System Design Document:

Song Recorder with Playback

University at Buffalo, The State University at New York
EE478 Fall 2019

Written by: Serena LaFave
Engineer: Serena LaFave

Table of Contents

List of Figures

1. Introduction

1.1 Overview

This system was developed and simulated using Xilinx Vivado, then programmed and demonstrated on the Zybo Z7 FPGA.

The system has two modes, recording and playback, which are selected based on a slider switch. Each of the four pushbuttons on the board activate a musical note and play it through headphones connected to the HPH OUT jack.

During recording mode, pressing any pushbutton will start the recording. Up to ten seconds of notes can be recorded, with a minimum note length of 0.25 seconds. Recording a new song will overwrite the last one. During playback mode, pressing any pushbutton will trigger the song to be played through the headphones.

Patterns are output on the four LED lights to indicate different statuses – a “1010” pattern indicates standby for both modes, all lights flashing on and off indicate active recording, and all lights on and steady indicate active playback.

1.2 Document Scope

The scope of this document is to provide an accurate description of the structure and functionality of the song recorder with playback for the final project of EE478 in the Fall Semester of 2019. This document discusses the inputs and outputs, detailed design of the system, and implementation on the FPGA device.

1.3 Intended Audience

The audience for this document is the EE478 session TAs and professor, as well as any GitHub visitors.

2. System Design Overview

2.1 System Block Diagram and Description

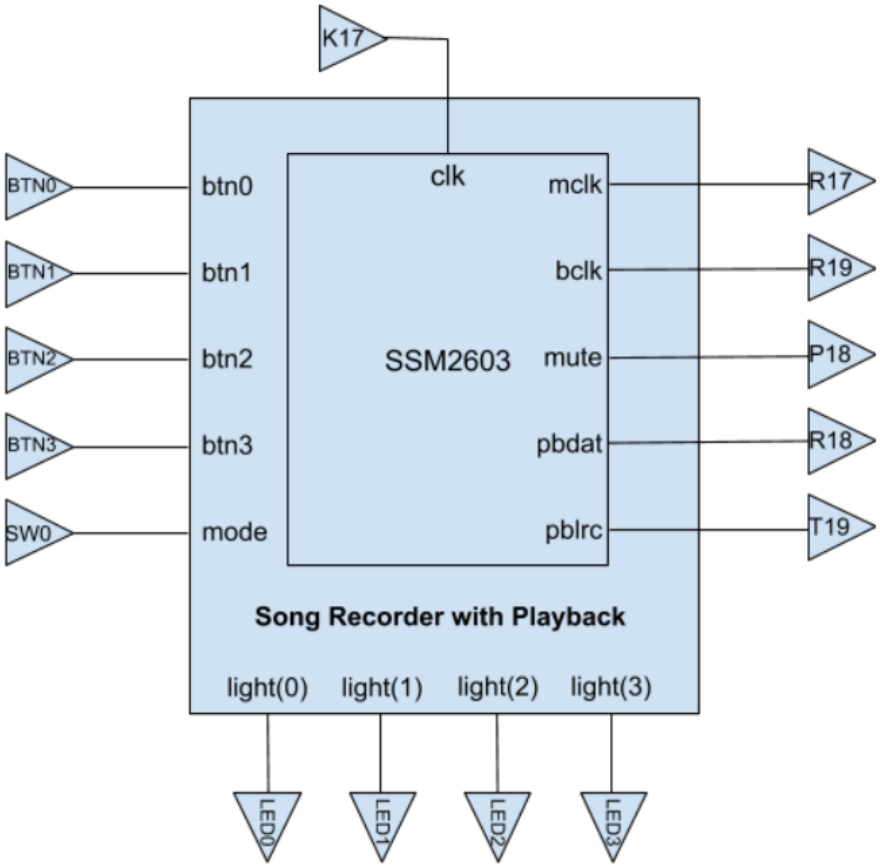

Figure 1: Song Recorder with Playback Block Diagram

The module has the main clock input from the board's pin K17, *clk*, which is sent directly to the SSM2603 audio codec. Pushbuttons BTN0 – BTN3 are inputs for the corresponding signals *btn0* – *btn3*. Slider switch SW0 determines signal *mode*. There are five outputs from the audio codec, *mclk*, *bclk*, *mute*, *pbdat*, and *pblrc*; the labels inside the output arrows for these signals are the names of the board's pins to which they are connected to. There are four output signals from signal *light*, which are connected to LED0 – LED3.

2.2 Song Recorder with Playback Module

The SSM2603 audio codec incorporated into the Zybo board is used to output the audio data. Because this codec requires a 12.288 MHz clock, a PLL is used to derive one from the 125 MHz main clock. The signal corresponding to this PLL clock, *mclk_clk*, is used in the sensitivity list of all processes included in this module.

The notes are created by changing the value of the internal signal *tone_terminal_count*. The codec requests data at a 48 kHz rate. Each time this data is requested, a counter signal, *tone_counter*, is incremented. Once it reaches half the specified value of *tone_terminal_count*, the data sent to the codec (*l_data* and *r_data*) are switched between zero amplitude and almost full amplitude. This creates a square wave whose frequency is determined by *tone_terminal_count*. Each pushbutton corresponds to a constant assigned to this signal, producing the musical notes C5, D5, E5, and F5.

