Report: Exercise 2

Group 9: Øyvin Richardsen, Sandor Zeestraten, Stian Habbestad

Norwegian University of Science and Technology
TDT4258 Energy Efficient Computer Design
March 14, 2013

Abstract

Contents

1 Introduction

2 Description and methodology

We started out by implementing the previous exercise that we had coded in assembly, only this time in C. This was an easy start for implementing an interrupt routine based on a program we already had good knowledge of. When we got this to work we were ready to enable and set up the ABDAC. First out we started with simply generating some noise, so as to easily check that we were able to make audible sound. Now we set out to make some sine waves as a first step of a synthesizer. We then used those sine waves to generate tones in a "audio.h" header file. This made the basis for simple synthesizing which we used to make simple sounds, and even the melody tetris. We spent some time trying to get rid of the random noise and tones we ended up with after playing our sounds. After tweaking the debounce delay properly and resetting the ABDAC this worked out nicely.

3 Results

4 Tests

Description We've created a few test scenarios in order to test different aspects and corner cases of our code. The main prerequisites were the STK1000 development board, JTAGICE MKII and a headset.

The setup of the board: The headset was connected to the minijack on the development board, the jumper 4 and 5 was set to "INT. DAC". Switches were connected to port C on the parallel I/O, and LED's on port B. Both the STK1000 and JTAGICE MKII debugger were connected and powered on. The tests were conducted by a person interacting with the switches wearing a headset, and another person logging the results.

Results Below is a table of the different tests we ran, the preconditions and the results.

Name	Preconditions	Description	Expected result	Test result
Steady-state test	Power is on, and the board is connected	Upload program to card, push reset switch	The board is powered and LED 7 should be on	Passed
Move LED right test	Program is active, only LED 7 is on	Push switch 6	LED 7 should be turned off and LED 6 should be turned on	Passed
Move LED left test	Program is active, only LED 6 is on	Push switch 7	LED 6 should be turned off and LED 7 should be turned on	Passed
Left LED wrap-around test	Program is active, only LED 7 is on	Push switch 67	LED 7 should be turned off and LED 0 should be turned on	Passed
Right LED wrap-around test	Program is active, only LED 0 is on	Push switch 6	LED 0 should be turned off and LED 7 should be turned on	Passed
Long push test	Program is active, only LED 7 is on	Push and hold switch 6 for a few seconds	LED 7 should be turned off and LED 6 should be turned on as soon as the switch is pushed	Passed

5 Evaluation of assignment

This assignment was more demanding yet also more fun than the first assignment. The learning curve was a bit harder as not all of us had coded much in C before, but it was a nice opportunity to learn.

6 Conclusion

7 Appendix

References

[1] AVR32 Architecture Document `http://www.atmel.com/images/doc32000.pdf`

[2] AT32AP7000 Datasheet `http://www.atmel.com/Images/doc32003.pdf`

[3] TDT4258 Compendium `http://www.idi.ntnu.no/emner/tdt4258/_media/kompendium.pdf`