

EE 415 Project Report for

Cascadable 4-Bit Comparator

By William Dixon

Mailbox 509

June 1, 2010

- INTRODUCTION 3**
- THE CASCADABLE 4-BIT COMPARATOR..... 4**
 - CONCEPT OF OPERATION.....4
 - LIMITATIONS.....5
 - POSSIBILITIES FOR IMPROVEMENT5
- CONSTRUCTION AND SIMULATIONS..... 6**
 - SCHEMATIC.....6
 - INITIAL LOGIC TESTING.....6
 - LAYOUT7
 - LVS AND PEX8
 - POST-LAYOUT SIMULATION8
 - COMMENTS ON INITIAL SCHEDULE8
- CONCLUSION 9**
- APPENDIX A: I/O AND TRUTH TABLE..... 10**
 - INPUTS AND OUTPUTS10
 - TRUTH TABLE11
- APPENDIX B: RESULTS OF POST-LAYOUT SIMULATION 12**
 - SUMMARY.....12
 - PLOTS AND DISCUSSION13
- APPENDIX C: FINAL LAYOUT..... 18**
- APPENDIX D: REFERENCES 19**

Introduction

A comparator is an electrical circuit that takes two values – either digital values composed of 1s and 0s or analog voltages – and changes its output to reflect the relationship or difference between those values. During normal operation, this particular comparator takes two 4-bit digital values “**A**” and “**B**” and then sets “high” one of three outputs: **A > B**, **A < B**, or **A = B**. This comparator also has the ability to be cascaded with identical comparators in order to build a circuit that compares two values having any integer multiple of 4 bits.

The comparing operation is implemented using a collection of various digital logic gates, which are themselves created using field-effect transistors. Thus, the entire comparator is created using hundreds of tiny, strategically-linked transistors and metal wires. The comparator circuit was not actually assembled in the real world; it was instead built and tested using computer software.

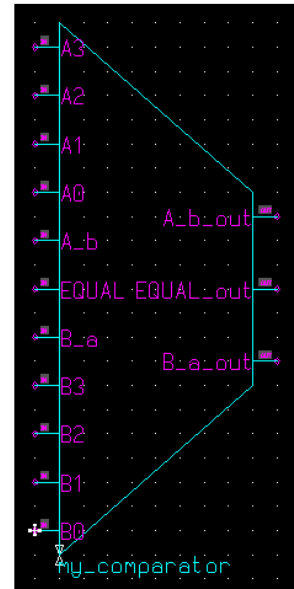
This report describes the comparator’s concept of operation and limitations as well as the builder’s method of construction and possibilities for improvement of the comparator. The appendices include descriptions of the circuit’s input and output (including a truth table relating these), simulation results for the final comparator circuit, and references mentioned in the report.

The Cascadable 4-Bit Comparator

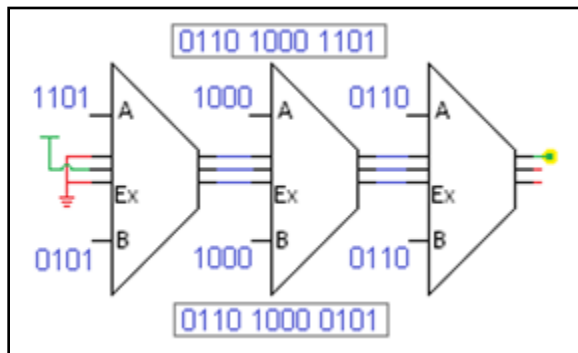
Concept of Operation

This cascadable 4-bit comparator normally takes two 4-bit digital values (“nibbles”) and changes the state of its output pins based on the difference between these values (see Appendix A for I/O description and truth table). During 4-bit operation, the expansion inputs A_b and B_a are tied to ground while the *EQUAL* input is tied to V_{DD} .

To construct a comparator that compares larger binary values, one must simply connect a number of these comparators in series, with the output of one stage connected to the expansion inputs of the next. The **A** and **B** inputs of the first stage will then be treated as the least significant bits and the inputs of the final stage will have its inputs treated as the most significant bits in the chain (the number gets “reversed” in a way).



4-Bit Comparator Symbol



Example of 12-Bit Cascaded Operation

One of the basic operations on which the comparator relies is the single-bit comparison operation. A one-bit comparator would have a “0” output unless both input bits were the same – either 1 or 0. This can be accomplished with a simple, familiar gate, XNOR.

A	B	Out
0	0	1
1	0	0
0	1	0
1	1	1

XNOR Truth Table

Although the 4-bit comparator does not explicitly use XNOR gates, it does perform the same basic function: it compares the corresponding

bits of inputs **A** and **B** and then uses these four results to figure the greater (or the equality) of the two numbers.

Limitations

The cascadable 4-bit comparator has aspects that limit its usefulness. The circuit schematic (see next page) shows that some logic gates used in the comparator have a rather large fan-out (especially the 4 main NOR gates). If the gates driving this level of fan-out are not sized properly, the speed at which they can switch their outputs between high and low (and with it the speed of the entire comparator) suffers. The intrinsic capacitances of the many transistors and the capacitance caused by relatively large lengths of interconnecting wire also add to the speed limitation of the circuit. Thus, the comparator will have a limited frequency up to which it can be operated reliably. Cascading comparators has a multiplicative effect on this limitation such that a 16-bit comparator made from a chain of 4-bit components may have a propagation delay around four times larger than a 4-bit part.

