Lab 5: Hierarchial Design and Formal Verification

Adam Sumner

Illinois Institute of Technology

ECE 429-01

Lab Date: October 14$^{\text{th}}$, 2015 **Due Date:** October 21$^{\text{st}}$, 2015

1 Introduction

The purpose of this lab is to introduce the student to hierarchal design and formal verification techniques that are essential when constructing complex circuits. A layout for a 2-input NAND gate will be built, and then a 2 input AND gate will be constructed using the schematics built from previous labs.

2 Theory/Pre-Lab

2.1 Theory

Since chip fabrication can be incredibly costly, it is a must to verify that the design will work correctly before sending it out for tapeout. Verifying functionality is one of the many important tasks that validates the success of a specific chip's design. While it is possible to give specific input to validate the functionality, there could potentially be a high number of inputs, producing hundreds of possibilities for output. Manual verification can become tedious and a waste of time. To resolve this issue, formal verification techniques have been proposed which can provide evidence for the functional correctness of a design. One such methodology is called equivalence checking. This technique takes two designs and verifies that both designs operate with the same functionality. LVS is one example of equivalence checking, as it shows where the layout is exactly the same as the schematic, while other factors such as transistor size are also verified. Because circuit designs can become complex, it's necessary to specify designs at a higher abstraction level. Due to the fact that the design in this lab is digital, boolean logic can be used for verification.

2.2 Pre-Lab

A stick diagram for a 2-input static CMOS NAND gate and a 2-input AND gate design were both sketched. They are shown in Figures 1 and 2.

