Computer Architecture & Design

Coen 316

Lab Experiment #4 (part 1 and part 2)

Andre Hei Wang Law

4017 5600

Section DN-X

Professor Dr. Fadi Alzouri

Performed on November 9, 2023

Due on November 23, 2023

“I certify that this submission is my original work and meets the Faculty’s Expectations of Originality.”

1) Introduction/Objectives

In the last Computer Architecture & Design COEN 316 lab, students worked on developing a functional CPU using previous lab’s knowledge as well as other components. For example, some added crucial components are a Control Unit, I-Cache, D-Cache, Sign Extended, etc. With these, the main objective is to design, implemented and test the interactions and functionalities of a CPU.

2) Theory

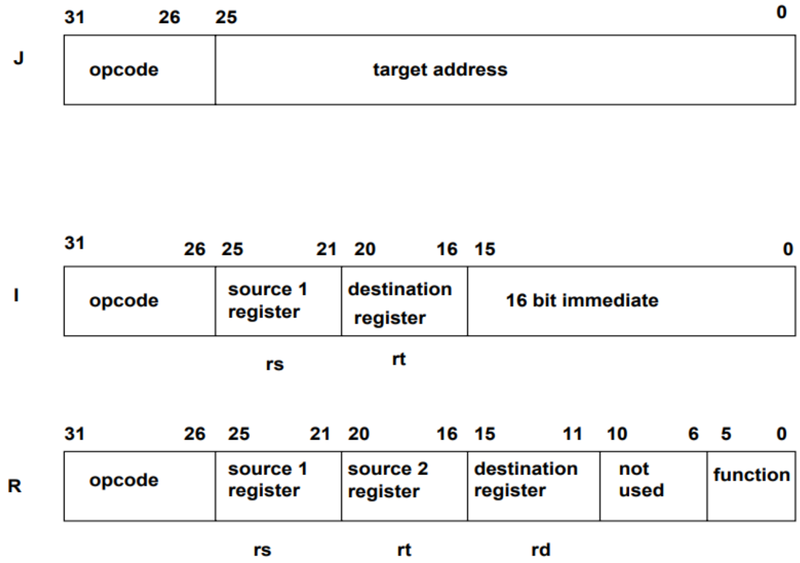
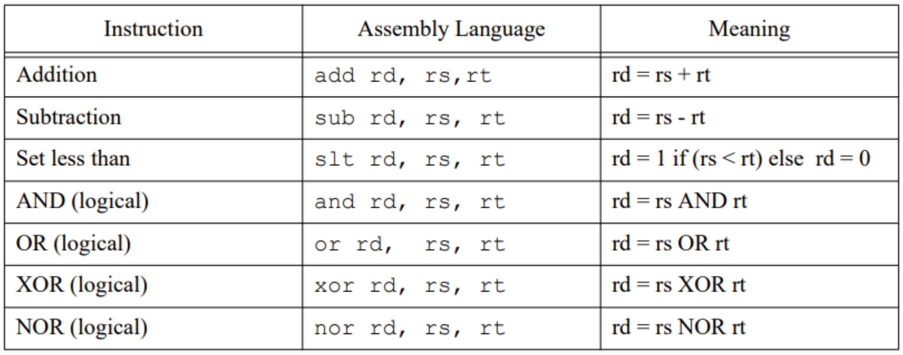


Figure 1: Instruction formats for R, I, and J type of instructions. [1]

Figure 1 is the MIPS processor instructions where J is the Jump Instruction, I is the Immediate Instruction and R is the Register Instruction.

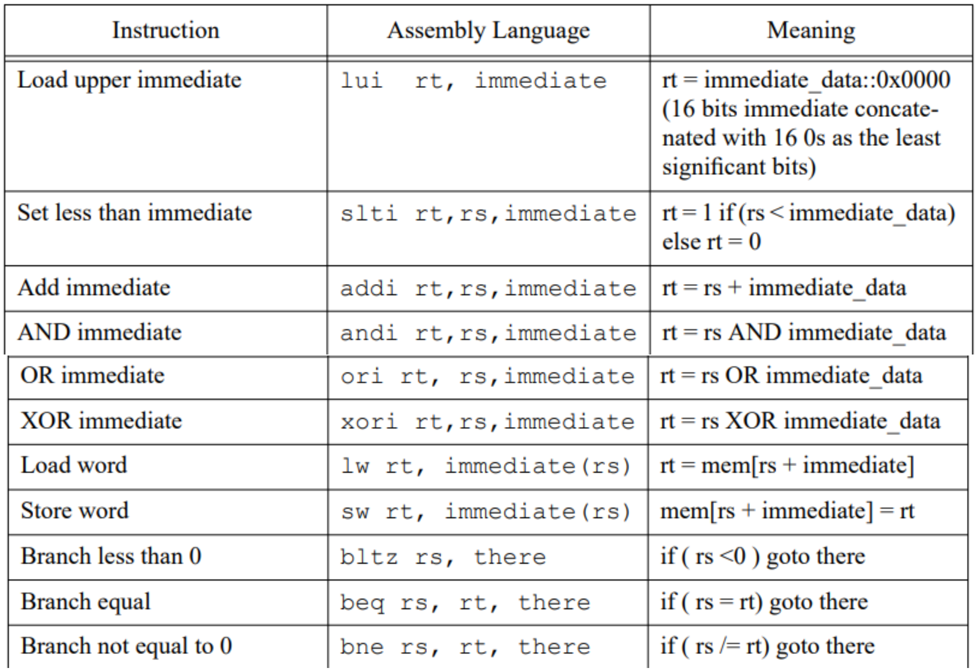
Table 1: Register instructions [2]



Instruction	Assembly Language	Meaning
Addition	`add rd, rs,rt`	rd = rs + rt
Subtraction	`sub rd, rs, rt`	rd = rs - rt
Set less than	`slt rd, rs, rt`	rd = 1 if (rs < rt) else rd = 0
AND (logical)	`and rd, rs, rt`	rd = rs AND rt
OR (logical)	`or rd, rs, rt`	rd = rs OR rt
XOR (logical)	`xor rd, rs, rt`	rd = rs XOR rt
NOR (logical)	`nor rd, rs, rt`	rd = rs NOR rt

Table 1 is Register Instruction information containing the assembly code as well as its meaning. There is a total of 7 of them.

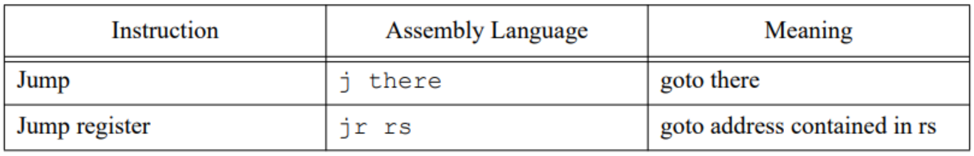
Table 2: Immediate instructions [2]



Instruction	Assembly Language	Meaning
Load upper immediate	`lui rt, immediate`	rt = immediate_data::0x0000 (16 bits immediate concatenated with 16 0s as the least significant bits)
Set less than immediate	`slti rt,rs,immediate`	rt = 1 if (rs < immediate_data) else rt = 0
Add immediate	`addi rt,rs,immediate`	rt = rs + immediate_data
AND immediate	`andi rt,rs,immediate`	rt = rs AND immediate_data
OR immediate	`ori rt, rs,immediate`	rt = rs OR immediate_data
XOR immediate	`xori rt,rs,immediate`	rt = rs XOR immediate_data
Load word	`lw rt, immediate(rs)`	rt = mem[rs + immediate]
Store word	`sw rt, immediate(rs)`	mem[rs + immediate] = rt
Branch less than 0	`bltz rs, there`	if (rs <0) goto there
Branch equal	`beq rs, rt, there`	if (rs = rt) goto there
Branch not equal to 0	`bne rs, rt, there`	if (rs /= rt) goto there

Table 2 is the Immediate Instruction which contains the assembly language code as well as its meaning. There is a total of 11 of them.

Table 3: Unconditional jump instructions [2]



Instruction	Assembly Language	Meaning
Jump	`j there`	goto there
Jump register	`jr rs`	goto address contained in rs

Table 3 is the remaining instruction for the initial 20. These two instructions represent the Jump Instructions and it has an assembly language code and its meaning.

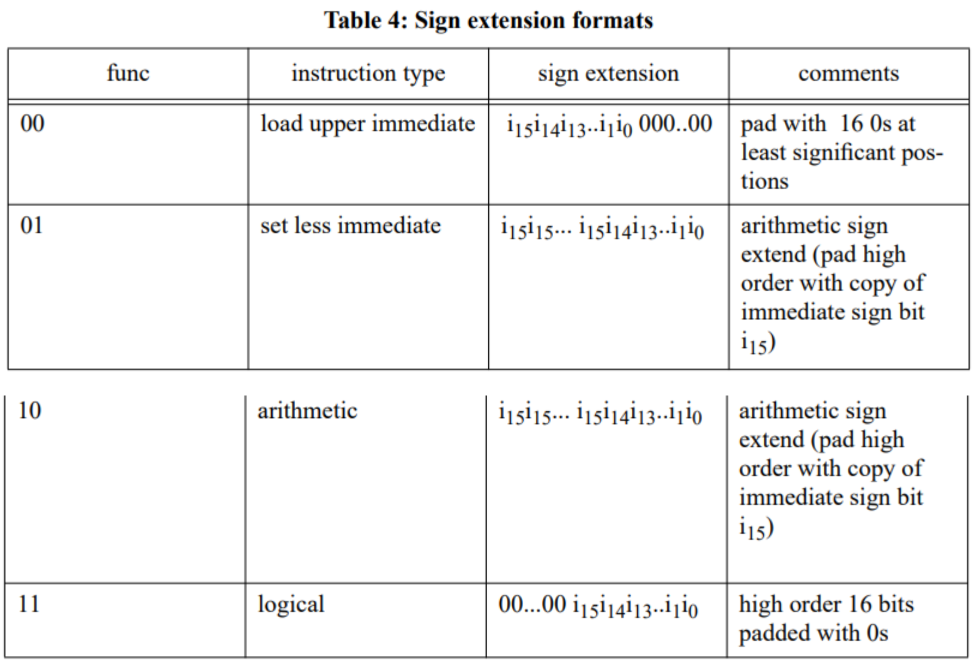


Table 4: Sign extension formats

func	instruction type	sign extension	comments
00	load upper immediate	$i_{15}i_{14}i_{13}..i_1i_0$ 000..00	pad with 16 0s at least significant postions
01	set less immediate	$i_{15}i_{15}$... $i_{15}i_{14}i_{13}..i_1i_0$	arithmetic sign extend (pad high order with copy of immediate sign bit i_{15})
10	arithmetic	$i_{15}i_{15}$... $i_{15}i_{14}i_{13}..i_1i_0$	arithmetic sign extend (pad high order with copy of immediate sign bit i_{15})
11	logical	00...00 $i_{15}i_{14}i_{13}..i_1i_0$	high order 16 bits padded with 0s

The above table 4 represent the four functions for the sign extension component.

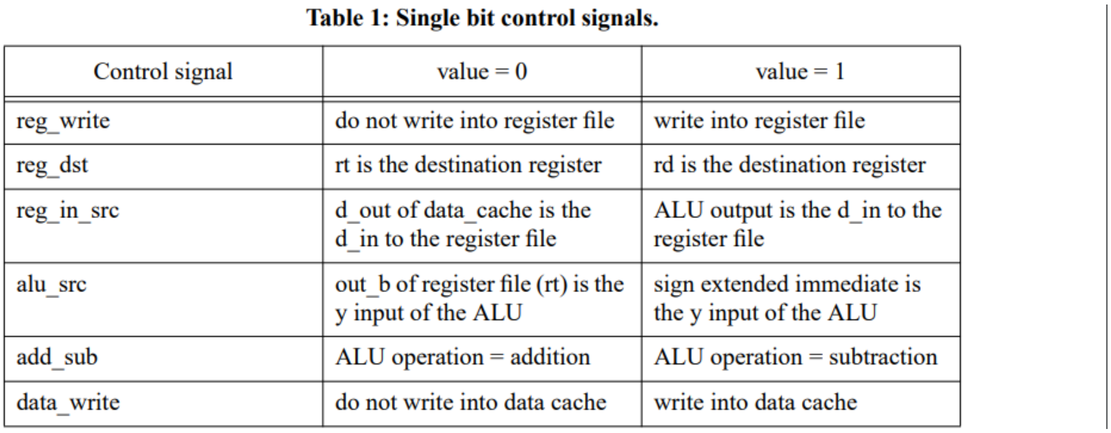


Table 1: Single bit control signals.

Control signal	value = 0	value = 1
reg_write	do not write into register file	write into register file
reg_dst	rt is the destination register	rd is the destination register
reg_in_src	d_out of data_cache is the d_in to the register file	ALU output is the d_in to the register file
alu_src	out_b of register file (rt) is the y input of the ALU	sign extended immediate is the y input of the ALU
add_sub	ALU operation = addition	ALU operation = subtraction
data_write	do not write into data cache	write into data cache

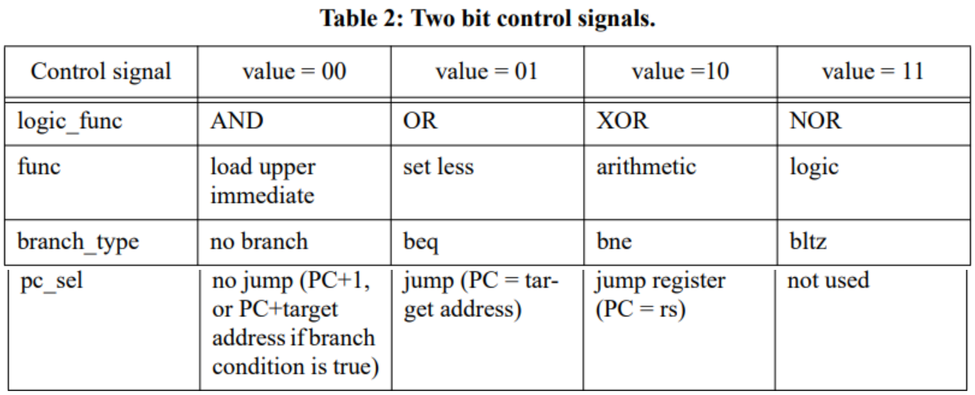


Table 2: Two bit control signals.

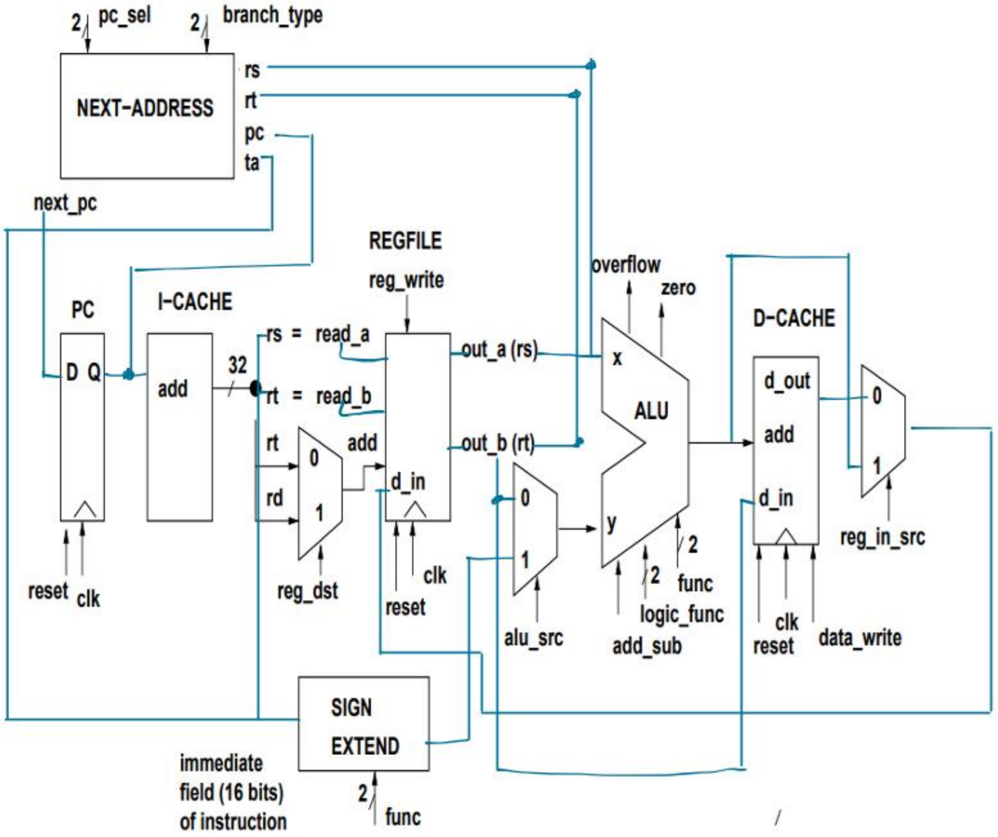
Control signal	value = 00	value = 01	value =10	value = 11
logic_func	AND	OR	XOR	NOR
func	load upper immediate	set less	arithmetic	logic
branch_type	no branch	beq	bne	bltz
pc_sel	no jump (PC+1, or PC+target address if branch condition is true)	jump (PC = target address)	jump register (PC = rs)	not used

Table 1 and 2 belongs to lab 4 part 2 and they represent 10 control signals used by the function of the control unit. It has a summary of their operations.

3) Tasks, Results and Discussion

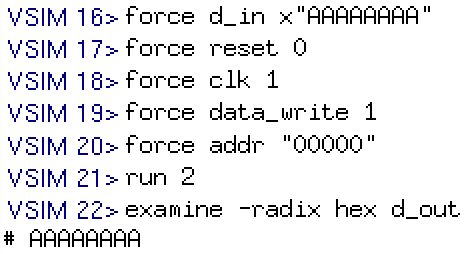
3.1) PART 1: CPU Datapath

CPU Datapath:



3.1.1) Data Cache

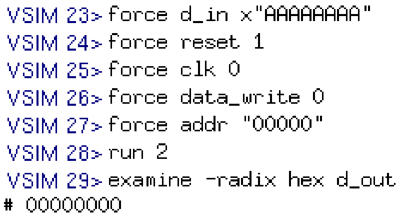
3.1.1.1) Initialize Value



```
VSIM 16> force d_in x"AAAAAAAA"
VSIM 17> force reset 0
VSIM 18> force clk 1
VSIM 19> force data_write 1
VSIM 20> force addr "00000"
VSIM 21> run 2
VSIM 22> examine -radix hex d_out
# AAAAAAAA
```

First step is to populate all values with rising edge clk and data_write 1. Observe d_out output corresponds to d_in of AAAAAAAA.

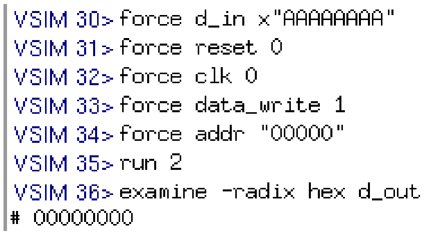
3.1.1.2) reset 1, clk 0 and data_write 0



```
VSIM 23> force d_in x"AAAAAAAA"
VSIM 24> force reset 1
VSIM 25> force clk 0
VSIM 26> force data_write 0
VSIM 27> force addr "00000"
VSIM 28> run 2
VSIM 29> examine -radix hex d_out
# 00000000
```

Testing the reset functionality. Result of 00000000 validates this functionality.

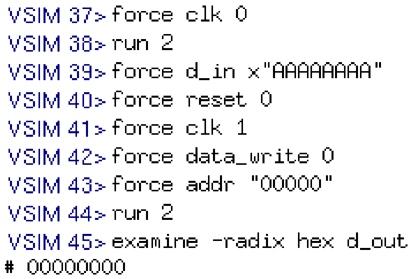
3.1.1.3) data_write 1, reset 0 and clk 0



```
VSIM 30> force d_in x"AAAAAAAA"
VSIM 31> force reset 0
VSIM 32> force clk 0
VSIM 33> force data_write 1
VSIM 34> force addr "00000"
VSIM 35> run 2
VSIM 36> examine -radix hex d_out
# 00000000
```

Testing the data_write functionality without rising edge clk. Result of 00000000 validates this functionality.

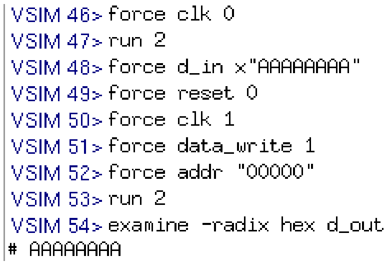
3.1.1.4) data_write 0, reset 0 and clk 1



```
VSIM 37> force clk 0
VSIM 38> run 2
VSIM 39> force d_in x"AAAAAAAA"
VSIM 40> force reset 0
VSIM 41> force clk 1
VSIM 42> force data_write 0
VSIM 43> force addr "00000"
VSIM 44> run 2
VSIM 45> examine -radix hex d_out
# 00000000
```

Testing the clk functionality without data_write. Result of 00000000 validates this functionality.

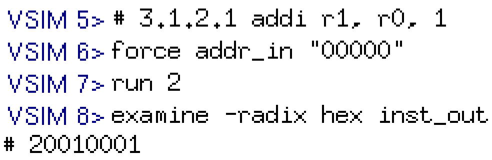
3.1.1.5) data_write 1, reset 0 and clk 1



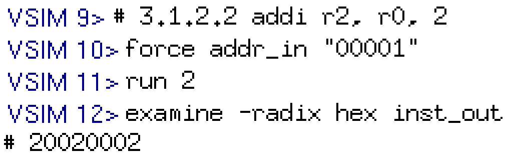
```
VSIM 46> force clk 0
VSIM 47> run 2
VSIM 48> force d_in x"AAAAAAAA"
VSIM 49> force reset 0
VSIM 50> force clk 1
VSIM 51> force data_write 1
VSIM 52> force addr "00000"
VSIM 53> run 2
VSIM 54> examine -radix hex d_out
# AAAAAAAA
```

Testing the clk functionality with data_write. Result of AAAAAAAA validates this functionality.

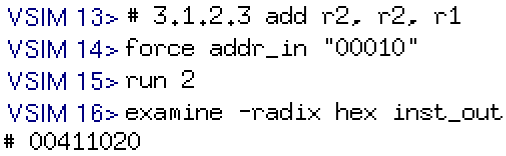
3.1.2) Instruction Cache



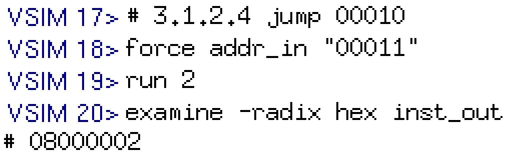
```
VSIM 5> # 3.1.2.1 addi r1, r0, 1
VSIM 6> force addr_in "00000"
VSIM 7> run 2
VSIM 8> examine -radix hex inst_out
# 20010001
```



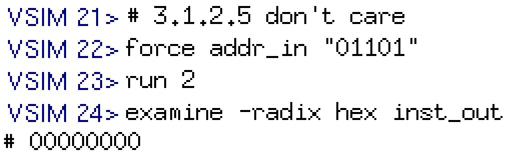
```
VSIM 9> # 3.1.2.2 addi r2, r0, 2
VSIM 10> force addr_in "00001"
VSIM 11> run 2
VSIM 12> examine -radix hex inst_out
# 20020002
```



```
VSIM 13> # 3.1.2.3 add r2, r2, r1
VSIM 14> force addr_in "00010"
VSIM 15> run 2
VSIM 16> examine -radix hex inst_out
# 00411020
```



```
VSIM 17> # 3.1.2.4 jump 00010
VSIM 18> force addr_in "00011"
VSIM 19> run 2
VSIM 20> examine -radix hex inst_out
# 08000002
```

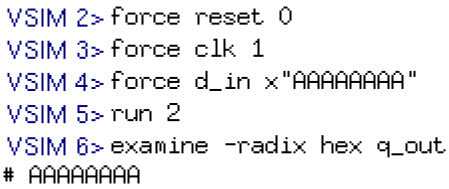


```
VSIM 21> # 3.1.2.5 don't care
VSIM 22> force addr_in "01101"
VSIM 23> run 2
VSIM 24> examine -radix hex inst_out
# 00000000
```

The i_cache.vhd works as intended since when prompted for a address input of 5 bits such as 00000, the output inst_out corresponds to the “hard-wired” program in machine code given in lab 4 part 1 manual.

