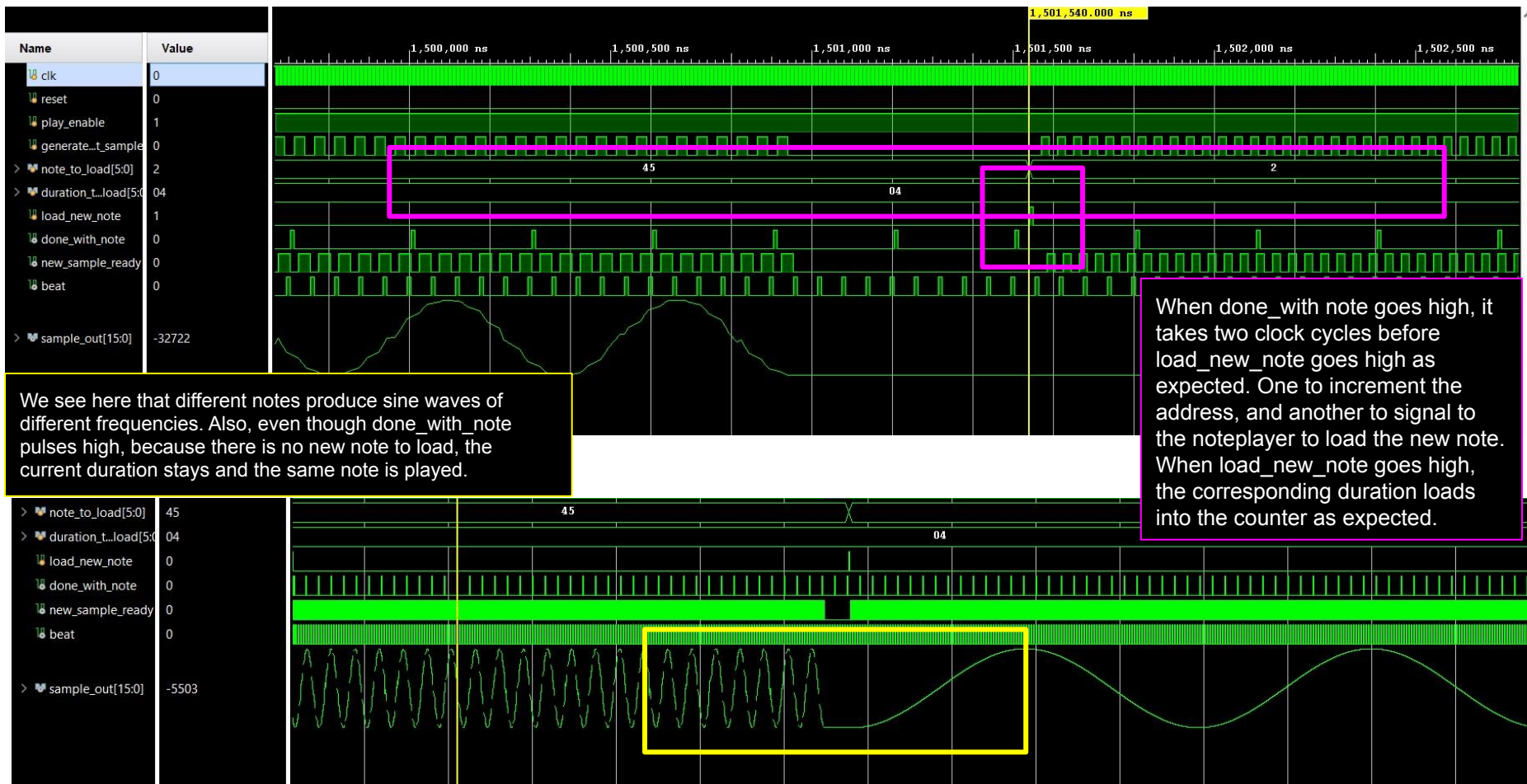# MCU

Note that as next_button is on, reset_player is on and the song number changes. When next is pressed on song 3, it wraps around to play song 0. Play is never on because when you move to a new song, it starts off on the paused state associated with that song and waits for play_button to go high.

