ECE 261 Project Presentation 2

8-bit Booth Multiplier

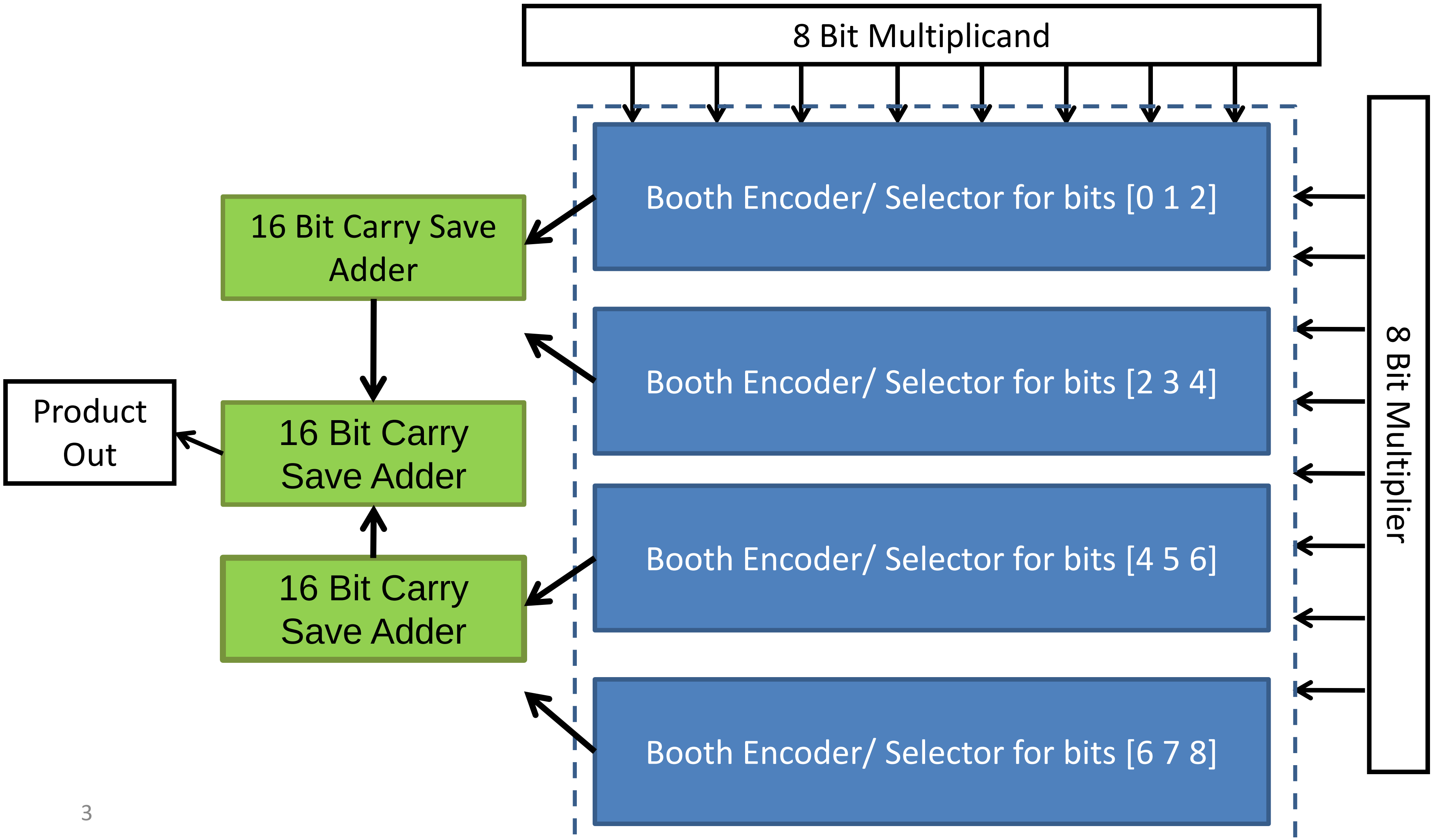
Eric Wang
Federico Gonzalez
Bryan Flemming
Jep Barbour

Abstract

The purpose of this project is to create a 8 by 8 multiplier using Booth's multiplication algorithm. The 8-bit multiplicand and 8-bit multiplier are input signals into four Booth encoders/selectors. After applying Booth's algorithm to the inputs, simple addition is done to produce a final output.

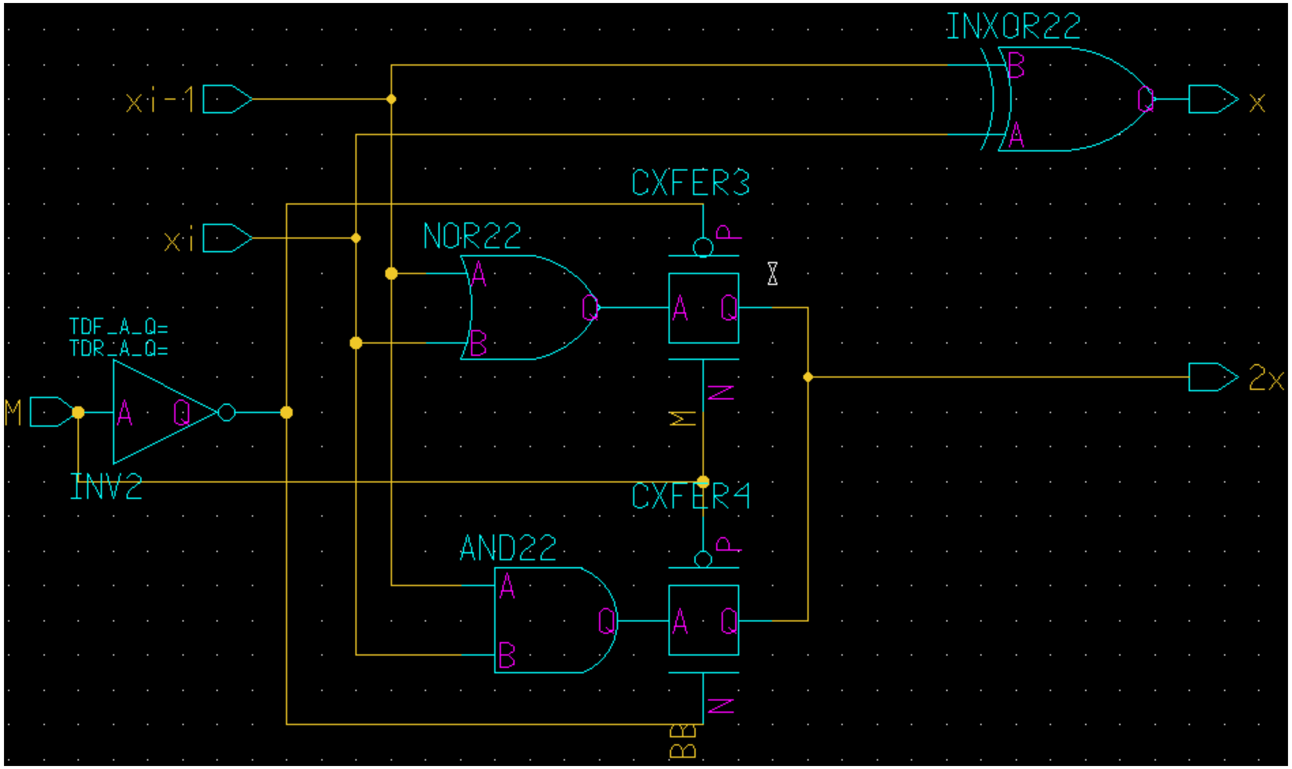
Our main goal is to produce a working 8 by 8 bit multiplier with correct simulations and layout while attempting to maximize the speed in which the multiplier performs the calculation.

