Multi-Channel Audio Recorder & Looper

Syed Ahmed, William Blackburn, Geoffrey Williston
Electrical and Computer Engineering Department
School of Engineering and Computer Science
Oakland University, Rochester, MI
sfahmed@oakland.edu, wblackbu@oakland.edu, gswillis@oakland.edu

***Abstract* — The multi-channel audio recorder and looper enables the recording and replay of audio to one of sixteen separate tracks. Four switches are used to determine the track to be recorded or played back, one switch enables the replay of audio from a selected track, and another switch enables recording to a selected track if replay has not been enabled. LEDs are used to indicate record and replay. The seven-segment display shows the selected track. The implementation of the multi-channel audio recorder and looper was motivated by a need to demonstrate both understanding of the circuit design methods covered in CSE 376 and the ability to gain proficiency in the use of peripherals to the Nexys-4 DDR board. This purpose was fulfilled. The successful implementation of this project required both the use of finite-state machines, registers, and multiplexers as well as operation of significant peripherals such as the on-board microphone, the audio output, and the DDR2 SDRAM external memory component.**

I. Introduction

The multi-channel audio recorder and looper serves as a useful simplified example of important problems in product design.

The project requires successfully integrating the behavior of different peripheral components into a single coherent design. The timing of microphone inputs, the timing of reading and writing to external memory, and the timing of audio output all had to be properly arranged.

Furthermore, the project required a sophisticated use of circuit design. A complex set of registers had to be placed in the audio input and output module to allow for recording and replay. A number of different counters proved to be unavoidable. Finally, a finite-state machine is needed for the control of replaying and recording.

II. Methodology

The overall design of the multi-channel audio recorder and looper is shown in Figure 1. For clarity, some changes have been made relative to a direct transcription of design. For example, the audio input/output module must send out a 2.38 MHz microphone clock to receive data from the microphone, the record switch is explicitly labeled in Figure 1 while in the design the record switch is only ever referred to by number, some of the modules are not explicitly separated in this manner in the VHDL code, etc. Figure 1 is faithful to the overall logic of the design. The algorithmic state machine chart is shown in Figure 2.

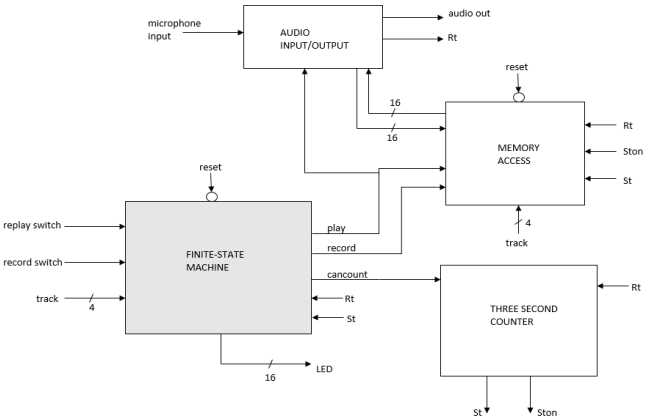

Figure 1: Figure 1 shows a simplified top-level diagram.

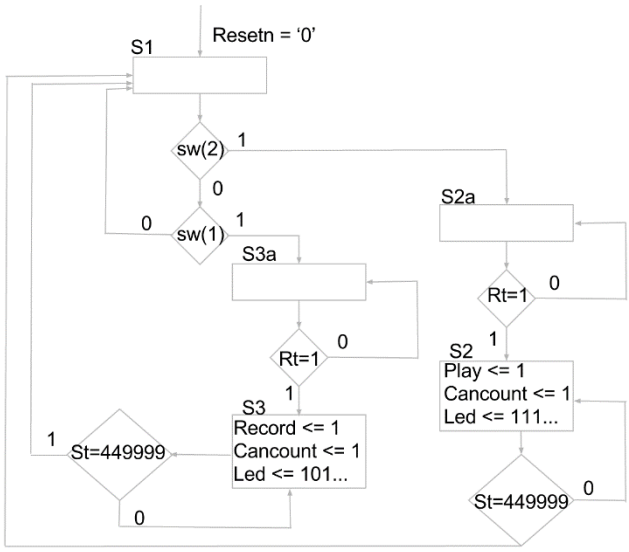

Figure 2: The algorithmic state machine chart of the finite-state machine.

