Griffin’s OH 2/26/22

COMS 1004 Introduction to Computer Science and Programming in Java

Quick Announcements

1. Problem Set 3 is due Monday 2/28/22. If you have not started on it please begin to do so
2. Programming Problem Set 3 is available and is due on Monday March 7th, it is harder than previous sets so please begin working on it early
3. Quiz 3 is on Tuesday March 8th please begin to study for it if you deem it necessary
4. The Collaborative Space in Math 207 is open from 1-5pm this Sunday (2/27/22)

Topics for the Week

1. Circuit Design and Utility
2. Even Parity
3. Borrow in and Borrow out bits
4. Assembly Instructions Simplified
5. Top-Down Class Design
6. Questions on the PSET

Circuit Design and Utility

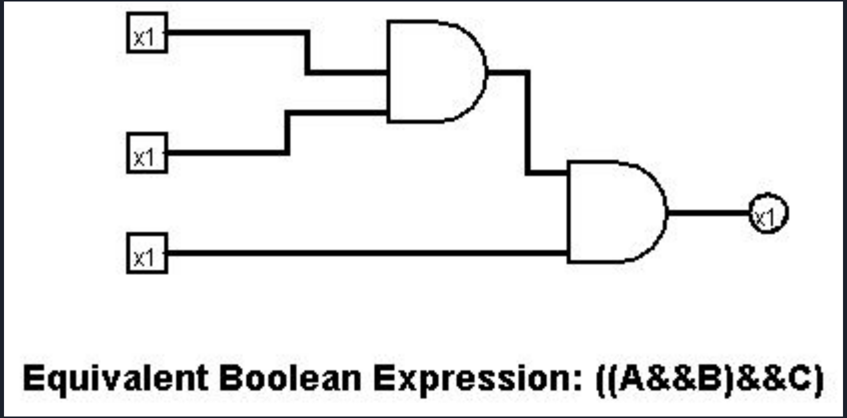
Circuits are an integral part of Computer Engineering and theoretical Computer Science and as such it is important for you to understand how circuits work and how to construct them.

Circuits have 4 key components:

1. Input Nodes
2. Wires
3. Logic Gates (see last week's slides)
4. Output Node

Circuit Design and Utility

In the image below, the three square nodes are the inputs for the circuit, with the single circular node being the output, two AND gates are used with wires to connect the nodes to the gates and the gates to the output. Now let's do a couple examples!



Circuit Design #1 Boolean to Circuit

Given the following boolean expression construct a circuit representation: (A&&B) || (C&&D)

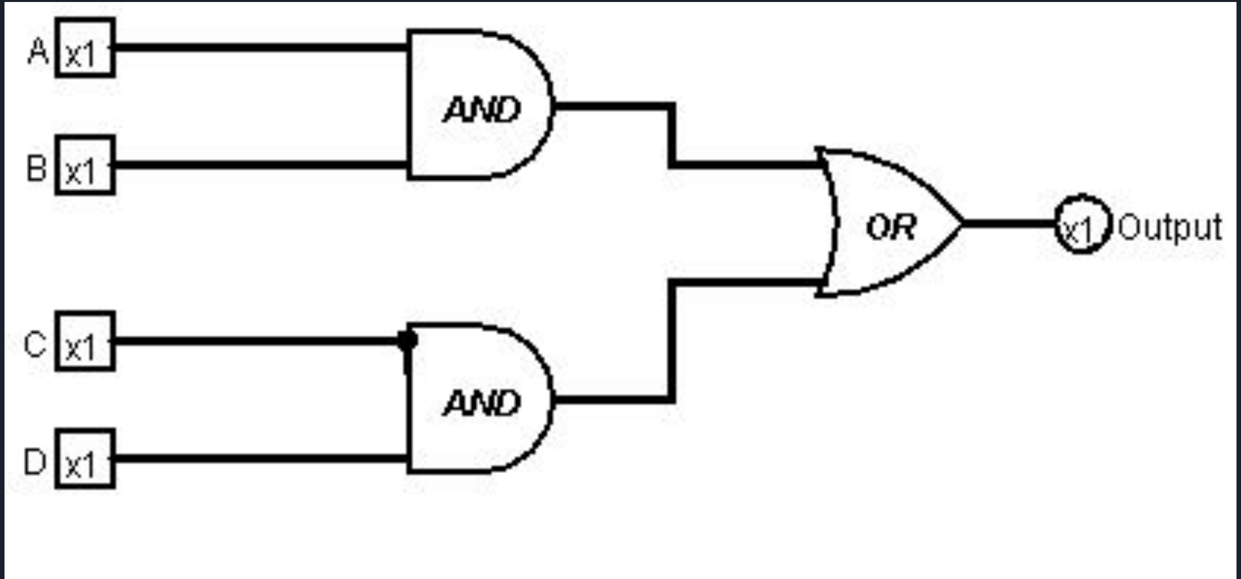
First Step: Determine the number of Inputs you have (In this case we have 4: A, B, C, and D)

