Experiment No. 05:

Input / Output Design

By: Theo Guidroz

Lab Partner: Daniyal Khan, Gregory Bonnema

Instructor: Dr. Jafar Saniie

TA: Xinrui Yu

ECE 441-001

Lab Date: 03-18-2020

Due Date: 04-03-2020

Acknowledgment: I acknowledge all of the work (including figures and codes) belongs to me and/or persons who are referenced.

Signature : 

I. Introduction

A. Purpose

The purpose of this lab is to gain knowledge on memory mapped I/O design and familiarize with interrupt handlers as well as implementing them.

B. Background

I/O is an important part of the computer system. Some common I/O devices are keyboards, mice, printers, and monitors. I/O designs are used to read and write data from a device to another. In this lab, students explore how the 68000 microprocessor interacts with I/O design. For this lab, the input device is a dip switch and the output are two 7 segment displays.

The MC68000 microprocessor is equipped with 3 interrupt signals (IPL0*, IPL1*, IPL2*) which provide 7 interrupt levels, level 0 being normal operating level. The status register contains three interrupt mask bits (I0, I1, I2) which are the logical complement of the interrupt hardware signals. The table below shows the list of interrupt level settings for the SANPER-1 ELU.

Interrupt Level	Interrupting Device	Vector No. (Decimal)
7	ABORT Switch	31
6	ACIA2 (Host Port)	30
5	ACIA1 (Terminal Port)	29
4	ACIA3, Speech, PIA	28
3	PI/T Parallel Ports $\overline{PIRQ}$	U.V
2	PI/T Timer ($\overline{TOUT}$)	U.V
1	System Expansion Board	A.V or U.V
0	Normal CPU Operation	Not Used

II. Lab Procedure and Equipment List

A. Equipment

Equipment

- SANPER-1 system
- PC with TUTOR software
- Breadboard
- DIP Switch
- 7-Segment Display (x2)
- ECE 441 Lab Kit (includes 74LS138, 7448, 74LS02, 74LS04, 74LS373, and different valued resistors)

B. Procedure

1. Implement the schematic on a breadboard
2. Connect the breadboard to the SANPER-1 ELU.
3. Use TUTOR's Memory Modify command to write data to the 7-segment displays.
4. Set the DIP switch to an initial value. Execute the program, and ensure the program correctly writes and the LCDs display the correct values as the program loops and writes values up until 99.

In this lab, the students started implementing the circuit on the first week following the schematic shown below. Once the circuit was completed, the program was launched but the LED segments did not light up. After an hour of debugging, the students realized that the 373 chips were not well wired and unfortunately, this could not be fixed due to time constriction. The second week of the lab did not take place due to classes moving online. To complete the lab, the only modification needed was to rewire the 373 chips. The rest of the circuit and the program seemed to work correctly.

