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

Tutorial

Homework report

**TITLE**

Simulation using HSPICE for 4-Bit Full Adder gate using Sub-circuit

Date of Performing Experiment: 22nd October 2019

Due Date: 29th October 2019

**Student ID: 14344331**

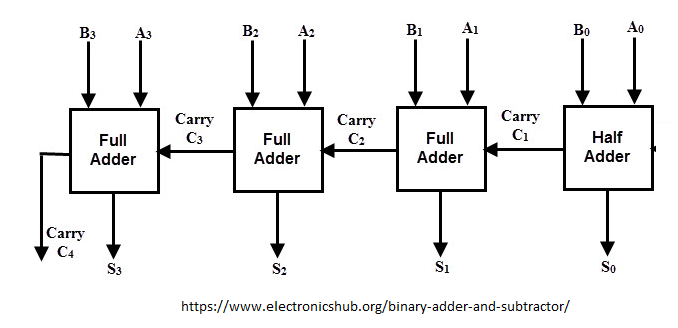
**Name: Prerana Samant**

# Objective:

To run HSPICE on Linux platform, to create netlist and simulation of 4-bit 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 a single bit adder. In a 4 bit adder, there will be 4 blocks of adders each individually acting as a Full Adder. For effective implementation of hardware, the first stage can be implemented using a Half Adder since, the first stage will not have a carry input.



The Half Adder will have below truth table for A0, B0

Equations:

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

Each full adder block above will operate as per below truth table for inputs A1, B1, C1, A2, B2, C2, A3, B3 and C3.

Equations:

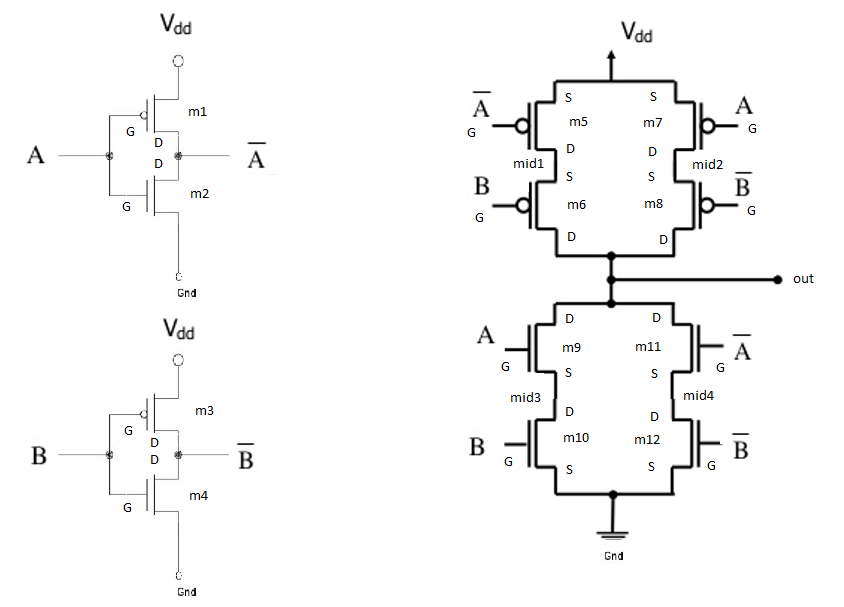
Truth Table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 |

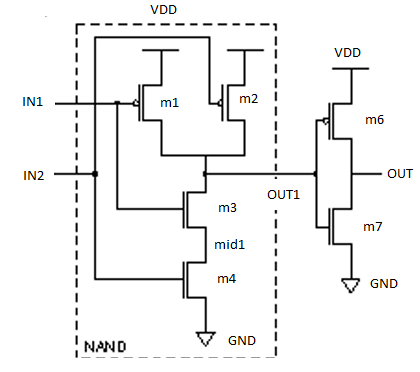
Equations:

The 4 bit adder can be implemented using below sub-circuits:

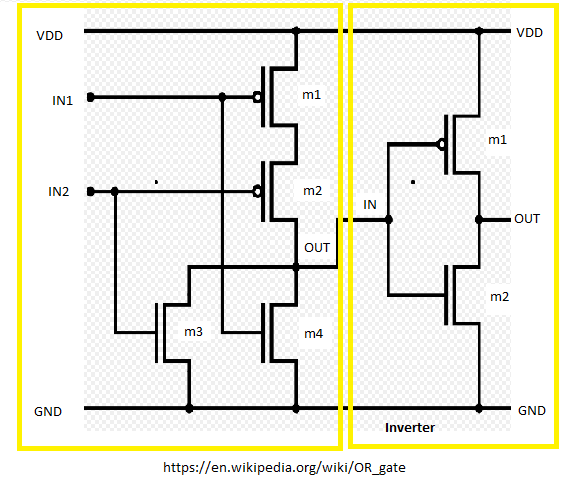
# CMOS implementation: XOR gate



# 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.
3. Make a sub-circuit block of Half Adder and Full Adder using above sub-circuits.
4. Connect the A0, B0 inputs to Half Adder. Cout of Half adder and A1, B1 to the first Full Adder block and so on.
5. Use the CMOS implementations in the figures above to implement the netlist of sub-circuits.
6. Define the voltage sources and nets for all the 8 inputs of the main 4-bit Full Adder block. Save it in the HSPICE folder by name “fa4bit.sp”. Extension “sp” is for input netlist.
7. 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 fa4bit.sp > fa4bit.lis

(inputfilename.sp > outputfilename.lis)

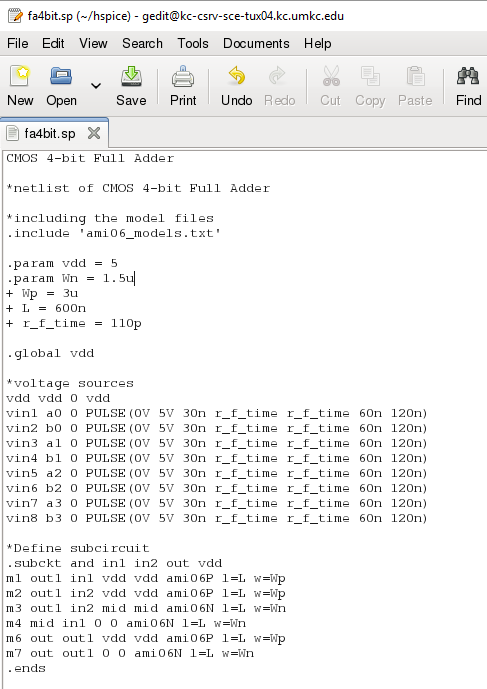
1. After this command you must get “hspice job concluded”. This means there are no errors. To check what errors are in the .sp file, open gedit🡪 open the .lis file
2. Now, simulate the netlist using CosmosScope. Type below command:

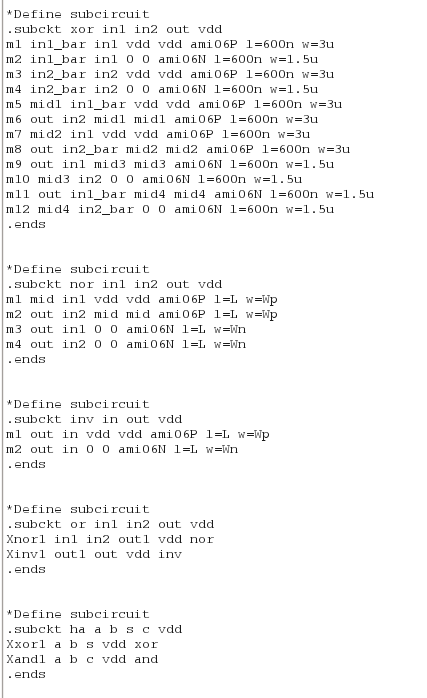
cscope &

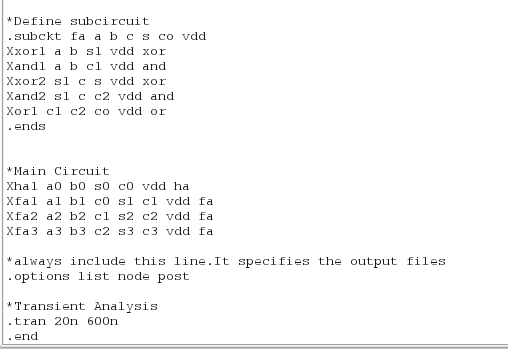
1. The simulator window will open. Click on File🡪 Open🡪Plotfiles.
2. Open the “fa4bit.tr” file. This file is the transient analysis graph data.
3. Select all the 8 inputs and outputs of main module and then click on Plot. Verify the graphs with truth table.
4. For simulating 4- bit full adder and easily 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 4 Bit Full Adder

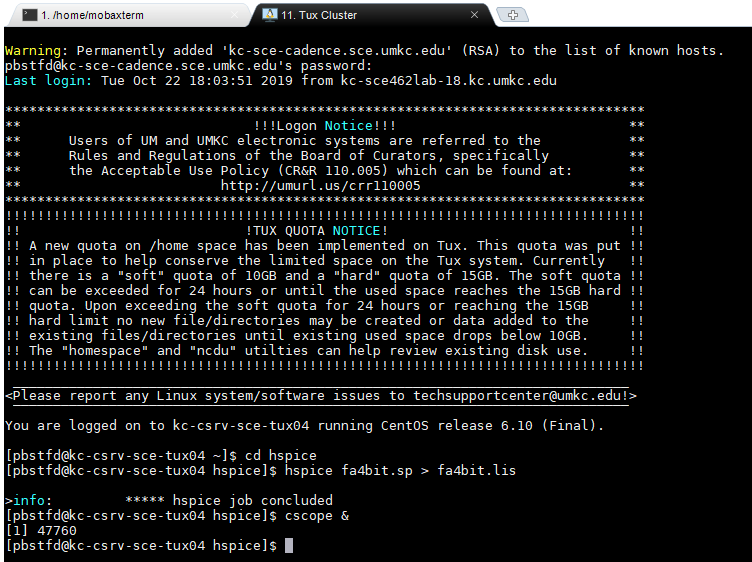
1. Gedit Code



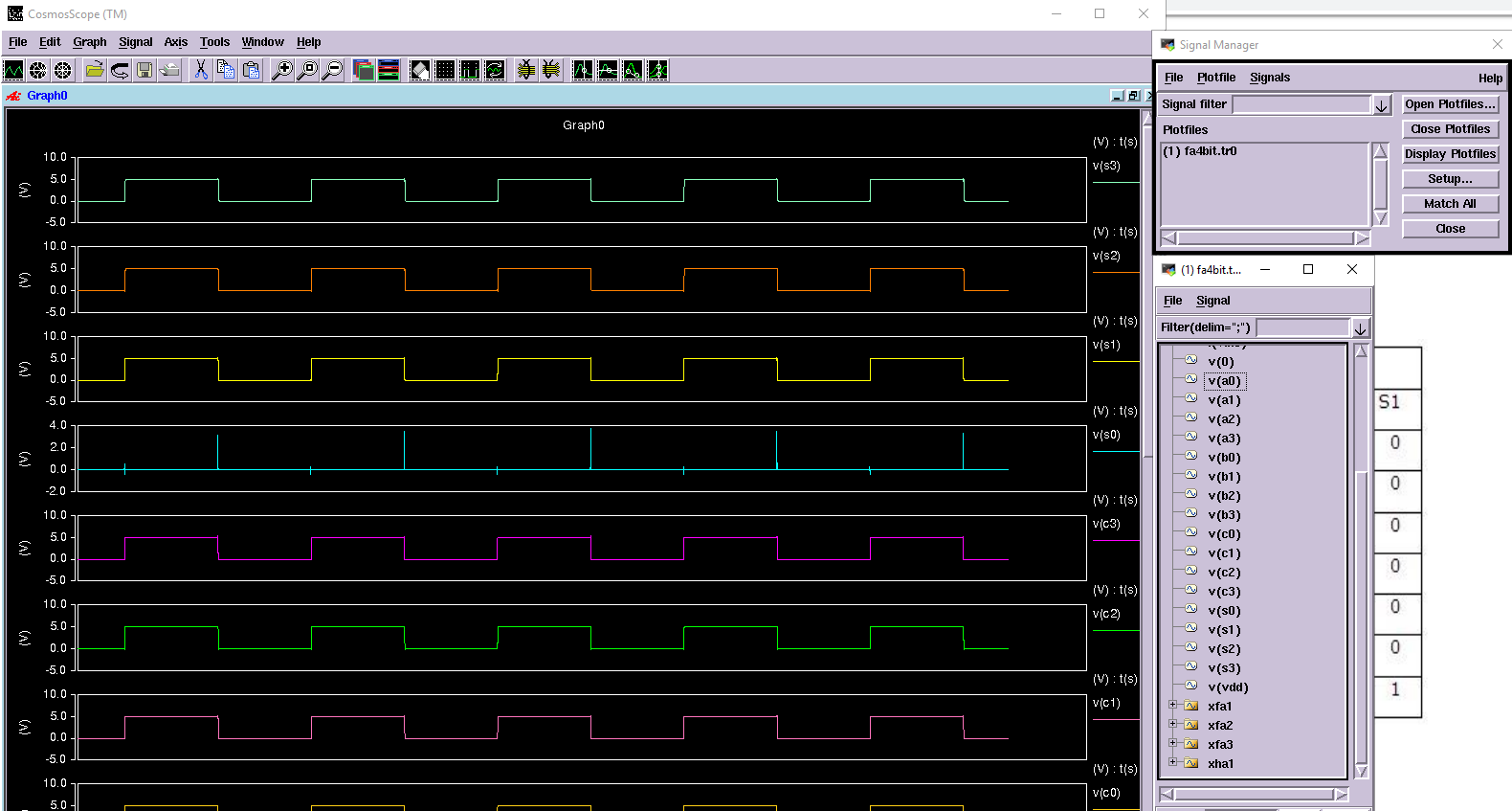




1. MobaXterm Job Done. Netlist successfully complied.



1. CosmosScope output: Simulating the output when all the inputs are 1. The outputs must be Sx=1 (except S0=0) Cx=1



# Discussion of Result

4-bit Full Adder: This block is implemented using 1 Half Adder and 3 Full Adders

Transistors required are as below:

1 Half Adder = 19 transistors.

1 Full Adder = 44 transistors. Hence, 3 FA = 132 transistors

Total Transistors required = 151.

# 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. By using a Half Adder for first bit we have saved total 25 transistors.