ELEC 2607 – L9O
Lab 1 – Phone Switch

Due: February 1, 2008

Matt Taylor – 100711744
(Introduction, Specifications, Design)

Ruqia Nur – 100716528
(Implementing and Testing, Summary, Commentary)

Introduction

In this lab we designed and built a telephone switch. From the prelab, we learned about multiplexers (MUXs) and demultiplexers (DeMUXs). Multiplexers pass digital signals from one of several inputs to an output. The output is received by the demultiplexer and switches it to corresponding output. The telephone switch we designed uses a pulsed dialer, a 4 input MUX and an 8 output DeMUX. The dialer is used to select one of the 4 MUX inputs, which represents the calling party, and it selects one of the 8 DeMUX outputs, which represents the called party.

This lab is useful in learning how signals can be passed through logic gates to a specific output. The lab reinforced our knowledge of gates and logic. We applied material learned in class about Boolean algebra and circuit logic to design a working telephone switch. Telephone switches were used in the early stages of telephone communication; they were originally manually operated at a central location. The digital switch we designed is closer to our modern day telephone system; however, our circuit only allows signals to travel in one direction. A more complex system would need to be created to allow bidirectional transmissions and analog to digital converters to simulate actual telephone systems.

Specifications

The input for our telephone switch is a push button switch, which when pressed, changed the outputs on S_0 and/or S_1. When the signal was changed we saw one of the 8 LEDs representing the DeMUX light up. The circuit that was designed had to be built using the Tektron Logic Lab; therefore we were limited in the number of gates we could use. The numbers of available gates were recorded in the prelab, and when designing the circuit, we made sure there were enough available. We tried to make a simple design making it easier to debug.

Design

The prelab taught us that different circuits using different gates can have the same logic; this was very valuable when designing our MUX and DeMUX. There were several different designs for each, some more complicated than others.

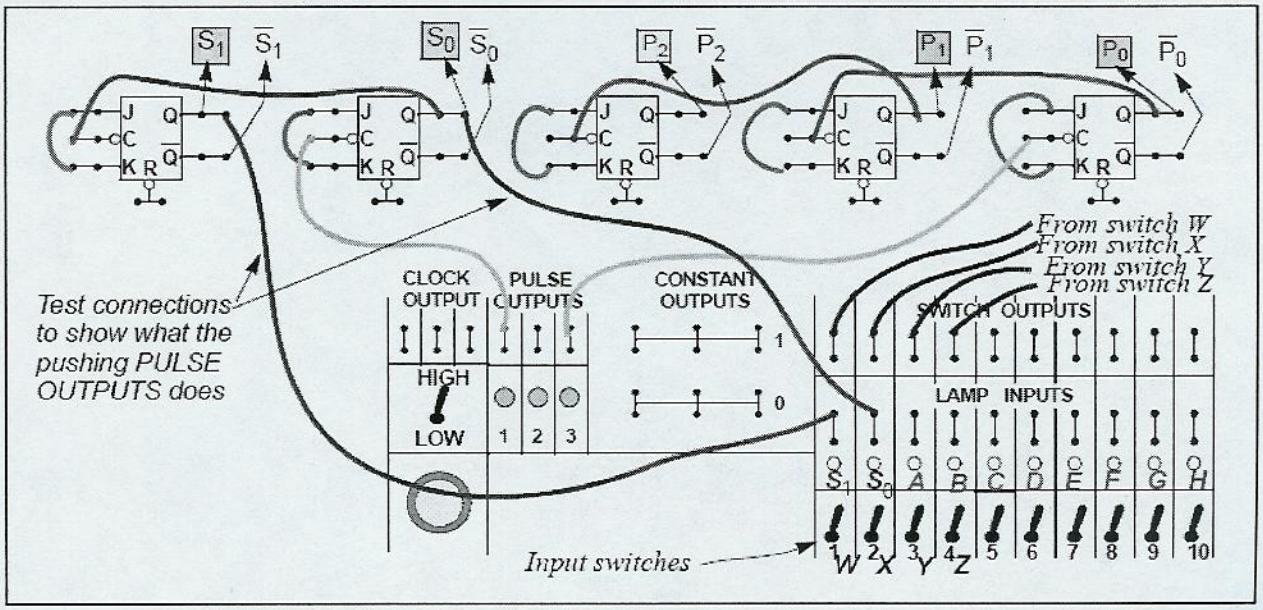
To design the 4-input MUX we decided to partition it into three 2-input MUXs. Two of the MUXs would feed into the third MUX, successfully creating a 4-input MUX. The most efficient design for the 4 input MUX makes good use of the Logic Lab board because it utilizes the boards Dual-Input AND-NOR gates. The MUX requires the use of 3 of these Dual-Input AND-NOR and 5 NOT gates. There are 6 NOTed inputs and outputs, however one NOT gate can be used for the S_0 inputs because 2 wires can be connected to NOT gate output.

We used a similar method for designing the DeMUX. The 8-output DeMUX was partitioned into one 2-output and two 4-output DeMUXs. The one 2-output DeMUX feed into two other DeMUXs, thus creating an 8-output DeMUX. The 2-output DeMUX was very simple and one required 2 ANDs and 1 NOT gate. Both of the 4-output DeMUXs were the same design; each used two of the widely available NAND gates, four NOR gates, and a single NOT gate. The design was simple and satisfied the specifications of the lab and was able to be created using the limited number of gates on the logic board.

The total number of gates used by both the MUX and DeMUX were available on the board, therefore our design was expected to work, and it did. The total number of gate required were as follows; 11 NOT gates, 2 AND gates, 4 NAND gates, 8 NOR gates, and 3 AND-NOR gates.

