Department of Computer Engineering
Digital Hardware Systems
CpE 3104 - Microprocessors

Laboratory Report

Laboratory Exercise No.:	5	**Date Performed:**	10/11/2022
Laboratory Exercise Title:	I/O Interfacing		
Name of Student:	Paul John Toral	**Document Version:**	1.0

Activity #1

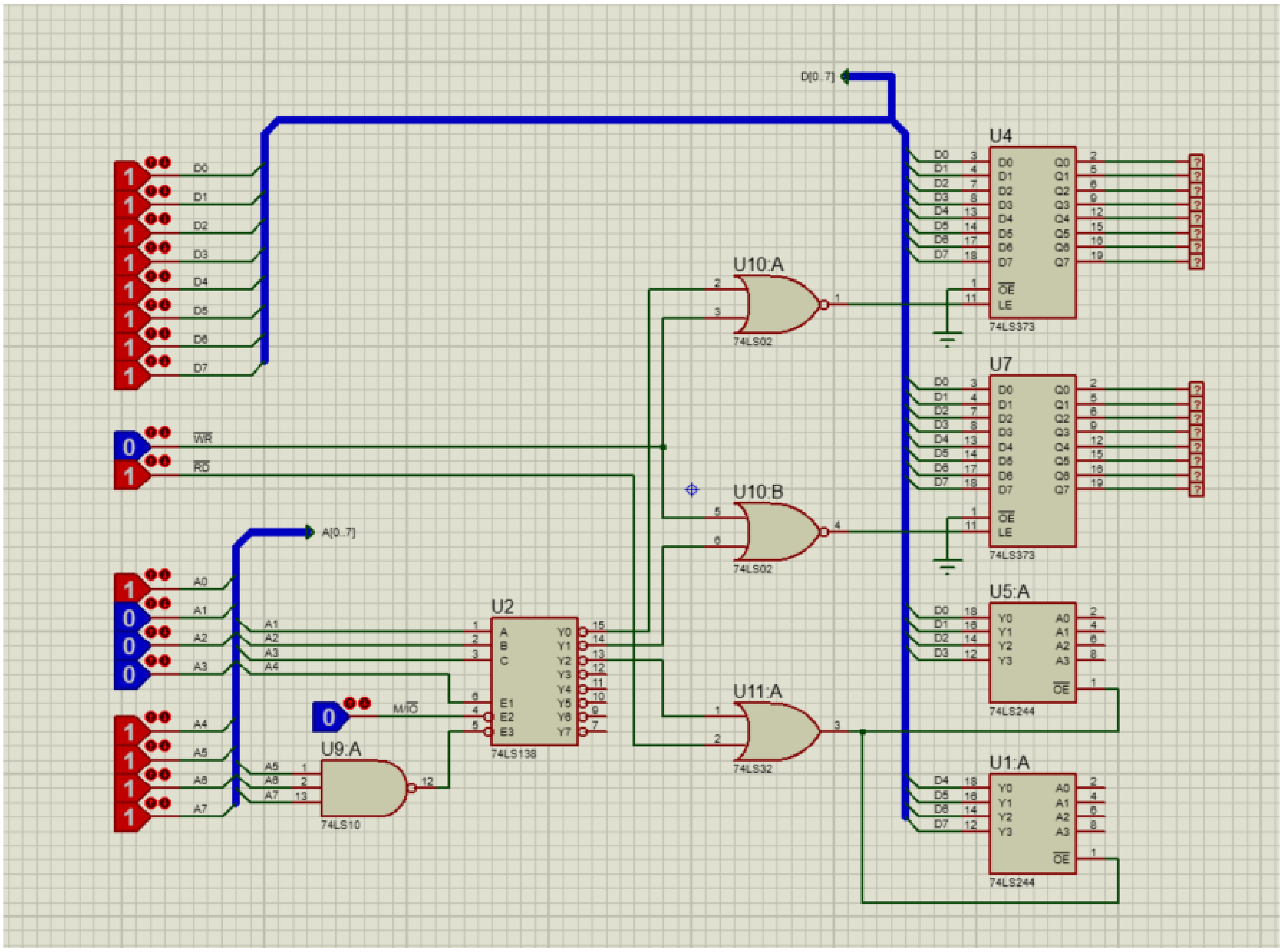


Figure 1. I/O Decoder

Table 1. Simulated Address Decoding and Ram write/read data

Address (A0-A7)	$\overline{WR}$	$\overline{RD}$	M/$\overline{IO}$	I/O Port Enabled
F0H	*0*	*1*	*0*	***PORTA***
F1H	1	0	0	**NO PORT ENABLED**
F4H	0	1	0	**NO PORT ENABLED**
F4H	0	1	1	**NO PORT ENABLED**
F5H	1	0	0	**PORTC**
F3H	0	1	0	**PORTB**
F2H	0	1	1	**NO PORT ENABLED**
02H	1	0	0	**NO PORT ENABLED**
65H	0	1	1	**NO PORT ENABLED**
F6H	1	0	0	**Command Register F6H**