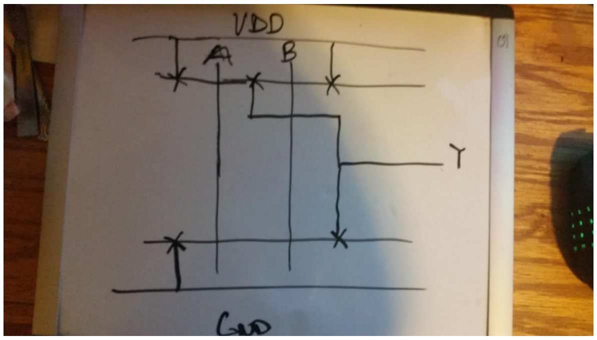


Figure 1: Stick Digram for Static CMOS 2-input NAND Gate

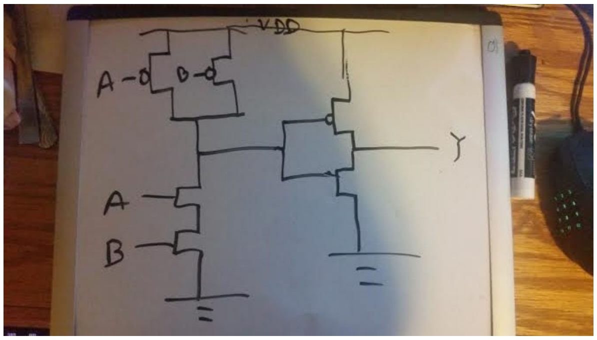


Figure 2: 2-input AND Gate Design

3 Implementation

3.1 Schematics

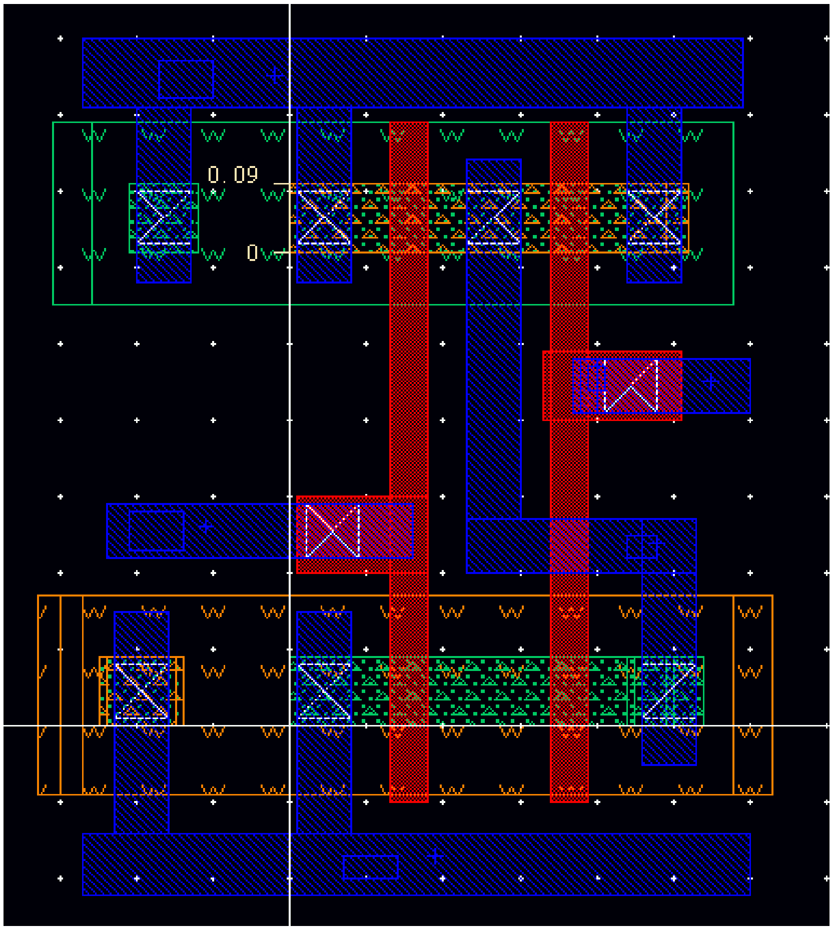


Figure 3: 2-input NAND Gate Layout

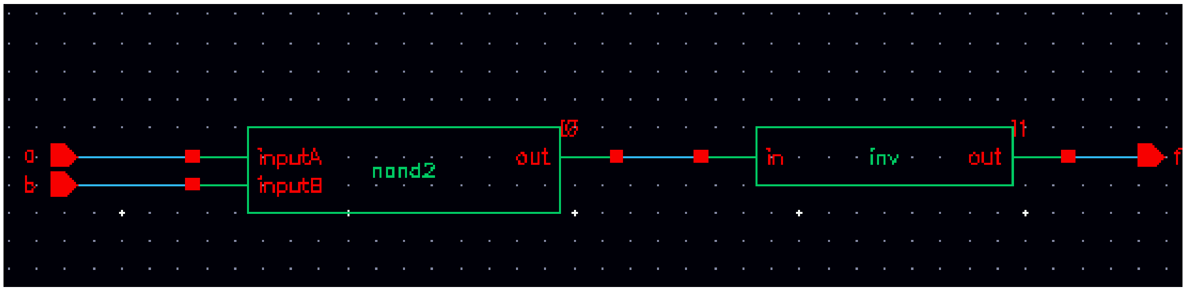


Figure 4: 2-input And Gate Schematic

