

컴퓨터 구조 (CSED311)

<Lab2 - Single Cycle CPU>

Team ID : 33

Student 1 : 곽민성 (Gwak Minseong, 20230840)

Student 2 : 김재환 (Kim Jaehwan, 20230499)

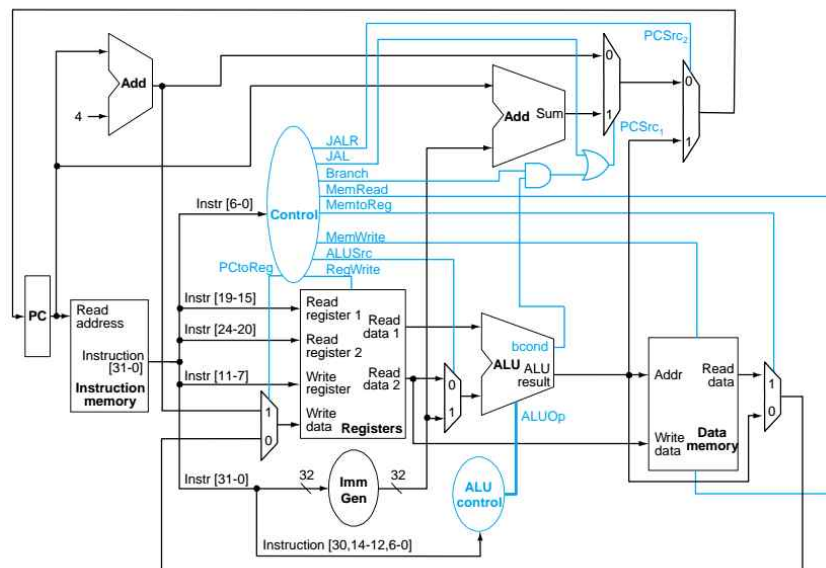
1. Introduction

이번 랩의 목표는 Single Cycle CPU를 Verilog를 통해 구현하는 것이다. 이 랩을 진행하기 위해선, Verilog의 문법과 동작, verilator의 사용법을 이해해야 한다. 우리는 이 랩을 성공적으로 진행하면서, Single Cycle CPU의 세부 구조와 동작을 정확하게 이해할 수 있다.

2. Design

우리는 아래 강의 노트의 CPU 구조를 차용하여 Single Cycle CPU를 구현할 수 있었다.

Datapath with Control



이를 통해 구성한 모듈은 다음과 같다.

모듈 이름	설명	Synchronous / Asynchronous
adder	두 수를 입력으로 받아, 더한 값을 반환한다.	Asynchronous
alu_ctrl_unit	instruction을 입력으로 받아, ALU가 수행해야 하는 동작을 정의하고, 반환한다.	Asynchronous
alu	두 수와 동작을 입력으로 받아, 산술 연산의 결과를 반환한다. Branch Instruction의 경우에, Branch Condition의 결과도 반환한다.	Asynchronous
control_unit	각 Instruction마다 다른 모듈에서 필요한 Control 신호들을 생성한다.	Asynchronous
cpu	다른 모든 모듈을 종합하고 연결해서, 하나의 독립적인 CPU로써 동작하도록 한다.	Synchronous
data_memory	메모리의 주소와 값을 입력으로 받아, 메모리상의 주소의 위치의 값을 반환하거나(mem_read), 그 주소에 들어온 값을 쓴다 (mem_write).	Asynchronous (Read), Synchronous (Write)
imm_gen	Instruction을 읽어서, 그 Instruction에 있는 Immediate value를 파싱해 반환한다.	Asynchronous
instruction_memory	Instruction 주소를 입력으로 받아, Instruction 메모리상의 주소의 위치의 값을 반환한다 (즉, Instruction을 반환한다.)	Asynchronous
mux	두 개의 입력과 1-bit의 sel 신호를 받아, sel 신호가 지시하는 둘 중 하나의 신호를 반환한다.	Asynchronous
pc	다음 클럭의 pc를 입력으로 받아서, clk이 변화함에 맞추어 pc(program counter)의 값을 다음 값으로 업데이트한다.	Synchronous
register_file	2개의 source 레지스터와 destination 레지스터의 번호, 입력값을 받아서 2개의 레지스터의 값을 반환하고, destination 레지스터에 값을 쓴다 (write_enable).	Asynchronous (Read), Synchronous (Write)

