

CENG 3420 Final (2021 Spring)

Name: Ng Chi Hon
ID: 1158116317

Q0 (0 marks)

- What is your last digit of your SID (0 is regarded as 10)? This value is defined as **NUM_1** in the whole question paper.
- What is your last two digits of your SID (00 is regarded as 100)? This value is defined as **NUM_2** in the whole question paper.
- What is your last three digits of your SID? This value is defined as **NUM_3** in the whole question paper.

Example: if your SID is 12345678, then $\text{NUM_1} = 8$, $\text{NUM_2} = 78$, $\text{NUM_3} = 678$.

Q1 (10 marks)

Select and fill the correct answer.

- RISC-V is a ____ and ____ ISA.

- A. little-endian RISC
- B. little-endian CISC
- C. big-endian RISC
- D. big-endian CISC

- Virtual memory systems use ____ mechanism.

- A. write-back
- B. write-through
- C. write-allocate

- Consider a virtual memory system including only one-level page table. Suppose that it takes 100 ns to access the memory, 20 ns to access the TLB (i.e., Translation-lookaside buffer) and the hit rate of TLB is approximately 80 %. So the average time to access the memory is about ____ ns. If the page fault rate is approximately 1 % and the page fault interruption needs 20 ms to handle this case, so the average time to access the memory is about ____ ns.

- D. 110 20134
- A. 120 20136
- B. 130 20138

100 ns

20

80%

170

4. A computer system adopts 32-bit single-word instructions and the address code is 12 bits. If 250 two-address instructions are defined, then the number of single-address instructions is _____

- A. 4 K
- B. 8 K
- C. 16 K
- D. 24 K**

5. The main frequency of a computer is 1.2GHz, and its instructions are divided into 4 categories. Their proportion in the benchmark are as shown below, So the MIPS of this

Type	proportion	CPI
A	50 %	2
B	20 %	3
C	10 %	4
D	20 %	5

computer system is _____

- A. 100
- B. 200
- C. 400**
- D. 600

1.2 · 10⁹ / (3)

6. The largest positive integer in IEEE754 single-precision floating-point format is _____

- A. $2^{126} - 2^{103}$
- B. $2^{127} - 2^{104}$
- C. $2^{127} - 2^{103}$
- D. $2^{128} - 2^{104}$**

7. _____ is a type of single address space multiprocessor in which some memory accesses are much faster than others depending on which processor asks for which word.

A. UMA

B. NUMA

8. Dividing 10101_2 by 11_2 , the quotient is and the remainder is .

A. 100_2 01_2

B. 111_2 01_2

C. 111_2 0_2

D. 100_2 0_2

9. ____ is edge triggered, thus it can be inserted between pipeline stages.

A. Latch

B. Flip-flop

10. ____ stores data as electric charge on a capacitor.

A. SRAM

B. DRAM

Q2 (10 marks)

In the post-PC era the really valuable resource is energy. Taking power dissipation into consideration in manufacturing a CPU becomes more and more important nowadays.

The Pentium 4 Prescott processor, released in 2004, had a clock rate of 3.6 GHz and voltage of 1.25 V. Assume that, on average, it consumed 10 W of static power and 90 W of dynamic power.

The Core i5 Ivy Bridge, released in 2012, had a clock rate of 3.4 GHz and voltage of 0.9 V. Assume that, on average, it consumed 30 W of static power and 40 W of dynamic power.

1. For each processor, find the average capacitive loads. (Hint: recall that how we approximate dynamic power)
2. Find the percentage of the total dissipated power comprised by static power and the ratio of static power to dynamic power for each CPU.
3. If the total dissipated power is to be reduced by 10 %, how much should the voltage be reduced to maintain the same leakage current? (Hint: power is defined as the product of voltage and current).

Q3 (5 marks)

Assume it takes one clock to send an address to DRAM memory and one clock to send the corresponding data back. DRAM has 8-cycle latency for first byte, and 4-cycle for each of subsequent bytes in the block. To transfer a 8-byte block, calculate the cycle number if we need:

1. Non-interleaving
2. 2-module interleaving

Q4 (15 marks)

Figure 1 is the datapath of a RISC-V core architecture.