3.1.3) PC

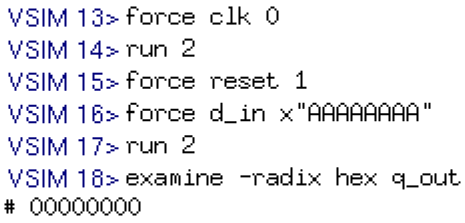
3.1.3.1) Initialize value



```
VSIM 2> force reset 0
VSIM 3> force clk 1
VSIM 4> force d_in x"AAAAAAAA"
VSIM 5> run 2
VSIM 6> examine -radix hex q_out
# AAAAAAAA
```

For pc.vhd, we first setup and initialize the values with these inputs. The current output is AAAAAAAA which corresponds to d_in.

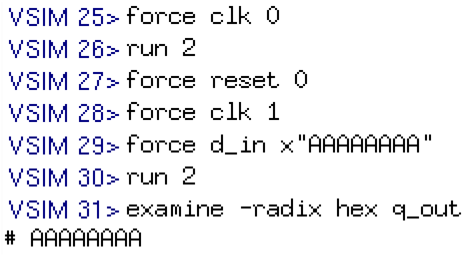
3.1.3.2) reset 1, clk 0



```
VSIM 13> force clk 0
VSIM 14> run 2
VSIM 15> force reset 1
VSIM 16> force d_in x"AAAAAAAA"
VSIM 17> run 2
VSIM 18> examine -radix hex q_out
# 00000000
```

Testing the reset functionality. Given reset 1, the output q_out will become 00000000.

3.1.3.3) reset 0, clk 1

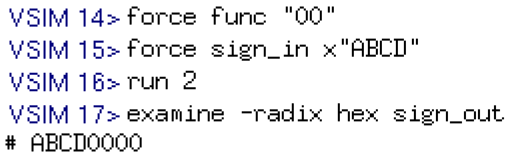


```
VSIM 25> force clk 0
VSIM 26> run 2
VSIM 27> force reset 0
VSIM 28> force clk 1
VSIM 29> force d_in x"AAAAAAAA"
VSIM 30> run 2
VSIM 31> examine -radix hex q_out
# AAAAAAAA
```

Testing the clk functionality. Given reset 0 and clk is a rising edge 1, the output will match d_in.

3.1.4) Sign Extended

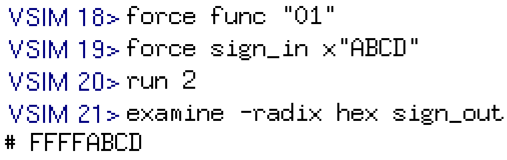
3.1.4.1) load upper immediate



```
VSIM 14> force func "00"
VSIM 15> force sign_in x"ABCD"
VSIM 16> run 2
VSIM 17> examine -radix hex sign_out
# ABCD0000
```

Correct behavior, the first 16 bits of sign_out correspond to sign_in while the rest are zeros.

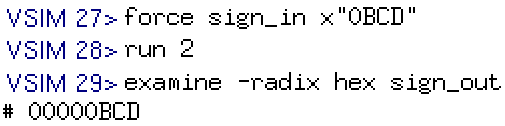
3.1.4.2) arithmetic sign extend (msb = 1)



```
VSIM 18> force func "01"
VSIM 19> force sign_in x"ABCD"
VSIM 20> run 2
VSIM 21> examine -radix hex sign_out
# FFFFABCD
```

Correct behavior, the first 16 bits of sign_out correspond to MSB of sign_in which is 1, so FFFF, while the rest correspond to sign_in ABCD.

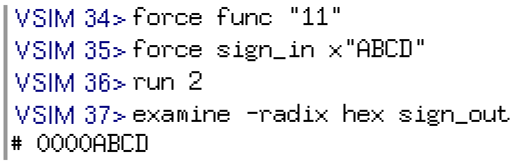
3.1.4.3) arithmetic sign extend (msb = 0)



```
VSIM 27> force sign_in x"0BCD"
VSIM 28> run 2
VSIM 29> examine -radix hex sign_out
# 00000BCD
```

Correct behavior, the first 16 bits of sign_out correspond to MSB of sign_in which is 0, so 0000, while the rest correspond to sign_in ABCD.

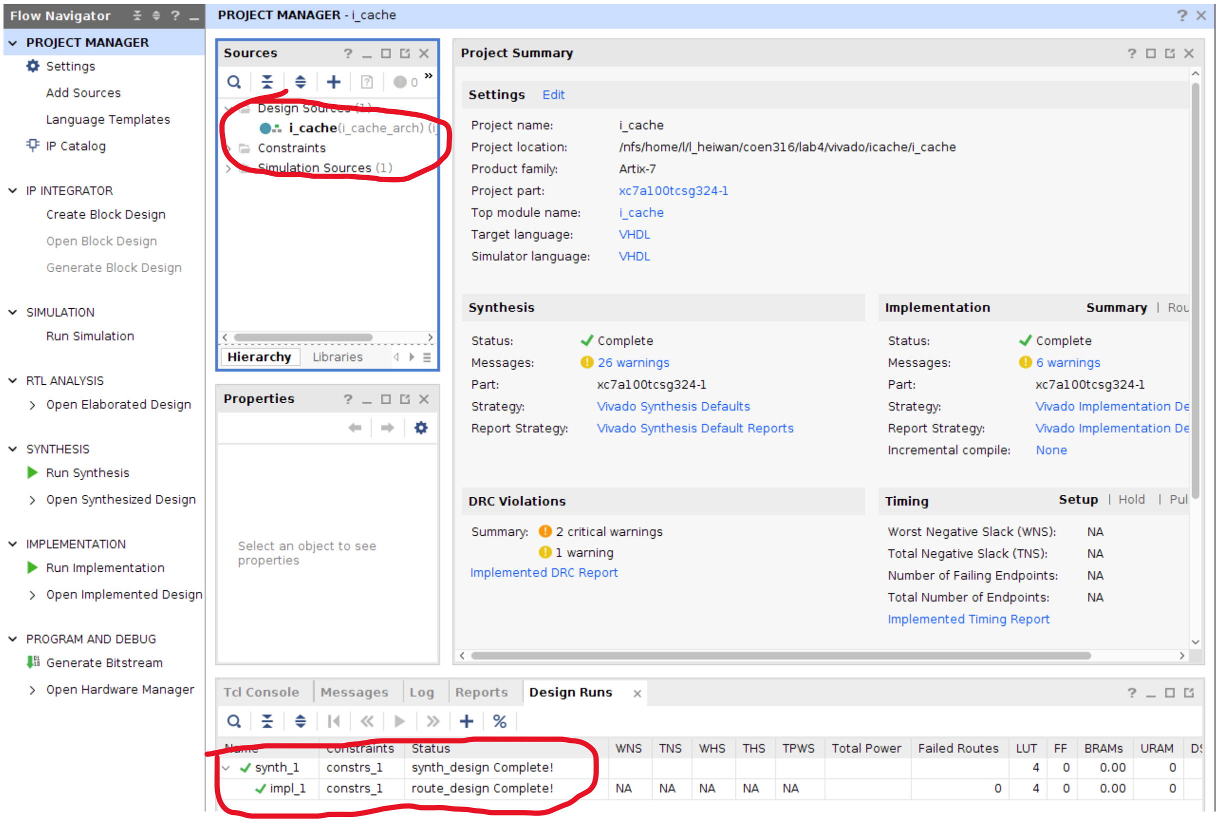
3.1.4.4) logical



```
VSIM 34> force func "11"
VSIM 35> force sign_in x"ABCD"
VSIM 36> run 2
VSIM 37> examine -radix hex sign_out
# 0000ABCD
```

Correct behavior, the first 16 bits of sign_out are all 0, while the rest correspond to sign_in ABCD.

3.1.5) Vivado Synthesis and Implementation (i_cache, d_cache and sign_extended)



Settings Edit

Project name: i_cache
Project location: /nfs/home/l/l_heiwan/coen316/lab4/vivado/icache/i_cache
Product family: Artix-7
Project part: xc7a100tcsg324-1
Top module name: i_cache
Target language: VHDL
Simulator language: VHDL

Synthesis

Status: ✓ Complete
Messages: 26 warnings
Part: xc7a100tcsg324-1
Strategy: Vivado Synthesis Defaults
Report Strategy: Vivado Synthesis Default Reports

Implementation Summary | Rou

Status: ✓ Complete
Messages: 6 warnings
Part: xc7a100tcsg324-1
Strategy: Vivado Implementation De
Report Strategy: Vivado Implementation De
Incremental compile: None

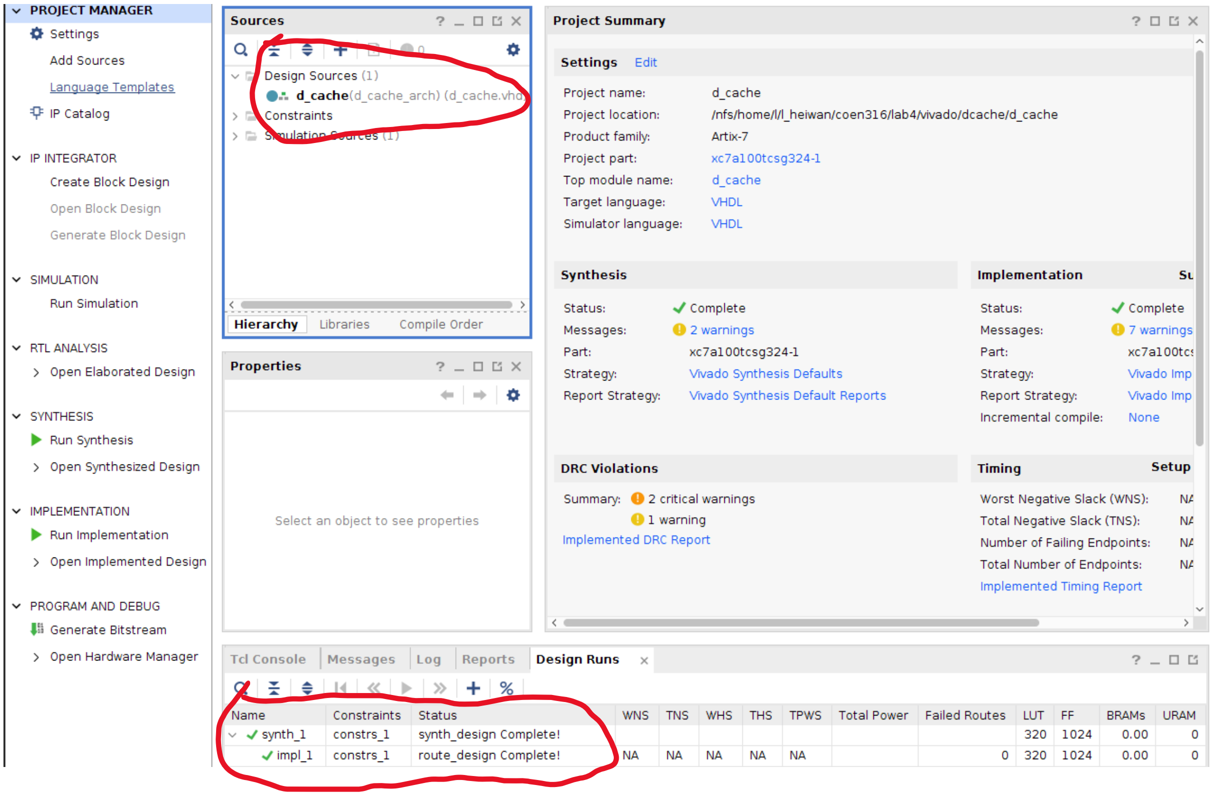
DRC Violations

Summary: 2 critical warnings
1 warning
Implemented DRC Report

Timing Setup | Hold | Pul

Worst Negative Slack (WNS): NA
Total Negative Slack (TNS): NA
Number of Failing Endpoints: NA
Total Number of Endpoints: NA
Implemented Timing Report

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	D
✓ synth_1	constrs_1	synth_design Complete!								4	0	0.00	0	
✓ impl_1	constrs_1	route_design Complete!	NA	NA	NA	NA	NA		0	4	0	0.00	0	



Design Sources (1)
d_cache(d_cache_arch) (d_cache.vhd
Constraints
Simulation Sources (1)

Project Summary

Settings Edit

Project name: d_cache
Project location: /nfs/home/l/l_heiwan/coen316/lab4/vivado/dcache/d_cache
Product family: Artix-7
Project part: xc7a100tcsg324-1
Top module name: d_cache
Target language: VHDL
Simulator language: VHDL

Synthesis

Status: ✓ Complete
Messages: 2 warnings
Part: xc7a100tcsg324-1
Strategy: Vivado Synthesis Defaults
Report Strategy: Vivado Synthesis Default Reports

Implementation

Status: ✓ Complete
Messages: 7 warnings
Part: xc7a100tcs
Strategy: Vivado Imp
Report Strategy: Vivado Imp
Incremental compile: None

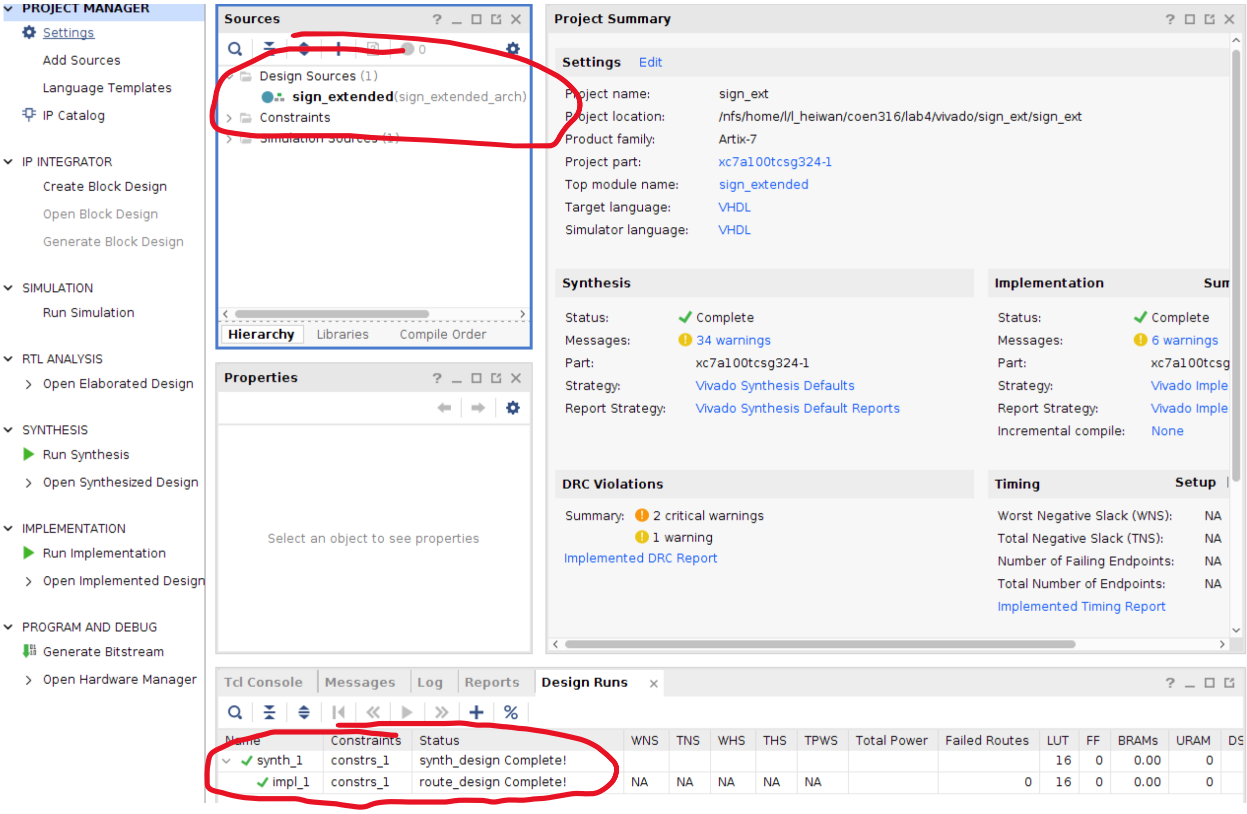
DRC Violations

Summary: 2 critical warnings
1 warning
Implemented DRC Report

Timing Setup

Worst Negative Slack (WNS): NA
Total Negative Slack (TNS): NA
Number of Failing Endpoints: NA
Total Number of Endpoints: NA
Implemented Timing Report

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM
✓ synth_1	constrs_1	synth_design Complete!								320	1024	0.00	0
✓ impl_1	constrs_1	route_design Complete!	NA	NA	NA	NA	NA		0	320	1024	0.00	0



Design Sources (1)
sign_extended(sign_extended_arch)
Constraints
Simulation Sources (1)

Project Summary

Settings Edit

Project name: sign_ext
Project location: /nfs/home/l/l_heiwan/coen316/lab4/vivado/sign_ext/sign_ext
Product family: Artix-7
Project part: xc7a100tcsg324-1
Top module name: sign_extended
Target language: VHDL
Simulator language: VHDL

Synthesis

Status: ✓ Complete
Messages: 34 warnings
Part: xc7a100tcsg324-1
Strategy: Vivado Synthesis Defaults
Report Strategy: Vivado Synthesis Default Reports

Implementation

Status: ✓ Complete
Messages: 6 warnings
Part: xc7a100tcsg
Strategy: Vivado Imple
Report Strategy: Vivado Imple
Incremental compile: None

DRC Violations

Summary: 2 critical warnings
1 warning
Implemented DRC Report

Timing Setup

Worst Negative Slack (WNS): NA
Total Negative Slack (TNS): NA
Number of Failing Endpoints: NA
Total Number of Endpoints: NA
Implemented Timing Report

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DS
✓ synth_1	constrs_1	synth_design Complete!								16	0	0.00	0	
✓ impl_1	constrs_1	route_design Complete!	NA	NA	NA	NA	NA		0	16	0	0.00	0	

3.2) Part 2: Datapath/Control Unit Integration

20 Instructions with Opcode and Function Fields and Control Signals:

Inst.	Op	Func	Reg_ write	Reg_ dst	Reg_ in_src	Alu_ src	Add_ sub	Data_ write	Logic_ func	Func	Branch_ type	Pc_ sel
Lui	001111	001111	1	0	1	1	0 (don’t care)	0	00 (don’t care)	00	00	00
Add	000000	100000	1	1	1	0	0	0	00	10	00	00
Sub	000000	100010	1	1	1	0	1	0	00	10	00	00
Slt	000000	101010	1	1	1	0	1	0	00	01	00	00
Addi	001000	001000	1	0	1	1	0	0	00	10	00	00
Slti	001010	001010	1	0	1	1	1	0	00	01	00	00
And	000000	100100	1	1	1	0	0	0	00	11	00	00
Or	000000	100101	1	1	1	0	0	0	01	11	00	00
Xor	000000	100110	1	1	1	0	0	0	10	11	00	00
Nor	000000	100111	1	1	1	0	0	0	11	11	00	00
Andi	001100	001100	1	0	1	1	0	0	00	11	00	00
Ori	001101	001101	1	0	1	1	0	0	01	11	00	00
Xori	001110	001110	1	0	1	1	0	0	10	11	00	00
Lw	100011	100011	1	0	0	1	0	0	10 (don’t care)	10	00	00
Sw	101011	101011	0	0	0	1	0	1	00	10	00	00
J	000010	000010	0	0	0	0	0	0	00	00	00	01
Jr	000000	001000	0	0	0	0	0	0	00	00	00	10
Bltz	000001	000110	0	0	0	0	0	0	0	0	11	00
Beq	000100	000100	0	0	0	0	0	0	00	00	01	00
bne	000101	000101	0	0	0	0	0	0	00	00	10	00

Op: To distinguish among the different instructions.

Func: To distinguish among the different instructions.

Reg_write: Determines which register is to be written into.

Reg_dst: Determines rt field or rd field register to be written into.

Reg_in_src: Mux for ALU and d_cache.

Alu_src: Only signed values care, rest are 0.

Add_sub: (lab 1) Only adder_subtract care, rest are 0.

Data_write: Active write signal.

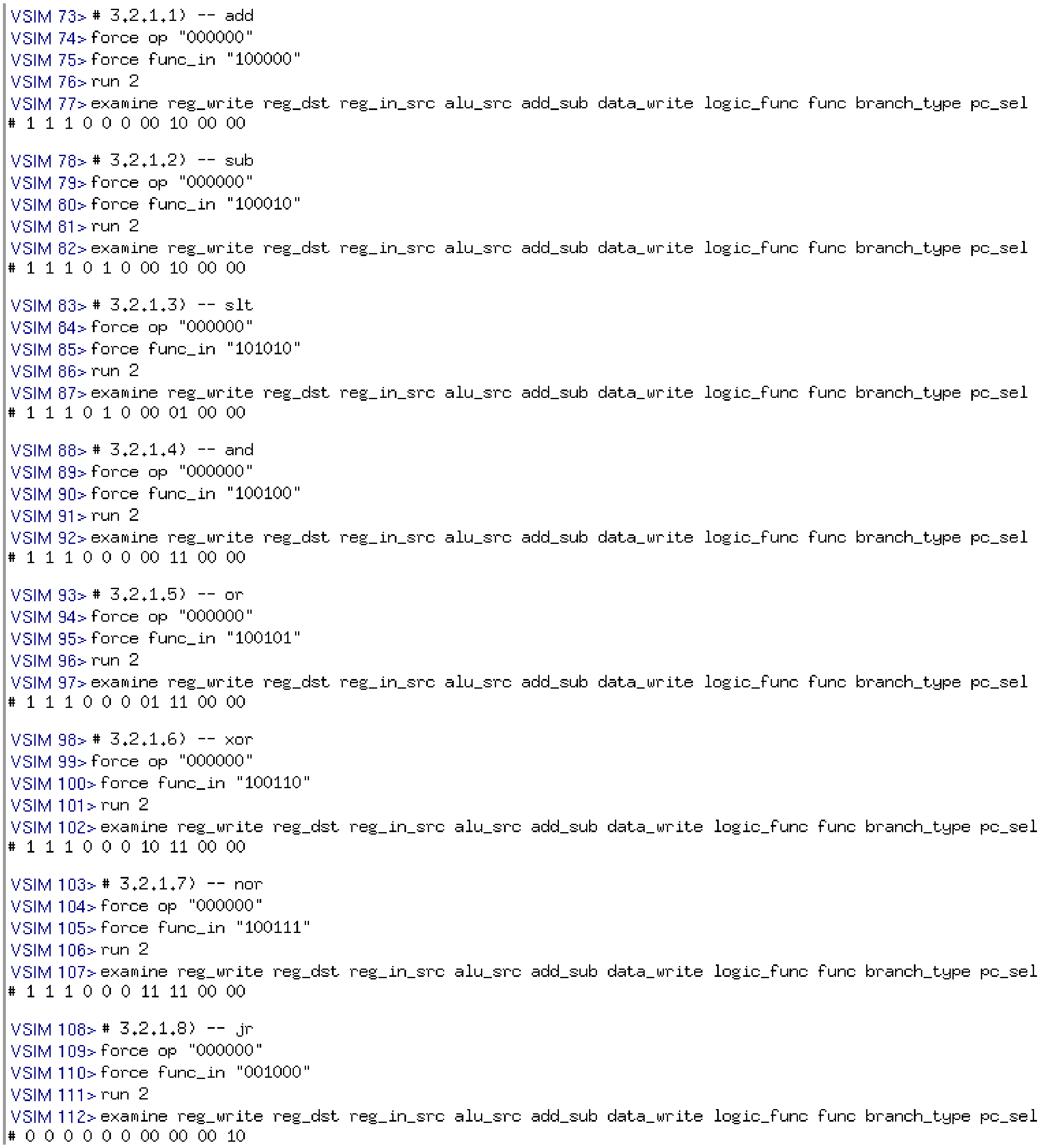
Logic_func: (lab 1) Only logic unit care, rest are 00.

