EE 443
Digital and Computer Analysis and Design Laboratory

Lab #7: *Completion and summary of components*
8/18/13

Prepared by:
Matt Van Veldhuizen

Lab Participation
Provide the percent participation in lab by each lab partner (normally, 50%, 50% for 2 partners):

Lab member name	Percent participation
Matt Van Veldhuizen	100%

Does your solution work the way it's supposed to work? ☐YES ☐NO[1]

[1]If your answer is NO, please explain in your report.

Instructor/TA comments and grading:

1. Objective and Background

The goal of this lab was to complete the last of the components necessary for the 16-bit RISC microprocessor, these last components where the last of the multiplexers needed for the control path. In addition to completing the last of the components, a table was created that describes each component and some of the properties each component had.

2. Equipment

Altera Quartus II

3. Procedure

The first set I did was complete the last of the components necessary for the microprocessor. These components included a 2-input 16-bit multiplexer called MUX2x16, which can be seen in Figure 1. A 2-input 3-bit multiplexer called MUX2x3, which can be seen in Figure 2. Lastly a component that combines the 3 most significant bits from the PC counter with the 12 bits of the jump address and a '0'. This last component was call PCJMP which can be seen in Figure 3. After the last of the components were created a table was created that described each component, what other components were used in making each component, the number of logic cells used, the percentage of the chip used, and the propagation delay range. This information can be seen in Table 1. The last step in the lab was to take an figure from the book (Figure 4.17), and annotate that figure, by labeling each of the components with the component name of our microprocessor. This can be seen in Figure 4.

4. Results

The component MUX2x16 had a max delay of 11.075ns, which can be seen in Table 2 and used less than 1% of the total logic elements and 11% of the total pins of the chip resources which can be seen in Figure 5. The simulation for MUX2x16 can be seen in Figure 6 The component MUX2x3 had a max delay of 9.474ns, which can be seen in Table 3 and used less than 1% of the total logic elements and 2% of the total pins of the chip resources, which can be seen in Figure 7. The simulation for MUX2x3 can be seen in Figure 8. The component PCJMP had a max delay of 8.918ns, which can be seen in Table 4 and used 0% of the total logic elements and 9% of the total pins of the chip resources, which can be seen in Figure 9. The simulation for PCJMP can be seen in Figure 10.

5. Discussion and Questions

This lab was pretty straightforward and simple. The last 2 multiplexers needed were simple to create. I did this by simply modifying the other multiplexers needed. The same goes for the PCJMP component. Making the component information table was probably the hardest part, only because I had to find all of the minimum propagation delay for each component. Other than that there were no design changes, other than some of the variable names for some of the components.

6. Conclusion

This lab was to complete the last of the components required to make the 16-bit RISC microprocessor. In addition to this a table was created that described each of the components that were created and displayed some basic information about them.

7. Attachments

```
1   library ieee;
2   use ieee.std_logic_1164.all;
3
4   entity MUX2x16 is
5   port(S : in std_logic_vector(0 downto 0);
6        X,Y : in std_logic_vector(15 downto 0);
7        O : out std_logic_vector(15 downto 0));
8   end;
9
10  architecture Logic of MUX2x16 is
11  begin
12  with S select O <= X when "0",
13                     Y when "1",
14                     "XXXXXXXXXXXXXXXX" when others;
15
16  end Logic;
```

Figure 1: MUX2x16.vhdl

```
1   library ieee;
2   use ieee.std_logic_1164.all;
3
4   entity MUX2x3 is
5   port(S : in std_logic_vector(0 downto 0);
6        X,Y : in std_logic_vector(2 downto 0);
7        O : out std_logic_vector(2 downto 0));
8   end;
9
10  architecture Logic of MUX2x3 is
11  begin
12  with S select O <= X when "0",
13                     Y when "1",
14                     "XXX" when others;
15
16  end Logic;
```

Figure 2: MIX2x3.vhdl

```
1   library ieee;
2   use ieee.std_logic_1164.all;
3
4   entity PCJMP is
5   port( PC_ADD   : in std_logic_vector(15 downto 0);
6         JMP_ADD  : in std_logic_vector(11 downto 0);
7         OUT_ADD  : out std_logic_vector(15 downto 0));
8   end;
9
10  architecture Logic of PCJMP is
11  begin
12     OUT_ADD <= PC_ADD(15 downto 13) & JMP_ADD & '0';
13  end Logic;
```