Appendix B shows that the maximum recorded propagation delay of the 4-bit comparator is approximately 1.8 nanoseconds, which suggests a maximum operating frequency (only considering propagation delay) of roughly 560 MHz; with an arbitrary 20% “safety factor” (for lack of a better term), this frequency falls to around 450 MHz. This speed would certainly not allow it to operate with the full clock frequency on a modern computer processor.

Possibilities for Improvement

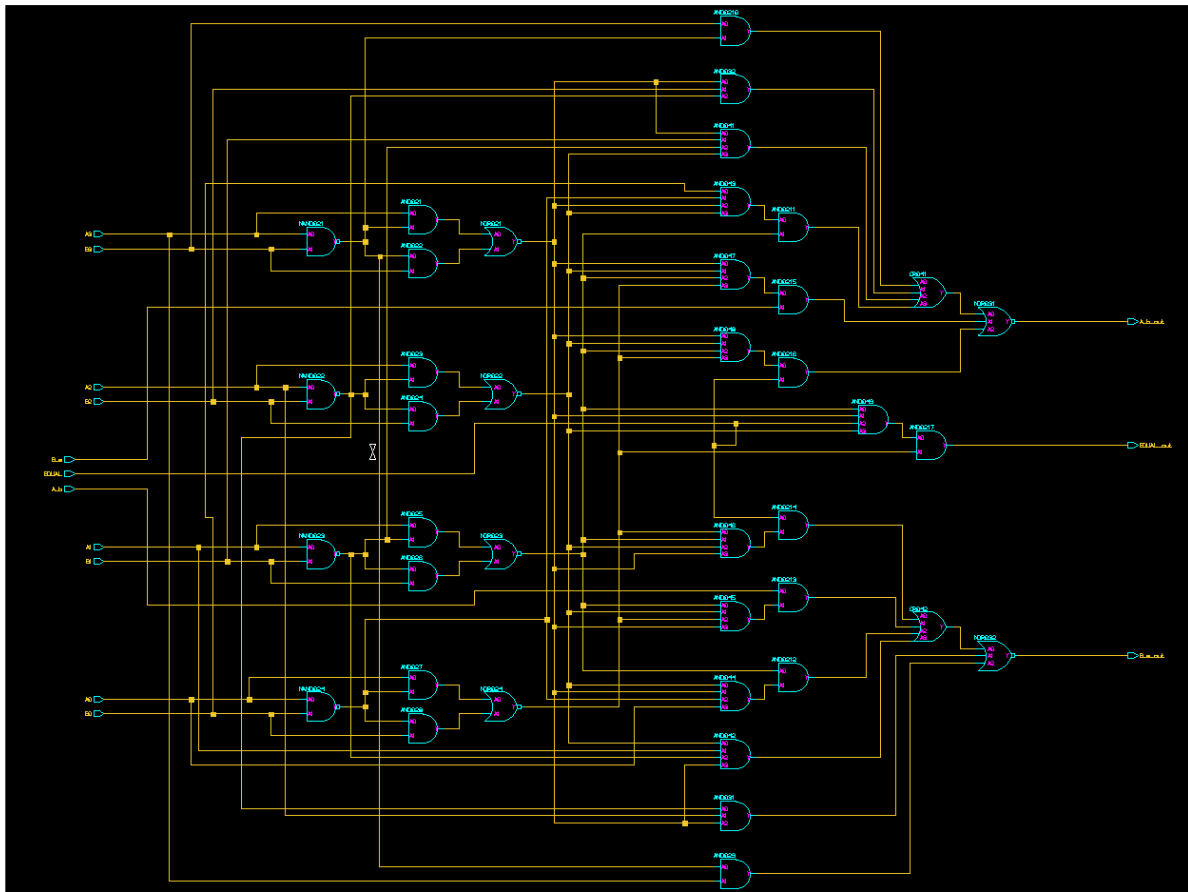
As can be seen in Appendix B, when the comparator experiences an abrupt falling edge (1 ps fall time), one of the outputs is momentarily pulled high; in this particular case, it takes 1.8 ns for that output to settle back to its intended value. If this response were eliminated, the overall delay would be reduced (in the falling edge case) to 1.2 ns – a substantial speed increase. The same 1.2 ns delay can be seen when an input experiences a rising edge.

The gates used to build this comparator were generic ones from the logic library of Design Architect. Especially in the case of gates driving large fan-out, if some of these gates were customized for this comparator application, a speed increase could result.

Construction and Simulations

Schematic

First, a schematic was created in Design Architect. This schematic was derived from the 7485 comparator circuit shown at the URL referenced in Appendix D. This schematic has a large number of wires interconnecting the various logic gates and therefore keeping track of them while building the circuit was essential. A large printout of the circuit was made and a green marker used to trace each wire that was placed on the schematic.