3. Observe the data in Table 1. What is the role of the control lines, and in I/O address decoding?

   The control lines allow us to separate the set of addresses to the necessary ports we want to access. They help segment the necessary port to be accessed. We can see that when RD is 1 and WR is 0, the operation is to write data to the output, oppositely when WR is 1 and RD is 0 we see that we are reading from PORT C although it is not explicitly shown there. The M/IO pin was low enable which means that it needs to be logic low in order for the read and write operation to work .

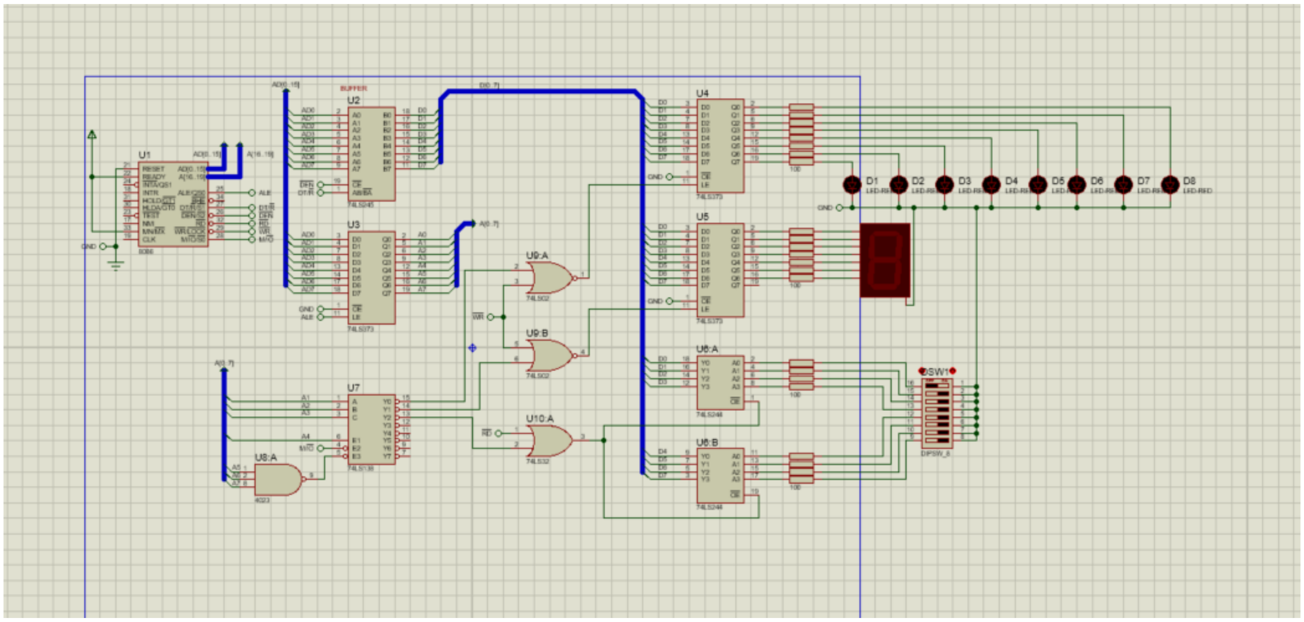
4. What do you think is the purpose of the latches and buffers?

   The main purpose of buffers is to pass data from its input to its output that is unchanged. On the other hand latches are flipflops that can be used to remember data or rather hold it so that it temporarily can be used in the future.

