John A. Gangemi
U68714612
CMOS - VLSI Design Lab
CDA4213L Fall 2014
9/12/2014

# LAB 1 : INTRODUCTION TO SPICE

## Objectives

Lab one is primarily a review of digital logic design concepts and how design tools such as HSpice can facilitate the verification and simulation process of very large scale integration design using complementary metal-oxide semiconductor technology.

Per instructions provided in the lab design document students are to implement four combinational logic circuits using predefined circuits included in a HSpice netlist file generously provided by the lab instructors.

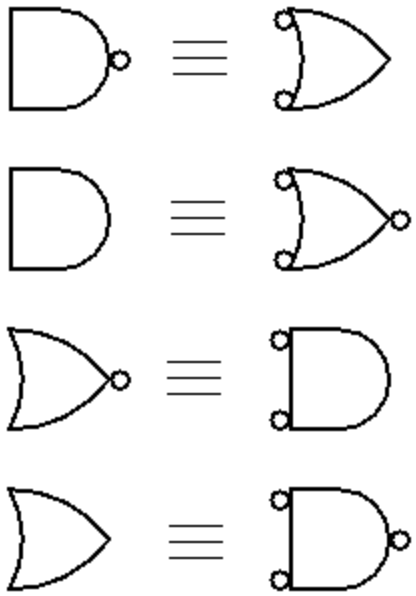
## Design and Analysis

The first step in the process of designing a combinational circuit is to develop a truth table and/or boolean expression that describes the output as a function of the inputs. Applying simplification techniques will provide a “minimal gate” realization of the circuit and through NAND/NOR conversion said circuit can adhere to the design specifications mentioned in the lab design document.

It should be noted that the minimal gate level representation of each combinational logic circuit is different than the final “design conforming” logic circuits depicted. Showing the NAND/NOR conversions was deemed unnecessary as the intermediate gate level representations were not used for implementing the HSpice netlists.

In regards to part three of the lab design document the combinational logic circuits were simulated with HSpice using VECTOR input simulation stimuli. The decision to choose such stimuli was based on its similarity to a truth table. This ensured that all possible combinations of inputs are tested in a 80 nanoseconds period with 10ns per row of the truth table.

NAND/NOR conversions:



I. 3-Input NAND

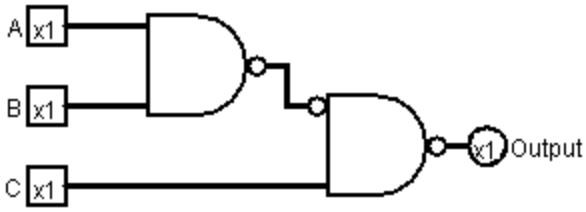
| A | B | C | Z |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

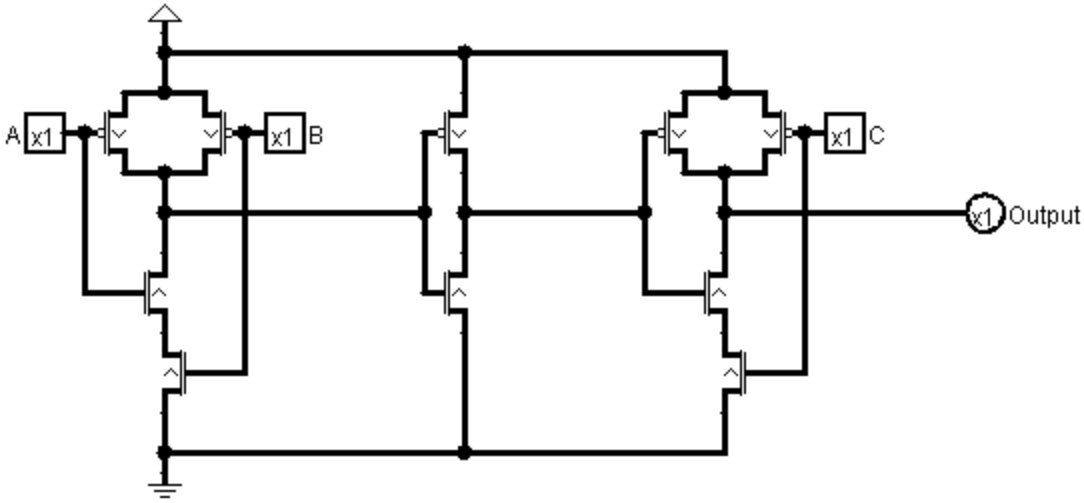
| | AB | | | |
|---|---|---|---|---|
| C | 0 0 | 0 1 | 1 1 | 1 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |

From the K-map the following boolean expression was derived:

$$Z = A' + B' + C'$$

Implementing the boolean expression for Z to create a gate-level representation lead to the following combinational logic circuit after applying NAND/NOR conversions. CMOS transistor level translation of the gate-network is also provided.