3. Implementation

우리는 위 구현된 모듈을 cpu 모듈에서 다음과 같이 wire를 이어서 구현하였다.

IF	Instruction Fetch	<pre>adder pc_adder1(.in0(instr_addr), // input .in1(4), // input .out(next_instr_addr_1) // output); adder pc_adder2(.in0(instr_addr), // input .in1(imm_gen_out), // input .out(next_instr_addr_2) // output); mux pc_mux1(.in0(next_instr_addr_1), // input .in1(next_instr_addr_2), // input .sel(PCSrc1), // input .out(next_instr_addr_3) // output); mux pc_mux_2(.in0(next_instr_addr_3), // input .in1(alu_out), // input .sel(PCSrc2), // input .out(next_instr_addr) // output); pc pc(.reset(reset), // input .clk(clk), // input .next_pc(next_instr_addr), // input .current_pc(instr_addr) // output); instruction_memory imem(.reset(reset), // input .clk(clk), // input .addr(instr_addr), // input .dout(instr) // output);</pre>
----	-------------------	--

ID	Instruction Decode	<pre> mux reg_mux(.in0(data_mux_out), // input .in1(next_instr_addr_1), // input .sel(ctrl_PCtoReg), // input .out(reg_mux_out) // output); register_file reg_file (.reset (reset), // input .clk (clk), // input .rs1 (instr[19:15]), // input .rs2 (instr[24:20]), // input .rd (instr[11:7]), // input .rd_din (reg_mux_out), // input .write_enable (ctrl_RegWrite), // input .is_ecall (ctrl_is_ecall), // input .rs1_dout (rs1_val), // output .rs2_dout (rs2_val), // output .print_reg (print_reg), // output .is_halted (is_halted) // output); // ----- Control Unit ----- control_unit ctrl_unit (.Instr(instr[6:0]), // input .JAL(ctrl_JAL), // output .JALR(ctrl_JALR), // output .Branch(ctrl_Branch), // output .MemRead(ctrl_MemRead), // output .MemtoReg(ctrl_MemtoReg), // output .MemWrite(ctrl_MemWrite), // output .ALUSrc(ctrl_ALUSrc), // output .RegWrite(ctrl_RegWrite), // output .PCtoReg(ctrl_PCtoReg), // output .is_ecall(ctrl_is_ecall) // output); imm_gen imm_gen(.Instr(instr), // input .imm_gen_out(imm_gen_out) // output); </pre>
----	--------------------	---

EX	ALU/Excute	<pre> alu_ctrl_unit alu_ctrl_unit (.Instr30(instr[30]), // input .funct3(instr[14:12]), // input .opcode(instr[6:0]), // input .alu_op(alu_op) // output); mux alu_mux(.in0(rs2_val), // input .in1(imm_gen_out), // input .sel(ctrl_ALUSrc), // input .out(alu_mux_out) // output); alu alu (.alu_op(alu_op), // input .alu_in_1(alu_in_1), // input .alu_in_2(alu_in_2), // input .alu_result(alu_out), // output .alu_bcond(bcond) // output); </pre>
MEM	Data Memory Access	<pre> data_memory dmem(.reset(reset), // input .clk(clk), // input .addr(alu_out), // input .din(rs2_val), // input .mem_read(ctrl_MemRead), // input .mem_write(ctrl_MemWrite), // input .dout(mem_out) // output); mux data_mux(.in0(alu_out), // input .in1(mem_out), // input .sel(ctrl_MemtoReg), // input .out(data_mux_out) // output); </pre>
WB	Write-Back	위 동작에서, memory 모듈로 다시 들어가는 선을 통해 Write-Back 된다.