Func: (lab 1) Only lui, slt, adder_substract and logic unit care, rest are 00.

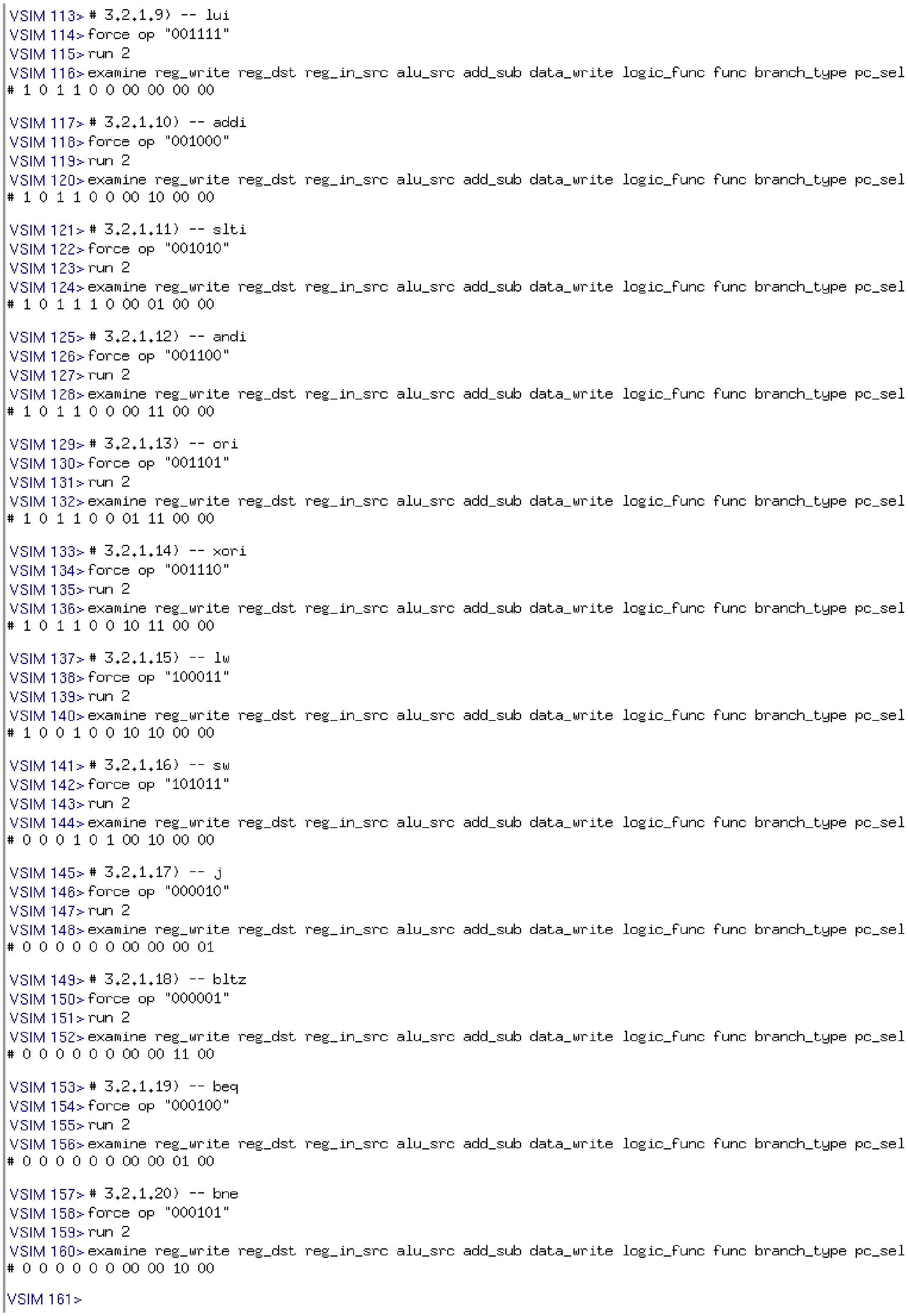
Branch_type: (lab 3) Only beq, bne and bltz care, rest are 00.

Pc_sel: (lab 3) Only jump and jump register care, rest are 00.

3.2.1) Control Unit



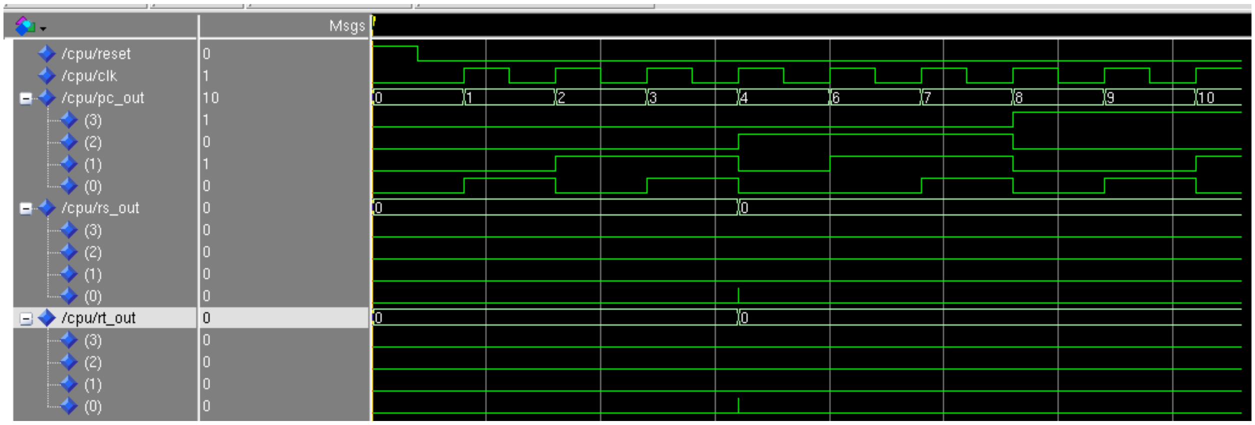
```
VSIM 73> # 3.2.1.1) -- add
VSIM 74> force op "000000"
VSIM 75> force func_in "100000"
VSIM 76> run 2
VSIM 77> examine reg_write reg_dst reg_in_src alu_src add_sub data_write logic_func func branch_type pc_sel
# 1 1 1 0 0 0 00 10 00 00

VSIM 78> # 3.2.1.2) -- sub
VSIM 79> force op "000000"
VSIM 80> force func_in "100010"
VSIM 81> run 2
VSIM 82> examine reg_write reg_dst reg_in_src alu_src add_sub data_write logic_func func branch_type pc_sel
# 1 1 1 0 1 0 00 10 00 00

VSIM 83> # 3.2.1.3) -- slt
VSIM 84> force op "000000"
VSIM 85> force func_in "101010"
VSIM 86> run 2
VSIM 87> examine reg_write reg_dst reg_in_src alu_src add_sub data_write logic_func func branch_type pc_sel
# 1 1 1 0 1 0 00 01 00 00

VSIM 88> # 3.2.1.4) -- and
VSIM 89> force op "000000"
VSIM 90> force func_in "100100"
VSIM 91> run 2
VSIM 92> examine reg_write reg_dst reg_in_src alu_src add_sub data_write logic_func func branch_type pc_sel
# 1 1 1 0 0 0 00 11 00 00

VSIM 93> # 3.2.1.5) -- or
VSIM 94> force op "000000"
VSIM 95> force func_in "100101"
VSIM 96> run 2
VSIM 97> examine reg_write reg_dst reg_in_src alu_src add_sub data_write logic_func func branch_type pc_sel
# 1 1 1 0 0 0 01 11 00 00

VSIM 98> # 3.2.1.6) -- xor
VSIM 99> force op "000000"
VSIM 100> force func_in "100110"
VSIM 101> run 2
VSIM 102> examine reg_write reg_dst reg_in_src alu_src add_sub data_write logic_func func branch_type pc_sel
# 1 1 1 0 0 0 10 11 00 00

VSIM 103> # 3.2.1.7) -- nor
VSIM 104> force op "000000"
VSIM 105> force func_in "100111"
VSIM 106> run 2
VSIM 107> examine reg_write reg_dst reg_in_src alu_src add_sub data_write logic_func func branch_type pc_sel
# 1 1 1 0 0 0 11 11 00 00

VSIM 108> # 3.2.1.8) -- jr
VSIM 109> force op "000000"
VSIM 110> force func_in "001000"
VSIM 111> run 2
VSIM 112> examine reg_write reg_dst reg_in_src alu_src add_sub data_write logic_func func branch_type pc_sel
# 0 0 0 0 0 0 00 00 00 10
```



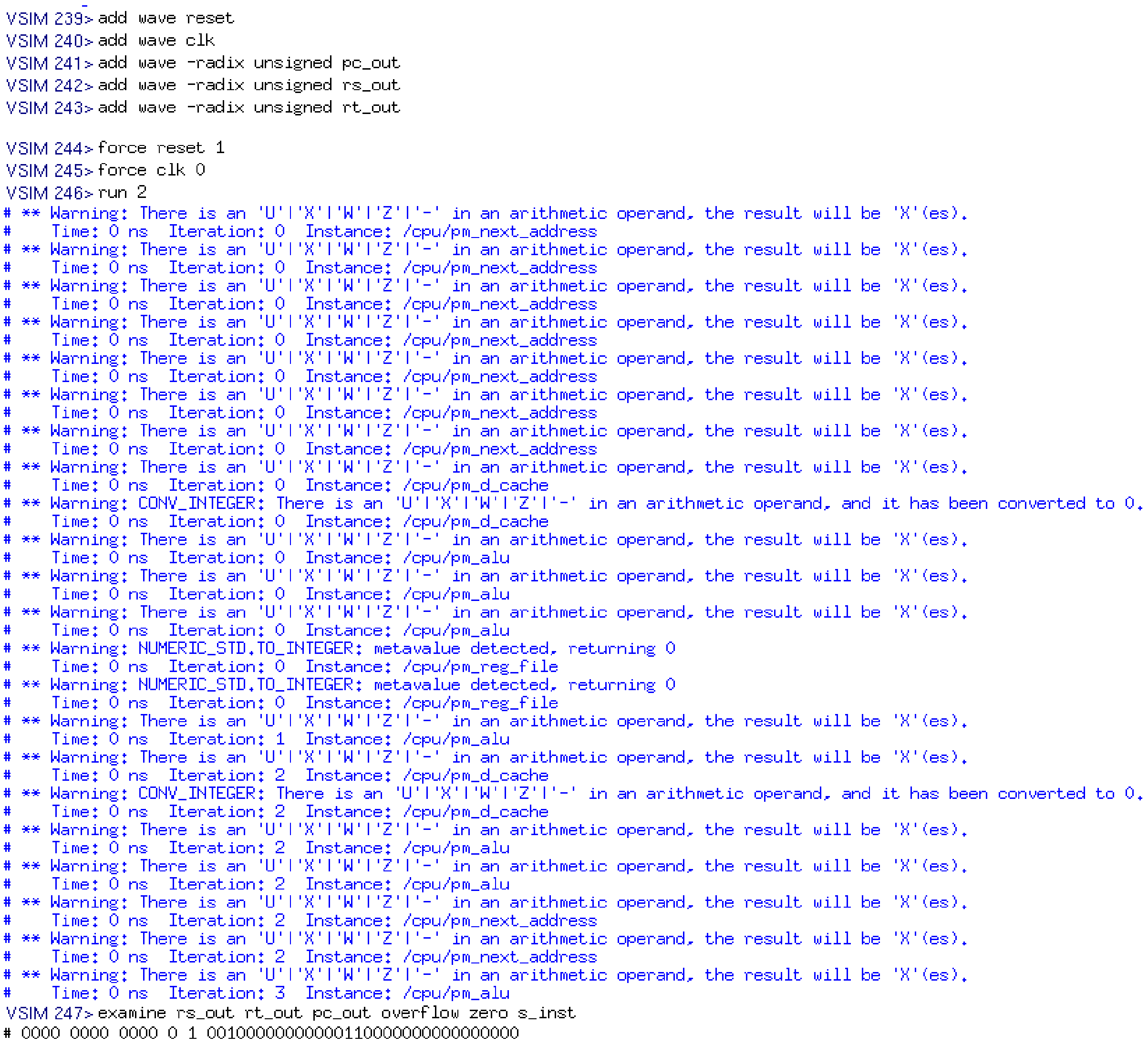
```
VSIM 113> # 3.2.1.9) -- lui
VSIM 114> force op "001111"
VSIM 115> run 2
VSIM 116> examine reg_write reg_dst reg_in_src alu_src add_sub data_write logic_func func branch_type pc_sel
# 1 0 1 1 0 0 00 00 00 00

VSIM 117> # 3.2.1.10) -- addi
VSIM 118> force op "001000"
VSIM 119> run 2
VSIM 120> examine reg_write reg_dst reg_in_src alu_src add_sub data_write logic_func func branch_type pc_sel
# 1 0 1 1 0 0 00 10 00 00

VSIM 121> # 3.2.1.11) -- slti
VSIM 122> force op "001010"
VSIM 123> run 2
VSIM 124> examine reg_write reg_dst reg_in_src alu_src add_sub data_write logic_func func branch_type pc_sel
# 1 0 1 1 1 0 00 01 00 00

VSIM 125> # 3.2.1.12) -- andi
VSIM 126> force op "001100"
VSIM 127> run 2
VSIM 128> examine reg_write reg_dst reg_in_src alu_src add_sub data_write logic_func func branch_type pc_sel
# 1 0 1 1 0 0 00 11 00 00

VSIM 129> # 3.2.1.13) -- ori
VSIM 130> force op "001101"
VSIM 131> run 2
VSIM 132> examine reg_write reg_dst reg_in_src alu_src add_sub data_write logic_func func branch_type pc_sel
# 1 0 1 1 0 0 01 11 00 00

VSIM 133> # 3.2.1.14) -- xori
VSIM 134> force op "001110"
VSIM 135> run 2
VSIM 136> examine reg_write reg_dst reg_in_src alu_src add_sub data_write logic_func func branch_type pc_sel
# 1 0 1 1 0 0 10 11 00 00

VSIM 137> # 3.2.1.15) -- lw
VSIM 138> force op "100011"
VSIM 139> run 2
VSIM 140> examine reg_write reg_dst reg_in_src alu_src add_sub data_write logic_func func branch_type pc_sel
# 1 0 0 1 0 0 10 10 00 00

VSIM 141> # 3.2.1.16) -- sw
VSIM 142> force op "101011"
VSIM 143> run 2
VSIM 144> examine reg_write reg_dst reg_in_src alu_src add_sub data_write logic_func func branch_type pc_sel
# 0 0 0 1 0 1 00 10 00 00

VSIM 145> # 3.2.1.17) -- j
VSIM 146> force op "000010"
VSIM 147> run 2
VSIM 148> examine reg_write reg_dst reg_in_src alu_src add_sub data_write logic_func func branch_type pc_sel
# 0 0 0 0 0 0 00 00 00 01

VSIM 149> # 3.2.1.18) -- bltz
VSIM 150> force op "000001"
VSIM 151> run 2
VSIM 152> examine reg_write reg_dst reg_in_src alu_src add_sub data_write logic_func func branch_type pc_sel
# 0 0 0 0 0 0 00 00 11 00

VSIM 153> # 3.2.1.19) -- beq
VSIM 154> force op "000100"
VSIM 155> run 2
VSIM 156> examine reg_write reg_dst reg_in_src alu_src add_sub data_write logic_func func branch_type pc_sel
# 0 0 0 0 0 0 00 00 01 00

VSIM 157> # 3.2.1.20) -- bne
VSIM 158> force op "000101"
VSIM 159> run 2
VSIM 160> examine reg_write reg_dst reg_in_src alu_src add_sub data_write logic_func func branch_type pc_sel
# 0 0 0 0 0 0 00 00 10 00

VSIM 161>
```

We tested the functionality of control_unit,vhd by forcing op and func_in values. By examining the output, we can observe that all 20 instructions match with the table.

3.2.2) CPU Datapath

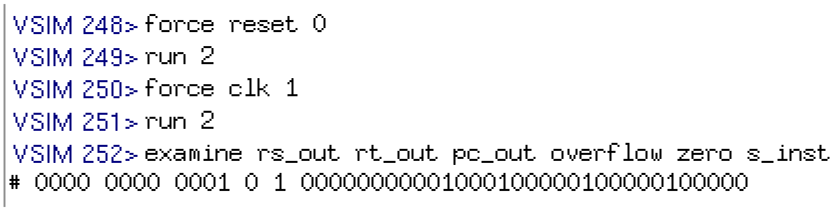


3.2.2.1) Arithmetic (w/ imm operands)



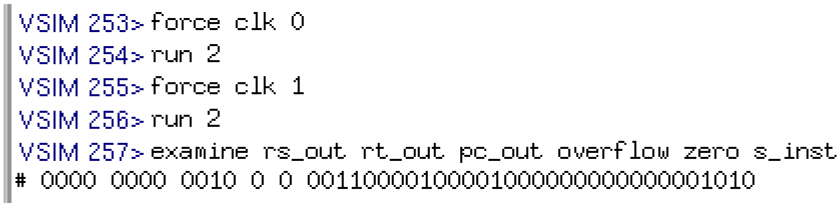
```
VSIM 239> add wave reset
VSIM 240> add wave clk
VSIM 241> add wave -radix unsigned pc_out
VSIM 242> add wave -radix unsigned rs_out
VSIM 243> add wave -radix unsigned rt_out

VSIM 244> force reset 1
VSIM 245> force clk 0
VSIM 246> run 2
# ** Warning: There is an 'U'|'X'|'W'|'Z'|'-' in an arithmetic operand, the result will be 'X'(es).
#    Time: 0 ns  Iteration: 0  Instance: /cpu/pm_next_address
# ** Warning: There is an 'U'|'X'|'W'|'Z'|'-' in an arithmetic operand, the result will be 'X'(es).
#    Time: 0 ns  Iteration: 0  Instance: /cpu/pm_next_address
# ** Warning: There is an 'U'|'X'|'W'|'Z'|'-' in an arithmetic operand, the result will be 'X'(es).
#    Time: 0 ns  Iteration: 0  Instance: /cpu/pm_next_address
# ** Warning: There is an 'U'|'X'|'W'|'Z'|'-' in an arithmetic operand, the result will be 'X'(es).
#    Time: 0 ns  Iteration: 0  Instance: /cpu/pm_next_address
# ** Warning: There is an 'U'|'X'|'W'|'Z'|'-' in an arithmetic operand, the result will be 'X'(es).
#    Time: 0 ns  Iteration: 0  Instance: /cpu/pm_next_address
# ** Warning: There is an 'U'|'X'|'W'|'Z'|'-' in an arithmetic operand, the result will be 'X'(es).
#    Time: 0 ns  Iteration: 0  Instance: /cpu/pm_next_address
# ** Warning: There is an 'U'|'X'|'W'|'Z'|'-' in an arithmetic operand, the result will be 'X'(es).
#    Time: 0 ns  Iteration: 0  Instance: /cpu/pm_next_address
# ** Warning: There is an 'U'|'X'|'W'|'Z'|'-' in an arithmetic operand, the result will be 'X'(es).
#    Time: 0 ns  Iteration: 0  Instance: /cpu/pm_d_cache
# ** Warning: CONV_INTEGER: There is an 'U'|'X'|'W'|'Z'|'-' in an arithmetic operand, and it has been converted to 0.
#    Time: 0 ns  Iteration: 0  Instance: /cpu/pm_d_cache
# ** Warning: There is an 'U'|'X'|'W'|'Z'|'-' in an arithmetic operand, the result will be 'X'(es).
#    Time: 0 ns  Iteration: 0  Instance: /cpu/pm_alu
# ** Warning: There is an 'U'|'X'|'W'|'Z'|'-' in an arithmetic operand, the result will be 'X'(es).
#    Time: 0 ns  Iteration: 0  Instance: /cpu/pm_alu
# ** Warning: There is an 'U'|'X'|'W'|'Z'|'-' in an arithmetic operand, the result will be 'X'(es).
#    Time: 0 ns  Iteration: 0  Instance: /cpu/pm_alu
# ** Warning: NUMERIC_STD.TO_INTEGER: metavalue detected, returning 0
#    Time: 0 ns  Iteration: 0  Instance: /cpu/pm_reg_file
# ** Warning: NUMERIC_STD.TO_INTEGER: metavalue detected, returning 0
#    Time: 0 ns  Iteration: 0  Instance: /cpu/pm_reg_file
# ** Warning: There is an 'U'|'X'|'W'|'Z'|'-' in an arithmetic operand, the result will be 'X'(es).
#    Time: 0 ns  Iteration: 1  Instance: /cpu/pm_alu
# ** Warning: There is an 'U'|'X'|'W'|'Z'|'-' in an arithmetic operand, the result will be 'X'(es).
#    Time: 0 ns  Iteration: 2  Instance: /cpu/pm_d_cache
# ** Warning: CONV_INTEGER: There is an 'U'|'X'|'W'|'Z'|'-' in an arithmetic operand, and it has been converted to 0.
#    Time: 0 ns  Iteration: 2  Instance: /cpu/pm_d_cache
# ** Warning: There is an 'U'|'X'|'W'|'Z'|'-' in an arithmetic operand, the result will be 'X'(es).
#    Time: 0 ns  Iteration: 2  Instance: /cpu/pm_alu
# ** Warning: There is an 'U'|'X'|'W'|'Z'|'-' in an arithmetic operand, the result will be 'X'(es).
#    Time: 0 ns  Iteration: 2  Instance: /cpu/pm_alu
# ** Warning: There is an 'U'|'X'|'W'|'Z'|'-' in an arithmetic operand, the result will be 'X'(es).
#    Time: 0 ns  Iteration: 2  Instance: /cpu/pm_next_address
# ** Warning: There is an 'U'|'X'|'W'|'Z'|'-' in an arithmetic operand, the result will be 'X'(es).
#    Time: 0 ns  Iteration: 2  Instance: /cpu/pm_next_address
# ** Warning: There is an 'U'|'X'|'W'|'Z'|'-' in an arithmetic operand, the result will be 'X'(es).
#    Time: 0 ns  Iteration: 3  Instance: /cpu/pm_alu
VSIM 247> examine rs_out rt_out pc_out overflow zero s_inst
# 0000 0000 0000 0 1 00100000000000110000000000000000
```

