

Laboratory 1. A Telephone Switch

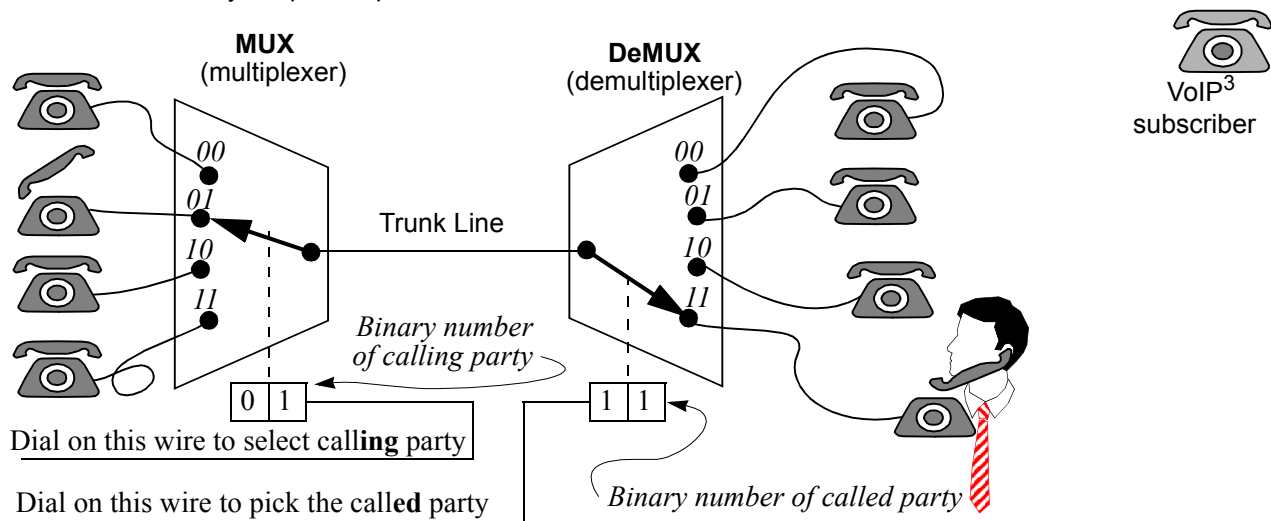
1.1 Overview;

In this lab you will build a telephone switch¹ that connects one subscriber to another. Since this is only a three-hour lab, we will make a lot of simplifications:

- The phone signals will have to be digital. In fact you will generate the signals by flicking a switch on and off very fast, and pretending it came from a microphone connected to an analog-to-digital converter.
- The conversations will go in only one direction, e.g. the person at the other end can't argue
- You will have four talking subscribers and four (later eight) listening subscribers.

The first system we will consider (Fig. 1), will allow only one conversation on the system at once. One could place a second system in parallel to allow two conversations (Fig. 2). Again you can imagine adding more parallel systems to allow more people to talk simultaneously. Telephone systems were typically designed to allow 10% of the subscribers to talk at once². They have 10 switches for 100 phones.

FIGURE 1: An overly simple telephone switch.

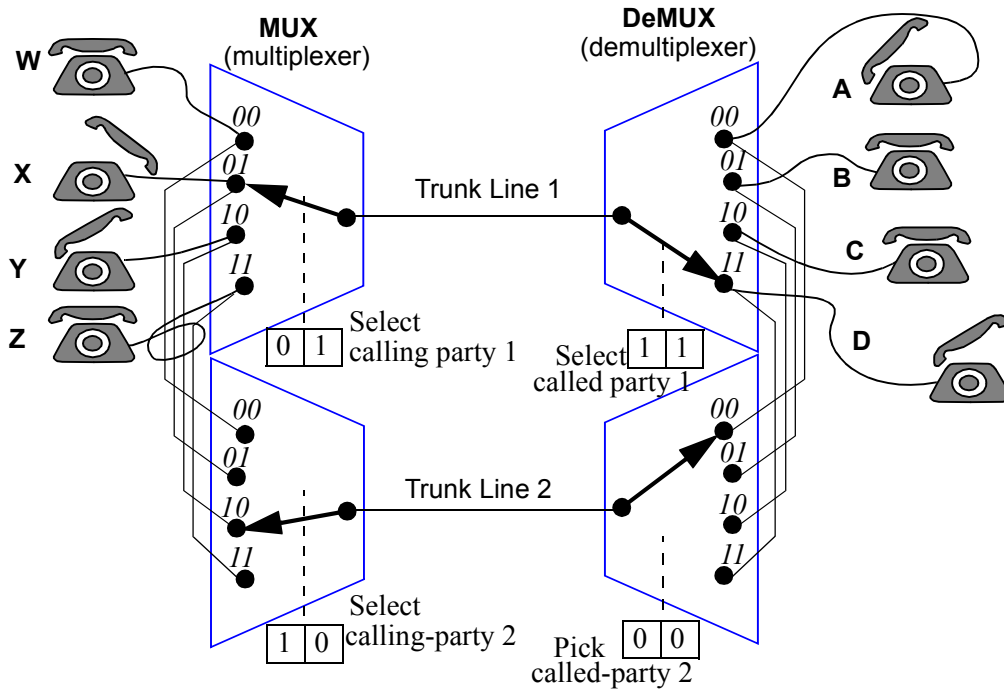


1. A good description of this type of telephone switch is given by Mark Csele at Niagra College: <http://technology.niagarac.on.ca/staff/mcsele/TelephoneSwitch.html>

2. It is highly unlikely that more than 10% of the phone lines are in use at once. However when the University of Calgary first put in its telephone registration system, so many people called it the first hour that the whole telephone exchange, which included the Foothills Hospital, was blocked for any new calls.

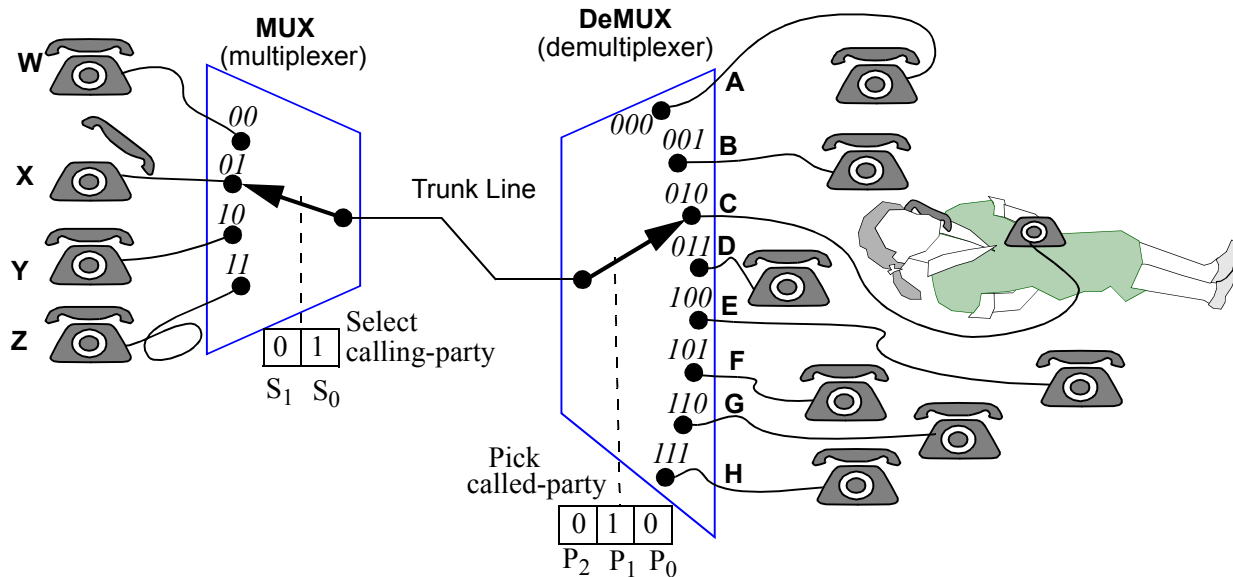
3. VoIP stands for voice over internet protocol and is the author's weak attempt at humor.

FIGURE 2: Placing two switches in parallel to allow two subscribers to make calls. One could place more switches in parallel but that gets to complicated for this lab.



A slightly different system could have eight called-subscribers with four calling-subscribers. This is a good system for telephone solicitation e.g. for sales people who call you at supper time.

FIGURE 3: Switch with 4 calling parties and 8 called parties.