Comparator Schematic (note the spaghetti-like mass of wires)

Initial Logic Testing

The schematic was tested using the analog / mixed signal (AMS) simulation function in Design Architect. The simulation put various digital values onto the **A** and **B** inputs of the comparator circuit and plotted these inputs along with the circuit's outputs. It also tested for correct

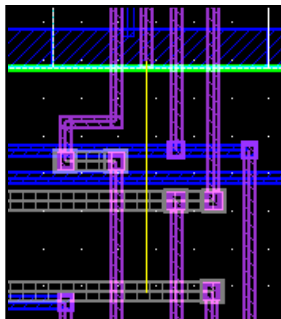
operation with respect to the circuit’s expansion inputs. The first simulation revealed a problem with the comparator’s logic; this was quickly found to be caused by a missing wire. Once the wire was put into place, the circuit performed as expected; however, no propagation delay was evident due to the fact that parasitics had not yet been included in the simulation.

Layout

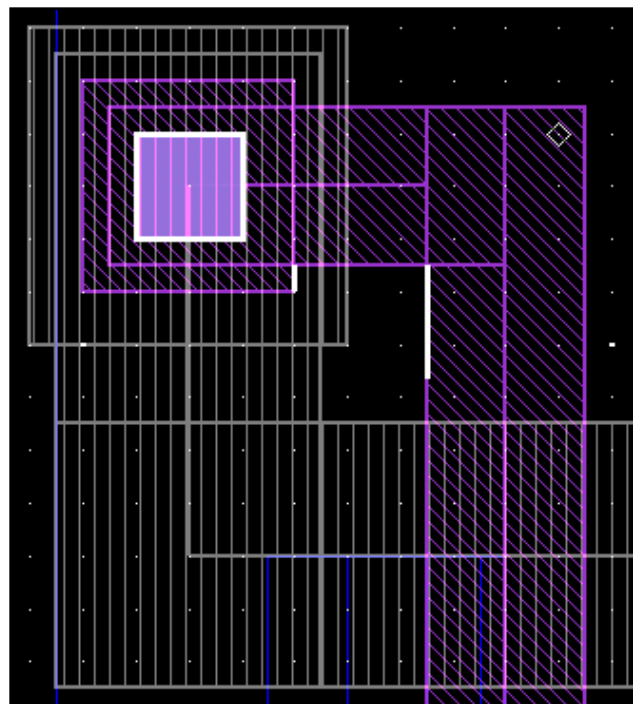
After the required viewpoints were created for the comparator, a layout was created using IC Station. Although the program’s automated functions were used to create a floor plan, place logic cells and ports, and route interconnecting wires, several issues had to be overcome.

IC Station’s automated wire routing function made several errors, some of which were design rules violations involving metal spacing. In order to correct these spacing errors, the wires were manually re-routed such that no design rules were violated.

In addition to design rules violations, the automated routing function failed to place several wires. These missing wires were indicated by yellow overflow lines left over after automated routing. Wires had to be placed by hand wherever a routing failure had occurred.



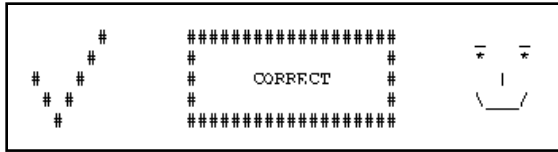
Routing Failure



Two Design Rules Violations: Metal2 and Metal3 Spacing (Metal 2 Highlighted)

LVS and PEX

After the initial layout was complete, the layout-versus-schematic check was run and passed. Extraction of parasitics initially failed, however, and it became clear that ports and nodes with names including special symbols like “:” and “>” cause errors during PEX. Thus, the schematic



Cheerful Indicator of Successful LVS Check

had to be edited and all prior steps repeated (save for the AMS) for the new layout. This new design passed the LVS check once again and parasitic extraction was completed using Calibre.

```
Warning: Undefined parameter "B_OUT" ignored in file "/home/wdixon/mgc/class/my_co
Error: Syntax Error in file "/home/wdixon/mgc/class/my_comparator_netlist" at line 3.
Error: Syntax Error in file "/home/wdixon/mgc/class/my_comparator_netlist" at line 56.
Error: Syntax Error in file "/home/wdixon/mgc/class/my_comparator_netlist" at line 74.
Error: Syntax Error in file "/home/wdixon/mgc/class/my_comparator_netlist" at line 75.
Error: Syntax Error in file "/home/wdixon/mgc/class/my_comparator_netlist" at line 129.
Error: Syntax Error in file "/home/wdixon/mgc/class/my_comparator_netlist" at line 192.
Error: Syntax Error in file "/home/wdixon/mgc/class/my_comparator_netlist" at line 210.
Error: Syntax Error in file "/home/wdixon/mgc/class/my_comparator_netlist" at line 211.
Error: Syntax Error in file "/home/wdixon/mgc/class/my_comparator_netlist" at line 265.
ERROR: Source could not be read.
```

Errors Caused by Special Characters in Port Names

Post-Layout Simulation

The results of the parasitic extraction were used to repeat the AMS simulation with parasitics included. This type of simulation uses parasitic parameters (internal and external capacitance values, higher-order transistor effects, equivalent resistance values) formatted to be used by SPICE (itself a circuit simulation tool) in order to mimic the comparator circuit’s real-life behavior.

Plots demonstrating the behavior of the comparator can be found in Appendix B. The AMS simulation with parasitics verified that the circuit’s logic works correctly and that its average propagation delay is approximately 1 nanosecond. Due to a glitch that can be seen in Appendix B, the current comparator circuit is not suitable for production.

Comments on Initial Schedule

Although the required tasks changed as I completed more of the project, the initial schedule included in the project proposal was surprisingly accurate and easy to abide by.

Conclusion

The goal of building a reasonably fast, cascadable 4-bit comparator has been achieved. The results of post-layout simulation suggest that the circuit has room for improvement, particularly its reliability (see the “hump” issue in Appendix B) and speed. Another possible improvement to the circuit would be expansion inputs that pull themselves down or up when left floating – particularly if this could be accomplished without much increase in static power dissipation.

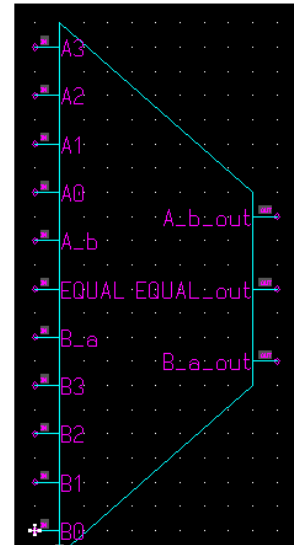
Overall, the project has been a success.

Appendix A: I/O and Truth Table

Inputs and Outputs

Excluding power rails, this comparator has 8 inputs, 3 expansion inputs, and 3 outputs, all of which are described below.

- Inputs
 - A3 – MSB of input value **A**
 - A2
 - A1
 - A0 – LSB of **A**
 - B3 – MSB of input value **B**
 - B2
 - B1
 - B0 – LSB of **B**
- Expansion Inputs
 - These are not used for 4-bit operation; however, for cascaded operation they can be connected to the previous stage's eponymous outputs. If the comparator senses that $\mathbf{A} = \mathbf{B}$, the expansion inputs determine the output.
 - $\mathbf{A} > \mathbf{B}$ (*A_b* in symbol) – Set low for 4-bit operation
 - $\mathbf{B} > \mathbf{A}$ (*B_a*) – Set low for 4-bit operation
 - $\mathbf{A} = \mathbf{B}$ (*EQUAL*) – Set high for 4-bit operation
- Outputs
 - $\mathbf{A} > \mathbf{B}$ (*A_b_out* in symbol)
 - $\mathbf{B} > \mathbf{A}$ (*B_a_out*)
 - $\mathbf{A} = \mathbf{B}$ (*EQUAL_out*)



4-Bit Comparator Symbol

Truth Table

Note that X signifies a “don’t care” state.

Expansion Inputs			Inputs								Outputs		
A > B	B > A	A = B	A3	B3	A2	B2	A1	B1	A0	B0	A > B	B > A	A = B
X	X	X	1	0	X	X	X	X	X	X	1	0	0
X	X	X	0	1	X	X	X	X	X	X	0	1	0
X	X	X	0	0	1	0	X	X	X	X	1	0	0
X	X	X	0	0	0	1	X	X	X	X	0	1	0
X	X	X	0	0	0	0	1	0	X	X	1	0	0
X	X	X	0	0	0	0	0	1	X	X	0	1	0
X	X	X	0	0	0	0	0	0	1	0	1	0	0
X	X	X	0	0	0	0	0	0	0	1	0	1	0
1	0	0	All Ax = Bx								1	0	0
0	1	0									0	1	0
0	0	1									0	0	1

