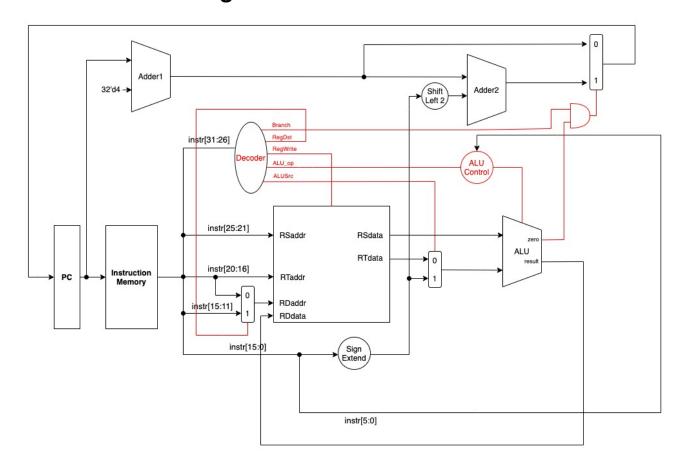
Report

Computer Organization Lab 2

Architecture Diagram



Module Description

· ProgramCounter:

To determine the output of PC by reset while reaching the edges of the clock

• Simple_Single_CPU:

To determine the I/Os of each modules and construct a model

· Adder:

To add up two inputs; here is to generate a new PC

Instr_Memory:

To obtain an instruction from the instruction memory

· Decoder:

To determine the controls by instruction op code

· Reg File:

To deal with the input datum: output the data from the register or write something into RD

· ALU Ctrl:

To determine ALU controls by ALU_ops and functs if the ALU_op represents R-type

• ALU:

To determine the operations under ALU with ALU controls

• MUX_2to1:

To determine the output by certain input controls

• Sign_Extend:

To extend a 16 bit number with the leading digit to a 32 bit number

Shift_Left_Two_32:

To make the number 4 times, to turn the number in terms of instructions into the one in terms of bytes

TestBench:

To test the program with some case datum

Waveform

R-type

• addu

Signals	Waves						
Time	9	6 sec	12 sec	18 sec	24 sec	30 sec 36 sec	42 sec
pc_out_o[31:0] =0000000C	XXXXXXXX	00000000		00000004	00000008	00000000	
RDaddr_i[4:0] =00	xx	00		01	06	00	
RDdata_i[31:0] =0000000A	XXXXXXXX	00000005		00000009	000 00007	0000000A	
RSaddr_i[4:0] =00	xx	02		04	ØA)00	
RSdata_o[31:0] =00000005	XXXXXXXX	00000002		00000004	FFFFFFF	00000005	
RTaddr_i[4:0] =00	xx	03		05	08	00	
RTdata_o[31:0] =00000005	XXXXXXXX	00000003		00000005	00000008	00000005	

• subu

Signals	Waves						
Time	9	6 sec	12 sec	18 sec	24 sec 30	sec 36 sec	42 sec
pc_out_o[31:0] =0000000C	XXXXXXX	00000000		00000004	0000008	999 9999C	
RDaddr_i[4:0] =00	xx	00		01	08	00	
RDdata_i[31:0] =000000000	XXXXXXX	FFFFFFF		000 00004	000 00005	000 00000	
RSaddr_i[4:0] =00	xx	02		09	01	00	
RSdata_o[31:0] =FFFFFFFF	XXXXXXXX	00000002		00000009	000 00004	FFFFFFF	
RTaddr_i[4:0] =00	xx	03		05	00		
RTdata_o[31:0] =FFFFFFFF	XXXXXXXX	00000003		000 00005	FFFFFFF		

and

Signals	Waves							
Time	3	6 sec	12 sec	18 sec	24 sec	30 sec	36 sec	42 sec
pc_out_o[31:0] =0000000C	XXXXXXXX	00000000		00000004	80000008		0000000C	
RDaddr_i[4:0] =00	xx	00		06	01		00	
RDdata_i[31:0] =00000006	XXXXXXXX	00000006		00000002	00000000		00000006	
RSaddr_i[4:0] =00	xx	08		00	08		00	
RSdata_o[31:0] =00000006	XXXXXXXX	FFFFFFE		00000006	80000008		00000006	
RTaddr_i[4:0] =00	xx	07		02	07		00	
RTdata_o[31:0] =00000006	XXXXXXXX	000 00007		00000002	00000007		00000006	