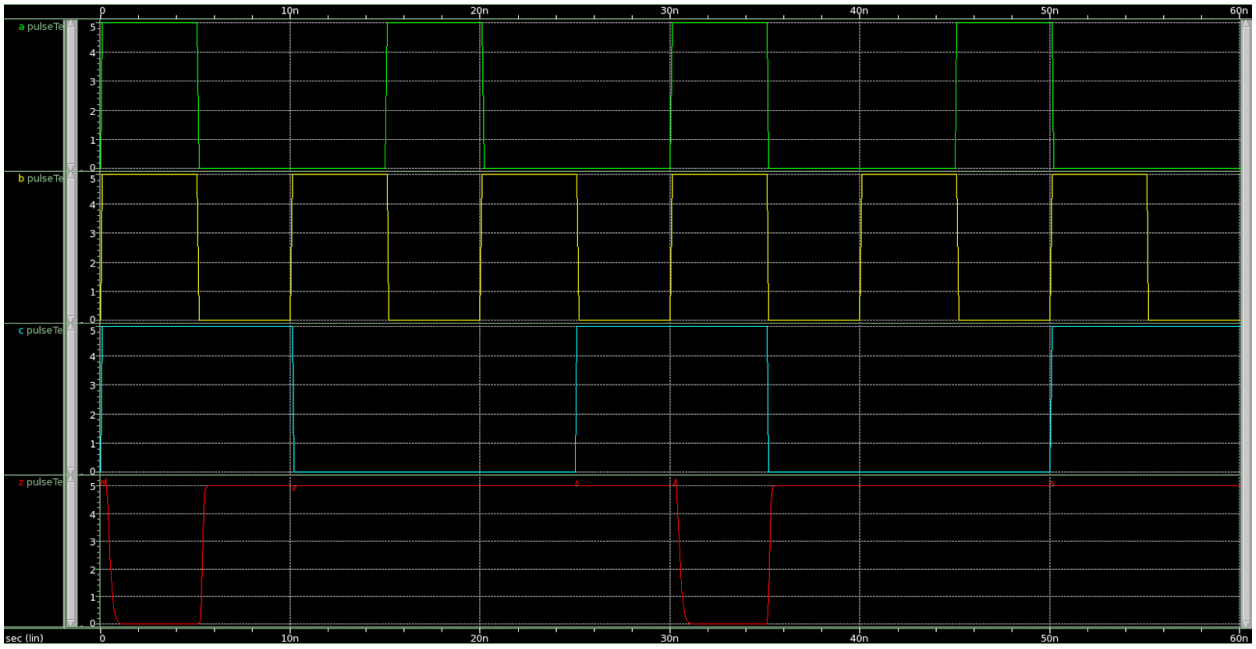


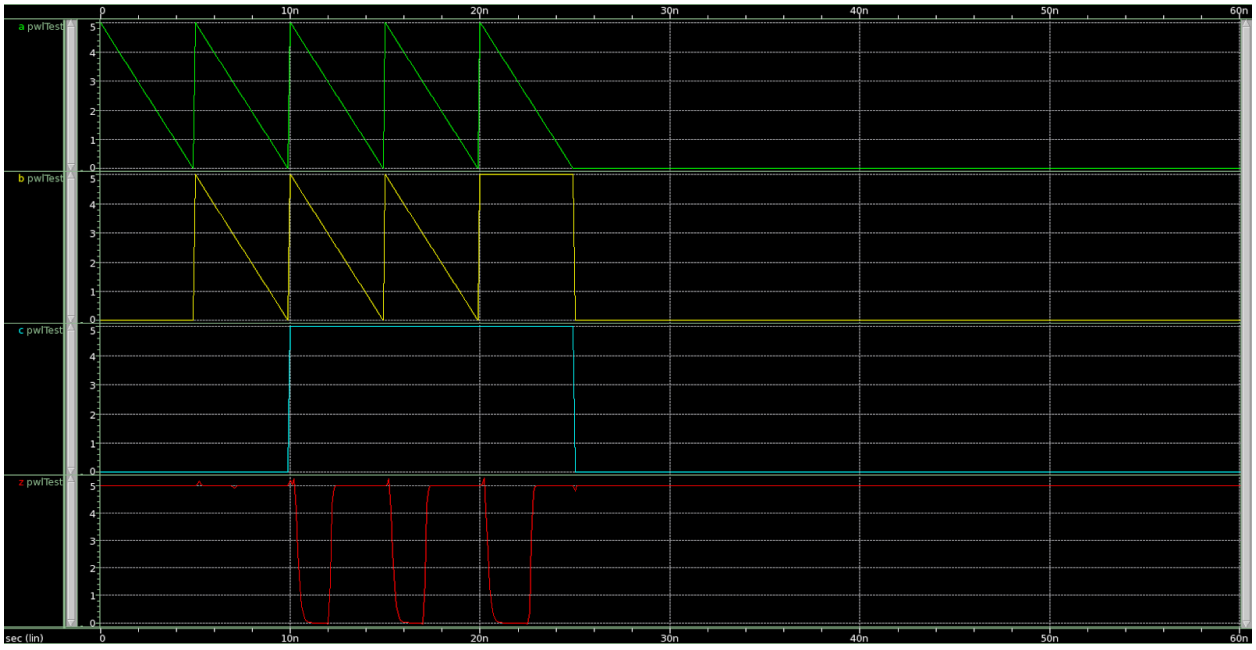

In regards to part two of the lab design document different input simulation stimuli were tested and the results verified against the logic developed from the truth table.

The green, yellow, and blue waveforms depicted are the inputs and the red waveform is the output. Accordingly the output is a function of the inputs and does not display erratic