EECS 373 Winter 2004

Lab 8: Analog to Digital Conversion

Requirements

Pre-lab: Individual answers to pre-lab questions are due at the beginning of your lab section during the week of March 15, 2004. Your group must also have an initial software design completed and entered into SingleStep.

In-lab: The In-lab demonstration sheet is due by Friday 5:30pm the week of March 15, 2004

Post-lab: Answers to post-lab questions are due at the beginning of your lab section during the week of March 22, 2004.

Value: This lab is worth 7% of your total grade.

Objectives

The purposes of this lab are:

1. To learn the operation of an analog-to-digital converter.
2. To observe the analog-to-digital conversion process and understand its parameters and limitations.
3. To increase your experience with reading device data sheets, interfacing digital I/O devices, and using interrupts.

Overview

Analog-to-digital converters (ADCs) allow microprocessor-based systems to sense and act on analog electrical signals, which commonly arise from sources such as a microphones, temperature sensors, etc. In this lab, you will interface the ADC on the lab expansion board to the MPC823 and study its operation.

Design Specification

Your system for Lab 8 will digitize an input analog signal continuously at a rate of 500 Hz and display the digital codes on the seven-segment display. In addition, pressing pushbutton S1 will buffer 500 consecutive samples (one second's worth) in memory.

The hardware for Lab 8 will simply interface the National Semiconductor ADC0808 converter found on the lab expansion board with the MPC823 via a single 8-bit read/write register. You can choose the specific address, but it should be in the same 0x0280XXXX range as your existing I/O devices. Beware of shadow address conflicts. A write to this device's register will initiate a conversion. The data value written selects one of the eight possible analog inputs. A read from the register returns the ADC output, i.e., the digital code corresponding to the input voltage level. Since the conversion itself requires a significant and variable amount of time, you will use the ADC's end-of-conversion signal to generate an interrupt.

The software for Lab 8 will be responsible for initiating conversions at the 500 Hz rate, collecting the digitized samples as they become available, and updating the seven-segment display and the memory array as appropriate. To maintain a precise sampling rate, you must initiate conversions at evenly spaced intervals, regardless of when the conversions complete.

Hardware Details

Print out the ADC0808 data sheet from the course web page and study it very carefully. Figure 5 on p. 7 gives the basic timing diagram for a conversion, and Figure 8 on p. 8 gives more important details. The values for the indicated timing parameters are listed on p. 4. Note that the "max" value for a "minimum" parameter is a worst-case minimum. For example, the minimum start pulse width is 100 ns for a typical device, but may be as much as 200 ns for some devices. In other words, your circuit should provide a start pulse of at least 200 ns.

The "typical application" diagram on p. 12 may be useful in understanding how your interface register should work. Note that this circuit only meets the minimum pulse width and setup and hold specifications if the bus is very slow (a few MHz at most). Your interface will be more complicated since you will need to extend the control signals to meet the timing specifications.

Interfacing to slower devices like the ADC0808 at a bus clock rate of 24MHz will require the insertion of many wait states. To minimize the number of wait states, the external bus clock will be slowed down to 10 MHz. The external bus clock is set to 10MHz with the SCCR, EBDF, control field, page 5-5 of the white book. The field must be set to "CLKOUT is GCLK2 divided by 2". Be sure to preserve the setings of the other fields of SCCR. Set this field in the intialiation section of your code. Although the external bus clock will be changed to 10MHz, the system clock, the source clock for the timers, remains at 24MHz.

The ADC0808 requires a clock to sequence the internal successive-approximation controller. You can use the bus clock from you PROC macro to generate the clock. Use a counter to divide the MPC823's bus clock down to a frequency within the ADC's specified range. The minimum and maximum clock rates are specified in the ADC0808 data sheet. The clock rate should be fast enough to keep the conversion time less then the sample interval. Remember, the bus clock is now set to 10MHz.

The Xilinx connections to the ADC are accessible through the LOCAL_DEVICES macro in the EECS373 library. The ADC data pins are wired on a common tristate 8-bit bus with the DACs and the SRAM; see the "Basic I/O" schematic (available on the web page) for details. This bus is represented by the D_OUT and D_IN connections and the tristate buffers are controlled by D_OUT_EN. If there is any confusion about the macro, use the hierarchy tool to look inside and see how it's connected to the Xilinx I/O pins. You should be able to match the pin numbers (the LOC= attributes) in the macro with the pin labels on the schematic.

The details of how S1 is handled are up to you. The important thing is that the 500 entries of the array represent 500 consecutive, in-order samples from the digitized signal, regardless of the duration of the S1 button press.