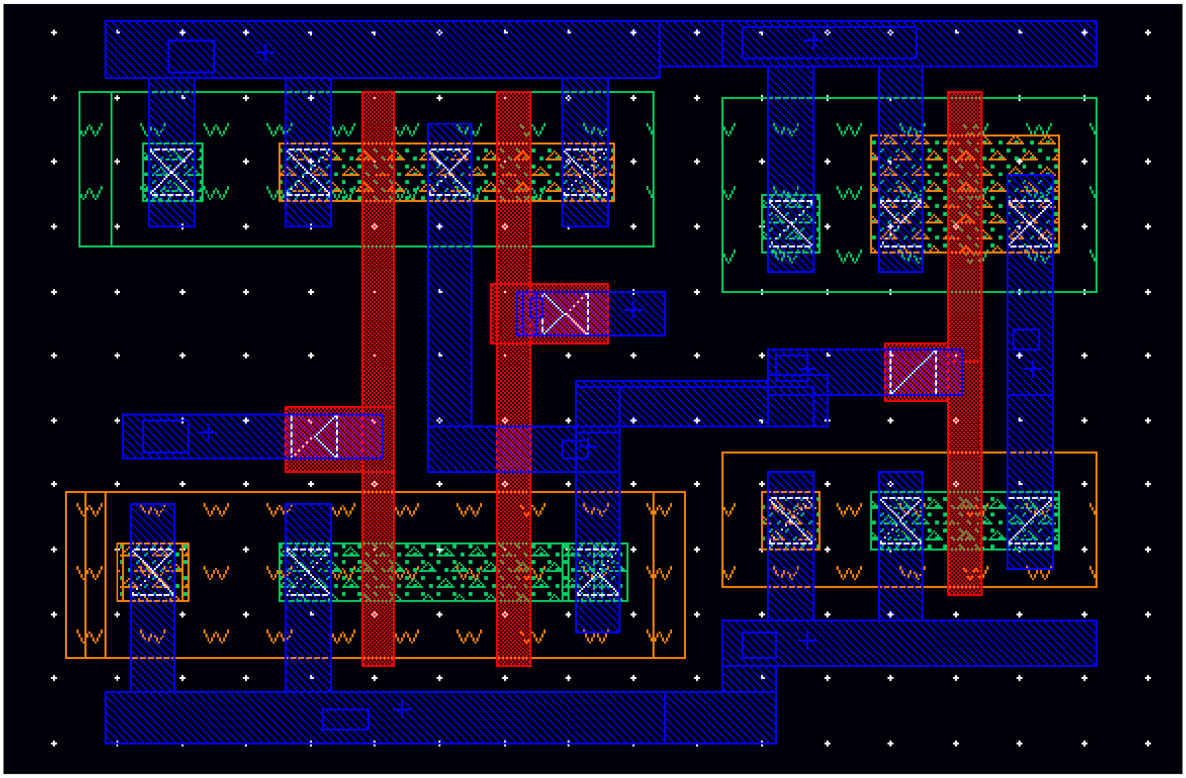


Figure 5: 2-input AND Gate Layout

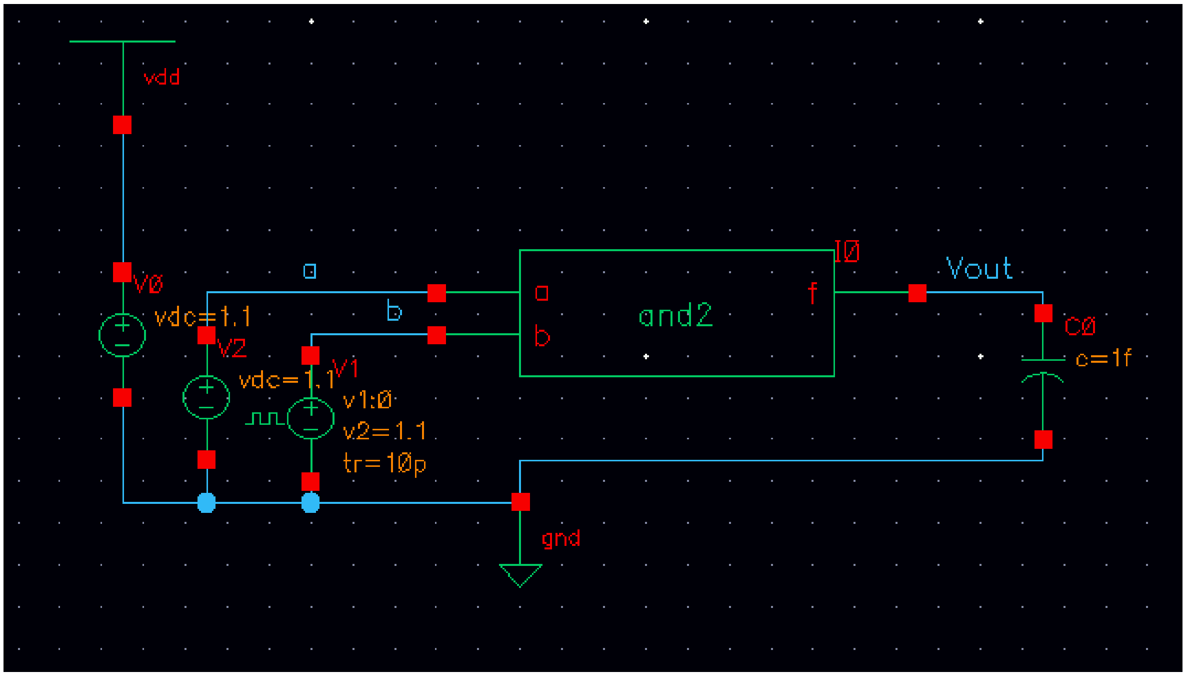


Figure 6: Test Circuit for 2-input AND Gate

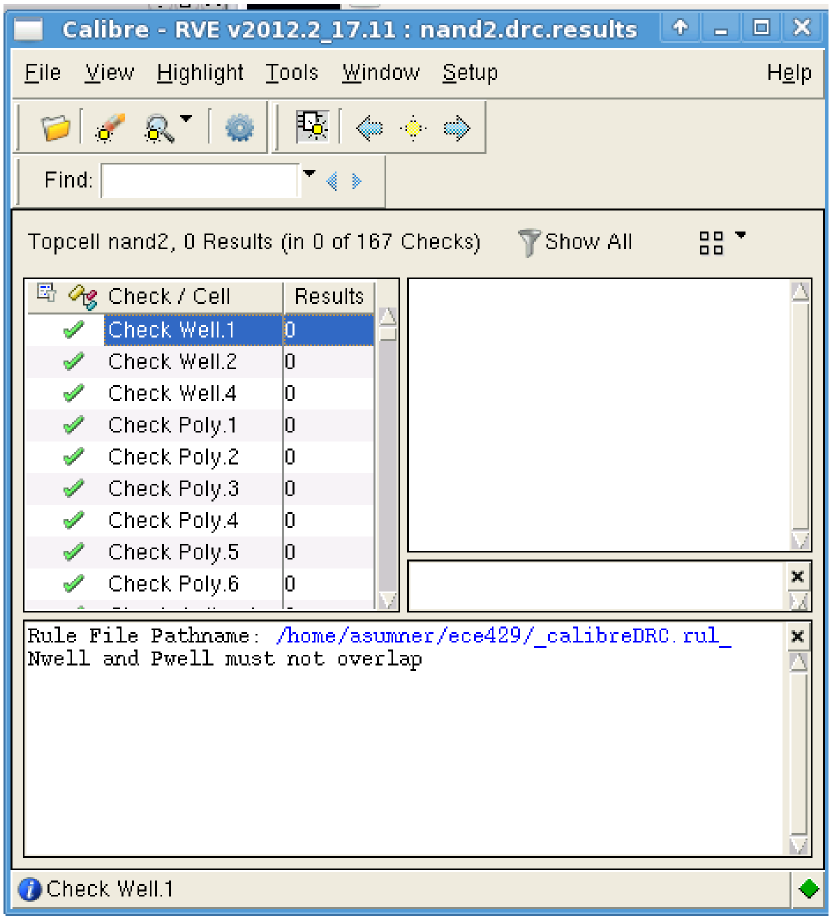
```
module and2(a,b,f);
  input a,b;
  output f;
  assign f=a&b;
endmodule
```

Figure 7: Verilog Model Code

3.2 Procedure