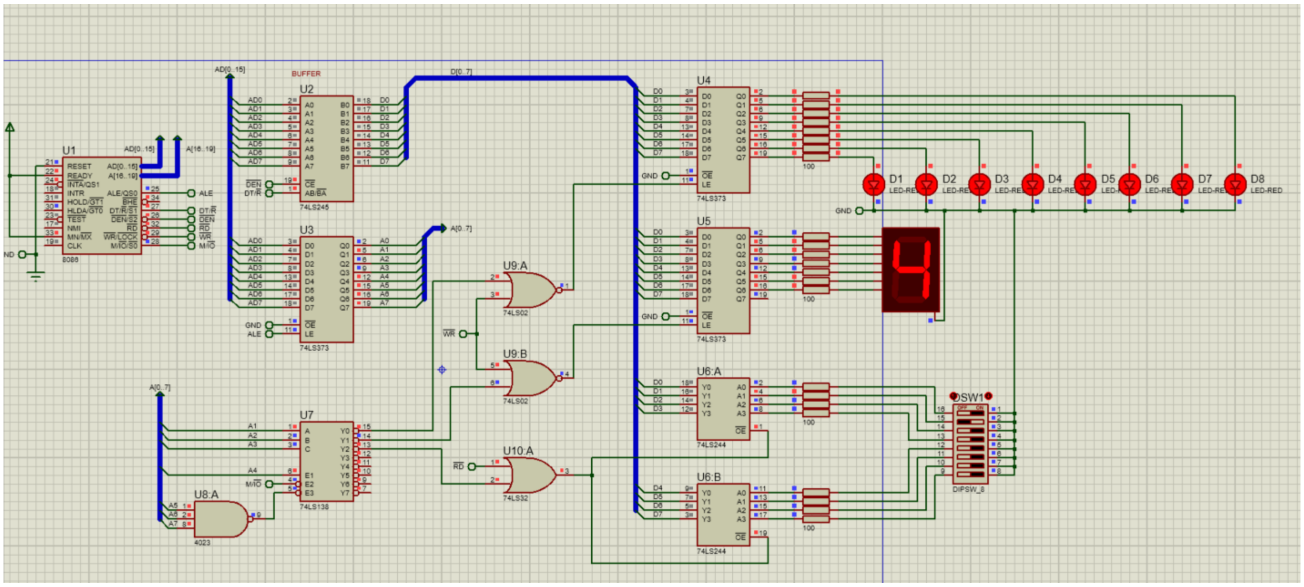
5. Based on the decoder circuit and I/O address range, is the I/O system “memory mapped” or “isolated”? Why?

   This is an isolated I/O system, as can be seen from the circuit architecture. This is because it has its own address space and is independent of the memory system (including the RAM and ROM).

