Laboratory 7 *Digital-to-Analog Converter*

Instructor: Dr. Lihong Zhang

1 Objectives

With the completion of this lab, you should be able to:

- Design interface between a DAC and the STK600 board
- Write assembly language programs to generate arbitrary analog signals

2 Introduction

Analog-to-Digital Converters (ADC) and Digital-to-Analog Converters (DAC) are extremely useful devices which enable us to interface the real world events (which are frequently analog) with microprocessors for monitoring or control. They are inexpensive but provide good precision and accuracy when compared to their pure analog counterparts.

There exist many types of ADCs and DACs differing in their internal construction, resolution, precision and accuracy, speed, compatibility with different microprocessors, and other capabilities like power supply requirements, unipolar/bipolar operation and coding. In this lab, you will interface a DAC to the STK600 board and test its operation. In next week's lab, you will operate on an ADC embedded within an AVR microcontroller.

Caution!

The DAC and AVR microcontroller that you will be using in this lab are sensitive to electrostatic fields and can be easily damaged. **Do not insert these devices into powered sockets.** Remove power before insertion or removal. Do not touch the pins of these devices with your fingers. It is always a good practice to switch off the power supply before you make any change to the hardware.

The DAC and ADC devices are normally expensive – please be careful not to ruin any chips.

It is always a good idea to measure the exact value of the required resistors or capacitors (if any) with an LCR meter before connecting these devices to DAC or ADC. This precaution may eliminate hours of frustration and debugging.

Note that, in this lab, a reference to a signal line/pin that is active low will have the * symbol after its name (for example, CS* or IOWR*).

3 Digital-to-Analog Converter

In this lab, we will use AD558JN as a voltage-output DAC chip. The data sheet of AD558JN is available in a PDF file on the D2L website. Read it fully before designing your circuit. This device is a simple 8-bit DAC.

3.1 Circuit Diagram

The AD558JN is to output (unipolar) analog signal in the range of 0V -- 2.56V. It will be used as an output port. Sketch a circuit diagram of the interface between AD558 and ATmega32 microcontroller (such as PORTD on the STK600 board), clearly showing all the interconnections and power supply connections. It is recommended you should NOT use PORTC for the interconnection with AD558 as some pins of ATmega32 PORTC on the STK600 board are reserved for certain communication/programming functionality.

Use ***Figure 1a*** on the AD558 data sheet (DIP packaging) for the pin names and numbers. Refer to ***Figure 3a*** to configure the device in the required mode. Connect the Chip Select (CS*) line and Chip Enable (CE*) input of the AD558 to the GND. The data bus lines and power supply points (+5V and ground) must also be connected.

3.2 Circuit wiring

Place the AD558 appropriately on the breadboard. Use a 10-wire cable (with a DIP connector on one end) to bring one ATmega32 port (such as PORTD on the STK600 board) to the breadboard. In addition, you need to use a 2-wire cable (with a 2-pin connector on one end) to bring the VTG (the target voltage on the STK600 board) and the ground to the breadboard.

Wire the circuit as in your diagram. Wire your circuit neatly – colour code your wires, and cut your wires to the right length. Make sure that all the ground points are connected together. Get the circuit checked out by your partner before powering up.

Question1: Turn the power ON and test the DAC for various output voltages. This can be done by using an oscilloscope or a DMM (digital multi-meter) to measure the output. ***Explain your observations.***

3.3 Software

Besides the connection made in Section 3.2, use a 10-wire cable to connect PORTB and SWITCHES headers on the STK600 board. Write an AVR assembly language program that increments one register, outputs it to the DAC port, delays a short while (say, a few us), and then repeats these operations above until a push-button switch is pressed (you can decide which push-button switch on the STK600 is used).

Question2: When you run this program, what output waveform do you obtain? You need to observe the waveform from an oscilloscope. Comment on your observation.

Question3: Assuming that the microprocessor poses no timing constraints, determine how fast can this device operate --- see Figure 7 of the AD558 data sheet and its timing specifications.

***Print out the .ASM file for your program, have a TA check for correct operation, and obtain a TA's signature*. Do the same things for Questions 4-7.**

Question4: Modify your program to produce the same waveform, but with half of the frequency. Submit the .ASM files of your program. Comment on your changes and results.

Question5: Modify your original program again to produce a waveform of double the frequency. Submit the .ASM files of your program. Comment on your changes and results.

Question6: Modify the original program to reduce the peak-to-peak voltage by one-half, but keep the same output waveform frequency. Submit the .ASM files of your program. Comment on your changes and results.

Question7: Create a program that outputs a triangle waveform with the maximum peak-to-peak voltage and the same frequency as Question2. Report on your results, and submit the .ASM file.

Question8: Based on the results of your programs, how could you generate a sinusoidal waveform? **Do not write a program**, but describe what would be involved in creating such a program.

4 Submission

At the end of the lab, submit the print-out of your files (.ASM, where appropriate). Also include your circuit diagram, answers to all the questions in the lab and prelab documents, and results.

Before leaving the lab, unwire your circuit, and return your devices to the IC kit at your bench.