Consider logic blocks used to implement the datapath as Figure 2 shown, and they have corresponding latency.

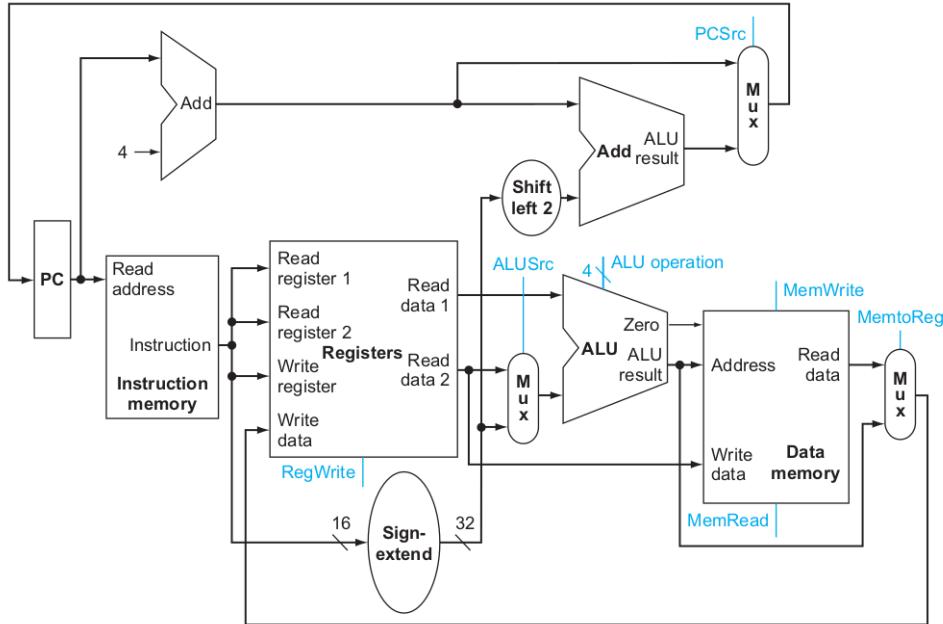


Figure 1: The datapath for the core RISC-V architecture combines the elements required by different instruction classes.

I-Mem/ D-Mem	Register File	Mux	ALU	Adder	Single Gate	Register Read	Register Setup	Sign Extension	Control
NUM_3 ps	200 ps	25 ps	150 ps	100 ps	NUM_1 ps	NUM_2 ps	50 ps	45 ps	40 ps

Figure 2: The latency of each logic block.

In Figure 2, **Register read** is the time needed after the rising clock edge for the new register value to appear on the output. This value applies to the PC only. **Register setup** is the amount of time a register's data input must be stable before the rising edge of the clock. This value is applied to both the PC and Register File.

1. What is the latency of an R-type instruction?
2. What is the latency of ld?
3. What is the latency of sd?
4. What is the latency of beq?
5. What is the latency of an I-type instruction ?
6. What is the minimum clock period for this CPU?

Q5 (15 marks)

Consider a RISC-V code snippet.

```

    li t0, 400
loop1: lw a0, 0(t0)
        lw a1, 400(t0)
        addi a2, a0, a1
        sw a2, 0(t0)
        addi t0, t0, -4
        bnez t0, loop1

```

Assume that the RISC-V design has following features,

- classical 5-stage pipeline: IF, ID, EX, MEM and WB
 - Each stage requires only one cycle
 - All memory references are hit in cache
1. Calculate how many clock cycles the machine will take execution of the code (Assume no bubble in the pipeline).
 2. Calculate how many clock cycles will take execution of the code without forwarding or bypassing when result of the branch instruction (new PC content) is available after the WB stage.
 3. Calculate how many clock cycles will take execution of the code with normal forwarding and bypassing when result of branch instruction (new PC content) is available after completion of the ID stage. Meanwhile, calculate the speedup based on the result of the answer of the first question.