In this lab you will design and build a Fig. 3 type of telephone switch using digital gates which are part of the Tektron Logic Lab trainer. These Labs, especially the digital gates they contain, are described on page 11 in this lab description.

1.2 Boolean Algebra Summary

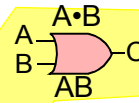
Boolean Algebra allows signals (variables) to be only 1 or 0.

We do not allow ~~2, 3, 14159, >1, 1+j1~~

Operations:

OR, not really addition, but written as + anyway;

$A+B = C$ means $C=1$ if either A OR B OR both are 1

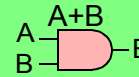


A	B	0	1
0	0	0	1
1	1	1	1

Inside the box shows the value of $C=A+B$

AND, not really multiplication, but written as \cdot or side by side;

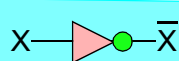
$A \cdot B = E$ or $AB=E$, means $E=1$ if A AND B are both 1.



A	B	0	1
0	0	0	0
1	0	0	1

value of $E = A \cdot B$

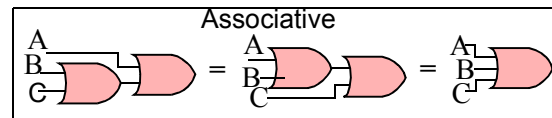
NOT, written as an overbar, \bar{x} ;
 $\bar{\bar{x}}$ = the opposite of x.



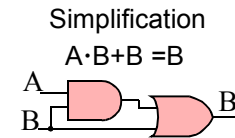
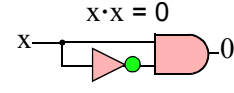
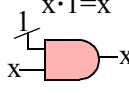
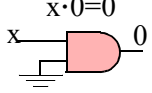
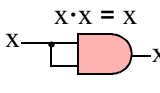
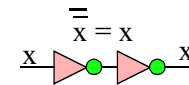
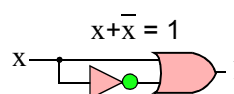
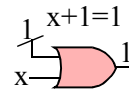
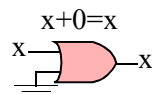
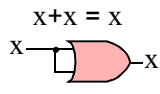
X	0	1
1	0	1

value of \bar{X}

Associative $(A+B)+C=A+(B+C)$, $(A \cdot B) \cdot C = A \cdot (B \cdot C)$
Commutative $A+B = B+A$, $A \cdot B = B \cdot A$
Distributive $A(B+C) = A \cdot B + A \cdot C$



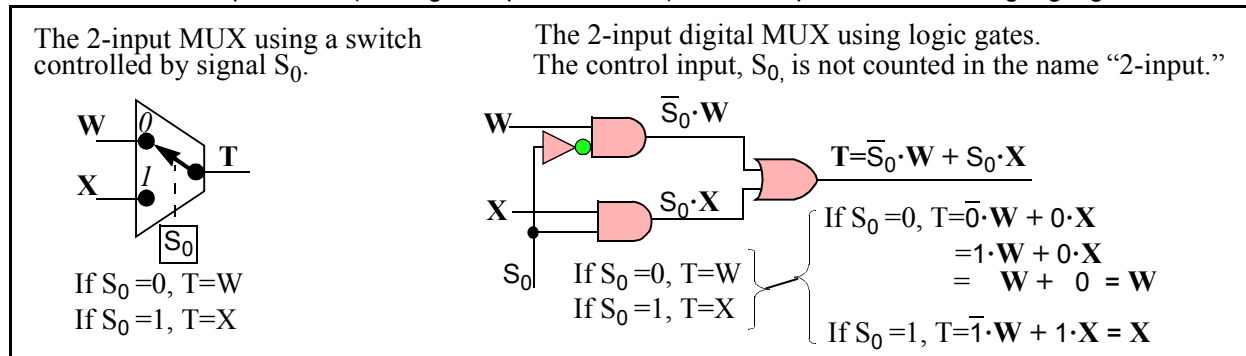
Theorems:



1.3 The Multiplexer (MUX)

Another type of gate which is very common, especially in this lab, is the multiplexer or MUX which is used to select one of several inputs to send to an output. The MUX can be made from the more basic AND OR and NOT gates.

FIGURE 4: The 2-input MUX (two signal inputs W and X) and its implementation using logic gates.



Read the equations in Fig. 4, until you understand the algebra of how T is switched between W and X.

The digital MUX has the disadvantage that it can only pass digital, not analog, signals. Further it can only pass these in one direction. These can both be overcome but not in this first lab. Wait until the transmission gate MUX is described in third year.

FIGURE 5: To construct a larger MUX we will use gate symbols with more than two input and with some of the inputs inverted..

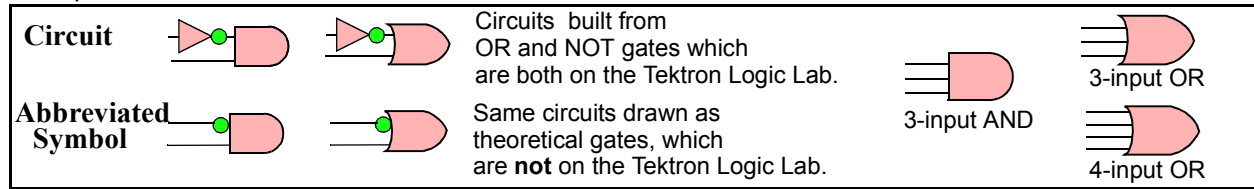


FIGURE 6: The 4 input MUX and its implementation using three 2-input MUXs.

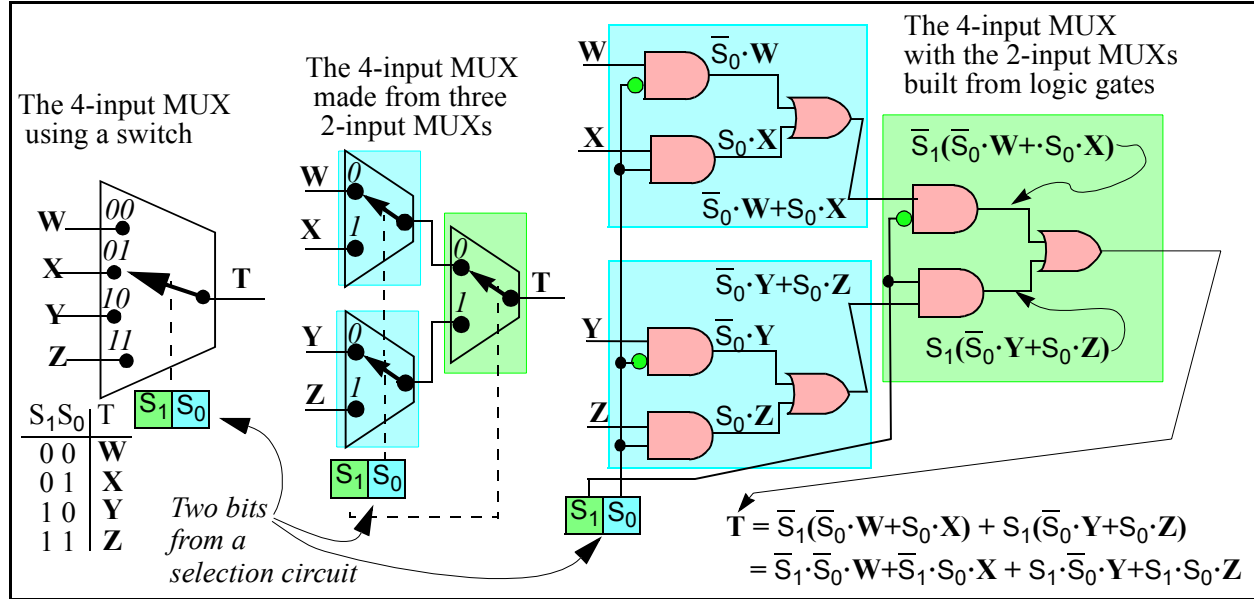
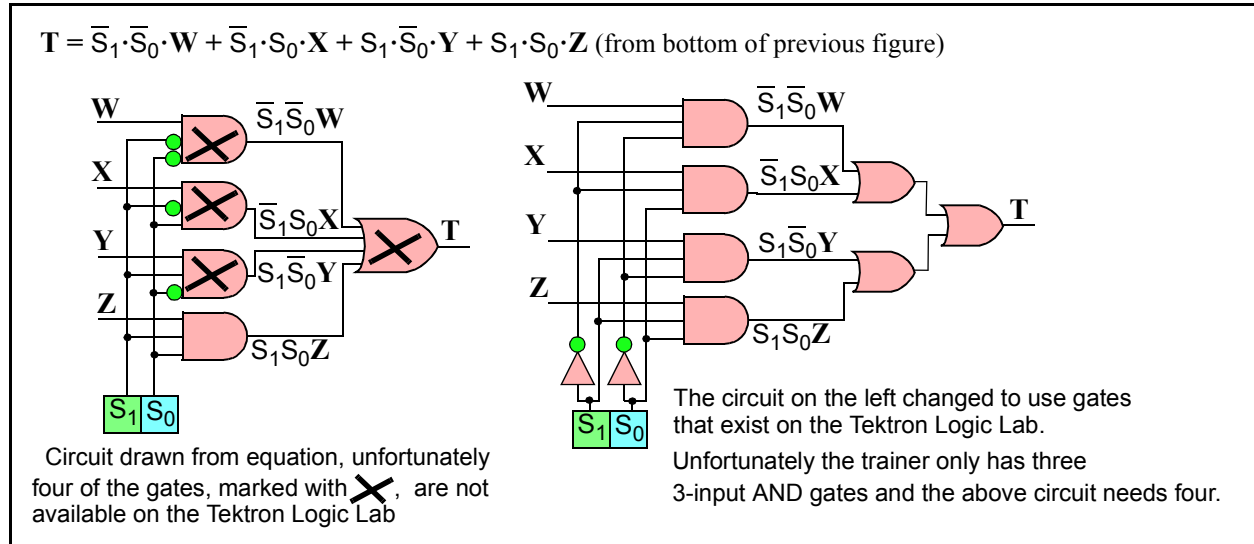


