CPE224 Lab 0 How to build sub-circuits and bus in Proteus

Background

The Basics of Digital Circuits

Analog circuits—circuits that accept and respond to voltages that vary continuously over a given range. Such analog circuits included rectifiers, filters, amplifiers, simple RC timers, oscillators, simple transistor switches, etc. Although each of these analog circuits is fundamentally important in its own right, these circuits lack an important feature—they cannot store and process bits of information needed to make complex logical decisions. To incorporate logical decision-making processes into a circuit, you need to use digital circuits.

In digital circuits there are only two voltage states present at any point within a circuit. These voltage states are either ***high*** or ***low***. The meaning of a voltage being high or low at a particular location within a circuit can signify a number of things. For example, it may represent the ***on*** or ***off*** state of a switch. It may represent one bit of a number, or whether an event has occurred, or whether some action should be taken. The high and low states can be represented as ***true*** and ***false*** statements, which are used in ***Boolean*** logic. In most cases, high = true and low = false. However, this does not have to be the case—you could make high = false and low = true. The decision to use one convention over the other is a matter left ultimately to the designer. In digital system, to avoid people getting confused over which convention is in use, the term ***positive true logic*** is used when high = true, while the term ***negative true logic*** is used when high = false.

In Boolean logic, the symbols **1** and **0** are used to represent ***true*** and ***false***, respectively. Now, unfortunately, 1 and 0 are also used in electronics to represent high and low voltage states, where high = 1 and low = 0. As you can see, things can get a bit confusing, especially if you are not sure which type of logic convention is being used, positive true or negative true logic.

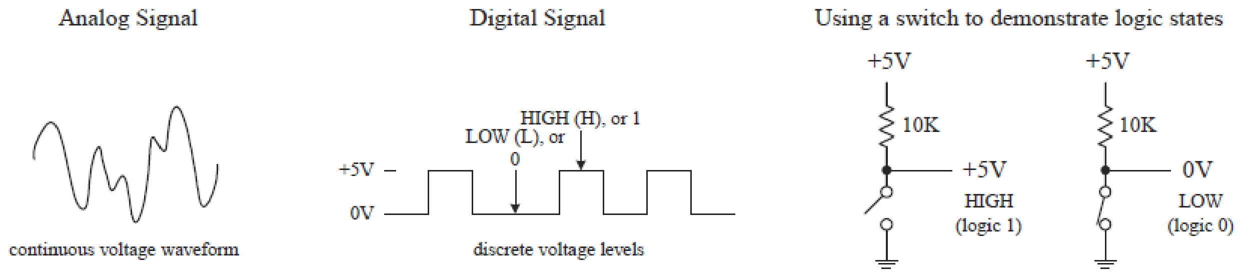
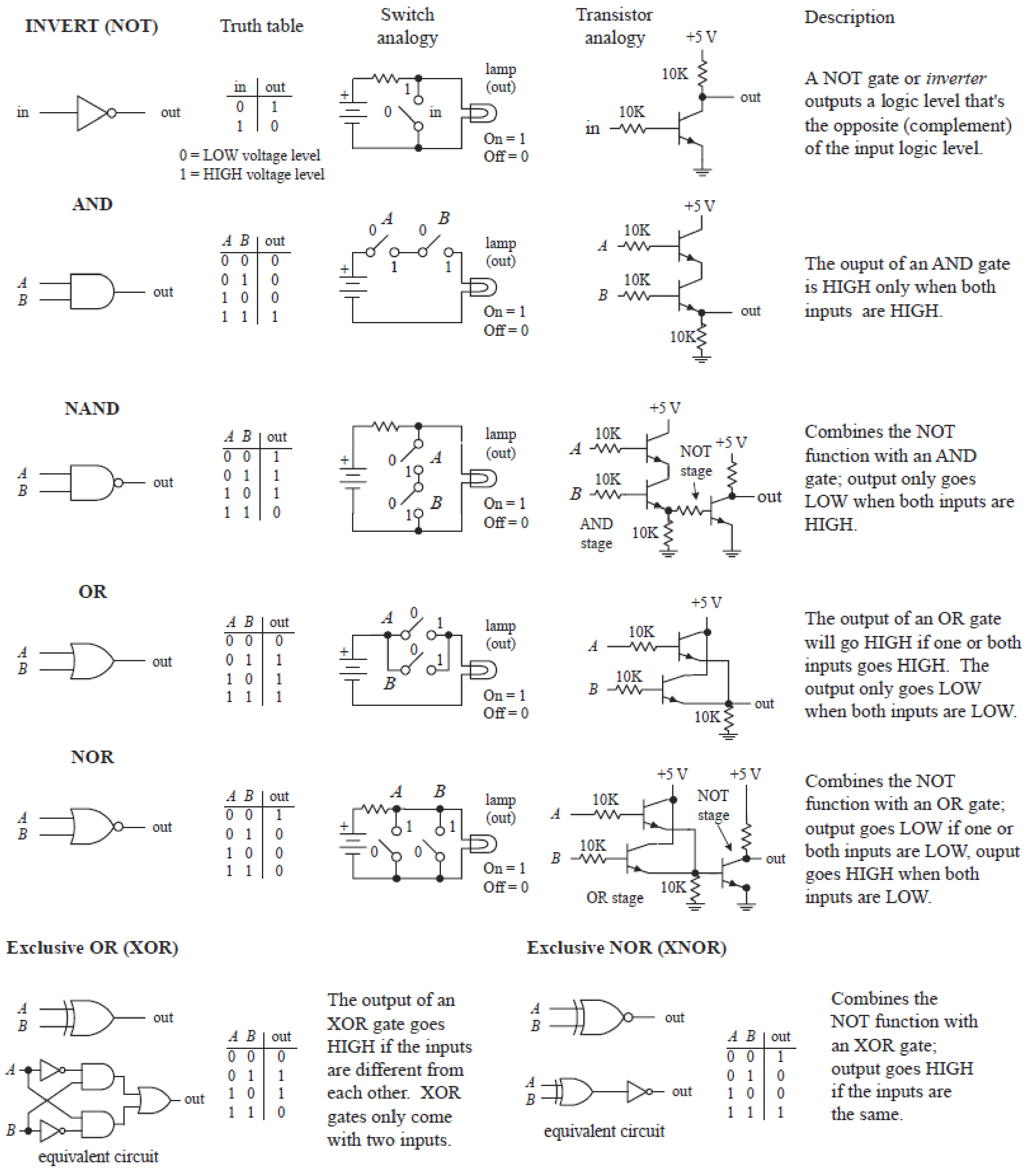