Although play_button stays high, play does not because each clock cycle will invert the state of play since the play button functions both as a play and pause button. For example, when play_button is pressed while play is high, it pauses the song and takes play low.
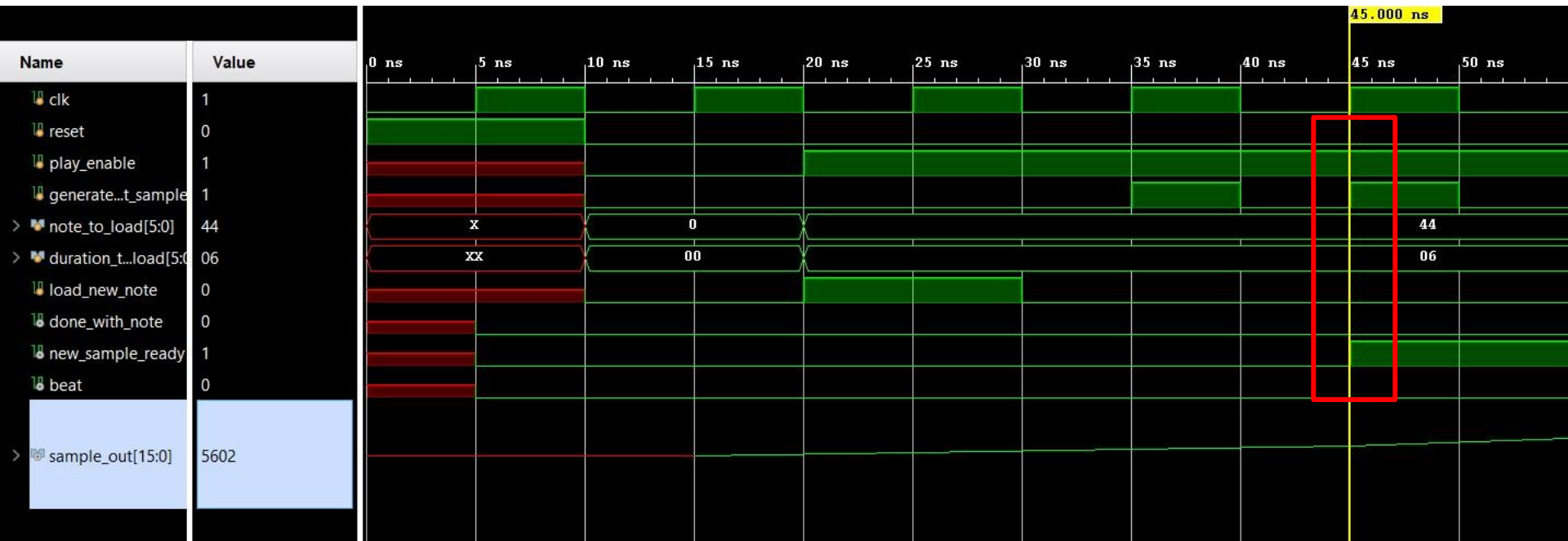
Similar to when next_button goes high, reset_player goes high, the song changes and play is off.

Song seems to go backwards from 1 to 0 which normally wouldn't happen - but this is due to reset being high. Whenever reset is high, no matter where it is, it takes the player back to the pause state associated to song 0.
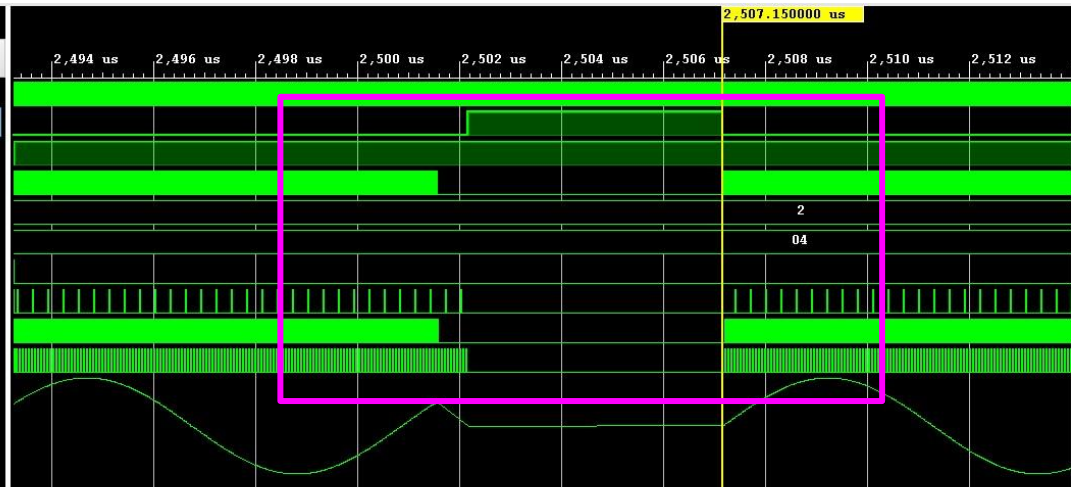
Note_Player