The pulse dialer circuit was supplied to us, see figure 1 for the wiring diagram provided by John Knight. The dialer was used as a clock counter to run the MUX and DeMUX. The clock counted in binary from 00 to 11, which selected a call party on the MUX. The clock for the DeMUX selected the called party. The counter was changed using the push button switch. See figure 2 for the schematic showing our design of the combined MUX and DeMUX.

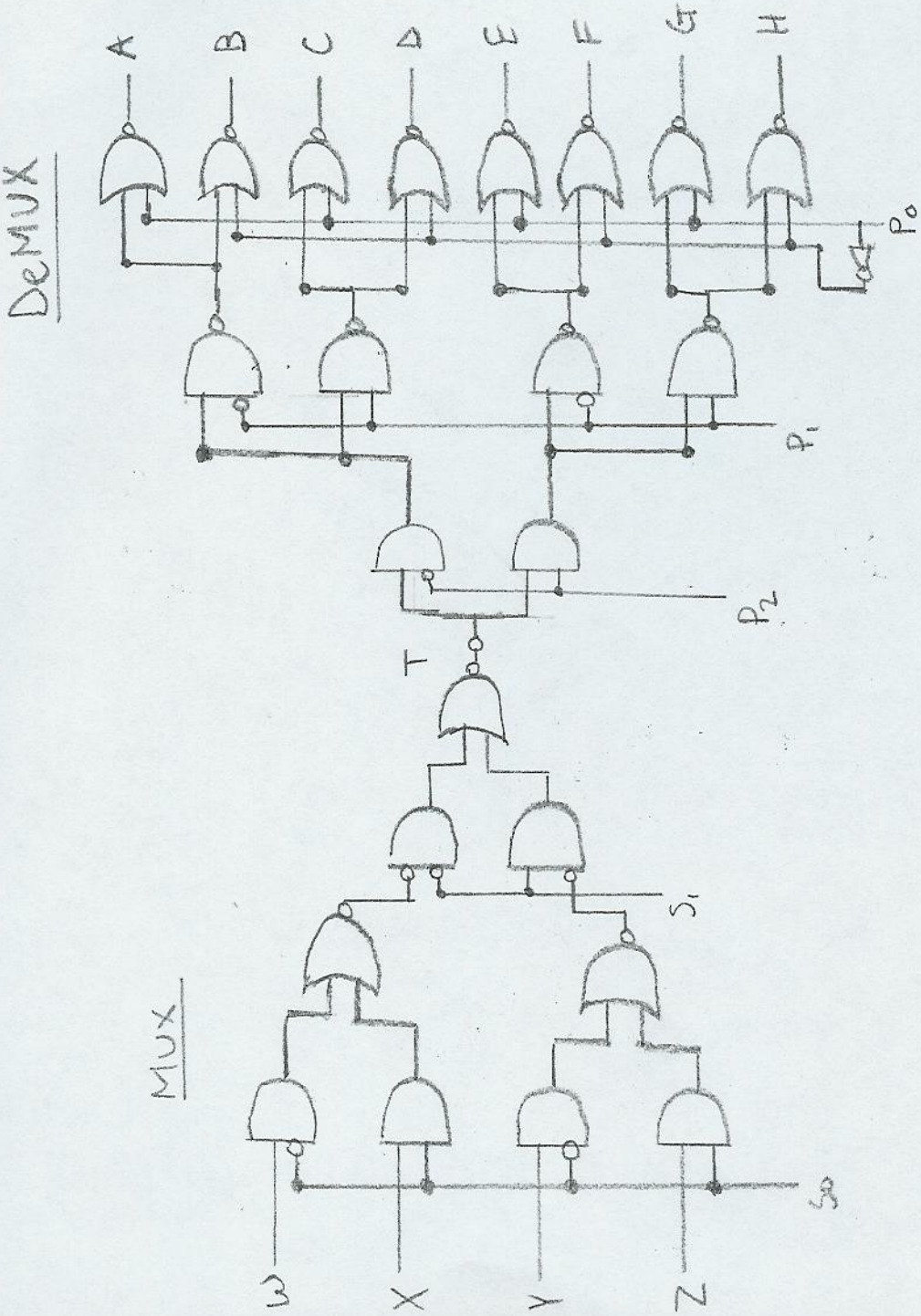
Figure 1: Pulse dialer wiring diagram provided in the Lab



Boolean equations ?

Truth table ?

Figure 2: 4-input MUX with 8-output DeMUX



Implementing and Testing

In this lab we used a pulsed dialer, a 4 input MUX and 8 output DeMUX to construct our telephone switch. The switch in its was built using the Tektron Logic Lab. Below is a list of parts that make the MUX and DeMUX.

MUX:

- 3 2-input digital MUX which consists of:
  - The control input S_0,
  - Inverter, which inverts S_0
  - S_0 and S_0 (inverted) are not considered as inputs hence two inputs
  - 2 AND gates for the inputs
  - 1 NOR Gate with the 2 AND gates as inputs.

DeMUX:

- 2 4-output DeMux which consists of:
  - 5 NOT, 2 AND, 4 NAND and 8 NOR gates
  - 4 NOR GATES are connected to 2 NAND gates and then fed into 1 of 2 AND gates, the other and gate is fed the same inputs.

Figures 1.1-1.3 are images of the MUX and DeMUX

In conducting this lab we were quick to notice the key importance of the wires, because we were only given a certain number of wires with varying lengths. This meant that we had to be careful of which wires we used where, and only using the longer ones where absolutely necessary. Fortunately we did not run into problems with wires because we recognized this problem early on.