Schematic Block Diagram

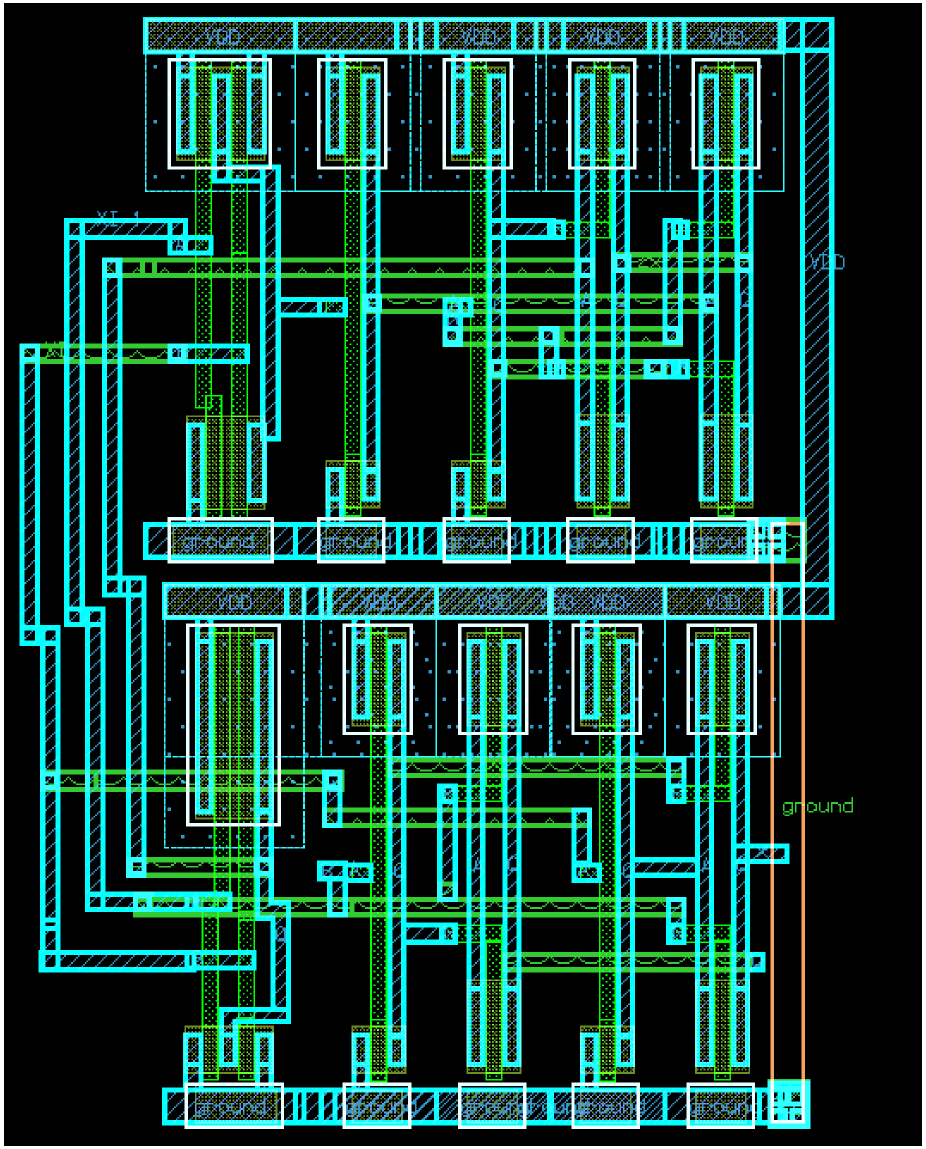


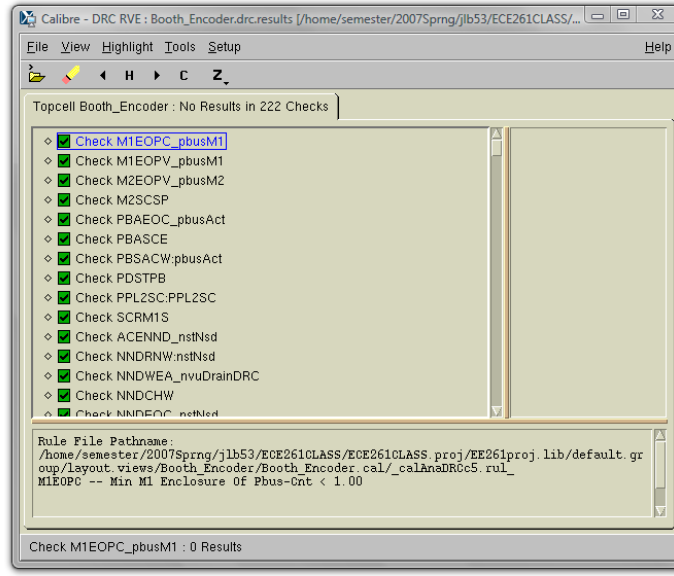
Booth Encoder

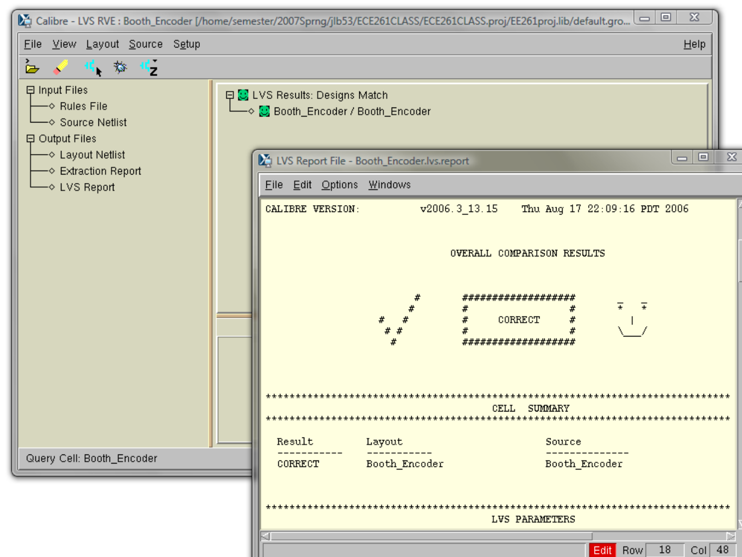
Schematic



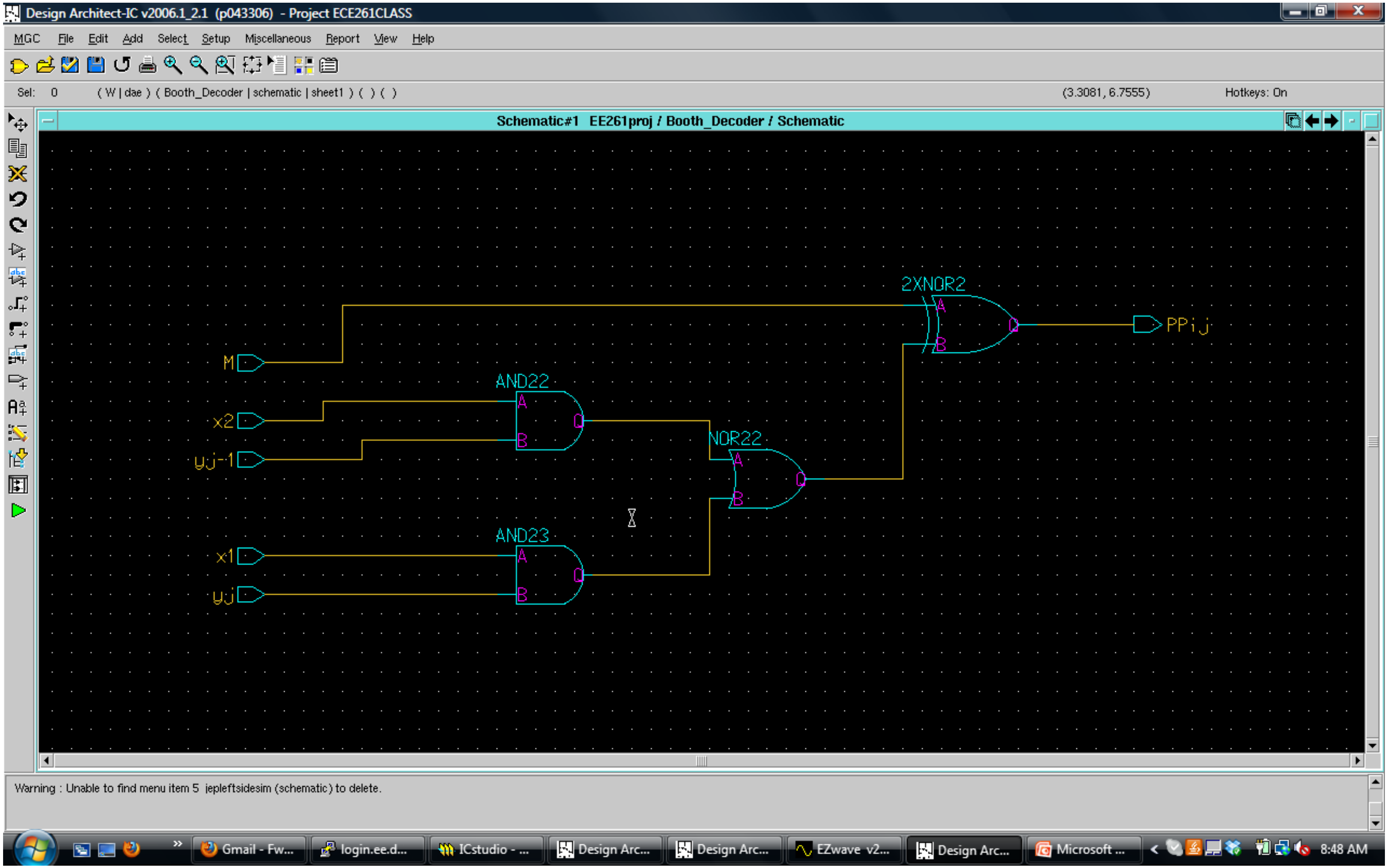
Booth Encoder



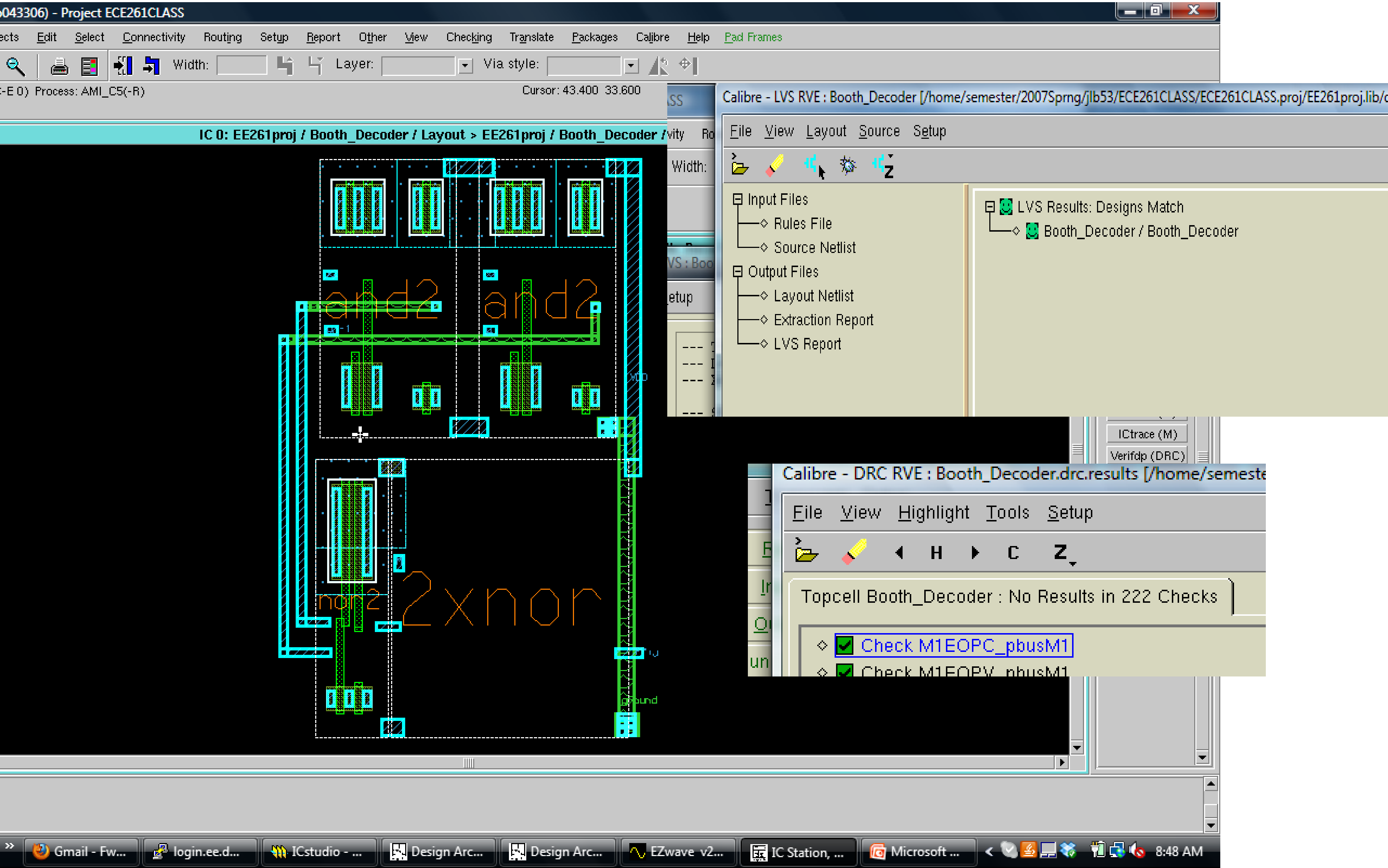




Booth Decoder Schematic

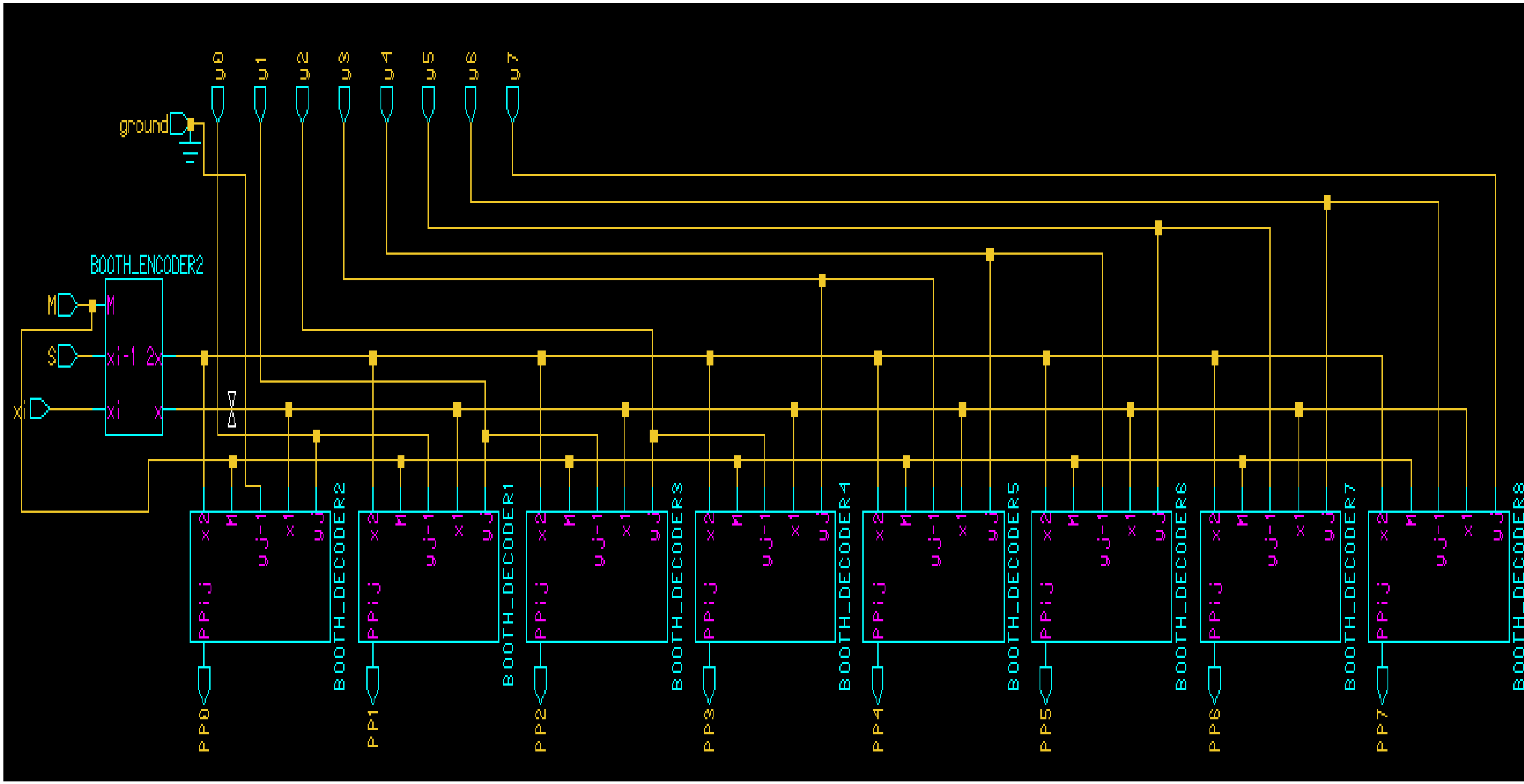


Booth Decoder Layout

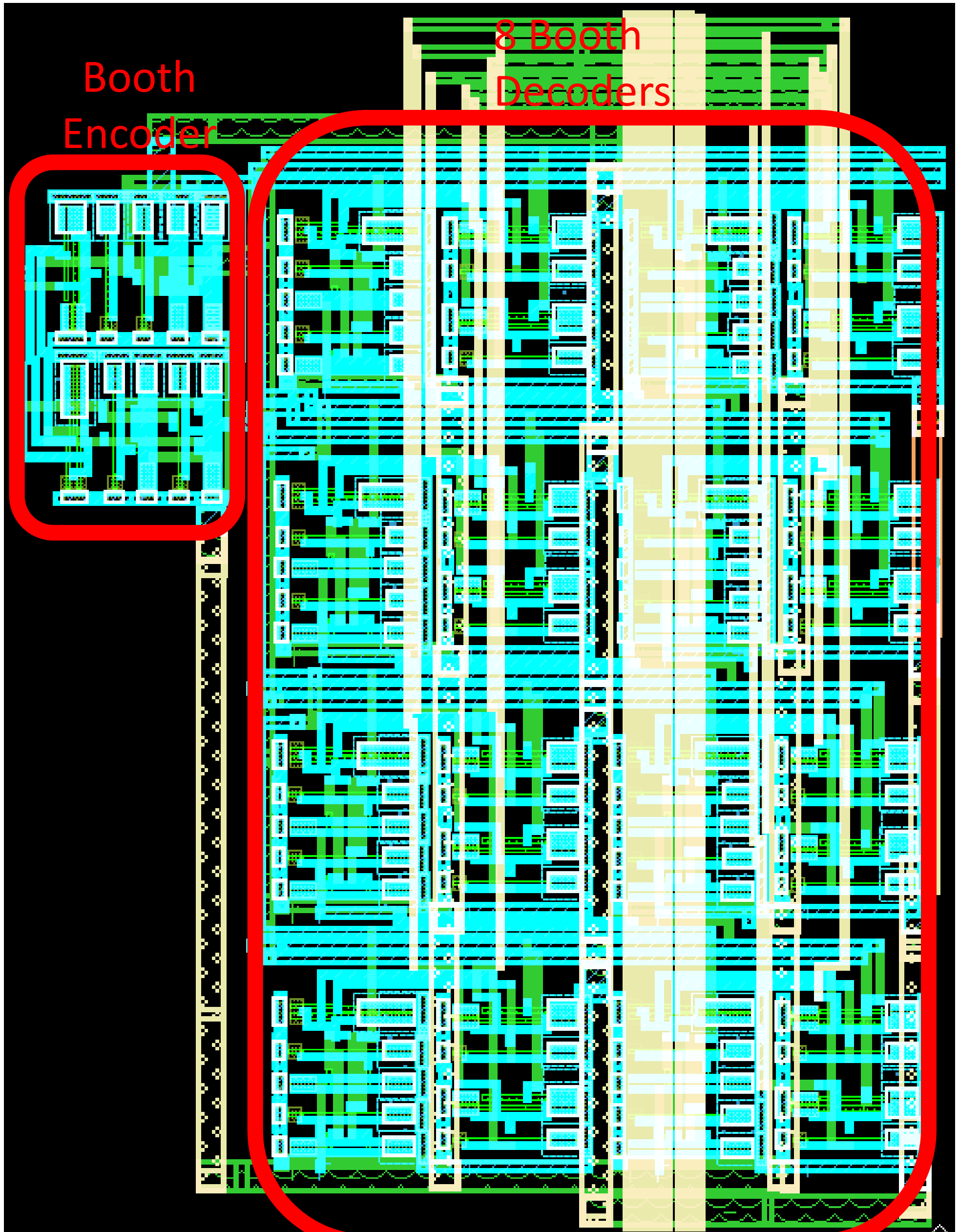


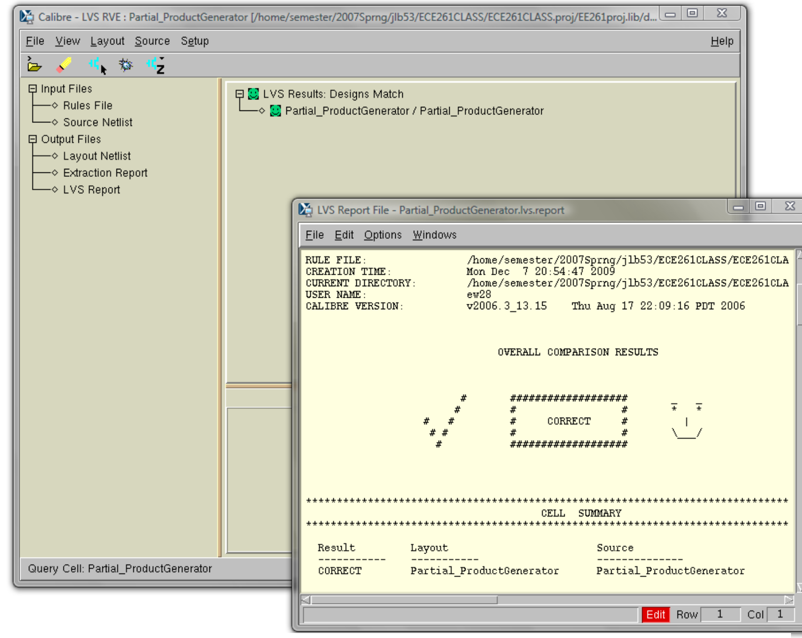
Partial product generator

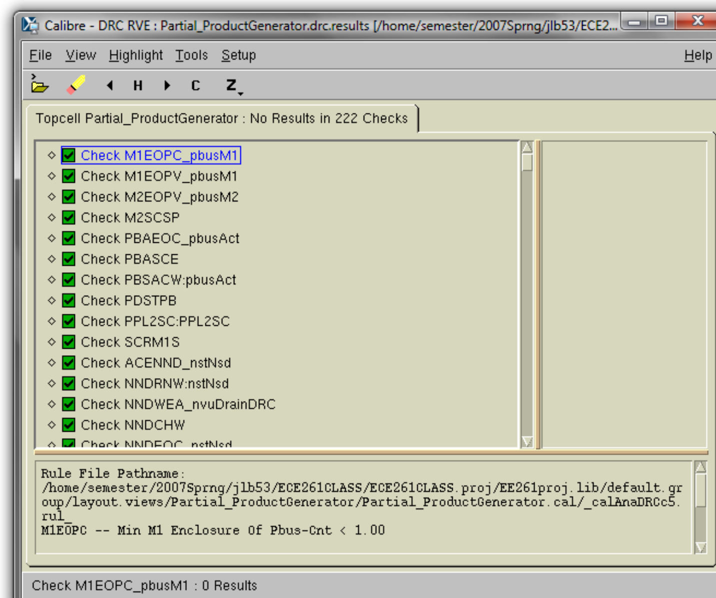
