Course Number: EC-ENGR 5590VL-0001 Special Topics in ECE

Tutorial

Lab report

**TITLE**

Simulation using HSPICE for Full Adder gate using Sub-circuit

Date of Performing Experiment: 22nd October 2019

Due Date: 29th October 2019

**Student ID: 14344331**

**Name: Prerana Samant**

# Objective:

To run HSPICE on Linux platform, to create netlist and simulation of Full Adder gate using the sub-circuit approach.

# Theory:

Full Adder takes into consideration the previous carry. So there are 3 inputs and 2 outputs.

For creating a Full Adder in CMOS logic, 2 XOR gate generates the SUM output and the 2 AND gates plus 1 OR gate is used to generate the CARRY output. Hence, we can first create sub-circuit of XOR gate, AND gate and the OR gate and use it to design Full Adder.

Alternatively, we can create sub-circuit of half adder and use two half adders to implement the full adder. I have used the first method to implement the Full adder.

The Full adder has only 3 inputs. One of the input is carry from previous state. It adds the current inputs with the previous carry.

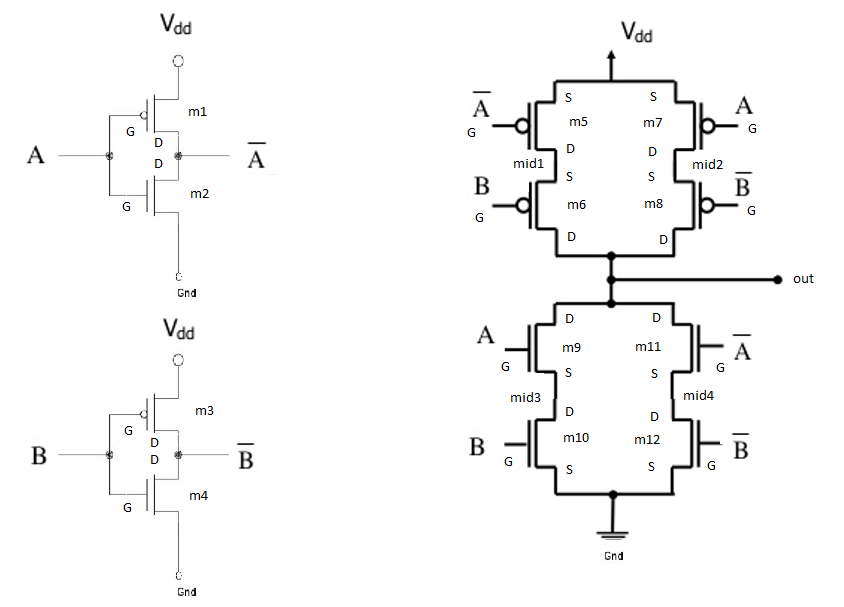
Truth Table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | Cin | Sum | Carry |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

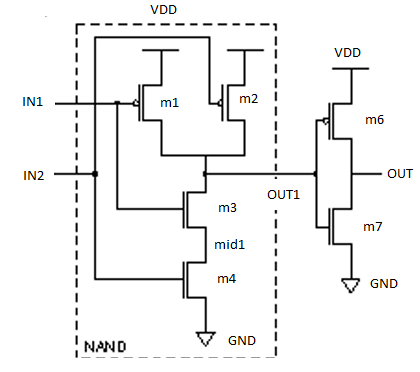
Equations:

The Full Adder sub-circuits can be implemented as below:

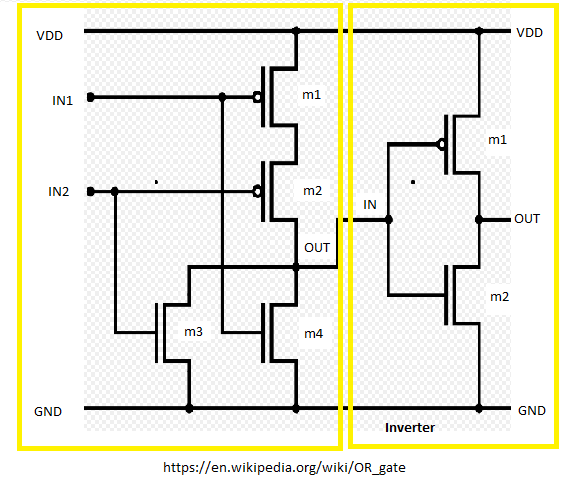
# CMOS implementation: XOR gate



# CMOS implementation: AND gate



# CMOS implementation: OR gate



# Procedure to Simulate in HSPICE:

1. Start MobaXterm like previous tutorials. Open New Cluster and enter password.
2. Enter the directory hspice.
3. Type below commands:

Cd hspice

gedit &

1. We require the input netlist file in HSPICE and has to be created beforehand and saved at a location where HSPICE is installed. Hence gedit is used to write a netlist input.
2. Type the netlist for sub-circuits, AND gate, XOR gate and the OR gate. For AND gate, use netlist of NAND gate and add an inverter at its output. OR gate can be implemented using NOR gate with an inverter. Use the CMOS implementations in the figures above to implement the netlist.
3. Define the voltage sources and nets for the main Full Adder block. Save it in the HSPICE folder by name “fa.sp”. Extension “sp” is for input netlist.
4. The MobaXterm will show some error but still the file gets saved. Close the TUX and Open a new TUX window. Enter password and directly type:

cd hspice

ls

Here you will find the file created.