An array of 40 elements, *rec_arr*, is used to store the notes during recording and make them available for playback. Each element is an integer with range 0 to 100 to allow for different *tone_terminal_count* values. A 4 Hz clock, *clk4*, is created using a counting process. Using *clk4* to cycle through the array results in each element being active for 0.25 seconds, totaling 10 seconds.

Recording mode consists of two states: *REC_IDLE* and *RECORDING*. *REC_IDLE* is the default when recording mode is activated by positioning the slider switch to 0 (signal *mode* = 0). This is done by initializing *rec_state*, the signal holding the current recording state, to *REC_IDLE* when the signal is declared. Also, the recording state is set to *REC_IDLE* whenever the mode is switched to playback by positioning the slider switch to 1 (signal *mode* = 1).

The *REC_IDLE* state is indicated by a "1010" LED pattern, i.e. LED1 and LED3 are on. Once a pushbutton is pressed, the state changes to *RECORDING*. In this state, the system cycles through *rec_arr* using signal *rec_cnt* to increment to the next element. If a pushbutton is activated during the 0.25 seconds of the current element, the element will be assigned to the respective *tone_terminal_count* value. If no pushbutton is activated, then the element will be assigned to 0 (through conditional statements, a zero-valued element will result in audio data with zero amplitude).

During the recording process, any pushbutton pressed will play its corresponding note through the headphones so the user can hear the song they are recording. Also during this time, all four LEDs blink on and off every 1.25 seconds to indicate active recording. After the 10 seconds of recording are concluded, the state returns to *REC_IDLE*. If a new note is pressed, the old song will be deleted and overwritten.

Playback mode consists of two states: *PLAY_IDLE* and *PLAYING*. *PLAY_IDLE* is the default when playback mode is activated. This is done by initializing *play_state*, the signal holding the current playback state, to *PLAY_IDLE* when the signal is declared. Also, the playback state is set to *PLAY_IDLE* whenever the mode is switched to recording.

The *PLAY_IDLE* state is indicated by a "1010" LED pattern. Once any pushbutton is pressed, the state changes to *PLAYING* and plays the previously recorded song through the headphones. This is done by cycling through *rec_arr* using signal *play_cnt* to increment to the next element. During this time, all four LEDs are on and steady to indicate active playback. After the 10 seconds of playback are concluded, the state will return to *PLAY_IDLE*.

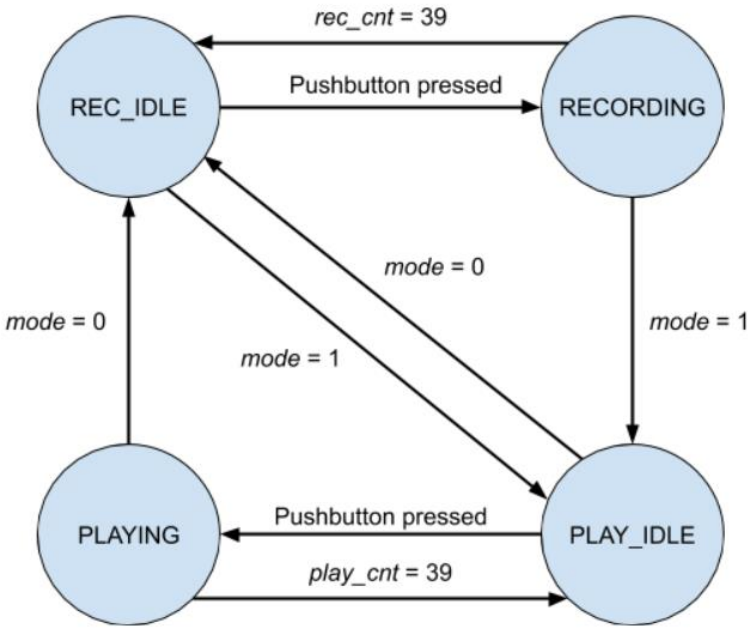

Figure 2: State Diagram of the Module.

3. Testing and Verification

3.1 Objective Verification

Functionality of the module was verified in deployment on the ZYNQ-7010 development board. When programmed, and with the slider switch at 0, the module output an LED pattern of "1010". When a pushbutton was pressed, its corresponding note was played through the headphones and all four LEDs began flashing on and off every 1.25 seconds. Any pushbutton pressed within the 10 seconds of recording also output the corresponding note. The LEDs then returned to the "1010" pattern. If a pushbutton was pressed again, the process would start over.

When the slider switch was changed to 1, the LEDs showed the "1010" pattern whether or not the recording mode had been idle or active. When a pushbutton was pressed, all four LEDs turned on and stayed steady for 10 seconds. If any notes had been recorded, it played the song back through the headphones. If the slider switch was changed to 0, even during active playback, the module would return to the idle state for recording.

Glossary

List of Abbreviations

- VHDL: VHSIC Hardware Description Language
- VHISC: Very High Speed Integrated Circuit
- FPGA: Field Programmable Gate Array
- BTN: Pushbutton
- SW: Switch
- LED: Light Emitting Diode
- HPH: Headphones
- PLL: Phase Lock Loop

Hardware References

- Zybo Z7 Development Board Reference Material https://reference.digilentinc.com/reference/programmable-logic/zybo-z7/reference-manual
- Xilinx Vivado Design Suite – HLx Editions https://www.xilinx.com/products/design-tools/vivado.html
- SSM2603 Audio Codec Data Sheet https://www.analog.com/media/en/technical-documentation/data-sheets/SSM2603.pdf