Fig. 6 shows one way to build a 4-input MUX out of gates, but it is certainly not the only way, and may not be the best implementation to build it in this lab. If one takes the equation for T derived at the bottom of the the figure, one can also implement the 4-input MUX with 3-input gates (Fig. 7).

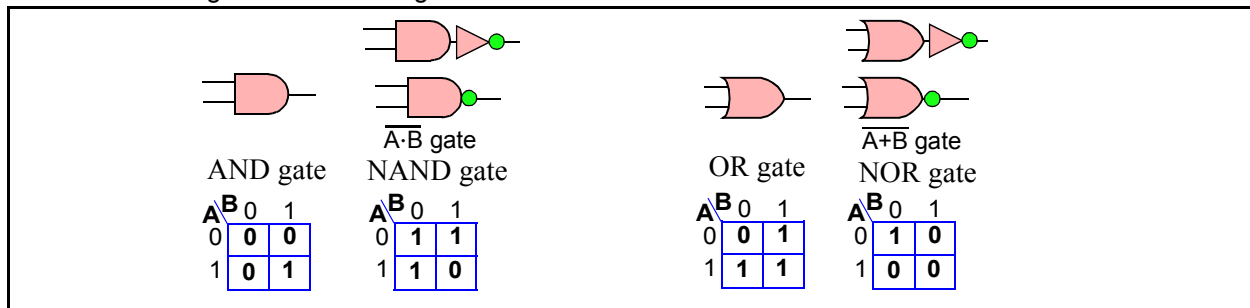
FIGURE 7: The 4 input MUX implemented using three 3-input gates.



The circuit in Fig. 7 using, AND and OR gates, is not directly useful. The Tektron Logic Labs do not have enough AND and ORs, rather they use gates called NANDs and NORs, described in Fig. 8. You will later learn that the gates used in practical digital circuits are almost always NANDs and NORs.

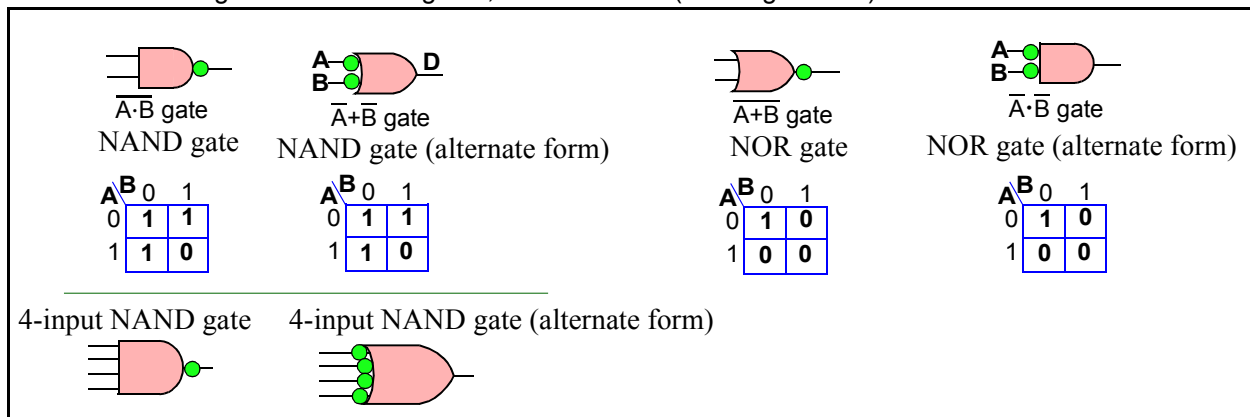
1.3.1 Using NAND and NOR gates

FIGURE 8: NOR gates and NAND gates.



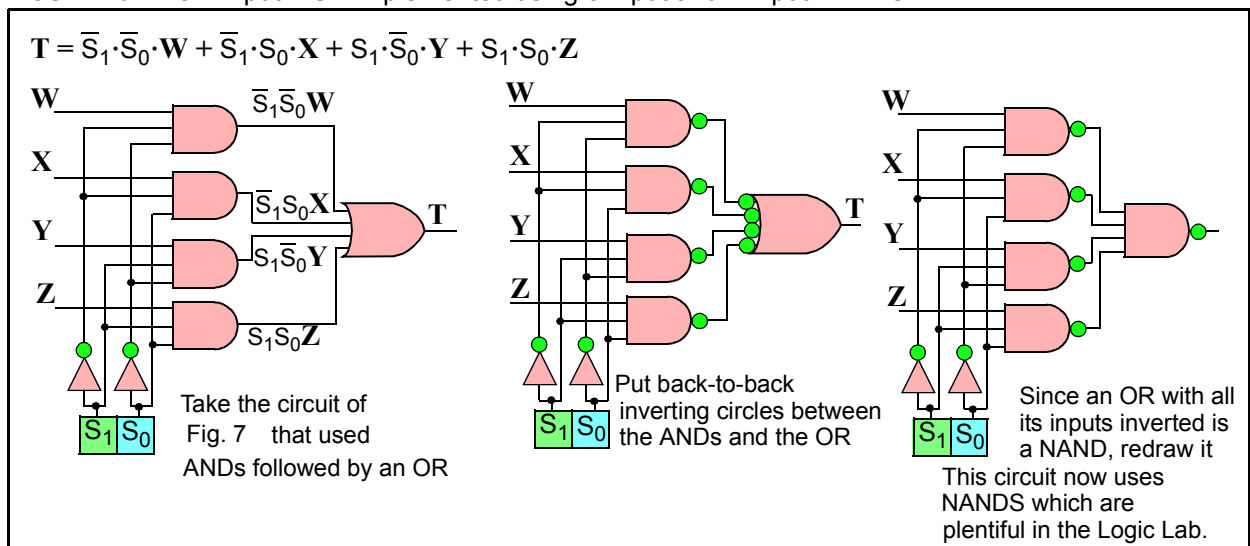
There is an alternate form for a NAND, and another for a NOR, which has the inverting circles on the inputs. Note especially that the alternate form of NAND contains an OR gate and the alternate form of NOR contains an AND gate.

FIGURE 9: NOR gates and NAND gates, alternate form (DeMorgan form).



It is easy to convert a circuit using ANDs followed by OR gates to one using NANDs. Take the circuit and placing inverting circles on the output of each NAND. To avoid changing the circuit operation, also place a circle on each input to the OR. These back-to-back circles cancel each other, so the output T is not affected. However the circles change the AND gates to NANDs, and the OR gate to another NAND.