Figure 3: PCJMP.vhdl

Table 1: MIPS-16 Components

VHDL Filename	Description	Components	Devices	Number of Logic Cells used	Presentage of chip used (%)	Propagation Delay Range (ns)

DCD3x8	3-to-8 positive-assert decoder with gate enable	none	Cyclone II EP2C35F672C6	8	< 1	5.774 - 11.087
MUX4x4	4-input 4-bit multiplexer	none	Cyclone II EP2C35F672C6	8	< 1	6.967 - 13.513
REG8	8-bit register component, with 8-bit input, output, 1-bit enable, and a clock	none	Cyclone II EP2C35F672C6	8	< 1	
REG8x16	8-by-16 register file (8 16-bit registers) with one registered input port (write) and two combination output ports (read)	REG8, MUX4x4, DCD3x8	Cyclone II EP2C35F672C6	168	< 1	9.789 - 13.885
ADD4	4-bit adder based on a commercial Binary Adder with Fast Carry	none	Cyclone II EP2C35F672C6	11	< 1	5.946 - 11.216
ALU4	4-bit ALU using a multiplexer approach with 4-bit inputs(1st & 2nd Register), 1-bit inputs(less, cin), and 3-bit select input. 3-bit output(3rd Register), 1-bit output(cout, overflow, set and zero)	ADD4, BWOR4, BWAND4, PINV4, MUX4x4	Cyclone II EP2C35F672C6	35	< 1	7.581 - 15.857
ALU16	Uses 4 ALU4s with 16-bit inputs(1st & 2nd Register), 3-bit select input. 16-bit output(3rd Register), 1-bit output(cout, overflow, zero)	ALU4	Cyclone II EP2C35F672C6	123	< 1	6.814 - 15.619
BWAND4	4-bit bitwise AND	none	Cyclone II EP2C35F672C6	4	< 1	5.133 - 9.435
BWOR4	4-bit bitwise OR	none	Cyclone II EP2C35F672C6	4	< 1	5.132 - 9.460
PINV4	4-bit programmable inverter	none	Cyclone II EP2C35F672C6	4	< 1	5.339 - 9.461
ADD16	Adds two 16-bit numbers, using ADD4	ADD4	Cyclone II EP2C35F672C6	37	< 1	6.580 - 25.065
DAT_MEM	Data memory, 16-bit inputs(Address and Value), 1-bit input(write & read enable), 16-bit output(value)		Cyclone II EP2C35F672C6	0	0	8.851 - 9.988
INCTWO	Adds 2 to a 16-bit number	ADD16	Cyclone II EP2C35F672C6	19	< 1	4.727 - 13.126
INS_MEM	Instruction memory	none	Cyclone II EP2C35F672C6	12	< 1	9.130 - 10.399
MUX3x16	3-input 16-bit multiplexer for jump instructions	none	Cyclone II EP2C35F672C6	32	< 1	9.171 - 11.890
SGNEXT6x16	sign extends a 6-bit number to a 16-bit number	none	Cyclone II EP2C35F672C6	0	0	4.802 - 9.135

SHLONE	shifts 16-bit number left by one bit	none	Cyclone II EP2C35F672C6	0	0	4.802 - 8.918
MUX2x16	2-input 16-bit multiplexer for jump instructions	none	Cyclone II EP2C35F672C6	16	< 1	6.691 - 11.075
MUX2x3	2-input 3-bit multiplexer for jump instructions	none	Cyclone II EP2C35F672C6	3	< 1	5.838 - 9.474
PCJMP	combines 3 MSB of PC with 12-bits of jmp address and '0'	none	Cyclone II EP2C35F672C6	0	0	7.926 - 8.918