Debugging this lab would be a very tedious process if the logic was not looked at. In the case of our lab, we had an issue where one wire was plugged into the wrong spot, and as a result we had every second light being skipped. We looked at what each of these outputs had in common logically and noticed that one of our S_0 inputs was being fed the wrong input. In this case we fixed this problem and it worked. This is the same approach that should be taken when debugging a circuit board. Rewiring would be very tedious.

Summary

The digital phone switch that we designed and created worked very well with little difficulty.

When designing the phone switch there were a few ways in which the end result could have been achieved. The multiplexer was created by using 2 dual input AND-NOR gates as the input to a NOR gate and an inverter was used to make the NOR and OR gate. However the same thing could have been done by using a single AND and OR gates. This was not a desirable choice because this requires a lot more wires and therefore more possibilities of errors. The demultiplexer was created using 4 NOR gates as inputs to 2 NAND gates and this was all used as 1 of 2 inputs for the AND gates (connected to the lamp inputs). It is clear that more single gates were used then in the multiplexer.

When conducting this lab, some impromptu innovation was conducted. This included using certain colors and lengths of wires for certain parts of the lab. We were quick to realize that the long wires should only be used where needed otherwise it would make the circuit hard to observe, as well as only a certain number of wires were given and if used carelessly you would run out.

Throughout the prelab, many different possibilities to create the gates needed in the MUX and DEMUX were conveyed. Relating to the course material, Demorgan's theorem could have been utilized in the manners listed below to achieve the same outputs (use of different logic gates):

1. NOR gate = AND gate with inverted inputs
2. NAND gate = OR gate with inverted inputs

In the end we learned how to make a MUX, DeMUX as well we learned how to create a switchboard and in the process used logic skills that were taught in class. The design of this lab was suitable for applying the material in the course.

However, the only suggestion would be to try and design a switch, which used a least amount of wires. The toughest part of this lab was trying to use the available gates to make the dialer and making sure that there were enough gates to complete the lab.

Commentary

3.0 An application in which one way signals save using large multiconductor cables is with television cable (CATV). This is because they can use cheaper wires as well, to reach an entire city, which allows more people to have access at a much cheaper cost. Another application of a one-way signals is the FM radio, however they use radio waves to reach listeners.

3.1 To build a system with 8 calling inputs, we could use two 4-input MUXs, the same type as the one designed above, which feed into a 2-input MUX. There would be 3 signals to allow for 3 bits of switching capacity, S_0, S_1, and S_2. See figure 3 for our designed schematic of how to build an 8-input MUX. This design could not be built using the Logic Lab board because there are not enough dual-input AND-NOR gates, other designs of the MUX could be used and built on the board, however it is unlikely that both a 8-input MUX and 8-output DeMUX could be built on the same board.

3.2 A broadcast call switch button will when pressed, cause all the output lights to turn on. For a 4-output DeMUX, the T output from the MUX will go into a AND gate, the other input on the AND will be from the push button broadcast switch. The output from the AND gate will travel into a 5-input OR gate, the other inputs will be A, B, C, and D from the DeMUX. The output from this OR gate will be split and connected to the LEDs from A, B, C, and D.

3.3 To dial any receiver on two labs, the Tektron boards need to be connected. This can be achieved by using the S2 and P3 switches and connecting the trunk line from Board 1 to MUX1 which outputs to MUX2 which is connected to both boards, as well as MUX1. See figure 4 for a schematic showing how to connect the lab boards.

3.4 In order for a hang up, the values of S_1 and S_0 must be set to 00. However, the reset switch which is wired into the JK flip flops which results in an inversion. This entails that the value of R be 0 in order for there to be an input. This is done by hooking the switch into a pulse output. Therefore, when the button is up it outputs a 0 and when it is down is out puts a 1. To hang up the button needs to be pressed down and then released.

Although the content of the lab was not difficult, there were some tricky parts to the lab. Wiring was an issue because there was a limited number of wires as well as sizes. repeated To make this lab better I would give larger wires incase a group runs out and has to take parts of the completed dialer apart and switch wires.