Appendix B: Results of Post-Layout Simulation

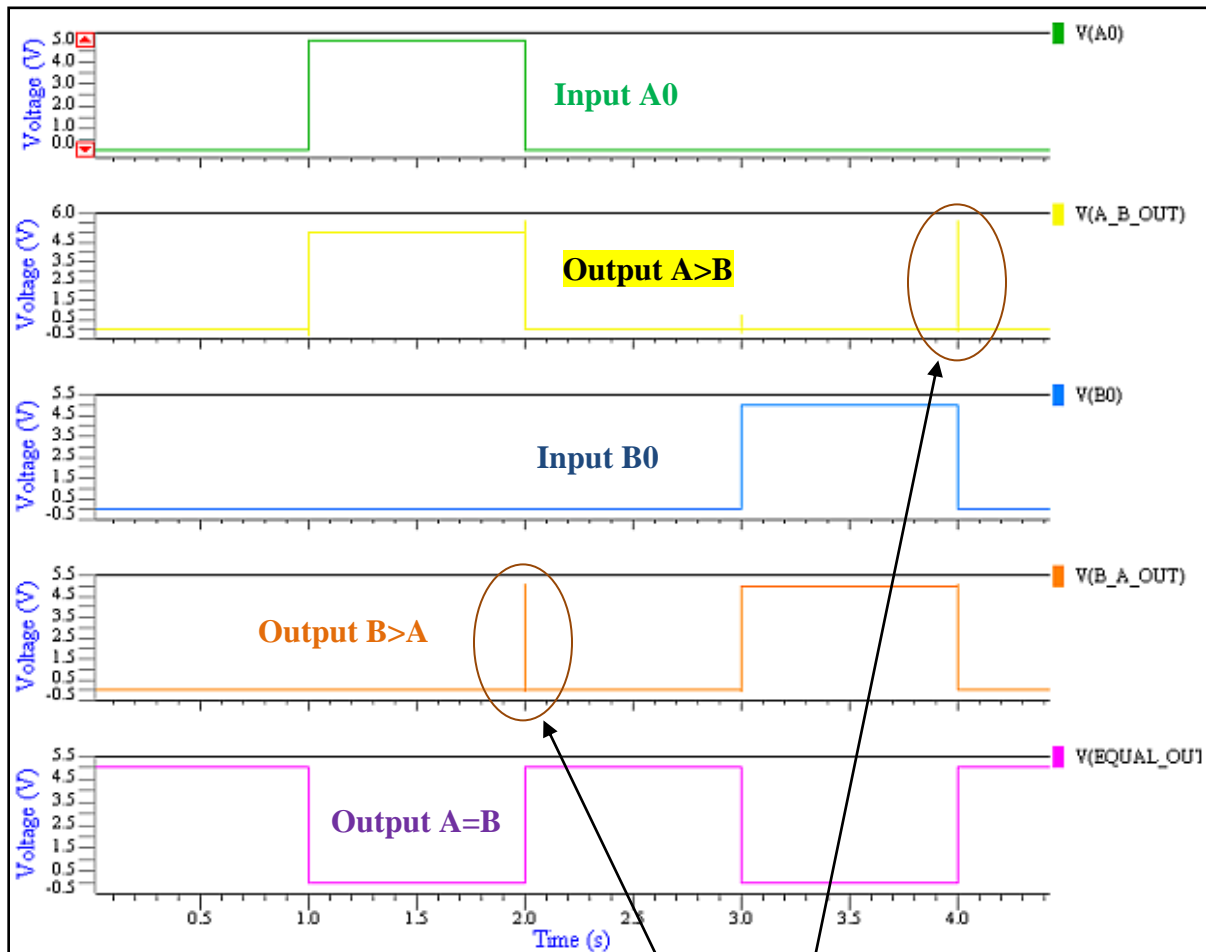
Since there are a very large number of input combinations that could be tested, only a handful were selected to test the characteristics of the comparator (the logic already having been verified pre-PEX). These include one input having a 1 ps H-L transition, an input having a 1 ps L-H transition, and multiple inputs switching states at the same time (1 ps transition times).

Summary

- Approximate L-H propagation delay – 1.2 ns (Worst case)
- Approximate H-L propagation delay – 1.8 ns (Worst case)
- The more inputs that switch simultaneously, the lower overall propagation delay
- Glitch when input goes H-L – Causes “hump” and extra delay
- Calibre gives power dissipation as 3.9 nW (likely static, not dynamic)