Figure 1 Analog signal and digital signal.

The exact voltages assigned to a high or low voltage states depend on the specific logic IC that is used (as it turns out, digital components are entirely IC based). As a general rule of thumb, +5 V is considered high, while 0 V (ground) is considered low. However this does not have to be the case. For example, some logic ICs may interrupt a voltage from +2.4 to +5 V as high and a voltage from +0.8 to 0 V as low. Other ICs may use an entirely different range. Again, these will be discussed in details later.

Logic Gates

Logic gates are the building blocks of digital electronics. The fundamental logic gates include the INVERT (NOT), AND, NAND, OR, NOR, exclusive OR (XOR), and exclusive NOR (XNOR) gates. Each of these gates performs a different logical operation. Figure 2 provides a description of what each logic gate does and gives a switch and transistor analogy for each gate.



INVERT (NOT)

in	out
0	1
1	0

0 = LOW voltage level
1 = HIGH voltage level

A NOT gate or *inverter* outputs a logic level that's the opposite (complement) of the input logic level.

AND

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

The ouput of an AND gate is HIGH only when both inputs are HIGH.

NAND

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

Combines the NOT function with an AND gate; output only goes LOW when both inputs are HIGH.

OR

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

The output of an OR gate will go HIGH if one or both inputs goes HIGH. The output only goes LOW when both inputs are LOW.

NOR

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

Combines the NOT function with an OR gate; output goes LOW if one or both inputs are LOW, ouput goes HIGH when both inputs are LOW.

Exclusive OR (XOR)

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

The output of an XOR gate goes HIGH if the inputs are different from each other. XOR gates only come with two inputs.

Exclusive NOR (XNOR)

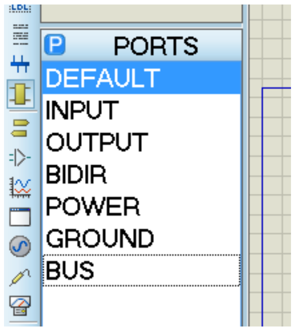
A	B	out
0	0	1
0	1	0
1	0	0
1	1	1

Combines the NOT function with an XOR gate; output goes HIGH if the inputs are the same.

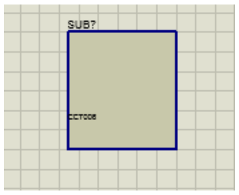
Figure 2 Basic logic gates.