Figure 3: Schematic of our designed 8-input MUX

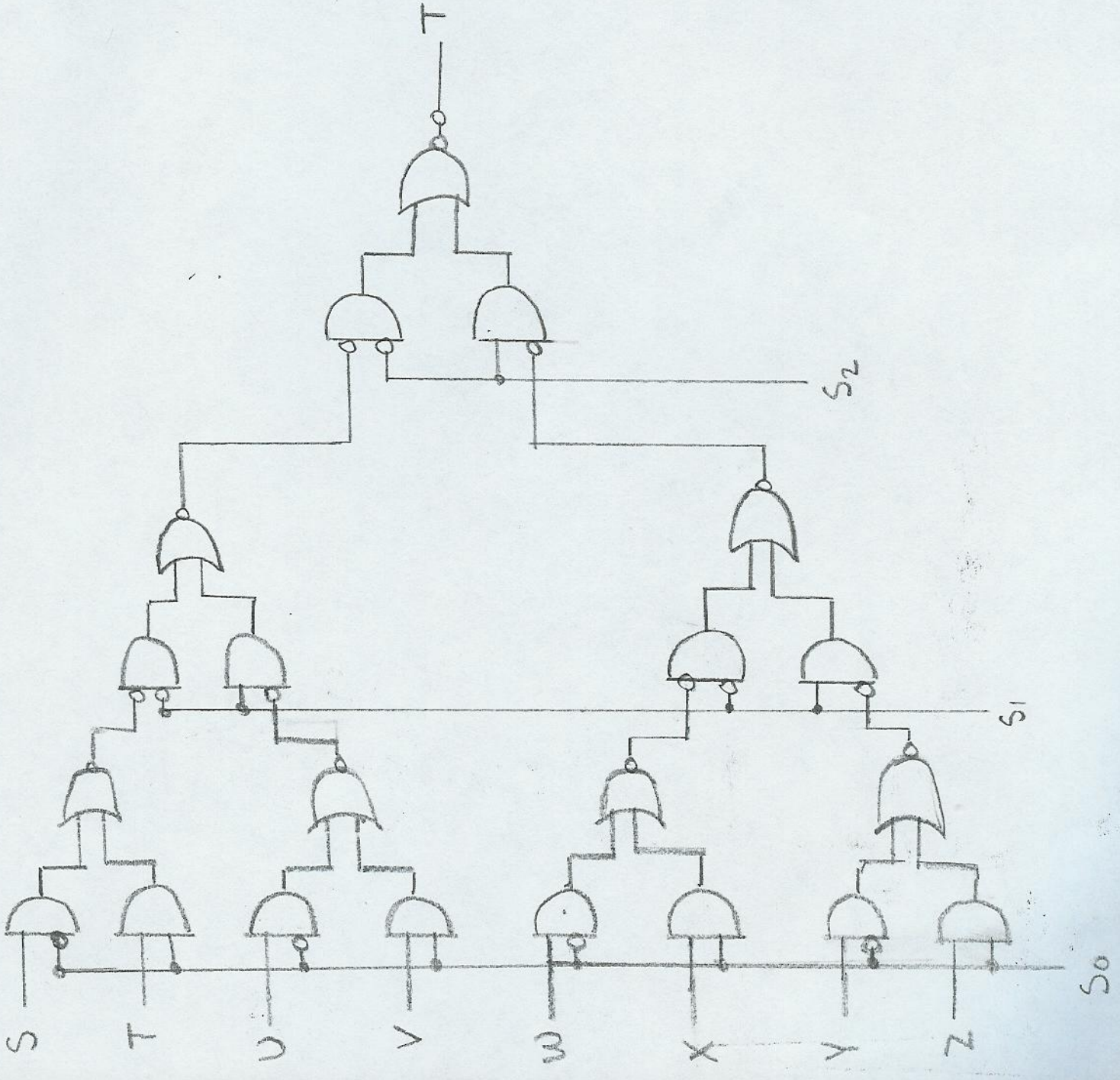
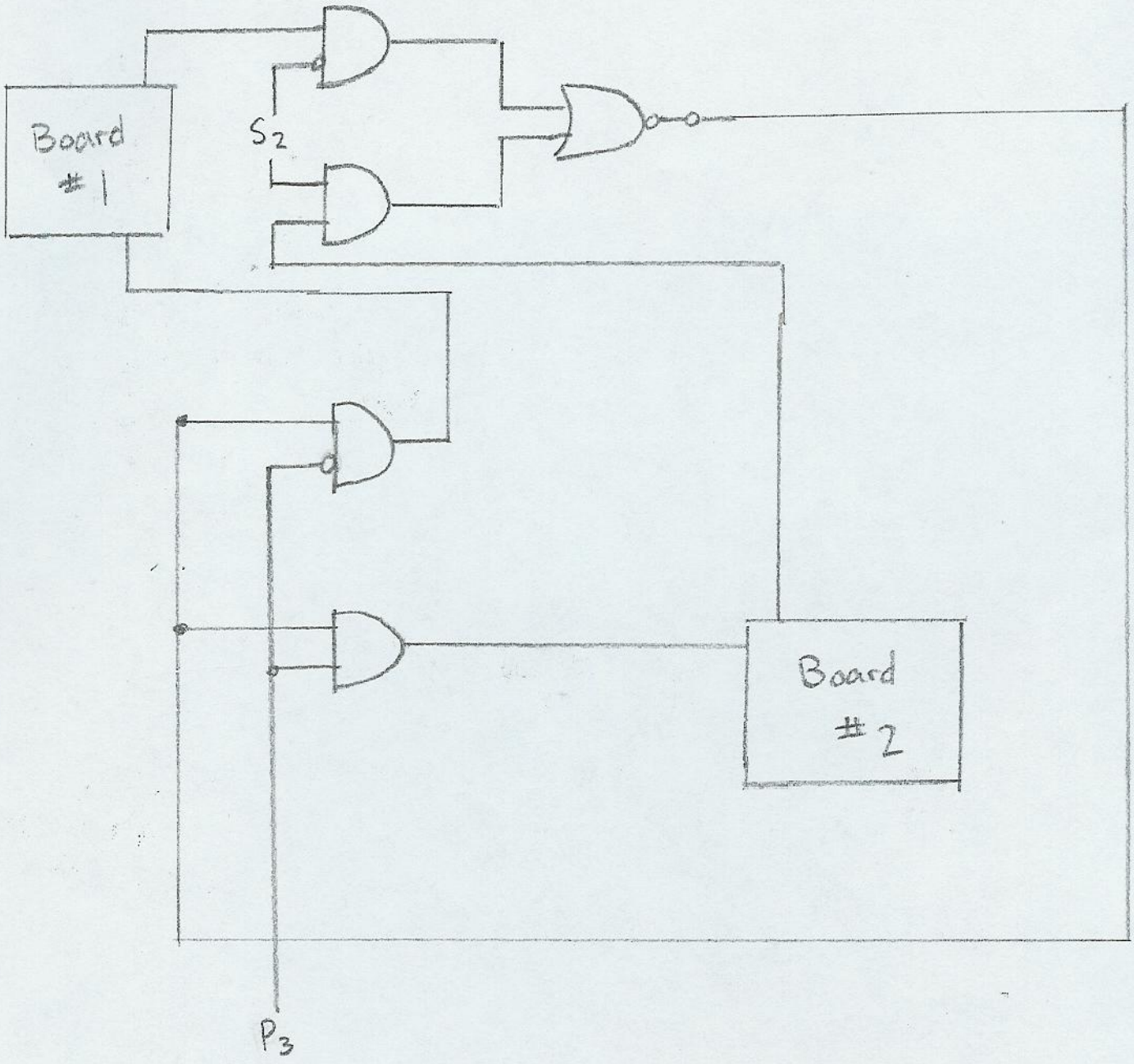