A NAND gate layout was created in Virtuoso and was verified using LVS. After this, a 2-input AND gate was designed using both the schematic and layout view. This was then verified using LVS, and it was also verified against the verilog model. Delay was then calculated for the AND gate design.

3.3 Results



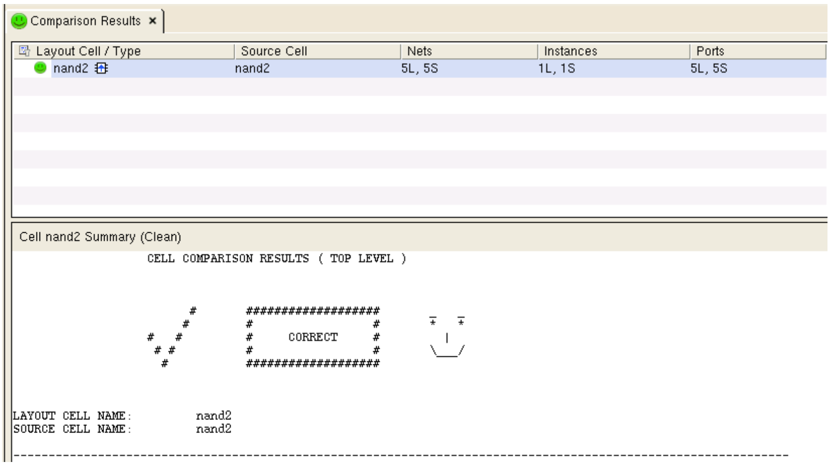
Topcell nand2, 0 Results (in 0 of 167 Checks)