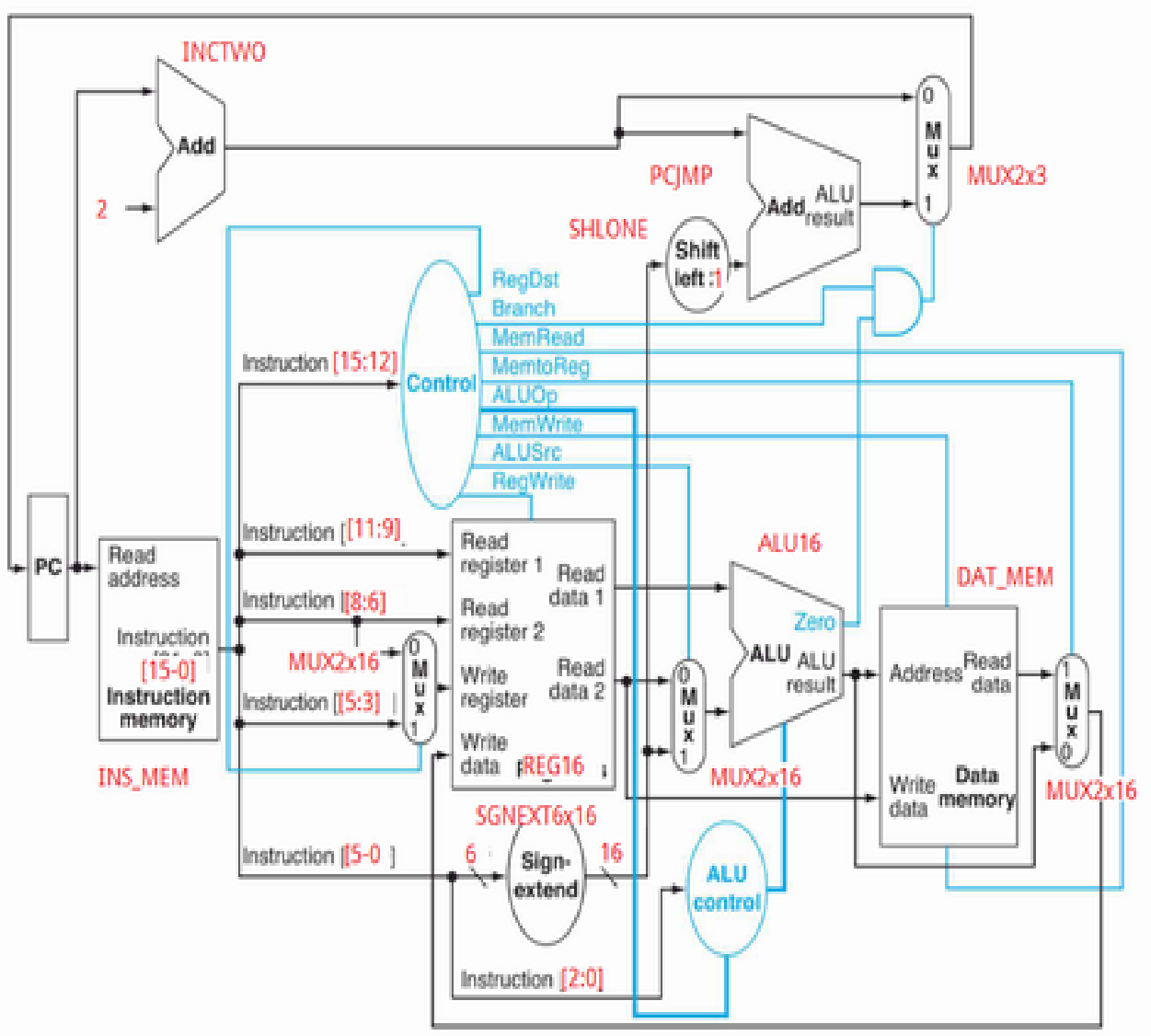

Figure 4: Processor Layout

Table 2: MUX2x16 Propagation Delay

	Input Port	Output Port	RR	RF	FR	FF
1	S[0]	O[3]	11.075	11.075	11.075	11.075
2	S[0]	O[9]	10.958	10.958	10.958	10.958
3	S[0]	O[0]	10.738	10.738	10.738	10.738
4	S[0]	O[13]	10.656	10.656	10.656	10.656
5	S[0]	O[12]	10.500	10.500	10.500	10.500
6	S[0]	O[11]	10.399	10.399	10.399	10.399
7	S[0]	O[10]	10.395	10.395	10.395	10.395
8	S[0]	O[15]	10.376	10.376	10.376	10.376
9	Y[1]	O[1]	10.374			10.374
10	S[0]	O[7]	10.309	10.309	10.309	10.309
11	Y[9]	O[9]	10.289			10.289
12	S[0]	O[5]	10.255	10.255	10.255	10.255
13	S[0]	O[14]	10.234	10.234	10.234	10.234
14	S[0]	O[8]	10.232	10.232	10.232	10.232
15	Y[12]	O[12]	10.231			10.231
16	S[0]	O[2]	10.219	10.219	10.219	10.219
17	S[0]	O[6]	10.207	10.207	10.207	10.207
18	X[9]	O[9]	10.022			10.022
19	X[3]	O[3]	9.844			9.844
20	Y[13]	O[13]	9.783			9.783
21	X[13]	O[13]	9.779			9.779
22	X[12]	O[12]	9.723			9.723

Total logic elements	16 / 33,216 (< 1 %)