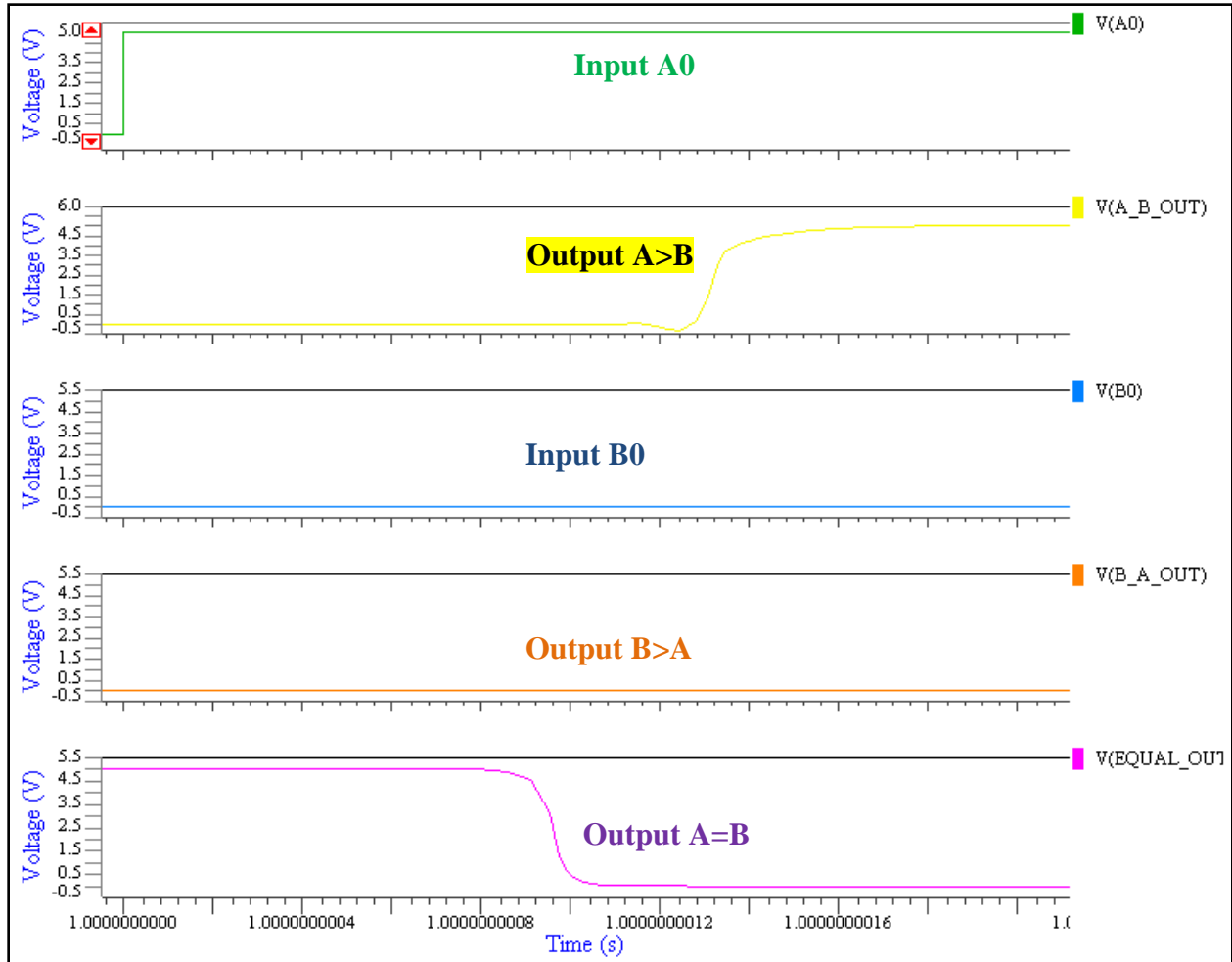
Plots and Discussion

This plot demonstrates the basic operation of the comparator. The MSBs of **A** and **B** are varied and the outputs of the circuit change as expected to indicate the inputs' relationship. Note the effect of the inputs' falling edges on the circuit's outputs.