Check / Cell	Results
✓ Check Well.1	0
✓ Check Well.2	0
✓ Check Well.4	0
✓ Check Poly.1	0
✓ Check Poly.2	0
✓ Check Poly.3	0
✓ Check Poly.4	0
✓ Check Poly.5	0
✓ Check Poly.6	0

```
Rule File Pathname: /home/asumner/ece429/_calibreDRC.rul_
Nwell and Pwell must not overlap
```

Check Well.1

Figure 8: 2-input NAND Gate DRC Results

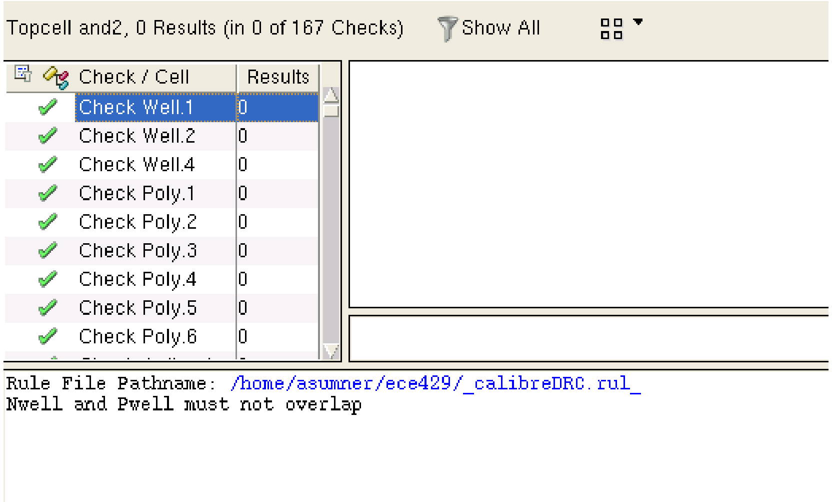


Layout Cell / Type	Source Cell	Nets	Instances	Ports
nand2	nand2	5L, 5S	1L, 1S	5L, 5S

Cell nand2 Summary (Clean)

```
                   CELL COMPARISON RESULTS ( TOP LEVEL )


                         #       ###################       _   _
                        #        #                 #       *   *
                    #   #        #     CORRECT     #         |
                     # #         #                 #       \___/
                      #          ###################


LAYOUT CELL NAME:         nand2
SOURCE CELL NAME:         nand2

--------------------------------------------------------------------------------------------------------------
```

Figure 9: 2-input NAND Gate LVS Results

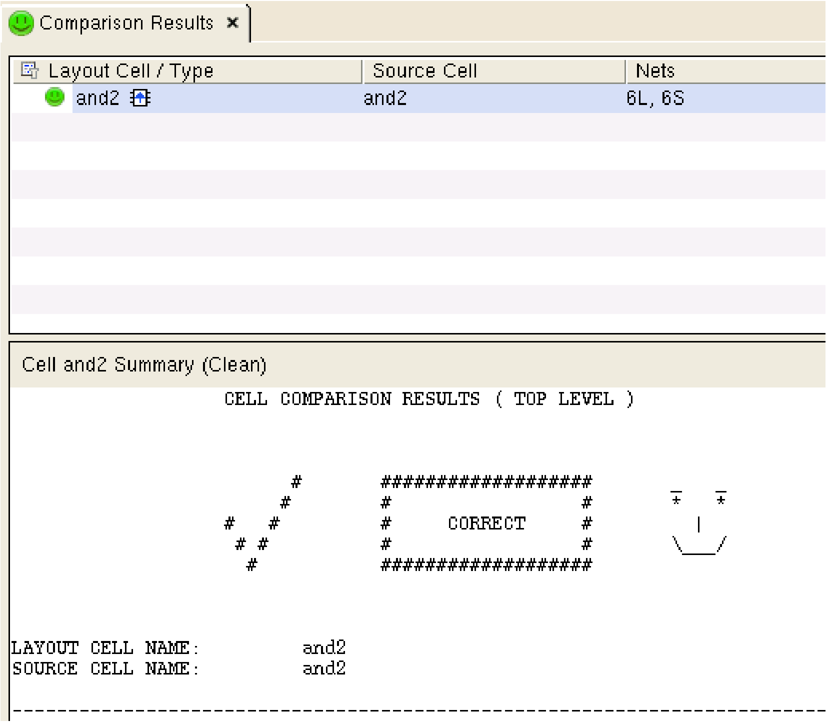


Topcell and2, 0 Results (in 0 of 167 Checks) Show All

	Check / Cell	Results
✓	Check Well.1	0
✓	Check Well.2	0
✓	Check Well.4	0
✓	Check Poly.1	0
✓	Check Poly.2	0
✓	Check Poly.3	0
✓	Check Poly.4	0
✓	Check Poly.5	0
✓	Check Poly.6	0

```
Rule File Pathname: /home/asumner/ece429/_calibreDRC.rul_
Nwell and Pwell must not overlap
```