Q6 (10 marks)

We have designed a 64-bit address direct-mapped cache, and the bits of address used to access the cache are as shown in Table 1

Table 1: Bits of the address to use in accessing the cache

Tag	Index	Offset
63-10	9-5	4-0

1. What is the block size of the cache in words?
2. Find the ratio between total bits required for such a cache design implementation over the data storage bits.
3. Beginning from power on, the following byte-addressed cache references are recorded as shown in . Find the hit ratio.

Table 2: Recorded byte-addressed cache references

Hex	00	04	10	84	E8	A0	400	1E	8C	C1C	B4	884
Dec	0	4	16	132	232	160	1024	30	140	3100	180	2180

Table 3: Virtual Memory System

Virtual Address (bits)	Physical DRAM Installed	Page Size	PTE Size (byte)
43	16 GB	4 KB	4

Q7 (15 marks)

Consider that we have a virtual memory system, and key parameters are listed as shown in table [3]

1. For a single-level page table, find the number of needed page table entries (PTEs) and the physical memory needed to store the page table.
2. Multi-level page table mechanism can reduce the physical memory consumption of page tables by only keeping active PTEs in physical memory. How many levels of page tables will be needed if the segment tables (the upper-level page tables) are allowed to be of unlimited size?
3. If the segments are limited to the 4KB page size, how many levels of page tables are needed?

Q8 (10 marks))

Consider the following portions of two different programs running at the same time on four processors in a share memory multiprocessor (SMP). Assume that before this code is running, both x and y are 0.

```

Core1: x = NUM_2;
Core2: y = NUM_3;
Core3: w = x + y + 1;
Core4: z = x + y;
```

1. What are all the possible resulting values of w, x, y, z ? For each possible outcome, explain how we might arrive at those values.
2. How could you make the execution more deterministic so that only one set of values is possible?

Q9 (10 marks)

We have designed a machine that can execute different instructions including ADD, ADDI, AND, BEQ, BNEQ, JUMP, LW, OR, SUB and SW. Figure [3] shows the finite state machine (FSM) of this multi-cycle design and every state requires one cycle to finish.

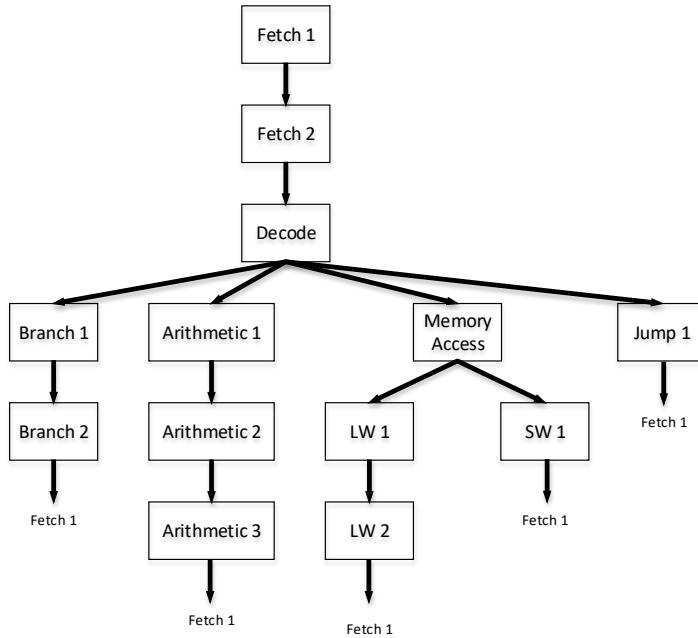


Figure 3: FSM 1

Instruction category	Instructions	Frequency
Arithmetic	ADD, ADDI, AND, OR, SUB	80% - $\frac{\text{NUM_2}}{2}\%$ - $\frac{\text{NUM_1}}{2}\%$
Branch	BEQ, BNEQ	$\frac{\text{NUM_2}}{2}\%$
Jump	JUMP	$\text{NUM_1}\%$
Memory Access	LW, SW	20% (i.e., 10% loads, 10% stores)

Table 4: Instructions categories & frequency

The instructions are broken down into categories as the Table 4 shows.