Pre-lab Assignment

1. The V_{REF} input on the ADC is fixed at 5V on the lab expansion board. Ideally, what change in ADC input voltage is equivalent to a 1 LSB change in the ADC output?
2. After a conversion, the ADC produces a value of 175. Assume that V_{REF} = 5V.
    a. If the ADC were ideal, what range of input analog voltages could generate this code?
    b. What is the range when you consider the error specification on the ADC data sheet?

3. Determine appropriate settings for the TMR1 prescaler and ICLK fields and the TRR1 register to generate an interrupt at a 500 Hz rate. Recall that the MPC823FADS board has a 24 MHz general system clock rate. (This will not change, even when the bus clock is slowed down to 10 MHz.)

4. Consider the timing diagram in Figure 5 of the ADC data sheet. Focus on the left portion where the analog input is specified and the conversion is initiated. Assuming a 10 MHz MPC823 bus clock, for how many bus cycles will START and ALE need to be asserted? How many additional cycles will be required to satisfy the ADC's hold time requirement? Draw a unified timing diagram combining an MPC823 bus write cycle (including the bus clock and all relevant MPC823 bus signals) with the START, ALE, and address signals of the ADC as shown on the left half of Figure 5.

5. Assuming a 10 MHz MPC823 bus clock, how many wait states will be required when reading from the ADC?

In-Lab Procedure

1. Add lines from the ADC's START, ALE, EOC, and CLOCK signals to the test points so they can be viewed on the logic analyzer. Download your ADC interface circuit to the Xilinx and write to the ADC register from the SingleStep command line. Set the SCCR, EBDF field to "CLKOUT is GCLK2 divided by 2" manually through the SingleStep register window. Note, GCLK2 does not refer to the GCLK2 on the PROC macro. Continue to use GCLK1 as the bus clock. It will be divided by 2. Verify that the START and ALE signals are asserted and measure their duration. Do they meet the minimum pulse width specification? Does EOC go low, then high? Is the bus clock running at 10Mhz? If not, debug your interface logic.

2. Once you observe the ADC control signals working properly in response to a write, verify that you can read a value from the ADC (again using the command line).

3. Once your hardware appears to work, debug your software. Again, verify one feature at a time in isolation before proceeding: the timer ISR, the EOC ISR, etc.

4. Build an adjustable voltage divider using the potentiometer and resistors available in the lab. The voltage dividers should already by built on a powered protoboard. See the board for the voltage divider and copy the schematic. By adjusting the potenioometer, measure the transition voltages for 0 - 1, 1 - 2, 3 - 4, 5 - 6, 7- 8.

5. Set up the waveform generator to output a 100 Hz sine wave, 4V peak-to-peak amplitude with a DC offset of 2.5 volts. That is, the sine wave should go down to 0.5V and up to 4.5V. **Use the oscilloscope to verify that the signal voltage is in this range *before* connecting it to the ADC.** If the input voltage goes below 0V or above 5V you could fry the ADC chip.

6. Press S1 to capture 500 samples in the memory array. Stop your program and read the samples out using the SingleStep command line. Do they look like they make sense? How many samples occur before the values cycle back to where they started? If everything looks OK, save the data in a file by using '>' to redirect the output of a 'read'. Make sure to save the output file for your lab report.

7. Repeat steps 7 and 8 for sine waves of 200 Hz, 400 Hz, and 800 Hz.

Demonstration 1.1: Demonstrate your system for the lab instructor.

Lab Report

1. Summarize the operation of your circuit and describe how you arrived at your design. Include a printout of your schematic.
2. Include a well-commented listing of your program.
3. Why is the particular voltage divider configuration and values necessary to make the measurements in part 4.
4. What were the output codes and transition voltages you recorded in step 4?
    a. Given these voltages and the digital output, what is the error at each transition relative to an ideal converter? Does this agree with the data sheet specification?
    b. What is the absolute error at each transition, including quantization error? (You should report the error at four points: just below and just above the transition voltage at each of the two transitions you measured.)
    c. What is the differential non-linearity for the step between the transitions? Does this agree with the data sheet specification?
5. Plot your data (using a spreadsheet or graphing program) for the 100 Hz, 200 Hz, 400 Hz, and 800 Hz sine waves. Are they all recognizable as sine waves? Estimate the signal frequency directly from each plot. What values do you arrive at? What does this tell you about the relationship between sampling rate and signal frequency?

Lab 8 Demonstration Sheet

Print this page and present it to your lab instructor when demonstrating the various lab sections. Turn this sheet in with your post lab or when your in lab demonstration is due. You are required to turn in only one demonstration sheet per group.

List Partners Names

_____

_____

_____

D1.1 Demonstrate your working system to your lab instructor.

Lab instructors initials: