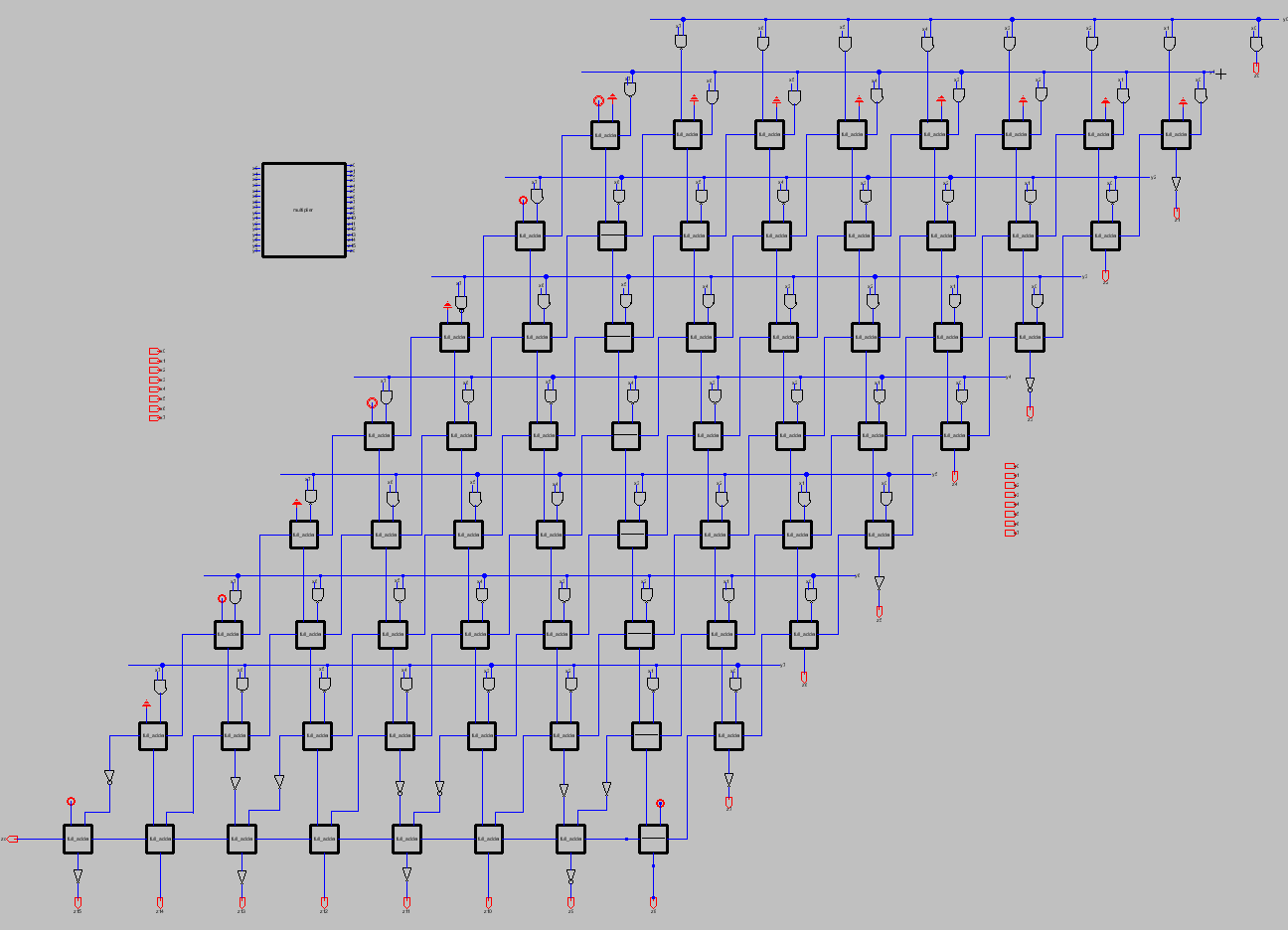
**EE5311 - Digital IC Design**

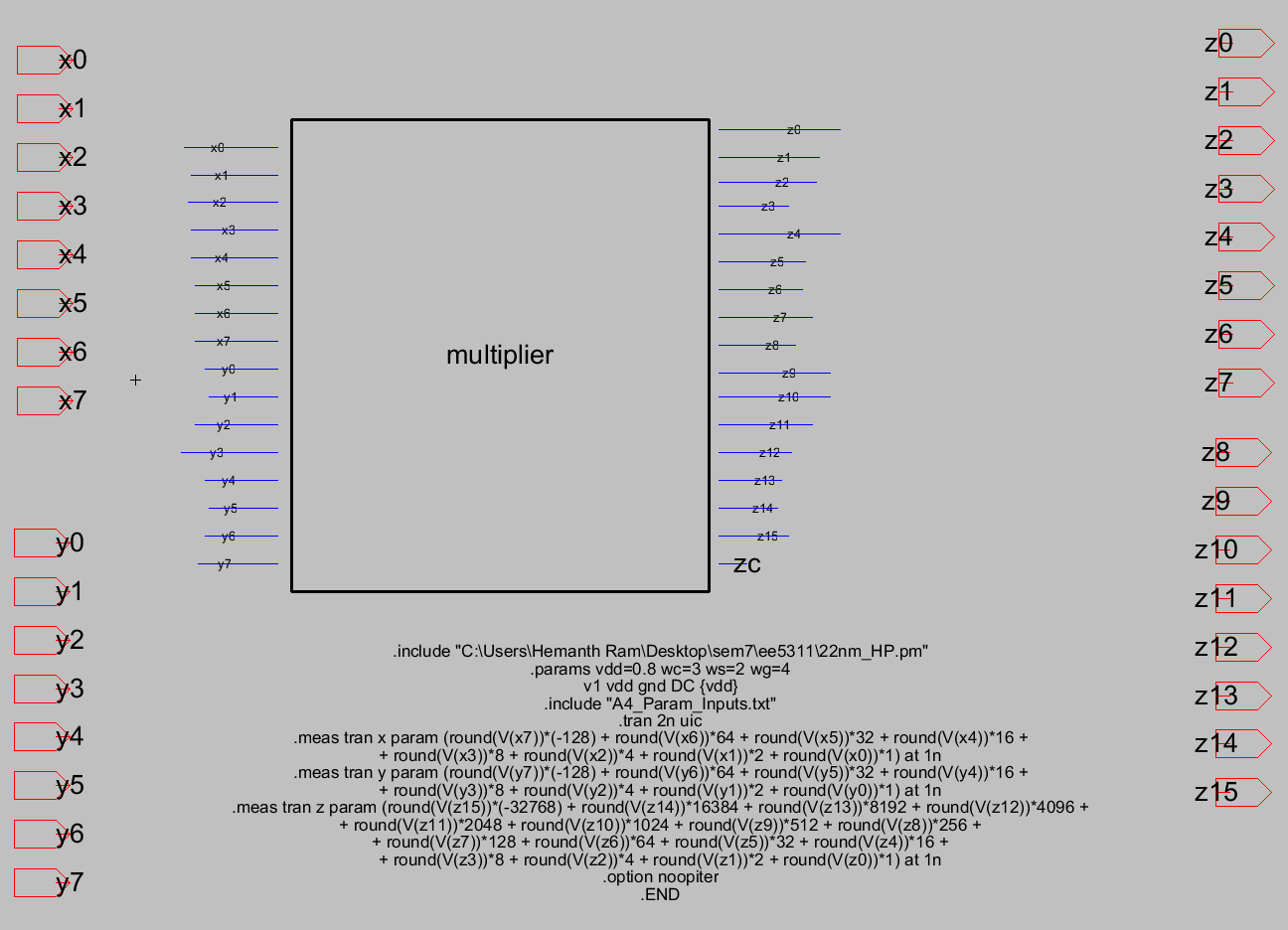
**Assignment 4 – Schematic of a signed 8-bit Carry Save Multiplier**