We see here that different notes produce sine waves of different frequencies. Also, even though done_with_note pulses high, because there is no new note to load, the current duration stays and the same note is played.

When done_with note goes high, it takes two clock cycles before load_new_note goes high as expected. One to increment the address, and another to signal to the noteplayer to load the new note. When load_new_note goes high, the corresponding duration loads into the counter as expected.

Note_Player pt2

Sample_out is 0 until new_sample_ready goes HIGH
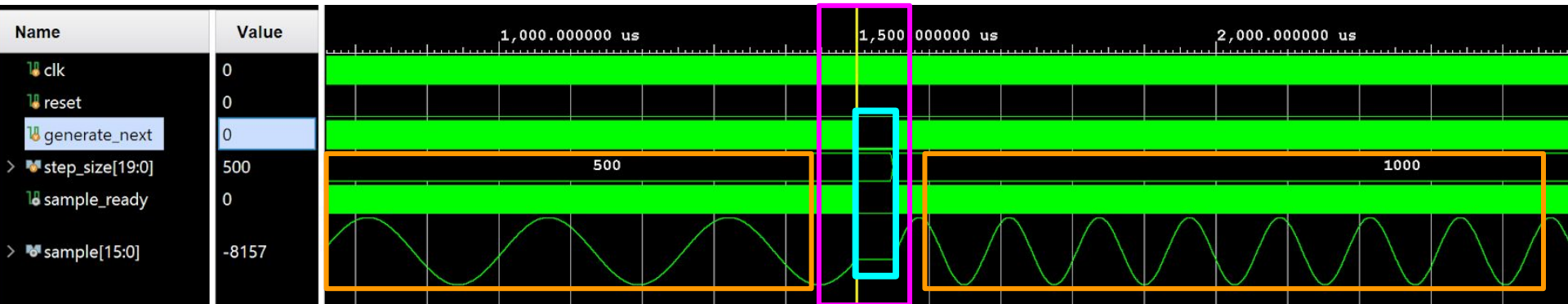
Note_Player pt3

When we toggle reset, the sample_out, new_sample_ready, and generate_next_sample all reset to 0 as expected. When reset turns back on, the note is loaded again.

When play_enable is toggled off, the duration counter stops counting and so the sample remains at whatever value it was previously until the enable is turned back on as expected

Sine_Reader

Name	Value
clk	0
reset	0
generate_next	0
step_size[19:0]	500
sample_ready	0
sample[15:0]	-8157