4. Discussion

1. adder와 alu

instruction address를 계산하기 위해서 덧셈 연산이 필요하다. 이를 동작할 수 있는 ALU 모듈이 있음에도 따로 adder 모듈을 사용하였다. ALU를 사용한 것에 비해 덜 복잡하며, 더 적은 수의 wire를 연결할 수 있다.

2. ecall 처리방법

ecall 명령을 수행할 때 x17 레지스터의 값이 10이면 정지해야하는데, 이를 구현할 때 두가지 방법을 생각하였다.

1. register file 모듈에 is_ecall 신호를 입력으로 넣고, register file 모듈 내에서 is_halted에 값을 할당한다.

2. register file 모듈에서 항상 x17의 값을 출력으로 내보내고, cpu 모듈 내에서 assign문을 통해 할당하기

2번 방법의 경우 레지스터의 값이 RF 모듈 밖으로 계속 출력되고 있다는 점이 불편하여 1번 방법으로 구현하였다.

3. 강의 자료에서는 대부분의 모듈 입출력 및 컨트롤 신호를 설명하고 있지만, alu control unit으로부터 만들어져 alu에 입력으로 들어가는 alu_opcode는 자세하게 설명되어 있지 않았다. 따라서 이번 랩 설계에서는 alu_op.v 파일에 직접 alu opcode를 정의하여 불러와 사용하였다.

4. x0 처리 방법

x0는 항상 값을 0으로 유지해야한다. write_enable (RegWrite) 신호가 있더라도 rd가 0인 경우에는 작성하지 않도록 구현하였다.

5. Conclusion

CPU를 적절히 구현하여 테스트 벤치에 대해서 RISC-V 시뮬레이터인 ripes와 동일한 시뮬레이션 결과를 얻었다. 사이클 수도 ripes와 일치함을 확인할 수 있었다.