3.2.2.2) Arithmetic (w/o imm operands)



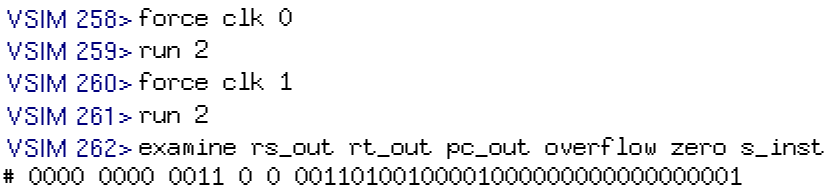
```
VSIM 248> force reset 0
VSIM 249> run 2
VSIM 250> force clk 1
VSIM 251> run 2
VSIM 252> examine rs_out rt_out pc_out overflow zero s_inst
# 0000 0000 0001 0 1 00000000001000100000100000100000
```

3.2.2.3) Logic (w/ imm operands)



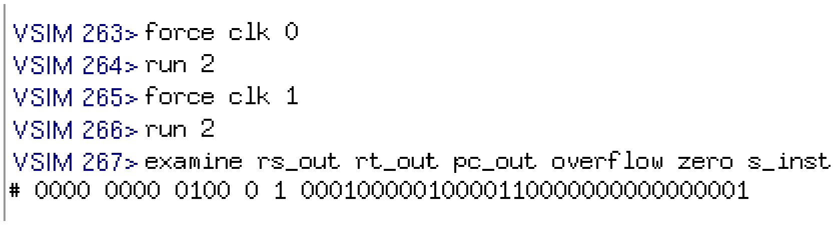
```
VSIM 253> force clk 0
VSIM 254> run 2
VSIM 255> force clk 1
VSIM 256> run 2
VSIM 257> examine rs_out rt_out pc_out overflow zero s_inst
# 0000 0000 0010 0 0 00110000100001000000000000001010
```

3.2.2.4) Logic (w/o imm operands)



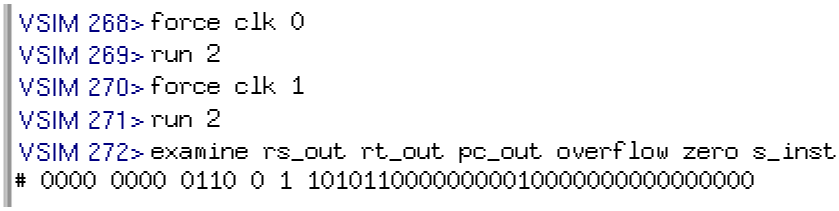
```
VSIM 258> force clk 0
VSIM 259> run 2
VSIM 260> force clk 1
VSIM 261> run 2
VSIM 262> examine rs_out rt_out pc_out overflow zero s_inst
# 0000 0000 0011 0 0 00110100100001000000000000000001
```

3.2.2.5) Conditional branch



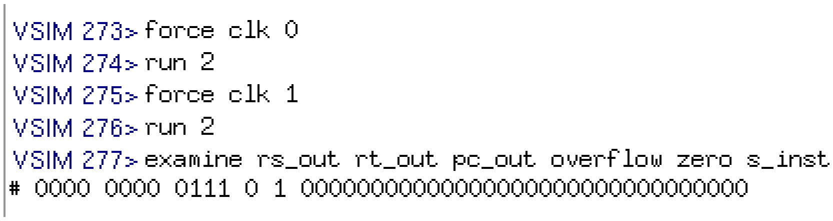
```
VSIM 263> force clk 0
VSIM 264> run 2
VSIM 265> force clk 1
VSIM 266> run 2
VSIM 267> examine rs_out rt_out pc_out overflow zero s_inst
# 0000 0000 0100 0 1 00010000010000110000000000000001
```

3.2.2.6) Memory access inst



```
VSIM 268> force clk 0
VSIM 269> run 2
VSIM 270> force clk 1
VSIM 271> run 2
VSIM 272> examine rs_out rt_out pc_out overflow zero s_inst
# 0000 0000 0110 0 1 10101100000000010000000000000000
```

3.2.2.7) Don’t care



```
VSIM 273> force clk 0
VSIM 274> run 2
VSIM 275> force clk 1
VSIM 276> run 2
VSIM 277> examine rs_out rt_out pc_out overflow zero s_inst
# 0000 0000 0111 0 1 00000000000000000000000000000000
```

Given that rest is 0 and there is a rising edge clk, the cpu will run through each i_cache machine code line. This can be seen from examine s_inst which match the i_cache machine code. For 3.2.2.7, the s_inst is all zeros as there are no more command in the i_cache. In this case the cpu will keep going outputting zeros.

4) Conclusions

In conclusion, lab 4 allowed for students to experiment on the intricacies of a CPU design through the usage of a CPU Datapath and subsequent components. By working progressively throughout lab 1 until now, students were able to deepen their understanding of CPU architecture. In the end, simulating and verifying the behavior of various instructions provided a comprehensive knowledge on computer architecture which is the main focus of the course COEN 316.

5) Appendix

5.1) VHDL Code

5.1.1) control_unit.vhd

```
library IEEE;
use IEEE.std_logic_1164.all;

entity control_unit is
port(
      op:                    in std_logic_vector(5 downto 0);
      func_in:          in std_logic_vector(5 downto 0);
      reg_write:        out std_logic;
      reg_dst:          out std_logic;
      reg_in_src:            out std_logic;
      alu_src:          out std_logic;
      add_sub:          out std_logic;
      data_write:            out std_logic;
      logic_func:            out std_logic_vector(1 downto 0);
      func:             out std_logic_vector(1 downto 0);
      branch_type:      out std_logic_vector(1 downto 0);
      pc_sel:                out std_logic_vector(1 downto 0));
end control_unit;
```

```
architecture control_unit_arch of control_unit is

signal control_signals: std_logic_vector(13 downto 0);

begin
    process(op, func_in, control_signals)
    begin
        case(op) is
        when "000000" =>
            case(func_in) is
            when "100000" => control_signals <= "11100000100000";
-- add
            when "100010" => control_signals <= "11101000100000";
-- sub
            when "101010" => control_signals <= "11101000010000";
-- slt
            when "100100" => control_signals <= "11100000110000";
-- and
            when "100101" => control_signals <= "11100001110000";
-- or
            when "100110" => control_signals <= "11100010110000";
-- xor
            when "100111" => control_signals <= "11100011110000";
-- nor
            when "001000" => control_signals <= "00000000000010";
-- jr
            when others   => control_signals <= "00000000000000";
            end case;
        when "001111" => control_signals <= "10110000000000"; --
lui
        when "001000" => control_signals <= "10110000100000"; --
addi
        when "001010" => control_signals <= "10111000010000"; --
slti
        when "001100" => control_signals <= "10110000110000"; --
andi
        when "001101" => control_signals <= "10110001110000"; --
ori
        when "001110" => control_signals <= "10110010110000"; --
xori
        when "100011" => control_signals <= "10010010100000"; -- lw
        when "101011" => control_signals <= "00010100100000"; -- sw
        when "000010" => control_signals <= "00000000000001"; -- j
        when "000001" => control_signals <= "00000000001100"; --
bltz
        when "000100" => control_signals <= "00000000000100"; --
beq
        when "000101" => control_signals <= "00000000001000"; --
```

```
bne
           when others   => control_signals <= "00000000000000";
           end case;

           reg_write  <= control_signals(13);
           reg_dst    <= control_signals(12);
           reg_in_src      <= control_signals(11);
           alu_src    <= control_signals(10);
           add_sub    <= control_signals(9);
           data_write      <= control_signals(8);
           logic_func      <= control_signals(7 downto 6);
           func       <= control_signals(5 downto 4);
           branch_type     <= control_signals(3 downto 2);
           pc_sel          <= control_signals(1 downto 0);
     end process;
end control_unit_arch;
```

5.1.2) cpu_entity.vhd

```
-- coen 316 lab 4
-- Andre Hei Wang Law
-- 4017 5600

library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_signed.all;

entity cpu is
port(reset:                in std_logic;
     clk:             in std_logic;
     rs_out, rt_out: out std_logic_vector(3 downto 0) := (others =>
'0'); -- output ports from reg. file
     pc_out:          out std_logic_vector(3 downto 0) := (others =>
'0');
     overflow, zero: out std_logic);
end cpu;

architecture datapath_cpu of cpu is
-- initialize components
     -- next address
     component next_address
     port(
           rt, rs:               in std_logic_vector(31 downto 0);
           pc:                   in std_logic_vector(31 downto 0);
           target_address:  in std_logic_vector(25 downto 0);
           branch_type:     in std_logic_vector(1 downto 0);
           pc_sel:               in std_logic_vector(1 downto 0);
           next_pc:         out std_logic_vector(31 downto 0));
     end component;

     -- pc
     component pc
     port(
           reset:           in std_logic;
           clk:        in std_logic;
           d_in:       in std_logic_vector(31 downto 0) := (others =>
'0');
           q_out:           out std_logic_vector(31 downto 0):= (others
=> '0'));
     end component;

     -- i_cache
     component i_cache
     port(
           addr_in:    in std_logic_vector(4 downto 0);
           inst_out:   out std_logic_vector(31 downto 0));
     end component;

     -- register file
     component reg_file
     port(
           din:  in std_logic_vector(31 downto 0);      -- data written
to specific register
           reset:      in std_logic;    -- asynchronous active-high
reset input
           clk:  in std_logic;    -- clock input
           write:      in std_logic;    -- write control signal
           read_a: in std_logic_vector(4 downto 0);    -- address for
read port A
           read_b: in std_logic_vector(4 downto 0);    -- address for
read port B
           write_address: in std_logic_vector(4 downto 0);  -- address
for write port
           out_a:      out std_logic_vector(31 downto 0);     -- data
output for read port A
           out_b:      out std_logic_vector(31 downto 0));    -- data
output for read port B
     end component;

     -- alu
     component alu
     port (
           x, y: in std_logic_vector(31 downto 0);           -- two
input operands
           add_sub:    in std_logic;                    -- 0=add, 1=sub
           logic_func:      in std_logic_vector(1 downto 0); -- 00=AND,
01=OR, 10=XOR, 11=NOR
           func:      in std_logic_vector(1 downto 0); -- 00=lui,
01=setless, 10=arith, 11=logic
           output:          out std_logic_vector(31 downto 0);
           overflow:  out std_logic;
           zero:      out std_logic);
      end component;

      -- d_cache
      component d_cache
      port(
           d_in:      in std_logic_vector(31 downto 0);
           reset:           in std_logic;
           clk:       in std_logic;
           data_write:      in std_logic;
           addr:      in std_logic_vector(4 downto 0);
           d_out:           out std_logic_vector(31 downto 0));
      end component;

      -- sign extend
      component sign_ext
      port(
           func:      in std_logic_vector(1 downto 0);
           sign_in:   in std_logic_vector(15 downto 0);
           sign_out:  out std_logic_vector(31 downto 0));
      end component;

      -- control unit
      component control_unit
      port(
           op:                   in std_logic_vector(5 downto 0);
           func_in:         in std_logic_vector(5 downto 0);
           reg_write:       out std_logic;
           reg_dst:         out std_logic;
           reg_in_src:           out std_logic;
           alu_src:         out std_logic;
           add_sub:         out std_logic;
           data_write:           out std_logic;
           logic_func:           out std_logic_vector(1 downto 0);
           func:            out std_logic_vector(1 downto 0);
           branch_type:     out std_logic_vector(1 downto 0);
           pc_sel:               out std_logic_vector(1 downto 0));
      end component;

-- signals
      -- control unit
      --signal s_op:             std_logic_vector(5 downto 0);
      --signal s_func_in:        std_logic_vector(5 downto 0);
      signal s_reg_write:        std_logic;
      signal s_reg_dst:          std_logic;
      signal s_reg_in_src:  std_logic;
      signal s_alu_src:          std_logic;
      signal s_add_sub:          std_logic;
      signal s_data_write:  std_logic;
      signal s_logic_func:  std_logic_vector(1 downto 0);
      signal s_func:             std_logic_vector(1 downto 0);
      signal s_branch_type: std_logic_vector(1 downto 0);
      signal s_pc_sel:      std_logic_vector(1 downto 0);

      signal s_inst:             std_logic_vector(31 downto 0);

      -- next address
      signal s_out_a:       std_logic_vector(31 downto 0);
      signal s_out_b:       std_logic_vector(31 downto 0);
      signal s_pc:          std_logic_vector(31 downto 0);
      signal s_next_pc:     std_logic_vector(31 downto 0);

      -- pc

      -- i_cache

      -- register file
      signal s_reg_in_src_out:   std_logic_vector(31 downto 0);
      signal s_reg_dst_out: std_logic_vector(4 downto 0);

      -- alu
      signal s_alu_src_out: std_logic_vector(31 downto 0);
      signal s_alu:              std_logic_vector(31 downto 0);

      -- d_cache
      signal s_d_out:            std_logic_vector(31 downto 0);

      -- sign extend
      signal s_sign_out:         std_logic_vector(31 downto 0);

      --   FOR pm_pc : "component_name" USE ENTITY work."entity_name"
("architecutre_name");
      FOR pm_pc : pc USE ENTITY work.pc (pc_arch);
      FOR pm_i_cache : i_cache USE ENTITY work.i_cache(i_cache_arch);
      FOR pm_reg_file : reg_file  USE ENTITY work.regfile (reg_arch);
      FOR pm_alu : alu USE ENTITY work.alu(arith_logic_unit);
      FOR pm_d_cache : d_cache USE ENTITY work.d_cache(d_cache_arch);
      FOR pm_sign_ext : sign_ext USE ENTITY
work.sign_extended(sign_extended_arch);
      FOR pm_next_address : next_address USE ENTITY
```

```
work.next_address(unit);
      FOR pm_control_unit : control_unit USE ENTITY
work.control_unit(control_unit_arch);

begin
      -- port map
      pm_pc: pc port map( -- pc
            reset => reset,
            clk => clk,
            d_in => s_next_pc,
            q_out => s_pc
      );

      pm_i_cache: i_cache port map( -- i_cache
            addr_in => s_pc(4 downto 0),
            inst_out => s_inst
      );

      pm_reg_file: reg_file port map( -- reg_file
            din => s_reg_in_src_out,
            reset => reset,
            clk => clk,
            write => s_reg_write,
            read_a => s_inst(25 downto 21),
            read_b => s_inst(20 downto 16),
            write_address => s_reg_dst_out,
            out_a => s_out_a,
            out_b => s_out_b
      );

      pm_alu: alu port map( -- alu
            x => s_out_a,
            y => s_alu_src_out,
            add_sub => s_add_sub,
            logic_func => s_logic_func,
            func => s_func,
            output => s_alu,
            overflow => overflow,
            zero => zero
      );

      pm_d_cache: d_cache port map( -- d_cache
            d_in => s_out_b,
            reset => reset,
            clk => clk,
            data_write => s_data_write,
            addr => s_alu(4 downto 0),
            d_out => s_d_out
      );

      pm_sign_ext: sign_ext port map( -- sign_ext
            func => s_func,
            sign_in => s_inst(15 downto 0),
            sign_out => s_sign_out
      );

      pm_next_address: next_address port map( -- next address
            rt => s_out_b,
            rs => s_out_a,
            pc => s_pc,
            target_address => s_inst(25 downto 0),
            branch_type => s_branch_type,
            pc_sel => s_pc_sel,
            next_pc => s_next_pc
      );

      pm_control_unit: control_unit port map( -- control_unit
            op => s_inst(31 downto 26),
            func_in => s_inst(5 downto 0),
            reg_write => s_reg_write,
            reg_dst => s_reg_dst,
            reg_in_src => s_reg_in_src,
            alu_src => s_alu_src,
            add_sub => s_add_sub,
            data_write => s_data_write,
            logic_func => s_logic_func,
            func => s_func,
            branch_type => s_branch_type,
            pc_sel => s_pc_sel
      );

      s_reg_dst_out <=      s_inst(20 downto 16) when (s_reg_dst = '0')
else
                                  s_inst(15 downto 11) when (s_reg_dst =
'1');

      s_alu_src_out <=s_out_b when (s_alu_src = '0') else
                                  s_sign_out when (s_alu_src = '1');
      s_reg_in_src_out <= s_d_out when (s_reg_in_src = '0') else
                                  s_alu when (s_reg_in_src = '1');

      rs_out <= s_out_a(3) & s_out_a(2) & s_out_a(1) & s_out_a(0);
      rt_out <= s_out_b(3) & s_out_b(2) & s_out_b(1) & s_out_b(0);
      pc_out <= s_pc(3) & s_pc(2) & s_pc(1) & s_pc(0);

end datapath_cpu;
```

5.1.3) d_cache.vhd

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_unsigned.all;

entity d_cache is
port(
     d_in:      in std_logic_vector(31 downto 0);
     reset:           in std_logic;
     clk:       in std_logic;
     data_write:      in std_logic;
     addr:      in std_logic_vector(4 downto 0);
     d_out:           out std_logic_vector(31 downto 0));
end d_cache;

architecture d_cache_arch of d_cache is

     type reg_array is array (0 to 31) of std_logic_vector(31 downto
0); -- 32 reg, each 32 bits
     signal registers : reg_array;

begin
     process(reset, clk)
     begin
          if (reset = '1') then -- reset (asynchronous)
               registers <= (others => (others => '0'));   -- clear
all reg. zeros
          elsif (rising_edge(clk) and data_write ='1') then-- click
(synchronous), rising edge
               registers(conv_integer(addr)) <= d_in;-- write
          end if;
     end process;

     process(addr, registers)
     begin
          d_out <= registers(conv_integer(addr));     -- read data
from addr
     end process;

end d_cache_arch;
```

5.1.4) i_cache.vhd

```
library IEEE;
use IEEE.std_logic_1164.all;
```

```
entity i_cache is
port(
     addr_in:   in std_logic_vector(4 downto 0);
     inst_out:  out std_logic_vector(31 downto 0));
end i_cache;

architecture i_cache_arch of i_cache is
begin
     process(addr_in)
     begin
          case(addr_in) is
               --when "00000" => inst_out <=
"00100000000000110000000000000000"; -- addi r3, r0, 0
               --when "00001" => inst_out <=
"00100000000000010000000000000000"; -- addi r1, r0, 0
               --when "00010" => inst_out <=
"00100000000000100000000000000101"; -- addi r2,r0,5
               --when "00011" => inst_out <=
"00000000001000100000100000100000"; -- add r1,r1,r2
               --when "00100" => inst_out <=
"00100000010000101111111111111111"; -- addi r2, r2, -1
               --when "00101" => inst_out <=
"00010000010000110000000000000001"; -- beq r2,r3 (+1) THERE
               --when "00110" => inst_out <=
"00001000000000000000000000000011"; -- jump 3  (LOOP)
               --when "00111" => inst_out <=
"10101100000000010000000000000000"; -- sw r1, 0(r0)
               --when "01000" => inst_out <=
"10001100000001000000000000000000"; -- lw r4, 0(r0)
               --when "01001" => inst_out <=
"00110000100001000000000000001010"; -- andi r4,r4, 0x000A
               --when "01010" => inst_out <=
"00110100100001000000000000000001"; -- ori r4,r4, 0x0001
               --when "01011" => inst_out <=
"00111000100001000000000000001011"; -- xori r4,r4, 0xB
               --when "01100" => inst_out <=
"00111000100001000000000000000000"; -- xori r4,r4, 0x0000

               -- values based on "machine_code.txt" found from TED's
directory
               when "00000" => inst_out <=
"00100000000000110000000000000000"; -- arithmetic (w/ imm operands),
addi r3, r0, 0
               when "00001" => inst_out <=
"00000000001000100000100000100000"; -- arithmetic (w/o imm operands),
add r1,r1,r2
               when "00010" => inst_out <=
"00110000100001000000000000001010"; -- logic (w/ imm operands), andi
```

```
r4,r4, 0x000A
                when "00011" => inst_out <= 
"00110100100001000000000000000001"; -- logic (w/o imm operands), ori 
r4,r4, 0x0001
                when "00100" => inst_out <= 
"00010000010000110000000000000001"; -- conditional branch, beq r2,r3 
(+1) THERE
                when "00101" => inst_out <= 
"00001000000000000000000000000100"; -- unconditional jump, jump 3 
(LOOP)
                when "00110" => inst_out <= 
"10101100000000010000000000000000"; -- memory access inst, sw r1, 
0(r0)
                when others =>  inst_out <= 
"00000000000000000000000000000000"; -- dont care
           end case;
     end process;
end i_cache_arch;
```

5.1.5) lab1_alu.vhd

```
-- coen 316 lab1
-- Andre Hei Wang Law
-- 4017 5600

library IEEE;
use IEEE.std_logic_1164.all;
use ieee.std_logic_signed.all;

entity alu is
     port (
     x, y: in std_logic_vector(31 downto 0);          -- two input 
operands
     add_sub:   in std_logic;                    -- 0=add, 1=sub
     logic_func:      in std_logic_vector(1 downto 0); -- 00=AND, 
01=OR, 10=XOR, 11=NOR
     func:      in std_logic_vector(1 downto 0); -- 00=lui, 
01=setless, 10=arith, 11=logic
     output:          out std_logic_vector(31 downto 0);
     overflow:  out std_logic;
     zero:      out std_logic);
end alu;

architecture arith_logic_unit of alu is
-- signal
signal adder_subtract_output, logic_unit_output: std_logic_vector(31 
downto 0);
```

```
begin
     -- adder_subtract (add_sub)
     process(x, y, add_sub)
     begin
          -- check for addition or substraction
          if (add_sub = '0') then
               adder_subtract_output <= x + y;        -- addition
          else
               adder_subtract_output <= x - y;        -- substraction
          end if;
     end process;

     -- LINE 40
     -- logic unit (logic_func)
     process(x, y, logic_func)
     begin
          case logic_func is
          when "00" =>
               logic_unit_output <= x AND y;     -- AND
          when "01" =>
               logic_unit_output <= x OR y;      -- OR
          when "10" =>
               logic_unit_output <= x XOR y;     -- XOR
          when others =>
               logic_unit_output <= not (x OR y);    -- NOR
          end case;
     end process;

     -- mux (func)
     process(y, adder_subtract_output, logic_unit_output, func)
     begin
          -- determines output
          case func is
          when "00" =>
               output <= y; -- y
          when "01" =>
               -- output = 1 if x < y
               -- output = 0 otherwise
               -- "000...MSB" of adder_subtract
               output <= "0000000000000000000000000000000" &
adder_subtract_output(31);
          when "10" =>
               output <= adder_subtract_output;   -- output of
adder_subtract
          when others =>
               output <= logic_unit_output;       -- output of logic
unit
          end case;
```

```
    end process;

    -- zero
    process(adder_subtract_output)
    begin
        -- zero = 1 when output of the adder_subtract unit is all
0s
        -- zero = 0 otherwise
        if adder_subtract_output =
"00000000000000000000000000000000" then
            zero <= '1';
        else
            zero <= '0';
        end if;
    end process;

