WASHINGTON STATE UNIVERSITY

CptS260: Introduction to Computer Architecture

Homework 6 Solution

School of Electrical and Computer Engineering
Spring 2021

Single Cycle Processor

Answer 1:

a. Different control signals for different types of instructions is in Table 1, and Table 2.

Instruction/Signal	RegDst	Jump	Branch	MemRead	MemtoReg	MemWrite
ADD	1	0	0	0	0	0
AND	1	0	0	0	0	0
J	0	1	X	0	0	0
BEQ	X	0	1	0	X	0
LD	0	0	0	1	1	0

Instruction/Signal	ALUOp	ALUSrc	RegWrite	ALUFunct
ADD	10	0	1	0010
AND	10	0	1	0000
J	XX	X	0	XXXX
BEQ	01	0	0	0110
LD	00	1	1	0010

b. The used blocks for each instruction is listed in Table 3.

	Instruction Memory	Register File	Sign-Extend	ALU	Data Memory	PC	ADD1	ADD2	Jump
ADD	✓	✓	×	✓	×	✓	✓	×	×
AND	✓	✓	×	✓	×	✓	✓	×	×
J	✓	×	×	×	×	✓	✓	×	✓
BEQ	✓	✓	✓	✓	×	✓	✓	✓	×
LD	✓	✓	✓	✓	✓	✓	✓	×	×

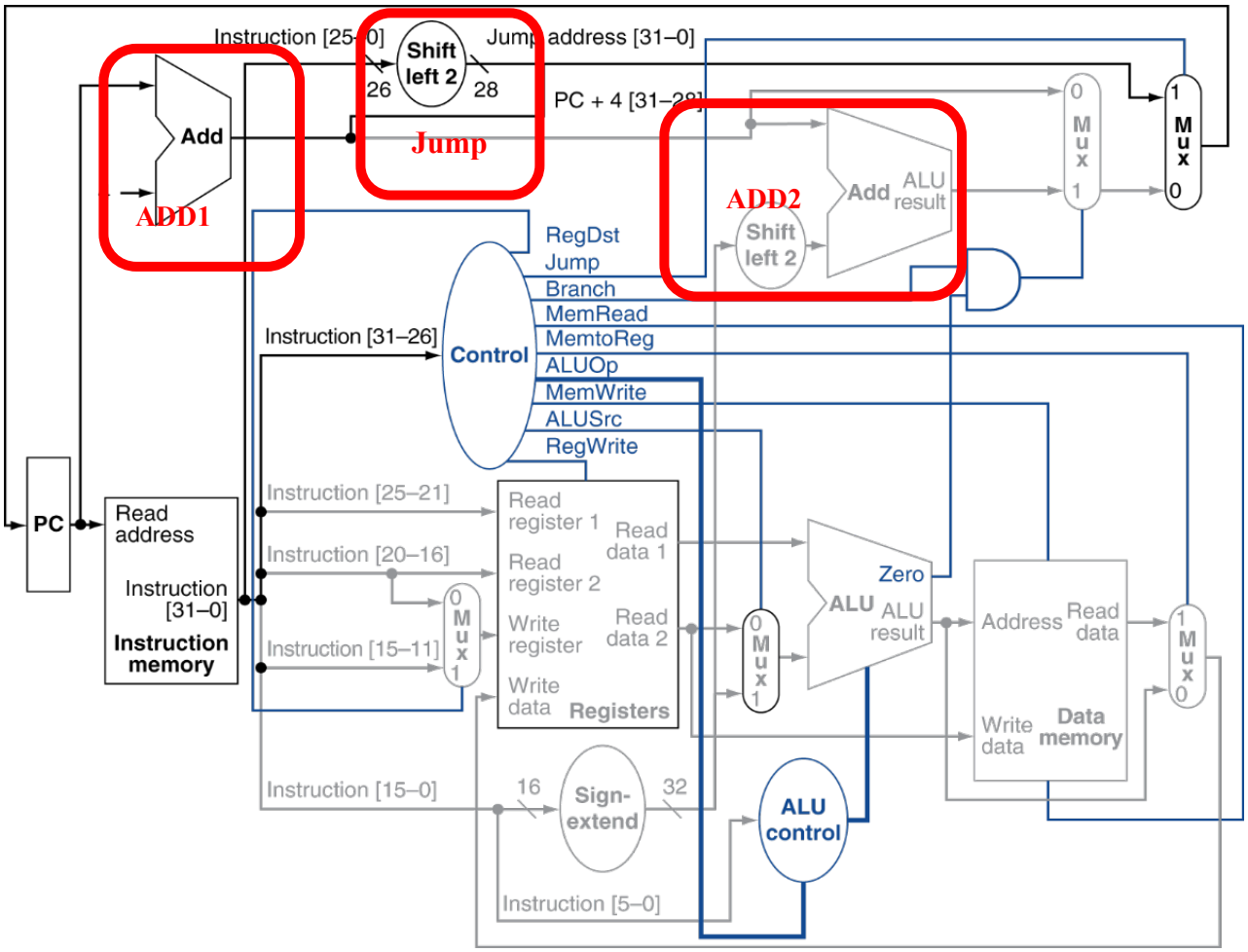

Figure 1. MIPS Single cycle processor

c. The value of PC is 0xF0000000 at first. The value of PC after each instruction is as follows:

