Autolayout Design Techniques (HDL-Layout) – Lab Number 7

**<INSERT SIGNOFF SHEET>**

Author : Brian Landy

Lab Instructor : Sayed Ashraf Mamun

TA : Andrew Fountain

TA : Sabrina Ly

Lab Section : Lab #7

Submission Date : 11/14/2019

Table of Contents

[ABSTRACT 3](#_Toc24564252)

[DESIGN METHODOLOGY 3](#_Toc24564253)

[RESULTS & ANALYSIS 3](#_Toc24564254)

[CONCLUSIONS 4](#_Toc24564255)

[QUESTIONS 4](#_Toc24564256)

[APPENDIX 4](#_Toc24564257)

[SOURCES 4](#_Toc24564258)

# ABSTRACT

This exercise involved using VHDL code and Pyxis Layout to generate a physical design of an n-bit ALU. The circuit was only described in the hardware description language. The procedure involved writing VHDL for a 1 bit and 16 bit ALU with 4 opcodes. The VHDL was functionally tested and the code was then used in Pyxis to create a layout that was then tested in eldo and viewed with Ezwave. There were many cases used to determine the performance of the 1 and 16 bit ALU’s. Overall the 1 bit ALU had a maximum input frequency of .372GHz and a max throughput frequency of .459 Ghz and the 16 bit had a maximum input frequency of .506Mhz and a maximum throughput frequency of .195Ghz. This exercise was based on the process of generating circuits based on VHDL with auto layout tools. Static and dynamic power was also measured for the auto routed circuits. The static power and dynamic power of a 1 bit ALU are 1999 nW AND 1470 uW. The static and dynamic power for the 16 bit ALU are 50834 nW and . It ended up a success as the process to go from a HDL file to an auto generated layout now routine and repeatable.

# DESIGN METHODOLOGY

# Procedure

The first part of the exercise consisted of creating the VHDL files that describe the circuits. The ALU was able to be done generically and all in one file using built in math libraries in the IEEE libs. One important thing to note is that when working with auto layout, normal VHDL doesn’t always cut it. All signals must be assigned for all cases. So for logical operations, the Carry out bit must be assigned as well and no static values are allowed. Only logic in can be used for logic out. The ALU is described best by table 1 which shows the operations it is capable of.

Table 1: ALU Operation

|  |  |  |  |
| --- | --- | --- | --- |
| **Opcode** | **Operation** | **Operands** | **Outputs** |
| 00 | AND | A and B | Output |
| 01 | OR | A or B | Output |
| 10 | ADD | A+B | Output, Cout |
| 11 | SUBTRACT | A-B | Output, Cout |

The opcode logic was implemented with some straightforward “if “cases and the generic VHDL code is shown in the appendix for the n bit alu. The code was tested in VSIM for functionality and correctness before the layout step. The results are discussed in the Results section.

To synthesize the circuit and use auto layout, a spectrum file was made for this VHDL. The code is shown in the appendix. This script generates the vhd files for the schematic step. The schematic step involves importing

Auto layout and auto synthesis steps involve opening all sheets of the circuit and then auto instantiating them in a new layout. Then autofp the layout, generate the standard cells, edit some port settings and set up routing preferences. This exercise succeeded when the area ratio was adjusted to .8. Power routing was done first to make the regular routing process simpler. The layouts were properly generated and ports were added to them. Then LVS was run for the 1 bit and the 16 bit ALUs. These both passed LVS successfully and were ready to simulate. PEX was used to extract netlist data for the 1 and 16 bit ALUs. Eldo was used to do circuit timing and power analysis and the results were shown in ezwave. Power was measure staticky and dynamically and the code to do such measurements are shown in the 1 and 16 bit layout test .cir files. This is discussed with results in the analysis section.

# RESULTS & ANALYSIS

Test benches written in VHDL were used to validate the 1 and 16 bit alu operation. Shown in figure 1 is the VHDL testbench of the 1 bit ALU.