Figure 4: Schematic showing how to connect two Tektron boards



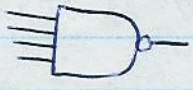
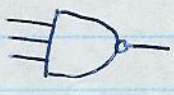
Matt Taylor
100711744

Elec 2607 - Lab 1 - Phone Switch

1.1 $A+B+C+D = (A+B)+(C+D)$

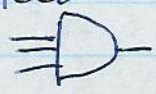
- Associative law

1.2

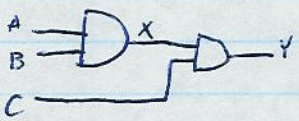
have  need 

- have one input on the 4-input NAND a logic 1

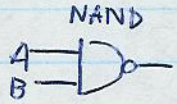
1.3

need  have  x2

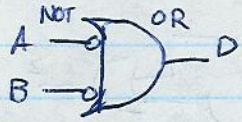
- arrange circuit:



1.4



$\overline{A \cdot B}$ gate



$\bar{A}+\bar{B}$ gate

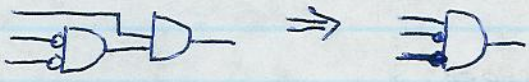
A	B	$A \cdot B$	$\overline{A \cdot B}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

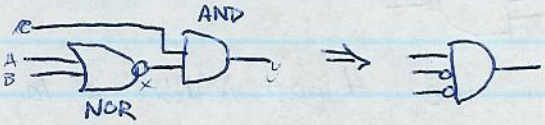
A	B	$\bar{A}$	$\bar{B}$	$\bar{A}+\bar{B}$
0	0	1	1	1
0	1	1	0	1
1	0	0	1	1
1	1	0	0	0

- both tables are the same therefore gates are the same

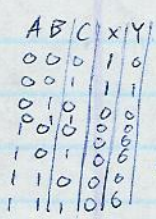
1.5 no question.

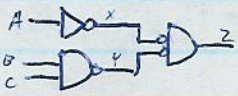
1.6 a)  3 input AND (M) ✓

b)  3 input AND w/ 2 NOT inputs (J) ✓

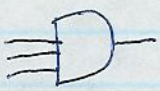
c) 

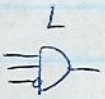
=

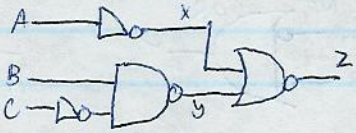


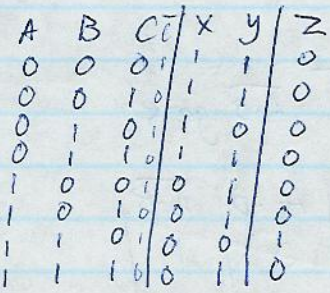
d) 

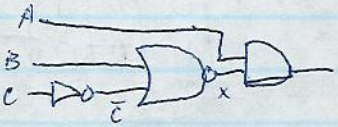
K 

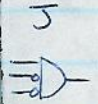
AND  (M)

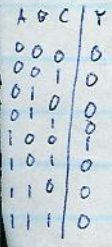
L 

e) 

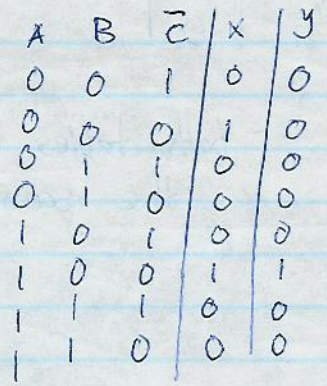
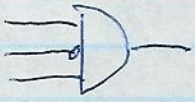
 (L) 

f) 

J 



M 

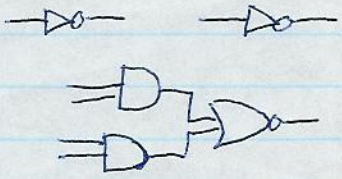
 (K) 

1.7 2-input AND 3-input AND 2-input OR 2-input XOR

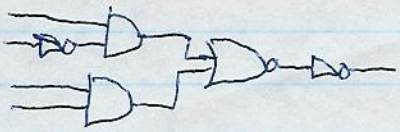
dual 2-input AND-NOR 2-input NAND 3-input NAND

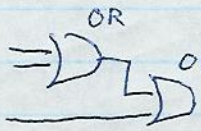
4-input NAND 2-input NOR NOT

1.8 given

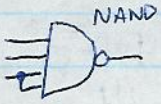


2-input MUX:



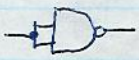
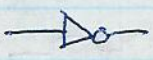
1.9 (A) 

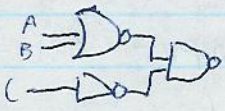
= (V) 

(B) 

= (X) 

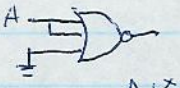
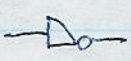
A	X
0	1
1	0

(C)  = (W) 

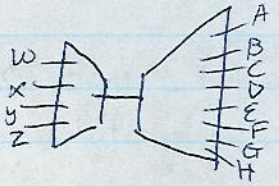
(D) 