behavior nor incorrect timing for each of the following screen dumps provided from WaveExplorer software.

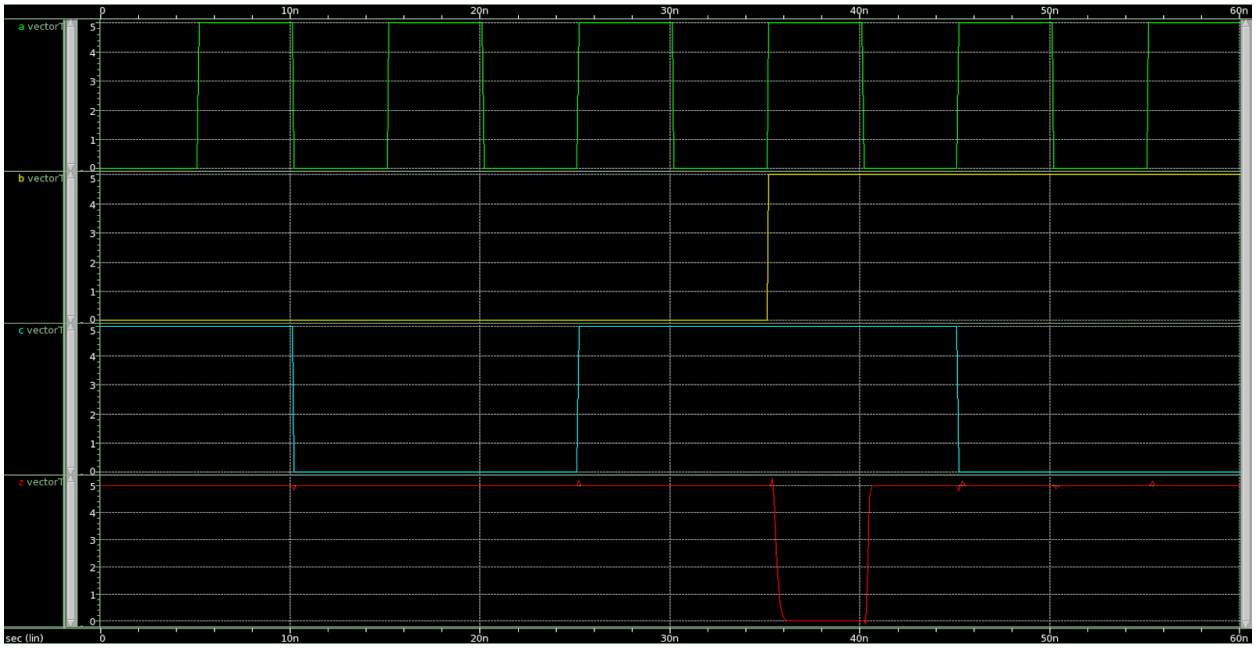
Simulating the design with HSpice using PULSE:




Simulating the design with HSpice using PWL:




Simulating the design with HSpice using VECTOR:




II. 3-Input NOR

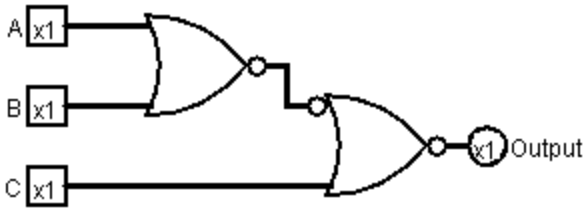
| A | B | C | Z |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

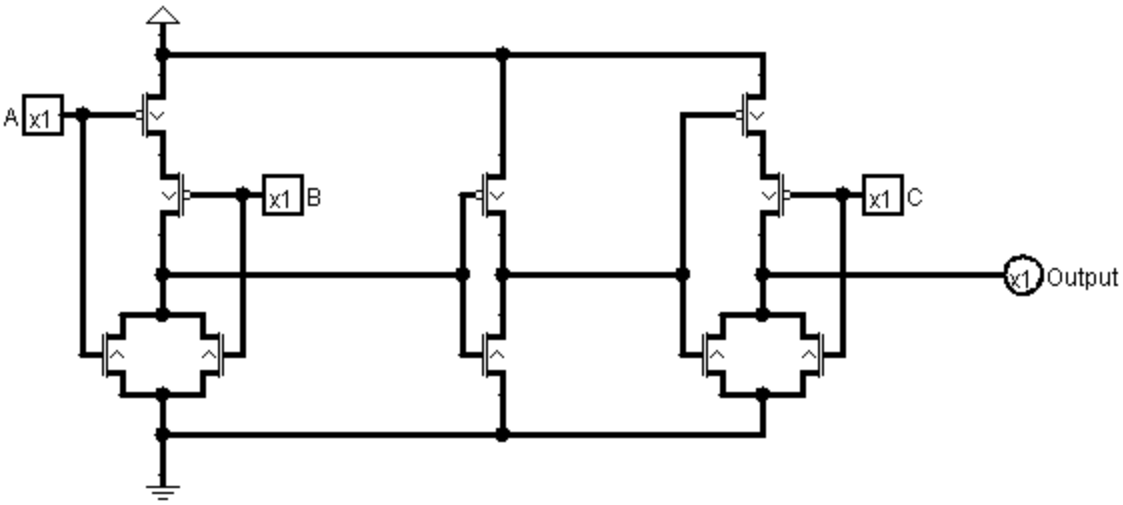
| | AB | | | |
|---|---|---|---|---|
| C | 0 0 | 0 1 | 1 1 | 1 0 |
| 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |

From the K-map the following boolean expression was derived:

Z = A'B'C'

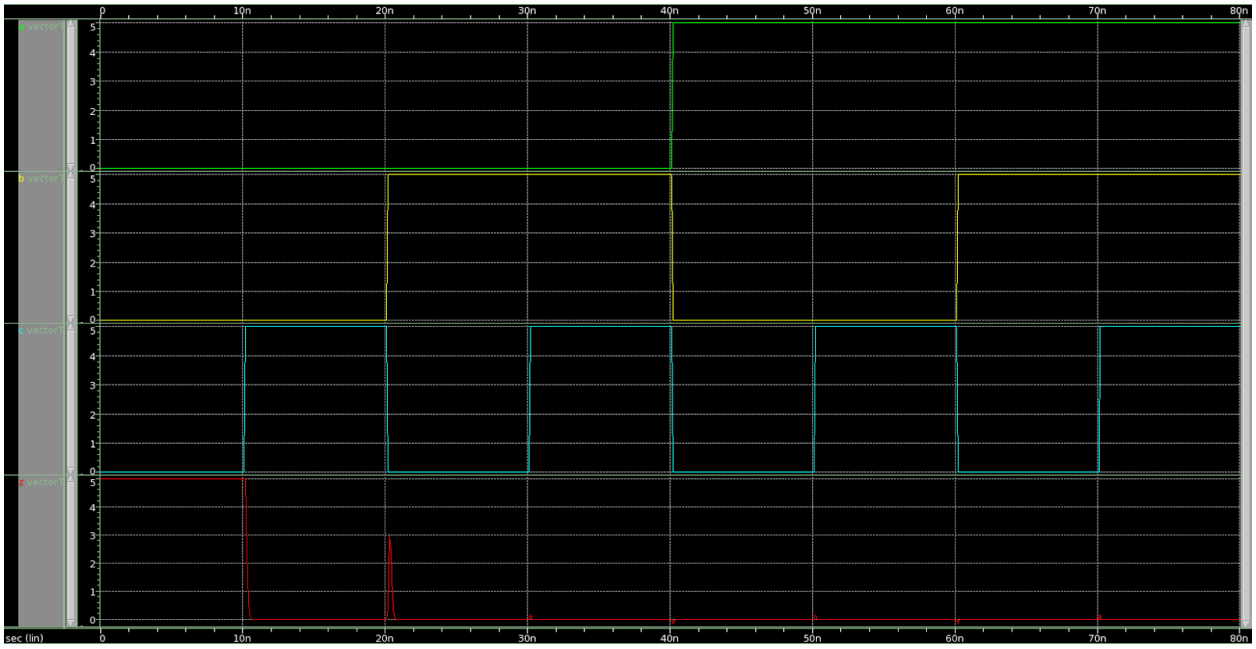
Implementing the boolean expression for Z to create a gate-level representation lead to the following combinational logic circuit after applying NAND/NOR conversions. CMOS transistor level translation of the gate-network is also provided.