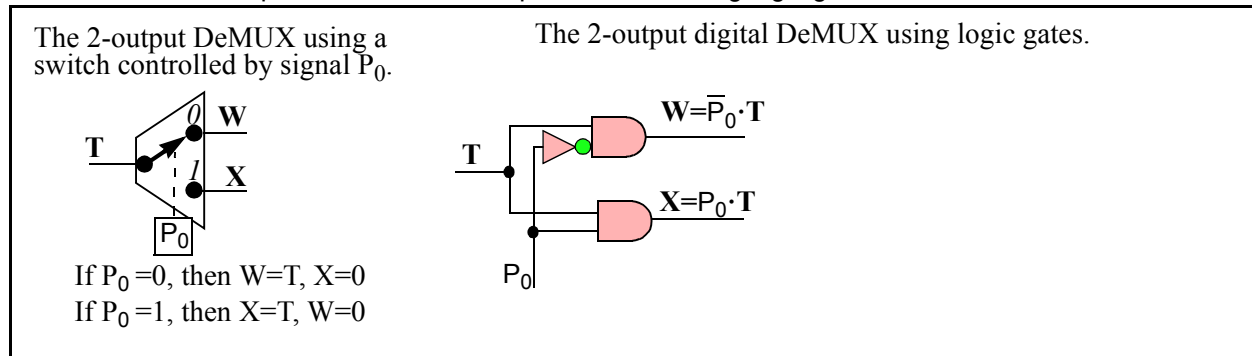
FIGURE 10: The 4-input MUX implemented using 3-input and 4-input NANDs.



1.4 The Demultiplexer (DeMUX)

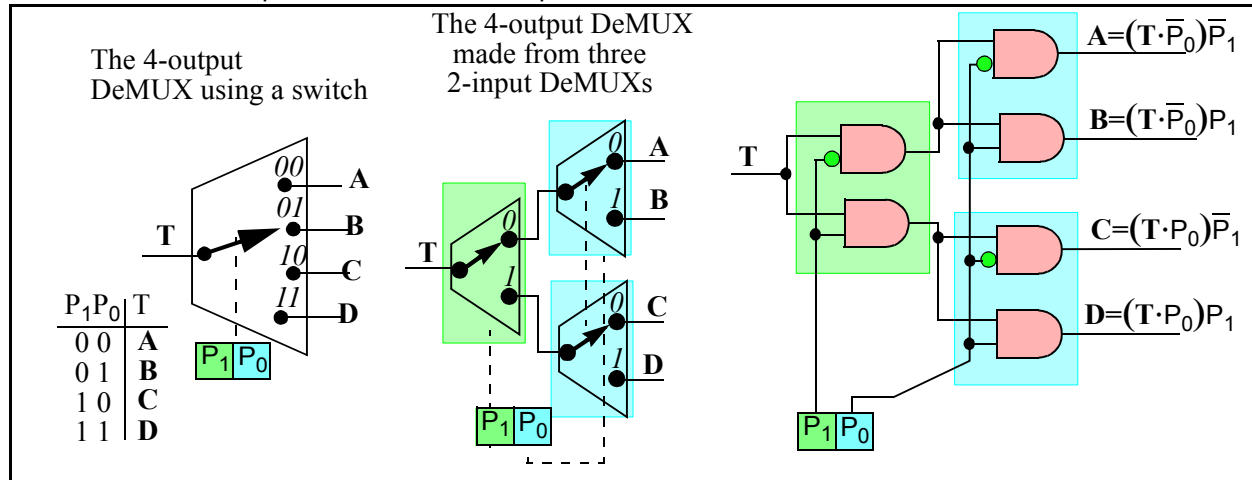
The DeMUX takes one input and switches it between two or more outputs.

FIGURE 11: The 2-output DeMUX and its implementation using logic gates.



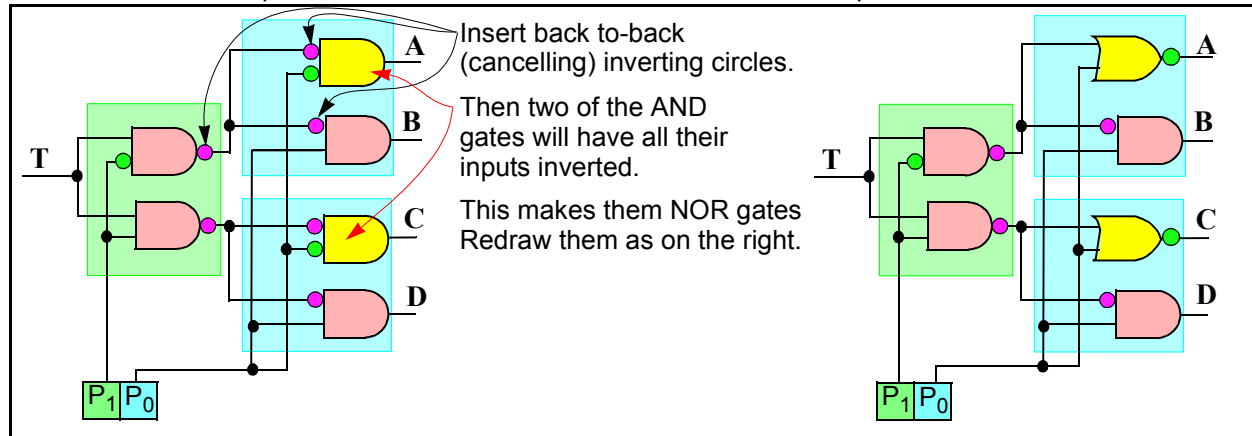
Read the equations in Fig. 11, until you understand how T is switched to either W or X.³

FIGURE 12: The 4-output DeMUX and one implementation.



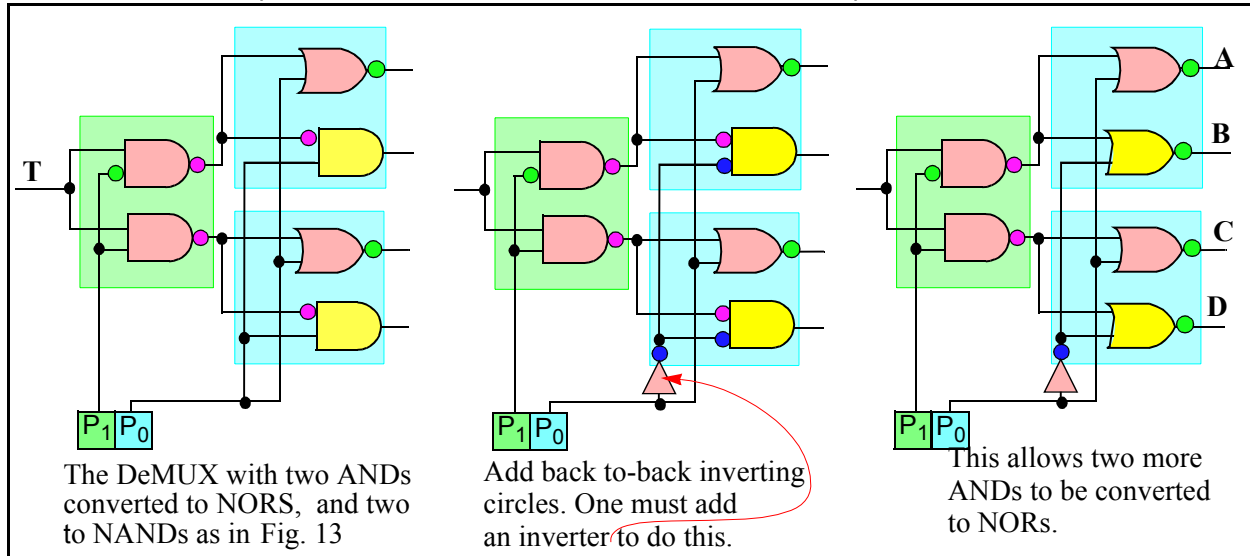
Changing this to use NAND and NOR gates is almost as easy as with the MUX. See Fig. 13.

FIGURE 13: The 4-output DeMUX converted to use NAND and NOR, step1.



3. In Fig. 11, when $P_0=0$, $W=T$, but with switches, as shown on the left, X will not be 0 but will be disconnected. We say it is "high impedance". In the gate circuits, on the right, X will be 0.

FIGURE 14: The 4 output DeMUX converted to use NAND and NOR, step2.



Here is another way to build a 4-output DeMUX.

FIGURE 15: The 4-output DeMUX, another implementation.