Procedure
Follow the instruction

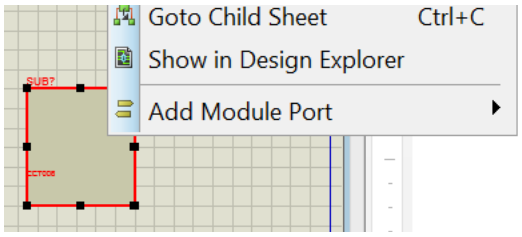
a. Add these following Parts: LOGICPROBE(BIG), LOGICSTATE, NOT
b. Create blank sheet (called root sheet)
c. Select PORTS Mode



d. Create square box in middle of sheet

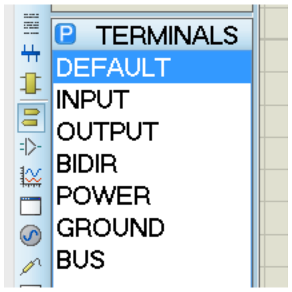


e. Right click on the box and choose Goto Child Sheet

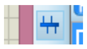


Now you are in Child sheet

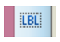
f. Use the TERMINAL Mode



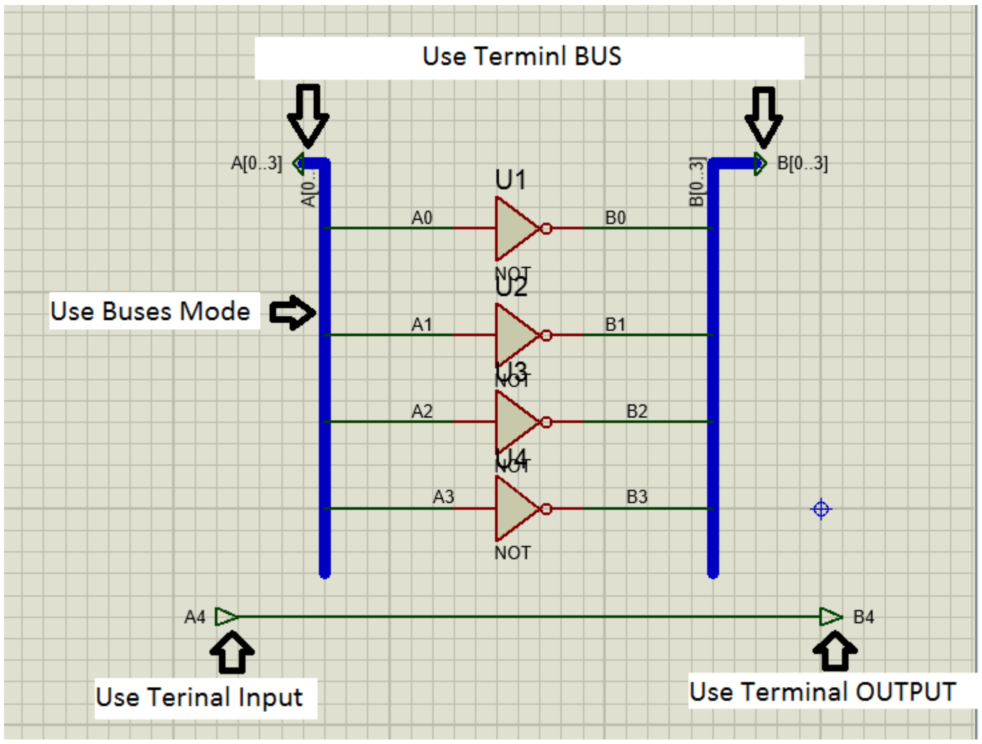
BUSES Mode



WRITE LABEL Mode



To create this circuit



Make TERMINAL Mode String Properties for

BUS output [A0..3] B[0..3]