The green, yellow, and blue waveforms depicted are the inputs and the red waveform is the output. Accordingly the output is a function of the inputs however it displays erratic behavior but not incorrect timing for the following screen dumps provided from WaveExplorer software.

Simulation with HSpice using VECTOR:




III. 3-Input XOR

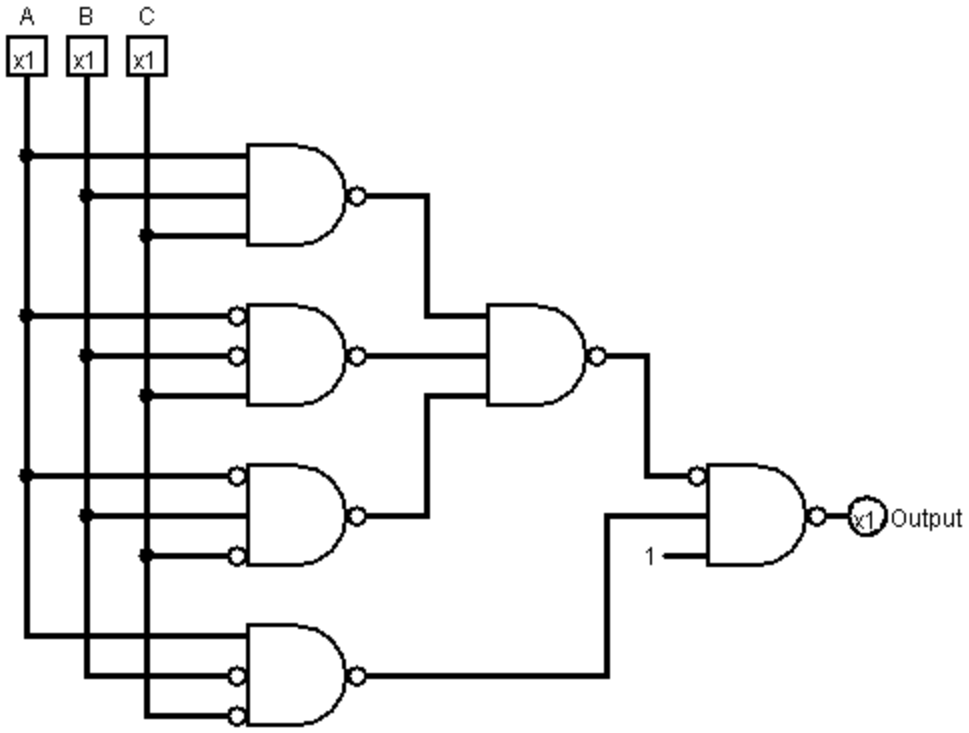
| A | B | Q = A XOR B | Z = Q XOR C |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 |

