2 The Music/Tone Generator

Generating a square/sine-wave of some frequency (between 20Hz to 20KHz) and connecting the output to a speaker creates an audible tone. Musical notes (A, B, C, $C^{\#}$, etc.) can be generated by designing a clock corresponding to that note's frequency. Of course, the quality of the sound depends upon amplification, filtering, signal processing, etc., but this project will focus on generating the tones. Selecting tones at different time intervals produces "music".

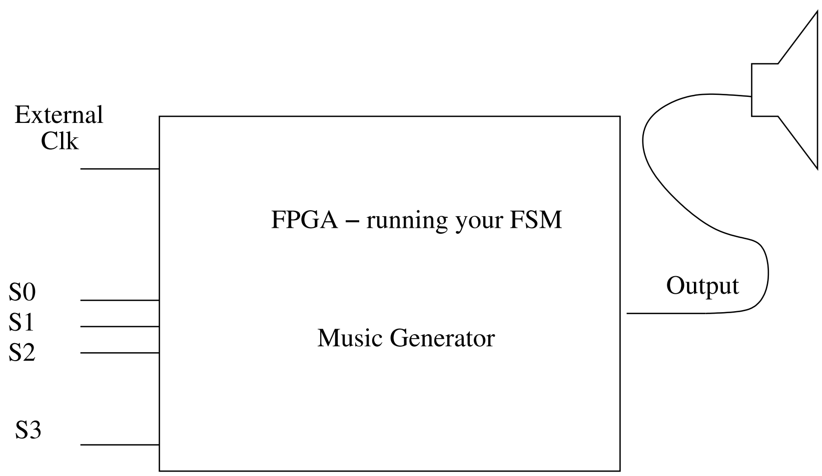

Figure 1: Block diagram for the music/tone generator.

In this project, you will design the music/tone generator shown in Fig. 1 and described below.

- Using the push-buttons $S0, S1, S2$, your circuit should play each of the 7 major notes.
- When $S3 = 0$, your circuit should play the note corresponding to the buttons pressed. When $S3 = 1$, you'll ignore the buttons and play a pre-recorded song stored in a ROM.
- When none of the buttons are pressed, your circuit will not play any note. When you press $S0$, you'll play note A. When you press $S1$, you can play B. When you press all three pushbuttons together you'll play the last note G. Therefore, by pressing a combination of these pushbuttons, you can create "music".
- When $S3 = 1$, you'll play a pre-recorded tune stored in a ROM. One example song provided is called *Habanera* from *Carmen*, an opera by Bizet. The composition (the notes, the frequency, and the duration of each note) is available as another document on CANVAS. Using the table, you'll hardcode a ROM that stores the frequency and duration of each note to play. While this may look complicated, it is just a bunch of counters selected by the FSM.
- A midi file corresponding to the tune is also uploaded on CANVAS, so you can compare what you hear to the actual tune.
- Your circuit should repeat the tune over an over as long as $S3 = 1$. When $S3 = 0$, your circuit should start reading the push-buttons. HINT: you might find it simpler to put $S3$ on a DIP switch.
- Speakers are available on the desks in the lab for testing and demonstration.
- For the "Note to Frequency" translation, refer to the site:

  `http://www.phy.mtu.edu/~suits/notefreqs.html`

Requirements

- **Milestone 1**: due in your discussion section the week of April 7th. You must show the TA a high-level ASM chart for your project and a detailed block diagram for your design.
- **Milestone 2**: due in your discussion section the week of April 14th. You must show the TA an ASM chart for your FSM control and Verilog code for your design though it does not need to be completely functional.
- **Final demonstration and report**: no later than April 23rd, you must demonstrate your working project to a TA. At or before your demonstration, you must turn in a project report that includes a high-level ASM chart for your design, a detailed block diagram for your datapath, ASM chart for your control, some key annotated simulation plots, and a brief description of any assumptions, important features, optimizations performed, and test strategy. At the same time, all Verilog source files for your project must be uploaded on CANVAS.
- **Extra credit:** if you demonstrate your working project and turn in all project materials by April 18th, you will receive 10 extra lab points.
- **Extra credit:** if you complete both projects before April 23rd, the second project will count towards up to 100 extra lab points.