Figure 10: 2-input AND Gate DRC Results

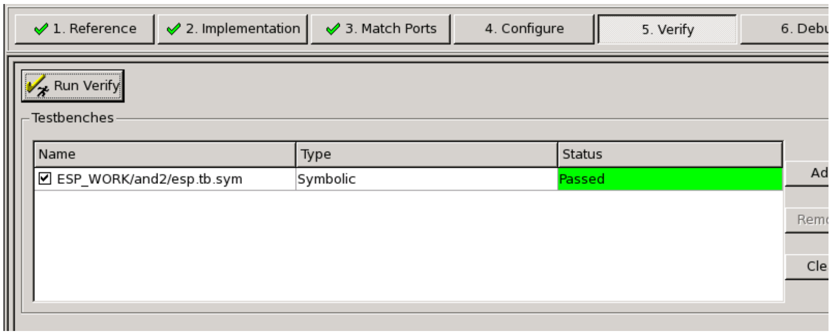


Layout Cell / Type	Source Cell	Nets
and2	and2	6L, 6S

Cell and2 Summary (Clean)

```
                    CELL  COMPARISON RESULTS ( TOP LEVEL )



                         #        ###################       _   _
                        #         #                 #       *   *
                   #   #          #     CORRECT     #         |
                    # #           #                 #       \___/
                     #            ###################


LAYOUT CELL NAME:          and2
SOURCE CELL NAME:          and2

--------------------------------------------------------------------------
```