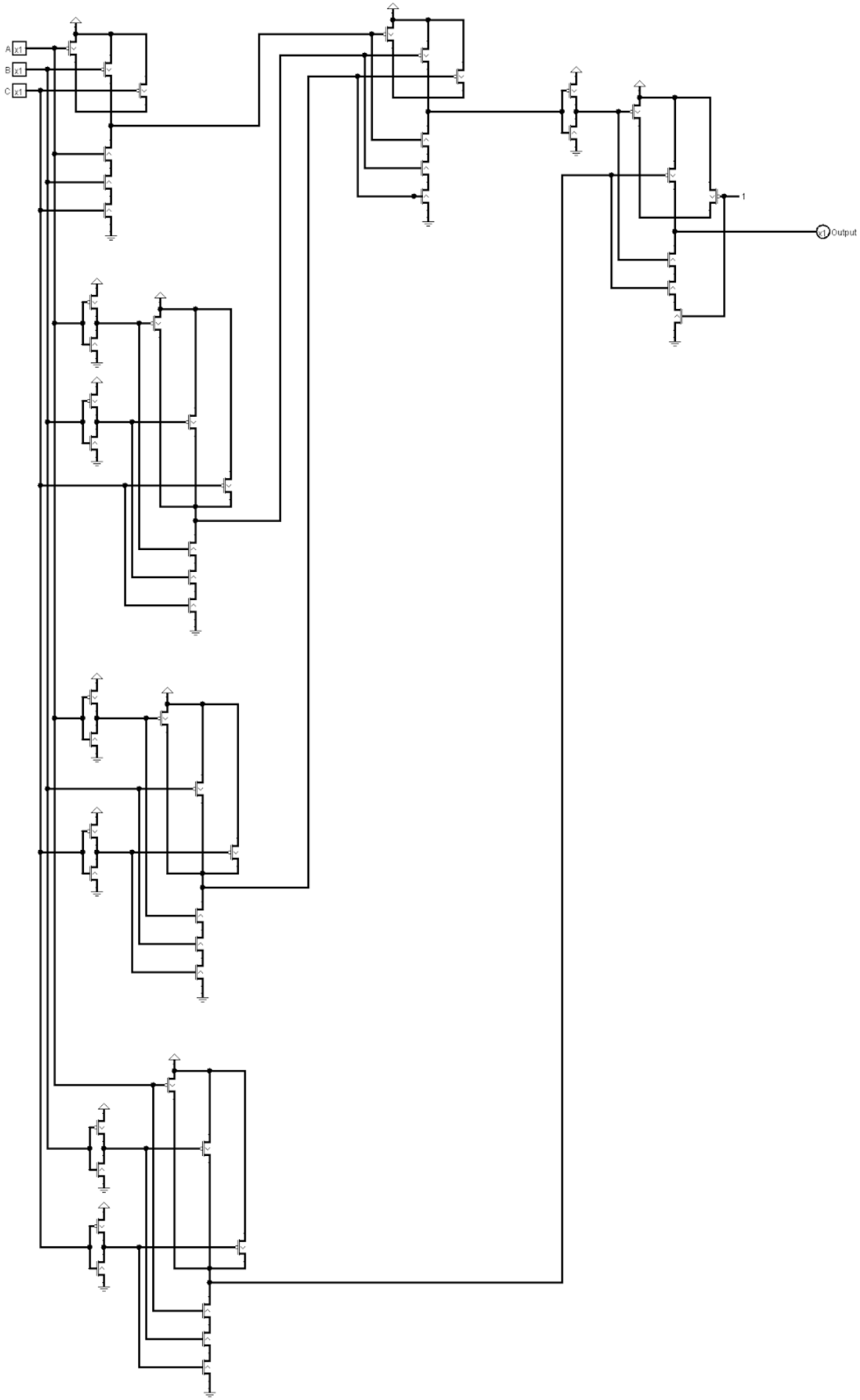
| | AB | | | |
|---|---|---|---|---|
| C | 0 0 | 0 1 | 1 1 | 1 0 |
| 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |

From the K-map the following boolean expression was derived:

Z = ABC + A'B'C + A'BC' + AB'C'

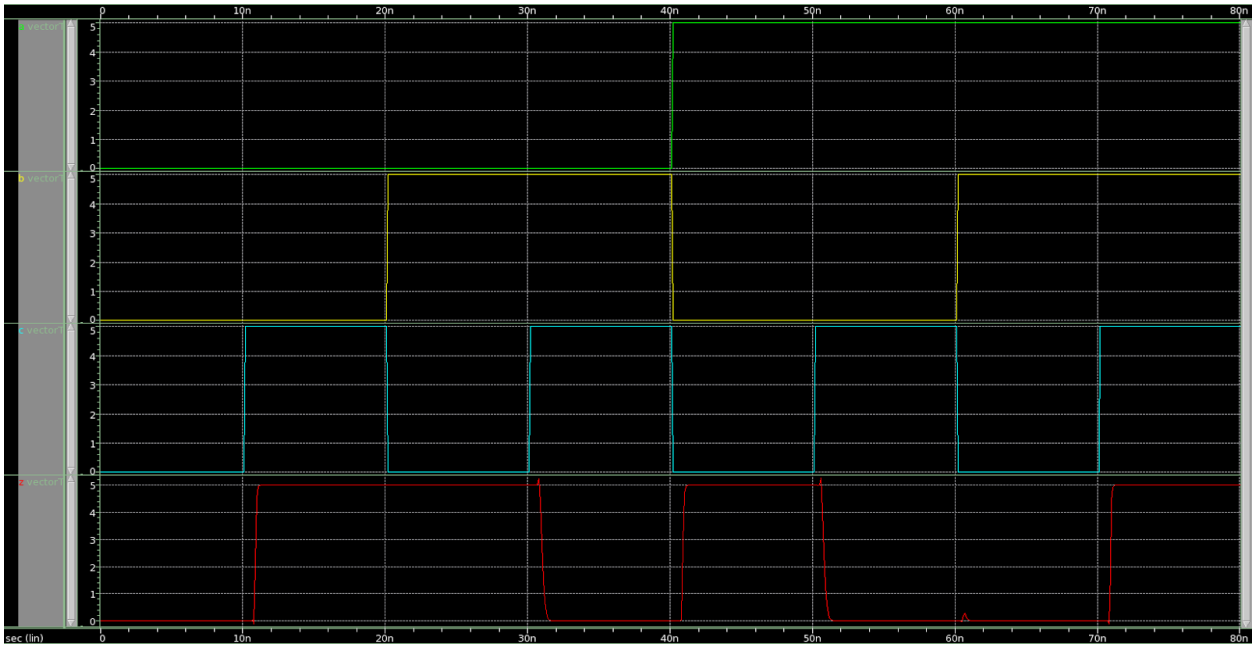
Implementing the boolean expression for Z to create a gate-level representation lead to the following combinational logic circuit after applying NAND/NOR conversions. CMOS transistor level translation of the gate-network is also provided.