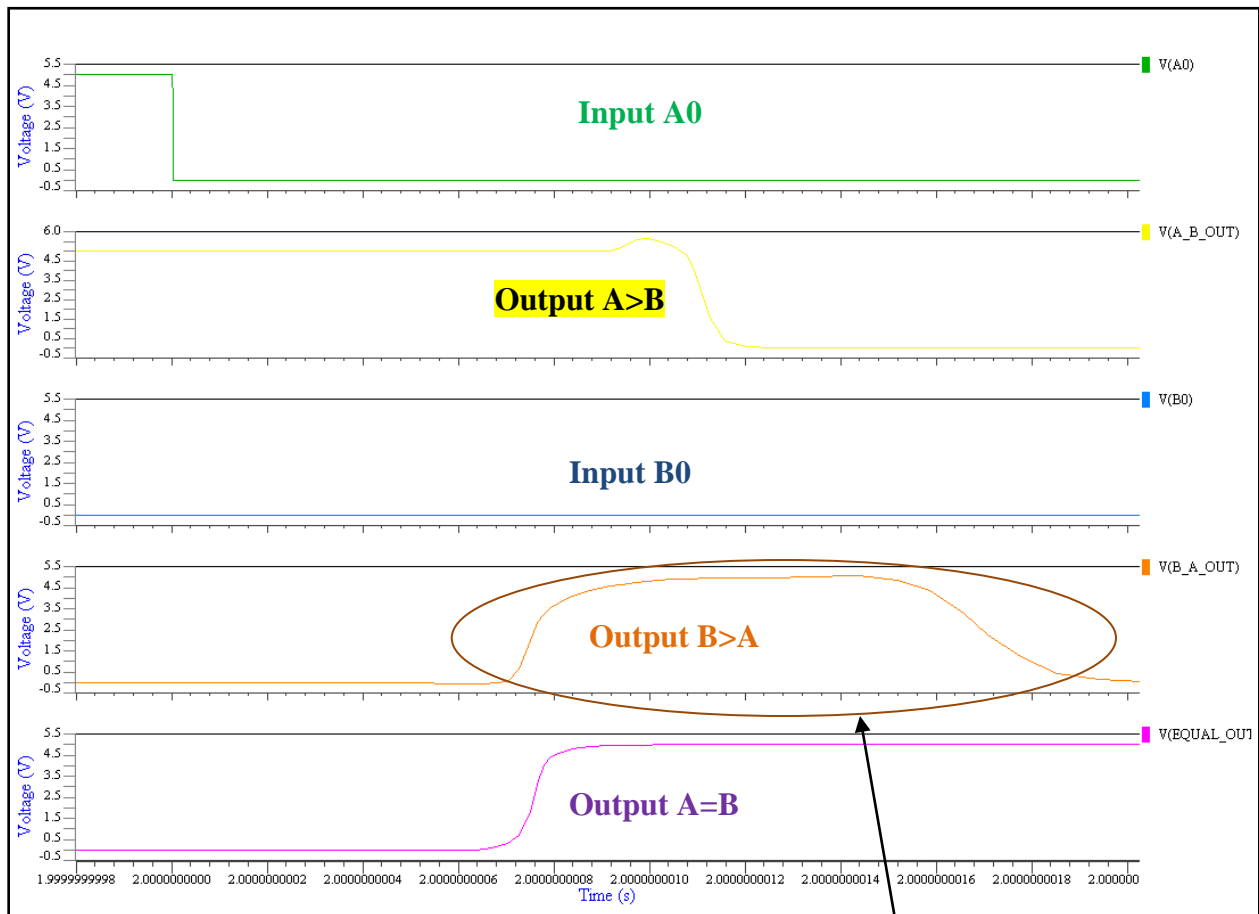
Basic Test of Comparator

This plot shows the result of a rising edge on the MSB of A. This simulation was found to be representative of any other single-input rising edge. The delay between the rising edge and the settling of the comparator's outputs appears to be close to 1.3 ns. Note that the *EQUAL_out* output has a much faster response than *A_b_out*.



Response to Single-Input Rising Edge

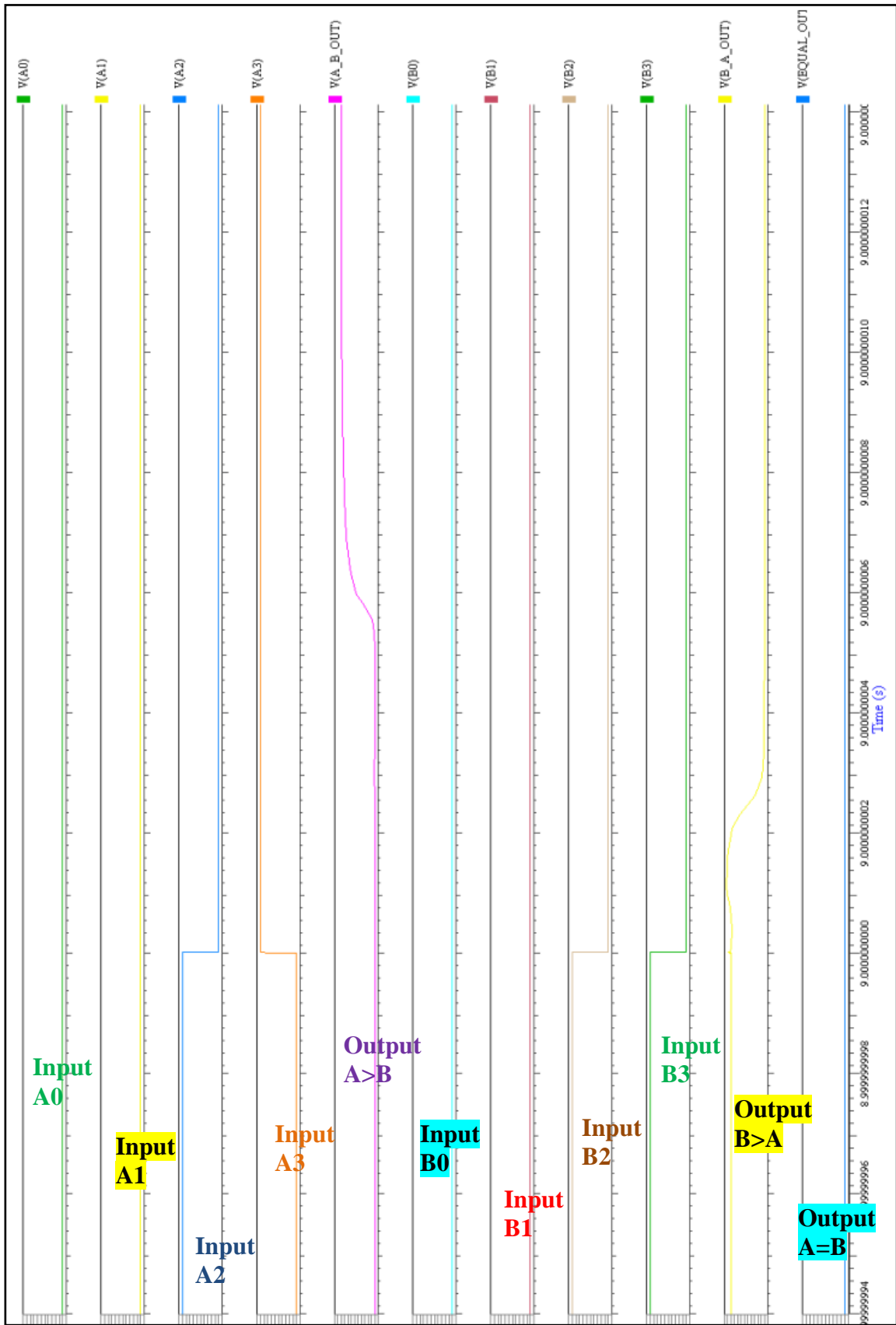
This plot shows the result of a falling edge on A0. This was found to be representative of any other single-input falling edge. Unlike the rising edge, the falling edge causes a response on all three outputs of the comparator, including one response (the “hump” on B_a_out here) that is unexpected. This “hump” causes a 1.8 ns delay between the falling edge and the settling of the outputs. This hump is likely caused by the delays of the intermediate gates between the input and output path; B_a_out is momentarily driven high and then pulled back down. This glitch makes the current comparator build unsuitable for production.