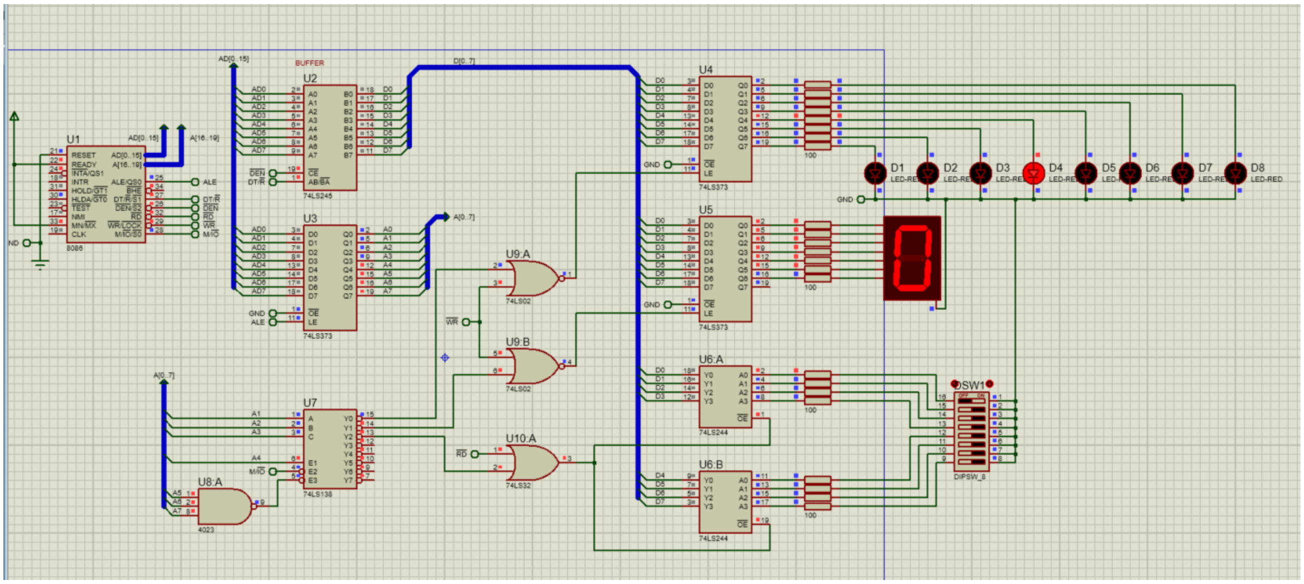
Activity #2



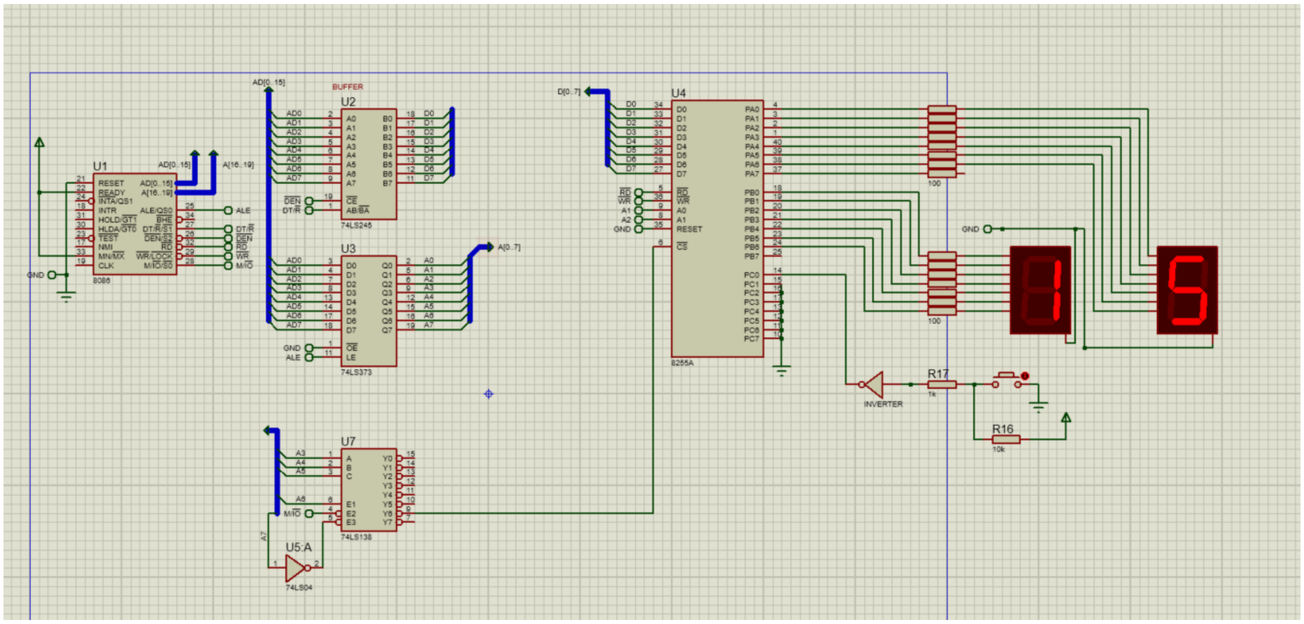
U1
8086
AD[0..15]
A[16..19]
BUFFER
U2
74LS245
U3
74LS373
D[0..7]
A[0..7]
U4
74LS373
U5
74LS373
U6:A
74LS244
U6:B
74LS244
U7
74LS138
U8:A
4023
U9:A
74LS02
U9:B
74LS02
U10:A
74LS32
D1
D2
D3
D4
D5
D6
D7
D8
LED-RED
DSW1
DIPSW_8
GND
ALE
WR
RD
M/IO
100



U1
8086
AD[0..15]
A[16..19]
BUFFER
U2
74LS245
U3
74LS373
D[0..7]
A[0..7]
U4
74LS373
U5
74LS373
U6:A
74LS244
U6:B
74LS244
U7
74LS138
U8:A
4023
U9:A
74LS02
U9:B
74LS02
U10:A
74LS32
D1
D2
D3
D4
D5
D6
D7
D8
LED-RED
DSW1
DIPSW_8
GND
ALE
WR
RD
M/IO
100



Activity #3:



3.

Address:

PORTA: **F0H**

PORTB: **F2H**

PORTC: **F4H**

COM_REG: **F6H**

4.

Command byte (in binary) 10001001

9, Personally I see that the 8255 peripheral is a nice and more convenient replacement of using primary latches and buffers for interfacing. It makes interfacing so much simpler. It can be seen that PORT A , B, and C are already found in the peripheral which saves us time in actually creating the circuitry.

10. Some advantages include a more convenient way to create circuitry, but on the other hand since it has fixed number of output and input ports, you can not adjust it. For example if you just need 4 pins from PORT B and there are 8 pins the 8255. Compared to actually creating the circuitry using latches and buffers we can adjust it according to the need through using different kinds of latches and buffers. Another great advantage of the 8255 is that you can program it to either be input or output depending on the command byte you said, this can not be done if you use regular latches and buffers.