EECS 2021: Computer Organization

Final Review

Final Exam review!

- Chapter 1 – multiple choice, performance calculation
- Chapter 2 – multiple choice, assembly language
- Chapter 3 – multiple choice, floating point calculations, multiplier, divider
- Chapter 4 – multiple choice, processor design, pipelining, hazards

Short answer questions

1. How many instructions could fit into a 256 byte memory unit, given a 32-bit architecture? **(2 marks)**

a) 256 **b)** 64

c) 32 **d)** 8

2. How many address bits are needed to specify each byte in a 512 byte memory unit? **(1 mark)**

a) 512 **b)** 8

c) 32 **d)** 9

3. How many bits do you shift a binary number in order to divide it by 4? **(1 mark)**

4. Why are the addresses of all RISC V instructions divisible by 4? **(1 mark)**

5. Assuming single precision IEEE 754 format, what decimal number is represent by this word:

   1 01111101 00100000000000000000000

6. Using 32-bit IEEE 754 single precision floating point with one(1) sign bit, eight (8) exponent bits and twenty three (23) mantissa bits, show the representation of -11/16 (or -0.6875).

7. In a computer architecture, groups of bits have no intrinsic meanings by themselves. What a bit pattern represents depends entirely on how it is used. The following shows bit patterns expressed in hexademical notation.

   a. 0x0C000000

   b. 0xC4630000

   What decimal number does the bit pattern represent if it is

   (i) a two's complement integer?

   (ii) An unsigned integer?

   (iii) a floating point number? Use the IEEE 754 standard.

Chapter 1 – Performance Calculation

Consider a processor that executes a scientific program. Based on the program profile, 50% of all instructions are floating point multiplication, 20% floating point division, and the remaining 30% are of other instruction types.

(a) Consider the idea to make the program execution 30% faster. This is done by making the divide instructions run 3 times faster or making the multiply run 8 times faster assuming the instructions' CPIs are identical. If you have to choose only speedup one instruction only, can you achieve the speed-up target?

(b) Consider the instruction speed-up in part (a), what is the the speed of the improved machine relative to the original machine?

Chapter 2 – Performance Calculation

Table 1 below shows the clock rate and Cycles Per Instruction (CPI) of four new processors that have the same instructions set.

Processor	Clock Rate	CPI
A	3.0 GHz	2.4
B	700 MHz	2.1
C	4.0 GHz	1.5
D	2.5 GHz	1.0

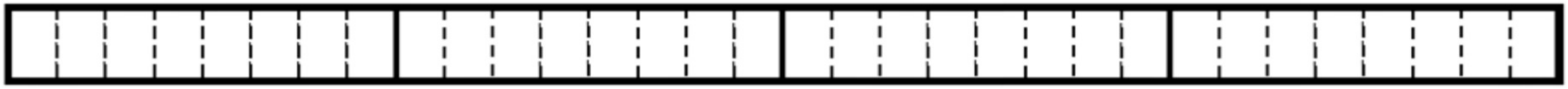
Table 1

(a) Which processor has the best performance based on CPU time? How many times faster is it compared to the other three processors?

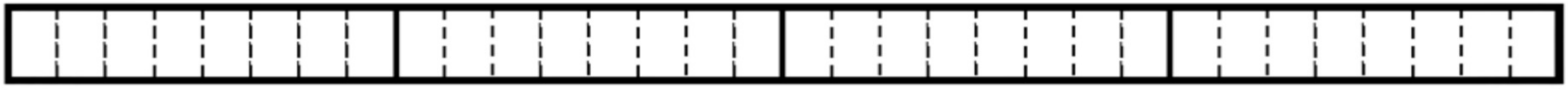
5. For the following assembly language instructions, write the equivalent machine code instruction in the space provided. You might find the reference information in the appendix helpful for this question. **(10 marks)**

a) add x13, x12, x3

b) lw x5, 20 (x3)



c) jal top (where top is at hexadecimal address 0xFF00



6. For the following machine code instructions, provide the equivalent assembly language instruction in the space provided. **(6 marks)**

