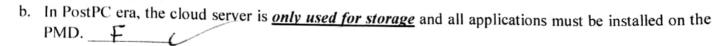


a. Super computers rely on parallelism to achieve very high performance for high end applications.



c. A CPU is designed with two adders, where only one adder is used for execution and the other one is used a spare in case the first adder becomes faulty. This is an example of improving performance via prediction.

d. Flash drive is faster and more expensive than the traditional magnetic disk drive.

(Ic)aug & (Ic.) same Q2. CPU-1 and CPU-2 have the same ISA and same compiler. The ISA has three instruction types: A, B, and C. The CPI for each instruction type in CPU-1 and CPU-2 are given in the table below. ProgramX is executed on both CPUs. The relative frequency of each instruction type in ProgramX is also given in the table. Answer the following questions: 1.5 < 3 points >

a. Compute  $\underline{CPI}_{avg}$  for ProgramX when executed on CPU-1 and CPU-2.

CP[aug CPUZ = 4x0.2 +1x0.4 +1x0.4

•	A	В	C
CPU-1 CPI	2	3	2
CPU-2 CPI	4	1	1
ProgramX Instruction%	20%	40%	40%

2.4 CPU-1 CPI<sub>avg</sub> =

CPU-2 CPI<sub>avg</sub>= 1.6

Given that the Clock Rate of CPU-1 is 3 times the Clock Rate of CPU-2, which CPU has bette performance when executing ProgramX? and by how much?

performance = 1



Q3. <u>Two compiled versions</u> (Version-1 and Version-2) of the same program are executed on the <u>same CPU</u>. The IC for each instruction type in Version-1 is given in the table. The CPI for each instruction type is given as well. Answer the following questions: 1.5 < 3 points>

How many <u>clock cycles</u> are needed for Version-1?

4.4				
CIK cycles -	TCVCOT -	2 v4	- 6 - 1	+2x5+2x2
0-,	TO YOUR	24	1 2 0 7	+ 425

Instruction Type	A	В	C	D
Version-1 IC	3	6	2	2
CPI	4	1	5	2

 $\Box$ 

1.5

Version-1 Clock Cycles = 321

Given that the performance of Version-2 is <u>4 times</u> the performance of Version-1, how many <u>clock</u> cycles are needed for Version-22

CPIA = 4x3 +1x6+5x2+2x2

Q4. Answer the following short questions:

a. Convert the following C-statement into RISC-V assembly language. Assume that "B" is an array of long long integers and its base address is mapped to register x8: <1 point>

$$B[5] = 0;$$

b. Convert the following C-statement into RISC-V assembly language. Assume that variable W is mapped to register x20 and variable Y is mapped to register x10: <1 point>

$$\overset{\chi_{\tau \omega}}{W} = \overset{\chi_{1o}}{Y};$$

$$W = Y;$$
RISC-V Assembly: Xori X20, Xw, Q

c. Convert the following RISC-V machine code instruction into assembly language. A list of opcodes and function fields is given on the right. <2 points>

Machine code instruction: 0000000011110011110100101001011

			/
Instruction	Opcode	Funct3	Funct6 or Funct7
add	0110011	000	0000000
lh	0000011	001	n.a.
bge	1100111	101	n.a.
- addi	0010011	000	n.a.