1,000.000000 us   1,500.000000 us   2,000.000000 us

500   1000
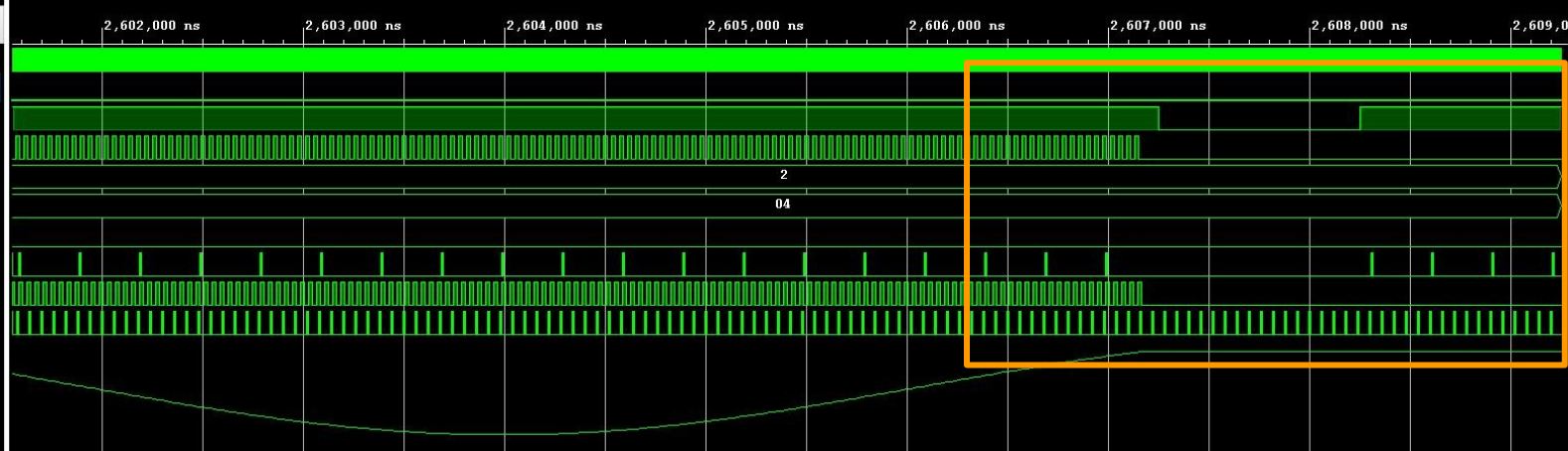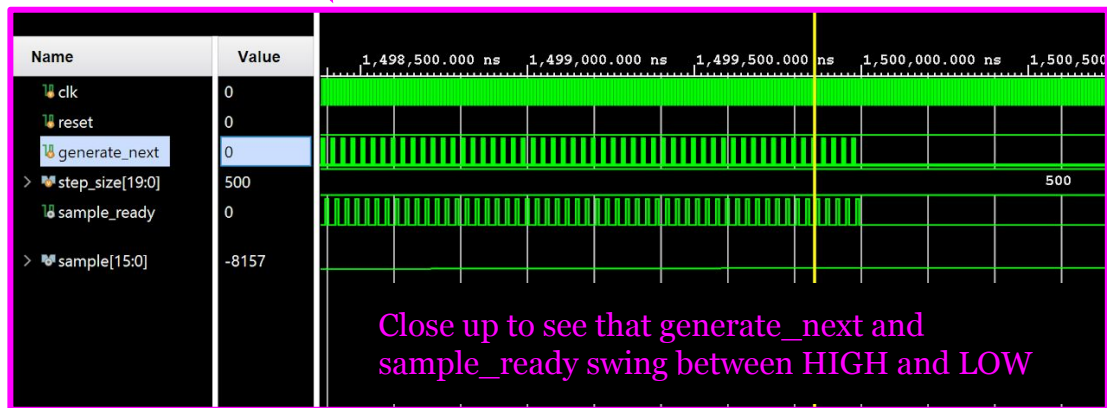