1. The model files “ami06\_models.txt” are already in the folder.
2. To create an output listing, type below command:

hspice fa.sp > fa.lis

(inputfilename.sp > outputfilename.lis)

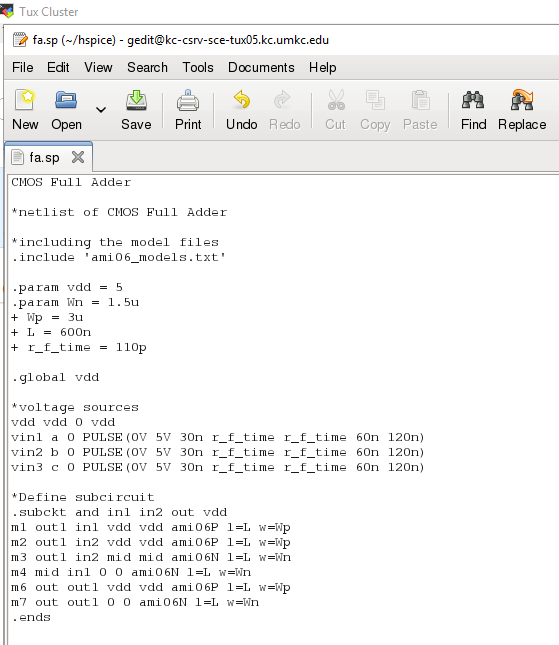
1. After this command you must get “hspice job concluded”. This means there are no errors. To check what errors are in the .sp file, open gedit🡪 open the .lis file
2. Now, simulate the netlist using CosmosScope. Type below command:

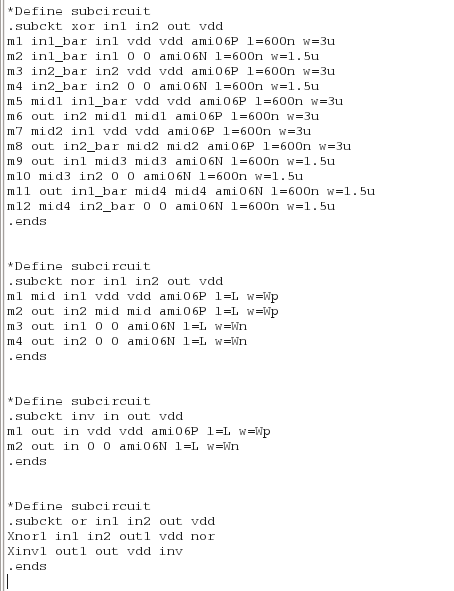
cscope &

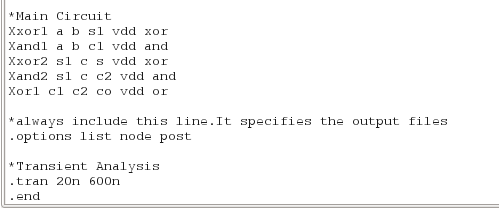
1. The simulator window will open. Click on File🡪 Open🡪Plotfiles.
2. Open the “fa.tr” file. This file is the transient analysis graph data.
3. Select the inputs and outputs of main module V (a), V (b) and V (c) and then click on Plot. Verify the graphs with truth table.
4. For simulating full adder and easily verifying the output, keep the time period of all the inputs same. So that the last state of truth table can be easily verified.