slli 0010011 001 000000 srli 0010011 101 000000

RISC-V Assembly: ----

2

d. Specify the contents of register x6 and memory location at address 2 (i.e. Memory[2]) in hexadecimal format after executing the following RISC-V code. Assume the initial values are as follows:  $x20 = 0x \ C7B2345D8A09FEB1, Memory[2] = 0xAD, Memory[3] = 0x74$ : <2 points> مذ سيده معد مطرا المعالم 2+0 1 sh x20, 2 (x0) 240 1 lb x6, 3 (x0) Memory[2] = 0 (c7.B2345D8  $(x6) = 0 \times 74$ e. Specify the contents of registers x17 and x18 in hexadecimal format after executing the following RISC-<2 points> V code: 0 x 7A/2D3 0 x 7A lui x17, 0x7A2D3 addi x18, x17, 0x€59 1000 1001 (x17) = -0.0x 7AOX7B 111 1011 000 11-11/11+ <2 points> f. Given the following RISC-V code: Instruction PC in decimal Loop: beq x6, x0,820 add x5, x5, x6 24 addi x6, x6, -1 28. jal x0, Loop 32 add x7, x5, x0 36 instruction will be executed next? Compute the value of the "Loop" immediate in the "jal" instruction at  $\underline{PC} = 32$ . Target addles = pctinnix2 -> 20 = 32 +imx2 Loop (in <u>decimal</u> format) = -66e. Given that the code on the right uses lock/unlock synchronization and two processes (P1 and P2) are executing the code in parallel, answer the following questions: <2 points> addi x5,x0,1 If <u>P1</u> is currently executing the addi x21, x21, 1 instruction as indicated try: by the black arrow, determine the value of the Lock variable in the lr.d x6, (x30) memory and determine which instruction P2 will probably execute next? bne x6,x0,try

Lock variable = -----

P2 will probably execute next (list all possible instructions): ----

Which instruction represents the 1<sup>st</sup> instruction in the ME region?

1st instruction in ME region is: --- bne x4 , xa, +(4)

sc.d x8,(x30),x5

bne x8,x0,try

ld x21, 40(x0)

■ addi x21, x21, 1

sd x21, 40(x0)

sd x0, 0(x30)

Q5. Given the C-language procedure on the right, answer the two questions below. Assume that the base address of "array A" is mapped to x10, variable "n" is mapped to x11/2 variable "key" is mapped to £122 variable "result" is mapped to x13, and variable ("i") is mapped to x27) Notice that x27 is a saved register.

a. Translate the procedure into RISC-V assembly language.

```
Sol X27, 0(3P)

Sol X13, 8(SP)

Sol X1, 16(SP)

Sol X1, 16(SP)

Sol X1, 24(SP)

addi X13, X0,0

addi X27, X0,0

Coldi X27, X0
                                    before Xzz, Xu, Evil 0.5
                                     5111 XMO X27,3 0.25
                            bed XIO, XIZILI X
Li. Addi X13, X13, 10.25
```

```
long long int REC (long long int A [ ].
long long int n, long long int key)
        int i, result;
        result = 0;
for (i = 0; i < n; i++) 1 \neq 1
                 if (A[i] == key)
                         result ++:
        return result;
```

< 6.5 points -

1d x,, 24 (SP) 1d x12 , 32 (SP) add: 50,50 9+40 Jal ( X 6, 0 (x)) Exit: - 1 - 1d X27, 0(SP)

addispispituo ject Xo, o (xi)

b. Given that register x11 is initialized to 100 and register x12 is initialized to 17, translate the following C-statement associated with the procedure above into RISC-V assembly language. Assume variable Number is mapped to register x9 and the starting address of "array A" in memory is 16.)

Number = REC (address of array A, 100, 17);

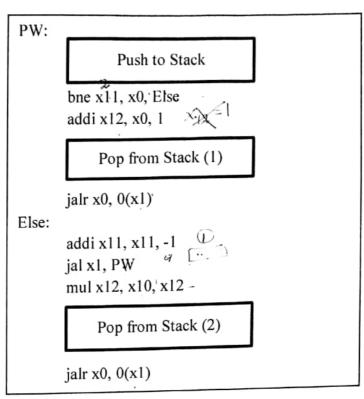
RISC-V Assembly:

1d X27,0CSP

1d X13,813P ld x, , 16 (5P)

(addi Xio IXa OX

Q6. Given the following <u>Non-leaf</u> procedure written in RISC-V assembly language, answer the questions below accordingly. The procedure has two arguments mapped to registers x10 and x11 and the return value is mapped to register x12.



- a. Determine whether each of the following statements is True or False?
  - Register x1 <u>must</u> be saved in the stack
  - Register x10 <u>must</u> be saved in the stack
  - Register x11 <u>must</u> be saved in the stack

F	X
T	X

b. Given the following procedure call, what is the <u>final value</u> in register x1 and x12 in <u>decimal</u> format? <u>Procedure call:</u>

PC in decimal	Instruction
40	addi x10, x0, 4
44	addi x11, x0, 2
48	jal x1, PW

$$(x1) = 48 + 4 = 52$$