**Member 1: Srivenkat A(EE18B038)  
Member 2: Hemanth Ram G K(EE18B132)  
Member 3: Sidesh S(EE18B032)**

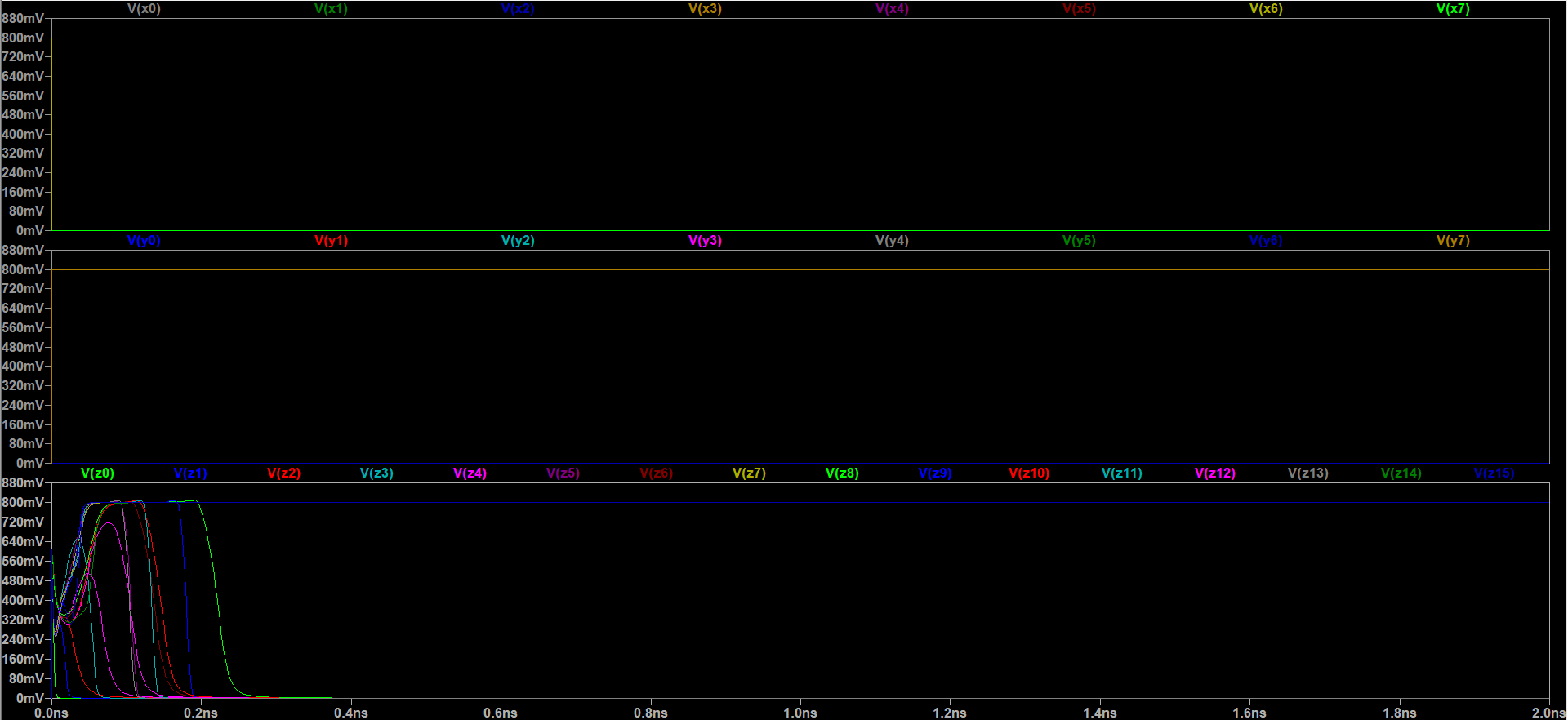
## Schematic of CSM



**Multiplier Testbench**



**Simulation output for 127 \* -128**

****

**Interpreted voltage values for 127 \* -128**

**x: (round(v(x7))\*(-128) + round(v(x6))\*64 + round(v(x5))\*32 + round(v(x4))\*16 + round(v(x3))\*8 + round(v(x2))\*4 + round(v(x1))\*2 + round(v(x0))\*1)=127**

**y: (round(v(y7))\*(-128) + round(v(y6))\*64 + round(v(y5))\*32 + round(v(y4))\*16 + round(v(y3))\*8 + round(v(y2))\*4 + round(v(y1))\*2 + round(v(y0))\*1)=-128**

**z: (round(v(z15))\*(-32768) + round(v(z14))\*16384 + round(v(z13))\*8192 + round(v(z12))\*4096 + round(v(z11))\*2048 + round(v(z10))\*1024 + round(v(z9))\*512 + round(v(z8))\*256 + round(v(z7))\*128 + round(v(z6))\*64 + round(v(z5))\*32 + round(v(z4))\*16 + round(v(z3))\*8 + round(v(z2))\*4 + round(v(z1))\*2 + round(v(z0))\*1)=-16256**