Total combinational functions	16 / 33,216 (< 1 %)
Dedicated logic registers	0 / 33,216 (0 %)
Total registers	0
Total pins	50 / 475 (11 %)
Total virtual pins	0
Total memory bits	0 / 483,840 (0 %)
Embedded Multiplier 9-bit elements	0 / 70 (0 %)
Total PLLs	0 / 4 (0 %)

Figure 5: MUX2x16 Chip Resources

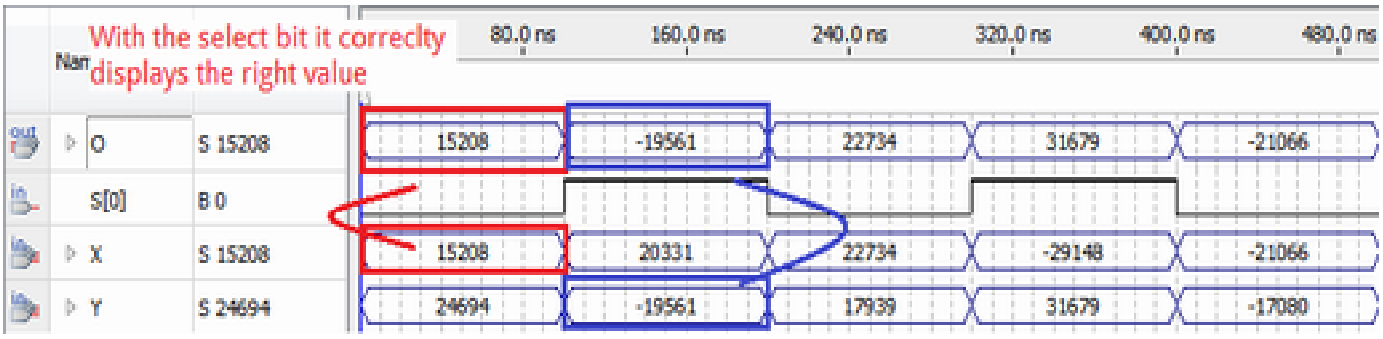

Figure 6: MUX2x16 Simulation Results

Table 3: MUX2x3 Delay

	Input Port	Output Port	RR	RF	FR	FF
1	S[0]	O[1]	9.474	9.474	9.474	9.474
2	X[2]	O[2]	9.422			9.422
3	Y[1]	O[1]	9.413			9.413
4	X[1]	O[1]	9.289			9.289
5	S[0]	O[2]	9.276	9.276	9.276	9.276
6	S[0]	O[0]	9.256	9.256	9.256	9.256
7	Y[2]	O[2]	9.135			9.135
8	X[0]	O[0]	6.194			6.194
9	Y[0]	O[0]	5.838			5.838

Total logic elements	3 / 33,216 (< 1 %)
Total combinational functions	3 / 33,216 (< 1 %)
Dedicated logic registers	0 / 33,216 (0 %)
Total registers	0
Total pins	11 / 475 (2 %)
Total virtual pins	0
Total memory bits	0 / 483,840 (0 %)
Embedded Multiplier 9-bit elements	0 / 70 (0 %)
Total PLLs	0 / 4 (0 %)