1. Calculate the average CPI.
2. Suppose we make a change to the design while we keep the same compiler and the same frequency of instructions. The new FSM is as shown in the Figure 4. Which design is better?

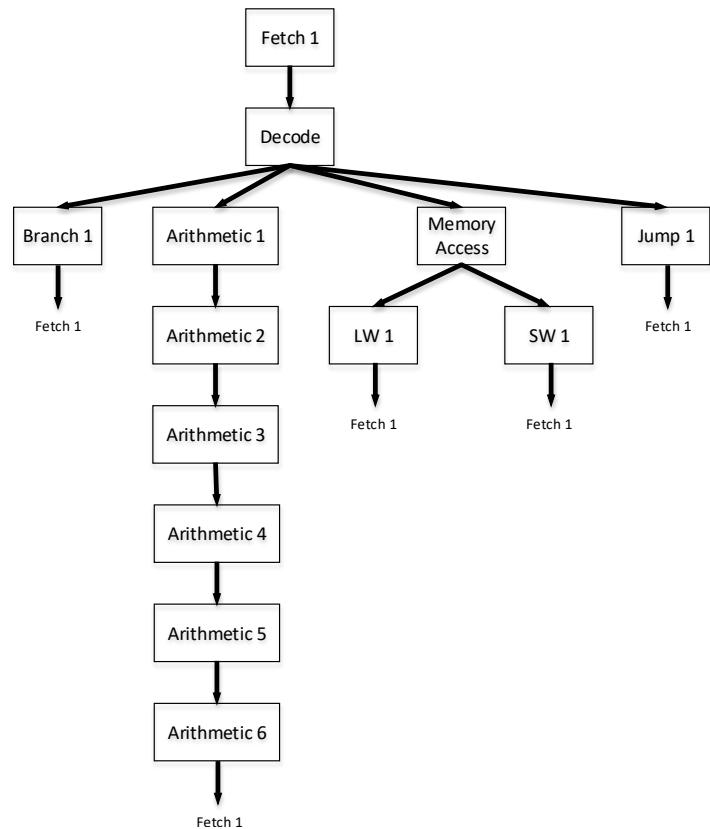


Figure 4: FSM 2

Q2 (10 marks)

In the post-PC era the really valuable resource is energy. Taking power dissipation into consideration in manufacturing a CPU becomes more and more important nowadays.

The Pentium 4 Prescott processor, released in 2004, had a clock rate of 3.6 GHz and voltage of 1.25 V. Assume that, on average, it consumed 10 W of static power and 90 W of dynamic power.

The Core i5 Ivy Bridge, released in 2012, had a clock rate of 3.4 GHz and voltage of 0.9 V. Assume that, on average, it consumed 30 W of static power and 40 W of dynamic power.

1. For each processor, find the average capacitive loads. (Hint: recall that how we approximate dynamic power)
2. Find the percentage of the total dissipated power comprised by static power and the ratio of static power to dynamic power for each CPU.
3. If the total dissipated power is to be reduced by 10 %, how much should the voltage be reduced to maintain the same leakage current? (Hint: power is defined as the product of voltage and current).