INPUT A4

OUTPUT B4

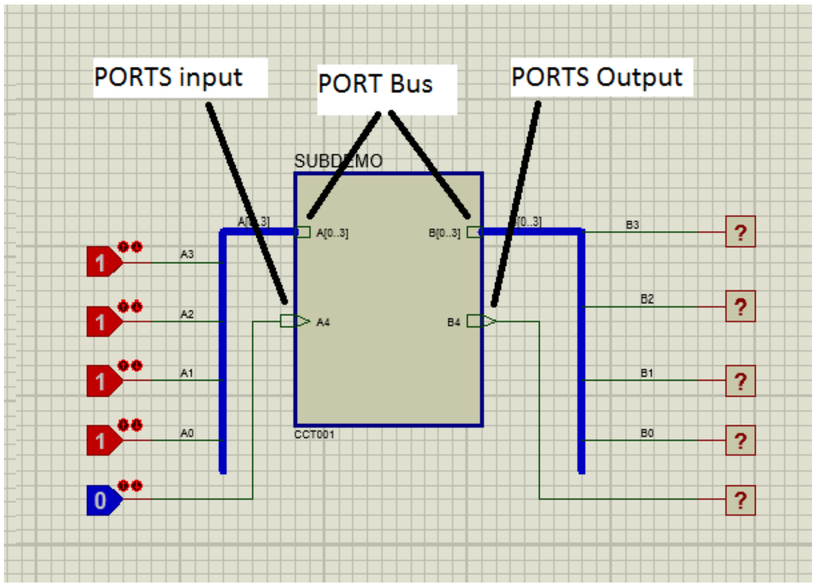
Use WRITE LABEL Mode for

BUS [A0..3] B[0..3]

All wires connected to all BUSes

g. Click right on empty on Child sheet and chose Exit to Parent sheet

h. Use PORTS Mode create this circuit



Make PORTS Mode String Properties for

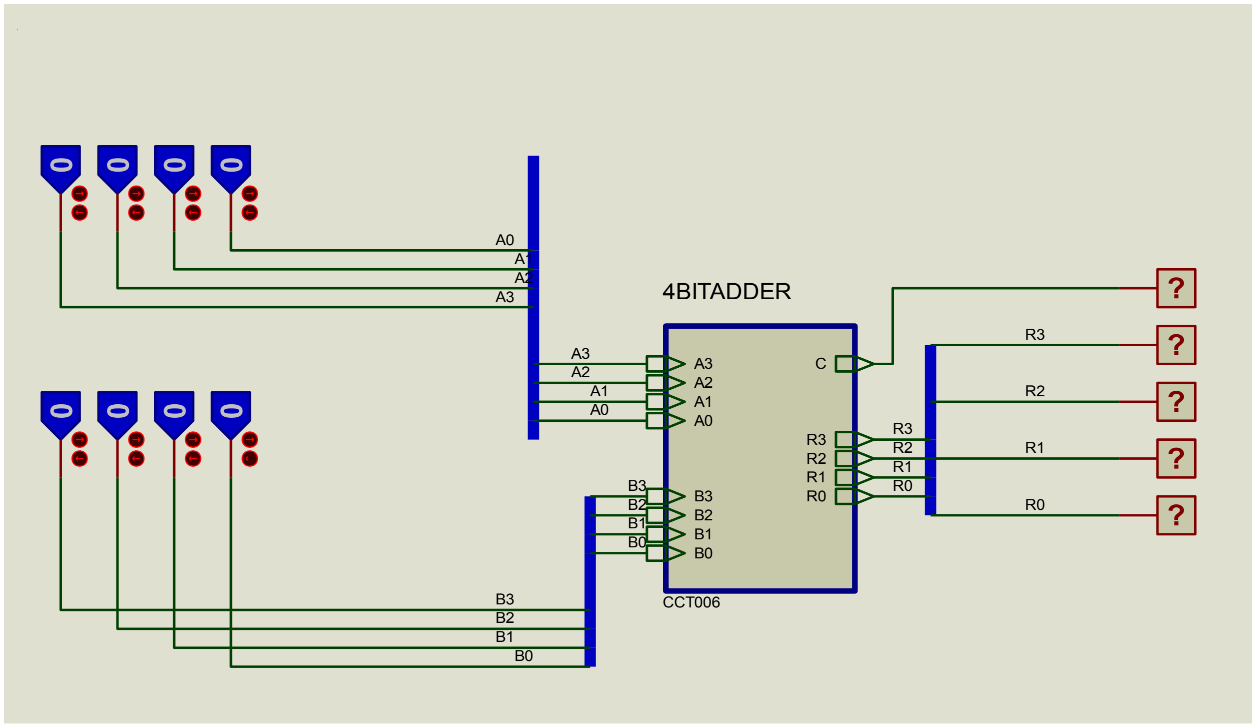
BUS output [A0..3] B[0..3]
INPUT A4
OUTPUT B4