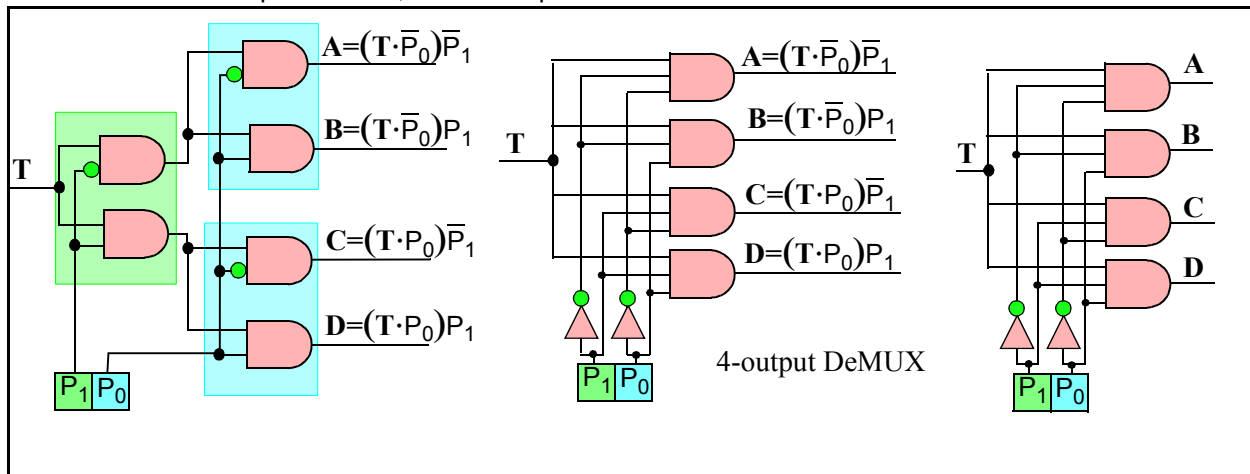
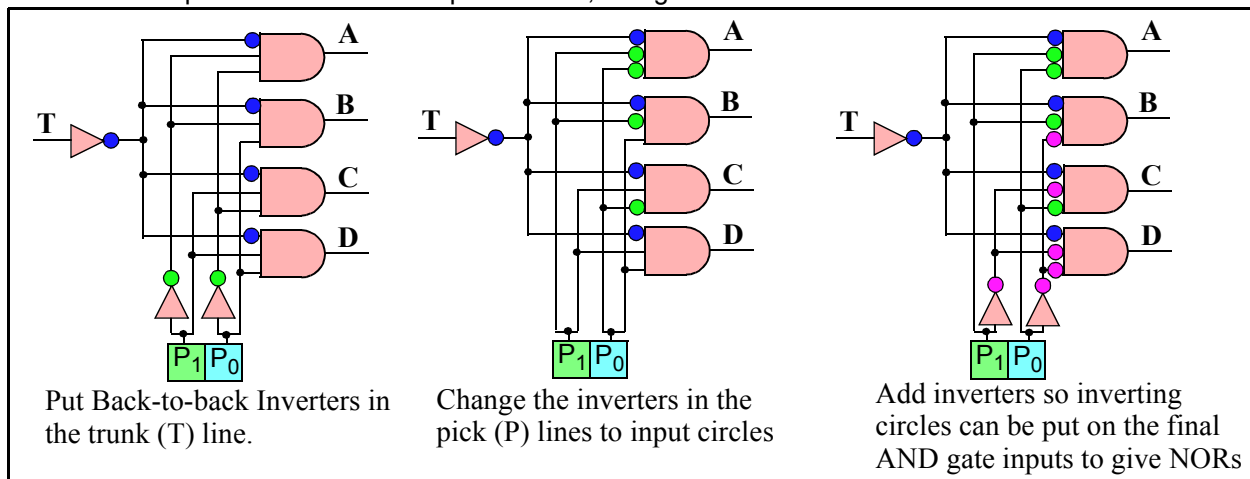


FIGURE 16: Steps to convert the 4-output DeMUX, of Fig. 14 to use NORs.



Prelab

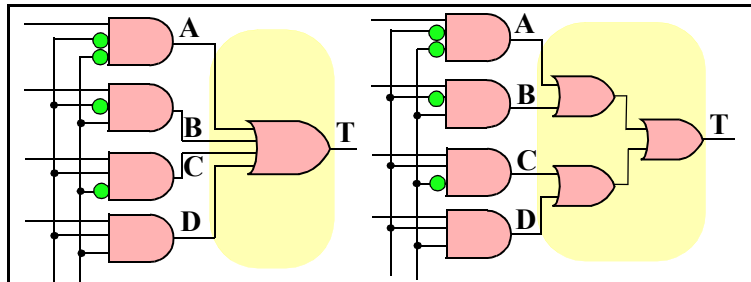
This must be prepared prior to the scheduled lab session.

It will be checked near the start of the lab.

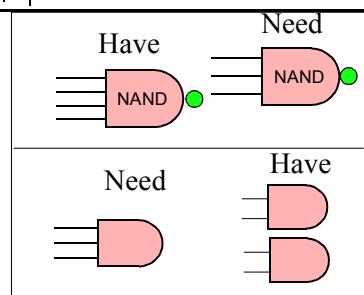
You will be asked to explain any questionable or unclear answers.

The TA will initial the prelab which must be attached as an appendix to the final report.

- 1.1 What law of Boolean Algebra allows us to write $A+B+C+D = (A+B)+(C+D)$ as is done in Fig. 7 and on the right.

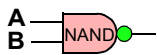


- 1.2 If you need a 3-input NAND gate and have only 4-input NANDs, gates, what can you do?



- 1.3 If you need a 3-input AND gate and only have 2-input ANDs, gates, what can you do? Warning, do not try this for NANDs.

- 1.4 One can prove two formulas are equal if they are equal for all possible input values. Write out a table to prove these two gates are equivalent.



$\overline{A \cdot B}$ gate

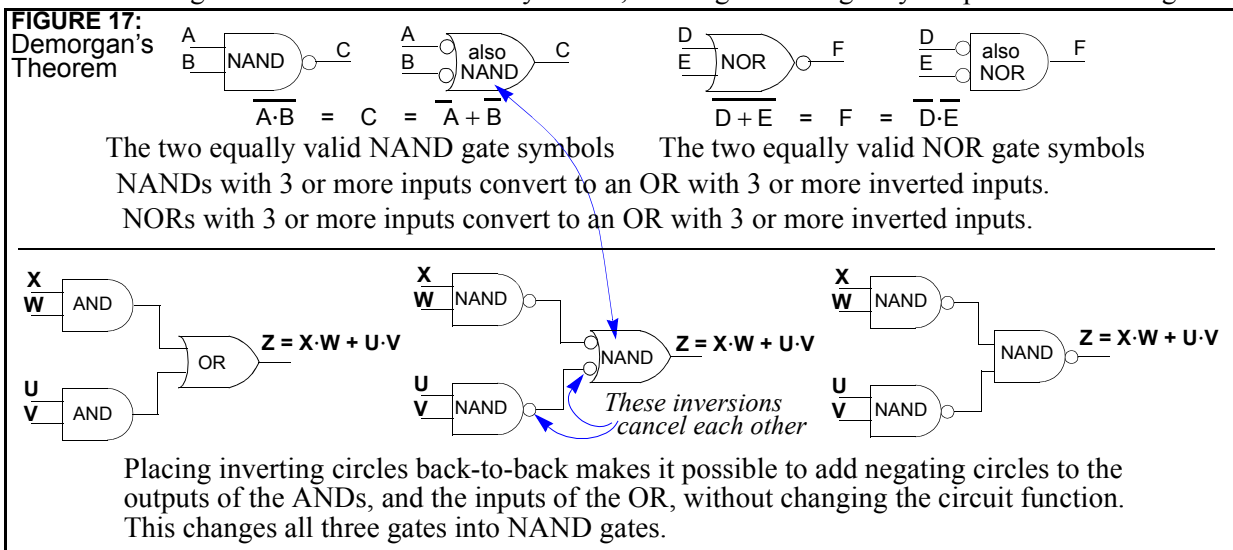
A	B	$A \cdot B$	$\overline{A \cdot B}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0