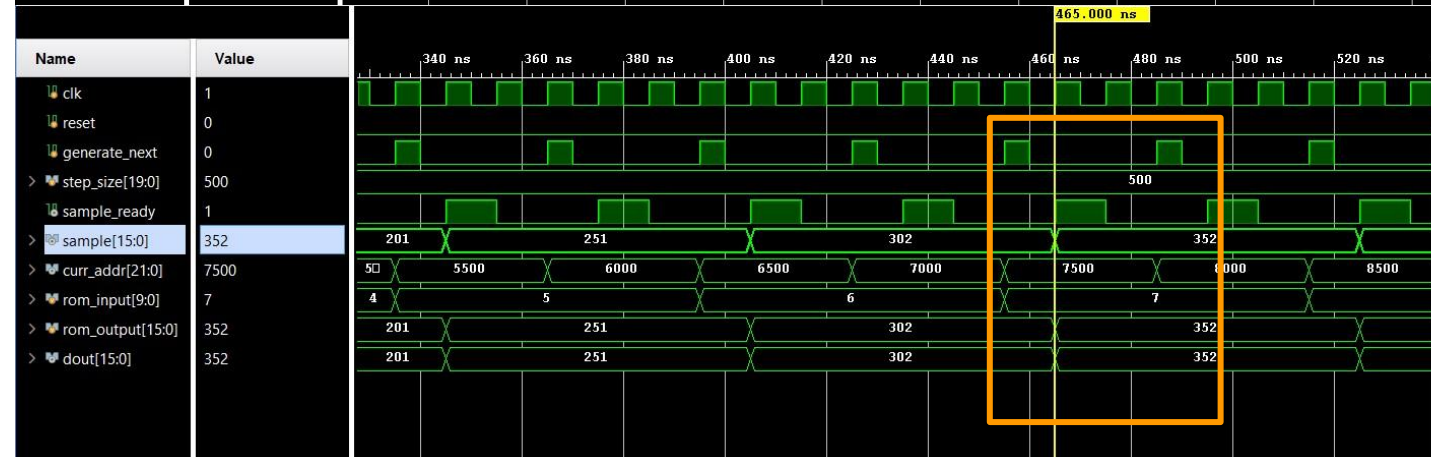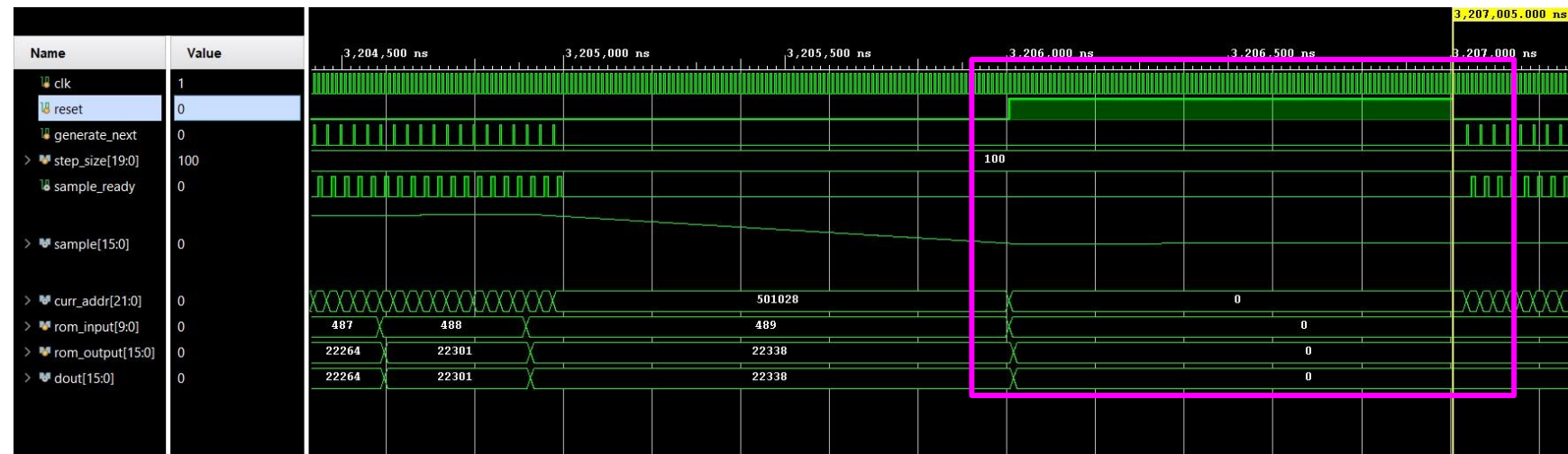
Note that with a different step_size we receive a waveform with a different frequency