Schematic



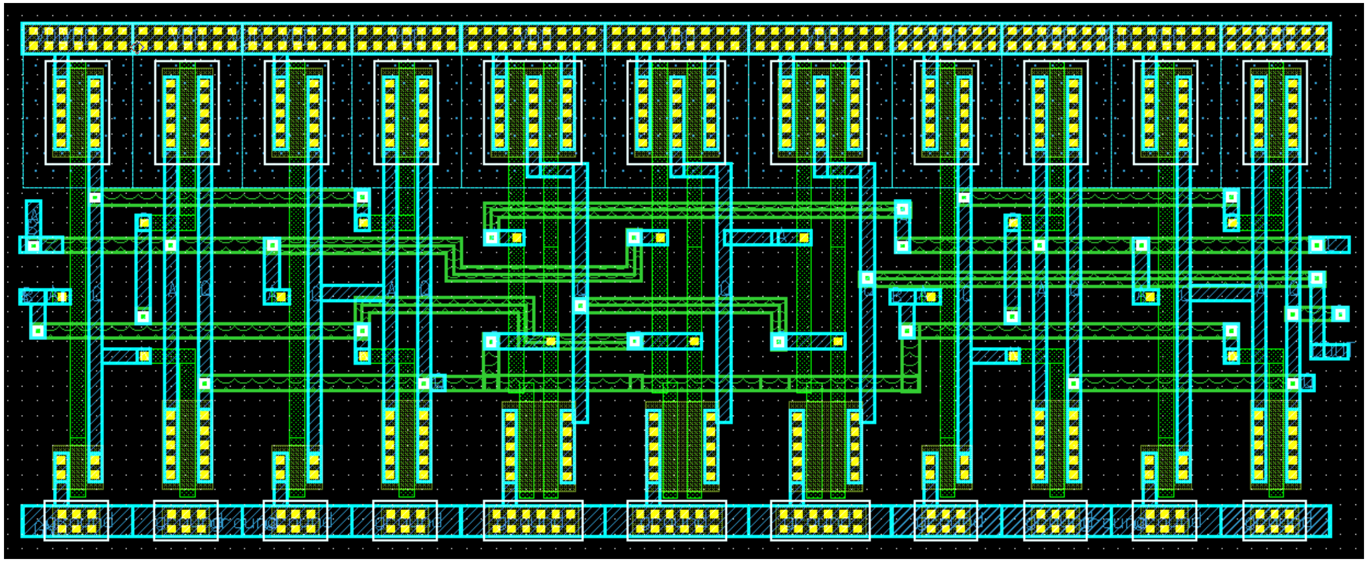
Partial product generator

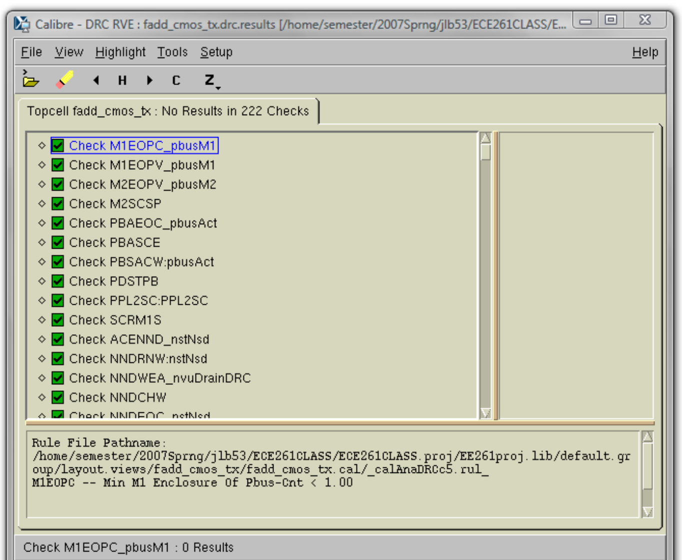


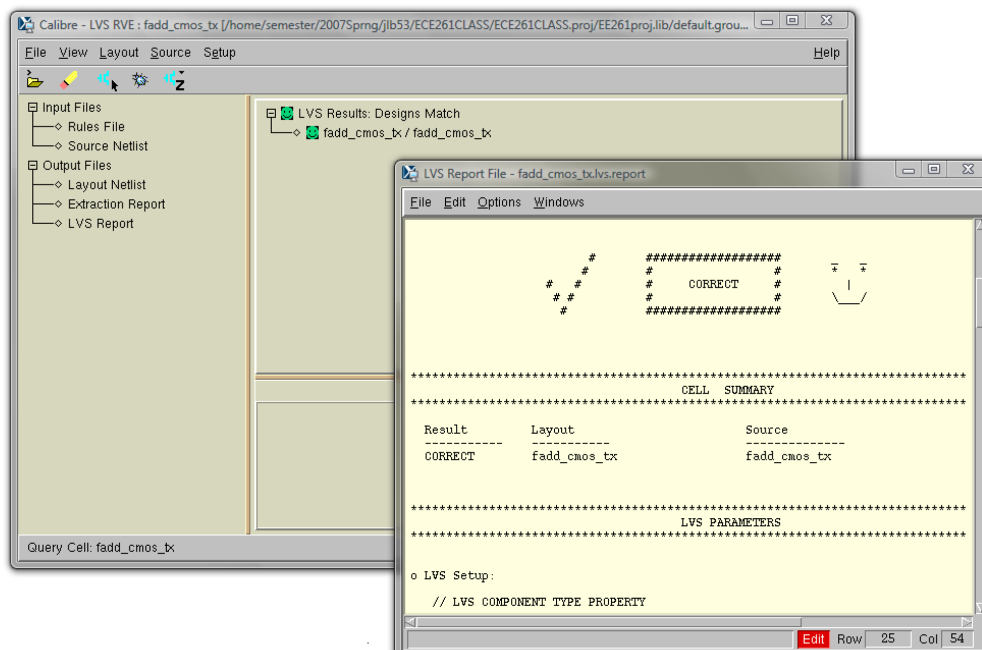




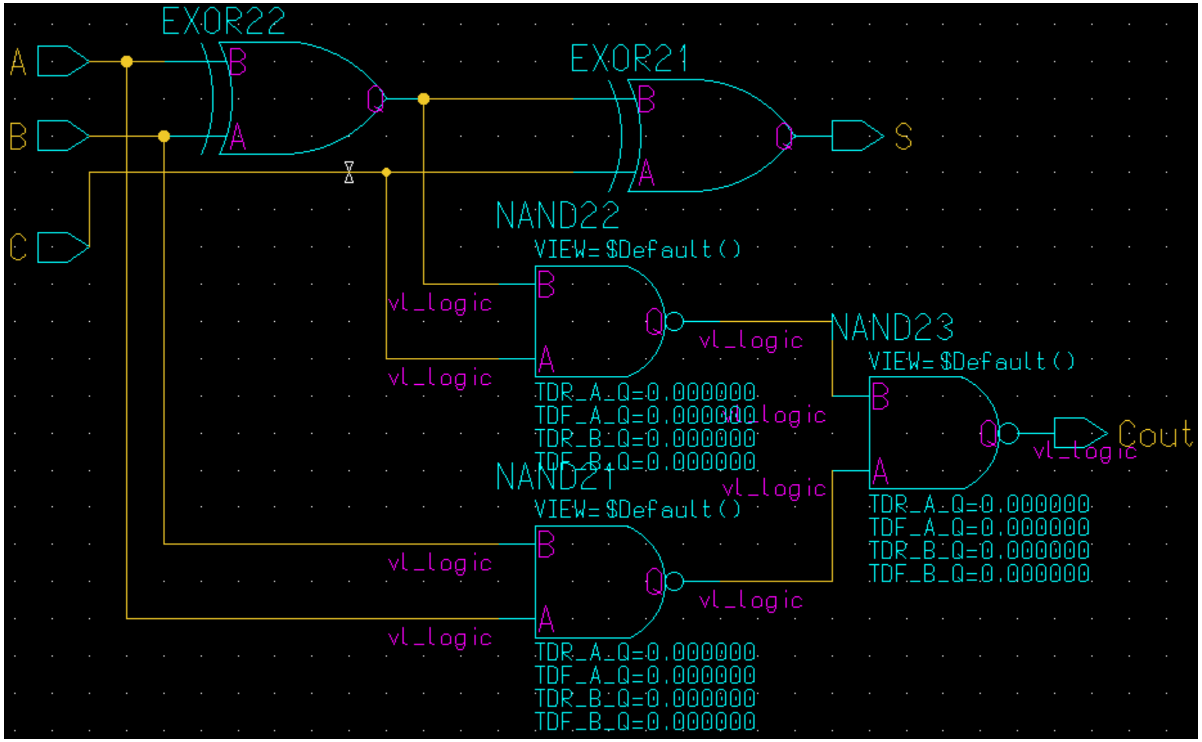
Full Adder



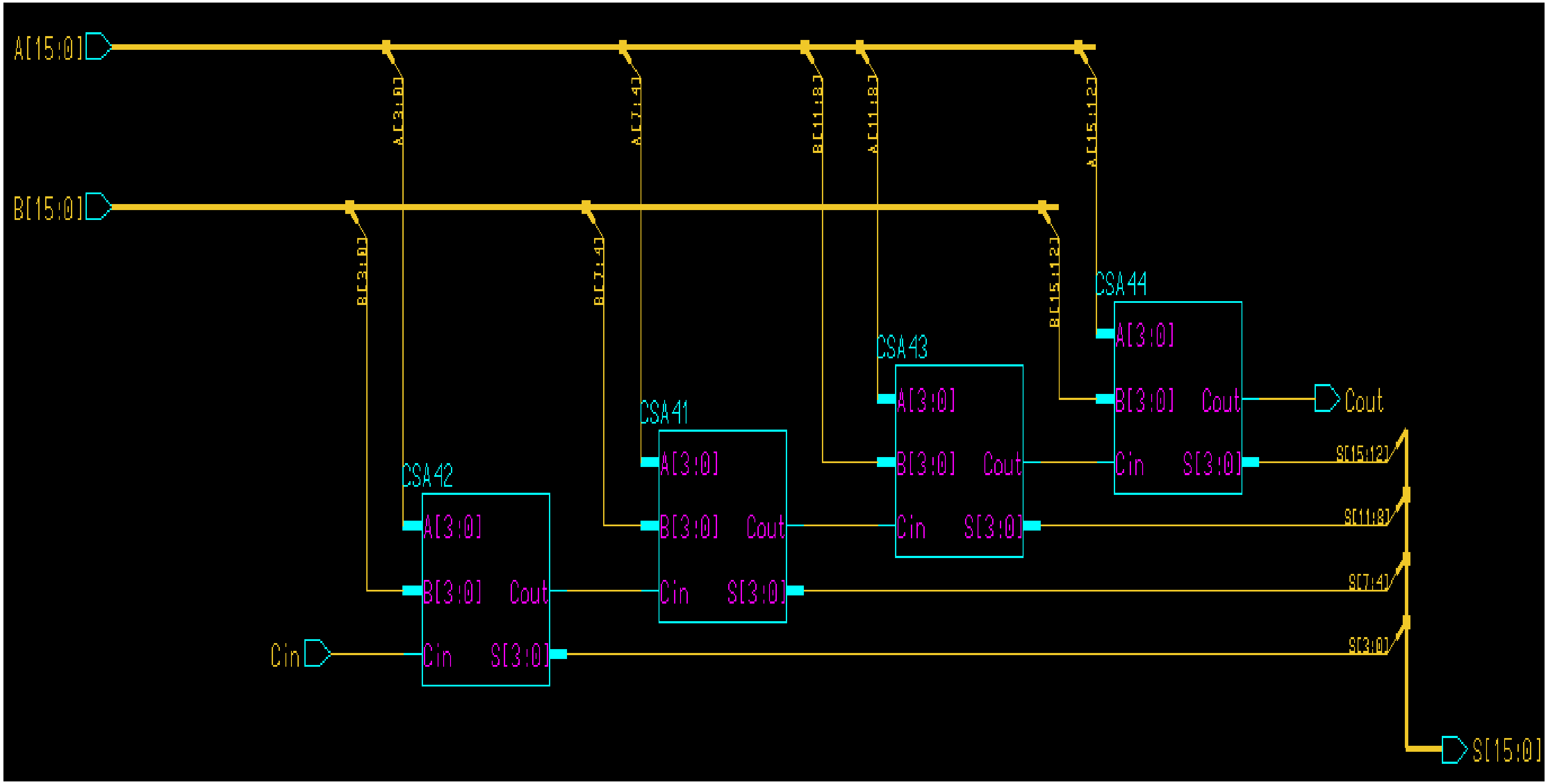

Calibre - DRC RVE : fadd_cmos_tx.drc.results [/home/semester/2007Sprng/jlb53/ECE261CLASS/E...
File View Highlight Tools Setup Help
Topcell fadd_cmos_tx : No Results in 222 Checks
Check M1EOPC_pbusM1
Check M1EOPV_pbusM1