As can be seen, the finite state machine of the recorder checks first for a request to replay and then checks for a request to record. If either request occurs, the finite state machine enters a transitionary period (either S2a or S3a) until the signal Rt is 1. This transitionary wait is to ensure the correct timing of the audio input/output buffer. After the transitionary wait, the finite-state machine sends an appropriate signal to either record or replay, indicates the process of recording or replaying, and enables a counter with output signal St, which counts for approximately 3 seconds.

During S2 and S3, the memory interface reads or writes audio information to the external memory in 16-bit increments. The timing of the signal St ensures that a novel set of 16 bits is read or written each time, and the value of the recording track and the signal St is used to determine the memory address to read or write from. If the audio recorder and looper is in the process of replay, the audio input/output module sends out the data retrieved from memory. Once the count is finished, the finite-state machine stops the process of replaying and recording and returns to the neutral state. If the request for record or replay is still active, the process begins again. The recorder holds samples that are approximately 3 seconds long.

A. *Reading From the Microphone*

In order to implement this project, the first challenge to overcome was successfully receiving input information from the microphone. To receive input from the microphone, a clock signal must first be sent out to the microphone. The relationship between input information and the microphone clock is shown in Figure 3. Figure 3 is taken from the Nexys-4 DDR manual [1].

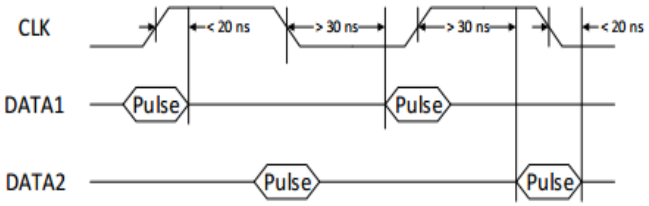

Figure 3: Figure 3 shows the relationship between the microphone clock and data input. The clock signal on top is not the 100 MHz clock but the clock defined for the microphone. Depending on a signal sent to the microphone, the window for reading values occurs on the rising edge (as in DATA1) or the falling edge (as in DATA2).

As can be seen in Figure 3, the microphone will only send out information in a short window of nanoseconds from when the microphone clock changes. The optimal frequency of the microphone clock was found to be approximately 2.4 MHz from the reference sheet of the microphone chip [2]. Using a simple counter, an approximately 2.38095 MHz clock signal was generated from the 100 MHz clock of the Nexys-4 DDR. A special signal was defined to mark the period immediately after the microphone clock changed. This signal was used to ensure that data from the microphone was stored for use only within the short window after the microphone clock changed.

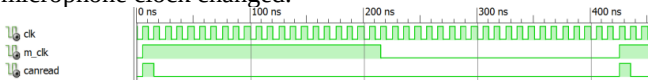

Figure 4: Figure 4 shows the 100 MHz clock, the microphone clock, and a signal to detect the window after the rising edge of the microphone clock.

B. *Audio Input and Output*

The step immediately following reading from the microphone is successfully outputting audio. Because both the microphone and the audio output used pulse density modulation, directly sending out bits received from the microphone or from memory would suffice for audio output as long as the output bit changed at the same rate as the microphone clock. Furthermore, the audio output should have a 16-bit buffer to accommodate the need of reading and writing to memory in 16-bit increments. With these concerns in mind, the initial design for the audio input/output module is shown in Figure 5.

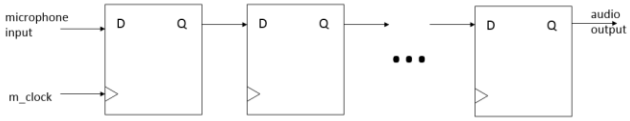

Figure 5: Figure 5 shows the initial design for the registers.

In order to more smoothly integrate the ability to read and write from memory, a series of three stages of 16-bit buffers was implemented in the final design, as shown in Figure 6.