	Our cpu	ripes																																																																																																			
basic	### SIMULATING ### TEST END SIM TIME : 58 TOTAL CYCLE : 28 FINAL REGISTER OUTPUT 0 00000000 1 00000000 2 00002ffc 3 00000000 4 00000000 5 00000000 6 00000000 7 00000000 8 00000000 9 00000000 10 00000013 11 00000003 12 ffffffff d7 13 00000037 14 00000013 15 00000026 16 0000001e 17 0000000a 18 00000000 19 00000000 20 00000000 21 00000000 22 00000000 23 00000000 24 00000000 25 00000000 26 00000000 27 00000000 28 00000000 29 00000000 30 00000000 31 00000000	<table><tr><th>Name</th><th>Alias</th><th>Value</th></tr><tr><td>x0</td><td>zero</td><td>0x00000000</td></tr><tr><td>x1</td><td>ra</td><td>0x00000000</td></tr><tr><td>x2</td><td>sp</td><td>0x00002ffc</td></tr><tr><td>x3</td><td>gp</td><td>0x00000000</td></tr><tr><td>x4</td><td>tp</td><td>0x00000000</td></tr><tr><td>x5</td><td>t0</td><td>0x00000000</td></tr><tr><td>x6</td><td>t1</td><td>0x00000000</td></tr><tr><td>x7</td><td>t2</td><td>0x00000000</td></tr><tr><td>x8</td><td>s0</td><td>0x00000000</td></tr><tr><td>x9</td><td>s1</td><td>0x00000000</td></tr><tr><td>x10</td><td>a0</td><td>0x00000013</td></tr><tr><td>x11</td><td>a1</td><td>0x00000003</td></tr><tr><td>x12</td><td>a2</td><td>0xffffffff d7</td></tr><tr><td>x13</td><td>a3</td><td>0x00000037</td></tr><tr><td>x14</td><td>a4</td><td>0x00000013</td></tr><tr><td>x15</td><td>a5</td><td>0x00000026</td></tr><tr><td>x16</td><td>a6</td><td>0x0000001e</td></tr><tr><td>x17</td><td>a7</td><td>0x0000000a</td></tr><tr><td>x18</td><td>s2</td><td>0x00000000</td></tr><tr><td>x19</td><td>s3</td><td>0x00000000</td></tr><tr><td>x20</td><td>s4</td><td>0x00000000</td></tr><tr><td>x21</td><td>s5</td><td>0x00000000</td></tr><tr><td>x22</td><td>s6</td><td>0x00000000</td></tr><tr><td>x23</td><td>s7</td><td>0x00000000</td></tr><tr><td>x24</td><td>s8</td><td>0x00000000</td></tr><tr><td>x25</td><td>s9</td><td>0x00000000</td></tr><tr><td>x26</td><td>s10</td><td>0x00000000</td></tr><tr><td>x27</td><td>s11</td><td>0x00000000</td></tr><tr><td>x28</td><td>t3</td><td>0x00000000</td></tr><tr><td>x29</td><td>t4</td><td>0x00000000</td></tr><tr><td>x30</td><td>t5</td><td>0x00000000</td></tr><tr><td>x31</td><td>t6</td><td>0x00000000</td></tr></table>	Name	Alias	Value	x0	zero	0x00000000	x1	ra	0x00000000	x2	sp	0x00002ffc	x3	gp	0x00000000	x4	tp	0x00000000	x5	t0	0x00000000	x6	t1	0x00000000	x7	t2	0x00000000	x8	s0	0x00000000	x9	s1	0x00000000	x10	a0	0x00000013	x11	a1	0x00000003	x12	a2	0xffffffff d7	x13	a3	0x00000037	x14	a4	0x00000013	x15	a5	0x00000026	x16	a6	0x0000001e	x17	a7	0x0000000a	x18	s2	0x00000000	x19	s3	0x00000000	x20	s4	0x00000000	x21	s5	0x00000000	x22	s6	0x00000000	x23	s7	0x00000000	x24	s8	0x00000000	x25	s9	0x00000000	x26	s10	0x00000000	x27	s11	0x00000000	x28	t3	0x00000000	x29	t4	0x00000000	x30	t5	0x00000000	x31	t6	0x00000000
	Name	Alias	Value																																																																																																		
	x0	zero	0x00000000																																																																																																		
	x1	ra	0x00000000																																																																																																		
	x2	sp	0x00002ffc																																																																																																		
	x3	gp	0x00000000																																																																																																		
	x4	tp	0x00000000																																																																																																		
	x5	t0	0x00000000																																																																																																		
	x6	t1	0x00000000																																																																																																		
	x7	t2	0x00000000																																																																																																		
	x8	s0	0x00000000																																																																																																		
	x9	s1	0x00000000																																																																																																		
	x10	a0	0x00000013																																																																																																		
	x11	a1	0x00000003																																																																																																		
	x12	a2	0xffffffff d7																																																																																																		
	x13	a3	0x00000037																																																																																																		
	x14	a4	0x00000013																																																																																																		
	x15	a5	0x00000026																																																																																																		
	x16	a6	0x0000001e																																																																																																		
	x17	a7	0x0000000a																																																																																																		
	x18	s2	0x00000000																																																																																																		
	x19	s3	0x00000000																																																																																																		
	x20	s4	0x00000000																																																																																																		
	x21	s5	0x00000000																																																																																																		
	x22	s6	0x00000000																																																																																																		
	x23	s7	0x00000000																																																																																																		
	x24	s8	0x00000000																																																																																																		
	x25	s9	0x00000000																																																																																																		
	x26	s10	0x00000000																																																																																																		
	x27	s11	0x00000000																																																																																																		
	x28	t3	0x00000000																																																																																																		
	x29	t4	0x00000000																																																																																																		
	x30	t5	0x00000000																																																																																																		
x31	t6	0x00000000																																																																																																			