$\overline{\overline{A} \cdot \overline{B}}$ gate

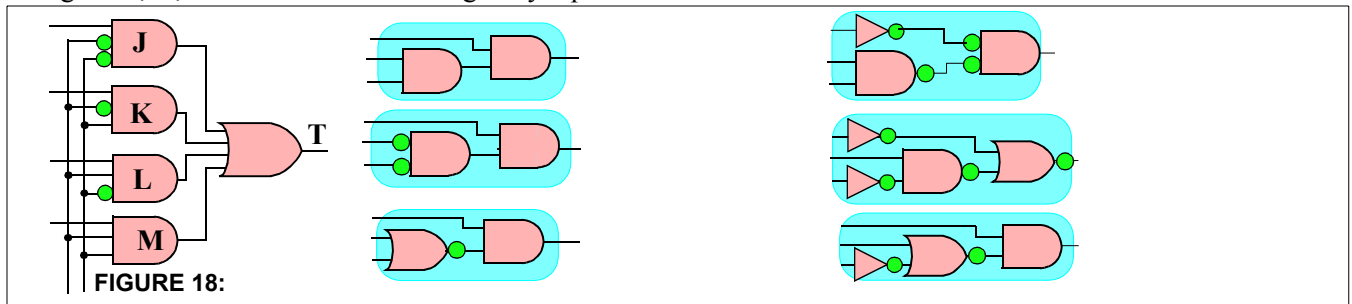
A	B	\overline{A}	\overline{B}	$\overline{\overline{A} \cdot \overline{B}}$
0	0	1	1	0
0	1	1	0	0
1	0	0	1	0
1	1	0	0	1

- 1.5 This equivalence of the two forms of NAND and also the two forms of NOR are called DeMorgan's theorems. Fig. 17 reviews these two equivalent forms. The lower part shows how to build Logic Lab circuits designed with ANDs followed by an OR, scarce gates, using only the plentiful NAND gates.

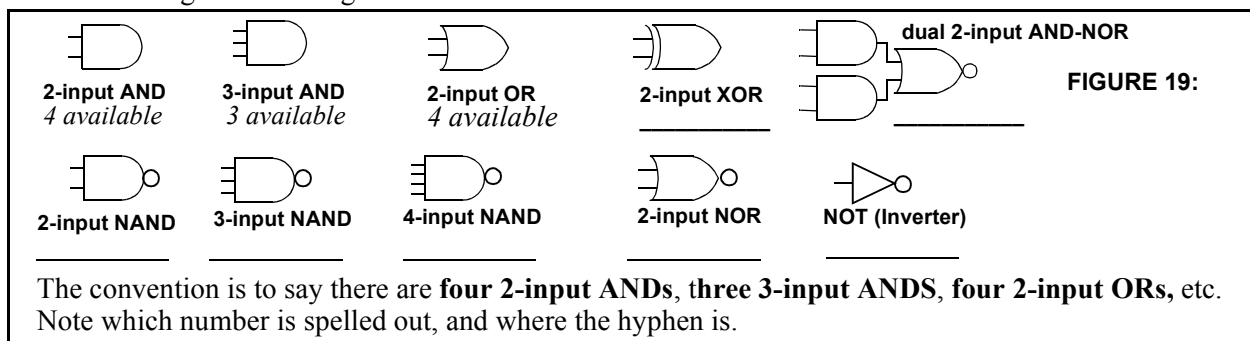


There is no question here. The above is a review for the next questions.

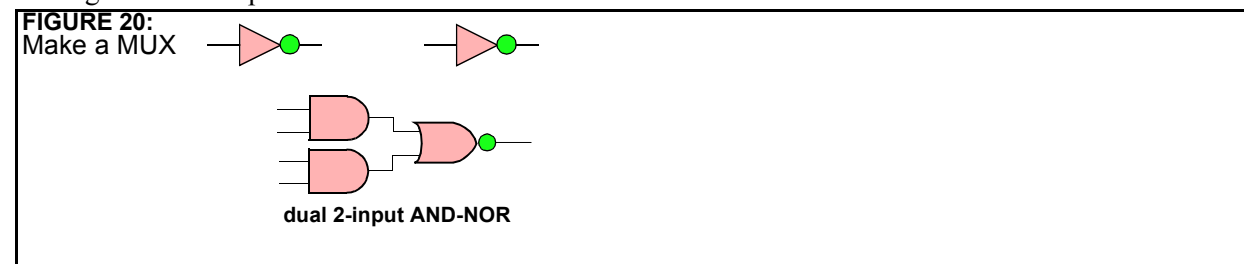
1.6 Transform each of the six groups of 2-input gates on the right of Fig. 18, into the one of the 3-input gates J, K, L or M to which it is logically equivalent.



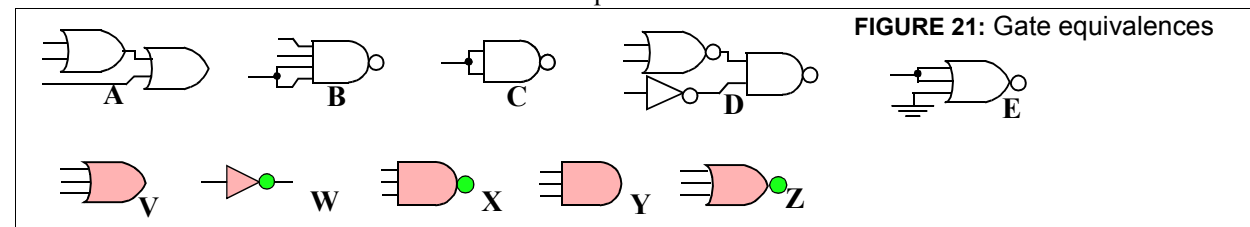
1.7 On the full page diagram of the Tektron Logic Lab on p. 11, count the gates of each type. Then add the missing counts to Fig. 19 below.



1.8 Convert the 2-input AND-NOR gate (of which there are 4 in the Logic Lab) and the two inverters in Fig. 20 to a 2-input MUX.



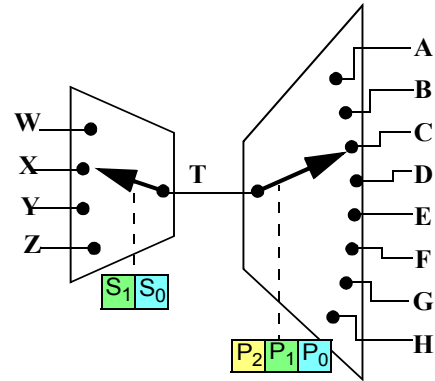
1.9 For each circuit in the top row, indicate which circuit in the bottom row is equivalent..



You are going to construct the 4 input-line, 8 output-line telephone switch of Fig. 3.

- 1.10 Fig. 22 shows the 'phone switch partitioned into a 4-input MUX and an 8-output DeMUX.
Following this example, partition a 4-input 8-output 'phone switch into one 4-input MUX, one 2-output DeMUX, and two 4-output DeMUXs

FIGURE 22:



- 1.11 Partition a 4-output DeMUX into three 2-output DeMUXs.

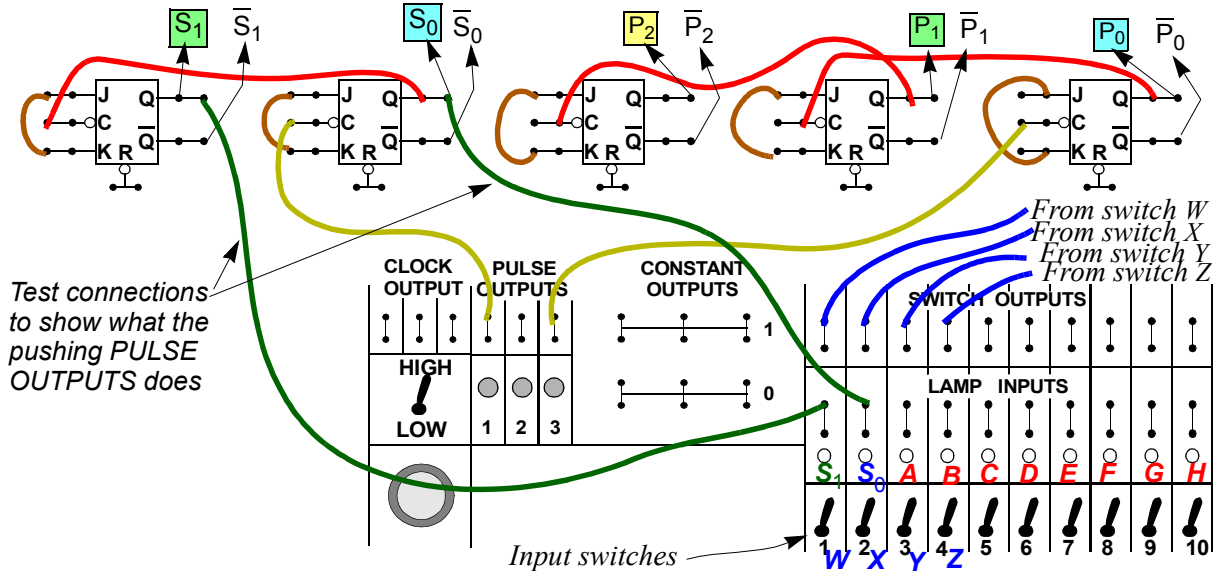
Design the 'Phone Switch

- 1.12 Design a 4-input MUX using gates available on the Logic Lab.
a) Show the circuit partitioned into one or more MUXs
b) Show the schematic diagram.
c) Make a list of the type and number of each type of gate used.
- 1.13 Design an 8 output-line DeMUX circuit for your telephone switch using gates available on the Logic Lab.
a) Show the circuit partitioned into one or more DeMUXs
b) Show the schematic diagram.
c) Make a list of the type and number of each type of gate used.
- 1.14 Add the total number of gates of each type used by the two circuits above. If you do not have enough gates of any type do a redesign.