00000001010000010000000110010011

3. In the space below, write a Verilog module called `counter` that takes in input signals called `clock`, `reset` and `enable`, and has a 4-bit output signal called `value`. **(3 marks)**

- Make the `value` output increment if `enable` is on when the clock goes high **(3 marks)**

1. In the spaces provided below, write the assembly language instruction(s) that corresponds to each of the tasks provided. **(12 marks total)**

a) Perform a right arithmetic shift on the value in `x3`. The number of bits to shift `x3` by is `5`. The result will be stored back into `x3`. **(3 marks)**

5. In the space below, write a short assembly language program that is a translation of the program on the right. You can assume that `i` has been placed on the top of the stack, and should be replaced by the return value before returning to the calling program. Make sure that you comment your code so that we understand what you're doing. **(8 marks)**

```
int make_even (int i) {
   if (i % 2 == 1)
      return i-1;
   else
      return i;
```

(a) The Multiplication Hardware shown in Figure 2 performs the sequential multiplication of the Multiplicand with the Multiplier.

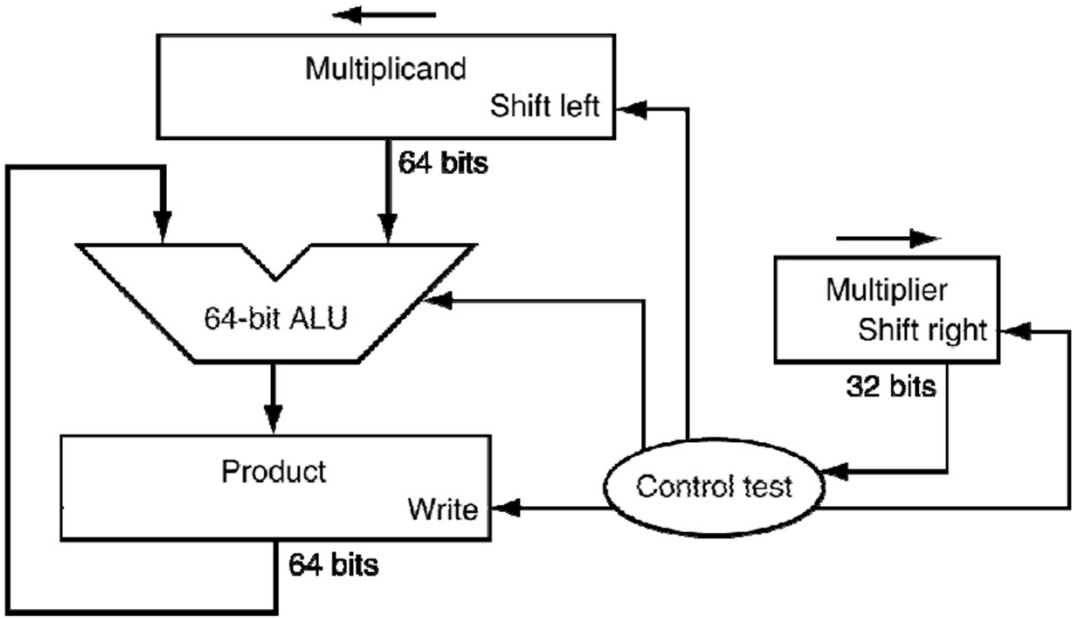


Figure 2: Multiplication Hardware

Describe the sequential multiplication process of 170 x 9 (Binary: $170_d = 1010\ 1010_b$, $9_d = 1001_b$) by filling up Table 3 given below. Write the MIPS instructions performed in each step in the 'Operation' column of the table. The first operation is shown as example.