Second Step: Determine what logic gates you'll need (In this case we will need 2 AND Gates and an OR gate)

Third Step: Split into smaller circuits, construct and then combine results together

Using this logic lets see what the final circuit will look like!

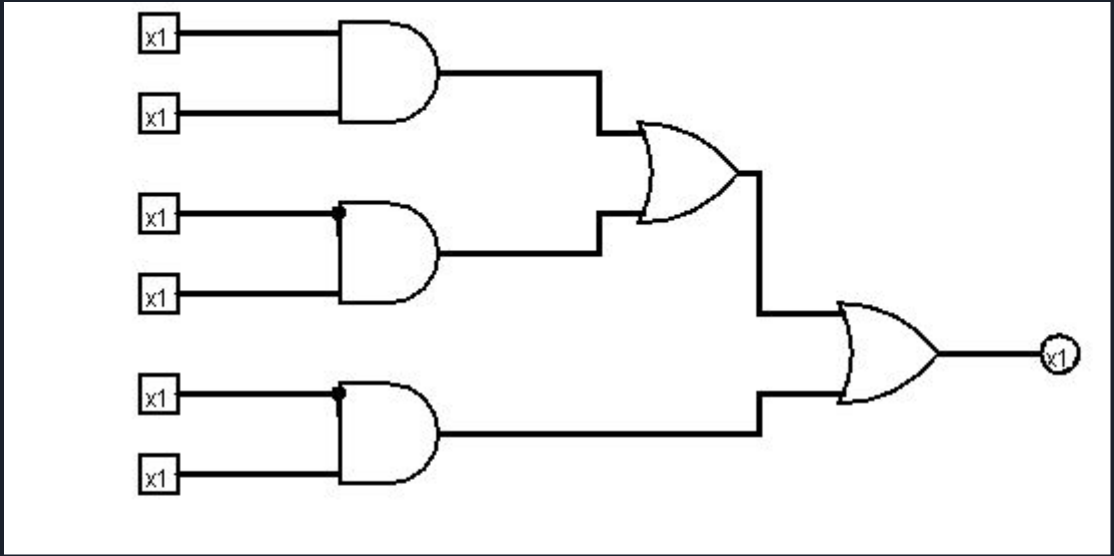
Circuit Design #1 Boolean to Circuit Solution



Equivalent Boolean: (A&&B) || (C&&D)

Circuit Design #2 Circuit to Boolean

Given the following Circuit, convert to the proper boolean expression:

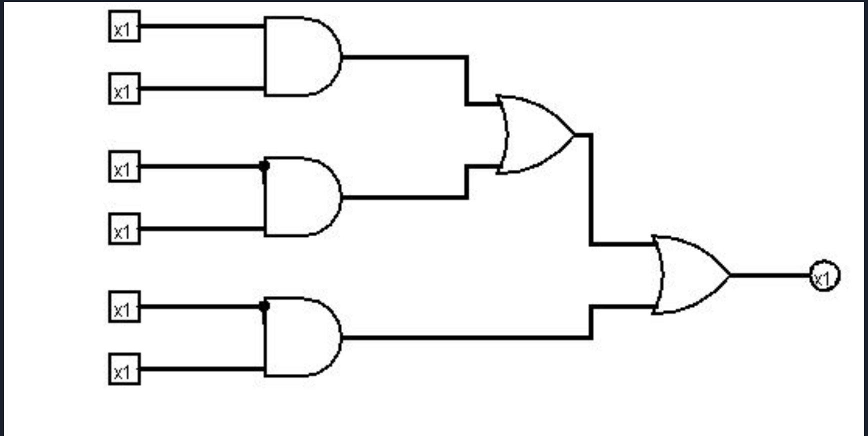


Circuit Design #2 Circuit to Boolean Solution

Step 1.) Recognize there are 3 pairs of inputs: A,B,C,D,E, and F

Step 2.) Connect each pair with their logic gate: A&&B C&&D E&&F

Step 3.) Connect each connections with their logic gate: (((A&&B) || (C&&D)) || (E&&F))



Equivalent Boolean Expression: (((A&&B) || (C&&D)) || (E&&F))

Even Parity Circuit

For the Even Parity Circuit, if there are an even number of bits set to 1 then the out bit is set to 0

If there is an odd number of bits set to 1 then the out bit is set to 1.