Figure 11: 2-input AND Gate LVS Results



Name	Type	Status
☑ ESP_WORK/and2/esp.tb.sym	Symbolic	Passed

Figure 12: Verilog Equivalence Validated

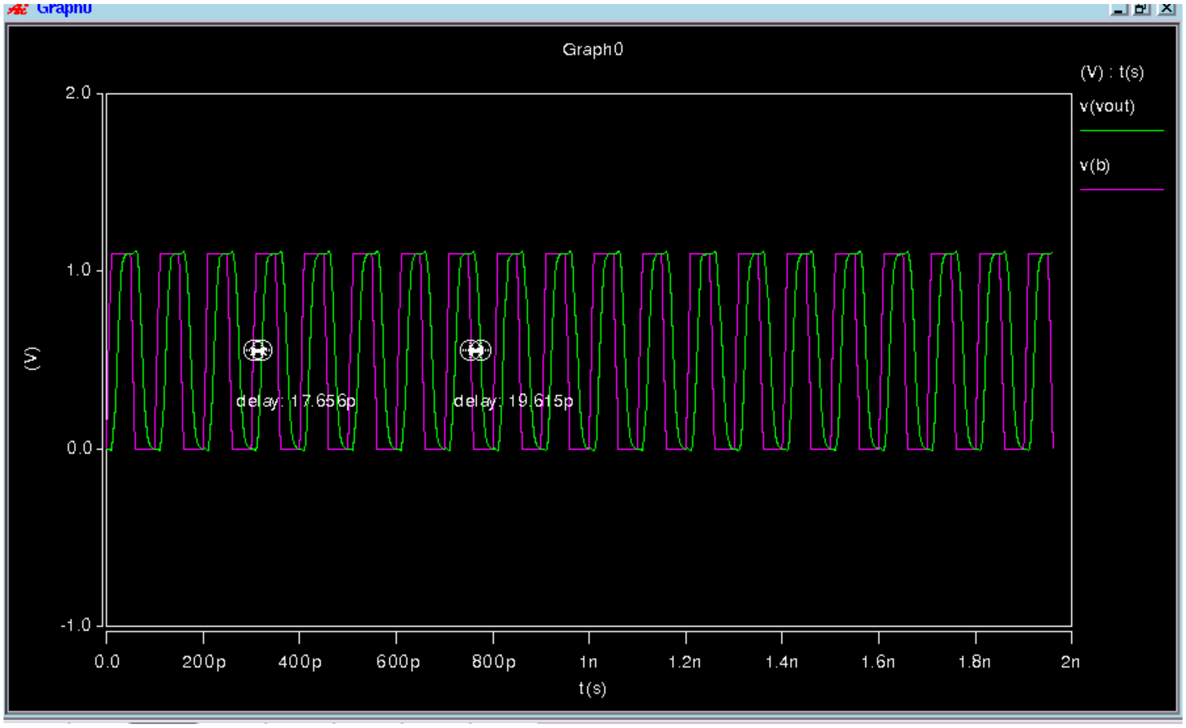


Figure 13: 10 → 11 Delay Measurements

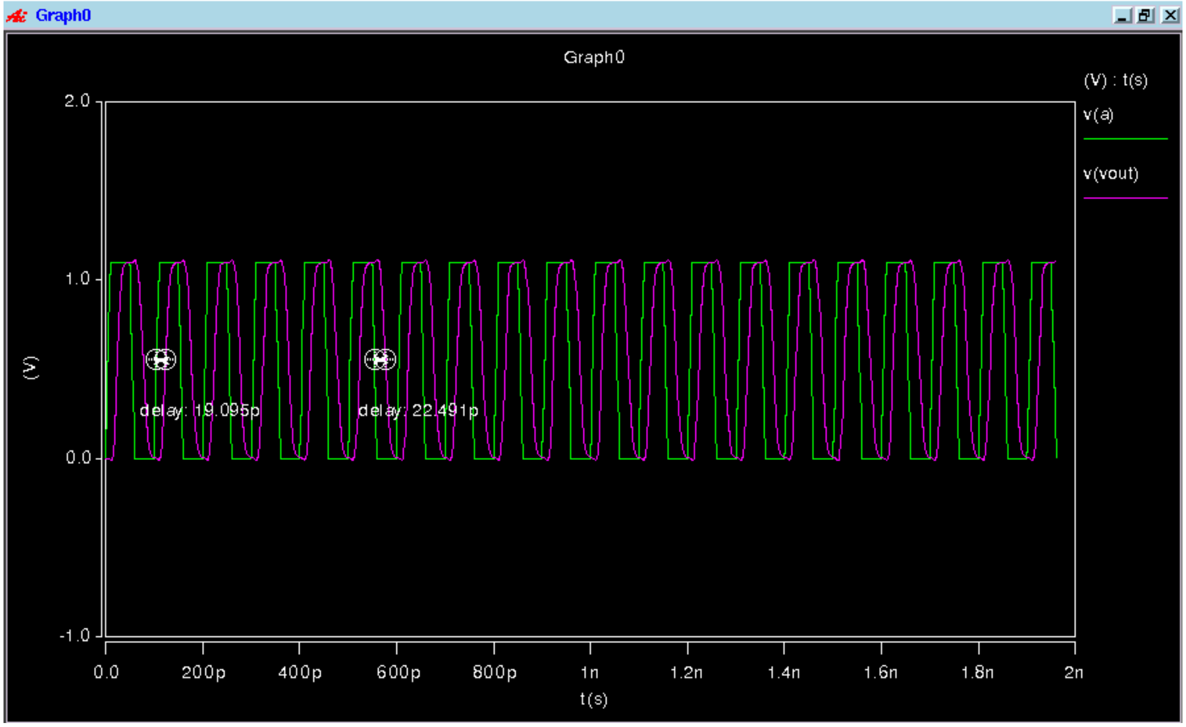


Figure 14: 01 → 11 Delay Measurements

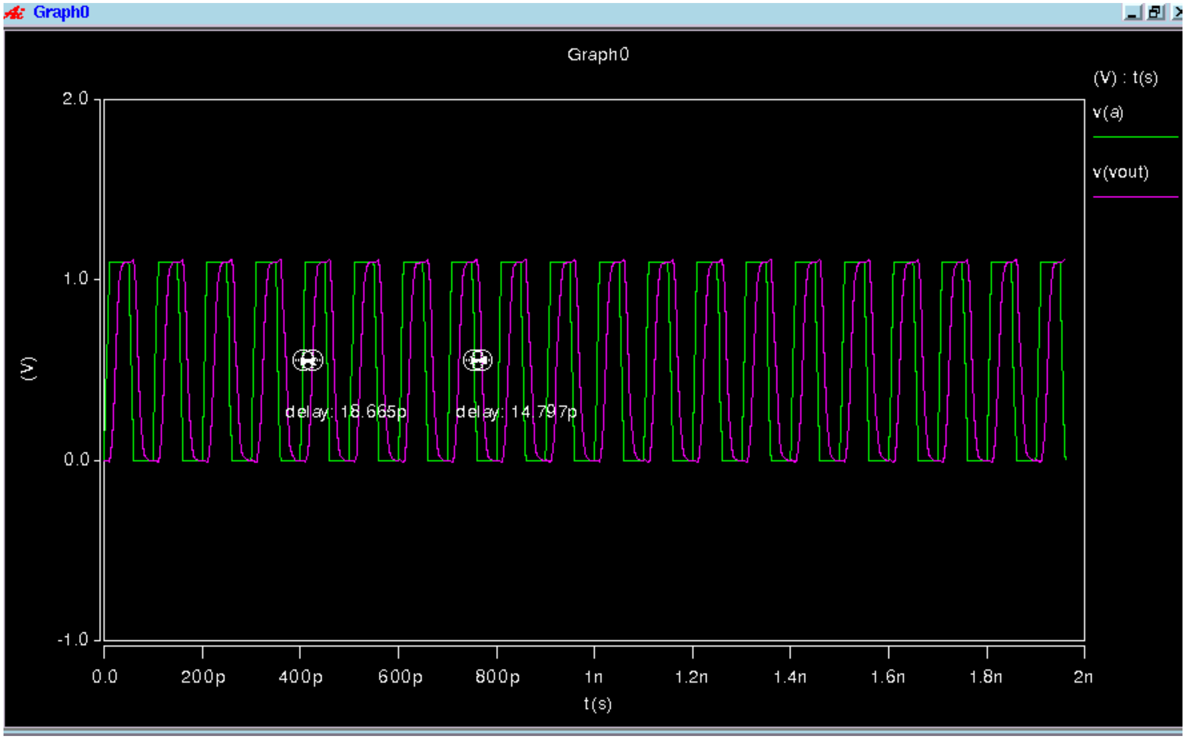


Figure 15: 00 → 11 Delay Measurements

w=90nm	Transition		
l=50nm	10→11	01→11	00→11
Rising Propogation Delay	17.656e-12 s	19.095e-12 s	18.665e-12 s
Falling Propogation Delay	19.615e-12 s	22.491e-12 s	14.797e-12 s
Average Power Consumption	3.17-05 W	3.26e-05 W	3.23e-05 W

Table 1: Delay and Power Consumption of Each Excition with 1f Load Capacitance

3.4 Discussion

Upon analysis of the results, it is clear that both a 2-input NAND Gate layout was successfully constructed, along with a 2-input AND Gate layout and schematic. Figures 8-12 show the validity of the equivalence checks while Figures 13-15 show the measurements of the delay. The results obtained are accurate and show that using previously created schematics, AND gate functionality was successfully obtained.

3.5 Questions

1. **Is there any other way to design an AND gate than combining an inverter with a NAND gate? Which one will you prefer?** Yes, it is possible to design an AND gate by using a 2-input NAND gate and then connecting its output to both inputs of another 2-input NAND Gate. This essentially turns the functionality of the second NAND gate into an inverter, producing the AND functionality desired. The preferred method is to use a NAND gate and an inverter since it utilizes less transistors.

2. **Do the input transitions leading to the propagation delays match your expectations? Why or why not?** Yes, it was expected that the delays would be much longer than that of a regular NAND gate due to the complexity of this circuit.

3. **Is there any relationship between the propagation delays you measured in this lab for the AND gate and those you measured in Lab 2 for the inverter and in Lab 4 for the NAND gate?** Yes it seems that the delays were additive, which makes sense since the two logic gates are daisy chained together.

4. **What is the difference between a schematic and a Verilog model?** A Verilog model is a way to represent a circuit purely with code. This makes it incredibly easy to quickly construct and simulate circuits. A schematic physically shows the layout of the circuit, requiring some sort of gui design software to lay the gates out.

4 Conclusions

Overall the lab was a success. A 2-input NAND gate layout was successfully created, and it was used in conjunction with a previously created Inverter schematic to design AND Gate functionality. Furthermore, the design was successfully verified using both LVS and Verilog equivalence checking. The AND gate may now be used in future designs.