Check M2EOPV_pbusM2
Check M2SCSP
Check PBAEOC_pbusAct
Check PBASCE
Check PBSACW:pbusAct
Check PDSTPB
Check PPL2SC:PPL2SC
Check SCRM1S
Check ACENND_nstNsd
Check NNDRNW:nstNsd
Check NNDWEA_nvuDrainDRC
Check NNDCHW
Rule File Pathname:
/home/semester/2007Sprng/jlb53/ECE261CLASS/ECE261CLASS.proj/EE261proj.lib/default.gr
oup/layout.views/fadd_cmos_tx/fadd_cmos_tx.cal/_calAnaDRCc5.rul_
M1EOPC -- Min M1 Enclosure Of Pbus-Cnt < 1.00
Check M1EOPC_pbusM1 : 0 Results


Calibre - LVS RVE : fadd_cmos_tx [/home/semester/2007Sprng/jlb53/ECE261CLASS/ECE261CLASS.proj/EE261proj.lib/default.grou...
File View Layout Source Setup Help
Input Files
Rules File
Source Netlist
Output Files
Layout Netlist
Extraction Report
LVS Report
LVS Results: Designs Match
fadd_cmos_tx / fadd_cmos_tx
Query Cell: fadd_cmos_tx
LVS Report File - fadd_cmos_tx.lvs.report
File Edit Options Windows
CORRECT
CELL SUMMARY
Result Layout Source
CORRECT fadd_cmos_tx fadd_cmos_tx
LVS PARAMETERS
o LVS Setup:
// LVS COMPONENT TYPE PROPERTY
Edit Row 25 Col 54

