

# Lab #8

## *Interface to a 7-Segment LED*

### **PURPOSE**

This lab introduces interfacing to 7-segment LEDs. Upon completion of this lab, students will be able to do the following:

- Write programs that perform counting and code conversion using a 7-segment display.

### **PREPARATION**

Prior to the scheduled lab session, read the following section from your textbook:

- Section 9.7 (7-Segment LED Interface)

### **MATERIALS**

*Hardware:*

- 68KMB 68000-based computer
- I/O Board #2 for the 68KMB
- PC host computer
- RS232C serial interface cable
- 20-conductor ribbon cable

*MS-DOS Software:*

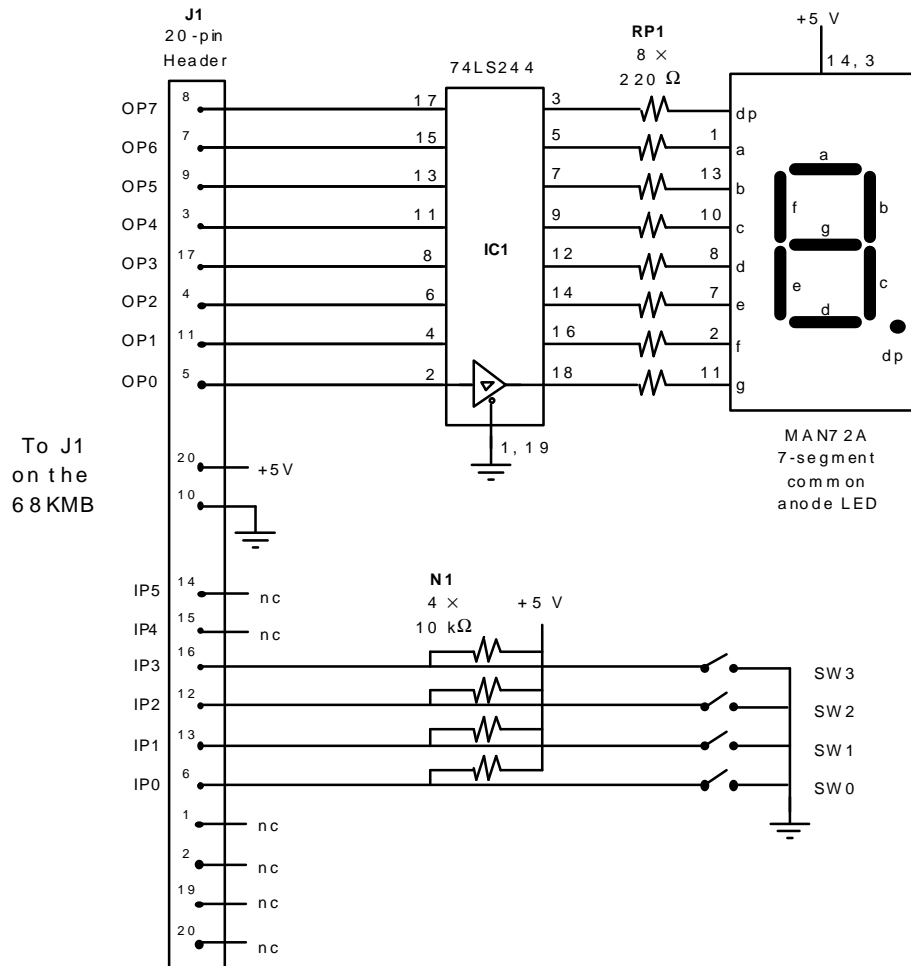
- A68K                      68000 cross assembler
- XLINK                     68000 linker, locator, conversion utility
- PC-VT                    VT100 terminal emulator
- EDIT                      MS-DOS text editor (or equivalent)

*68000 Programs:*

- LED7                      provided in 68KMB directory
- LED5HZ                  provided in 68KMB directory
- LED7A                    to be written
- RSEGMENT                to be written

## **INTRODUCTION**

This is the second in a series of labs to explore interfacing with the 68KMB. I/O Board #2 is a simple interface between the 68681 on the 68KMB and a 7-segment display and four switches. The schematic is illustrated in Figure 8-1.



**Figure 8-1. I/O Board #2**

I/O Board #2 is very similar to I/O Board #1. Instead of six input switches, I/O Board #2 only has four. These are used to input a 4-bit BCD or hexadecimal code. The output LEDs are combined in a 7-segment display driven by OP0-OP6. A decimal-point LED is driven by OP7. Electrically, the interface to the LEDs is identical to that in I/O Board #1.

## PROCEDURE

1. With the 68KMB powered-off, connect I/O Board #2 to J1.
2. Power-on the 68KMB and the PC host computer. Execute PC-VT and obtain the MON68K prompt from the 68KMB.

3. Use MON68K's memory modify command to access the 68681's output port registers. Experiment with turning on the segments of the LED display one at a time. For example, turn on segment *a*, then segment *b*, etc. Review lab #7, if necessary.

Demonstrate to your lab instructor that you are comfortable using MON68K commands to turn the LEDs on and off.

8.1 

4. An example program called LED7 is presented in Example 9-7 in your textbook. LED7 reads a 4-bit hexadecimal code from the input switches on I/O Board #2 and outputs the corresponding 7-segment pattern to the display. Review the software listing and the description of the program to gain an understanding of its operation.

Run the program and demonstrate it to your lab instructor. Be prepared to answer questions on the operation of this program.

8.2 

5. Make a copy of LED7.SRC and call it LED7A.SRC. Put your name and the date in comment lines at the top of the source file. Modify the new program such that only BCD codes are displayed. If a code in the range \$1010 to \$1111 is read from the switches, the display should appear blank.

Demonstrate the program to your lab instructor.

8.3 

6. An example program called LED5HZ is presented in Example 9-8 in your textbook. LED5HZ counts from 0 to F repeatedly at 5 Hz. The count is displayed on the 7-

segment display on I/O Board #2. Review the software listing and the description of the program to gain an understanding of its operation.

Run the program and demonstrate it to your lab instructor. Be prepared to answer questions on the operation of this program.

8.4 

7. Write a program to operate with the 7-segment display in I/O Board #2. The program should light segments one at a time, following a pattern around the outside of the display (*a*, *b*, *c*, ... *f*, *a*, *b*, and so on). The frequency of rotation should be 5 Hz. That is, segment *a* is on for 200 ms, segment *b* is on for 200 ms, and so on.

Bonus: Place the output instructions in an interrupt service routine and time the rotation using the 68681 timer. Example 9-6 in your textbook illustrates how to create timed interrupts using the 68681 timer. Review this if necessary.

Call the new program RSEGMENT. Put your name and the date in comment lines at the top. Demonstrate the program to your lab instructor.

8.5 

## CONCLUSION

This lab has introduced code conversions, counting, and timed I/O on the 68KMB using the 68681's parallel input port, parallel output port, and timer.