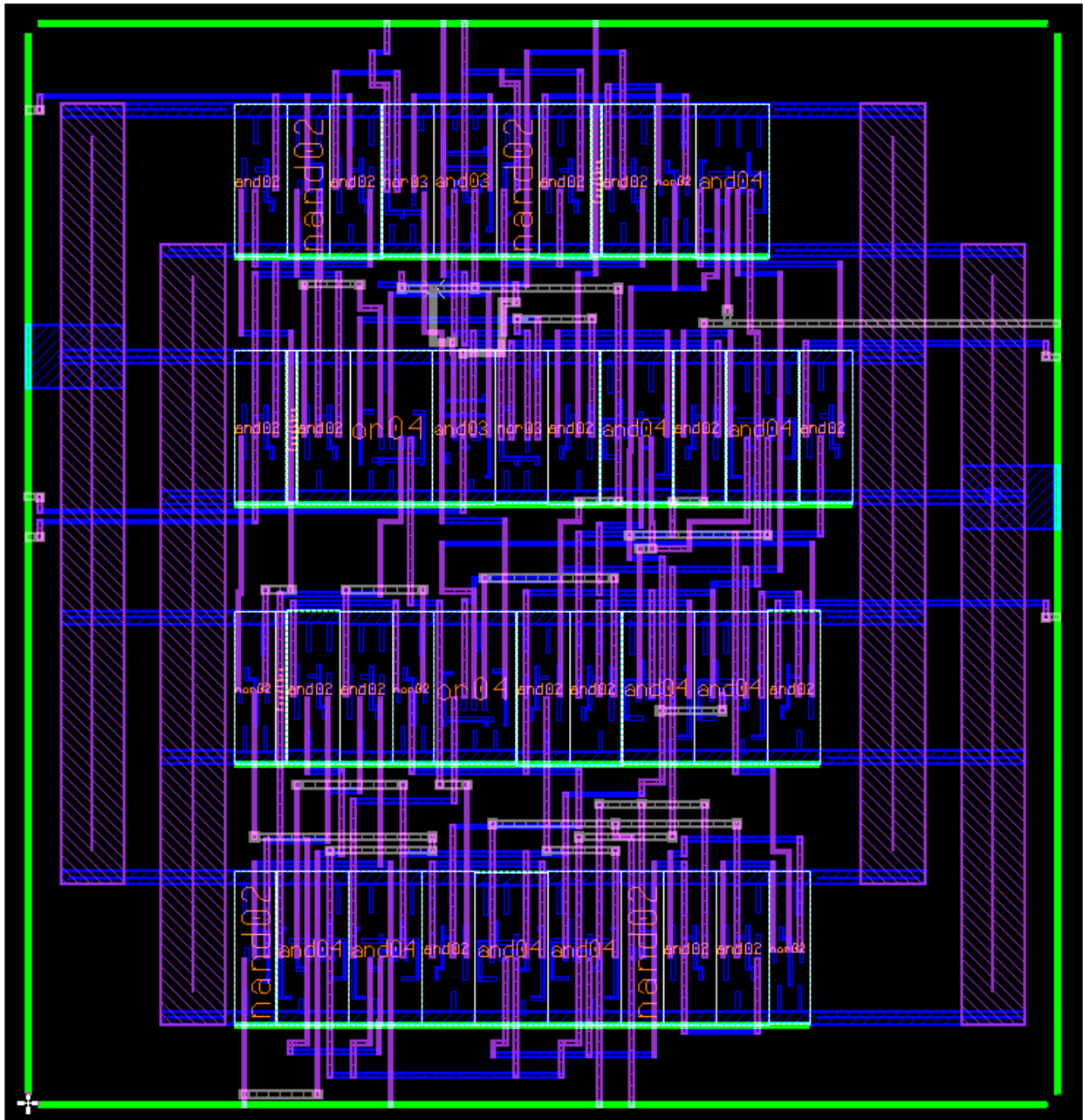
Response to Single-Input Falling Edge

“Hump” due to input’s falling edge

The plot on the following page shows the circuit's response to the changing of several inputs at the same time. The time it takes for the outputs to settle is approximately 0.7 ns, less than the 1.2 nanoseconds for a single-input change. This suggests that as more outputs change simultaneously, the propagation delay becomes smaller – this was confirmed by a small number of further simulations. Thus, the worst-case propagation delay occurs, in general, when a single input changes.



Appendix C: Final Layout



Final Layout

Appendix D: References

7485 Comparator Schematic and Logic Simulation:

<http://tams-www.informatik.uni-hamburg.de/applets/hades/webdemos/20-arithmetic/45-compare/7485-comparator.html>

Link to PDF Version of this File:

http://www.angelfire.com/oh4/thevault/VLSI_comparator.PDF