EGCP 2110-01
Microprocessors
Laboratory #9

ADC and Interrupt Modes

Prepared By:
Alex Laird
Collin Barrett

on

Saturday, November 7th, 2009

Low-Level Source Code

Below is the assembly code that we ensured worked in the lab. We wrote it prior to having access to the Z-80 Microprocessor to test it. Upon presentation, our code ran correctly.

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;Created by: Alex Laird and Collin Barrett
;Date: Nov. 3, 2009
;Class: Microprocessors
;Lab 9: ADC and Interrupt Modes
;Purpose: To study, via implementation, the use of interrupts.
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;point the program start to the beginning of memory
      ORG 1800H

;disable and initialize the interrupt in ROM
      DI
      LD    A, 00H
      LD    (1F41H), A
      LD    A, 19H
      LD    (1F42H), A
      IM 1

MAIN:
      LD    HL, 0000H
      LD    B, 05H
;loop until an interrupt is received, gathering average of five
LOOP:
      EI
      OUT   (0C8H), A
      HALT
      DEC   B
      JP    NZ, LOOP
      CALL  OUTPUT
      JP    MAIN

;define the interrupt
      ORG   1900H
;disable interrupt
      DI
;read from the ADC and add to average
      IN    A, (0C8H)
      LD    E, A
      LD    D, 00H
      ADD   HL, DE
      RETI

;output results
OUTPUT:
      LD    A, H
      SLA   A
      SLA   A
      SLA   A
      SLA   A
```

```
;multiply by 8  by shifting L left and rotating the carry into H
      LD    B, 00H
      LD    C, L
      LD    H, 00H
      SLA   L
      RL    H
      SLA   L
      RL    H
      SLA   L
      RL    H
;multiply by 2 by adding twice
      ADC   HL, BC
      ADC   HL, BC
;combine ones and tenths
      OR    H

;output the BCD and return
      OUT   (0C0H), A
      RET
```

Program Overview

The program uses an interrupt to gather the voltage being fed into the ADC from the 10k pot. Interrupt Mode 1 is used, which means the Z80 jumps to ROM location 0038H and initializes the interrupt mode. Once the interrupt is initialized, the Z80 jumps to the memory location specified at 1F41H and 1F42H. 1900H, the address of our ISR (Interrupt Service Routine) is stored at these memory locations prior to calling IM 1.

The main loop continues looping until an interrupt signal is received. Once an interrupt signal is received, the voltage that the ADC has stored is gathered and accumulated with previously gathered voltages. Five voltages are gathered to average them together. Once five voltages have been gathered, the averaged voltage is output to the BCD.

In order to calculate the voltage stored on the ADC, we used the equation $(x/256)*5.0 = v$, where x is the number given by the ADC to represent the voltage and v is the output voltage between 0.0 and 5.0. Since the gathered voltage must be multiplied by five, it was simpler to gather five voltages and average (thus not having to divide by five) than to multiplying each gathered voltage by five.

Table 1 shows the actual voltage, read from a multimeter, compared to the voltage output to the BCD (which was calculated by our program).

	BCD Voltage (V)	Multimeter Voltage (V)	Percent Difference
Minimum (0.0) V	0.0	$6.43 * 10^{-3}$	.00064%
1.0 V	1.0	1.0108	1.08%
2.0 V	2.0	1.9945	0.275%
3.0 V	3.0	2.9953	0.157%
4.0 V	4.0	3.970	0.75%
Maximum (5.0) V	4.9	4.957	1.163%

Table 1 Comparable Voltages

The average percent difference was 0.685%, meaning our output voltage to the BCD is very nearly accurate, and most of the inaccuracy lies in rounding errors that are limited by the BCD and ADC themselves.