The Dialer (Already designed for you)

These are circuits to act a telephone dial. There are two, one to select the calling party and one to pick the called party (Fig. 3). They counts, in binary, the number of times you push a button and apply that as S_1S_0 to select the called party and another button generates $P_2P_1P_0$ to pick the called party. You might also check the comment in Sect. 3.4, p. 14

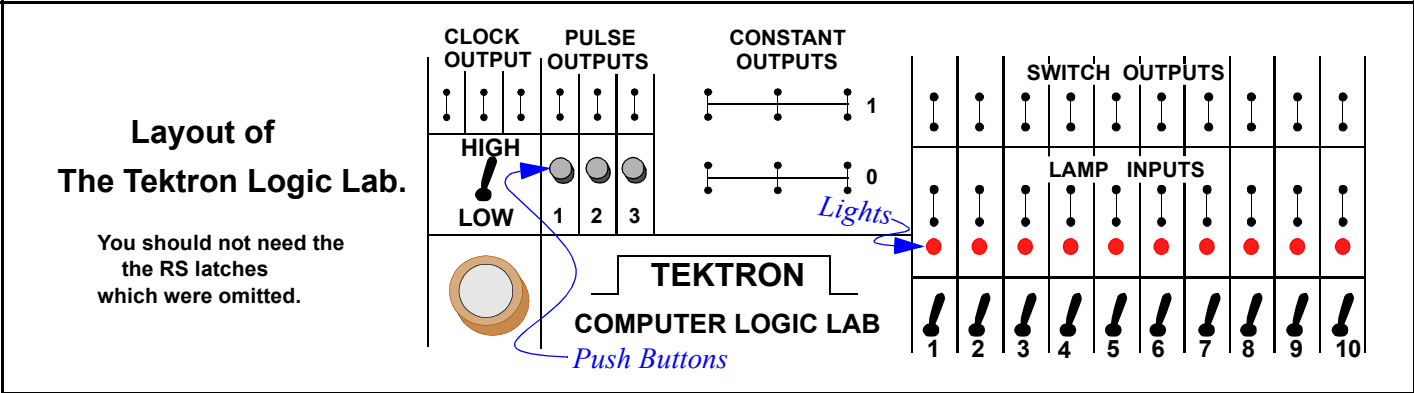
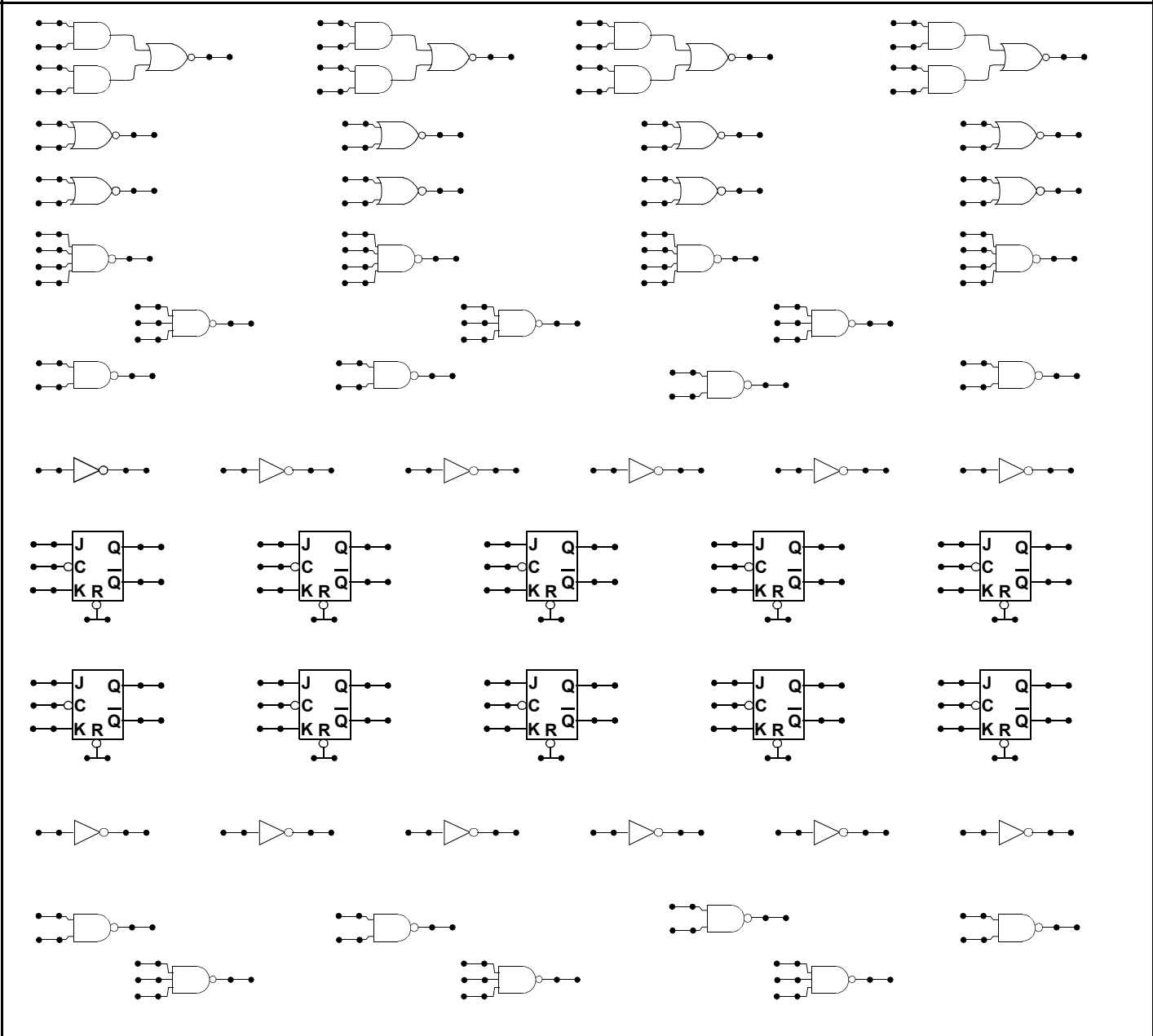
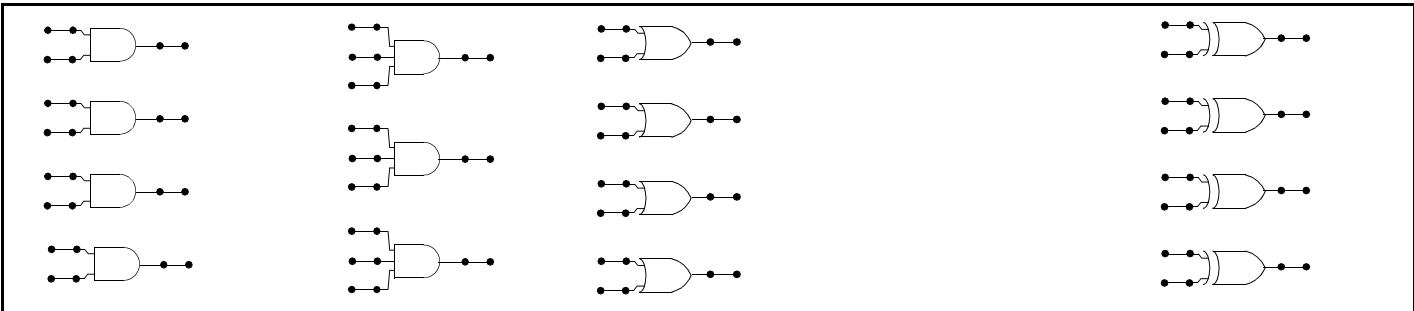
These circuits will be studied later. For this lab you need only make the connections.