loop	### SIMULATING ### TEST END SIM TIME : 446 TOTAL CYCLE : 222 FINAL REGISTER OUTPUT 0 00000000 1 00000000 2 00002ffc 3 00000000 4 00000000 5 00000000 6 00000000 7 00000000 8 00000000 9 00000000 10 00000000 11 00000000 12 00000000 13 00000000 14 0000000a 15 00000009 16 0000005a 17 0000000a 18 00000000 19 00000000 20 00000000 21 00000000 22 00000000 23 00000000 24 00000000 25 00000000 26 00000000 27 00000000 28 00000000 29 00000000 30 00000000 31 00000000	<table><tr><th>Name</th><th>Alias</th><th>Value</th></tr><tr><td>x0</td><td>zero</td><td>0x00000000</td></tr><tr><td>x1</td><td>ra</td><td>0x00000000</td></tr><tr><td>x2</td><td>sp</td><td>0x00002ffc</td></tr><tr><td>x3</td><td>gp</td><td>0x00000000</td></tr><tr><td>x4</td><td>tp</td><td>0x00000000</td></tr><tr><td>x5</td><td>t0</td><td>0x00000000</td></tr><tr><td>x6</td><td>t1</td><td>0x00000000</td></tr><tr><td>x7</td><td>t2</td><td>0x00000000</td></tr><tr><td>x8</td><td>s0</td><td>0x00000000</td></tr><tr><td>x9</td><td>s1</td><td>0x00000000</td></tr><tr><td>x10</td><td>a0</td><td>0x00000000</td></tr><tr><td>x11</td><td>a1</td><td>0x00000000</td></tr><tr><td>x12</td><td>a2</td><td>0x00000000</td></tr><tr><td>x13</td><td>a3</td><td>0x00000000</td></tr><tr><td>x14</td><td>a4</td><td>0x0000000a</td></tr><tr><td>x15</td><td>a5</td><td>0x00000009</td></tr><tr><td>x16</td><td>a6</td><td>0x0000005a</td></tr><tr><td>x17</td><td>a7</td><td>0x0000000a</td></tr><tr><td>x18</td><td>s2</td><td>0x00000000</td></tr><tr><td>x19</td><td>s3</td><td>0x00000000</td></tr><tr><td>x20</td><td>s4</td><td>0x00000000</td></tr><tr><td>x21</td><td>s5</td><td>0x00000000</td></tr><tr><td>x22</td><td>s6</td><td>0x00000000</td></tr><tr><td>x23</td><td>s7</td><td>0x00000000</td></tr><tr><td>x24</td><td>s8</td><td>0x00000000</td></tr><tr><td>x25</td><td>s9</td><td>0x00000000</td></tr><tr><td>x26</td><td>s10</td><td>0x00000000</td></tr><tr><td>x27</td><td>s11</td><td>0x00000000</td></tr><tr><td>x28</td><td>t3</td><td>0x00000000</td></tr><tr><td>x29</td><td>t4</td><td>0x00000000</td></tr><tr><td>x30</td><td>t5</td><td>0x00000000</td></tr><tr><td>x31</td><td>t6</td><td>0x00000000</td></tr></table>	Name	Alias	Value	x0	zero	0x00000000	x1	ra	0x00000000	x2	sp	0x00002ffc	x3	gp	0x00000000	x4	tp	0x00000000	x5	t0	0x00000000	x6	t1	0x00000000	x7	t2	0x00000000	x8	s0	0x00000000	x9	s1	0x00000000	x10	a0	0x00000000	x11	a1	0x00000000	x12	a2	0x00000000	x13	a3	0x00000000	x14	a4	0x0000000a	x15	a5	0x00000009	x16	a6	0x0000005a	x17	a7	0x0000000a	x18	s2	0x00000000	x19	s3	0x00000000	x20	s4	0x00000000	x21	s5	0x00000000	x22	s6	0x00000000	x23	s7	0x00000000	x24	s8	0x00000000	x25	s9	0x00000000	x26	s10	0x00000000	x27	s11	0x00000000	x28	t3	0x00000000	x29	t4	0x00000000	x30	t5	0x00000000	x31	t6	0x00000000
Name	Alias	Value																																																																																																			
x0	zero	0x00000000																																																																																																			
x1	ra	0x00000000																																																																																																			
x2	sp	0x00002ffc																																																																																																			
x3	gp	0x00000000																																																																																																			
x4	tp	0x00000000																																																																																																			
x5	t0	0x00000000																																																																																																			
x6	t1	0x00000000																																																																																																			
x7	t2	0x00000000																																																																																																			
x8	s0	0x00000000																																																																																																			
x9	s1	0x00000000																																																																																																			
x10	a0	0x00000000																																																																																																			
x11	a1	0x00000000																																																																																																			
x12	a2	0x00000000																																																																																																			
x13	a3	0x00000000																																																																																																			
x14	a4	0x0000000a																																																																																																			
x15	a5	0x00000009																																																																																																			
x16	a6	0x0000005a																																																																																																			
x17	a7	0x0000000a																																																																																																			
x18	s2	0x00000000																																																																																																			
x19	s3	0x00000000																																																																																																			
x20	s4	0x00000000																																																																																																			
x21	s5	0x00000000																																																																																																			
x22	s6	0x00000000																																																																																																			
x23	s7	0x00000000																																																																																																			
x24	s8	0x00000000																																																																																																			
x25	s9	0x00000000																																																																																																			
x26	s10	0x00000000																																																																																																			
x27	s11	0x00000000																																																																																																			
x28	t3	0x00000000																																																																																																			
x29	t4	0x00000000																																																																																																			
x30	t5	0x00000000																																																																																																			
x31	t6	0x00000000																																																																																																			