$$\begin{array}{lll} 3.6 \text{ GHz} & 1.25 \text{ V} & 10 \text{ W} \\ & & 90 \text{ W} \\ 3.4 \text{ GHz} & 0.9 \text{ V} & 30 \text{ W} \\ & & 40 \text{ W} \end{array}$$

$$\begin{aligned} 1) \text{ For Pentium 4, average capacitive load} &= \frac{2P}{V^2 F} \\ &= \frac{180}{(1.25V)^2 (3.6 \cdot 10^9)} \\ &= 3.2 \cdot 10^{-8} \text{ F} \end{aligned}$$

$$\begin{aligned} \text{For i5, average capacitive load} &= \frac{2P}{V^2 F} = \frac{240 \text{ W}}{(0.9 \text{ V})^2 (3.4 \cdot 10^9)} \\ &= 2.9 \cdot 10^{-8} \text{ F} \end{aligned}$$

$$2) \text{ Pentium 4, percentage static power of total dissipated power} = \frac{10 \text{ W}}{10 \text{ W} + 90 \text{ W}} \cdot 100\% = 10\%$$

$$\text{ratio of static power to dynamic power} = \frac{10 \text{ W}}{90 \text{ W}} = \frac{1}{9} / 0.1$$

$$3) \text{ i5, percentage static power of total dissipated power} = \frac{30 \text{ W}}{30 \text{ W} + 40 \text{ W}} \cdot 100\% = 42.86\%$$

$$\text{ratio of static power to dynamic power} = \frac{30 \text{ W}}{40 \text{ W}} \cdot 100\% = \frac{3}{4} / 0.75$$

$$3) \text{ By } (\text{Static}_{\text{new}} + \text{Dynamic}_{\text{new}}) / (\text{static}_{\text{old}} + \text{Dynamic}_{\text{old}}) = 0.9$$

$$\& \text{Power} = V \cdot I \quad (\text{current same})$$

$$\text{Static}_{\text{new}} = V_{\text{new}} \cdot \left(\frac{10}{1.25}\right) = 8V_{\text{new}}$$

$$\text{Dynamic}_{\text{new}} = 0.9 \cdot 100 - 8V_{\text{new}}$$

$$\therefore V_{\text{new}} = \sqrt{\frac{(90 - 8V_{\text{new}})}{(3.2 \cdot 10^8 \cdot 3.6 \cdot 10^9)}} = 0.85V_{//}$$

$$\text{Static}_{\text{new}} = V_{\text{new}} \cdot (30/0.9) = 33.3V_{\text{new}}$$

$$\text{Dynamic}_{\text{new}} = 0.9 \cdot 70 - 33.3V_{\text{new}}$$

$$\therefore V_{\text{new}} = \sqrt{\frac{(0.9 \cdot 70 - 33.3V_{\text{new}})}{(2.9 \cdot 10^8 \cdot 3.4 \cdot 10^9)}} = 0.64V_{//}$$

Q3 (5 marks)

(1)

Assume it takes one clock to send an address to DRAM memory and one clock to send the corresponding data back. DRAM has 8-cycle latency for first byte, and 4-cycle for each of subsequent bytes in the block. To transfer a 8-byte block, calculate the cycle number if we need:

(1)

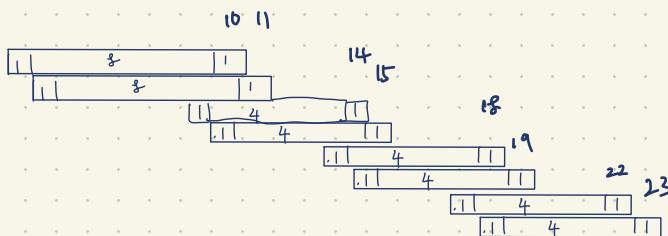
1. Non-interleaving
2. 2-module interleaving

3

(8)

(4)

$$1 + 1 + 8 + 4 \cdot 7 = 38 //$$



$$\therefore \text{Ans} = 23 //$$

Q4 (15 marks)

Figure 1 is the datapath of a RISC-V core architecture.

Consider logic blocks used to implement the datapath as Figure 2 shown, and they have corresponding latency.