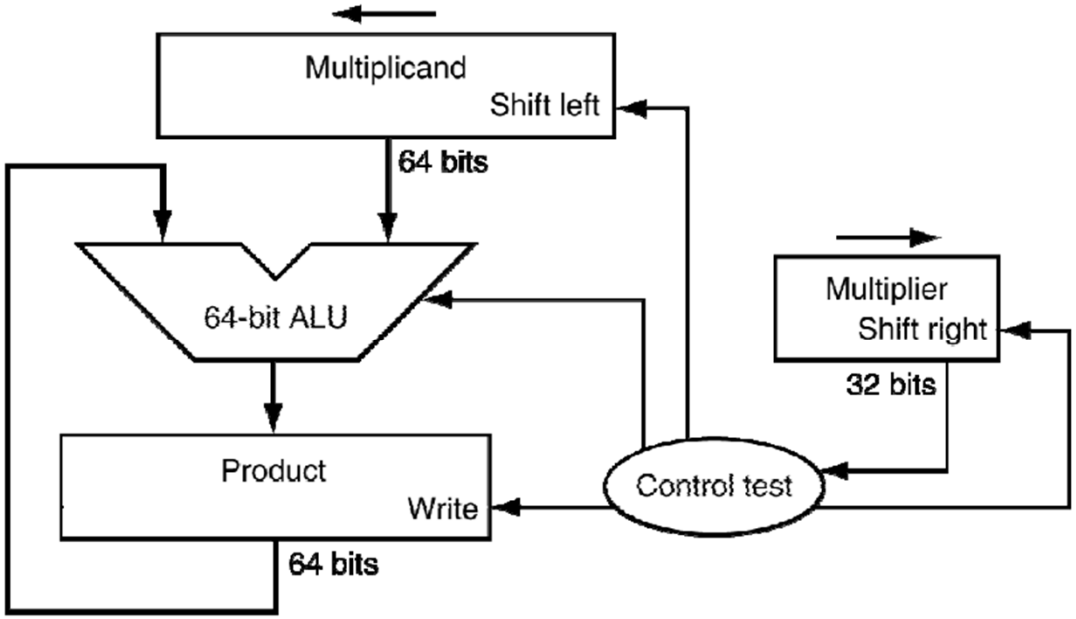
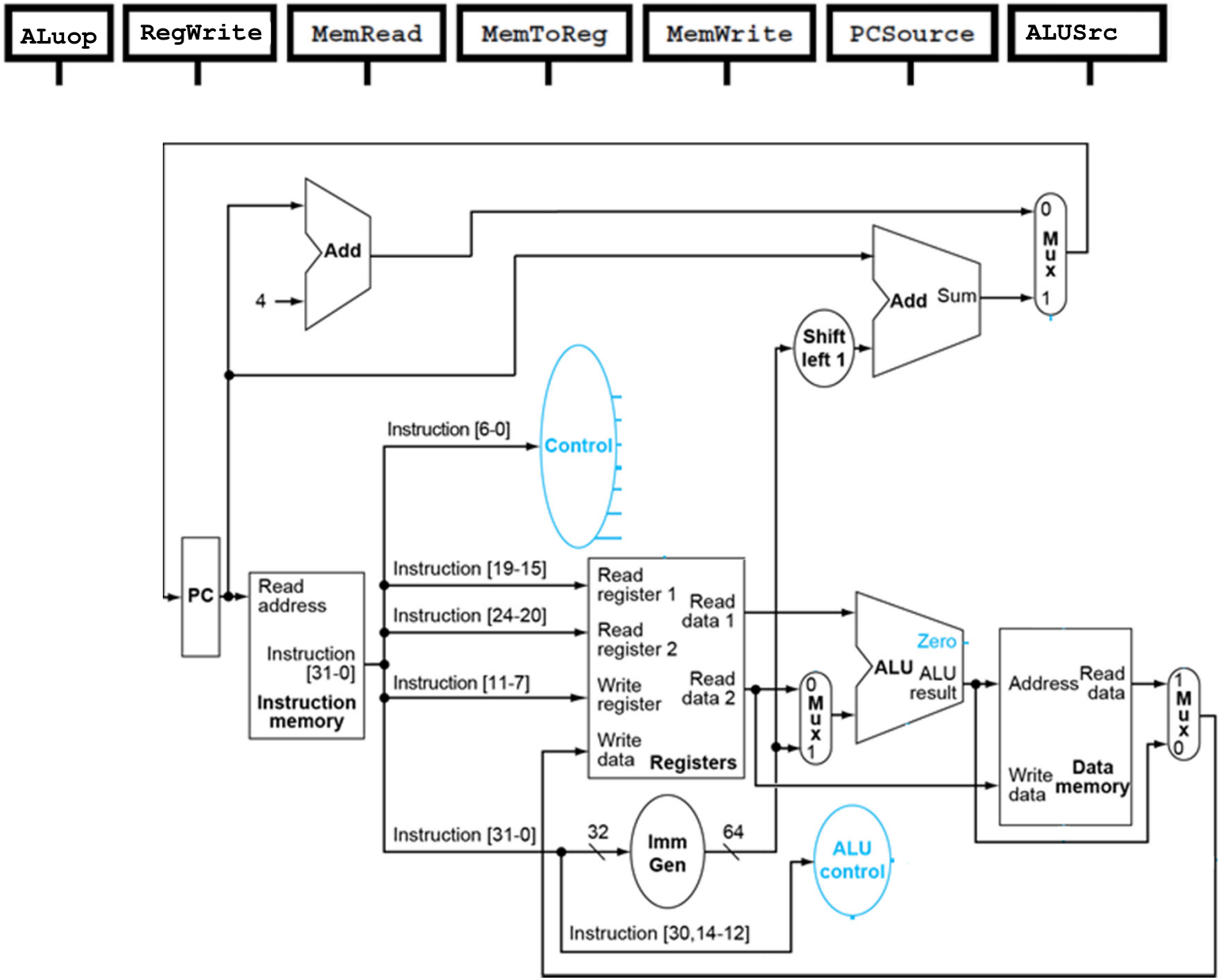


