Lab 4: MIPS Datapath for R-type Instructions

****

1 Objective

The objective of this lab is to design a MIPS datapath for R-type instructions (specified in Table 1) as illustrated in Figures 1 to 6.

2 Required Knowledge

- How to write VHDL code using the behavioral and structural models.

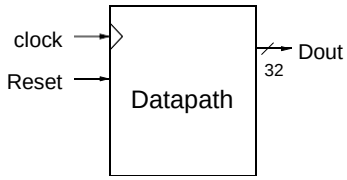

Figure 1: Top-level block diagram of Datapath

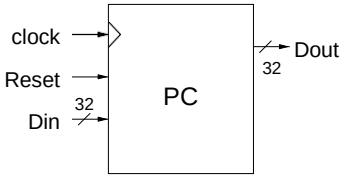

Figure 2: Program counter register (PC)

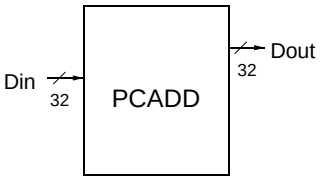

Figure 3: Program counter adder (PCADD)

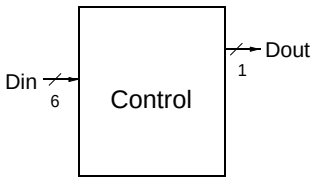

Figure 4: Control

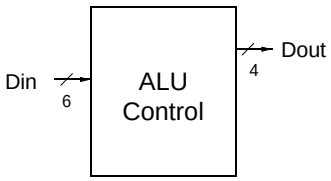

Figure 5: ALU control

- How to use packages in VHDL.
- How to use the Xilinx Vivado Design Suite to write VHDL code, create block designs, add VHDL modules to block designs and create test benches.

3 Design

The objective of this lab is to design the MIPS datapath (and control unit) to implement the R-type instructions listed in Table 1. To accomplish this, the following components must first be designed using behavioral VHDL:

- The Program Counter (PC) (Figure 2)
- The Control Unit (Figure 4)
- The ALU Control Unit (Figure 5)
- The PCADD (Figure 3)

In addition, a working design for the ALU, Register-File and Instruction Memory are provided on the Beachboard. You will have to **modify** the provided ALU to add the functionality for the *slt* instruction. The provided Register-File and Instruction memory units **must be** used as they are pre-loaded with initial values and the test program, respectively.

The datapath should be designed as a block design. Figure 6 illustrates the connections between the components. Table 2 includes all the components that must be designed, as well as the datapath, and the method by which the design is to be implemented.