III. Results and Analysis

A. Discussion

Programs

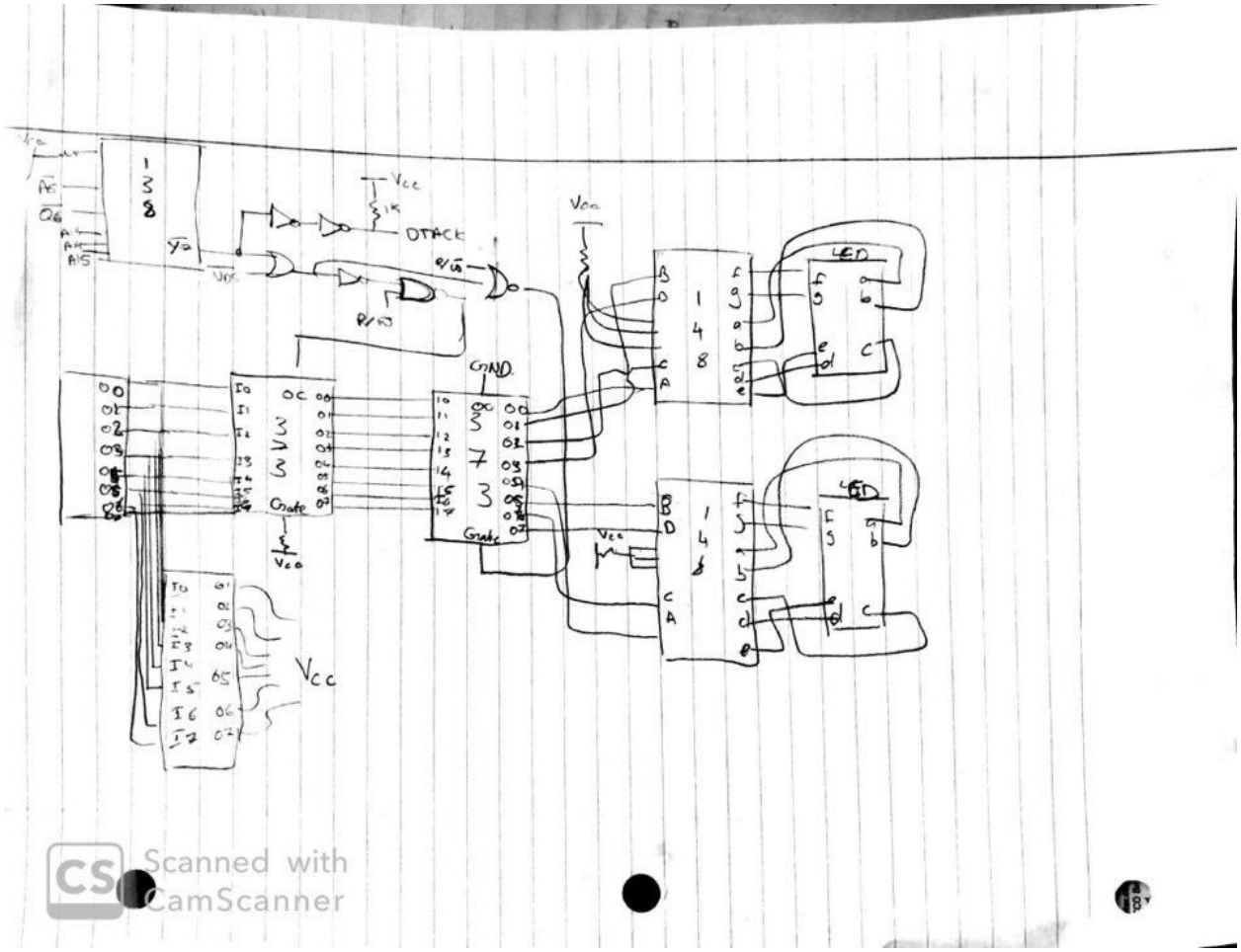
```
 *-----------------------------------------------------------
* Title      : Interrupt Service Routine
* Written by : Theo Gudiroz
* Date       : 03/11/2020
* Description: Read data from DIP switch, write data to two 7-segment
displays
*-----------------------------------------------------------
    ORG    $900

START:                  ; first instruction of program
    MOVEA.L #$6E000,A1      ;initialize the address
    MOVE.B (A1),D0          ;Read the value
    MOVE.L  #$1,D1          ;Initialize min
    MOVE.L  #$99,D2         ;Initialize max
    MOVE.W  #0,D3           ;Initialize delay
    MOVE.W  #$FFFF,D4       ;Initialize delay max

CHECK:
    CMP.B   D2,D0           ;Is counter max?
    BEQ     DONE            ;If yes, branch to done
    MOVE.B  D0,$6E000       ;Write to LED

WAIT:
    ADDI.W  #1,D3           ;Increment counter
    CMP.W   D3,D4           ;Is the delay done?
    BEQ     COUNT           ;If yes, branch to count
    BRA     WAIT            ;Else loop

COUNT:
    ABCD    D1,D0           ;Increment counter by 1
    MOVE.W  #0,D3           ;Reinitialize delay
    BRA     CHECK           ;Branch to check

DONE:
    MOVE.B  D0,$6E000       ;Write last value to LED
    MOVE.B  #228,D6         ;Exit to TUTOR

    END     START
```