Figure 2: Multiplication Hardware

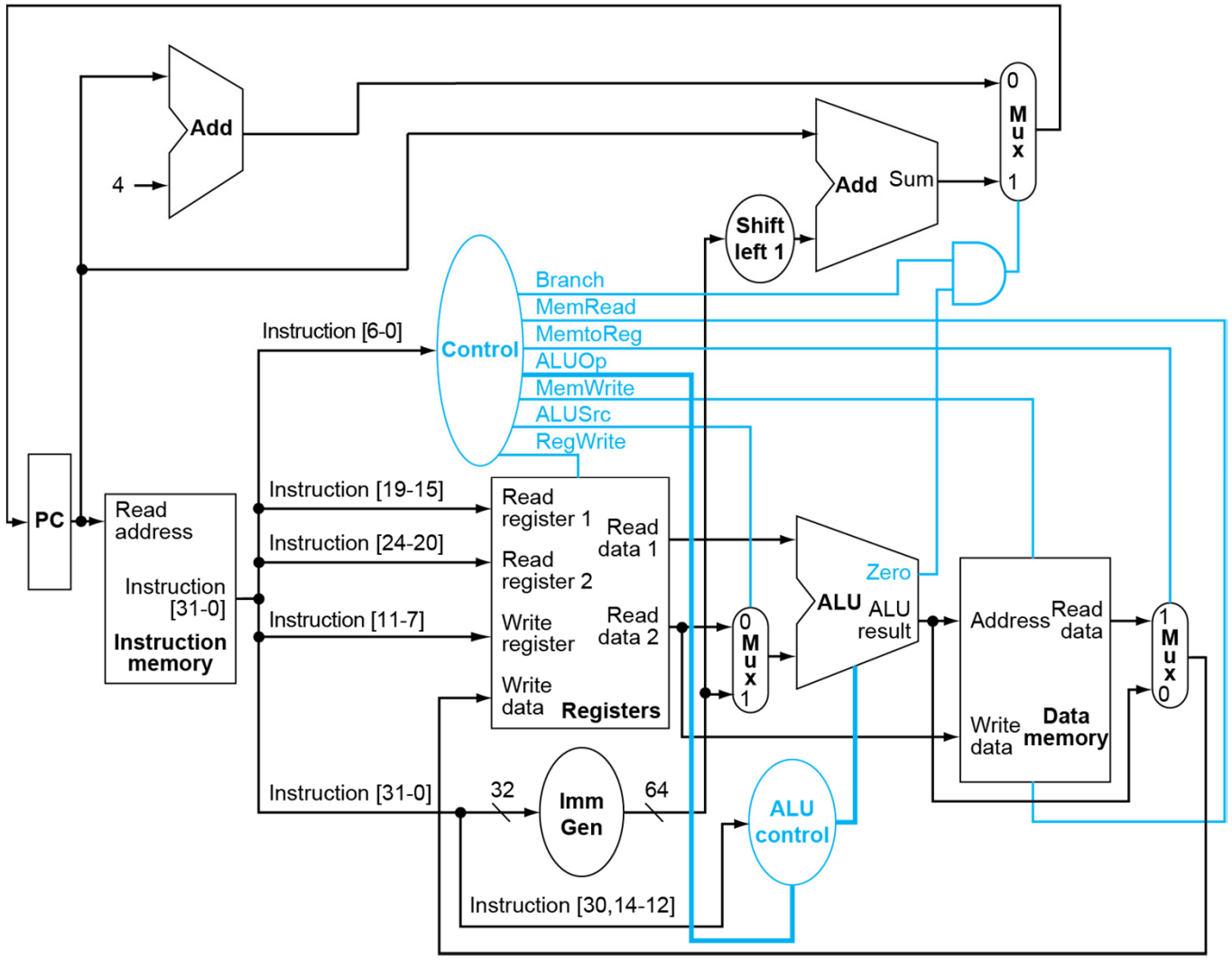
Operation	Multiplicand (M_D)	Multiplier (M_R)	Product (P)
Initial values	1010 1010	1001	0000 0000 0000
P = M_D sll M_D, srl M_R	1 0101 0100	0100	0000 1010 1010
P = P sll M_D, srl M_R	10 1010 1000	0010	0000 1010 1010
P = P sll M_D, srl M_R	101 0101 0000	0001	0000 1010 1010
P = P + M_D sll M_D, srl M_R	1010 1010 0000	0000	1011 1111 1010


ALuop
RegWrite
MemRead
MemToReg
MemWrite
PCSource
ALUSrc
Add
4
Add
Sum
Shift left 1
0
M
u
x
1
Control
Instruction [6-0]
PC
Read address
Instruction [31-0]
Instruction memory
Instruction [19-15]
Instruction [24-20]
Instruction [11-7]
Read register 1
Read register 2
Write register
Write data
Read data 1
Read data 2
Registers
Zero
ALU
ALU result
Address
Read data
Write data
Data memory
Instruction [31-0]
32
Imm Gen
64
ALU control
Instruction [30,14-12]

4. Consider the datapaths below. For each of the following operations, highlight the path that the data needs to take, from start to finish. **(12 marks)**

a) Store the value in `x7` at the current stack pointer location.

b) Increment the program counter by the immediate value.



7. For each of the processor tasks below, indicate what the values of the following control unit signals will be by filling in the boxes next to each signal with the signal values. **(12 marks)**

- If a control signal doesn't affect the operation, fill in its value with an X.
- For ALUOp, if you don't know the values, just write what kind of operation is taking place.

a) Reduce the program counter by the value given in the current instruction's immediate bits.

b) Load the content of base address + offset into register x5.