Layout the Circuit for the Logic Lab

Wiring from a schematic is the sign of an overconfident amateur. You need a wiring diagram.

- 1.15 Make several copies of the Logic Lab layout.



1.16 Take the circuit at the MUX/DeMUX level, and colour on the layout, an area around the gates used for each MUX-DeMUX. See Fig. 23. Wiring is easier if you can group the gates of one colour together.

1.17 For each MUX or DeMUX make a wiring diagram on a copy of the Logic Lab layout. Fig. 24 is an example.

On diagrams you should draw in only short leads. Give long leads names and just write the name at both end connections, as illustrated in Fig. 24.

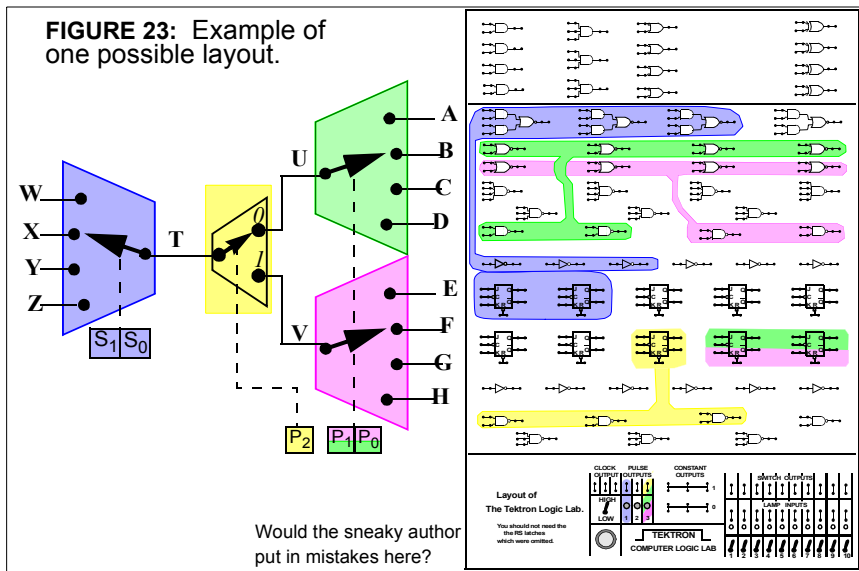
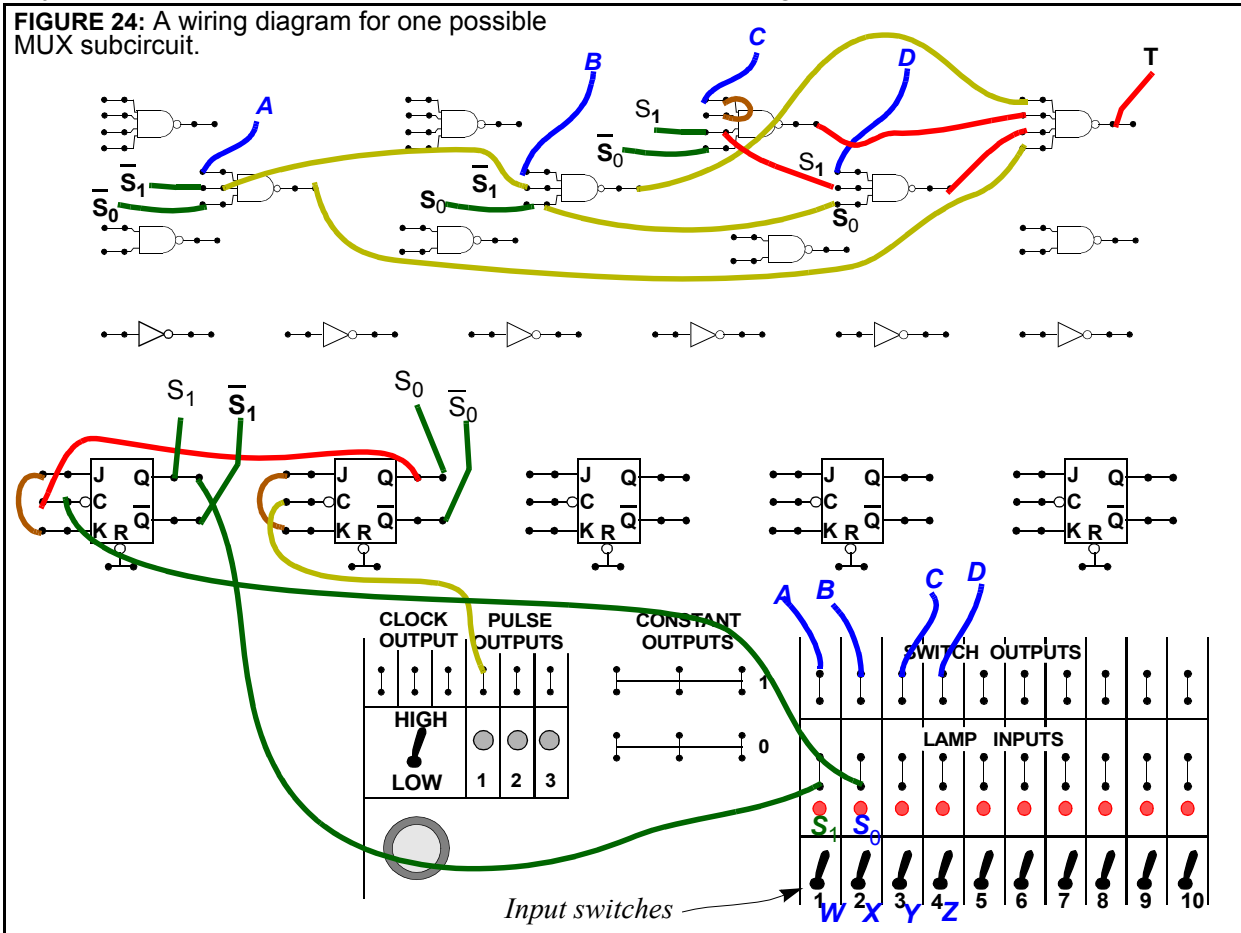


FIGURE 24: A wiring diagram for one possible MUX subcircuit.



1.18 At the start of the lab period, you must have this prelab ready to present to the TA. Further he/she must approve it before you will be given any wires (or marks).

END OF PRELAB

Debugging

Most of the troubles are poorly inserted wires. Reread Step 2.1(p. 13).

A symptom of poor connections is lights that change when you place the flat of your hand on top of the mess of wires.

Take a spare light and connect a long lead on it. You can touch the other end of the lead to any other lead to tell if that other lead is 1 or 0.

Sometimes the big white plastic connection boards are loose in the connector in the wooden frame. Push the heel of your hand on the plastic on the side of the board and push them in.

If you are really having trouble it may be useful to use two trainers. Place them side by side and connect the GROUND connections by a lead. This will give you some extra debugging lights, and some less cluttered space.

Check out and Clean Up

When your circuit works, demonstrate it to the teaching assistant.

After demonstrating clean up your board, sort the leads, and put them back in their bag.

We have a contract with Lord Voldemort to deal with students who do not clean up. He zeros their checkout mark.

You must attach your initialled Prelab as an Appendix to your final report. However it is not acceptable as one of the major sections.

Additions, Thoughts and Ideas for the Commentary Part of the Report

- 3.0 Can you think of an application where consolidations of one-way signals might save using large multiconductor cables?
This hint may fire up your thinking. Either that or steal it.
- 3.1 Could you design and show how to build a system with 8 calling inputs instead of 4.
Then could you find enough gates to put it on the Logic Lab?
- 3.2 Can you add a *broadcast call* switch which delivers the signal to all of A, B, C, and D when the switch is on?
- 3.3 Could you, by using another group's Logic Lab, make a system that, by using 16 pushes, dial any receiver on two labs?
- 3.4 This phone needs a "hang up" control. The S_1S_0 control just counts 00->01->10->11->00->01- . . . to hang up the phone, in this case return to $S_1S_0 = 00$, one can connect the third push button to the R inputs on the two JK flip flops. This will reset the flip flops to 00 when the push button output is "0". Is the push-button output "0" when it is up or pushed down?
If this is not the right way up, how could you easily fix it?

A typical circuit, but not to be taken as an outstanding model for layout