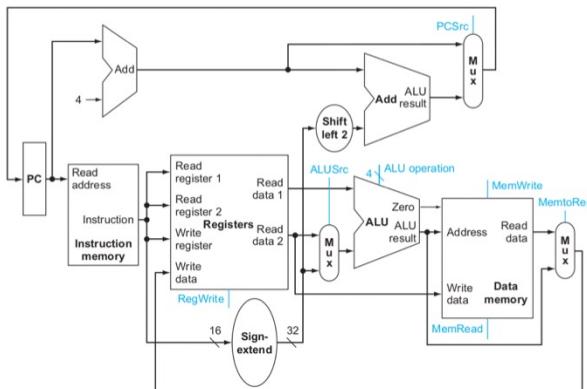


Figure 1: The datapath for the core RISC-V architecture combines the elements required by different instruction classes.

I-Mem/ D-Mem	Register File	Mux	ALU	Adder	Single Gate	Register Read	Register Setup	Sign Extension	Control
NUM_3 ps	200 ps	25 ps	150 ps	100 ps	NUM_1 ps	NUM_2 ps	50 ps	45 ps	40 ps

(317)

Figure 2: The latency of each logic block.

In Figure 2, **Register read** is the time needed after the rising clock edge for the new register value to appear on the output. This value applies to the PC only. **Register setup** is the amount of time a register's data input must be stable before the rising edge of the clock. This value is applied to both the PC and Register File.

- What is the latency of an R-type instruction?
- What is the latency of `ld`?
- What is the latency of `sd`?
- What is the latency of `beq`?
- What is the latency of an I-type instruction ?
- What is the minimum clock period for this CPU?

$$4) \text{ 1) R-type : } 17 + 317 + 200 + 25 + 150 + 25 + 50 = 784 \text{ ps}$$

$$2) \text{ ld } = 17 + 317 + 200 + 25 + 150 + 317 + 25 + 50 = 1101 \text{ ps}$$

$$3) \text{ sd } = 17 + 317 + 200 + 150 + 25 + 317 = 905 \text{ ps}$$

$$4) \text{ beq } = 17 + 317 + 200 + 25 + 150 + 7 + 25 + 50 = 791 \text{ ps}$$

$$5) \text{ I-type } = 17 + 317 + 200 + 25 + 150 + 25 + 50 = 784 \text{ ps}$$

6) 1161 ps_s



Q5 (15 marks)

Consider a RISC-V code snippet.

```
    li t0, 400      1
loop1: lw a0, 0(t0)  +
        lw a1, 400(t0) 6
        addi a2, a0, a1
        sw a2, 0(t0)
        addi t0, t0, -4
        bnez t0, loop1
```

$$400/4 = 100 \text{ time}$$

Assume that the RISC-V design has following features,

- classical 5-stage pipeline: IF, ID, EX, MEM and WB
- Each stage requires only one cycle
- All memory references are hit in cache

1. Calculate how many clock cycles the machine will take execution of the code (Assume no bubble in the pipeline).
2. Calculate how many clock cycles will take execution of the code without forwarding or bypassing when result of the branch instruction (new PC content) is available after the WB stage.
3. Calculate how many clock cycles will take execution of the code with normal forwarding and bypassing when result of branch instruction (new PC content) is available after completion of the ID stage. Meanwhile, calculate the speedup based on the result of the answer of the first question.

1) No. of instruction run = $1 + (6 \cdot (400/4))$
= 601

No of cycle = $601 \cdot 5 = 3005$ clock cycle

2)

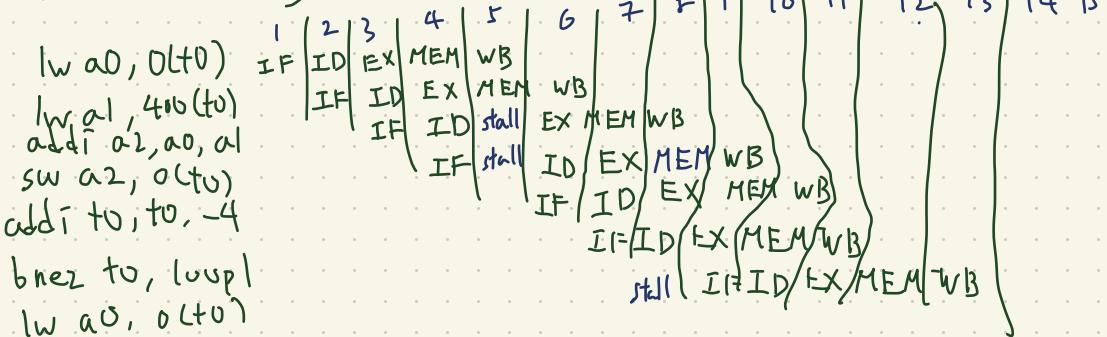
```
    li t0, 400
loop1: lw a0, 0(t0)
        lw a1, 400(t0)
        addi a2, a0, a1
        sw a2, 0(t0)
        addi t0, t0, -4
        bnez t0, loop1
```