    -- overflow
    process(adder_subtract_output, add_sub, x, y)
    begin
        -- overflow = 1 when:
            -- 1. Adding two positive num (x + y) and a negative
result is obtained
            -- 2. Adding two negative numbers (-x + -y) and a
possitive result is obtained
            -- 3. Substracting two operands of opposite sign (-x -
y)
            -- 4. Substracting two operands of opposite sign (x -
-y)

        -- 1. and 2. Addition
        if add_sub = '0' then
            if (x(31) = '0' and y(31) = '0' and
adder_subtract_output(31) = '1') or
               (x(31) = '1' and y(31) = '1' and
adder_subtract_output(31) = '0') then
                overflow <= '1'; -- Overflow detected
            else
                overflow <= '0'; -- No overflow
            end if;
            --overflow <= (not(x(31)) AND not(y(31)) AND
adder_subtract_output(31)) OR (x(31) AND y(31) AND
not(adder_subtract_output(31)));

        -- 3. and 4. Substraction
        elsif add_sub = '1' then
            if (x(31) = '0' and y(31) = '1' and
adder_subtract_output(31) = '1') or
               (x(31) = '1' and y(31) = '0' and
adder_subtract_output(31) = '0') then
                overflow <= '1'; -- Overflow detected
            else
                overflow <= '0'; -- No overflow
            end if;
            --overflow <= (x(31) AND not(y(31)) AND
not(adder_subtract_output(31))) OR (not(x(31)) AND y(31) AND
adder_subtract_output(31));
        end if;
    end process;
end arith_logic_unit;
```

5.1.6) lab2_register_file.vhd

```
-- coen 316 lab 2
-- Andre Hei Wang Law
-- 4017 5600

library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_unsigned.all;
use IEEE.numeric_std.all;

entity regfile is
    port (
    din:  in std_logic_vector(31 downto 0);      -- data written to
specific register
    reset:    in std_logic;    -- asynchronous active-high reset
input
    clk:  in std_logic;    -- clock input
    write:    in std_logic;    -- write control signal
    read_a: in std_logic_vector(4 downto 0);    -- address for read
port A
    read_b: in std_logic_vector(4 downto 0);    -- address for read
port B
    write_address: in std_logic_vector(4 downto 0);  -- address for
write port
    out_a:    out std_logic_vector(31 downto 0);    -- data output
for read port A
    out_b:    out std_logic_vector(31 downto 0));   -- data output
for read port B
end regfile;

architecture reg_arch of regfile is
    type reg_array is array (0 to 31) of std_logic_vector(31 downto
0); -- 32 reg, each 32 bits
    signal registers : reg_array;    -- register file, all zeros
```

```
begin
      process(reset, clk, write)
      begin
            if (reset = '1') then -- reset (asynchronous)
                  registers <= (others => (others => '0'));   -- clear
all reg. zeros
            elsif (clk'event and clk='1') then     -- click
(synchronous), rising edge
                  if (write = '1') then       -- check for active write
signal
                        registers(to_integer(unsigned(write_address))) <=
din; -- write
                  end if;
            end if;
      end process;
      -- reading (asynchronous)
      process(read_a, read_b, registers)
      begin
            out_a <= registers(to_integer(unsigned(read_a))); -- read
data from read_a
            out_b <= registers(to_integer(unsigned(read_b))); -- read
data from read_b
      end process;
end architecture reg_arch;
```

5.1.7) lab3_next_addr.vhd

```
-- coen 316 lab 3
-- Andre Hei Wang Law
-- 4017 5600

library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_signed.all;

entity next_address is
port(
      -- inputs
      rt, rs:                 in std_logic_vector(31 downto 0);
      pc:                     in std_logic_vector(31 downto 0);
      target_address:   in std_logic_vector(25 downto 0);
      branch_type:      in std_logic_vector(1 downto 0);
      pc_sel:                 in std_logic_vector(1 downto 0);
            -- output
      next_pc:          out std_logic_vector(31 downto 0));
end next_address;
```

```
architecture unit of next_address is

-- signal
signal tar_addr_signed_32: std_logic_vector(31 downto 0);
signal branch_offset_signed_32:  std_logic_vector(31 downto 0);
signal branch_offset: std_logic_vector(31 downto 0);

begin
    -- calculate "pseudo-direct" addressing to fill entire 32 bit
    -- target address for "jump" instructions
    tar_addr_signed_32(31 downto 26) <= (others =>
target_address(25)); -- fill with ones or zeros
    tar_addr_signed_32(25 downto 0) <= target_address; -- fill the
rest

    -- branch offset for no unconditional jump instructions
    branch_offset_signed_32(31 downto 16) <= (others =>
target_address(15)); -- fill with ones or zeros
    branch_offset_signed_32(15 downto  0) <= target_address(15 downto
0); -- fill the rest

    -- PC_sel functionality
    process(pc_sel, rs, branch_offset, pc, tar_addr_signed_32)
    begin
        if (pc_sel = "00") then -- no unconditional jump
            next_pc <= branch_offset + pc + 1;
        elsif (pc_sel = "01") then -- jump
            next_pc <= tar_addr_signed_32;
        elsif (pc_sel = "10") then -- jump register
            next_pc <= rs;
        elsif (pc_sel = "11") then -- not used
            -- do nothing
        end if;
    end process;

    -- branch_type functionality
    process(branch_type, rs, rt, branch_offset,
branch_offset_signed_32)
    begin
        if (branch_type = "00") then -- no branch
            branch_offset <= (others => '0');
        elsif (branch_type = "01") then -- beq (equal to 0)
            if (rs = rt) then
                branch_offset <= branch_offset_signed_32;
            else
                branch_offset <= (others => '0');
            end if;
        elsif (branch_type = "10") then -- bne (not equal to 0)
```

```
               if (rs /= rt) then
                    branch_offset <= branch_offset_signed_32;
               else
                    branch_offset <= (others => '0');
               end if;
          elsif (branch_type = "11") then -- bltz (less then 0)
               if (rs(31) = '1') then -- means rs is negative, or < 0
                    branch_offset <= branch_offset_signed_32;
               else
                    branch_offset <= (others => '0');
               end if;
          end if;
     end process;
end unit;
```

5.1.8) pc.vhd

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_signed.all;

entity pc is
port(
     reset:              in std_logic;
     clk:           in std_logic;
     d_in:          in std_logic_vector(31 downto 0);
     q_out:              out std_logic_vector(31 downto 0));
end pc;

architecture pc_arch of pc is
begin
     process(reset, clk)
     begin
          if (reset = '1') then -- reset (asynchronous)
               q_out <= (others => '0');  -- clear all
          elsif (clk'event and clk='1') then    -- click
(synchronous), rising edge
               q_out <= d_in;
          end if;
     end process;
end pc_arch;
```

5.1.9) sign_extended.vhd

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_signed.all;

entity sign_extended is
port(
      func:      in std_logic_vector(1 downto 0);
      sign_in:   in std_logic_vector(15 downto 0);
      sign_out:  out std_logic_vector(31 downto 0));
end sign_extended;

architecture sign_extended_arch of sign_extended is

begin
      -- mux (func)
      process(sign_in, func)
      begin
           case(func) is
           when "00" =>
                 sign_out(31 downto 16) <= sign_in; -- load upper
immediate
                 sign_out(15 downto  0) <= (others => '0');
           when "01" =>
                 sign_out(31 downto 16) <= (others => sign_in(15)); --
sign extend
                 sign_out(15 downto  0) <= sign_in; -- fill rest
           when "10" =>
                 sign_out(31 downto 16) <= (others => sign_in(15)); --
sign extend
                 sign_out(15 downto  0) <= sign_in; -- fill rest
           when others =>
                 sign_out(31 downto 16) <= (others => '0'); -- logical
                 sign_out(15 downto  0) <= sign_in;
           end case;
      end process;

end sign_extended_arch;
```

5.2) Constrain .xdc File

```
# Ted Obuchowicz
# XDC file for complete CPU

set_property CLOCK_DEDICATED_ROUTE  FALSE [ get_nets clk ] ;

# set the reset to the left most switch

set_property -dict { PACKAGE_PIN V10 IOSTANDARD LVCMOS33 }
[ get_ports { reset }  ] ;
```

```
# use the centre pushbutton as the clock

set_property -dict { PACKAGE_PIN N17 IOSTANDARD LVCMOS33 }
[ get_ports { clk }  ] ;


# set pc_out to left hand LEDS

set_property -dict { PACKAGE_PIN V11   IOSTANDARD LVCMOS33 }
[ get_ports { pc_out[3] }  ] ;
set_property -dict { PACKAGE_PIN V12 IOSTANDARD LVCMOS33 }
[ get_ports { pc_out[2] }  ] ;
set_property -dict { PACKAGE_PIN V14 IOSTANDARD LVCMOS33 }
[ get_ports { pc_out[1] }  ] ;
set_property -dict { PACKAGE_PIN V15 IOSTANDARD LVCMOS33 }
[ get_ports { pc_out[0] }  ] ;

# set overflow and zero immediately after pc_out

set_property -dict { PACKAGE_PIN T16 IOSTANDARD LVCMOS33 }
[ get_ports { overflow }  ] ;
set_property -dict { PACKAGE_PIN U14 IOSTANDARD LVCMOS33 }
[ get_ports { zero }  ] ;


# set rs_out and rt_out to the right hand LEDS  with a gap between the
two

# rs_out
set_property -dict { PACKAGE_PIN V16 IOSTANDARD LVCMOS33 }
[ get_ports { rs_out[3] }  ] ;
set_property -dict { PACKAGE_PIN U16 IOSTANDARD LVCMOS33 }
[ get_ports { rs_out[2] }  ] ;
set_property -dict { PACKAGE_PIN U17 IOSTANDARD LVCMOS33 }
[ get_ports { rs_out[1] }  ] ;
set_property -dict { PACKAGE_PIN V17 IOSTANDARD LVCMOS33 }
[ get_ports { rs_out[0] }  ] ;

# rt_out
set_property -dict { PACKAGE_PIN N14  IOSTANDARD LVCMOS33 }
[ get_ports { rt_out[3] }  ] ;
set_property -dict { PACKAGE_PIN J13 IOSTANDARD LVCMOS33 }
[ get_ports { rt_out[2] }  ] ;
set_property -dict { PACKAGE_PIN K15 IOSTANDARD LVCMOS33 }
[ get_ports { rt_out[1] }  ] ;
set_property -dict { PACKAGE_PIN H17 IOSTANDARD LVCMOS33 }
[ get_ports { rt_out[0] }  ] ;
```

5.3) DO File

5.3.1) Do File Control Unit

```
# 3.1.1 Data Cache

add wave *

# 3.2.1.1) -- add
force op "000000"
force func_in "100000"
run 2
examine reg_write reg_dst reg_in_src alu_src add_sub data_write logic_func func branch_type pc_sel

# 3.2.1.2) -- sub
force op "000000"
force func_in "100010"
run 2
examine reg_write reg_dst reg_in_src alu_src add_sub data_write logic_func func branch_type pc_sel

# 3.2.1.3) -- slt
force op "000000"
force func_in "101010"
run 2
examine reg_write reg_dst reg_in_src alu_src add_sub data_write logic_func func branch_type pc_sel

# 3.2.1.4) -- and
force op "000000"
force func_in "100100"
run 2
examine reg_write reg_dst reg_in_src alu_src add_sub data_write logic_func func branch_type pc_sel

# 3.2.1.5) -- or
force op "000000"
force func_in "100101"
run 2
examine reg_write reg_dst reg_in_src alu_src add_sub data_write logic_func func branch_type pc_sel

# 3.2.1.6) -- xor
force op "000000"
force func_in "100110"
run 2
```

```
examine reg_write reg_dst reg_in_src alu_src add_sub data_write
logic_func func branch_type pc_sel

# 3.2.1.7) -- nor
force op "000000"
force func_in "100111"
run 2
examine reg_write reg_dst reg_in_src alu_src add_sub data_write
logic_func func branch_type pc_sel

# 3.2.1.8) -- jr
force op "000000"
force func_in "001000"
run 2
examine reg_write reg_dst reg_in_src alu_src add_sub data_write
logic_func func branch_type pc_sel

# 3.2.1.9) -- lui
force op "001111"
run 2
examine reg_write reg_dst reg_in_src alu_src add_sub data_write
logic_func func branch_type pc_sel

# 3.2.1.10) -- addi
force op "001000"
run 2
examine reg_write reg_dst reg_in_src alu_src add_sub data_write
logic_func func branch_type pc_sel

# 3.2.1.11) -- slti
force op "001010"
run 2
examine reg_write reg_dst reg_in_src alu_src add_sub data_write
logic_func func branch_type pc_sel

# 3.2.1.12) -- andi
force op "001100"
run 2
examine reg_write reg_dst reg_in_src alu_src add_sub data_write
logic_func func branch_type pc_sel

# 3.2.1.13) -- ori
force op "001101"
run 2
examine reg_write reg_dst reg_in_src alu_src add_sub data_write
logic_func func branch_type pc_sel

# 3.2.1.14) -- xori
```

```
force op "001110"
run 2
examine reg_write reg_dst reg_in_src alu_src add_sub data_write
logic_func func branch_type pc_sel

# 3.2.1.15) -- lw
force op "100011"
run 2
examine reg_write reg_dst reg_in_src alu_src add_sub data_write
logic_func func branch_type pc_sel

# 3.2.1.16) -- sw
force op "101011"
run 2
examine reg_write reg_dst reg_in_src alu_src add_sub data_write
logic_func func branch_type pc_sel

# 3.2.1.17) -- j
force op "000010"
run 2
examine reg_write reg_dst reg_in_src alu_src add_sub data_write
logic_func func branch_type pc_sel

# 3.2.1.18) -- bltz
force op "000001"
run 2
examine reg_write reg_dst reg_in_src alu_src add_sub data_write
logic_func func branch_type pc_sel

# 3.2.1.19) -- beq
force op "000100"
run 2
examine reg_write reg_dst reg_in_src alu_src add_sub data_write
logic_func func branch_type pc_sel

# 3.2.1.20) -- bne
force op "000101"
run 2
examine reg_write reg_dst reg_in_src alu_src add_sub data_write
logic_func func branch_type pc_sel
```

5.3.2) Do File CPU Unit

```
add wave reset
add wave clk
add wave -radix unsigned pc_out
add wave -radix unsigned rs_out
add wave -radix unsigned rt_out

force reset 1
force clk 0
run 2
examine rs_out rt_out pc_out overflow zero s_inst

force reset 0
run 2
force clk 1
run 2
examine rs_out rt_out pc_out overflow zero s_inst

force clk 0
run 2
force clk 1
run 2
examine rs_out rt_out pc_out overflow zero s_inst

force clk 0
run 2
force clk 1
run 2
examine rs_out rt_out pc_out overflow zero s_inst

force clk 0
run 2
force clk 1
run 2
examine rs_out rt_out pc_out overflow zero s_inst

force clk 0
run 2
force clk 1
run 2
examine rs_out rt_out pc_out overflow zero s_inst

force clk 0
run 2
force clk 1
run 2
examine rs_out rt_out pc_out overflow zero s_inst

force clk 0
run 2
force clk 1
run 2
examine rs_out rt_out pc_out overflow zero s_inst
```

```
force clk 0
run 2
force clk 1
run 2
examine rs_out rt_out pc_out overflow zero s_inst

force clk 0
run 2
force clk 1
run 2
examine rs_out rt_out pc_out overflow zero s_inst
```

5.3.3) Do File d_cache

```
# 3.1.1 Data Cache

add wave *

# 3.1.1.1 initialize value
force d_in x"AAAAAAAA"
force reset 0
force clk 1
force data_write 1
force addr "00000"
run 2
examine -radix hex d_out

# 3.1.1.2 reset 1, clk 0 and data_write 0
force d_in x"AAAAAAAA"
force reset 1
force clk 0
force data_write 0
force addr "00000"
run 2
examine -radix hex d_out

# 3.1.1.3 data_write 1, reset 0 and clk 0
force d_in x"AAAAAAAA"
force reset 0
force clk 0
force data_write 1
force addr "00000"
run 2
examine -radix hex d_out

# 3.1.1.4 data_write 0, reset 0 and clk 1
```

```
force clk 0
run 2
force d_in x"AAAAAAAA"
force reset 0
force clk 1
force data_write 0
force addr "00000"
run 2
examine -radix hex d_out

# 3.1.1.5 data_write 1, reset 0 and clk 1
force clk 0
run 2
force d_in x"AAAAAAAA"
force reset 0
force clk 1
force data_write 1
force addr "00000"
run 2
examine -radix hex d_out
```

5.3.4) Do File i_cache

```
# 3.1.2 Instructions Cache

add wave *

# 3.1.2.1 addi r1, r0, 1
force addr_in "00000"
run 2
examine -radix hex inst_out

# 3.1.2.2 addi r2, r0, 2
force addr_in "00001"
run 2
examine -radix hex inst_out

# 3.1.2.3 add r2, r2, r1
force addr_in "00010"
run 2
examine -radix hex inst_out

# 3.1.2.4 jump 00010
force addr_in "00011"
run 2
examine -radix hex inst_out
```

```
# 3.1.2.5 don’t care
force addr_in "01101"
run 2
examine -radix hex inst_out
```

5.3.5) Do File pc

```
# 3.1.3 Pc

add wave *

# 3.1.3.1 initialize value
force reset 0
force clk 1
force d_in x"AAAAAAAA"
run 2
examine -radix hex q_out

# 3.1.3.2 reset 1, clk 0
force clk 0
run 2
force reset 1
force d_in x"AAAAAAAA"
run 2
examine -radix hex q_out

# 3.1.3.3 reset 0, clk 1
force clk 0
run 2
force reset 0
force clk 1
force d_in x"AAAAAAAA"
run 2
examine -radix hex q_out
```

5.3.6) Do File sign_ext

```
# 3.1.4 Sign extended

add wave *

# 3.1.4.1 load upper immediate
force func "00"
force sign_in x"ABCD"
run 2
examine -radix hex sign_out
```

```
# 3.1.4.2 set less immediate
force func "01"
force sign_in x"ABCD"
run 2
examine -radix hex sign_out

# 3.1.4.3 arithmetic
force func "10"
force sign_in x"0BCD"
run 2
examine -radix hex sign_out

# 3.1.4.4 logical
force func "11"
force sign_in x"ABCD"
run 2
examine -radix hex sign_out
```

5.3) Synthesis Log

```
*** Running vivado
    with args -log cpu.vds -m64 -product Vivado -mode batch -messageDb vivado.pb -notrace -source cpu.tcl


****** Vivado v2018.2 (64-bit)
  **** SW Build 2258646 on Thu Jun 14 20:02:38 MDT 2018
  **** IP Build 2256618 on Thu Jun 14 22:10:49 MDT 2018
    ** Copyright 1986-2018 Xilinx, Inc. All Rights Reserved.

source cpu.tcl -notrace
Command: synth_design -top cpu -part xc7a100tcsg324-1
Starting synth_design
Attempting to get a license for feature 'Synthesis' and/or device 'xc7a100t'
INFO: [Common 17-349] Got license for feature 'Synthesis' and/or device 'xc7a100t'
INFO: Launching helper process for spawning children vivado processes
INFO: Helper process launched with PID 421
---------------------------------------------------------------------------------
Starting RTL Elaboration : Time (s): cpu = 00:00:02 ; elapsed = 00:00:04 . Memory (MB): peak = 1468.590 ; gain
= 86.727 ; free physical = 71004 ; free virtual = 153913
---------------------------------------------------------------------------------
WARNING: [Synth 8-2489] overwriting existing secondary unit i_cache_arch
[/nfs/home/l/l_heiwan/coen316/lab4/i_cache_quiz.vhd:10]
INFO: [Synth 8-638] synthesizing module 'cpu' [/nfs/home/l/l_heiwan/coen316/lab4/cpu_entity.vhd:17]
INFO: [Synth 8-3491] module 'pc' declared at '/nfs/home/l/l_heiwan/coen316/lab4/pc.vhd:5' bound to instance
'pm_pc' of component 'pc' [/nfs/home/l/l_heiwan/coen316/lab4/cpu_entity.vhd:165]
INFO: [Synth 8-638] synthesizing module 'pc' [/nfs/home/l/l_heiwan/coen316/lab4/pc.vhd:13]
INFO: [Synth 8-256] done synthesizing module 'pc' (1#1) [/nfs/home/l/l_heiwan/coen316/lab4/pc.vhd:13]
INFO: [Synth 8-3491] module 'i_cache' declared at '/nfs/home/l/l_heiwan/coen316/lab4/i_cache.vhd:4' bound to
instance 'pm_i_cache' of component 'i_cache' [/nfs/home/l/l_heiwan/coen316/lab4/cpu_entity.vhd:172]
INFO: [Synth 8-638] synthesizing module 'i_cache' [/nfs/home/l/l_heiwan/coen316/lab4/i_cache_quiz.vhd:10]
INFO: [Synth 8-256] done synthesizing module 'i_cache' (2#1)
[/nfs/home/l/l_heiwan/coen316/lab4/i_cache_quiz.vhd:10]
INFO: [Synth 8-3491] module 'regfile' declared at '/nfs/home/l/l_heiwan/coen316/lab4/lab2_register_file.vhd:10'
bound to instance 'pm_reg_file' of component 'reg_file' [/nfs/home/l/l_heiwan/coen316/lab4/cpu_entity.vhd:177]
INFO: [Synth 8-638] synthesizing module 'regfile' [/nfs/home/l/l_heiwan/coen316/lab4/lab2_register_file.vhd:23]
INFO: [Synth 8-256] done synthesizing module 'regfile' (3#1)
[/nfs/home/l/l_heiwan/coen316/lab4/lab2_register_file.vhd:23]
INFO: [Synth 8-3491] module 'alu' declared at '/nfs/home/l/l_heiwan/coen316/lab4/lab1_alu.vhd:9' bound to
instance 'pm_alu' of component 'alu' [/nfs/home/l/l_heiwan/coen316/lab4/cpu_entity.vhd:189]
INFO: [Synth 8-638] synthesizing module 'alu' [/nfs/home/l/l_heiwan/coen316/lab4/lab1_alu.vhd:20]
INFO: [Synth 8-256] done synthesizing module 'alu' (4#1) [/nfs/home/l/l_heiwan/coen316/lab4/lab1_alu.vhd:20]
```