Questions

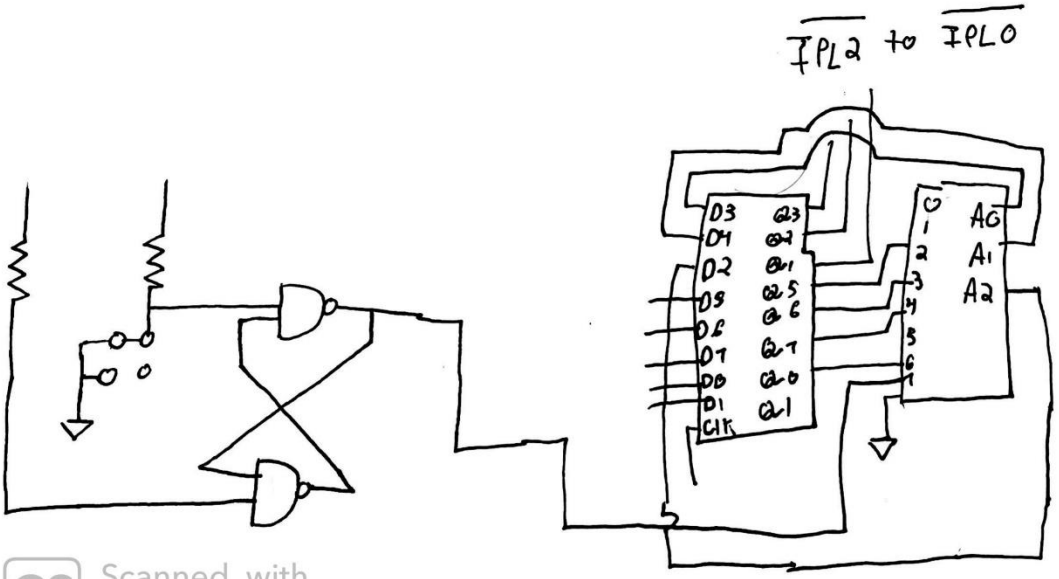
1. **A commented listing for the programs of Prelim #3.**

Section above.

2. **A schematic diagram of your hardware design for Prelim #1.**



3. **Review Chapter 7, "Hardware Description" of the Motorola Educational Computer Board User's Manual. Also, review Chapter 8 and examine Figure 8-3, Sheet 2 of 3 of the MC68000 Educational Computer Board Schematic Diagram, and redraw only the ABORT switch circuitry. Describe in detail how this ABORT Switch circuitry operates.**



The inputs to the Set/Reset are asserted low, and the flip flop sets the outputs to assert them high. The output is sent to pin 7 of the 74LS148 chip, which asserts the 3 outputs of the encoder all to high. These three outputs (A2 to A0) are fed into the 74LS273 (D Flip Flops) and the 3 outputs of these flip flops are all asserted low and fed into the interrupt control signals of the 68000.

4. **The Exception Vector Table has vectors assigned for Uninitialized Interrupts and Spurious Interrupts. Discuss how each of these types of interrupts occurs. Discuss the significance and applications of each of these types of interrupts.**

   If the CPU sends an interrupt acknowledge after receiving an interrupt request but no device responds to acknowledge, a spurious interrupt is generated. It prevents the CPU from waiting forever for an external device to respond. An uninitialized interrupt is generated when a peripheral has not been configured yet, but it generates an interrupt. Peripherals designed for 68000 based systems automatically supply the uninitialized vector 0x0F.

5. **Discuss the differences between Auto Vectored and User Vectored interrupts. For how many of each type does the MC68000 allow?**

   Auto-vectored interrupts are associated with peripherals designed for 8-bit processors because 8-bit processors cannot provide a vector during an IACK cycle. There are 7 auto-vectored interrupts. User-vector interrupts are intended for peripherals that can provide an 8-bit vector number and there are 256 of them.

6. **Discuss the events that occur during an Interrupt Acknowledge (IACK) Bus Cycle.**

   a. The peripheral signals on its interrupt line.
   b. The line is encoded into an interrupt level.
   c. The CPU completes the current instruction then saves its current state onto the stack.
   d. The level is compared to the SR.
   e. If the incoming level is higher, the interrupt is served.
   f. If the incoming level is equal or lower, no change occurs.

IV. Conclusions

In this lab, students understood the science behind the I/O mapped design. Given one more session, the group would have debugged the circuit and got the counter to work but due to unforeseen events, this did not happen.

References

[1] Experiment 5 Lab Manual