= (Z) 

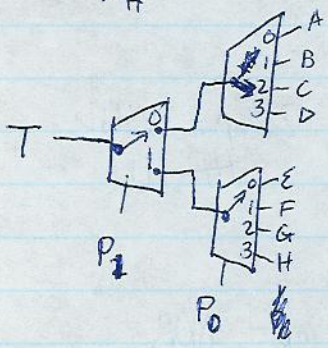
A	B	C	X	Y	Z
0	0	0	1	1	0
0	0	1	1	0	0
0	1	0	1	0	0
0	1	1	0	0	1
1	0	0	0	0	1
1	0	1	0	0	1
1	1	0	0	0	1
1	1	1	0	0	1

(E) A  = (W) 

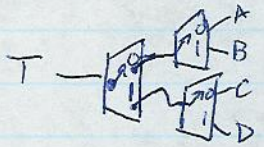
A	X
1	0
0	1

1.10 

partition into 4 input MUX
one 2 output DeMUX
two 4 output DeMUX

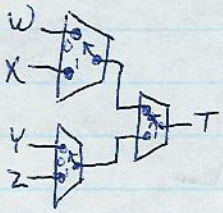


1.11 4 output DeMUX into three 2 output DeMUX

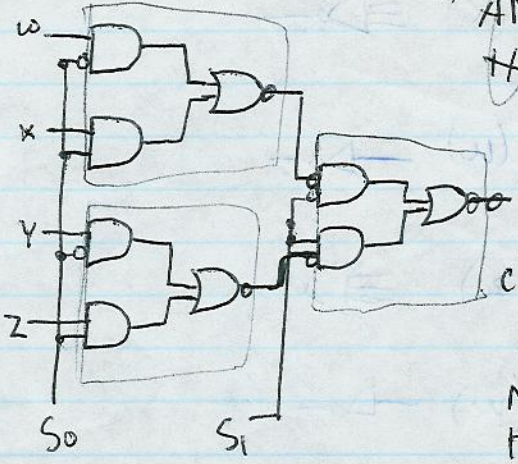


1.12 4-input MUX

a) partitioned MUXs



b) schematic



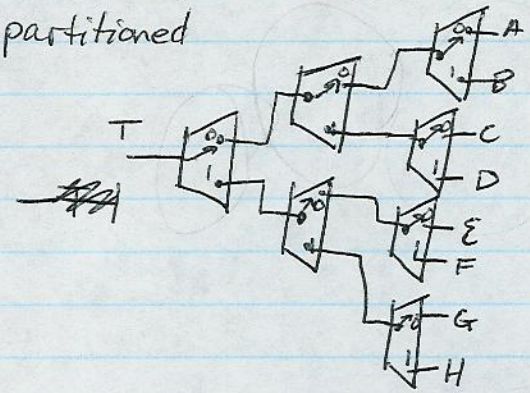
c) AND NOT OR
1111 111 1

c) DUAL INPUT AND-NOR
111

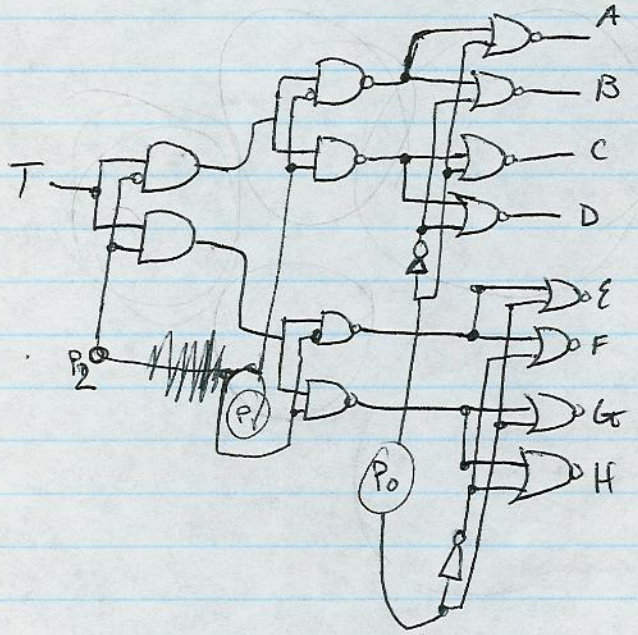
NOT
1111

1.13 8 output DeMUX

a) partitioned



b)



