21800612
임건호

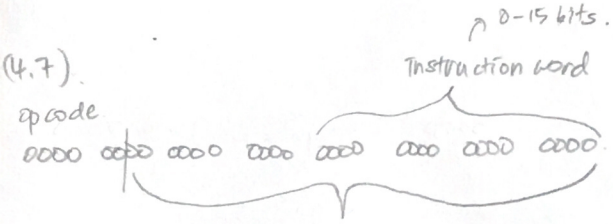
(4.2).

4.2.1 ① Instruction memory.
② Write port in registers.
③ both register read ports.
④ ALU (to add Rd and Rc together)

4.2.2
This one can work using existing blocks. So, no need for new functional blocks.

4.2.3
Do not need any of new signals. It only needs to be change in Control signals.

(4.7).



4.7.1
Sign-extend
↳ 0000 0000 0000 0000 0000 0000 0001 0100.

"Shift left 2"
↳ 0001 1000 1000 0000 0000 0000 0101 0000.

4.7.2
ALU Op : 00
InstructionOp : 010100

4.7.3.
New PC : PC+4
Path : PC → Add (PC+4) → branch Mux → Jump Mux → PC

4.7.4.
Wr Reg Mux : 2 or 0 (RegDst = X)
ALU Mux : 20
Branch Mux : PC+4
Mem / ALU Mux : X
Jump Mux : PC+4

4.7.5.
ALU : -3 and 20
Add (PC+4) : PC and 4
Add (Branch) : PC+4 and 20*4

4.7.6.
↳ Read Register 1, Read Register 2, Write Register, Write data, RegWrite

4.14.1

Pipeline Cycles.

Instructions	1	2	3	4	5	6	7	8	9	10	11	12	13	14
LW R2, 0(R1).	IF	ID.	EX	MEM	WB.									
BEQ R2, R0, Label2 (NT)		IF	ID	***	EX	MEM	WB.							
LW R3, 0(R2).			IF		ID	EX	MEM	WB						
BEQ R3, R0, Label1 (T).					IF	ID	***	EX.	MEM	WB				
BEQ R2, R0, Label2 (T).						IF	***	ID	EX	MEM	WB.			
SW R1, 0(R2)								IF	ID.	EX	MEM	WB		

4.14.2

Pipeline Cycles.

Instructions	1	2	3	4	5	6	7	8	9	10	11	12	13	14
LW R2, 0(R1)	IF	ID.	EX	MEM	WB.									
BEQ R2, R0, Label2 (NT).		IF	ID	***	EX	MEM	WB.							
LW R3, 0(R2)			IF	***	ID	EX	MEM	WB.						
BEQ R3, R0, Label1 (T).					IF	ID	EX	MEM	WB					
ADD R1, R3, R1						IF	ID	EX.	MEM	WB.				
BEQ R2, R0, Label2 (T).							IF	ID	EX	MEM	WB			
LW R3, 0(R2)								IF	ID.	EX	MEM	WB.		
SW R1, 0(R2)									IF	ID	EX	MEM	WB	

4.14.3

```
        LW  R2, 0(R1)
Label1: BEZ R2, Label2   # not taken once, then taken.
        LW  R3, 0(R2)
        BEZ R3, Label1   # taken.
        ADD R1, R3, R1
Label2: SW  R1, 0(R2)
```