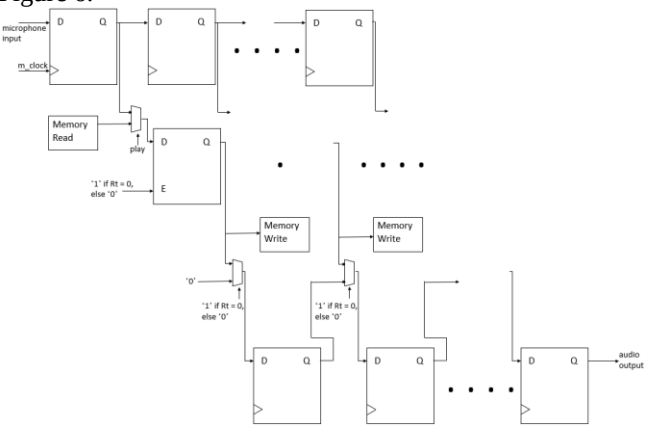

Figure 6: Figure 6 shows the final design for the three stages of buffer.

The signal Rt in Figure 6 counts the rising edges of the microphone clock. That Rt is 0 indicates that the microphone clock has experienced sixteen rising edges. As a result, this is the time to move each stage of the buffer down a level.

When replaying is not enabled, the first set of buffers accumulates 16 bits of microphone audio input, the second set of buffers stores sixteen bits of audio, and the third set of buffers sends out 16 bits of audio one bit at a time.

When replaying is enabled, the middle set of buffers has its values replaced with values from memory, which are to be played out.

C. *Memory*

The significant inputs and outputs to memory are shown in Figure 1.

As suggested in Figure 1, the address is specified by the three second count and the selected track. The ability to read and write from memory is determined using a signal called

"Ston", which takes into account the state of the registers in the audio input/output module. The timing of the registers is important because one wishes to replace the middle set of buffers at the correct point in time.

The memory makes use of code provided by the Digilent employee Mihaita Nagy [3]. The code was essentially used as a black box.

D. *Finite-State Machine, LEDs, and Seven-Segment Display*

The finite state machine is fairly simple. Its operations are as described in Figure 2. From the discussion of the buffers in section II.B of this paper, it should now be clear why the transitional states S2a and S3a exist. These transitionary states allow for proper timing of access to the buffer registers.

The finite state machine controls the LEDs directly and sends out a different signal based on the state. The seven-segment display just takes in the track selected and returns the hex value for display.

III. Experimental Setup

The multi-channel audio recorder and looper was first tested using two pairs of earbud headphones, of the same kind usually used to hear music from a laptop computer or iPod. A Nexys-4 DDR board was programmed using the design previously described. One set of headphones was attached to the audio output of the Nexys-4 DDR board. The other set of audio headphones was connected to a laptop computer. Music was played out of the computer earbud headphones to simulate speech, and the computer earbud headphones were placed near the onboard microphone. Audio output was heard from the Nexys-4 DDR board audio output headphones. Multiple kinds of sample songs were recorded and replayed on different tracks using this method.

A second test was conducted using the same methodology as the first, but with speakerphones attached to the Nexys-4 DDR board instead of headphones.

IV. Results

The recording and replay was successfully implemented with both headphones and speakers. Recording and replay successfully used multiple tracks. This capability was demonstrated both in the lab and in the classroom demonstration. The operation of the audio looper can be observed at this link:
https://www.youtube.com/watch?v=6ccvVDBzQx0

The audio output had some static, which was likely due to problems in timing memory reads with buffer shifts. There were some difficulties recording music at higher pitches, which was likely due to implicit limitations in hardware. The highest frequency at which the microphone can record is limited. In spite of these issues, the basic functions of the project were demonstrated.

Conclusions

This project successfully uses a finite state machine, counters, and registers to interface with peripherals of the Nexys-4 DDR board. Audio input, audio output, and external memory are all critical components utilized in the design.

There are a few additional features to be implemented. Static can be removed through better timing of reads and writes to memory or digital signal processing. The length of the recording can be increased by increasing the length of the count and modifying the way the memory component generates addresses for external memory.

References

[1] "Reference Manual," Nexys 4 DDR Resource Center, 11-Sep-2014. [Online]. Available at: https://reference.digilentinc.com/nexys4-ddr:refmanual.

[2] "ADMP421 Datasheet and Product Info," Analog Devices. [Online]. Available at: http://www.analog.com/en/products/obsolete/admp421.html.

[3] S. Lowe and M. Nagy , "SRAM to DDR Component," Nexys 4 DDR Resource Center, 09-Feb-2016. [Online]. Available at: https://reference.digilentinc.com/nexys4-ddr:sram.