non-controlflow

```
### SIMULATING ###
TEST END
SIM TIME : 446
TOTAL CYCLE : 222
FINAL REGISTER OUTPUT
0 00000000
1 00000000
2 00002ffc
3 00000000
4 00000000
5 00000000
6 00000000
7 00000000
8 00000000
9 00000000
10 00000000
11 00000000
12 00000000
13 00000000
14 0000000a
15 00000009
16 0000005a
17 0000000a
18 00000000
19 00000000
20 00000000
21 00000000
22 00000000
23 00000000
24 00000000
25 00000000
26 00000000
27 00000000
28 00000000
29 00000000
30 00000000
31 00000000
```

Name	Alias	Value
x0	zero	0x00000000
x1	ra	0x00000000
x2	sp	0x00002ffc
x3	gp	0x00000000
x4	tp	0x00000000
x5	t0	0x00000000
x6	t1	0x00000000
x7	t2	0x00000000
x8	s0	0x00000000
x9	s1	0x00000000
x10	a0	0x0000000a
x11	a1	0x0000003f
x12	a2	0xffffffffl
x13	a3	0x0000002f
x14	a4	0x0000000e
x15	a5	0x0000002l
x16	a6	0x0000000a
x17	a7	0x0000000a
x18	s2	0x00000000
x19	s3	0x00000000
x20	s4	0x00000000
x21	s5	0x00000000
x22	s6	0x00000000
x23	s7	0x00000000
x24	s8	0x00000000
x25	s9	0x00000000
x26	s10	0x00000000
x27	s11	0x00000000
x28	t3	0x00000000
x29	t4	0x00000000
x30	t5	0x00000000
x31	t6	0x00000000

위와 같이 정상 동작하는 모습을 확인할 수 있다.

6. References

[1] Lecture notes.