If we wanted to convert this to a boolean expression it would look like this:

Output = (A && !B && !C) || (!A && B && !C) || (!A && !B && C) || (A && B && C)

If we wanted to, we could build a circuit and even write a Java program to simulate this behavior, but of course since this is too closely related to Odd Parity, those exercise are left to the reader

Borrow in and Borrow Out Bits

For one of your problems this week you have to implement an one-bit subtractor, please Friday's review session for more help starting out this process A concept that trips up students however, including myself when I saw this in my Departmental Project for AOE last semester was the concept of the Borrow In bit and the Borrow Out bit.

It is easier to think of this process with Base 10 subtraction since it works the same way. Just ask yourself 2 questions:

1. Did the previous column need to do a borrow? (If yes that is a borrow in)
2. Did this column's subtraction need to do a borrow? (If yes that is a borrow out)

Here an example of this in decimal: 110 - 98

For the first column, there is no previous column, so no borrow in. But there is a borrow out - we needed to borrow because 0 < 8. The result is 2.

For the second column, there is a borrow in from the first column, and a borrow out since 0 < 9. The result is 1.

For the third column there is a borrow in, but no borrow out (0 >= 0).

Assembly Instructions Simplified

The following useful assembly instructions that you are able to use for the PSet (pgs 253 and 254 of the Invitation Textbook):

1. LOAD X
2. STORE X
3. MOVE X,Y
4. ADD X,Y,Z
5. ADD X,Y
6. COMPARE X,Y
7. JUMP X
8. JUMPGT/LT/EQ/GE/LE/NEQ
9. HALT

We will do a couple examples of these Reference (v,w,x,y,z = 200,201.202,203, and 204 respectively)

Assembly Instructions Example #1

Write out the Assembly Instructions for the following code segment:

if(x < y) then set v to z else set v to y

Memory Location	OP Code	Address Field	Comment
50	COMPARE	202, 203	Compares the values of x and y
51	JUMPLT	54	Jump to Location 54 if x < y
52	MOVE	203, 200	Copies contents of y onto v
53	JUMP	55	Jump to location 55
54	MOVE	204, ,200	Copies the contents of z onto v

Assembly Instruction Example #2

Write the Assembly code instructions for the following code segment:

while(y >= z) set y to the value of w+y set z to the value of z+y End of Loop

Memory Location	OP Code	Address Field	Comment
50	COMPARE	203, 204	Compares the values of y and z
51	JUMPLT	59	Jump to Location 59 if y < z
52	LOAD	203	Register R now has the contents of y
53	ADD	201	Register R now has the contents of y+w
54	STORE	203	The value has been stored to y

Assembly Instructions Example #2

Memory Location	OP Code	Address Field	Comment
55	LOAD	204	Register R now has the contents of z
56	ADD	203	Register R now has the contents of z+y
57	STORE	204	The value has been stored to z
58	JUMP	50	Jump to location 50 to repeat loop
59	….	…	….

Top-Down Class Design

Last week someone asked me how I went about designing all my projects from the very small to the very large. The process I described was Top-Down Class Design. In Top-Down you start with a very broad problem you want to solve and you continually break it down into subproblems until you get to a point where you can solve the subproblems to eventually solve the whole problem. Here is a good example:

Imagine you wanted to simulate a Movie Theater well to simulate a theater, you need the main theater class, but then you also have to handle each screen in the theater, and each screen has a movie associated with it so you have to handle that too, and each movie has different showtimes etc….

Link to the Video Version

Thanks for reading! If you would like to watch this in video form or see past videos click the link!

Griffin's OH Video Link

View other resources that I have posted in the files section on courseworks

Griffin's OH Materials

Have a wonderful week and I hope to see you soon! As always if you have any questions please ask on EDStem or email me at gcn2106@columbia.edu