**Fig. 1: 1 bit ALU VHDL Functional Simulation**

Figure 1 will serve as the baseline operation and expected output of the layout. The 16 bit alu was also simulated and is shown in figure 2.

**Fig. 2: 16 bit ALU VHDL Functional Simulation**

By using the same VHDL code, schematics could be generated for the hardware. The schematics are generated and shown in Pyxis schematic. Figure 3 shows the generated schematic for the 1 bit ALU. All gates fit onto one sheet.

**Fig. 3: 1 bit ALU Generated Schematic**

Vhdl tb and func verification

Show shematics

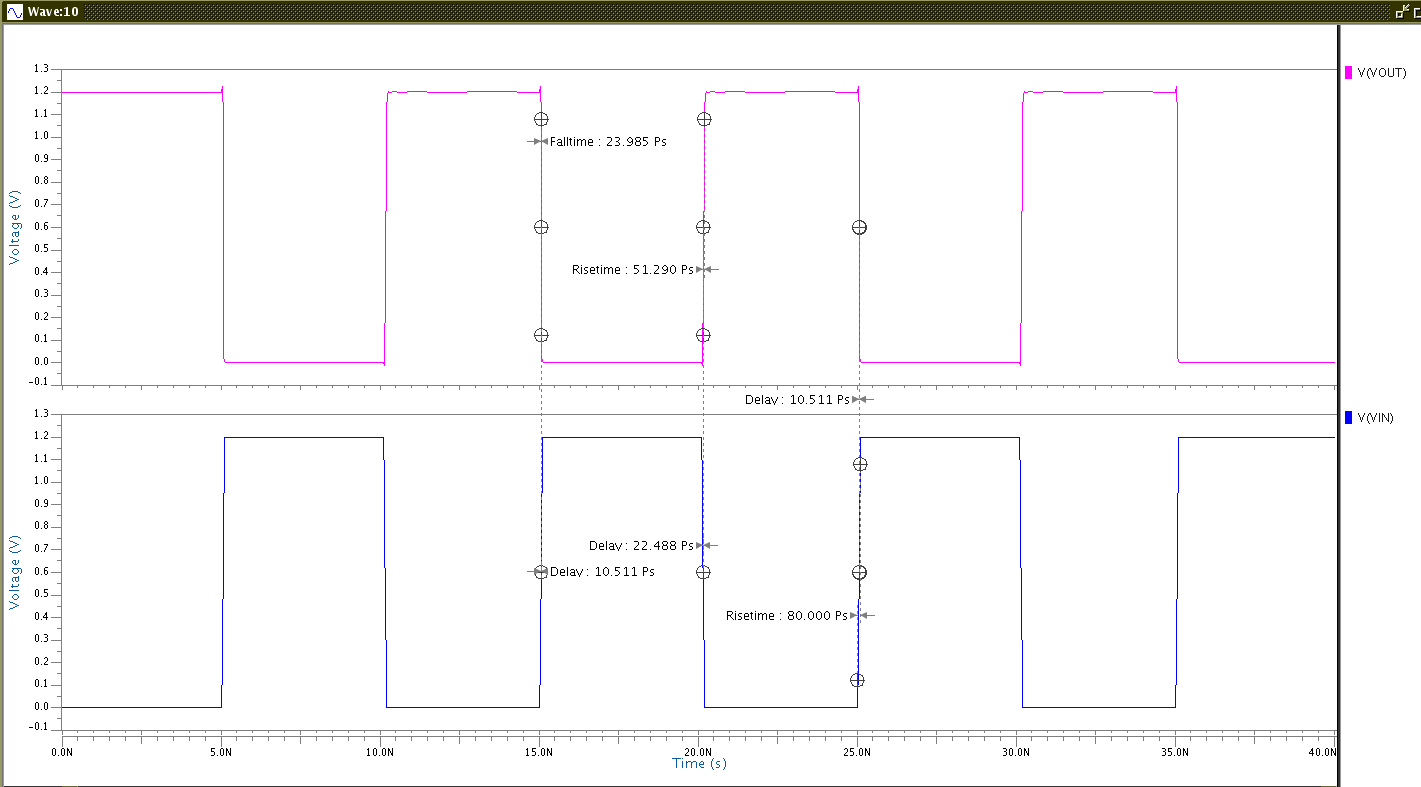
Show generated layouts

Simulate with eldo and show the simulations

Then show the measurments

Determine worstcase opcode for each

Talk about measuring power



**Fig. 5:** Plotted CMOS Inverter Vin and Vout with 40ns transient analysis with profile (T = 25 degrees C, 1.2V Pulse on Vin swept)

Table 1: Schematic Inverter Results

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **VDD [V]** | **Temp [°C]** | **Load**  **Capacitance [fF]** | **Rise Time[S]** | **Fall Time[S]** | **TP,HL [S]** | **TP,LH [S]** |
| Nominal | 1.2 | 25 | 0 | 5.13E-11 | 2.40E-11 | 2.25E-11 | 1.05E-11 |
| Nominal | 1.2 | 25 | 120 | 1.74E-09 | 9.08E-10 | 4.65E-10 | 8.15E-10 |
| Worst Case | 1.08 | 125 | 0 | 5.39E-11 | 4.26E-11 | 2.33E-11 | 1.28E-11 |
| Worst Case | 1.08 | 125 | 120 | 2.04E-09 | 1.32E-09 | 9.58E-10 | 6.03E-10 |

Equations 1 and 2 are the method of solving for these critical frequencies.

**Equation 1.** F Input Max calculation

**Equation 2.** F throughput max calculation

These equations were used to generate the worst case frequency information in table 3.

Table 3: Inverter Frequencies

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Type** | **VDD [V]** | **Temp [°C]** | **Load**  **Capacitance [fF]** | **Finput, max [Hz]** | **FThroughput, Max [Hz]** |
| Schematic | 1.08 | 125 | 0 | 1.037E+10 | 2.771E+10 |