```
INFO: [Synth 8-3491] module 'd_cache' declared at '/nfs/home/l/l_heiwan/coen316/lab4/d_cache.vhd:5' bound to
instance 'pm_d_cache' of component 'd_cache' [/nfs/home/l/l_heiwan/coen316/lab4/cpu_entity.vhd:200]
INFO: [Synth 8-638] synthesizing module 'd_cache' [/nfs/home/l/l_heiwan/coen316/lab4/d_cache.vhd:15]
WARNING: [Synth 8-614] signal 'data_write' is read in the process but is not in the sensitivity list
[/nfs/home/l/l_heiwan/coen316/lab4/d_cache.vhd:21]
INFO: [Synth 8-256] done synthesizing module 'd_cache' (5#1) [/nfs/home/l/l_heiwan/coen316/lab4/d_cache.vhd:15]
INFO: [Synth 8-3491] module 'sign_extended' declared at '/nfs/home/l/l_heiwan/coen316/lab4/sign_extended.vhd:5'
bound to instance 'pm_sign_ext' of component 'sign_ext' [/nfs/home/l/l_heiwan/coen316/lab4/cpu_entity.vhd:209]
INFO: [Synth 8-638] synthesizing module 'sign_extended'
[/nfs/home/l/l_heiwan/coen316/lab4/sign_extended.vhd:12]
INFO: [Synth 8-256] done synthesizing module 'sign_extended' (6#1)
[/nfs/home/l/l_heiwan/coen316/lab4/sign_extended.vhd:12]
INFO: [Synth 8-3491] module 'next_address' declared at '/nfs/home/l/l_heiwan/coen316/lab4/lab3_next_addr.vhd:9'
bound to instance 'pm_next_address' of component 'next_address'
[/nfs/home/l/l_heiwan/coen316/lab4/cpu_entity.vhd:215]
INFO: [Synth 8-638] synthesizing module 'next_address'
[/nfs/home/l/l_heiwan/coen316/lab4/lab3_next_addr.vhd:22]
INFO: [Synth 8-256] done synthesizing module 'next_address' (7#1)
[/nfs/home/l/l_heiwan/coen316/lab4/lab3_next_addr.vhd:22]
INFO: [Synth 8-3491] module 'control_unit' declared at '/nfs/home/l/l_heiwan/coen316/lab4/control_unit.vhd:4'
bound to instance 'pm_control_unit' of component 'control_unit'
[/nfs/home/l/l_heiwan/coen316/lab4/cpu_entity.vhd:225]
INFO: [Synth 8-638] synthesizing module 'control_unit' [/nfs/home/l/l_heiwan/coen316/lab4/control_unit.vhd:20]
INFO: [Synth 8-256] done synthesizing module 'control_unit' (8#1)
[/nfs/home/l/l_heiwan/coen316/lab4/control_unit.vhd:20]
INFO: [Synth 8-256] done synthesizing module 'cpu' (9#1) [/nfs/home/l/l_heiwan/coen316/lab4/cpu_entity.vhd:17]
---------------------------------------------------------------------------------
Finished RTL Elaboration : Time (s): cpu = 00:00:03 ; elapsed = 00:00:06 . Memory (MB): peak = 1512.230 ; gain
= 130.367 ; free physical = 71016 ; free virtual = 153924
---------------------------------------------------------------------------------

Report Check Netlist:
+------+------------------+-------+---------+-------+------------------+
|      |Item              |Errors |Warnings |Status |Description       |
+------+------------------+-------+---------+-------+------------------+
|1     |multi_driven_nets |      0|        0|Passed |Multi driven nets |
+------+------------------+-------+---------+-------+------------------+
---------------------------------------------------------------------------------
Start Handling Custom Attributes
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
Finished Handling Custom Attributes : Time (s): cpu = 00:00:03 ; elapsed = 00:00:06 . Memory (MB): peak =
1512.230 ; gain = 130.367 ; free physical = 71015 ; free virtual = 153923
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
Finished RTL Optimization Phase 1 : Time (s): cpu = 00:00:03 ; elapsed = 00:00:06 . Memory (MB): peak =
1512.230 ; gain = 130.367 ; free physical = 71015 ; free virtual = 153923
---------------------------------------------------------------------------------
INFO: [Device 21-403] Loading part xc7a100tcsg324-1
INFO: [Project 1-570] Preparing netlist for logic optimization

Processing XDC Constraints
Initializing timing engine
Parsing XDC File [/nfs/home/l/l_heiwan/coen316/lab4/givenToUs/cpu.xdc]
Finished Parsing XDC File [/nfs/home/l/l_heiwan/coen316/lab4/givenToUs/cpu.xdc]
INFO: [Project 1-236] Implementation specific constraints were found while reading constraint file
[/nfs/home/l/l_heiwan/coen316/lab4/givenToUs/cpu.xdc]. These constraints will be ignored for synthesis but will
be used in implementation. Impacted constraints are listed in the file [.Xil/cpu_propImpl.xdc].
Resolution: To avoid this warning, move constraints listed in [.Xil/cpu_propImpl.xdc] to another XDC file and
exclude this new file from synthesis with the used_in_synthesis property (File Properties dialog in GUI) and
re-run elaboration/synthesis.
Completed Processing XDC Constraints

INFO: [Project 1-111] Unisim Transformation Summary:
No Unisim elements were transformed.

Constraint Validation Runtime : Time (s): cpu = 00:00:00 ; elapsed = 00:00:00 . Memory (MB): peak = 1892.141 ;
gain = 0.000 ; free physical = 70673 ; free virtual = 153582
---------------------------------------------------------------------------------
Finished Constraint Validation : Time (s): cpu = 00:00:15 ; elapsed = 00:01:06 . Memory (MB): peak = 1892.141 ;
gain = 510.277 ; free physical = 70824 ; free virtual = 153732
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
Start Loading Part and Timing Information
---------------------------------------------------------------------------------
```

```
Loading part: xc7a100tcsg324-1
---------------------------------------------------------------------------------
Finished Loading Part and Timing Information : Time (s): cpu = 00:00:15 ; elapsed = 00:01:06 . Memory (MB):
peak = 1892.141 ; gain = 510.277 ; free physical = 70824 ; free virtual = 153732
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
Start Applying 'set_property' XDC Constraints
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
Finished applying 'set_property' XDC Constraints : Time (s): cpu = 00:00:15 ; elapsed = 00:01:06 . Memory (MB):
peak = 1892.141 ; gain = 510.277 ; free physical = 70825 ; free virtual = 153734
---------------------------------------------------------------------------------
INFO: [Synth 8-5546] ROM "inst_out" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[0]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[1]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[2]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[3]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[4]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[5]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[6]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[7]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[8]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[9]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[10]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[11]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[12]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[13]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[14]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[15]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[16]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[17]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[18]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[19]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[20]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[21]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[22]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[23]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[24]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[25]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[26]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[27]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[28]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[29]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[30]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[31]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5818] HDL ADVISOR - The operator resource <adder> is shared. To prevent sharing consider
applying a KEEP on the output of the operator [/nfs/home/l/l_heiwan/coen316/lab4/lab1_alu.vhd:30]
INFO: [Synth 8-5546] ROM "registers_reg[0]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[1]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[2]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[3]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[4]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[5]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[6]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[7]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[8]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[9]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[10]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[11]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[12]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[13]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[14]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[15]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[16]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[17]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[18]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[19]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[20]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[21]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[22]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[23]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[24]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[25]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[26]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[27]" won't be mapped to RAM because it is too sparse
```

```
INFO: [Synth 8-5546] ROM "registers_reg[28]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[29]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[30]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "registers_reg[31]" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-5546] ROM "control_signals" won't be mapped to RAM because it is too sparse
WARNING: [Synth 8-327] inferring latch for variable 'next_pc_reg'
[/nfs/home/l/l_heiwan/coen316/lab4/lab3_next_addr.vhd:43]
WARNING: [Synth 8-327] inferring latch for variable 'branch_offset_reg'
[/nfs/home/l/l_heiwan/coen316/lab4/lab3_next_addr.vhd:57]
--------------------------------------------------------------------------------
Finished RTL Optimization Phase 2 : Time (s): cpu = 00:00:16 ; elapsed = 00:01:07 . Memory (MB): peak =
1892.141 ; gain = 510.277 ; free physical = 70815 ; free virtual = 153724
--------------------------------------------------------------------------------

Report RTL Partitions:
+-+--------------+------------+----------+
| |RTL Partition |Replication |Instances |
+-+--------------+------------+----------+
+-+--------------+------------+----------+
--------------------------------------------------------------------------------
Start RTL Component Statistics
--------------------------------------------------------------------------------
Detailed RTL Component Info :
+---Adders :
           3 Input     32 Bit       Adders := 2
+---XORs :
           2 Input     32 Bit         XORs := 1
+---Registers :
                       32 Bit    Registers := 65
+---Muxes :
           2 Input     32 Bit        Muxes := 12
           4 Input     32 Bit        Muxes := 3
           9 Input     14 Bit        Muxes := 1
          14 Input     14 Bit        Muxes := 1
           2 Input      5 Bit        Muxes := 1
           2 Input      1 Bit        Muxes := 67
           4 Input      1 Bit        Muxes := 1
--------------------------------------------------------------------------------
Finished RTL Component Statistics
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Start RTL Hierarchical Component Statistics
--------------------------------------------------------------------------------
Hierarchical RTL Component report
Module cpu
Detailed RTL Component Info :
+---Muxes :
           2 Input     32 Bit        Muxes := 2
           2 Input      5 Bit        Muxes := 1
Module pc
Detailed RTL Component Info :
+---Registers :
                       32 Bit    Registers := 1
Module i_cache
Detailed RTL Component Info :
+---Muxes :
           2 Input     32 Bit        Muxes := 1
Module regfile
Detailed RTL Component Info :
+---Registers :
                       32 Bit    Registers := 32
+---Muxes :
           2 Input      1 Bit        Muxes := 32
Module alu
Detailed RTL Component Info :
+---Adders :
           3 Input     32 Bit       Adders := 1
+---XORs :
           2 Input     32 Bit         XORs := 1
+---Muxes :
           4 Input     32 Bit        Muxes := 2
           2 Input     32 Bit        Muxes := 1
           2 Input      1 Bit        Muxes := 1
Module d_cache
Detailed RTL Component Info :
```

```
+---Registers :
                       32 Bit    Registers := 32
+---Muxes :
           2 Input      1 Bit        Muxes := 32
Module sign_extended
Detailed RTL Component Info :
+---Muxes :
           4 Input     32 Bit        Muxes := 1
Module next_address
Detailed RTL Component Info :
+---Adders :
           3 Input     32 Bit       Adders := 1
+---Muxes :
           2 Input     32 Bit        Muxes := 8
           2 Input      1 Bit        Muxes := 2
           4 Input      1 Bit        Muxes := 1
Module control_unit
Detailed RTL Component Info :
+---Muxes :
           9 Input     14 Bit        Muxes := 1
          14 Input     14 Bit        Muxes := 1
---------------------------------------------------------------------------------
Finished RTL Hierarchical Component Statistics
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
Start Part Resource Summary
---------------------------------------------------------------------------------
Part Resources:
DSPs: 240 (col length:80)
BRAMs: 270 (col length: RAMB18 80 RAMB36 40)
---------------------------------------------------------------------------------
Finished Part Resource Summary
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
Start Cross Boundary and Area Optimization
---------------------------------------------------------------------------------
Warning: Parallel synthesis criteria is not met
INFO: [Synth 8-5546] ROM "pm_control_unit/control_signals" won't be mapped to RAM because it is too sparse
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[14][4]' (FDCE) to
'pm_reg_file/registers_reg[13][4]'
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[13][4]' (FDCE) to
'pm_reg_file/registers_reg[11][4]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[11][4] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[14][5]' (FDCE) to
'pm_reg_file/registers_reg[13][5]'
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[13][5]' (FDCE) to
'pm_reg_file/registers_reg[11][5]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[11][5] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[14][6]' (FDCE) to
'pm_reg_file/registers_reg[13][6]'
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[13][6]' (FDCE) to
'pm_reg_file/registers_reg[11][6]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[11][6] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[14][7]' (FDCE) to
'pm_reg_file/registers_reg[13][7]'
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[13][7]' (FDCE) to
'pm_reg_file/registers_reg[11][7]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[11][7] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[14][8]' (FDCE) to
'pm_reg_file/registers_reg[13][8]'
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[13][8]' (FDCE) to
'pm_reg_file/registers_reg[11][8]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[11][8] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[14][9]' (FDCE) to
'pm_reg_file/registers_reg[13][9]'
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[13][9]' (FDCE) to
'pm_reg_file/registers_reg[11][9]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[11][9] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[14][10]' (FDCE) to
'pm_reg_file/registers_reg[13][10]'
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[13][10]' (FDCE) to
'pm_reg_file/registers_reg[11][10]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[11][10] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[14][11]' (FDCE) to
'pm_reg_file/registers_reg[13][11]'
```

```
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[13][11]' (FDCE) to
'pm_reg_file/registers_reg[11][11]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[11][11] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[14][12]' (FDCE) to
'pm_reg_file/registers_reg[13][12]'
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[13][12]' (FDCE) to
'pm_reg_file/registers_reg[11][12]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[11][12] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[14][13]' (FDCE) to
'pm_reg_file/registers_reg[13][13]'
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[13][13]' (FDCE) to
'pm_reg_file/registers_reg[11][13]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[11][13] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[14][14]' (FDCE) to
'pm_reg_file/registers_reg[13][14]'
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[13][14]' (FDCE) to
'pm_reg_file/registers_reg[11][14]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[11][14] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[14][15]' (FDCE) to
'pm_reg_file/registers_reg[13][15]'
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[13][15]' (FDCE) to
'pm_reg_file/registers_reg[11][15]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[11][15] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[14][16]' (FDCE) to
'pm_reg_file/registers_reg[13][16]'
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[13][16]' (FDCE) to
'pm_reg_file/registers_reg[11][16]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[11][16] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[14][17]' (FDCE) to
'pm_reg_file/registers_reg[13][17]'
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[13][17]' (FDCE) to
'pm_reg_file/registers_reg[11][17]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[11][17] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[14][18]' (FDCE) to
'pm_reg_file/registers_reg[13][18]'
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[13][18]' (FDCE) to
'pm_reg_file/registers_reg[11][18]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[11][18] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[14][19]' (FDCE) to
'pm_reg_file/registers_reg[13][19]'
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[13][19]' (FDCE) to
'pm_reg_file/registers_reg[11][19]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[11][19] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[14][20]' (FDCE) to
'pm_reg_file/registers_reg[13][20]'
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[13][20]' (FDCE) to
'pm_reg_file/registers_reg[11][20]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[11][20] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[14][21]' (FDCE) to
'pm_reg_file/registers_reg[13][21]'
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[13][21]' (FDCE) to
'pm_reg_file/registers_reg[11][21]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[11][21] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[14][22]' (FDCE) to
'pm_reg_file/registers_reg[13][22]'
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[13][22]' (FDCE) to
'pm_reg_file/registers_reg[11][22]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[11][22] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[14][23]' (FDCE) to
'pm_reg_file/registers_reg[13][23]'
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[13][23]' (FDCE) to
'pm_reg_file/registers_reg[11][23]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[11][23] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[14][24]' (FDCE) to
'pm_reg_file/registers_reg[13][24]'
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[13][24]' (FDCE) to
'pm_reg_file/registers_reg[11][24]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[11][24] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[14][25]' (FDCE) to
'pm_reg_file/registers_reg[13][25]'
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[13][25]' (FDCE) to
'pm_reg_file/registers_reg[11][25]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[11][25] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[14][26]' (FDCE) to
'pm_reg_file/registers_reg[13][26]'
```

```
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[13][26]' (FDCE) to
'pm_reg_file/registers_reg[11][26]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[11][26] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[14][27]' (FDCE) to
'pm_reg_file/registers_reg[13][27]'
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[13][27]' (FDCE) to
'pm_reg_file/registers_reg[11][27]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[11][27] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[14][28]' (FDCE) to
'pm_reg_file/registers_reg[13][28]'
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[13][28]' (FDCE) to
'pm_reg_file/registers_reg[11][28]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[11][28] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[14][29]' (FDCE) to
'pm_reg_file/registers_reg[13][29]'
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[13][29]' (FDCE) to
'pm_reg_file/registers_reg[11][29]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[11][29] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[14][30]' (FDCE) to
'pm_reg_file/registers_reg[13][30]'
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[13][30]' (FDCE) to
'pm_reg_file/registers_reg[11][30]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[11][30] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[14][31]' (FDCE) to
'pm_reg_file/registers_reg[13][31]'
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[13][31]' (FDCE) to
'pm_reg_file/registers_reg[11][31]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[11][31] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[14][0]' (FDCE) to
'pm_reg_file/registers_reg[13][0]'
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[13][0]' (FDCE) to
'pm_reg_file/registers_reg[11][0]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[11][0] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[14][1]' (FDCE) to
'pm_reg_file/registers_reg[13][1]'
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[13][1]' (FDCE) to
'pm_reg_file/registers_reg[11][1]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[11][1] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[14][2]' (FDCE) to
'pm_reg_file/registers_reg[13][2]'
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[13][2]' (FDCE) to
'pm_reg_file/registers_reg[11][2]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[11][2] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[14][3]' (FDCE) to
'pm_reg_file/registers_reg[13][3]'
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[13][3]' (FDCE) to
'pm_reg_file/registers_reg[11][3]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[11][3] )
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[8][31]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[8][30]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[8][29]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[8][28]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[8][27]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[8][26]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[8][25]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[8][24]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[8][23]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[8][22]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[8][21]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[8][20]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[8][19]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[8][18]) is unused and will be removed
```

```
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[8][17]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[8][16]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[8][15]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[8][14]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[8][13]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[8][12]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[8][11]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[8][10]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[8][9]) is unused and will be removed from
module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[8][8]) is unused and will be removed from
module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[8][7]) is unused and will be removed from
module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[8][6]) is unused and will be removed from
module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[8][5]) is unused and will be removed from
module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[8][4]) is unused and will be removed from
module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[8][3]) is unused and will be removed from
module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[8][2]) is unused and will be removed from
module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[8][1]) is unused and will be removed from
module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[8][0]) is unused and will be removed from
module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[9][31]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[9][30]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[9][29]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[9][28]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[9][27]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[9][26]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[9][25]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[9][24]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[9][23]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[9][22]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[9][21]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[9][20]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[9][19]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[9][18]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[9][17]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[9][16]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[9][15]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[9][14]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[9][13]) is unused and will be removed
from module cpu.
```

```
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[9][12]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[9][11]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[9][10]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[9][9]) is unused and will be removed from
module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[9][8]) is unused and will be removed from
module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[9][7]) is unused and will be removed from
module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[9][6]) is unused and will be removed from
module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[9][5]) is unused and will be removed from
module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[9][4]) is unused and will be removed from
module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[9][3]) is unused and will be removed from
module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[9][2]) is unused and will be removed from
module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[9][1]) is unused and will be removed from
module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[9][0]) is unused and will be removed from
module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[10][31]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[10][30]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[10][29]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[10][28]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[10][27]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[10][26]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[10][25]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[10][24]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[10][23]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[10][22]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[10][21]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[10][20]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[10][19]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[10][18]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[10][17]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[10][16]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[10][15]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[10][14]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[10][13]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[10][12]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[10][11]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[10][10]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[10][9]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[10][8]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[10][7]) is unused and will be removed
```

```
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[10][6]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[10][5]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[10][4]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[10][3]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[10][2]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[10][1]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[10][0]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[12][31]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[12][30]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[12][29]) is unused and will be removed
from module cpu.
WARNING: [Synth 8-3332] Sequential element (pm_reg_file/registers_reg[12][28]) is unused and will be removed
from module cpu.
INFO: [Common 17-14] Message 'Synth 8-3332' appears 100 times and further instances of the messages will be
disabled. Use the Tcl command set_msg_config to change the current settings.
---------------------------------------------------------------------------------
Finished Cross Boundary and Area Optimization : Time (s): cpu = 00:00:28 ; elapsed = 00:01:20 . Memory (MB):
peak = 1908.156 ; gain = 526.293 ; free physical = 70786 ; free virtual = 153698
---------------------------------------------------------------------------------

Report RTL Partitions:
+-+--------------+------------+----------+
| |RTL Partition |Replication |Instances |
+-+--------------+------------+----------+
+-+--------------+------------+----------+
---------------------------------------------------------------------------------
Start Applying XDC Timing Constraints
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
Finished Applying XDC Timing Constraints : Time (s): cpu = 00:00:35 ; elapsed = 00:01:39 . Memory (MB): peak =
1908.156 ; gain = 526.293 ; free physical = 70656 ; free virtual = 153568
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
Start Timing Optimization
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
Finished Timing Optimization : Time (s): cpu = 00:00:37 ; elapsed = 00:01:40 . Memory (MB): peak = 1908.156 ;
gain = 526.293 ; free physical = 70658 ; free virtual = 153570
---------------------------------------------------------------------------------

Report RTL Partitions:
+-+--------------+------------+----------+
| |RTL Partition |Replication |Instances |
+-+--------------+------------+----------+
+-+--------------+------------+----------+
---------------------------------------------------------------------------------
Start Technology Mapping
---------------------------------------------------------------------------------
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[6][8]' (FDCE) to
'pm_reg_file/registers_reg[7][8]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[7][8] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[6][9]' (FDCE) to
'pm_reg_file/registers_reg[7][9]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[7][9] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[6][10]' (FDCE) to
'pm_reg_file/registers_reg[7][10]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[7][10] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[6][11]' (FDCE) to
'pm_reg_file/registers_reg[7][11]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[7][11] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[6][20]' (FDCE) to
'pm_reg_file/registers_reg[7][20]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[7][20] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[6][21]' (FDCE) to
'pm_reg_file/registers_reg[7][21]'
```

```
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[7][21] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[6][22]' (FDCE) to
'pm_reg_file/registers_reg[7][22]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[7][22] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[6][23]' (FDCE) to
'pm_reg_file/registers_reg[7][23]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[7][23] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[6][4]' (FDCE) to
'pm_reg_file/registers_reg[7][4]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[7][4] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[6][5]' (FDCE) to
'pm_reg_file/registers_reg[7][5]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[7][5] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[6][6]' (FDCE) to
'pm_reg_file/registers_reg[7][6]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[7][6] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[6][7]' (FDCE) to
'pm_reg_file/registers_reg[7][7]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[7][7] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[6][12]' (FDCE) to
'pm_reg_file/registers_reg[7][12]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[7][12] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[6][13]' (FDCE) to
'pm_reg_file/registers_reg[7][13]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[7][13] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[6][14]' (FDCE) to
'pm_reg_file/registers_reg[7][14]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[7][14] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[6][15]' (FDCE) to
'pm_reg_file/registers_reg[7][15]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[7][15] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[6][16]' (FDCE) to
'pm_reg_file/registers_reg[7][16]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[7][16] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[6][17]' (FDCE) to
'pm_reg_file/registers_reg[7][17]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[7][17] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[6][18]' (FDCE) to
'pm_reg_file/registers_reg[7][18]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[7][18] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[6][19]' (FDCE) to
'pm_reg_file/registers_reg[7][19]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[7][19] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[6][24]' (FDCE) to
'pm_reg_file/registers_reg[7][24]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[7][24] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[6][25]' (FDCE) to
'pm_reg_file/registers_reg[7][25]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[7][25] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[6][26]' (FDCE) to
'pm_reg_file/registers_reg[7][26]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[7][26] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[6][27]' (FDCE) to
'pm_reg_file/registers_reg[7][27]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[7][27] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[6][28]' (FDCE) to
'pm_reg_file/registers_reg[7][28]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[7][28] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[6][29]' (FDCE) to
'pm_reg_file/registers_reg[7][29]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[7][29] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[6][30]' (FDCE) to
'pm_reg_file/registers_reg[7][30]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[7][30] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[6][31]' (FDCE) to
'pm_reg_file/registers_reg[7][31]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[7][31] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[6][0]' (FDCE) to
'pm_reg_file/registers_reg[7][0]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[7][0] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[6][1]' (FDCE) to
'pm_reg_file/registers_reg[7][1]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[7][1] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[6][2]' (FDCE) to
'pm_reg_file/registers_reg[7][2]'
```