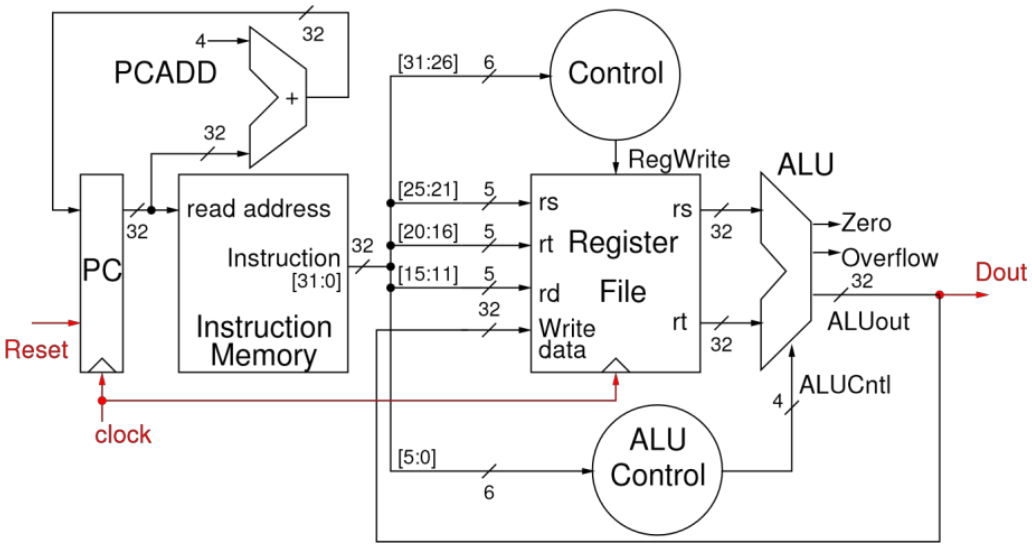

Figure 6: Datapath for R-type instructions

Table 1: R-type instructions to implement

No	Operation	Mnemonic	$Opcode_{hex}$	$Function_{hex}$
1	Add	*add*	00	20
2	Add unsigned	*addu*	00	21
3	Subract	*sub*	00	22
4	Subract unsigned	*subu*	00	23
5	And	*and*	00	24
6	Or	*or*	00	25
7	Xor	*xor*	00	26
8	Nor	*nor*	00	27
9	Set less than	*slt*	00	2A
10	Set less than unsigned	*sltu*	00	2B

**For splitting a bus into smaller width buses, use the Slice IP.*

**Use the specified names for both file and input and output ports in case of datapath.*

4 Testing

The registers in the register file are initialized to values shown in Table 3. The instruction memory contains the test program which is shown in below. Calculate the final values of the registers after the completion of the execution of the test program and complete Table 3.

Table 2: Components to design

No	Component	VHDL model	Design model
1	PC	Behavioral	HDL
2	PCADD	Behavioral	HDL
3	Control Unit	Behavioral	HDL
4	ALU Control Unit	Behavioral	HDL
5	Datapath	-	Block Design

Build a testbench with a clock period of *20ns* and run the simulation for 1 clock cycles with reset='1' and for 10 clock cycles with reset='0'. Compare the final values of the registers from the simulation with the values you calculated in Table 3.

The test program in instruction memory

add $t0, $t1, $t2
addu $t1, $t2, $t3
and $t2, $t3, $t4
nor $t3, $t4, $t5
or $t4, $t5, $t6
xor $t5, $t6, $t7
sub $s0, $s1, $s2
subu $s1, $s2, $s3
slt $s2, $s3, $s4
sltu $s3, $s4, $s5

Table 3. Initial values of registers

No	Register	Calculated		Simulated	
		Initial value$_{hex}$	Final value$_{hex}$	Initial value$_{hex}$	Final value$_{hex}$
0	$zero	00000000		00000000	
1	$at	00000000		00000000	
2	v_0$	00000000		00000000	
3	v_1$	00000000		00000000	

4	$a0	00000000		00000000	
5	$a1	00000000		00000000	
6	$a2	00000000		00000000	
7	$a3	00000000		00000000	
8	$t0	00000009		00000009	
9	$t1	0000000A		0000000A	
10	$t2	0000000B		0000000B	
11	$t3	0000000C		0000000C	
12	$t4	0000000D		0000000D	
13	$t5	0000000E		0000000E	
14	$t6	0000000F		0000000F	
15	$t7	00000010		00000010	
16	$s0	00000011		00000011	
17	$s1	00000012		00000012	
18	$s2	00000013		00000013	
19	$s3	00000014		00000014	
20	$s4	00000015		00000015	
21	$s5	00000016		00000016	
22	$s6	00000017		00000017	
23	$s7	00000018		00000018	
24	$t8	00000019		00000019	
25	$t9	0000001A		0000001A	

26	$k0	00000000		00000000	
27	$k1	00000000		00000000	
28	$gp	00000000		00000000	
29	$sp	00000000		00000000	
30	$fp	00000000		00000000	
31	$ra	00000000		00000000	

5 Lab Deliverables

Submit a ZIPPED file with the following syntax rtypedatapath_LastName(s).zip.

The zipped file should include the following files:

1. Lab 4 Project folder (folder containing all the files of your design including Xilinx generated files).
2. Completed Table 3
3. A comprehensive lab report describing:
   - Your method for designing the datapath components
   - The new HDL codes developed for each datapath component
   - Required modification on the provided ALU to add the functionality for the *slt* instruction
   - The testbench developed to validate your design
4. Snapshot of your datapath design
5. Final simulation waveform

Submit the zipped file through the Beachboard. The submission due date of your work is **August 13 (08/13/2021), 11:59 PM.**