**Interpreted voltage values for -128 \* 127**

**x: (round(v(x7))\*(-128) + round(v(x6))\*64 + round(v(x5))\*32 + round(v(x4))\*16 + round(v(x3))\*8 + round(v(x2))\*4 + round(v(x1))\*2 + round(v(x0))\*1)=-128**

**y: (round(v(y7))\*(-128) + round(v(y6))\*64 + round(v(y5))\*32 + round(v(y4))\*16 + round(v(y3))\*8 + round(v(y2))\*4 + round(v(y1))\*2 + round(v(y0))\*1)=127**

**z: (round(v(z15))\*(-32768) + round(v(z14))\*16384 + round(v(z13))\*8192 + round(v(z12))\*4096 + round(v(z11))\*2048 + round(v(z10))\*1024 + round(v(z9))\*512 + round(v(z8))\*256 + round(v(z7))\*128 + round(v(z6))\*64 + round(v(z5))\*32 + round(v(z4))\*16 + round(v(z3))\*8 + round(v(z2))\*4 + round(v(z1))\*2 + round(v(z0))\*1)=-16256**

**Interpreted voltage values for 0 \* 127**

**x: (round(v(x7))\*(-128) + round(v(x6))\*64 + round(v(x5))\*32 + round(v(x4))\*16 + round(v(x3))\*8 + round(v(x2))\*4 + round(v(x1))\*2 + round(v(x0))\*1)=0**

**y: (round(v(y7))\*(-128) + round(v(y6))\*64 + round(v(y5))\*32 + round(v(y4))\*16 + round(v(y3))\*8 + round(v(y2))\*4 + round(v(y1))\*2 + round(v(y0))\*1)=127**

**z: (round(v(z15))\*(-32768) + round(v(z14))\*16384 + round(v(z13))\*8192 + round(v(z12))\*4096 + round(v(z11))\*2048 + round(v(z10))\*1024 + round(v(z9))\*512 + round(v(z8))\*256 + round(v(z7))\*128 + round(v(z6))\*64 + round(v(z5))\*32 + round(v(z4))\*16 + round(v(z3))\*8 + round(v(z2))\*4 + round(v(z1))\*2 + round(v(z0))\*1)=0**

**Interpreted voltage values for -128 \* 0**

**x: (round(v(x7))\*(-128) + round(v(x6))\*64 + round(v(x5))\*32 + round(v(x4))\*16 + round(v(x3))\*8 + round(v(x2))\*4 + round(v(x1))\*2 + round(v(x0))\*1)=-128**

**y: (round(v(y7))\*(-128) + round(v(y6))\*64 + round(v(y5))\*32 + round(v(y4))\*16 + round(v(y3))\*8 + round(v(y2))\*4 + round(v(y1))\*2 + round(v(y0))\*1)=0**

**z: (round(v(z15))\*(-32768) + round(v(z14))\*16384 + round(v(z13))\*8192 + round(v(z12))\*4096 + round(v(z11))\*2048 + round(v(z10))\*1024 + round(v(z9))\*512 + round(v(z8))\*256 + round(v(z7))\*128 + round(v(z6))\*64 + round(v(z5))\*32 + round(v(z4))\*16 + round(v(z3))\*8 + round(v(z2))\*4 + round(v(z1))\*2 + round(v(z0))\*1)=0**

**Interpreted voltage values for -12 \* 13**

**x: (round(v(x7))\*(-128) + round(v(x6))\*64 + round(v(x5))\*32 + round(v(x4))\*16 + round(v(x3))\*8 + round(v(x2))\*4 + round(v(x1))\*2 + round(v(x0))\*1)=-12**

**y: (round(v(y7))\*(-128) + round(v(y6))\*64 + round(v(y5))\*32 + round(v(y4))\*16 + round(v(y3))\*8 + round(v(y2))\*4 + round(v(y1))\*2 + round(v(y0))\*1)=13**

**z: (round(v(z15))\*(-32768) + round(v(z14))\*16384 + round(v(z13))\*8192 + round(v(z12))\*4096 + round(v(z11))\*2048 + round(v(z10))\*1024 + round(v(z9))\*512 + round(v(z8))\*256 + round(v(z7))\*128 + round(v(z6))\*64 + round(v(z5))\*32 + round(v(z4))\*16 + round(v(z3))\*8 + round(v(z2))\*4 + round(v(z1))\*2 + round(v(z0))\*1)=-156**

**Interpreted voltage values for -1 \* -1**

**x: (round(v(x7))\*(-128) + round(v(x6))\*64 + round(v(x5))\*32 + round(v(x4))\*16 + round(v(x3))\*8 + round(v(x2))\*4 + round(v(x1))\*2 + round(v(x0))\*1)=-1**

**y: (round(v(y7))\*(-128) + round(v(y6))\*64 + round(v(y5))\*32 + round(v(y4))\*16 + round(v(y3))\*8 + round(v(y2))\*4 + round(v(y1))\*2 + round(v(y0))\*1)=-1**

**z: (round(v(z15))\*(-32768) + round(v(z14))\*16384 + round(v(z13))\*8192 + round(v(z12))\*4096 + round(v(z11))\*2048 + round(v(z10))\*1024 + round(v(z9))\*512 + round(v(z8))\*256 + round(v(z7))\*128 + round(v(z6))\*64 + round(v(z5))\*32 + round(v(z4))\*16 + round(v(z3))\*8 + round(v(z2))\*4 + round(v(z1))\*2 + round(v(z0))\*1)=1**

**Interpreted voltage values for -128 \* -128**

**x: (round(v(x7))\*(-128) + round(v(x6))\*64 + round(v(x5))\*32 + round(v(x4))\*16 + round(v(x3))\*8 + round(v(x2))\*4 + round(v(x1))\*2 + round(v(x0))\*1)=-128**

**y: (round(v(y7))\*(-128) + round(v(y6))\*64 + round(v(y5))\*32 + round(v(y4))\*16 + round(v(y3))\*8 + round(v(y2))\*4 + round(v(y1))\*2 + round(v(y0))\*1)=-128**

**z: (round(v(z15))\*(-32768) + round(v(z14))\*16384 + round(v(z13))\*8192 + round(v(z12))\*4096 + round(v(z11))\*2048 + round(v(z10))\*1024 + round(v(z9))\*512 + round(v(z8))\*256 + round(v(z7))\*128 + round(v(z6))\*64 + round(v(z5))\*32 + round(v(z4))\*16 + round(v(z3))\*8 + round(v(z2))\*4 + round(v(z1))\*2 + round(v(z0))\*1)=16384**