• or

Signals	Waves						
Time)	6 sec	12 sec	18 sec	24 sec	30 sec 36 sec	42 sec
pc_out_o[31:0] =0000000C	XXXXXXX	00000000		00000004	00000008	9999999C	
RDaddr_i[4:0] =00	xx	00		06	01	00	
RDdata_i[31:0] =FFFFFFFF	XXXXXXX	FFFFFFF			A0000000	FFFFFFF	
RSaddr_i[4:0] =00	xx	ØB		00	08	00	
RSdata_o[31:0] =FFFFFFFF	XXXXXXX	FFFFFFFE		FFFFFFF	00000008	FFFFFFF	
RTaddr_i[4:0] =00	xx	07		0Z		.00	
RTdata_o[31:0] =FFFFFFFF	XXXXXXXX	00000007		00000002		FFFFFFF	

slt

Signals	Waves							
Time	9	6 sec	12 sec	18 sec	24 sec	30 sec	36 sec	42 sec
pc_out_o[31:0] =0000000C	xxxxxxx	00000000		00000004	00000008		0000000C	
RDaddr_i[4:0] =00	xx	00		01	02		.00	
RDdata_i[31:0] =00000000	XXXXXXXX	00000001		00000000	00000001		00000000	
RSaddr_i[4:0] =00	XX	02		08	ØA.		00	
RSdata_o[31:0] =00000001	XXXXXXXX	00000002		00000008	FFFFFFF		00000001	
RTaddr_i[4:0] =00	XX	04)06	04		00	
RTdata o[31:0] =00000001	XXXXXXXXX	00000004		00000006	00000004		00000001	

• sra

Signals	Waves								
Time)	6 sec	2 sec	18	sec 24 sec	30	sec 36	sec	42 sec
pc_out_o[31:0] =0000000C	XXXXXXXX	00000000		00000004		80000008	0000	999C	
RDaddr_i[4:0] =00	xx	02		08		06	.00		
RDdata_i[31:0] =00000000	XXXXXXXX	00000000		00000003		FFFFFFF	0000	9999	
RSaddr_i[4:0] =00	xx	99							
RSdata_o[31:0] =000000000	XXXXXXXX	00000000							
RTaddr_i[4:0] =00	xx	03		07		08	99		
RTdata_o[31:0] =000000000	XXXXXXXX	00000003		00000007		FFFFFFE	0000	0000	

srav

Signals	Waves									
Time	9	6 sec	12 sec	18	sec 24	sec	30 sec	36	sec 42 s	ec
pc_out_o[31:0] =0000000C	XXXXXXXX	00000000		00000004			00000008	0000	999C	
RDaddr_i[4:0] =00	xx	02		08			96	00		
RDdata_i[31:0] =000000000	XXXXXXXX	000 00000		00000003			FFFFFFF	0000	9999	
RSaddr_i[4:0] =00	xx	03		01			03	00		
RSdata_o[31:0] =000000000	XXXXXXX	00000003		00000001			000 00003	0000	9999	
RTaddr_i[4:0] =00	xx	03		07			0B	00		
RTdata_o[31:0] =000000000	XXXXXXXX	00000003		00000007			FFFFFFE	0000	9999	

I-type • addi

Signals	Waves						
Time)	6 sec	12 sec	18 sec	24 sec 30 sec	36 sec	42 sec
pc_out_o[31:0] =0000000C	XXXXXXXX	00000000		00000004	000 00008	0000000C	
RDaddr_i[4:0] =00	xx	00		02	06	00	
RDdata_i[31:0] =FFFF2912	XXXXXXXX	FFFF9489		FFFFADAB	FFFFA6BD	FFFF2912	
RSaddr_i[4:0] =00	xx	02		00			
RSdata_o[31:0] =FFFF9489	XXXXXXXX	00000002		FFFF9489			
RTaddr_i[4:0] =00	xx	00		02	06	00	
RTdata_o[31:0] =FFFF9489	XXXXXXXX	00000000	FFFF9489	00000002	000 00006	FFFF9489	

sltiu

Signals	Waves							
Time	3	6 sec	12 sec	18 sec	24 sec	30 sec	36 sec	42 sec
pc_out_o[31:0] =0000000C	XXXXXXXX	00000000		00000004		90000008	0000000C	
RDaddr_i[4:0] =00	xx	00		02		03	00	
RDdata_i[31:0] =000000000	XXXXXXXX	00000000		00000001		00000000		
RSaddr_i[4:0] =00	xx	02		08		0A	00	
RSdata_o[31:0] =000000000	xxxxxxx	00000002		00000008		FFFFFFF	00000000	
RTaddr_i[4:0] =00	xx	00		02		03	00	
RTdata_o[31:0] =000000000	XXXXXXX	00000000		00000002		00000003	00000000	