A
x1
B
x1
C
x1
1
x1
Output

The green, yellow, and blue waveforms depicted are the inputs and the red waveform is the output. The waveforms show the appropriate timings for the circuit.

Simulating the design with HSpice using VECTOR:



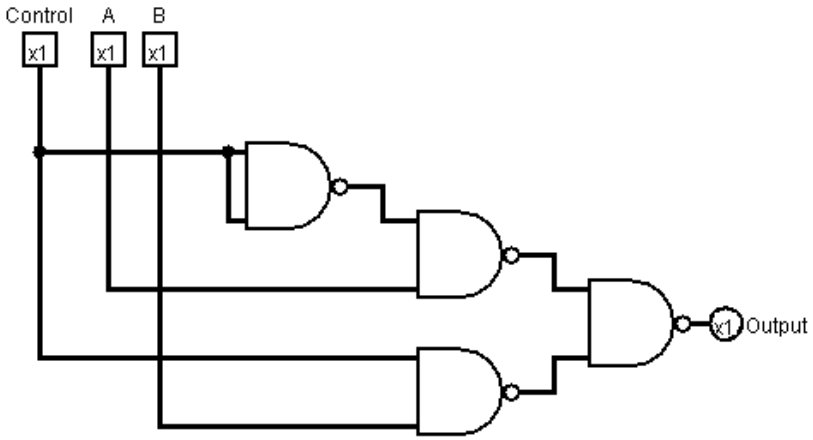

IV. 2-to-1 MUX

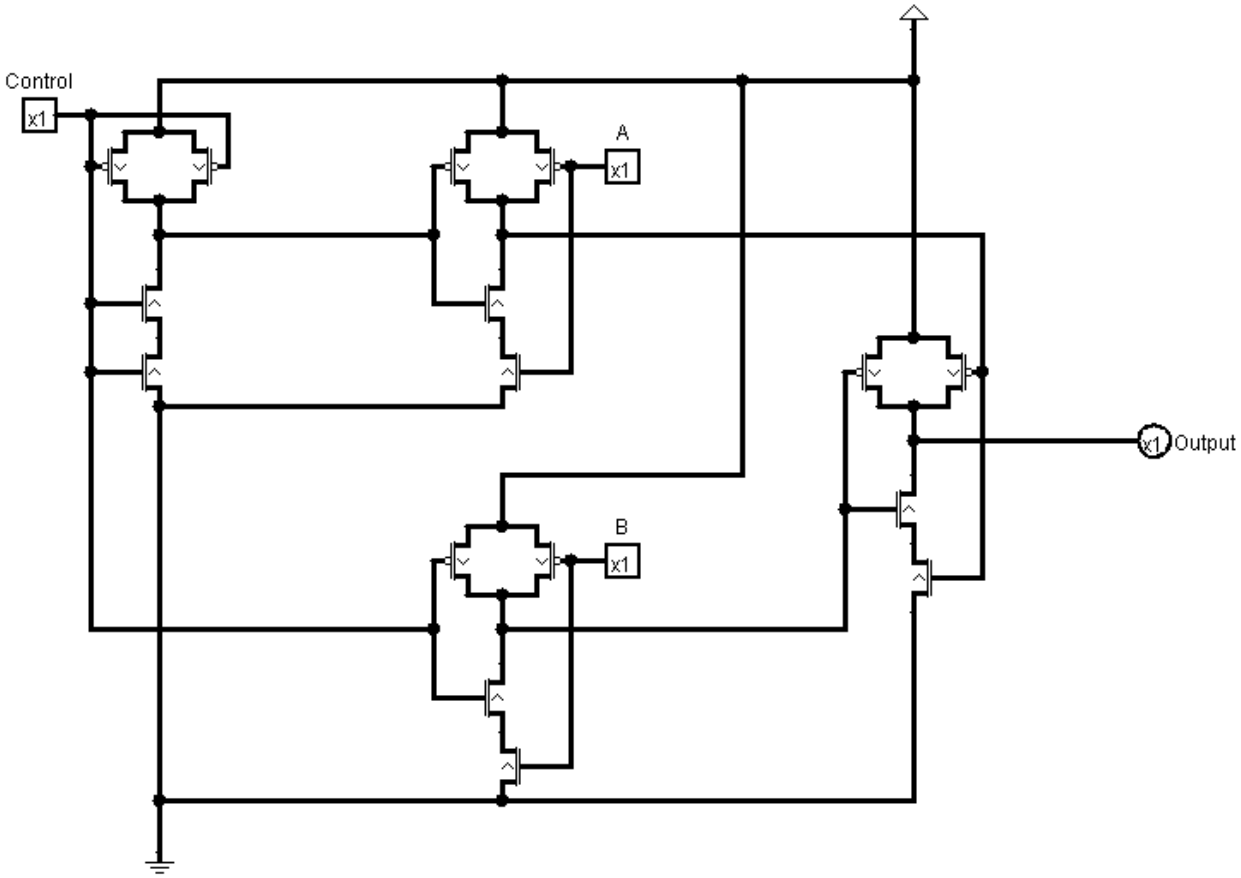
| S | A | B | Z |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