**Interpreted voltage values for 103 \* -57**

**x: (round(v(x7))\*(-128) + round(v(x6))\*64 + round(v(x5))\*32 + round(v(x4))\*16 + round(v(x3))\*8 + round(v(x2))\*4 + round(v(x1))\*2 + round(v(x0))\*1)=103**

**y: (round(v(y7))\*(-128) + round(v(y6))\*64 + round(v(y5))\*32 + round(v(y4))\*16 + round(v(y3))\*8 + round(v(y2))\*4 + round(v(y1))\*2 + round(v(y0))\*1)=-57**

**z: (round(v(z15))\*(-32768) + round(v(z14))\*16384 + round(v(z13))\*8192 + round(v(z12))\*4096 + round(v(z11))\*2048 + round(v(z10))\*1024 + round(v(z9))\*512 + round(v(z8))\*256 + round(v(z7))\*128 + round(v(z6))\*64 + round(v(z5))\*32 + round(v(z4))\*16 + round(v(z3))\*8 + round(v(z2))\*4 + round(v(z1))\*2 + round(v(z0))\*1)=-5871**

**Interpreted voltage values for -50 \* 50**

**x: (round(v(x7))\*(-128) + round(v(x6))\*64 + round(v(x5))\*32 + round(v(x4))\*16 + round(v(x3))\*8 + round(v(x2))\*4 + round(v(x1))\*2 + round(v(x0))\*1)=-50**

**y: (round(v(y7))\*(-128) + round(v(y6))\*64 + round(v(y5))\*32 + round(v(y4))\*16 + round(v(y3))\*8 + round(v(y2))\*4 + round(v(y1))\*2 + round(v(y0))\*1)=50**

**z: (round(v(z15))\*(-32768) + round(v(z14))\*16384 + round(v(z13))\*8192 + round(v(z12))\*4096 + round(v(z11))\*2048 + round(v(z10))\*1024 + round(v(z9))\*512 + round(v(z8))\*256 + round(v(z7))\*128 + round(v(z6))\*64 + round(v(z5))\*32 + round(v(z4))\*16 + round(v(z3))\*8 + round(v(z2))\*4 + round(v(z1))\*2 + round(v(z0))\*1)=-2500**

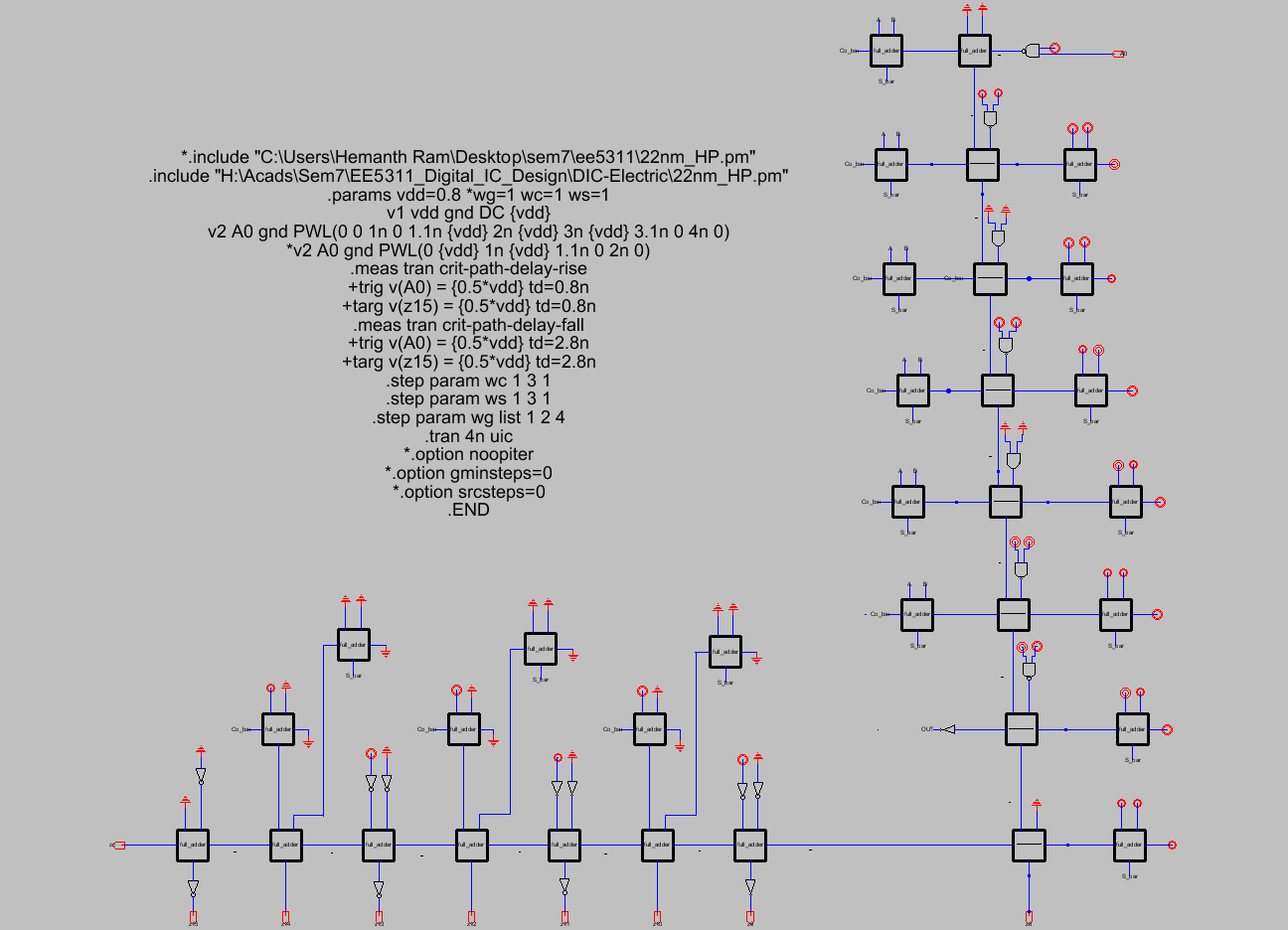
**Interpreted voltage values for 79 \* 81**

**x: (round(v(x7))\*(-128) + round(v(x6))\*64 + round(v(x5))\*32 + round(v(x4))\*16 + round(v(x3))\*8 + round(v(x2))\*4 + round(v(x1))\*2 + round(v(x0))\*1)=79**

