

Control ROM

By Brian and Cody

Why is the component important?

- It drives the various operations of the processor
- It controls which components are enabled based on the instruction
- It takes the instruction sent from the instruction memory and uses it to direct all other components to accomplish the task

What instructions
does the
component
support

- All of them
- ADD
- AND
- MOVL
- MOVS
- CMP
- JA
- JALR
- JALR 1 1
- This component is required for instructions to function because this is where instructions are processed

Implementation

- This component uses data flow style.
- Using data flow allows values to be assigned for the output based on the output from the decoder which is a set instruction
- We used conditional statements and the assign statement to assign the values

Test Cases

```

1 module CONTROLROMTEST();
2
3 reg [63:0] in;
4 wire [6:0] out;
5
6 CONTROLROM DUT(in, out);
7
8
9
10
11
12
13 initial
14 begin
15 $display("AND Input: 0000000000000000000000000000000000000000000000000000000000000000");
16 in <= 64'b0000000000000000000000000000000000000000000000000000000000000000;
17 #1000
18 $display("Output: %d%d%d%d%d%d", out[6], out[5], out[4], out[3], out[2], out[1], out[0]);
19
20 $display("ADD Input: 0000100000000000000000000000000000000000000000000000000000000000");
21 in <= 64'b0000100000000000000000000000000000000000000000000000000000000000;
22 #1000
23 $display("Output: %d%d%d%d%d%d", out[6], out[5], out[4], out[3], out[2], out[1], out[0]);
24
25 $display("MOVL Input: 0000000000010000000000000000000000000000000000000000000000000000");
26 in <= 64'b0000000000010000000000000000000000000000000000000000000000000000;
27 #1000
28 $display("Output: %d%d%d%d%d%d", out[6], out[5], out[4], out[3], out[2], out[1], out[0]);
29
30 $display("MOVS Input: 0000000000001000000000000000000000000000000000000000000000000000");
31 in <= 64'b0000000000001000000000000000000000000000000000000000000000000000;
32 #1000
33 $display("Output: %d%d%d%d%d%d", out[6], out[5], out[4], out[3], out[2], out[1], out[0]);
34
35 $display("CMP Input: 0000000000000000000000000000000000000000000000000000000000000000");
36 in <= 64'b0000000000000000000000000000000000000000000000000000000000000000;
37 #1000
38 $display("Output: %d%d%d%d%d%d", out[6], out[5], out[4], out[3], out[2], out[1], out[0]);
39
40 $display("JA Input: 0000000000000000100000000000000000000000000000000000000000000000");
41 in <= 64'b0000000000000000010000000000000000000000000000000000000000000000;
42 #1000
43 $display("Output: %d%d%d%d%d%d", out[6], out[5], out[4], out[3], out[2], out[1], out[0]);
44
45
46
47
48
49 end
50
51 endmodule

```

Test Cases

[illegible]

Issues

- We had to learn the syntax for conditional statements, binary numbers, and the assign block
- We had to finish the decoder to get the values for the conditional statements

Gate Delay

- Structurally the Control ROM is seven NOR gates working in parallel. Each NOR has a delay of 2 and since the gates do not rely on each other the total delay for the Control ROM is 2.

The background features a series of thin, curved lines in shades of gray, creating a sense of motion and depth. These lines are more prominent on the left side of the image and fade towards the right.

The End