| Schematic | 1.08 | 125 | 120 | 2.976E+08 | 6.404E+08 |
| Layout | 1.08 | 125 | 0 | 8.331E+09 | 1.647E+10 |
| Layout | 1.08 | 125 | 120 | 2.994E+08 | 6.349E+08 |

# CONCLUSIONS

Overall this exercise was a success in demonstrating auto layouts for an ALU. However, the circuit generated could be improved about by optimizing the layout process. This sort of improvement comes from optimizing layout area and other parts of the circuit. This exercise involved using VHDL code and Pyxis Layout to generate a physical design of an n-bit ALU. The circuit was only described in the hardware description language. The procedure involved writing VHDL for a 1 bit and 16 bit ALU with 4 opcodes. The VHDL was functionally tested and the code was then used in Pyxis to create a layout that was then tested in eldo and viewed with Ezwave. There were many cases used to determine the performance of the 1 and 16 bit ALU’s. Overall the 1 bit ALU had a maximum input frequency of .372GHz and a max throughput frequency of .459 Ghz and the 16 bit had a maximum input frequency of .506Mhz and a maximum throughput frequency of .195Ghz. This exercise was based on the process of generating circuits based on VHDL with auto layout tools. Static and dynamic power was also measured for the auto routed circuits. The static power and dynamic power of a 1 bit ALU are 1999 nW AND 1470 uW. The static and dynamic power for the 16 bit ALU are 50834 nW and . It ended up a success as the process to go from a HDL file to an auto generated layout now routine and repeatable.

# QUESTIONS

* Generate a 4-bit version of your ALU and compare the ADD function performance to your 4-bit mirror adder. How much faster/slower is the ALU (use percentages toillustrate differences)? Why?

# APPENDIX

N bit alu

Spectrum file

Put in netlists and talk about extraction

Circuit files layout\_test.cir

# SOURCES

Rit CMPE Digital IC design lab 7 manual, RIT