**y: (round(v(y7))\*(-128) + round(v(y6))\*64 + round(v(y5))\*32 + round(v(y4))\*16 + round(v(y3))\*8 + round(v(y2))\*4 + round(v(y1))\*2 + round(v(y0))\*1)=81**

**z: (round(v(z15))\*(-32768) + round(v(z14))\*16384 + round(v(z13))\*8192 + round(v(z12))\*4096 + round(v(z11))\*2048 + round(v(z10))\*1024 + round(v(z9))\*512 + round(v(z8))\*256 + round(v(z7))\*128 + round(v(z6))\*64 + round(v(z5))\*32 + round(v(z4))\*16 + round(v(z3))\*8 + round(v(z2))\*4 + round(v(z1))\*2 + round(v(z0))\*1)=6399**

**Critical Path of CSM modelled separately**



**Inputs to Full Adders in Critical Path**

1. Input Cin is closer than A, B to output and critical input is connected to Cin in every full adder.
2. For an edge in Cin, sum delay for different combinations of A, B in a Full Adder:

|  |  |  |
| --- | --- | --- |
| Inputs to (A,B ) \ Cin Edge Type | Rising | Falling |
| 00 | 62.6ps | 26.7ps |
| 01 | 46.9ps | 34.7ps |
| 10 | 43.3ps | 35.2ps |
| 11 | 30.1ps | 49.4ps |

So, for max delay in sum propagation over both rising and falling edges, fix A=0, B=0.

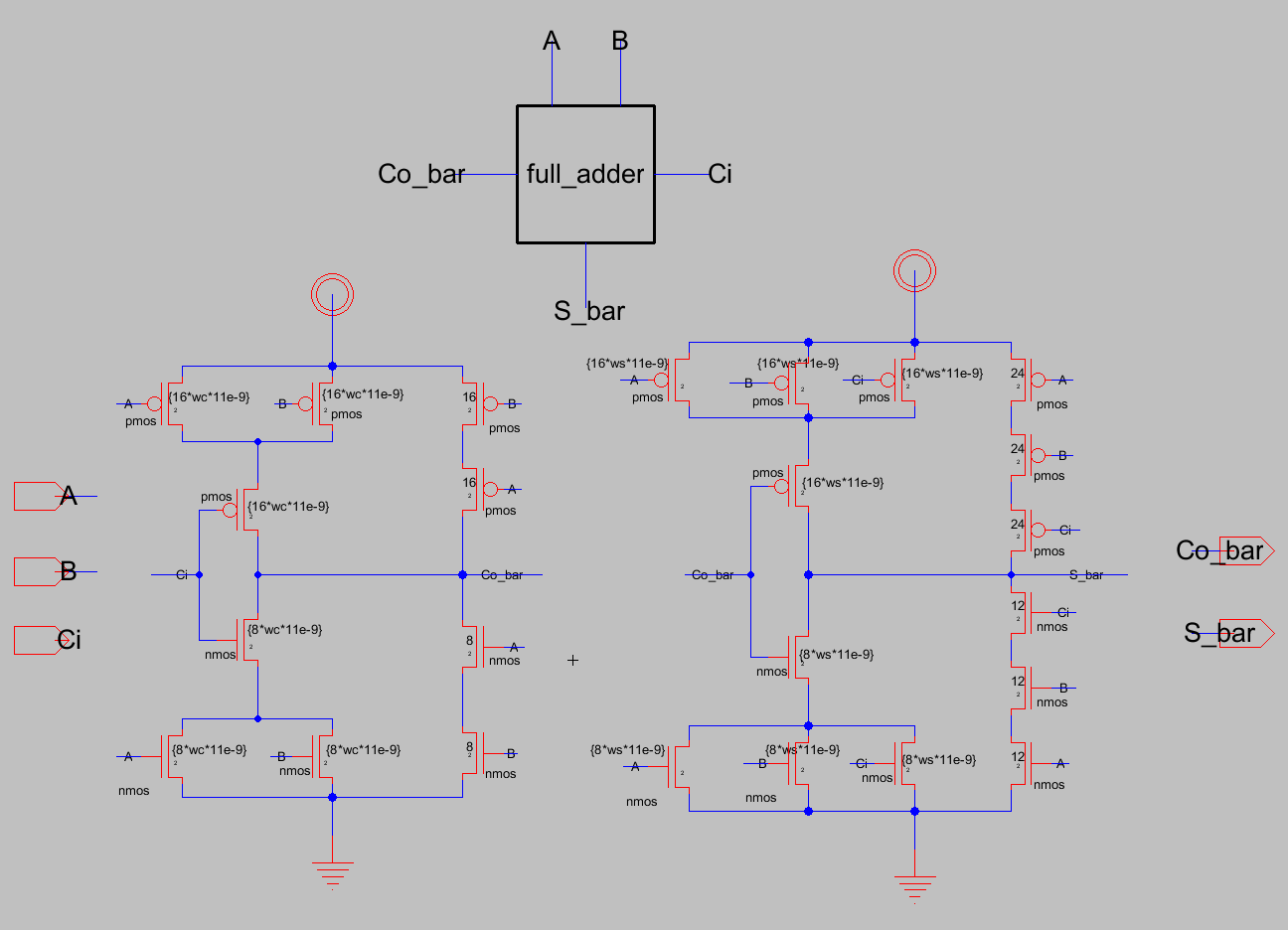
1. For an edge in Carry in, Full Adder should be in propagation stage for max delay. Delays for different combinations of A/B in Full Adder:

|  |  |  |
| --- | --- | --- |
| Inputs to (A, B) \ Cin Edge Type | Rising | Falling |
| 01 | 30.2ps | 9.37ps |
| 10 | 27.7ps | 9.27ps |