Figure 7: MUX2x3 Chip Resources

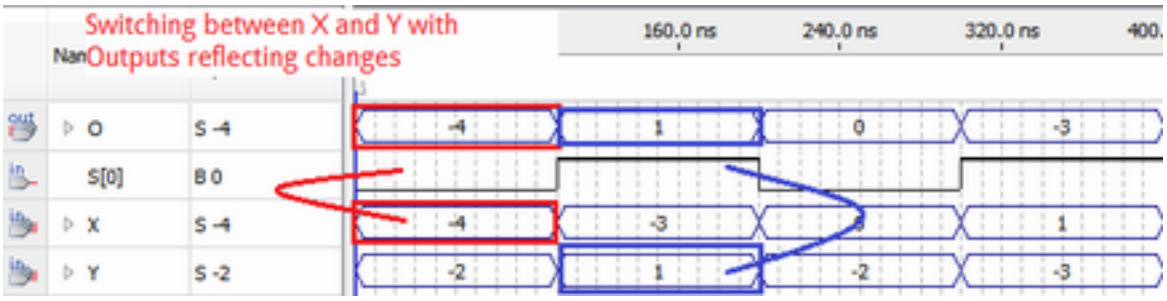

Figure 8: MUX2x3 Simulation Results

Table 4: PCJMP Delay

	Input Port	Output Port	RR	RF	FR	FF
1	JMP_ADD[1]	OUT_ADD[2]	8.918			8.918
2	JMP_ADD[2]	OUT_ADD[3]	8.905			8.905
3	JMP_ADD[0]	OUT_ADD[1]	8.859			8.859
4	PC_ADD[15]	OUT_ADD[15]	8.835			8.835
5	JMP_ADD[9]	OUT_ADD[10]	8.823			8.823
6	JMP_ADD[3]	OUT_ADD[4]	8.807			8.807
7	JMP_ADD[4]	OUT_ADD[5]	8.801			8.801
8	JMP_ADD[10]	OUT_ADD[11]	8.588			8.588
9	JMP_ADD[6]	OUT_ADD[7]	8.170			8.170
10	JMP_ADD[5]	OUT_ADD[6]	8.140			8.140
11	JMP_ADD[11]	OUT_ADD[12]	8.010			8.010
12	PC_ADD[13]	OUT_ADD[13]	7.980			7.980
13	JMP_ADD[7]	OUT_ADD[8]	7.944			7.944
14	JMP_ADD[8]	OUT_ADD[9]	7.926			7.926
15	PC_ADD[14]	OUT_ADD[14]	7.926			7.926

Total logic elements	0 / 33,216 (0 %)
Total combinational functions	0 / 33,216 (0 %)
Dedicated logic registers	0 / 33,216 (0 %)
Total registers	0
Total pins	44 / 475 (9 %)
Total virtual pins	0
Total memory bits	0 / 483,840 (0 %)
Embedded Multiplier 9-bit elements	0 / 70 (0 %)
Total PLLs	0 / 4 (0 %)

Figure 9: PCJMP Chip Resources

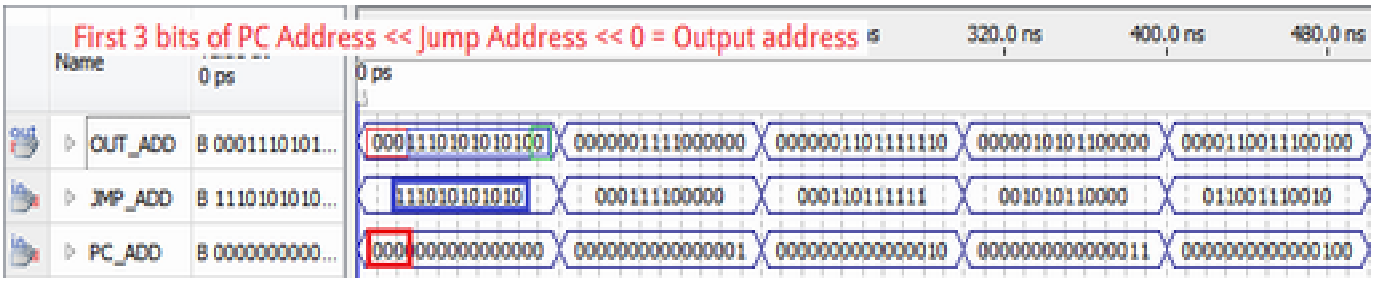

Figure 10: PCJMP Simulation Results