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
lw a0, 0(t0)	IF	ID	EX	MEM	WB										
lw a1, 4(t0)(t0)	IF	ID	EX	MEM	WB										
addi a2, a0, a1	IF	ID	stall	stall	EX	MEM	WB								
sw a2, 0(t0)	IF	stall	stall	ID	EX	stall	MEM	WB							
addi t0, t0, -4	IF	ID	stall	EX	MEM	WB									
bneq t0, 0(loop)	IF	stall	ID	stall	stall	EX	MEM	WB							

Each loop need 15 clock cycle

$$\text{Total clock cycle} = 1 + 15 \cdot (400/4) = 1501 \text{ clock cycle}$$

3) With forwarding



$$\therefore \text{Total clock cycle} = 1 + 8 \cdot (400/4) = 801 \text{ clock cycle}$$

Q6 (10 marks)

We have designed a 64-bit address direct-mapped cache, and the bits of address used to access the cache are as shown in Table 1.

Table 1: Bits of the address to use in accessing the cache

Tag	Index	Offset
63-10	9-5	4-0

- What is the block size of the cache in words?
- Find the ratio between total bits required for such a cache design implementation over the data storage bits.
- Beginning from power on, the following byte-addressed cache references are recorded as shown in . Find the hit ratio.

Table 2: Recorded byte-addressed cache references

Hex	00	04	10	84	E8	A0	400	1E	8C	C1C	B4	884	(12)
Dec	0	4	16	132	232	160	1024	30	140	3100	180	2180	

63-10

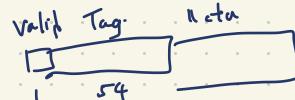
(524)

9-8

(1)

4-0

(5)



1) 4 words

2) By index = 5, Cache line = $2^5 = 32$ lines in cache

$$32 \cdot 4 \cdot 8 \cdot 8 + 54 \cdot 32 + 1 \cdot 32$$

$$\text{ratio} = \frac{32 \cdot 4 \cdot 8 \cdot 8}{2^5 \cdot 4 \cdot 8 \cdot 8} = \frac{9952}{8192} = 1.21\%$$

3) By observation, 0x04, 0x10, 0x8C and 0xB4 hit

$$\text{hit ratio} = \frac{4}{12} = 0.33\%$$

Q7 (15 marks)

Consider that we have a virtual memory system, and key parameters are listed as shown in table 3

- For a single-level page table, find the number of needed page table entries (PTEs) and the physical memory needed to store the page table.
- Multi-level page table mechanism can reduce the physical memory consumption of page tables by only keeping active PTEs in physical memory. How many levels of page tables will be needed if the segment tables (the upper-level page tables) are allowed to be of unlimited size?
- If the segments are limited to the 4KB page size, how many levels of page tables are needed?