NOT	AND	NAND	NOR
~~HH~~	11	1111	~~HH~~ 111

1.14

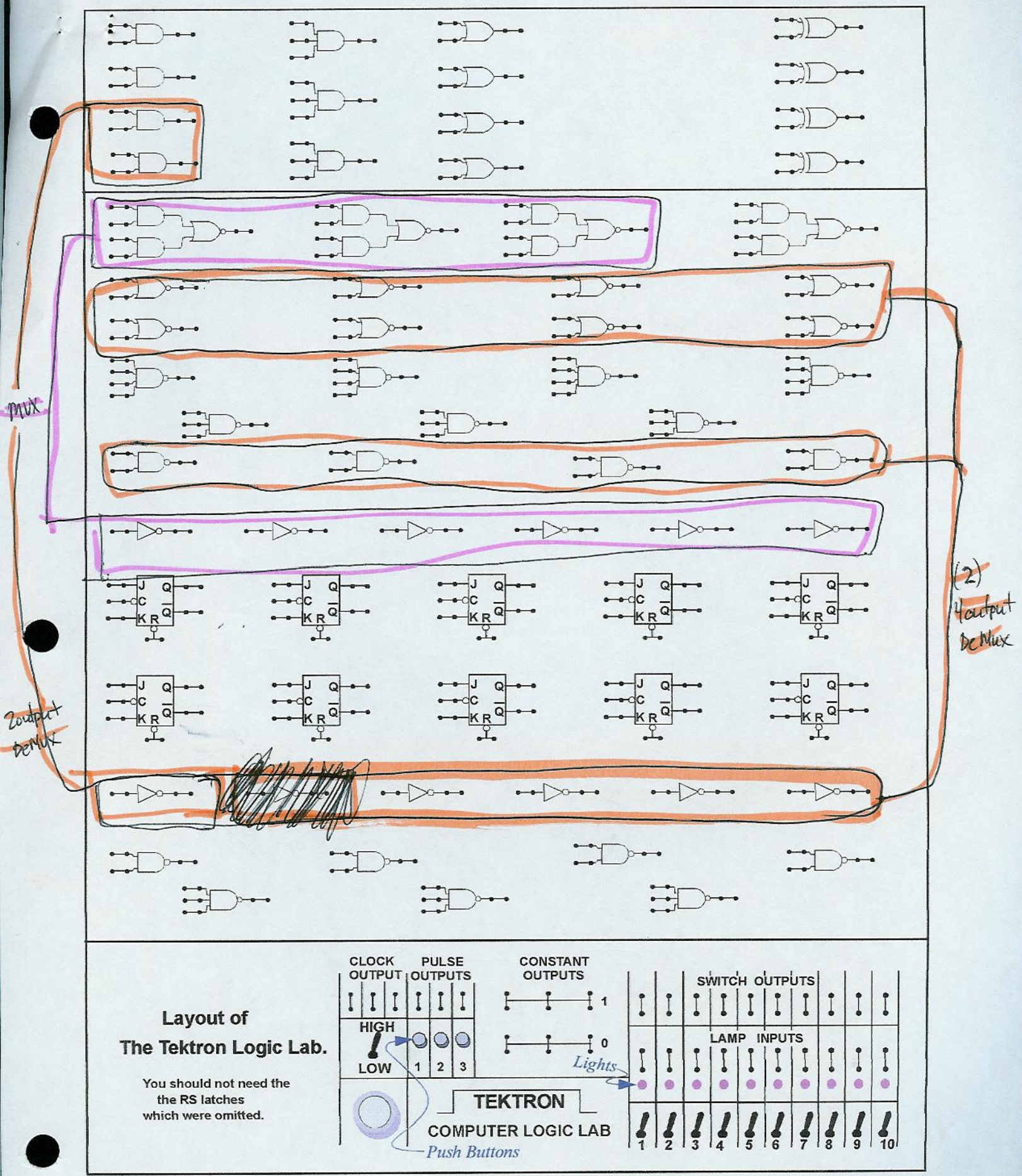
NOT ✓	AND ✓	NAND ✓	NOR ✓	AND-NOR ✓
~~HH~~ ~~HH~~ 1	11	1111	~~HH~~ 111	111

