

Cowgirl Platform :: Component Tests

thebeekeeper

March 2006

1 Overview

OK! I've got most of the hardware elements written, now I just have to make sure they work.

Contents

1 Overview	1
2 Program Counter : 100%	2
2.1 ROM	2
3 Control : 60%	3
4 Registers : 100%	4
4.1 Writing to Registers	4
4.2 Reading From Registers	4
5 ALU : 80%	5
5.1 Arithmetic	5
5.2 Logical	5
6 Shifter : 5%	6
7 System : 50%	7

2 Program Counter : 100%

This one ought to be pretty simple.

Code	Complete
Testing	Working

Table 1: Program Counter Status

So it's looking pretty good.

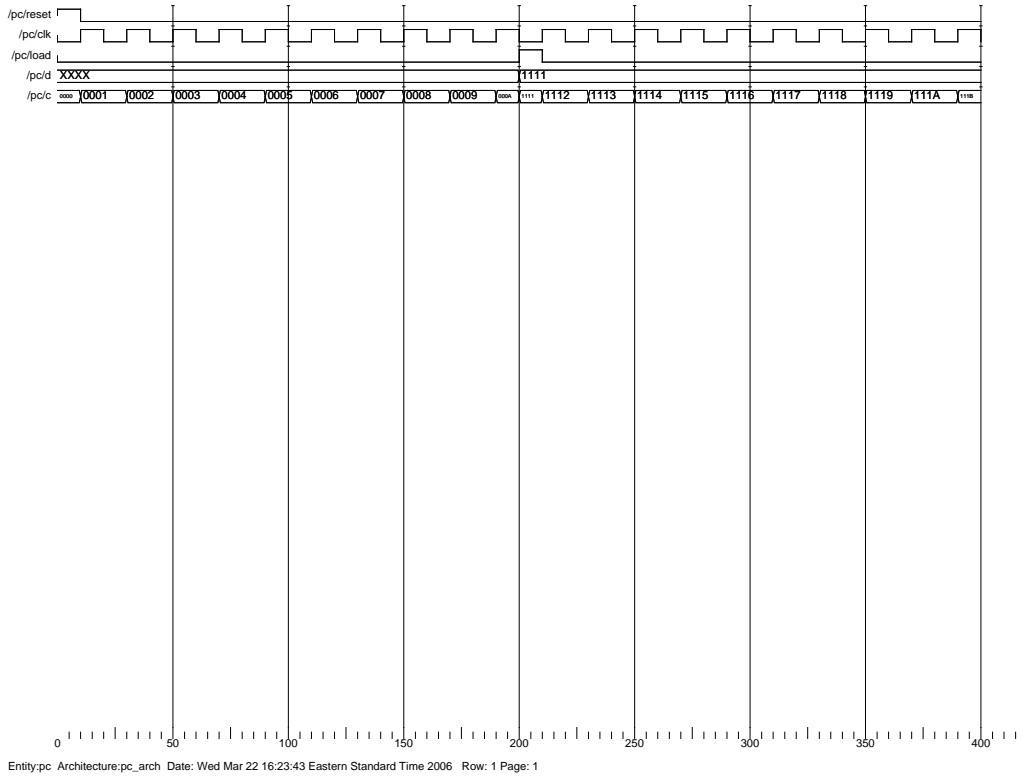


Figure 1: Program Counter Simulation Trace

2.1 ROM

This is working better than anything

3 Control : 60%

What a mess

4 Registers : 100%

I can't get the value to be updated on the rising edge of the clock. It always happens on the falling edge.

OK, so when I test a single register, it works as intended. AKA, the value in the register gets set on the rising edge of the clock. Actually, it is transitioning on the rising edge, I was just setting the value in my do file at the falling edge. Oops! So the registers seem to work correctly!

4.1 Writing to Registers

When **wr_en** is high, the value present at **d** is written to the register specified by **addr_a**. This operation takes place at the rising edge of the clock.

4.2 Reading From Registers

There are two registers that can be read simultaneously, a and b. These values are always present at the output ports. The value at **a_o** is the value that is stored in register a, addressed by **addr_a**, and the value at **b_o** is the value that is stored in register b, addressed by **addr_b**.

5 ALU : 80%

5.1 Arithmetic

Why won't it add? Ok, I added the input signals to the sensitivity list in arithmetic.vhdl, and now it adds just fine. I haven't put the divide operation in yet, since I don't think it exists in std_logic_arith. I haven't completely tested multiplication yet. The case I put in sim_alu.do overflows.

5.2 Logical

Works, 100%! w00t w00t!

6 Shifter : 5%

Not yet implemented.

7 System : 50%

Yikes