# Tables/Graphs for Full Adder Schematic

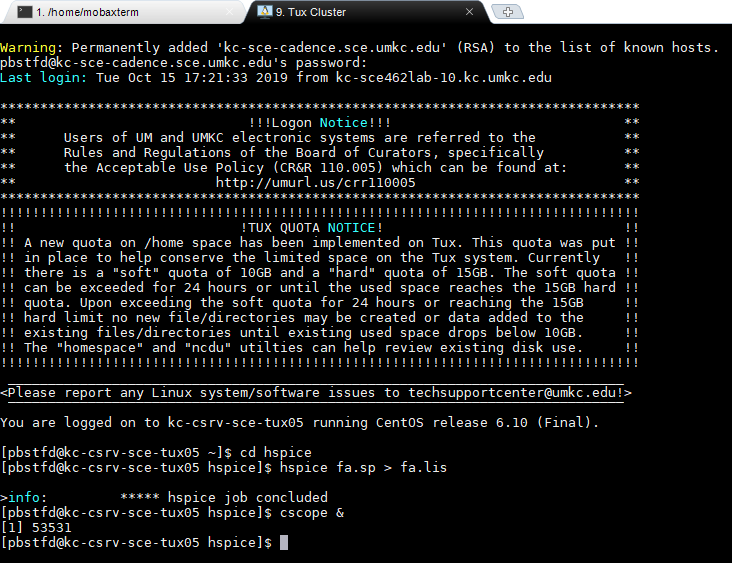
1. Gedit Code



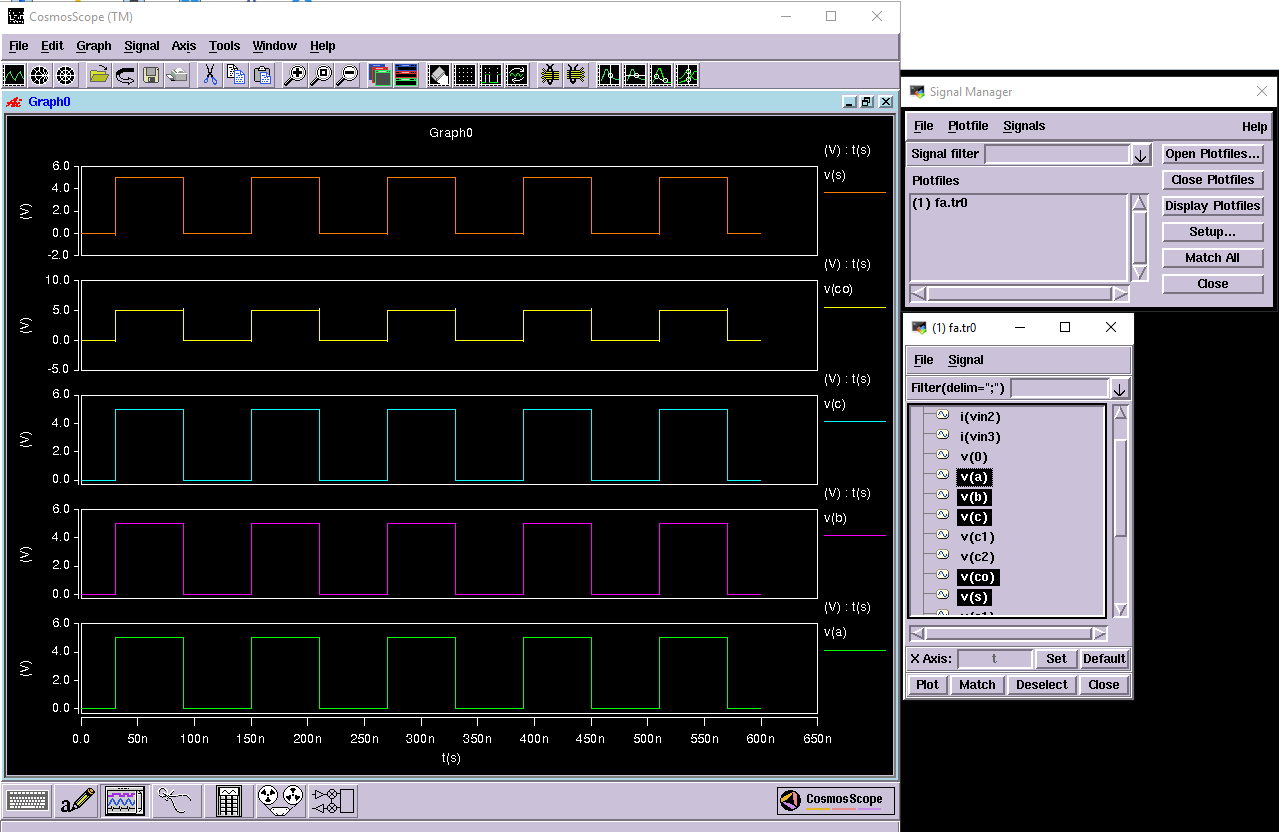




1. MobaXterm Job Done. Netlist successfully complied.



1. CosmosScope output: Simulating the output when all the inputs are 1. The outputs must be S=1 C=1



# Discussion of Result

Full Adder Gate: The SUM is basically the XOR function of the three inputs, A, B and Cin. Carry\_out is the sum of the two AND functions. Hence, 2 XOR gates, 2 AND gates and 1 OR gate is required for implementation. Transistors required are as below:

AND: 7 X 2 =14 transistors

XOR: 12 X 2 = 24 transistors

OR: 6 Transistors

Total transistors required are 44.

# Conclusion

Through this lab tutorial we conclude that to implement a complicated circuit, we can break it down into smaller modules and use sub-circuit approach to build bigger modules. Full adder can be built using two Half adders or using 2 XOR gates, 2 AND gates and 1 OR gate.