1 Introduction

In this lab you will apply the debugging techniques from the previous two labs to extend the ADC-PWM audio processing program to implement stereo input (two channels) and stereo output. In doing so, you should use the debugging and performance analysis techniques from previous labs where necessary. Additionally, you will need to consult the datasheet (AT91SAM7L128/64 Preliminary) for technical details related to the ADC and PWM.

Recall from lab02 and lab03, we have created a system that samples analog audio signals and recreates equivalent analog signals in a pulse-width modulated (PWM) waveform. This system inputs a single analog input and outputs a single PWM output. In this lab, you will extend this system to input and output two signals simultaneously. The finished design should sample and retransmit a stereo audio signal received from a stereo audio jack. See the Wikipedia page for a description and history of stereophonic sound (http://en.wikipedia.org/wiki/Stereo).

2 Lab Procedure

2.1 Download and Run lab03 Project

1. Unzip the lab04 source code from the lab04.zip file attached with this lab assignment. Note the similarities between lab04 and the source code from lab02 and lab03.
2. Use IAR Embedded Workbench IDE to make, download, and run the program.

2.2 Modify the Code

1. Modify the source code to implement stereo sound.
2. We have left many of the design decisions up to you, so please comment your code sufficiently in order to document which input pins you use, which output pins you use, and any other design decisions you make.
3. Use the debugging skills you have learned in the previous labs to make your job easier.
4. Consult the datasheet (AT91SAM7L128/64 Preliminary) for technical details about both the ADC and PWM peripherals, including setting the register bits.

2.3 Debug and Test

1. Use breakpoints, the watch window, debug variables, debug pins, the oscilloscope, etc. to thoroughly debug and test your program. We will provide speakers and speaker connections for testing.
2. Use an output pin (see lab03) to ensure that the interrupt service routine (ISR) completes before another interrupt is triggered.
3. Demonstrate your functioning system to a TA.

TA Initial Lab04 Box 1

2.4 Compare to lab03

1. Measure the frequency and period of the PWM ISR.
2. Calculate the slack (amount of time per interrupt period where the ISR is not executing)
3. Compare these results to those obtained from lab03, before extending the system to stereo.

3 Lab Report

Your lab report for lab04 should include answers to the following questions:

1. Turn in a copy of your modified main.c file with the lab report. Do not forget to put your name at the top of the file.
2. What is the largest bit-width you were able to use for the duty cycle of each PWM output? Why?
3. What are the frequency, period, and slack of the interrupt service routine? Compare to the results found in lab03 (the mono system). Account for any changes.
4. Did you notice any "bleeding" of one input channel to another? (i.e. were you able to hear the left channel input through the right channel output, or visa versa?) If so, explain why.
5. What are the system limitations to processing even more channels?
6. With only four analog input pins, how might you extend the system to input eight channels?