Use WRITE LABEL Mode for
BUS [A0..3] B[0..3]
All wires connected to all BUSes

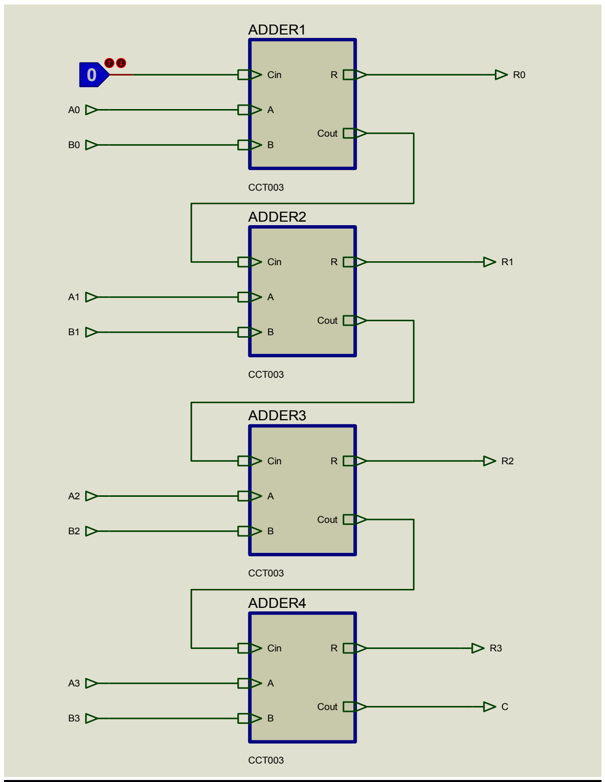
i. TEST your circuit

Challenge Lab Build the 4bit Full adder from four 1bit full adder sub-circuit

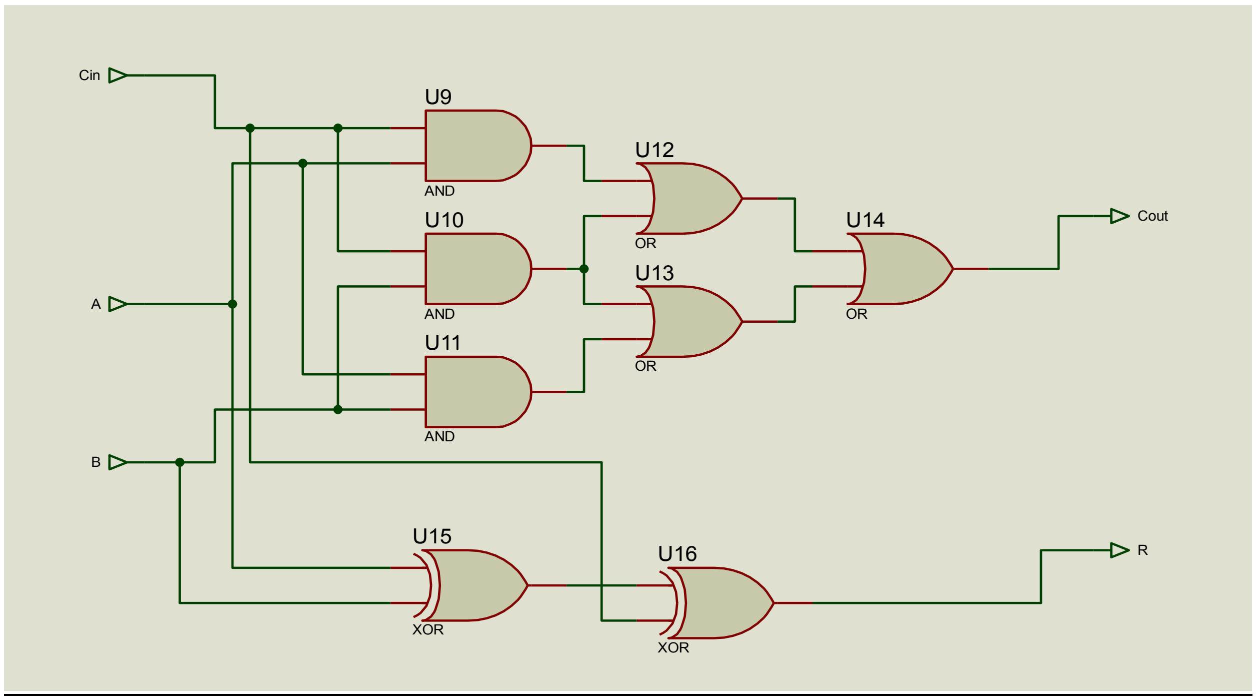
Root Sheet 1



Child Sheet 1



Child Sheet 2



PARTS

OR, AND, XOR, LOGICPROBR (BIG), LOGICSTATE.