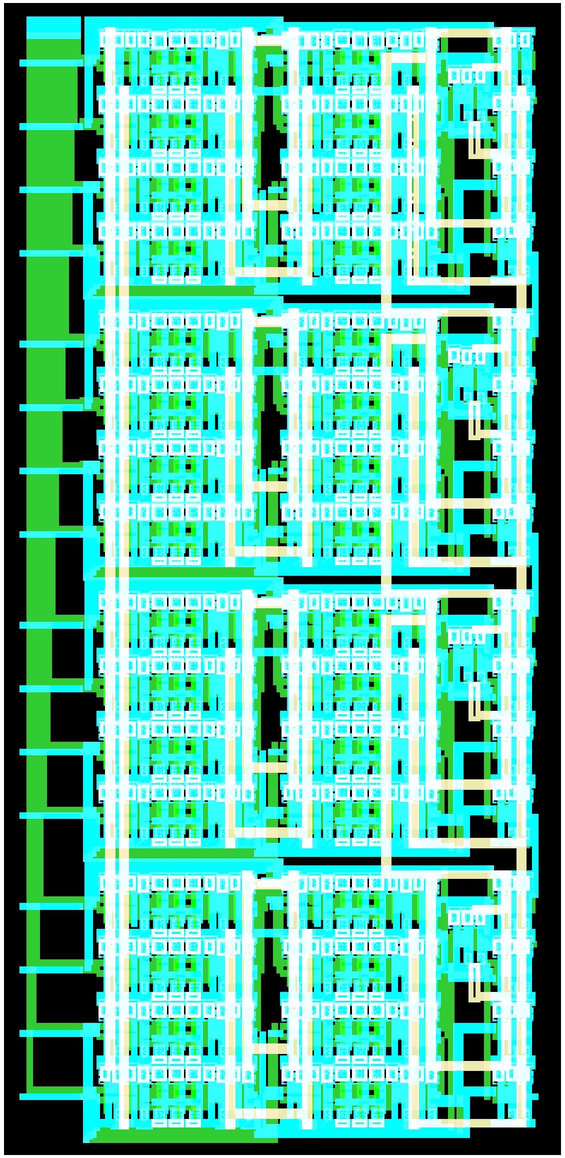
Full Adder

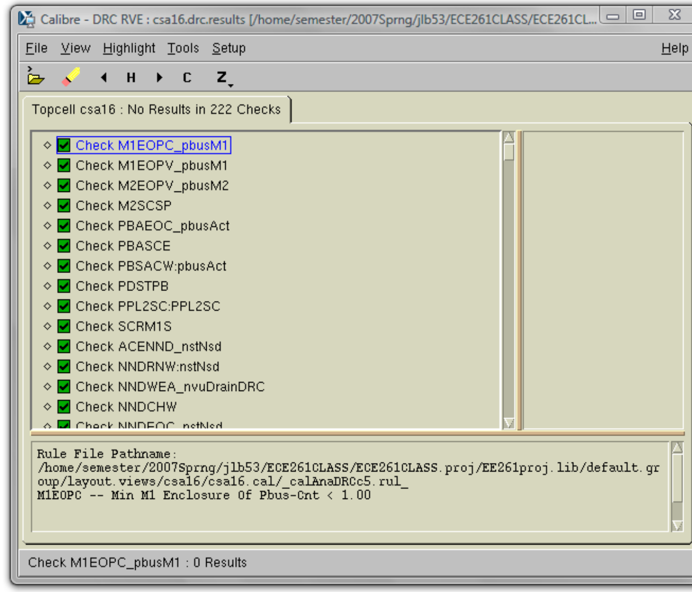

EXOR22
EXOR21
A
B
C
S
Cout
NAND22
VIEW=$Default()
NAND21
VIEW=$Default()
NAND23
VIEW=$Default()
vl_logic
TDR_A_Q=0.000000
TDF_A_Q=0.000000
TDR_B_Q=0.000000
TDF_B_Q=0.000000