Table 3: Virtual Memory System

Virtual Address (bits)	Physical DRAM Installed	Page Size	PTE Size (byte)
43	16 GB	4 KB	4

$$1) \text{ virtual address} = 43 \text{ bit}$$

$$\text{Physical memory} = 16 \text{ GB}$$

$$\text{Page Size} = 4 \text{ KB} = 2^{12} \cdot 8 = 2^{15} \text{ bit}$$

$$\text{Page entry} = 4 \text{ B} = 2^2 \cdot 8 = 2^5 \text{ bit}$$

$$\text{PTE} = 43 - 15 = 28 \text{ bits} = 2^{18} \text{ k entries},$$

$$\text{Physical memory need} = 28 \text{ bit} \cdot 4 \text{ byte} = 2^{18} \cdot 2^2 = 2^{20} \text{ KB},$$

2) If upper-level page table is unlimited size, then there can be unlimited page indexed per page, Then only one-level page setup is enough

3) $\frac{\text{Page Size}}{\text{PTE Size}} = \frac{4 \text{ KB}}{4 \text{ B}} = 2^{10}$ page indexed per page, for 2^{28} PTE,
By $\lceil (28/10)^3 \rceil$, we need 3-level page setup

Q8 ((10 marks))

Consider the following portions of two different programs running at the same time on four processors in a share memory multiprocessor (SMP). Assume that before this code is running, both x and y are 0.

Core1: $x = \text{NUM_2};$

Core2: $y = \text{NUM_3};$

Core3: $w = x + y + 1;$

Core4: $z = x + y;$

1. What are all the possible resulting values of w, x, y, z ? For each possible outcome, explain how we might arrive at those values.
2. How could you make the execution more deterministic so that only one set of values is possible?

$$x = 17$$

$$y = 317$$

$$w = x + y + 1$$

$$z = x + y$$

1)	x	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17
	y	317	317	317	317	317	317	317	317	317	317	317	317	317	317	317
	w	1	18	318	335	1	1	1	18	318	335	1	1	18	318	335
	z	0	0	0	0	17	317	334	17	317	334	17	317	334	17	317

2) Set synchronization instruction after each operation so that all core see same value on all node

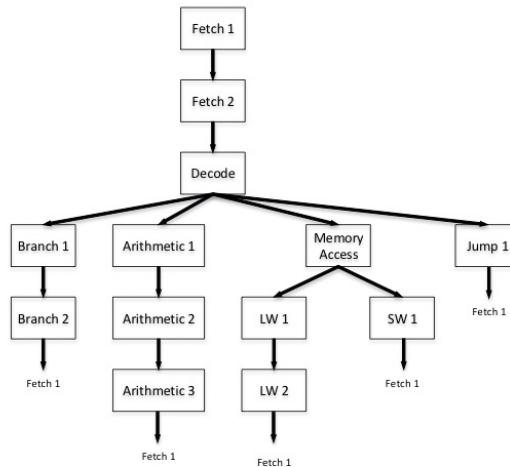


Figure 3: FSM 1

Instruction category	Instructions	Frequency
Arithmetic	ADD, ADDI, AND, OR, SUB	80% - $\frac{15}{2} \cdot 2\%$ - $\frac{1}{2} \cdot 1\%$ = 64.5%
Branch	BEQ, BNEQ	$\frac{17}{2} \cdot 2\% = 17\%$
Jump	JUMP	$\frac{7}{2} \cdot 1\% = 3.5\%$
Memory Access	LW, SW	20% (i.e., 10% loads, 10% stores)

Table 4: Instructions categories & frequency

The instructions are broken down into categories as the Table 4 shows.

1. Calculate the average CPI.
2. Suppose we make a change to the design while we keep the same compiler and the same frequency of instructions. The new FSM is as shown in the Figure 4. Which design is better?

1) Average CPI = $64.5 \cdot 6 + 8.5 \cdot 5 + 7 \cdot 4 + 10 \cdot 6 + 10 \cdot 5$
 ≈ 5.675 cycle

2) For Design 2 , Average CPI = $64.5 \cdot 8 + 8.5 \cdot 3 + 7 \cdot 3 + 10 \cdot 4 + 10 \cdot 4$
 ≈ 6.425 cycle

Although Branch, Jump & Memory Access reduce cycle, since proportion of Arithmetic is highest & it increase cycle, Average CPI of Design 1 < Average CPI of Design 2 , So Design 1 is better.