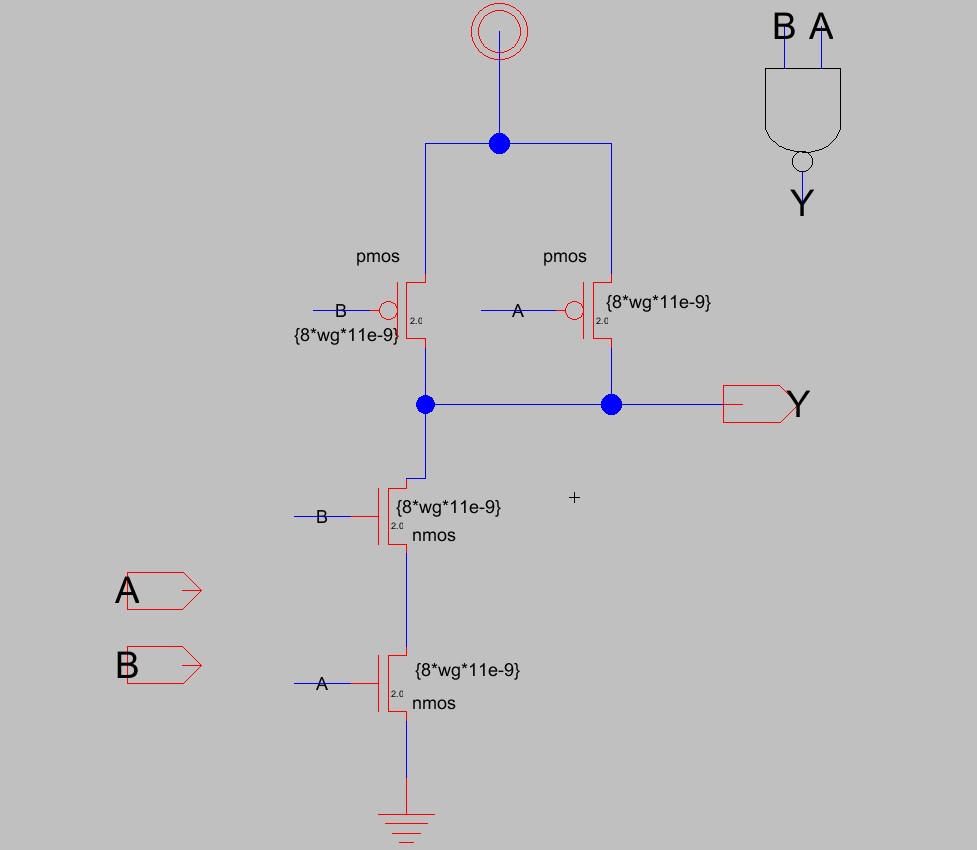
So, for rising/falling edge in Cin, A=0, B=1 for max delay in carry propagation