16 bit Carry Save Adder

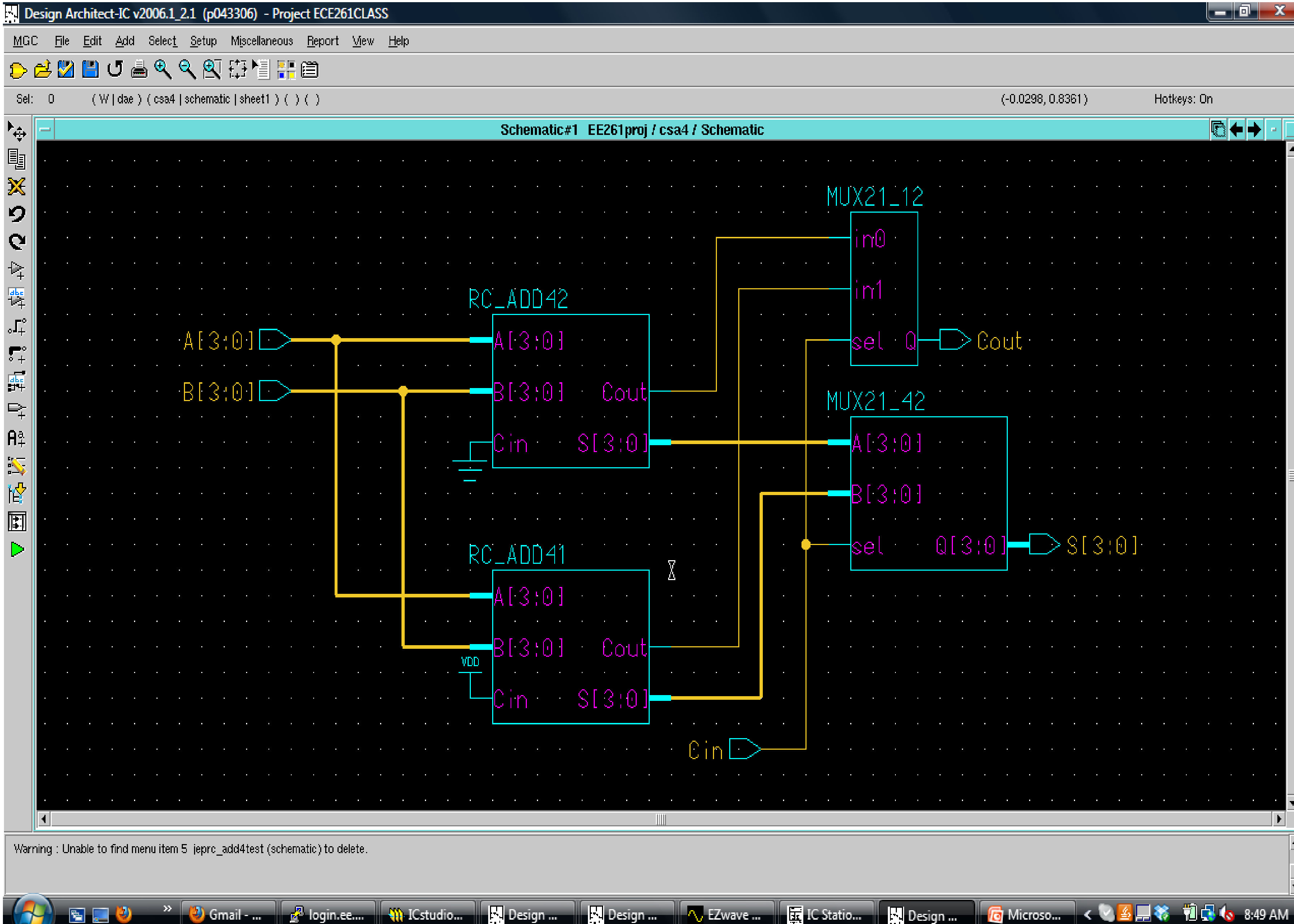


16 bit Carry Save Adder

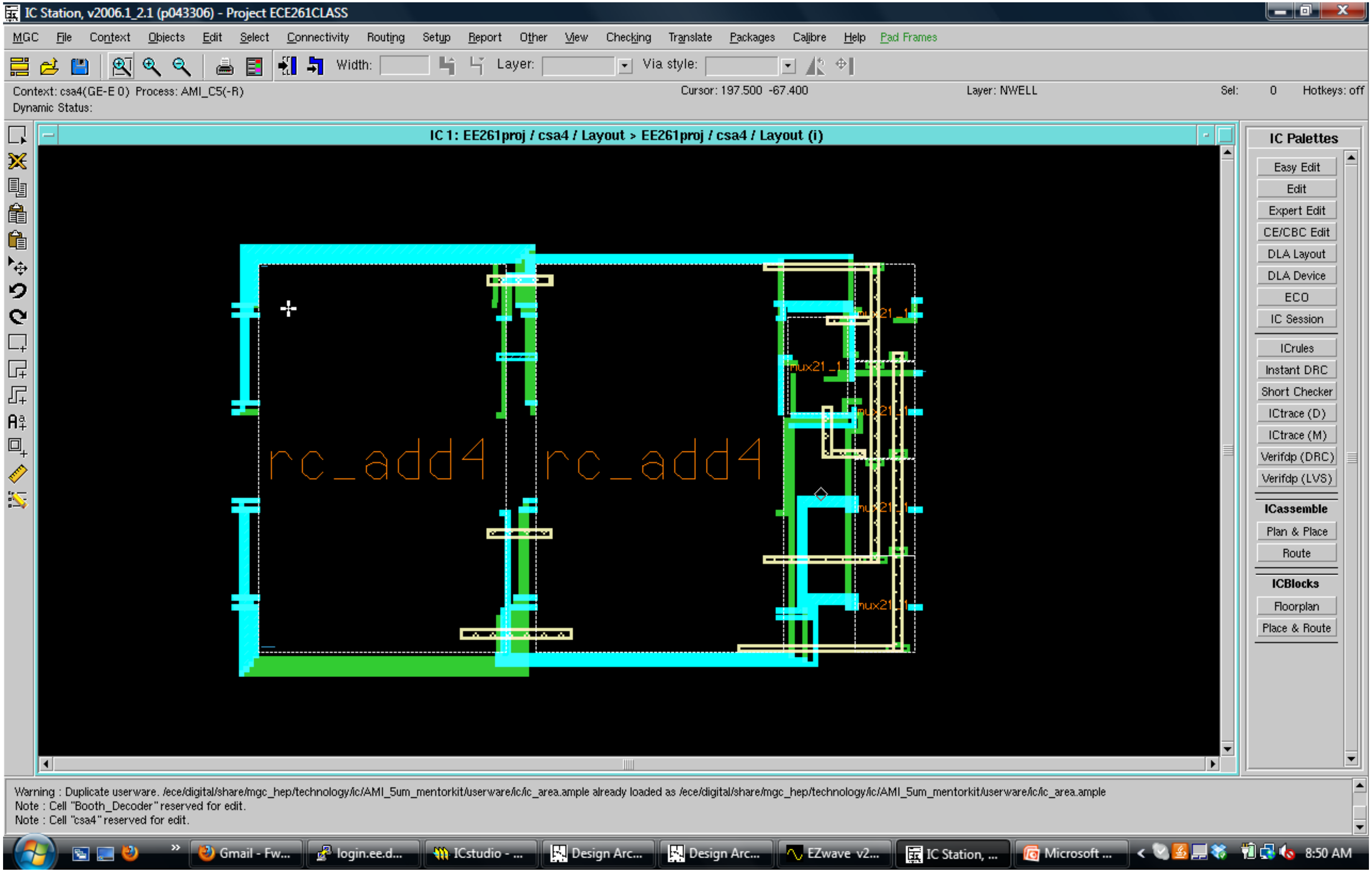




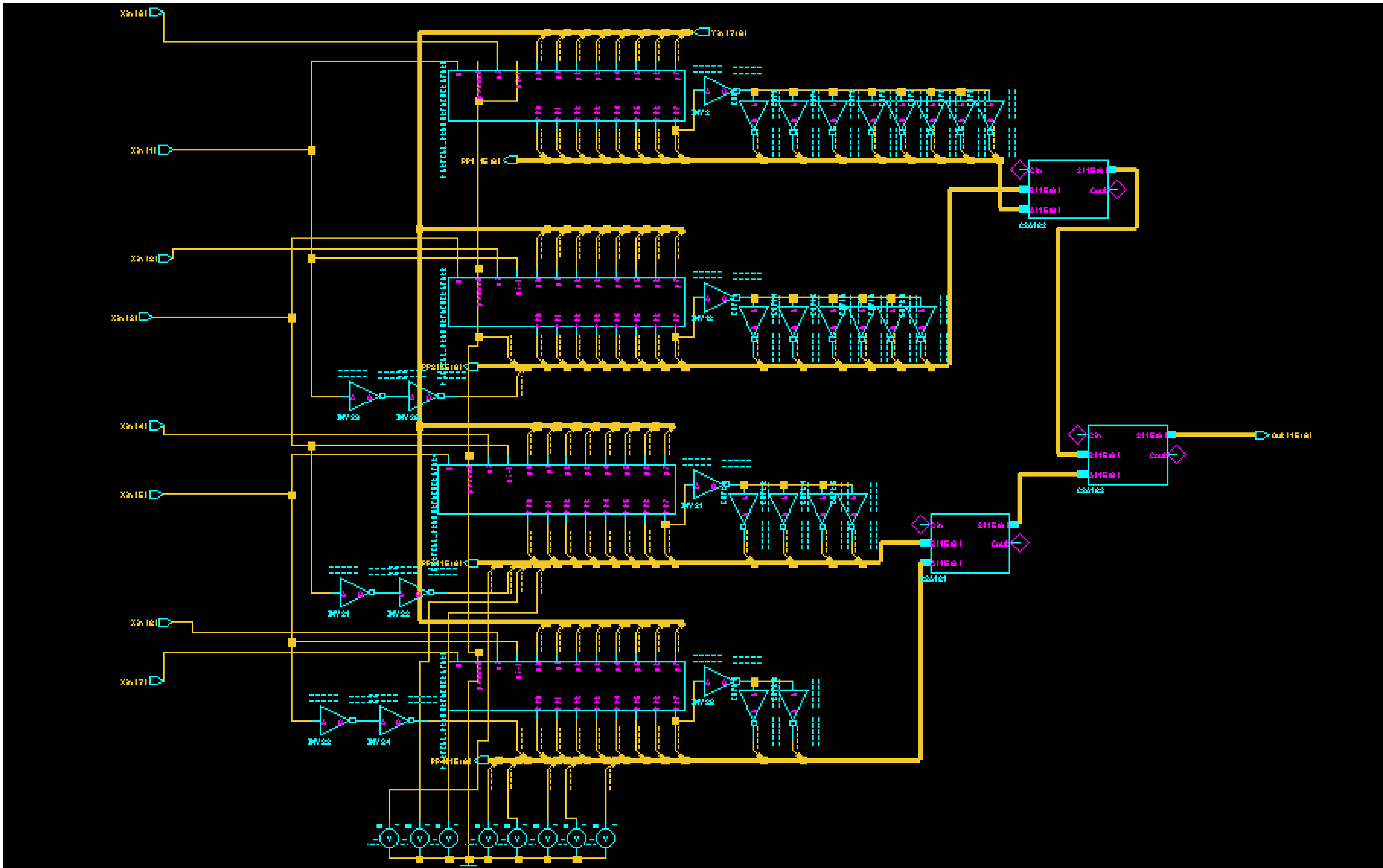
4 bit Carry Save Adder



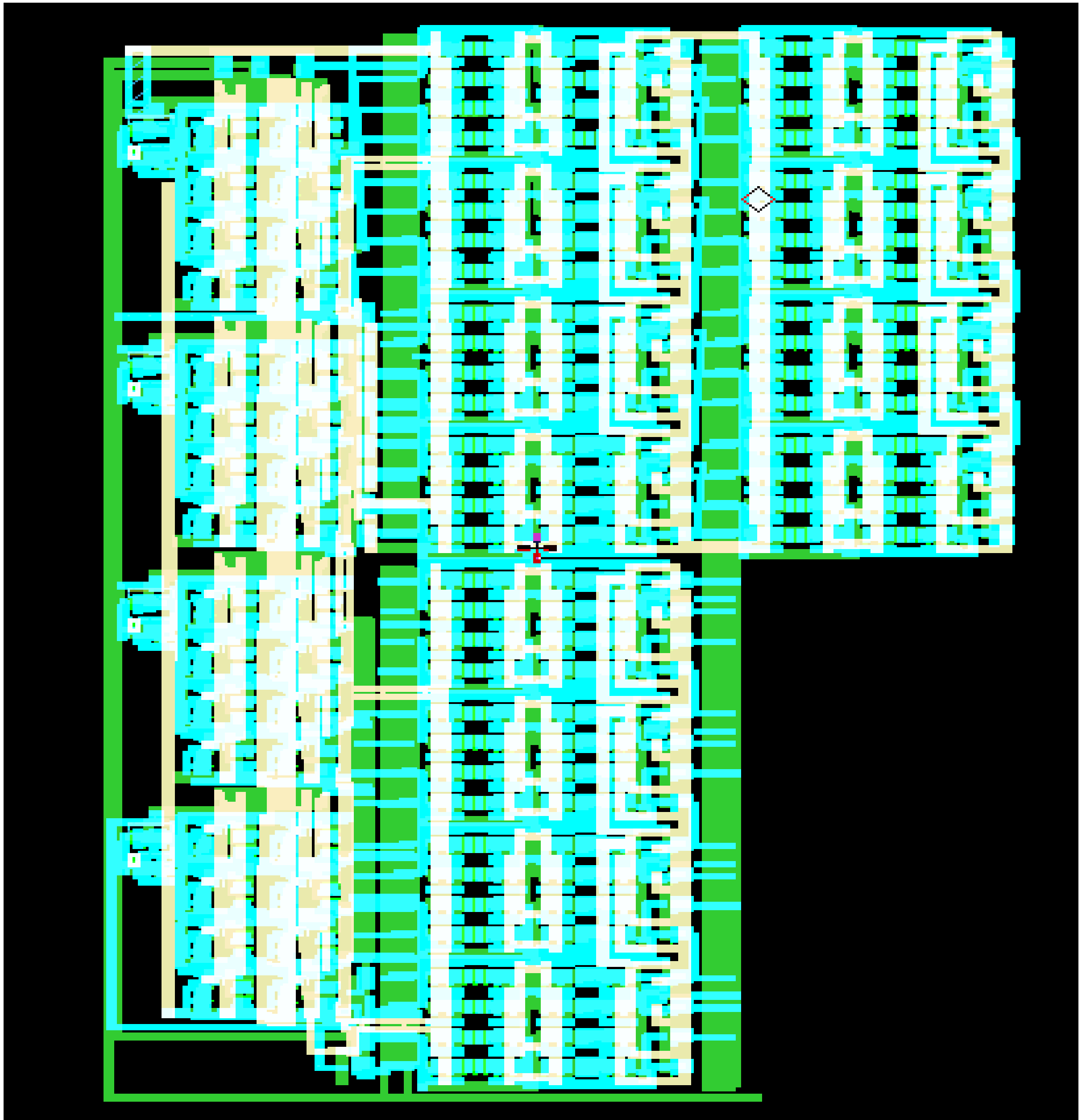
4 bit Carry Save Adder



Top Level Diagram



Top Level Layout



Functional Simulations

- Our Eldo would not simulate the top level diagram with all three adders included
- Therefore, for simulations, we deleted the third adder, and took the outputs at the ends of the inner adders.
- To verify functionality through simulation, we simply took the two outputs we found and added them ourselves.