Branch _____ MemRead _____ MemWrite _____

MemToReg _____ ALUOp _____ ALUSrc _____ RegWrite _____

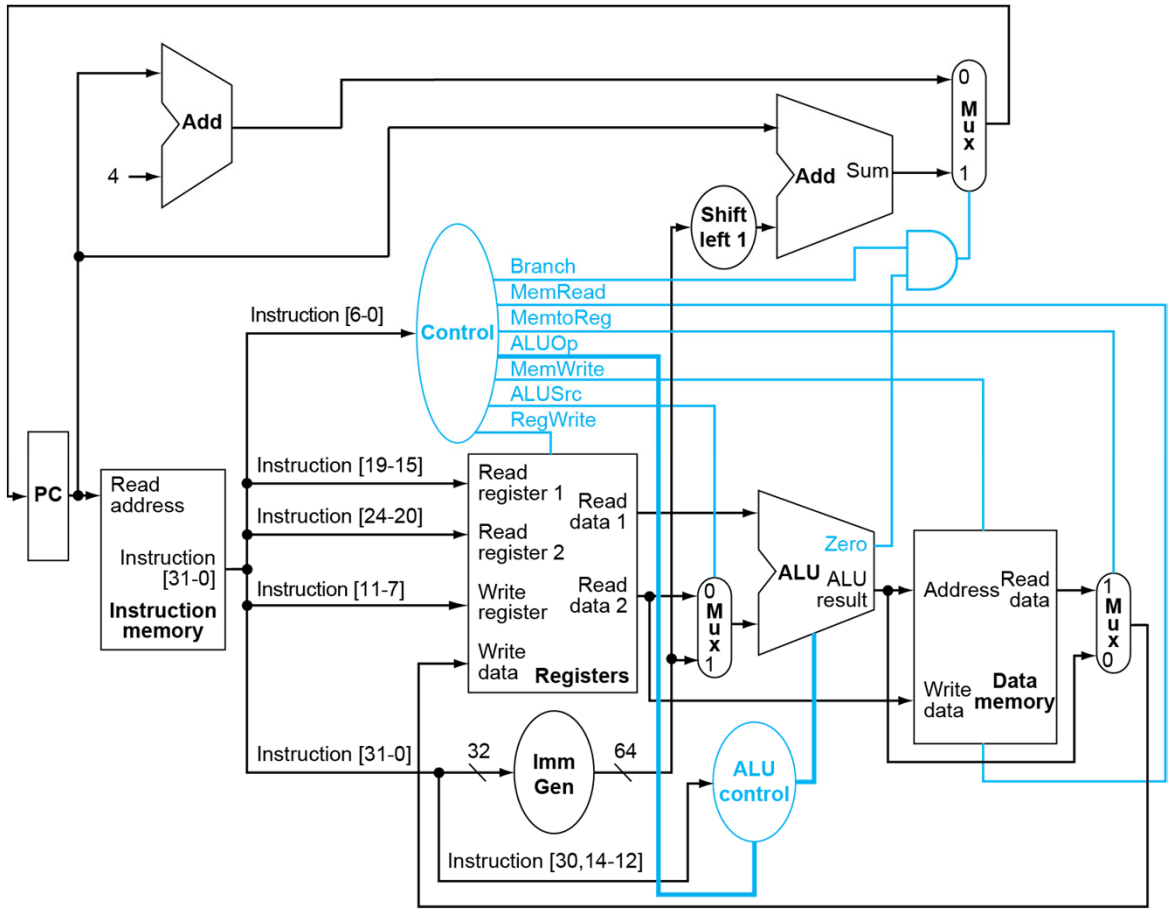
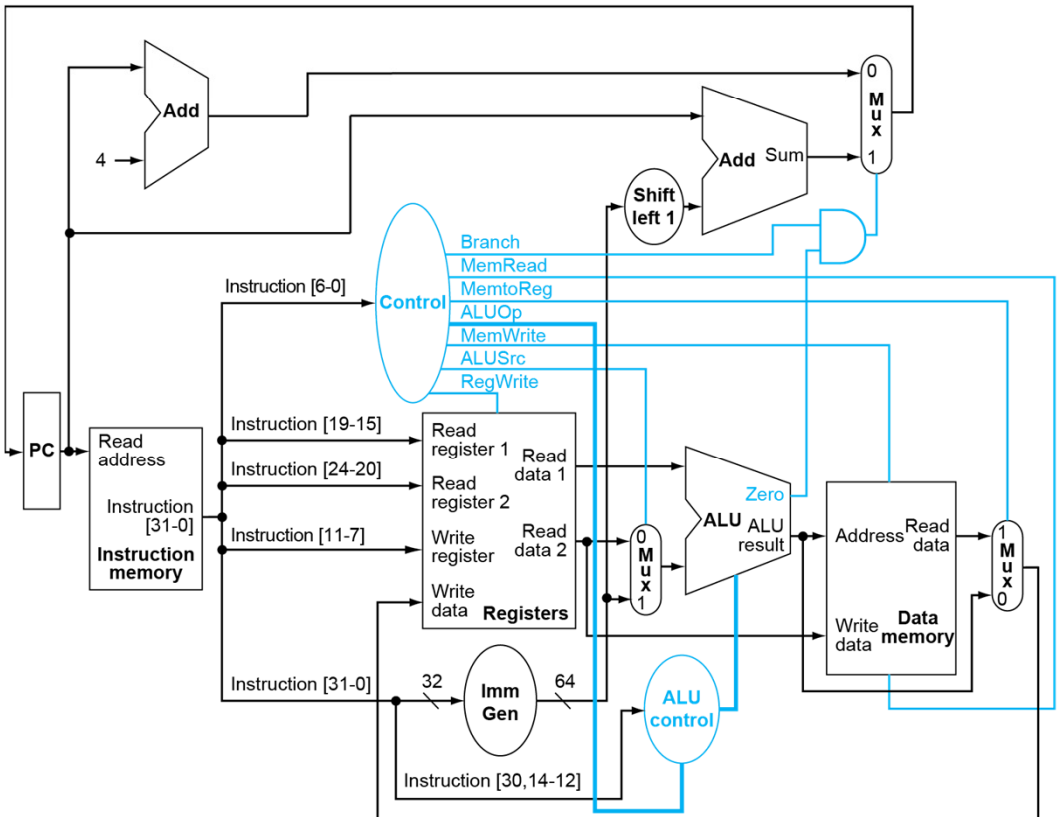