## Sizes of FA, NAND, AND and INV gates used are parameterised

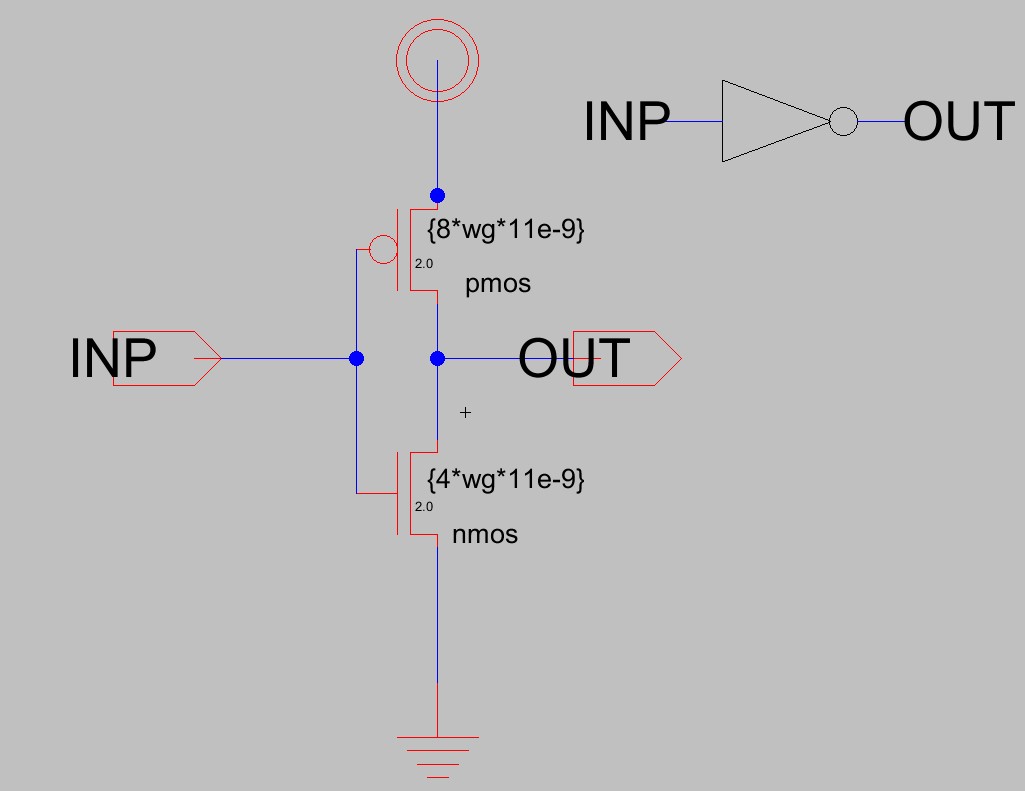
**Schematic of Full Adder used**



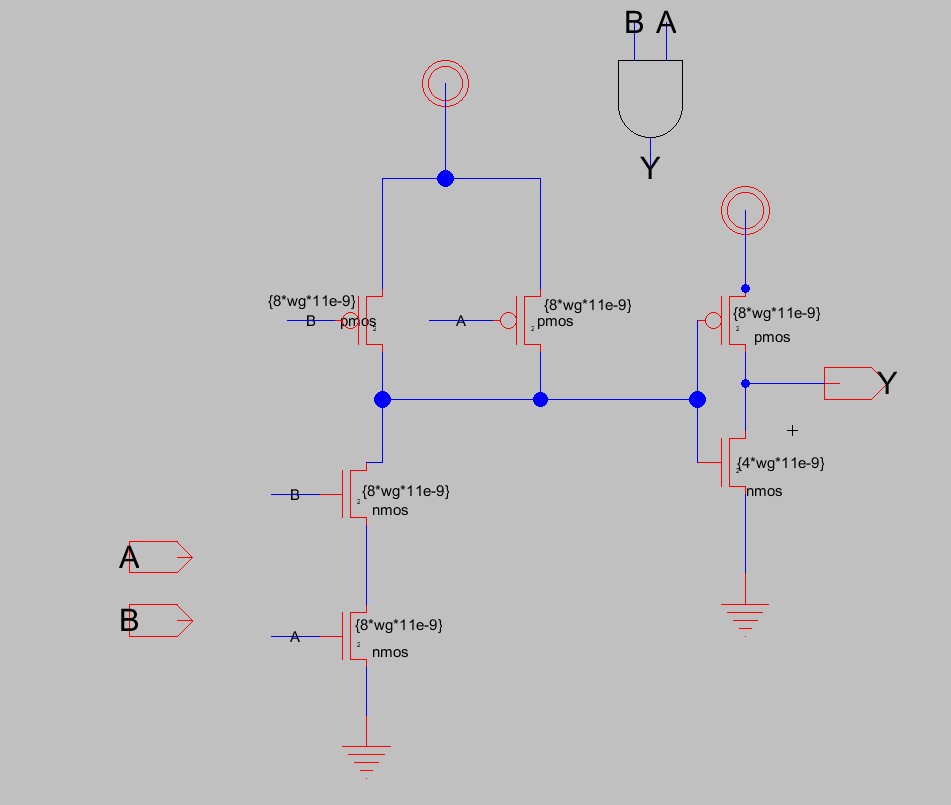
**Schematic of NAND gate used**



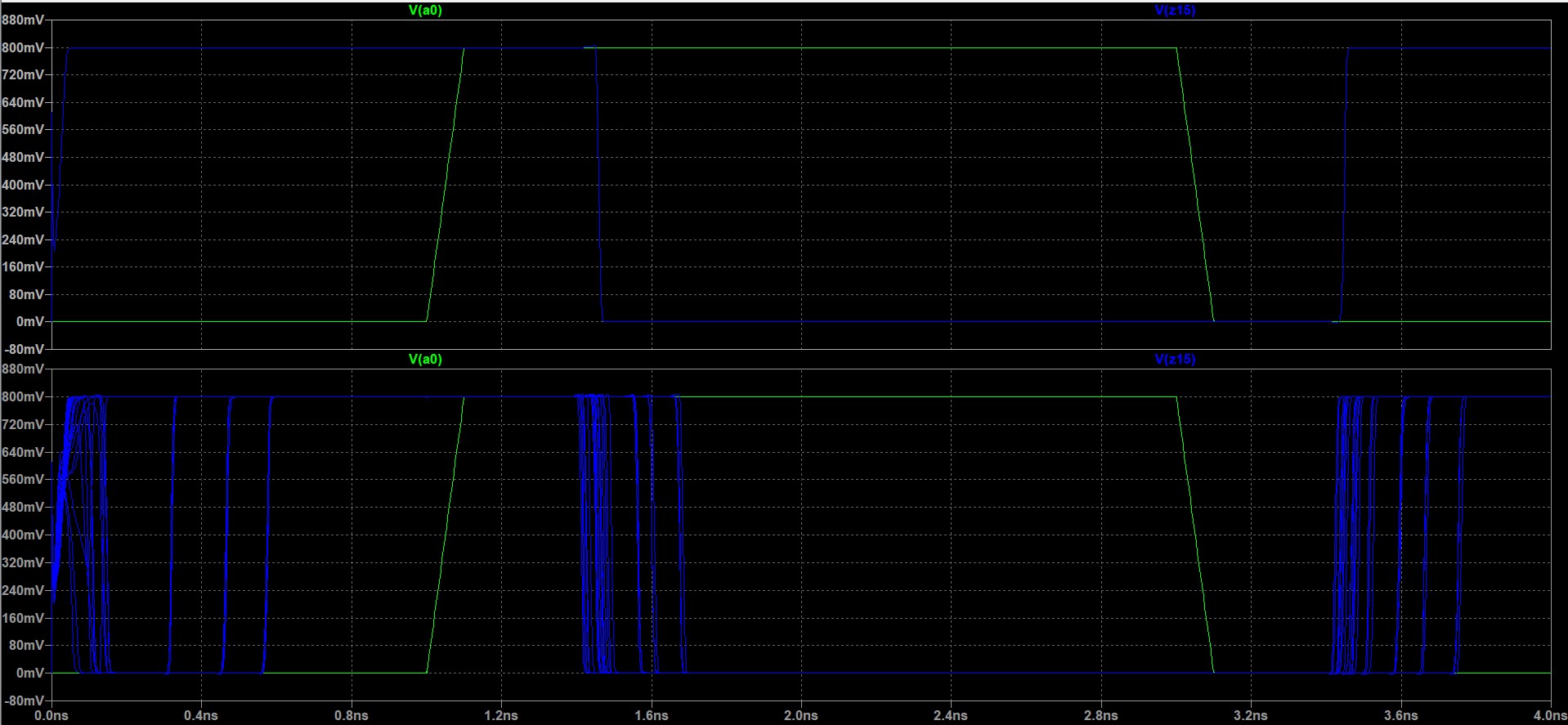
**Schematic of Inverter used**



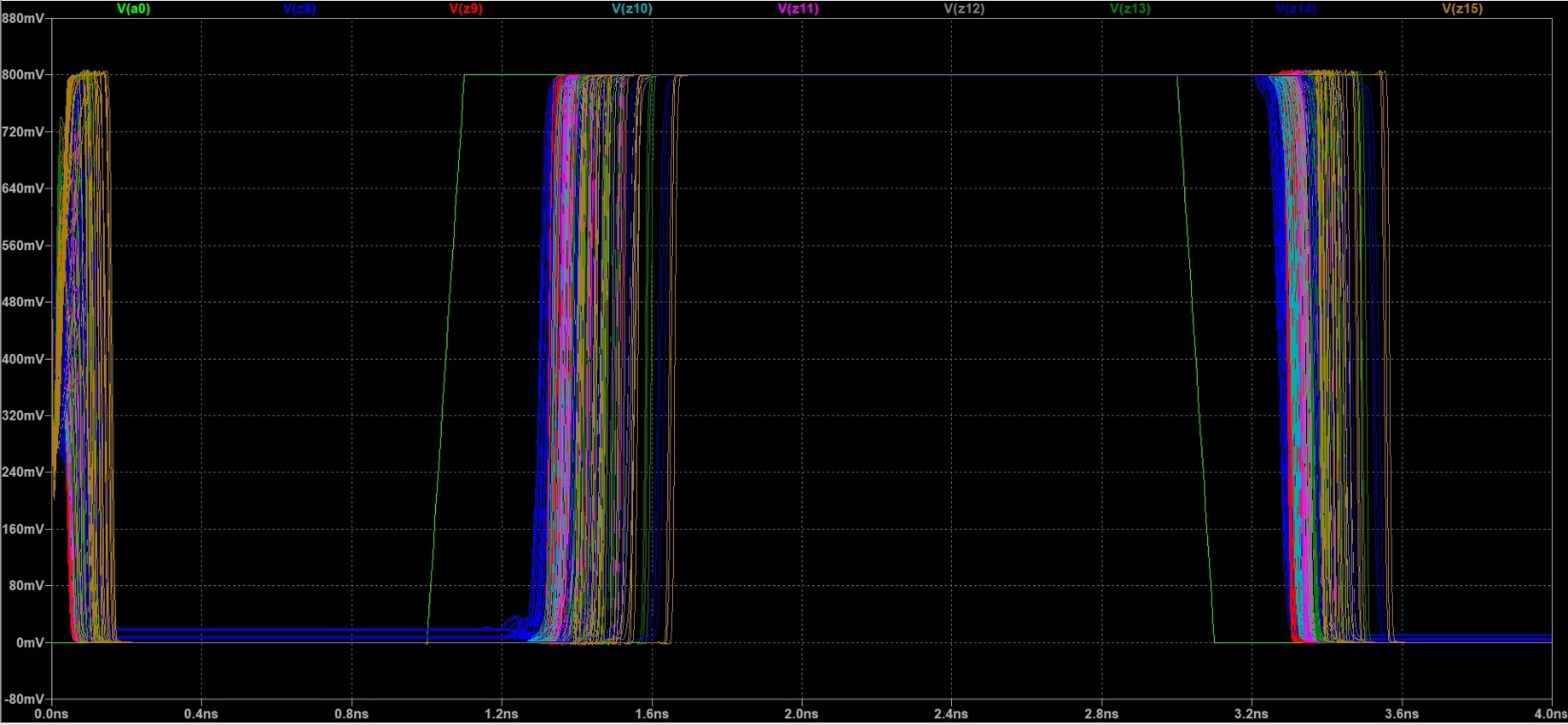
**Schematic of AND gate used**



**Simulation of Critical Path Delay swept across possible gate size combinations**



**Simulation of Vector Merge outputs for critical input**

****

**Critical Path Delay tabulated across gate size combinations**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **step** |  |  |  | **Crit-Path-Rise** | **Crit-Path-Fall** | **Max t\_pd** | **Area (unit\_sq)** |
| 1 | wc=1 | ws=1 | wg=1 | 5.15E-10 | 5.47E-10 | 5.47E-10 | 9049 |
| 2 | wc=2 | ws=1 | wg=1 | 4.23E-10 | 4.22E-10 | 4.23E-10 | 9751 |
| 3 | wc=3 | ws=1 | wg=1 | 4.10E-10 | 3.98E-10 | 4.10E-10 | 10679 |
| 4 | wc=1 | ws=2 | wg=1 | 5.58E-10 | 6.19E-10 | 6.19E-10 | 11315 |
| 5 | wc=2 | ws=2 | wg=1 | 4.14E-10 | 4.37E-10 | 4.37E-10 | 12017 |
| 6 | wc=3 | ws=2 | wg=1 | 3.77E-10 | 3.86E-10 | 3.86E-10 | 12945 |
| 7 | wc=1 | ws=3 | wg=1 | 6.34E-10 | 7.11E-10 | 7.11E-10 | 13899 |
| 8 | wc=2 | ws=3 | wg=1 | 4.44E-10 | 4.76E-10 | 4.76E-10 | 14601 |
| 9 | wc=3 | ws=3 | wg=1 | 3.87E-10 | 4.06E-10 | 4.06E-10 | 15529 |
| 10 | wc=1 | ws=1 | wg=2 | 5.12E-10 | 5.44E-10 | 5.44E-10 | 9049 |
| 11 | wc=2 | ws=1 | wg=2 | 4.16E-10 | 4.17E-10 | 4.17E-10 | 9751 |
| 12 | wc=3 | ws=1 | wg=2 | 4.01E-10 | 3.91E-10 | 4.01E-10 | 10679 |