Test 1:

- Multiplier input: 00001010 = 10
- Multiplicand input: 00000101 = 5

- Output 1: 1111111111011110
- Output 2: 0000000001010100
- Sum of outputs: 0000000000110010 = 50

- As we can see, the output equals the product of the two inputs

Test 2

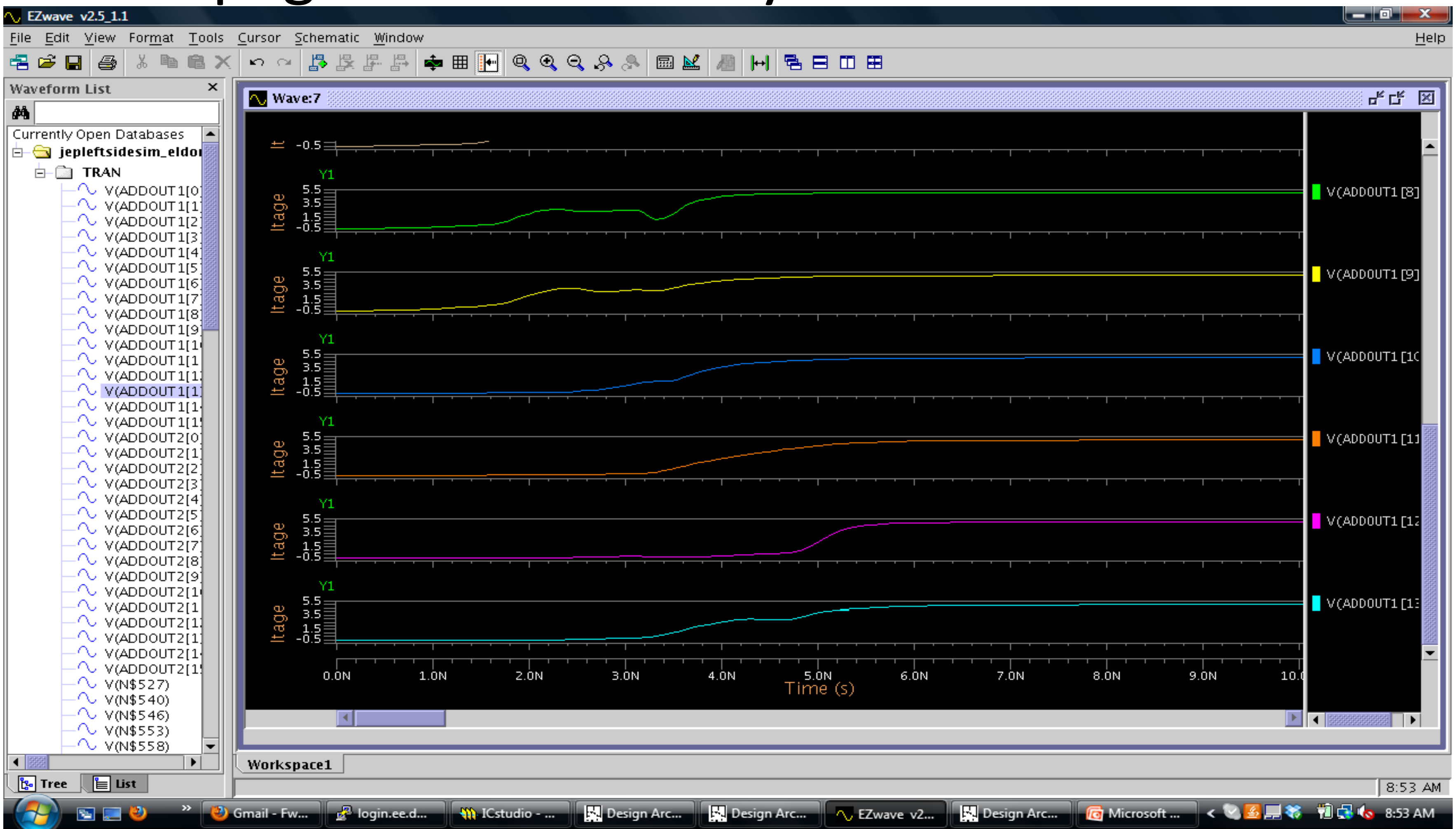
- Multiplier input: 01011101 = 93
- Multiplicand input: 0001101010 = 26

- Output 1: 1111111110100110
- Output 2: 0000100111000100
- Sum of outputs: 0000100101101010 = 2418

- Once again the final output is equal to the product of the inputs as expected.

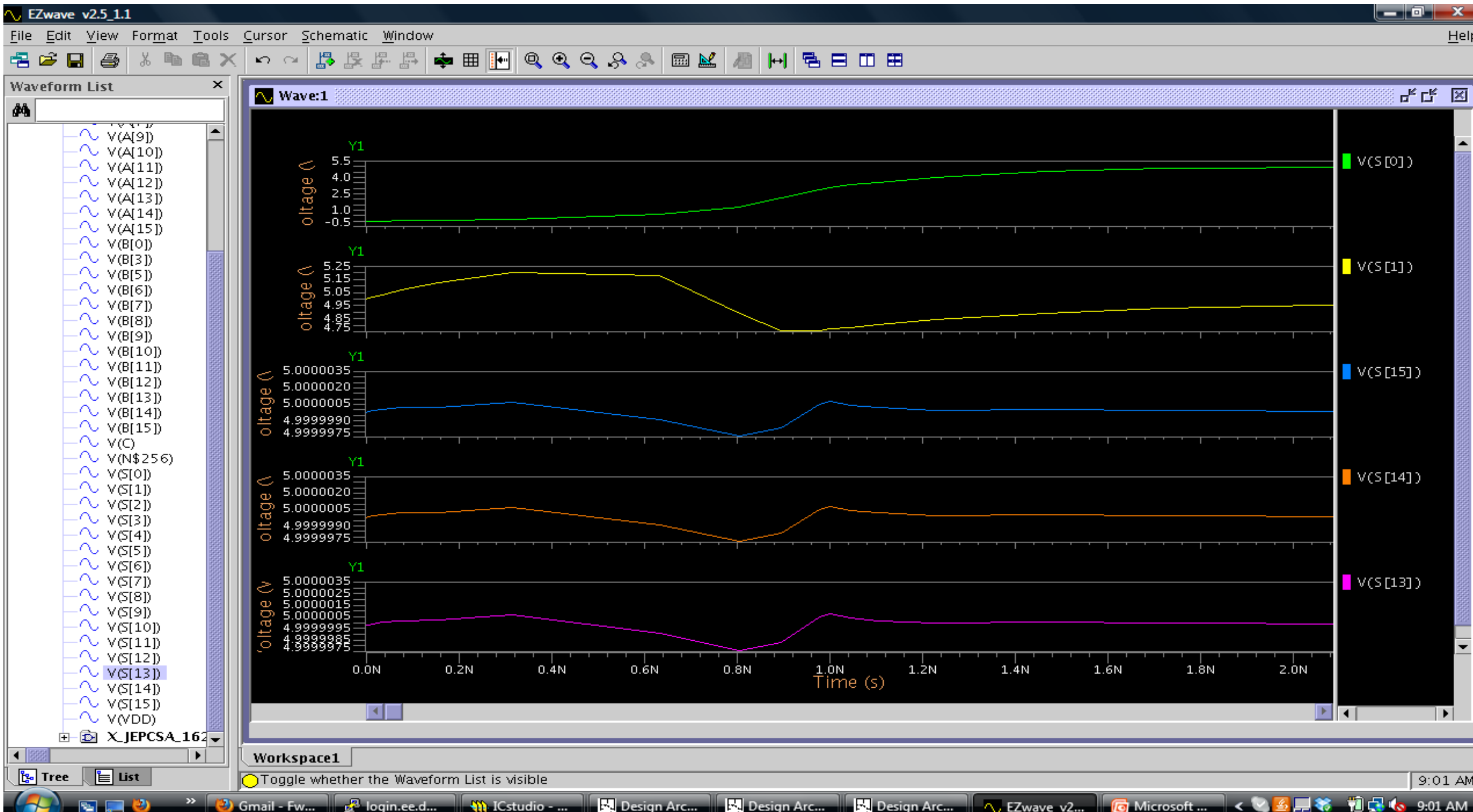
Timing Simulations

- Propagation time? Only 6ns worst case



Propagation Time continued

- Extra adder! 2ns additional propagation time.



Power Consumption

- Simulation of top level block (minus the final adder of course) showed an average power consumption of 73.72 nano Watts.
- Simulation of the adder gave a power consumption of 21.24 nW.
- Therefore the total power consumption of our circuit is expected to be an average of 94.96nW.