1.15 printed

1.16 see attached

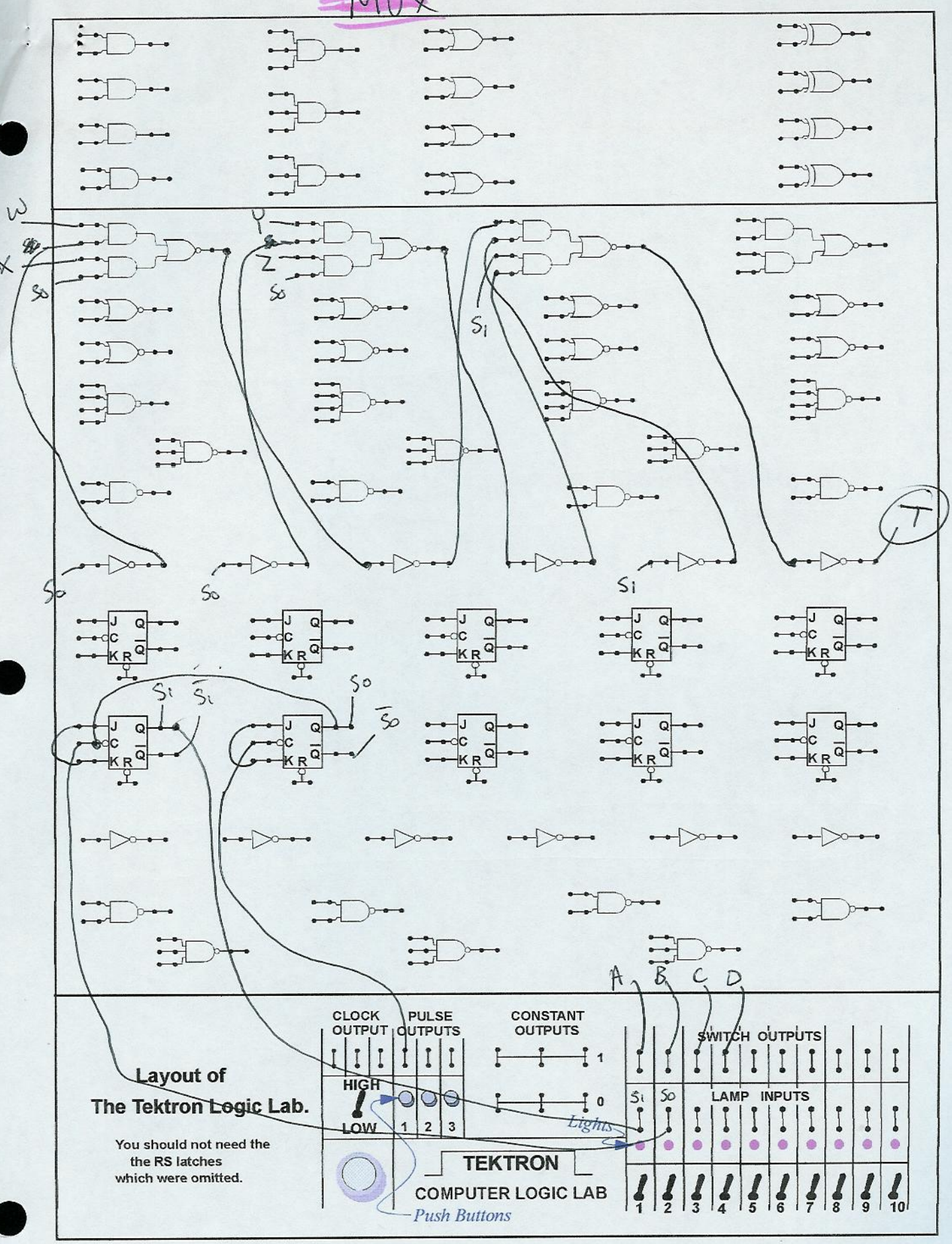
1.17 see attached

1.18 OK

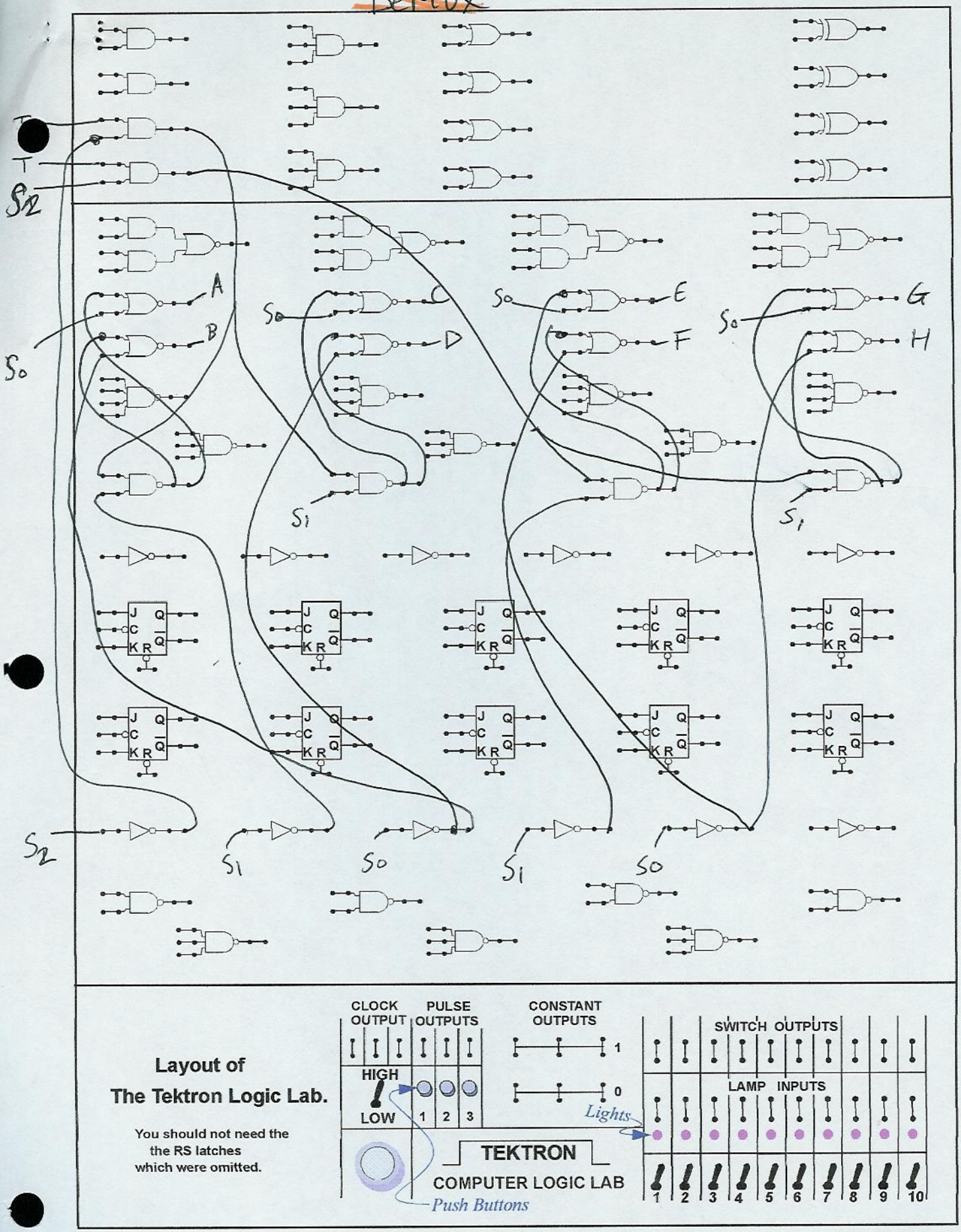


Layout of The Tektron Logic Lab.

You should not need the the RS latches which were omitted.


MUX
Layout of
The Tektron Logic Lab.
You should not need the
the RS latches
which were omitted.
CLOCK OUTPUT
PULSE OUTPUTS
CONSTANT OUTPUTS
HIGH
LOW
1
2
3
1
0
Lights
SWITCH OUTPUTS
LAMP INPUTS
TEKTRON
COMPUTER LOGIC LAB
Push Buttons
1 2 3 4 5 6 7 8 9 10

DeMUX



Layout of The Tektron Logic Lab.

You should not need the the RS latches which were omitted.