From the truth table the following expression was derived using the laws of Boolean algebra:

$$Z = S'AB' + S'AB + SA'B + SAB$$
$$Z = S'(AB' + AB) + S(A'B + AB)$$
$$Z = S'A + SB$$

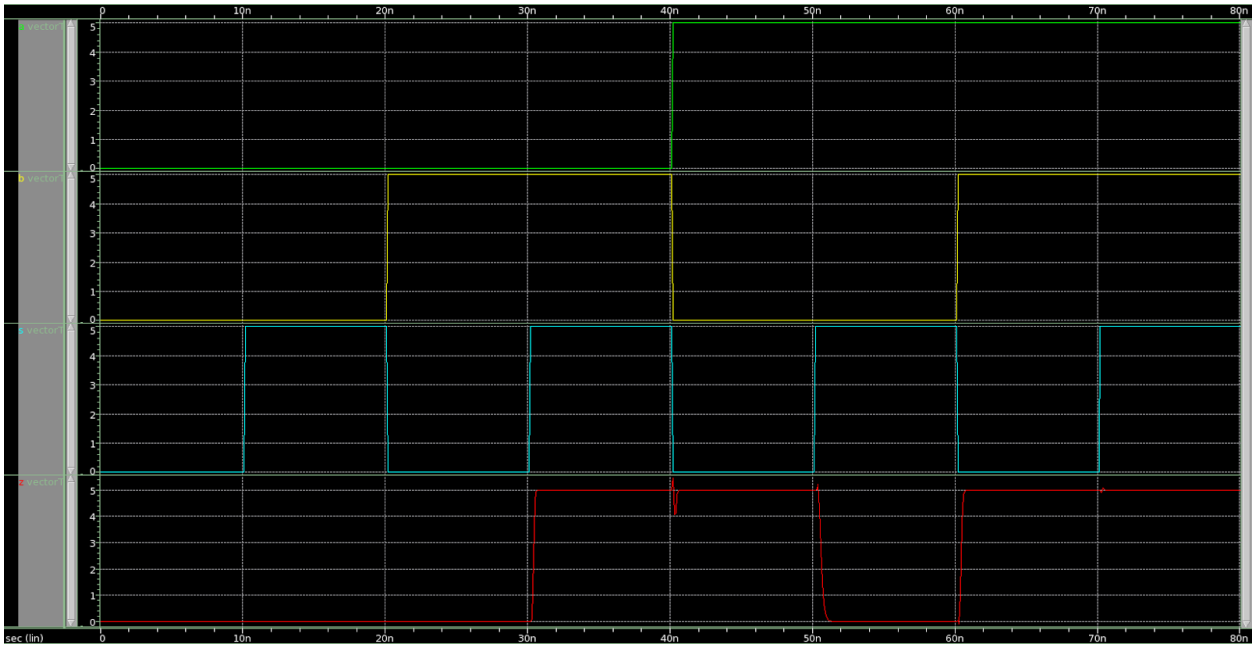
Implementing the boolean expression for Z to create a gate-level representation lead to the following combinational logic circuit after applying NAND/NOR conversions. CMOS transistor level translation of the gate-network is also provided.


Control
A
B
x1
x1
x1
x1
Output


Control
x1
A
x1
B
x1
x1
Output

The green, yellow, and blue waveforms depicted are the inputs and the red waveform is the output. The waveforms show the appropriate timings for the circuit but there exists a small “glitch” at the 40 nanosecond mark for the output.

Simulating the design with HSpice using VECTOR:




Observations

By using the HSpice software extensively the past week I do feel that the knowledge gained will benefit tremendously in future HSpice simulations. I fully understand the importance and correct use of the three main files used in simulation (netlist, spice deck, and vector).