1) For ADD, AND, and LD instructions, the next value doesn't depend on the instruction, and always the computer should execute the next instruction in the memory. Therefore: $PC \leftarrow PC + 4$, 0xF0000000 + 4 = 0xF0000004

2) For jump instruction, the next value is PC(31:28) [Instruction(25:0) shift left 2]

J 0x3FFFFFF →

Instruction (25:0) = 0x3FFFFFF = 0b11111111111111111111111111

If we shift left this number by 2 bits, then we get this number:

0b11111111111111111111111100

Finally, we should concatenate the four most significant bit of to this **value:**

PC = 0xF0000000 = 0b1111000000000000000000000000000

Therefore, the new value of PC is:

0b11111111111111111111111111100 = 0xFFFFFFFC

3) For BEQ, we should check the condition, if it is true(as it is here), we should calculate the new value for PC using the relative PC addressing:

$$PC_{new} = Signextend\big(Instruction(15:0)\big) \ll 2 \ + (PC_{old} + 4))$$

$$Instruction(15:0) = \ 0xFFFF \rightarrow sign - extend \ = \ 0xFFFFFFFF \rightarrow$$

$$shift\ left\ 2 \ = \ 0xFFFFFFFC$$

$$PC \ + \ 4 \ = \ 0xF0000004$$

$$PC_{new} = 0xFFFFFFFC + 0xF0000004 = \ 0xF0000000$$

Answer 2:

a. There are two different paths

1) Path to read an instruction from memory (green path)

$$Delay_{green} = Reg(PC) + I - MEM = 90 + 200 = 290ps$$

2) Ptah to update the PC value to PC + 4 (red path)

$$Delay_{red} = Reg(PC) + ADDER + MUX = 90 + 70 + 20 = 180ps$$

These paths are shown in Figure 2. The minimum is the delay of critical path which is 290ps in this case.

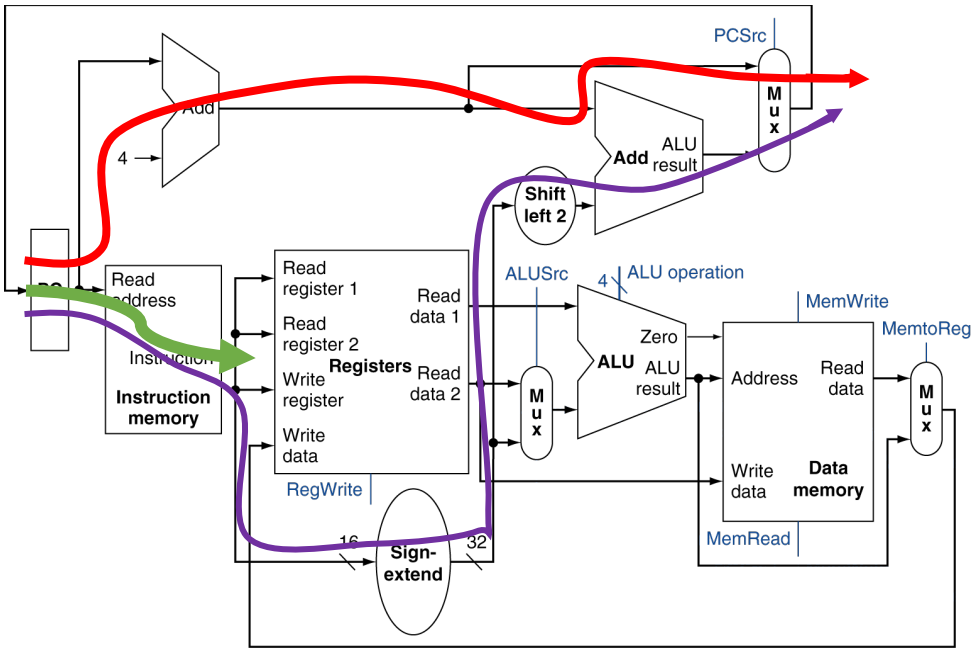

Figure 2. MIPS datapath without control unit.

b. The purple path in Figure 2 shows the path to perform the unconditional branch instruction. The difference with conditional branch is that we don't need to compare two operands.

Therefore there is no need to use ALU, and consequently the path is not through the ALU block.

$$Delay_{purple} = Reg(PC) + I - MEM + Signextend + Shiftleft + ADDER + 1 \times MUX = 90 + 200 + 10 + 15 + 70 + 20 = 405ps$$

If we do the same analysis but using Figure 1 where two MUX are shown due to addition of the control unit and therefore need for separating JUMP from other branches, then the delay will be:

$$Delay = Reg(PC) + I - MEM + Signextend + Shiftleft + ADDER + 2 \times MUX = 90 + 200 + 10 + 15 + 70 + 40 = 425ps$$

c. Jump, Branch

d. ALUSrc signal determines the source of second operand. For I-type instruction (i.e., addi, lw, sw, etc.), ALUSrc should be one as we need the immediate value. On the other hand, ALUSrc should be zero as we need a register as a second operand for R-type instructions (i.e., add, and, beq, etc.)

Answer 3:

Difference between this instruction and regular LW is that the address is based on the data of two registers instead of the constant value and one register.

a. The same set of blocks as ordinary LW instruction is need, except that we don't need the sign-extend block, as we don't need the constant value, and both operands come from register file. The set of blocks for LW are listed in question 1.
b. No new block is needed. We only need to allocate an instruction code for LWI.
c. The control unit should be redesigned to generate correct control signals for the new instruction.