| 13 | wc=1 | ws=2 | wg=2 | 5.51E-10 | 6.13E-10 | 6.13E-10 | 11315 |
| 14 | wc=2 | ws=2 | wg=2 | 4.05E-10 | 4.29E-10 | 4.29E-10 | 12017 |
| 15 | wc=3 | ws=2 | wg=2 | 3.66E-10 | 3.77E-10 | 3.77E-10 | 12945 |
| 16 | wc=1 | ws=3 | wg=2 | 6.25E-10 | 7.03E-10 | 7.03E-10 | 13899 |
| 17 | wc=2 | ws=3 | wg=2 | 4.33E-10 | 4.66E-10 | 4.66E-10 | 14601 |
| 18 | wc=3 | ws=3 | wg=2 | 3.75E-10 | 3.95E-10 | 3.95E-10 | 15529 |
| 19 | wc=1 | ws=1 | wg=4 | 5.15E-10 | 5.48E-10 | 5.48E-10 | 9049 |
| 20 | wc=2 | ws=1 | wg=4 | 4.16E-10 | 4.17E-10 | 4.17E-10 | 9751 |
| 21 | wc=3 | ws=1 | wg=4 | 4.00E-10 | 3.90E-10 | 4.00E-10 | 10679 |
| 22 | wc=1 | ws=2 | wg=4 | 5.52E-10 | 6.13E-10 | 6.13E-10 | 11315 |
| 23 | wc=2 | ws=2 | wg=4 | 4.03E-10 | 4.27E-10 | 4.27E-10 | 12017 |
| 24 | wc=3 | ws=2 | wg=4 | 3.62E-10 | 3.73E-10 | 3.73E-10 | 12945 |
| 25 | wc=1 | ws=3 | wg=4 | 6.24E-10 | 7.02E-10 | 7.02E-10 | 13899 |
| 26 | wc=2 | ws=3 | wg=4 | 4.29E-10 | 4.63E-10 | 4.63E-10 | 14601 |
| 27 | wc=3 | ws=3 | wg=4 | 3.69E-10 | 3.90E-10 | 3.90E-10 | 15529 |

For Cout=3x, Sum=2x, Nand=4x min delay is observed.

For Cout=3x, Sum=2x, Nand=1x, only a 3.5% delay improvement is observed but a much smaller nand gate is used.

But combination Cout=3x, Sum=1x, Nand-1x has only a 6% higher delay and 18% lesser area. So, chosen sizes are 3x for carry out block, 1x for sum block, 1x for the standard gates.

**Chosen Gate sizes**

**Full Adder:**

**Cout-bar: 3x**

**Sum-bar: 1x**

**Standard Cells:**

**NAND, AND, INV: 1x**

**--End--**