• beq

Signals	Waves								
Time	3	6 sec	12 sec	18 sec	24 sec	30	sec 36	sec	42 sec
pc_out_o[31:0] =00000010	XXXXXXX	00000000		00000008		0000000C	0000	0010	
RDaddr_i[4:0] =00	xx	00		07		00			
RDdata_i[31:0] =00000000	XXXXXXXX	00000000		00000001		00000000			
RSaddr_i[4:0] =00	xx	00		02		00			
RSdata_o[31:0] =00000000	XXXXXXXX	000 00000		00000002		00000000			
RTaddr_i[4:0] =00	xx	00		07		00			
RTdata_o[31:0] =000000000	XXXXXXXX	000000000		00000007		00000000			

• lui

Signals	Waves						
Time	9	6 sec	12 sec	18 sec	24 sec	30 sec 36 sec	42 sec
pc_out_o[31:0] =0000000C	XXXXXXXX	00000000		00000004	00000008	0000000C	
RDaddr_i[4:0] =00	xx	00		04	08	00	
RDdata_i[31:0] =000000000	XXXXXXX	DEAD0000		BEEF0000	2EDA0000	00000000	
RSaddr_i[4:0] =00	xx	00					
RSdata_o[31:0] =DEAD0000	xxxxxxx	00000000	DEAD0000				
RTaddr_i[4:0] =00	xx	00		04	08	00	
RTdata_o[31:0] =DEAD0000	XXXXXXXX	00000000	DEAD0000	00000004	00000008	DEAD@000	

• ori

Signals	Waves						
Time	9	6 sec	12 sec	18 sec	24 sec	30 sec 36 sec	42 sec
pc_out_o[31:0] =0000000C	xxxxxxx	00000000		000 00004	00000008	9999999C	
RDaddr_i[4:0] =00	xx	00		04	08	00	
RDdata_i[31:0] =0000DEAD	XXXXXXX	0000DEAD		FFFFFFF	00002EDE	00000EAD	
RSaddr_i[4:0] =00	xx	00		ØB	06	00	
RSdata_o[31:0] =0000DEAD	XXXXXXXX	00000000	0000DEAD	FFFFFFE	00000006	00000EAD	
RTaddr_i[4:0] =00	xx	00		04	08	90	
RTdata_o[31:0] =0000DEAD	XXXXXXXX	00000000	0000DEAD	00000004	00000008	00000EAD	

• bne

Signals	Waves							
Time	9	6 sec	12 sec	18 sec	24 sec	30 sec	36 sec	42 sec
pc_out_o[31:0] =00000010	XXXXXXX	000 00000		00000008	00000000		00000010	
RDaddr_i[4:0] =00	xx	00		07	00			
RDdata_i[31:0] =000000000	XXXXXXXX	00000001			00000000			
RSaddr_i[4:0] =00	xx	01		02	00			
RSdata_o[31:0] =000000000	XXXXXXXX	00000001		000 0000 Z	00000000			
RTaddr_i[4:0] =00	xx	80		07	99			
RTdata_o[31:0] =000000000	xxxxxxxx	00000000		000 000007	00000000			

Questions

• What is the difference between "input [15:0] input_0" and "input [0:15] input_0" inside the module?

Way it uses the LSB and the MSB

- What is the meaning of "always" block in Verilog?
 Used to describe events that should happen under certain conditions
- What are the advantages and disadvantages of port connection by order and port connection by name in Verilog?
 - Connection by order
 - Advantages:
 - More simplified
 - · Disadvantages:
 - · Easily cause an error due to unawareness of wrong order
 - · Connection by name
 - Advantages:
 - Reduce the risk of an inadvertent error because a net was connected to the wrong port
 - · Better document the intent of the design
 - · Disadvantages:
 - Too verbose

Contribution

- · ALU module, ALU Ctrl module, Simple Single CPU module, Adder module
- test case generation
- · draw the architecture diagram

However, we debug together, which includes bugs in ALU, ALU_Ctrl.

Discussions

這次的作業雖然繁瑣,sra和一些sign的問題也困擾了我們很久,再來是debug底到快崩潰,但是這次Lab也讓我更徹底了解整個CPU的運作流程,等於是再複習了一次期中考的內容,希望之後的Lab也能繼續像這次一樣令我收穫滿滿。另外隊友很讚,非常棒。