```
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[7][2] )
INFO: [Synth 8-3886] merging instance 'pm_reg_file/registers_reg[6][3]' (FDCE) to
'pm_reg_file/registers_reg[7][3]'
INFO: [Synth 8-3333] propagating constant 0 across sequential element (\pm_reg_file/registers_reg[7][3] )
---------------------------------------------------------------------------------
Finished Technology Mapping : Time (s): cpu = 00:00:38 ; elapsed = 00:01:41 . Memory (MB): peak = 1923.773 ;
gain = 541.910 ; free physical = 70658 ; free virtual = 153570
---------------------------------------------------------------------------------

Report RTL Partitions:
+-+--------------+------------+----------+
| |RTL Partition |Replication |Instances |
+-+--------------+------------+----------+
+-+--------------+------------+----------+
---------------------------------------------------------------------------------
Start IO Insertion
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
Start Flattening Before IO Insertion
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
Finished Flattening Before IO Insertion
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
Start Final Netlist Cleanup
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
Finished Final Netlist Cleanup
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
Finished IO Insertion : Time (s): cpu = 00:00:38 ; elapsed = 00:01:42 . Memory (MB): peak = 1923.777 ; gain =
541.914 ; free physical = 70658 ; free virtual = 153570
---------------------------------------------------------------------------------

Report Check Netlist:
+------+------------------+-------+---------+-------+------------------+
|      |Item              |Errors |Warnings |Status |Description       |
+------+------------------+-------+---------+-------+------------------+
|1     |multi_driven_nets |      0|        0|Passed |Multi driven nets |
+------+------------------+-------+---------+-------+------------------+
---------------------------------------------------------------------------------
Start Renaming Generated Instances
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
Finished Renaming Generated Instances : Time (s): cpu = 00:00:38 ; elapsed = 00:01:42 . Memory (MB): peak =
1923.777 ; gain = 541.914 ; free physical = 70658 ; free virtual = 153570
---------------------------------------------------------------------------------

Report RTL Partitions:
+-+--------------+------------+----------+
| |RTL Partition |Replication |Instances |
+-+--------------+------------+----------+
+-+--------------+------------+----------+
---------------------------------------------------------------------------------
Start Rebuilding User Hierarchy
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
Finished Rebuilding User Hierarchy : Time (s): cpu = 00:00:38 ; elapsed = 00:01:42 . Memory (MB): peak =
1923.777 ; gain = 541.914 ; free physical = 70658 ; free virtual = 153570
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
Start Renaming Generated Ports
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
Finished Renaming Generated Ports : Time (s): cpu = 00:00:38 ; elapsed = 00:01:42 . Memory (MB): peak =
1923.777 ; gain = 541.914 ; free physical = 70658 ; free virtual = 153570
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
Start Handling Custom Attributes
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
Finished Handling Custom Attributes : Time (s): cpu = 00:00:38 ; elapsed = 00:01:42 . Memory (MB): peak =
1923.777 ; gain = 541.914 ; free physical = 70658 ; free virtual = 153570
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
```

```
Start Renaming Generated Nets
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
Finished Renaming Generated Nets : Time (s): cpu = 00:00:38 ; elapsed = 00:01:42 . Memory (MB): peak =
1923.777 ; gain = 541.914 ; free physical = 70658 ; free virtual = 153570
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
Start Writing Synthesis Report
---------------------------------------------------------------------------------

Report BlackBoxes:
+-+--------------+----------+
| |BlackBox name |Instances |
+-+--------------+----------+
+-+--------------+----------+

Report Cell Usage:
+------+-------+------+
|      |Cell   |Count |
+------+-------+------+
|1     |BUFG   |     1|
|2     |CARRY4 |    13|
|3     |LUT2   |    13|
|4     |LUT3   |     1|
|5     |LUT4   |    65|
|6     |LUT5   |   125|
|7     |LUT6   |   451|
|8     |MUXF7  |   163|
|9     |MUXF8  |    62|
|10    |FDCE   |  1221|
|11    |IBUF   |     2|
|12    |OBUF   |    14|
+------+-------+------+

Report Instance Areas:
+------+------------------+------------+------+
|      |Instance          |Module      |Cells |
+------+------------------+------------+------+
|1     |top               |            |  2131|
|2     |  pm_alu          |alu         |    17|
|3     |  pm_d_cache      |d_cache     |  1470|
|4     |  pm_next_address |next_address |     2|
|5     |  pm_pc           |pc          |   369|
|6     |  pm_reg_file     |regfile     |   256|
+------+------------------+------------+------+
---------------------------------------------------------------------------------
Finished Writing Synthesis Report : Time (s): cpu = 00:00:38 ; elapsed = 00:01:42 . Memory (MB): peak =
1923.777 ; gain = 541.914 ; free physical = 70658 ; free virtual = 153570
---------------------------------------------------------------------------------
Synthesis finished with 0 errors, 0 critical warnings and 829 warnings.
Synthesis Optimization Runtime : Time (s): cpu = 00:00:28 ; elapsed = 00:00:46 . Memory (MB): peak = 1923.777 ;
gain = 162.004 ; free physical = 70712 ; free virtual = 153624
Synthesis Optimization Complete : Time (s): cpu = 00:00:38 ; elapsed = 00:01:42 . Memory (MB): peak =
1923.781 ; gain = 541.914 ; free physical = 70723 ; free virtual = 153635
INFO: [Project 1-571] Translating synthesized netlist
INFO: [Netlist 29-17] Analyzing 240 Unisim elements for replacement
INFO: [Netlist 29-28] Unisim Transformation completed in 0 CPU seconds
INFO: [Project 1-570] Preparing netlist for logic optimization
INFO: [Opt 31-138] Pushed 0 inverter(s) to 0 load pin(s).
INFO: [Project 1-111] Unisim Transformation Summary:
No Unisim elements were transformed.

INFO: [Common 17-83] Releasing license: Synthesis
267 Infos, 104 Warnings, 0 Critical Warnings and 0 Errors encountered.
synth_design completed successfully
synth_design: Time (s): cpu = 00:00:40 ; elapsed = 00:01:44 . Memory (MB): peak = 1955.793 ; gain = 586.656 ;
free physical = 70709 ; free virtual = 153621
WARNING: [Constraints 18-5210] No constraint will be written out.
INFO: [Common 17-1381] The checkpoint
'/nfs/home/l/l_heiwan/coen316/lab4/vivado/part2/lab4_part2/lab4_part2.runs/synth_1/cpu.dcp' has been generated.
INFO: [runtcl-4] Executing : report_utilization -file cpu_utilization_synth.rpt -pb cpu_utilization_synth.pb
report_utilization: Time (s): cpu = 00:00:00.07 ; elapsed = 00:00:00.14 . Memory (MB): peak = 1979.812 ; gain =
0.000 ; free physical = 70708 ; free virtual = 153620
INFO: [Common 17-206] Exiting Vivado at Wed Nov 22 20:05:28 2023...
```

5.4) Implementation Log

```
*** Running vivado
    with args -log cpu.vdi -applog -m64 -product Vivado -messageDb vivado.pb -mode batch -source cpu.tcl -
notrace


****** Vivado v2018.2 (64-bit)
  **** SW Build 2258646 on Thu Jun 14 20:02:38 MDT 2018
  **** IP Build 2256618 on Thu Jun 14 22:10:49 MDT 2018
    ** Copyright 1986-2018 Xilinx, Inc. All Rights Reserved.

source cpu.tcl -notrace
Command: link_design -top cpu -part xc7a100tcsg324-1
Design is defaulting to srcset: sources_1
Design is defaulting to constrset: constrs_1
INFO: [Netlist 29-17] Analyzing 240 Unisim elements for replacement
INFO: [Netlist 29-28] Unisim Transformation completed in 0 CPU seconds
INFO: [Project 1-479] Netlist was created with Vivado 2018.2
INFO: [Device 21-403] Loading part xc7a100tcsg324-1
INFO: [Project 1-570] Preparing netlist for logic optimization
Parsing XDC File [/nfs/home/l/l_heiwan/coen316/lab4/givenToUs/cpu.xdc]
Finished Parsing XDC File [/nfs/home/l/l_heiwan/coen316/lab4/givenToUs/cpu.xdc]
INFO: [Opt 31-138] Pushed 0 inverter(s) to 0 load pin(s).
INFO: [Project 1-111] Unisim Transformation Summary:
No Unisim elements were transformed.

7 Infos, 0 Warnings, 0 Critical Warnings and 0 Errors encountered.
link_design completed successfully
link_design: Time (s): cpu = 00:00:11 ; elapsed = 00:00:58 . Memory (MB): peak = 1704.566 ; gain = 330.414 ;
free physical = 70839 ; free virtual = 153748
Command: opt_design
Attempting to get a license for feature 'Implementation' and/or device 'xc7a100t'
INFO: [Common 17-349] Got license for feature 'Implementation' and/or device 'xc7a100t'
Running DRC as a precondition to command opt_design

Starting DRC Task
INFO: [DRC 23-27] Running DRC with 8 threads
INFO: [Project 1-461] DRC finished with 0 Errors
INFO: [Project 1-462] Please refer to the DRC report (report_drc) for more information.

Time (s): cpu = 00:00:02 ; elapsed = 00:00:04 . Memory (MB): peak = 1778.594 ; gain = 74.027 ; free physical =
70828 ; free virtual = 153737

Starting Cache Timing Information Task
INFO: [Timing 38-35] Done setting XDC timing constraints.
Ending Cache Timing Information Task | Checksum: 132e5b778

Time (s): cpu = 00:00:15 ; elapsed = 00:01:08 . Memory (MB): peak = 2251.094 ; gain = 472.500 ; free physical =
70383 ; free virtual = 153292

Starting Logic Optimization Task

Phase 1 Retarget
INFO: [Opt 31-138] Pushed 0 inverter(s) to 0 load pin(s).
INFO: [Opt 31-49] Retargeted 0 cell(s).
Phase 1 Retarget | Checksum: 132e5b778

Time (s): cpu = 00:00:00.14 ; elapsed = 00:00:00.09 . Memory (MB): peak = 2251.094 ; gain = 0.000 ; free
physical = 70399 ; free virtual = 153308
INFO: [Opt 31-389] Phase Retarget created 0 cells and removed 0 cells

Phase 2 Constant propagation
INFO: [Opt 31-138] Pushed 0 inverter(s) to 0 load pin(s).
Phase 2 Constant propagation | Checksum: 132e5b778

Time (s): cpu = 00:00:00.18 ; elapsed = 00:00:00.12 . Memory (MB): peak = 2251.094 ; gain = 0.000 ; free
physical = 70399 ; free virtual = 153308
INFO: [Opt 31-389] Phase Constant propagation created 0 cells and removed 0 cells

Phase 3 Sweep
Phase 3 Sweep | Checksum: 120e2a9de
```

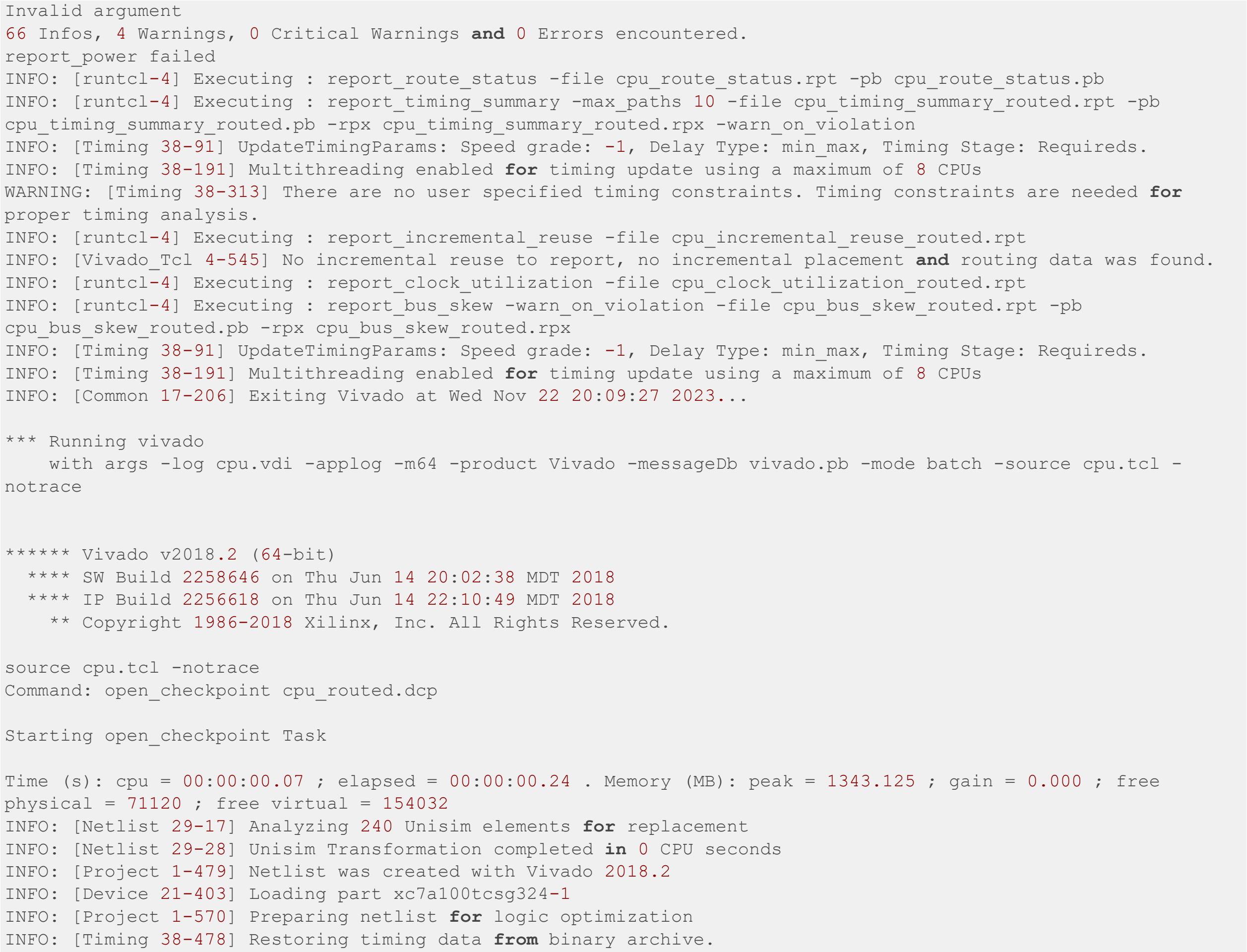
```
Time (s): cpu = 00:00:00.20 ; elapsed = 00:00:00.15 . Memory (MB): peak = 2251.094 ; gain = 0.000 ; free
physical = 70399 ; free virtual = 153308
INFO: [Opt 31-389] Phase Sweep created 0 cells and removed 0 cells

Phase 4 BUFG optimization
Phase 4 BUFG optimization | Checksum: 120e2a9de

Time (s): cpu = 00:00:00.23 ; elapsed = 00:00:00.18 . Memory (MB): peak = 2251.094 ; gain = 0.000 ; free
physical = 70399 ; free virtual = 153308
INFO: [Opt 31-662] Phase BUFG optimization created 0 cells of which 0 are BUFGs and removed 0 cells.

Phase 5 Shift Register Optimization
Phase 5 Shift Register Optimization | Checksum: 15f26dba6

Time (s): cpu = 00:00:00.26 ; elapsed = 00:00:00.21 . Memory (MB): peak = 2251.094 ; gain = 0.000 ; free
physical = 70399 ; free virtual = 153308
INFO: [Opt 31-389] Phase Shift Register Optimization created 0 cells and removed 0 cells

Phase 6 Post Processing Netlist
Phase 6 Post Processing Netlist | Checksum: 15f26dba6

Time (s): cpu = 00:00:00.27 ; elapsed = 00:00:00.22 . Memory (MB): peak = 2251.094 ; gain = 0.000 ; free
physical = 70399 ; free virtual = 153308
INFO: [Opt 31-389] Phase Post Processing Netlist created 0 cells and removed 0 cells

Starting Connectivity Check Task

Time (s): cpu = 00:00:00 ; elapsed = 00:00:00 . Memory (MB): peak = 2251.094 ; gain = 0.000 ; free physical =
70399 ; free virtual = 153308
Ending Logic Optimization Task | Checksum: 15f26dba6

Time (s): cpu = 00:00:00.28 ; elapsed = 00:00:00.23 . Memory (MB): peak = 2251.094 ; gain = 0.000 ; free
physical = 70399 ; free virtual = 153308

Starting Power Optimization Task
INFO: [Pwropt 34-132] Skipping clock gating for clocks with a period < 2.00 ns.
Ending Power Optimization Task | Checksum: 15f26dba6

Time (s): cpu = 00:00:00.02 ; elapsed = 00:00:00.06 . Memory (MB): peak = 2251.094 ; gain = 0.000 ; free
physical = 70399 ; free virtual = 153308

Starting Final Cleanup Task
Ending Final Cleanup Task | Checksum: 15f26dba6

Time (s): cpu = 00:00:00 ; elapsed = 00:00:00 . Memory (MB): peak = 2251.094 ; gain = 0.000 ; free physical =
70399 ; free virtual = 153308
INFO: [Common 17-83] Releasing license: Implementation
23 Infos, 0 Warnings, 0 Critical Warnings and 0 Errors encountered.
opt_design completed successfully
opt_design: Time (s): cpu = 00:00:17 ; elapsed = 00:01:12 . Memory (MB): peak = 2251.094 ; gain = 546.527 ;
free physical = 70399 ; free virtual = 153308
INFO: [Timing 38-480] Writing timing data to binary archive.
Writing placer database...
Writing XDEF routing.
Writing XDEF routing logical nets.
Writing XDEF routing special nets.
Write XDEF Complete: Time (s): cpu = 00:00:00.06 ; elapsed = 00:00:00.02 . Memory (MB): peak = 2283.105 ; gain
= 0.000 ; free physical = 70395 ; free virtual = 153305
INFO: [Common 17-1381] The checkpoint
'/nfs/home/l/l_heiwan/coen316/lab4/vivado/part2/lab4_part2/lab4_part2.runs/impl_1/cpu_opt.dcp' has been
generated.
INFO: [runtcl-4] Executing : report_drc -file cpu_drc_opted.rpt -pb cpu_drc_opted.pb -rpx cpu_drc_opted.rpx
Command: report_drc -file cpu_drc_opted.rpt -pb cpu_drc_opted.pb -rpx cpu_drc_opted.rpx
INFO: [IP_Flow 19-234] Refreshing IP repositories
INFO: [IP_Flow 19-1704] No user IP repositories specified
INFO: [IP_Flow 19-2313] Loaded Vivado IP repository '/CMC/tools/xilinx/Vivado_2018.2/Vivado/2018.2/data/ip'.
INFO: [DRC 23-27] Running DRC with 8 threads
INFO: [Coretcl 2-168] The results of DRC are in file
/nfs/home/l/l_heiwan/coen316/lab4/vivado/part2/lab4_part2/lab4_part2.runs/impl_1/cpu_drc_opted.rpt.
report_drc completed successfully
report_drc: Time (s): cpu = 00:00:04 ; elapsed = 00:00:07 . Memory (MB): peak = 2363.148 ; gain = 80.031 ; free
physical = 70401 ; free virtual = 153310
Command: place_design
Attempting to get a license for feature 'Implementation' and/or device 'xc7a100t'
```

```
INFO: [Common 17-349] Got license for feature 'Implementation' and/or device 'xc7a100t'
INFO: [DRC 23-27] Running DRC with 8 threads
INFO: [Vivado_Tcl 4-198] DRC finished with 0 Errors
INFO: [Vivado_Tcl 4-199] Please refer to the DRC report (report_drc) for more information.
Running DRC as a precondition to command place_design
INFO: [DRC 23-27] Running DRC with 8 threads
INFO: [Vivado_Tcl 4-198] DRC finished with 0 Errors
INFO: [Vivado_Tcl 4-199] Please refer to the DRC report (report_drc) for more information.

Starting Placer Task
INFO: [Place 30-611] Multithreading enabled for place_design using a maximum of 8 CPUs

Phase 1 Placer Initialization

Phase 1.1 Placer Initialization Netlist Sorting
Netlist sorting complete. Time (s): cpu = 00:00:00 ; elapsed = 00:00:00 . Memory (MB): peak = 2363.148 ; gain =
0.000 ; free physical = 70392 ; free virtual = 153302
Phase 1.1 Placer Initialization Netlist Sorting | Checksum: ca97ba35

Time (s): cpu = 00:00:00.01 ; elapsed = 00:00:00.02 . Memory (MB): peak = 2363.148 ; gain = 0.000 ; free
physical = 70392 ; free virtual = 153302
Netlist sorting complete. Time (s): cpu = 00:00:00 ; elapsed = 00:00:00.01 . Memory (MB): peak = 2363.148 ;
gain = 0.000 ; free physical = 70392 ; free virtual = 153302

Phase 1.2 IO Placement/ Clock Placement/ Build Placer Device
INFO: [Timing 38-35] Done setting XDC timing constraints.
WARNING: [Place 30-574] Poor placement for routing between an IO pin and BUFG. This is normally an ERROR but
the CLOCK_DEDICATED_ROUTE constraint is set to FALSE allowing your design to continue. The use of this override
is highly discouraged as it may lead to very poor timing results. It is recommended that this error condition
be corrected in the design.

        clk_IBUF_inst (IBUF.O) is locked to IOB_X0Y82
        clk_IBUF_BUFG_inst (BUFG.I) is provisionally placed by clockplacer on BUFGCTRL_X0Y0
Resolution: Poor placement of an IO pin and a BUFG has resulted in the router using a non-dedicated path
between the two.  There are several things that could trigger this DRC, each of which can cause unpredictable
clock insertion delays that result in poor timing.  This DRC could be caused by any of the following: (a) a
clock port was placed on a pin that is not a CCIO-pin (b)the BUFG has not been placed in the same half of the
device or SLR as the CCIO-pin (c) a single ended clock has been placed on the N-Side of a differential pair
CCIO-pin.
Phase 1.2 IO Placement/ Clock Placement/ Build Placer Device | Checksum: 4eb80967

Time (s): cpu = 00:00:01 ; elapsed = 00:00:00.85 . Memory (MB): peak = 2363.148 ; gain = 0.000 ; free physical
= 70390 ; free virtual = 153300

Phase 1.3 Build Placer Netlist Model
Phase 1.3 Build Placer Netlist Model | Checksum: 14e35c75a

Time (s): cpu = 00:00:02 ; elapsed = 00:00:00.92 . Memory (MB): peak = 2363.148 ; gain = 0.000 ; free physical
= 70391 ; free virtual = 153301

Phase 1.4 Constrain Clocks/Macros
Phase 1.4 Constrain Clocks/Macros | Checksum: 14e35c75a

Time (s): cpu = 00:00:02 ; elapsed = 00:00:00.92 . Memory (MB): peak = 2363.148 ; gain = 0.000 ; free physical
= 70391 ; free virtual = 153301
Phase 1 Placer Initialization | Checksum: 14e35c75a

Time (s): cpu = 00:00:02 ; elapsed = 00:00:00.93 . Memory (MB): peak = 2363.148 ; gain = 0.000 ; free physical
= 70391 ; free virtual = 153301

Phase 2 Global Placement

Phase 2.1 Floorplanning
Phase 2.1 Floorplanning | Checksum: 14e35c75a

Time (s): cpu = 00:00:02 ; elapsed = 00:00:00.95 . Memory (MB): peak = 2363.148 ; gain = 0.000 ; free physical
= 70395 ; free virtual = 153304
WARNING: [Place 46-29] place_design is not in timing mode. Skip physical synthesis in placer
Phase 2 Global Placement | Checksum: fc646420