Table 6.1 shows the latencies of individual components of the MIPS datapath. The cummulative latencies affect the clock cycle time of the entire datapath.

Table 6.1: Datapath components' latencies

I-Mem	Add	Mux	ALU	Regs	D-Mem	Sign-Extend	Shift-Left-2
200ps	70ps	20ps	90ps	90ps	250ps	15ps	10ps

(a) What is the clock cycle time if the only types of instructions we need to support are ALU instructions (ADD, AND, etc.)?



6. Consider the following fragment of RISC V code:

```
sw   x16, 12(x6)
lw   x16,  8(x6)
beq  x5,   x4,  loop     ; Assume R5 != R4
add  x5,   x1,    x4
slt  x5,   x15,   x4
```

Assume that all branches are perfectly predicted (such that this eliminates all control hazards) and that no delay slots are used. What is the total execution time of this instruction sequence in the 5-stage pipeline?

Instruction	Pipeline Stage	Cycles

sw x16, 12(x6)
lw x16, 8(x6)
beq x5, x4, loop
add x5, x1, x4
slt x5, x15, x4
	IF ID EX MEM WB
	IF ED EX MEM WB
	IF ID EX MEM WB
	*** *** IF ID EX MEM WB
	IF ID EX MEM WB
		11

Practice #1

- How do you write an assembly language program that performs the following task:

```
int series() {
    int n = 0;
    for (int j=0; j<100; j++)
       n += j;
    return n;
}
```

Practice #2

- How do you write an assembly language program that performs the following task:

```
int fact(int i) {
   if (i <= 0)  return 1;
   else return i * fact(i-1);
}
```

Some Final Thoughts

The End