While generate_next is LOW, the sample value is held steady and sample_ready stays LOW.

Name	Value
clk	0
reset	0
generate_next	0
step_size[19:0]	500
sample_ready	0
sample[15:0]	-8157

1,498,500.000 ns   1,499,000.000 ns   1,499,500.000 ns   1,500,000.000 ns   1,500,500

500

Close up to see that generate_next and sample_ready swing between HIGH and LOW
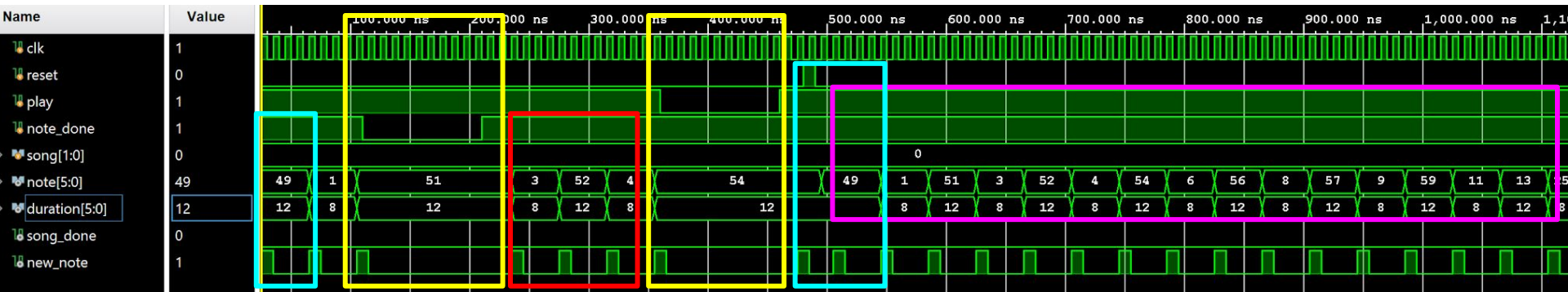
Sine_Reader pt2

When reset is toggled on, the sample resets to 0 and sample_ready and generate_next both got to 0 until reset is turned off as expected.

Sample_out changes only when sample_ready pulses high as expected.

Song_Reader

While play is LOW, song_reader holds the note it is on and doesn't move on.

Each time new_note goes HIGH, the note will change, and when it is LOW it holds on to the current note.

The first note in song0 is 49 as demonstrated in the first blue box, so when reset goes HIGH it starts over with note 49 as demonstrated in the second blue box.

While play is HIGH, song_reader is able to move on and process each note.

Song_Reader pt. 2
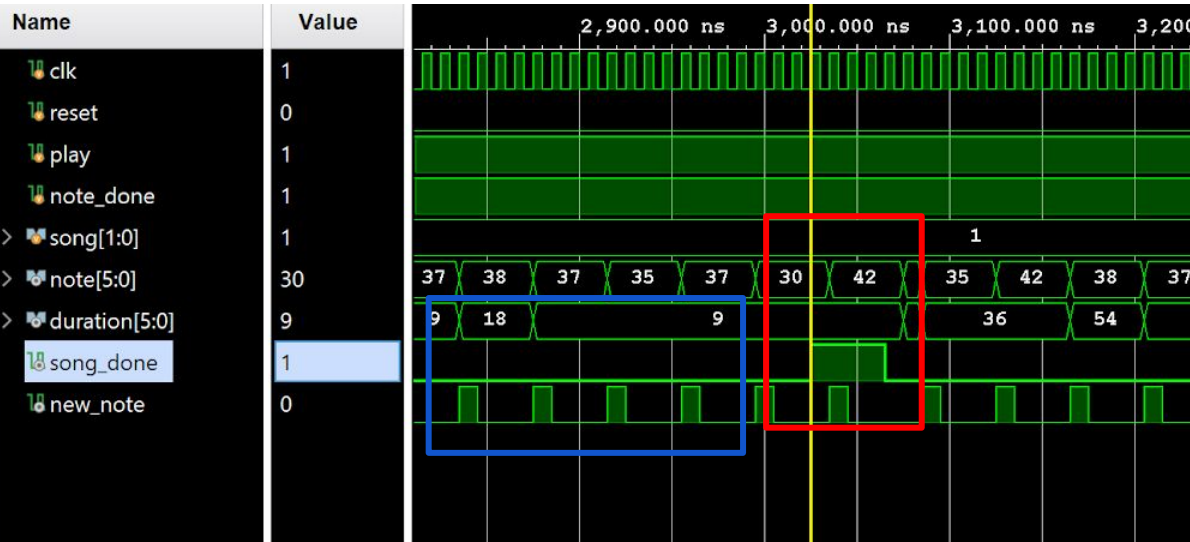
When the final note in song1 is played, song_done goes high. Note that the song does not change because song is an input and it is hardcoded to be 1 here. In the system itself you would normally see it transition to the paused state associated to song2.

Although the durations are different, new note still happens at the same rate and that is because in normally it would be note_player who notifies song_reader note_done must go HIGH. For testbench purposes, we just keep note_done HIGH.