Time (s): cpu = 00:00:09 ; elapsed = 00:00:03 . Memory (MB): peak = 2418.168 ; gain = 55.020 ; free physical =
70353 ; free virtual = 153262

Phase 3 Detail Placement
Phase 3.1 Commit Multi Column Macros
Phase 3.1 Commit Multi Column Macros | Checksum: fc646420

Time (s): cpu = 00:00:09 ; elapsed = 00:00:03 . Memory (MB): peak = 2418.168 ; gain = 55.020 ; free physical =
70353 ; free virtual = 153262

Phase 3.2 Commit Most Macros & LUTRAMs
Phase 3.2 Commit Most Macros & LUTRAMs | Checksum: 120db6594

Time (s): cpu = 00:00:09 ; elapsed = 00:00:03 . Memory (MB): peak = 2418.168 ; gain = 55.020 ; free physical =
70352 ; free virtual = 153262

Phase 3.3 Area Swap Optimization
Phase 3.3 Area Swap Optimization | Checksum: 12cbbff89

Time (s): cpu = 00:00:09 ; elapsed = 00:00:03 . Memory (MB): peak = 2418.168 ; gain = 55.020 ; free physical =
70352 ; free virtual = 153262

Phase 3.4 Pipeline Register Optimization
Phase 3.4 Pipeline Register Optimization | Checksum: 12cbbff89

Time (s): cpu = 00:00:09 ; elapsed = 00:00:03 . Memory (MB): peak = 2418.168 ; gain = 55.020 ; free physical =
70352 ; free virtual = 153262

Phase 3.5 Small Shape Detail Placement
Phase 3.5 Small Shape Detail Placement | Checksum: 144f226f7

Time (s): cpu = 00:00:10 ; elapsed = 00:00:04 . Memory (MB): peak = 2418.168 ; gain = 55.020 ; free physical =
70354 ; free virtual = 153264

Phase 3.6 Re-assign LUT pins
Phase 3.6 Re-assign LUT pins | Checksum: 144f226f7

Time (s): cpu = 00:00:10 ; elapsed = 00:00:04 . Memory (MB): peak = 2418.168 ; gain = 55.020 ; free physical =
70354 ; free virtual = 153264

Phase 3.7 Pipeline Register Optimization
Phase 3.7 Pipeline Register Optimization | Checksum: 144f226f7

Time (s): cpu = 00:00:10 ; elapsed = 00:00:04 . Memory (MB): peak = 2418.168 ; gain = 55.020 ; free physical =
70354 ; free virtual = 153264
Phase 3 Detail Placement | Checksum: 144f226f7

Time (s): cpu = 00:00:10 ; elapsed = 00:00:04 . Memory (MB): peak = 2418.168 ; gain = 55.020 ; free physical =
70354 ; free virtual = 153264

Phase 4 Post Placement Optimization and Clean-Up

Phase 4.1 Post Commit Optimization
Phase 4.1 Post Commit Optimization | Checksum: 144f226f7

Time (s): cpu = 00:00:10 ; elapsed = 00:00:04 . Memory (MB): peak = 2418.168 ; gain = 55.020 ; free physical =
70354 ; free virtual = 153264

Phase 4.2 Post Placement Cleanup
Phase 4.2 Post Placement Cleanup | Checksum: 144f226f7

Time (s): cpu = 00:00:10 ; elapsed = 00:00:04 . Memory (MB): peak = 2418.168 ; gain = 55.020 ; free physical =
70354 ; free virtual = 153263

Phase 4.3 Placer Reporting
Phase 4.3 Placer Reporting | Checksum: 144f226f7

Time (s): cpu = 00:00:10 ; elapsed = 00:00:04 . Memory (MB): peak = 2418.168 ; gain = 55.020 ; free physical =
70354 ; free virtual = 153263

Phase 4.4 Final Placement Cleanup
Phase 4.4 Final Placement Cleanup | Checksum: 1b809c133

Time (s): cpu = 00:00:10 ; elapsed = 00:00:04 . Memory (MB): peak = 2418.168 ; gain = 55.020 ; free physical =
70354 ; free virtual = 153263
Phase 4 Post Placement Optimization and Clean-Up | Checksum: 1b809c133

Time (s): cpu = 00:00:10 ; elapsed = 00:00:04 . Memory (MB): peak = 2418.168 ; gain = 55.020 ; free physical =
70354 ; free virtual = 153263
```

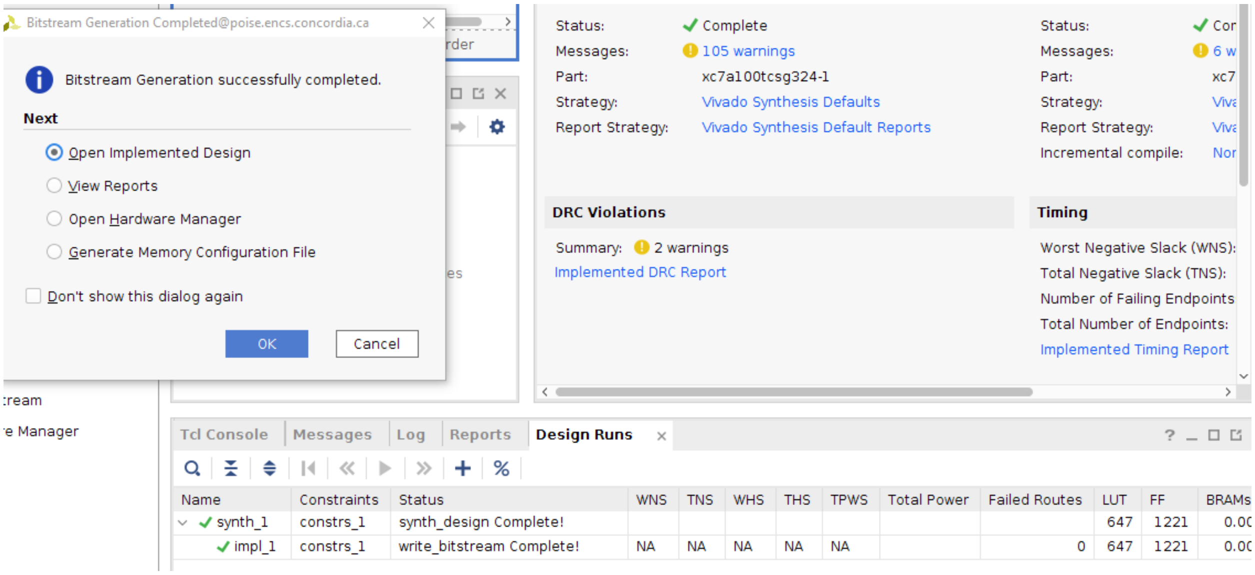
```
Ending Placer Task | Checksum: f6ad9808

Time (s): cpu = 00:00:10 ; elapsed = 00:00:04 . Memory (MB): peak = 2418.168 ; gain = 55.020 ; free physical =
70372 ; free virtual = 153282
INFO: [Common 17-83] Releasing license: Implementation
41 Infos, 2 Warnings, 0 Critical Warnings and 0 Errors encountered.
place_design completed successfully
place_design: Time (s): cpu = 00:00:12 ; elapsed = 00:00:08 . Memory (MB): peak = 2418.168 ; gain = 55.020 ;
free physical = 70372 ; free virtual = 153282
INFO: [Timing 38-480] Writing timing data to binary archive.
Writing placer database...
Writing XDEF routing.
Writing XDEF routing logical nets.
Writing XDEF routing special nets.
Write XDEF Complete: Time (s): cpu = 00:00:00.37 ; elapsed = 00:00:00.13 . Memory (MB): peak = 2418.168 ; gain
= 0.000 ; free physical = 70366 ; free virtual = 153278
INFO: [Common 17-1381] The checkpoint
'/nfs/home/l/l_heiwan/coen316/lab4/vivado/part2/lab4_part2/lab4_part2.runs/impl_1/cpu_placed.dcp' has been
generated.
INFO: [runtcl-4] Executing : report_io -file cpu_io_placed.rpt
report_io: Time (s): cpu = 00:00:00.07 ; elapsed = 00:00:00.17 . Memory (MB): peak = 2418.168 ; gain = 0.000 ;
free physical = 70366 ; free virtual = 153276
INFO: [runtcl-4] Executing : report_utilization -file cpu_utilization_placed.rpt -pb cpu_utilization_placed.pb
report_utilization: Time (s): cpu = 00:00:00.07 ; elapsed = 00:00:00.14 . Memory (MB): peak = 2418.168 ; gain =
0.000 ; free physical = 70370 ; free virtual = 153281
INFO: [runtcl-4] Executing : report_control_sets -verbose -file cpu_control_sets_placed.rpt
report_control_sets: Time (s): cpu = 00:00:00.03 ; elapsed = 00:00:00.10 . Memory (MB): peak = 2418.168 ; gain
= 0.000 ; free physical = 70369 ; free virtual = 153279
Command: route_design
Attempting to get a license for feature 'Implementation' and/or device 'xc7a100t'
INFO: [Common 17-349] Got license for feature 'Implementation' and/or device 'xc7a100t'
Running DRC as a precondition to command route_design
INFO: [DRC 23-27] Running DRC with 8 threads
WARNING: [DRC PLCK-12] Clock Placer Checks: Poor placement for routing between an IO pin and BUFG.
Resolution: Poor placement of an IO pin and a BUFG has resulted in the router using a non-dedicated path
between the two.  There are several things that could trigger this DRC, each of which can cause unpredictable
clock insertion delays that result in poor timing.  This DRC could be caused by any of the following: (a) a
clock port was placed on a pin that is not a CCIO-pin (b)the BUFG has not been placed in the same half of the
device or SLR as the CCIO-pin (c) a single ended clock has been placed on the N-Side of a differential pair
CCIO-pin.
 This is normally an ERROR but the CLOCK_DEDICATED_ROUTE constraint is set to FALSE allowing your design to
continue. The use of this override is highly discouraged as it may lead to very poor timing results. It is
recommended that this error condition be corrected in the design.

        clk_IBUF_inst (IBUF.O) is locked to IOB_X0Y82
        clk_IBUF_BUFG_inst (BUFG.I) is provisionally placed by clockplacer on BUFGCTRL_X0Y0
INFO: [Vivado_Tcl 4-198] DRC finished with 0 Errors, 1 Warnings
INFO: [Vivado_Tcl 4-199] Please refer to the DRC report (report_drc) for more information.


Starting Routing Task
INFO: [Route 35-254] Multithreading enabled for route_design using a maximum of 8 CPUs
Checksum: PlaceDB: 53719dcc ConstDB: 0 ShapeSum: a33bfa3c RouteDB: 0

Phase 1 Build RT Design
Phase 1 Build RT Design | Checksum: 12884b526

Time (s): cpu = 00:00:24 ; elapsed = 00:00:21 . Memory (MB): peak = 2460.773 ; gain = 42.605 ; free physical =
70210 ; free virtual = 153121
Post Restoration Checksum: NetGraph: 2b6571ad NumContArr: fd1f4379 Constraints: 0 Timing: 0

Phase 2 Router Initialization
INFO: [Route 35-64] No timing constraints were detected. The router will operate in resource-optimization mode.

Phase 2.1 Fix Topology Constraints
Phase 2.1 Fix Topology Constraints | Checksum: 12884b526

Time (s): cpu = 00:00:24 ; elapsed = 00:00:21 . Memory (MB): peak = 2466.762 ; gain = 48.594 ; free physical =
70179 ; free virtual = 153089

Phase 2.2 Pre Route Cleanup
Phase 2.2 Pre Route Cleanup | Checksum: 12884b526

Time (s): cpu = 00:00:24 ; elapsed = 00:00:21 . Memory (MB): peak = 2466.762 ; gain = 48.594 ; free physical =
70179 ; free virtual = 153089
```

```
 Number of Nodes with overlaps = 0
Phase 2 Router Initialization | Checksum: 1612e7d2a

Time (s): cpu = 00:00:25 ; elapsed = 00:00:21 . Memory (MB): peak = 2476.027 ; gain = 57.859 ; free physical =
70169 ; free virtual = 153080

Phase 3 Initial Routing
Phase 3 Initial Routing | Checksum: f5895377

Time (s): cpu = 00:00:26 ; elapsed = 00:00:21 . Memory (MB): peak = 2480.027 ; gain = 61.859 ; free physical =
70169 ; free virtual = 153080

Phase 4 Rip-up And Reroute

Phase 4.1 Global Iteration 0
 Number of Nodes with overlaps = 150
 Number of Nodes with overlaps = 1
 Number of Nodes with overlaps = 0
Phase 4.1 Global Iteration 0 | Checksum: 14242920f

Time (s): cpu = 00:00:28 ; elapsed = 00:00:22 . Memory (MB): peak = 2480.027 ; gain = 61.859 ; free physical =
70168 ; free virtual = 153079
Phase 4 Rip-up And Reroute | Checksum: 14242920f

Time (s): cpu = 00:00:28 ; elapsed = 00:00:22 . Memory (MB): peak = 2480.027 ; gain = 61.859 ; free physical =
70168 ; free virtual = 153079

Phase 5 Delay and Skew Optimization
Phase 5 Delay and Skew Optimization | Checksum: 14242920f

Time (s): cpu = 00:00:28 ; elapsed = 00:00:22 . Memory (MB): peak = 2480.027 ; gain = 61.859 ; free physical =
70168 ; free virtual = 153079

Phase 6 Post Hold Fix

Phase 6.1 Hold Fix Iter
Phase 6.1 Hold Fix Iter | Checksum: 14242920f

Time (s): cpu = 00:00:28 ; elapsed = 00:00:22 . Memory (MB): peak = 2480.027 ; gain = 61.859 ; free physical =
70168 ; free virtual = 153079
Phase 6 Post Hold Fix | Checksum: 14242920f

Time (s): cpu = 00:00:28 ; elapsed = 00:00:22 . Memory (MB): peak = 2480.027 ; gain = 61.859 ; free physical =
70168 ; free virtual = 153079

Phase 7 Route finalize

Router Utilization Summary
  Global Vertical Routing Utilization    = 0.296644 %
  Global Horizontal Routing Utilization  = 0.356067 %
  Routable Net Status*
  *Does not include unroutable nets such as driverless and loadless.
  Run report_route_status for detailed report.
  Number of Failed Nets               = 0
  Number of Unrouted Nets             = 0
  Number of Partially Routed Nets     = 0
  Number of Node Overlaps             = 0

Congestion Report
North Dir 1x1 Area, Max Cong = 34.2342%, No Congested Regions.
South Dir 1x1 Area, Max Cong = 44.1441%, No Congested Regions.
East Dir 1x1 Area, Max Cong = 42.6471%, No Congested Regions.
West Dir 1x1 Area, Max Cong = 38.2353%, No Congested Regions.

------------------------------
Reporting congestion hotspots
------------------------------
Direction: North
----------------
Congested clusters found at Level 0
Effective congestion level: 0 Aspect Ratio: 1 Sparse Ratio: 0
Direction: South
----------------
Congested clusters found at Level 0
Effective congestion level: 0 Aspect Ratio: 1 Sparse Ratio: 0
```

```
Direction: East
----------------
Congested clusters found at Level 0
Effective congestion level: 0 Aspect Ratio: 1 Sparse Ratio: 0
Direction: West
----------------
Congested clusters found at Level 0
Effective congestion level: 0 Aspect Ratio: 1 Sparse Ratio: 0

Phase 7 Route finalize | Checksum: 14242920f

Time (s): cpu = 00:00:28 ; elapsed = 00:00:22 . Memory (MB): peak = 2480.027 ; gain = 61.859 ; free physical =
70169 ; free virtual = 153079

Phase 8 Verifying routed nets

 Verification completed successfully
Phase 8 Verifying routed nets | Checksum: 14242920f

Time (s): cpu = 00:00:28 ; elapsed = 00:00:22 . Memory (MB): peak = 2482.027 ; gain = 63.859 ; free physical =
70168 ; free virtual = 153079

Phase 9 Depositing Routes
Phase 9 Depositing Routes | Checksum: 16ee01053

Time (s): cpu = 00:00:28 ; elapsed = 00:00:22 . Memory (MB): peak = 2482.027 ; gain = 63.859 ; free physical =
70169 ; free virtual = 153079
INFO: [Route 35-16] Router Completed Successfully

Time (s): cpu = 00:00:28 ; elapsed = 00:00:22 . Memory (MB): peak = 2482.027 ; gain = 63.859 ; free physical =
70203 ; free virtual = 153113

Routing Is Done.
INFO: [Common 17-83] Releasing license: Implementation
54 Infos, 3 Warnings, 0 Critical Warnings and 0 Errors encountered.
route_design completed successfully
route_design: Time (s): cpu = 00:00:30 ; elapsed = 00:00:26 . Memory (MB): peak = 2482.027 ; gain = 63.859 ;
free physical = 70203 ; free virtual = 153113
INFO: [Timing 38-480] Writing timing data to binary archive.
Writing placer database...
Writing XDEF routing.
Writing XDEF routing logical nets.
Writing XDEF routing special nets.
Write XDEF Complete: Time (s): cpu = 00:00:00.34 ; elapsed = 00:00:00.13 . Memory (MB): peak = 2482.027 ; gain
= 0.000 ; free physical = 70196 ; free virtual = 153110
INFO: [Common 17-1381] The checkpoint
'/nfs/home/l/l_heiwan/coen316/lab4/vivado/part2/lab4_part2/lab4_part2.runs/impl_1/cpu_routed.dcp' has been
generated.
INFO: [runtcl-4] Executing : report_drc -file cpu_drc_routed.rpt -pb cpu_drc_routed.pb -rpx cpu_drc_routed.rpx
Command: report_drc -file cpu_drc_routed.rpt -pb cpu_drc_routed.pb -rpx cpu_drc_routed.rpx
INFO: [DRC 23-27] Running DRC with 8 threads
INFO: [Coretcl 2-168] The results of DRC are in file
/nfs/home/l/l_heiwan/coen316/lab4/vivado/part2/lab4_part2/lab4_part2.runs/impl_1/cpu_drc_routed.rpt.
report_drc completed successfully
INFO: [runtcl-4] Executing : report_methodology -file cpu_methodology_drc_routed.rpt -pb
cpu_methodology_drc_routed.pb -rpx cpu_methodology_drc_routed.rpx
Command: report_methodology -file cpu_methodology_drc_routed.rpt -pb cpu_methodology_drc_routed.pb -rpx
cpu_methodology_drc_routed.rpx
INFO: [Timing 38-35] Done setting XDC timing constraints.
INFO: [Timing 38-35] Done setting XDC timing constraints.
INFO: [DRC 23-133] Running Methodology with 8 threads
INFO: [Coretcl 2-1520] The results of Report Methodology are in file
/nfs/home/l/l_heiwan/coen316/lab4/vivado/part2/lab4_part2/lab4_part2.runs/impl_1/cpu_methodology_drc_routed.rpt
.
report_methodology completed successfully
INFO: [runtcl-4] Executing : report_power -file cpu_power_routed.rpt -pb cpu_power_summary_routed.pb -rpx
cpu_power_routed.rpx
Command: report_power -file cpu_power_routed.rpt -pb cpu_power_summary_routed.pb -rpx cpu_power_routed.rpx
WARNING: [Power 33-232] No user defined clocks were found in the design!
Resolution: Please specify clocks using create_clock/create_generated_clock for sequential elements. For pure
combinatorial circuits, please specify a virtual clock, otherwise the vectorless estimation might be inaccurate
INFO: [Timing 38-35] Done setting XDC timing constraints.
Running Vector-less Activity Propagation...

Finished Running Vector-less Activity Propagation
```



```
Invalid argument
66 Infos, 4 Warnings, 0 Critical Warnings and 0 Errors encountered.
report_power failed
INFO: [runtcl-4] Executing : report_route_status -file cpu_route_status.rpt -pb cpu_route_status.pb
INFO: [runtcl-4] Executing : report_timing_summary -max_paths 10 -file cpu_timing_summary_routed.rpt -pb
cpu_timing_summary_routed.pb -rpx cpu_timing_summary_routed.rpx -warn_on_violation
INFO: [Timing 38-91] UpdateTimingParams: Speed grade: -1, Delay Type: min_max, Timing Stage: Requireds.
INFO: [Timing 38-191] Multithreading enabled for timing update using a maximum of 8 CPUs
WARNING: [Timing 38-313] There are no user specified timing constraints. Timing constraints are needed for
proper timing analysis.
INFO: [runtcl-4] Executing : report_incremental_reuse -file cpu_incremental_reuse_routed.rpt
INFO: [Vivado_Tcl 4-545] No incremental reuse to report, no incremental placement and routing data was found.
INFO: [runtcl-4] Executing : report_clock_utilization -file cpu_clock_utilization_routed.rpt
INFO: [runtcl-4] Executing : report_bus_skew -warn_on_violation -file cpu_bus_skew_routed.rpt -pb
cpu_bus_skew_routed.pb -rpx cpu_bus_skew_routed.rpx
INFO: [Timing 38-91] UpdateTimingParams: Speed grade: -1, Delay Type: min_max, Timing Stage: Requireds.
INFO: [Timing 38-191] Multithreading enabled for timing update using a maximum of 8 CPUs
INFO: [Common 17-206] Exiting Vivado at Wed Nov 22 20:09:27 2023...

*** Running vivado
    with args -log cpu.vdi -applog -m64 -product Vivado -messageDb vivado.pb -mode batch -source cpu.tcl -
notrace


****** Vivado v2018.2 (64-bit)
  **** SW Build 2258646 on Thu Jun 14 20:02:38 MDT 2018
  **** IP Build 2256618 on Thu Jun 14 22:10:49 MDT 2018
    ** Copyright 1986-2018 Xilinx, Inc. All Rights Reserved.

source cpu.tcl -notrace
Command: open_checkpoint cpu_routed.dcp

Starting open_checkpoint Task

Time (s): cpu = 00:00:00.07 ; elapsed = 00:00:00.24 . Memory (MB): peak = 1343.125 ; gain = 0.000 ; free
physical = 71120 ; free virtual = 154032
INFO: [Netlist 29-17] Analyzing 240 Unisim elements for replacement
INFO: [Netlist 29-28] Unisim Transformation completed in 0 CPU seconds
INFO: [Project 1-479] Netlist was created with Vivado 2018.2
INFO: [Device 21-403] Loading part xc7a100tcsg324-1
INFO: [Project 1-570] Preparing netlist for logic optimization
INFO: [Timing 38-478] Restoring timing data from binary archive.
```



Bitstream Generation successfully completed.

Next

● Open Implemented Design

○ View Reports

○ Open Hardware Manager

○ Generate Memory Configuration File

□ Don't show this dialog again

OK Cancel

Status: ✓ Complete
Messages: 105 warnings
Part: xc7a100tcsg324-1
Strategy: Vivado Synthesis Defaults
Report Strategy: Vivado Synthesis Default Reports

Status: ✓ Cor
Messages: 6 w
Part: xc7
Strategy: Viva
Report Strategy: Viva
Incremental compile: Nor

DRC Violations

Summary: 2 warnings
Implemented DRC Report

Timing

Worst Negative Slack (WNS):
Total Negative Slack (TNS):
Number of Failing Endpoints
Total Number of Endpoints:
Implemented Timing Report

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs
✓ synth_1	constrs_1	synth_design Complete!								647	1221	0.0
✓ impl_1	constrs_1	write_bitstream Complete!	NA	NA	NA	NA	NA		0	647	1221	0.0