Computer Architecture I Mid-Term

Chinese Name: _____

Pinyin Name: _____

Student ID: _____

E-Mail ... @shanghaitech.edu.cn: _____

Question	Points	Score
1	1	
2	3	
3	10	
4	7	
5	6	
6	8	
7	4	
8	2	
9	9	
10	13	
11	21	
12	16	
Total:	100	

- This test contains 23 numbered pages, including the cover page, printed on both sides of the sheet.
- We will use gradescope for grading, so only answers filled in at the obvious places will be used.
- Use the provided blank paper for calculations and then copy your answer here.
- Please turn **off** all cell phones, smartwatches, and other mobile devices. Remove all hats and headphones. Put everything in your backpack. Place your backpacks, laptops and jackets out of reach.
- Unless told otherwise always assume a 32bit machine.
- The total estimated time is 120 minutes.

- You have 120 minutes to complete this exam. The exam is closed book; no computers, phones, or calculators are allowed. You may use two A4 pages (front and back) of handwritten notes in addition to the provided green sheet.
- There may be partial credit for incomplete answers; write as much of the solution as you can. We will deduct points if your solution is far more complicated than necessary. When we provide a blank, please fit your answer within the space provided.
- Do **NOT** start reading the questions/ open the exam until we tell you so!

[1] 1. First Task (worth one point): Fill in you name
Fill in your name and email on the front page and your ShanghaiTech email on top of every page (without @shanghaitech.edu.cn) (so write your email in total 23 times).

2. **Various Questions**

3 (a) Name the 6 Great Ideas in Computer Architecture as taught in the lectures.

Solution:
1. Abstraction (Layers of Representation/Interpretation)
2. Moores Law (Designing through trends)
3. Principle of Locality (Memory Hierarchy)
4. Parallelism
5. Performance Measurement and Improvement
6. Dependability via Redundancy

3. **C Basics**

For this part, numbers are represented in 2's complement and stored in little endian. Except for decimals, please keep the leading zeros for full representation of the specified data length. All answers should adhere to the required format indicated by the subscripts.

4 (a) Consider the following numbers are stored in a `signed short`, figure out the arithmetic operations and fill the blanks.

$(\text{-}813)_{10} >> (3)_{10} = (\_\_\_\_\_)_{16}$ $(02D7)_{16} + (00D6)_{16} = (\_\_\_\_\_)_{2}$

$(\text{-}1)_{10}$ & $(\text{-}2)_{10} = (\_\_\_\_\_)_{10}$ $(FD94)_{16} - (727)_{10} = (\_\_\_\_\_)_{16}$

Solution:
$(\text{-}813)_{10} >> (3)_{10} = (FF9A)_{16}$ $(02D7)_{16} + (00D6)_{16} = (0000001110101101)_{2}$

$(\text{-}1)_{10}(\text{-}2)_{10} = (\text{-}2)_{10}$ $(FD94)_{16} - (727)_{10} = (FABD)_{16}$

6 (b) Read the declaration of the following `union` in C.

```
typedef union{
   uint32_t number;
   uint8_t bytes[4];
   struct {
      unsigned int x : 7;
      unsigned int y : 5;
      unsigned int z : 20;
   } data;
} DataType;
```

1. What is the value of `sizeof (DataType)`? _____

> **Solution:** 4

2. Consider the assignment below, fill the following blanks.

```
DataType v;
v.number = 0x08C13D72;
```

`v.data.x` = (_____)$_{16}$ `v.data.y` = (_____)$_{16}$ `v.data.z` = (_____)$_{16}$

> **Solution:**
> `v.data.x` = $(72)_{16}$ `v.data.y` = $(1A)_{16}$ `v.data.z` = $(08C13)_{16}$

3. Consider another assignment below, fill the following blanks.

```
DataType v;
v.data.x = 0x7A;
v.data.y = 0x03;
v.data.z = 0xCEF3D;
```

`v.bytes[0]` = (_____)$_{16}$ `v.bytes[1]` = (_____)$_{16}$

`v.bytes[2]` = (_____)$_{16}$ `v.bytes[3]` = (_____)$_{16}$

> **Solution:**
> `v.bytes[0]` = $(FA)_{16}$ `v.bytes[1]` = $(D1)_{16}$
>
> `v.bytes[2]` = $(F3)_{16}$ `v.bytes[3]` = $(CE)_{16}$

4. **Memory in C**

Consider the following C program, fill in the blanks.

```
#define MAX_NAME_LEN 50
int num_people = 0;
void add_people(char **list){
   char name2[] = "Van";
   list[num_people] = calloc(MAX_NAME_LEN, sizeof(char));
   strcpy(list[num_people], name2);
   num_people += 1;
}
int main(){
   const int list_size = 100;
   char **name_list = malloc(sizeof(char *) * list_size);
   char *name1 = "Billy";
   add_people(name_list);
   add_people(name_list);
   return 0;
```

```
16 }
```

4 (a) Fill in <, >, = or `can't decide` for these four questions based on what the given C expressions evaluate to. You cannot assume `malloc` return heap address sequentially in C standard.

`name_list` _____ `&list_size`

`&name_list` _____ `&num_people`

`name_list[1]` _____ `name_list`

`&name1` _____ `&list`

Solution: 1. <
2. >
3. `can't decide`
4. >

3 (b) Fill in `static`, `stack`, `heap` or `code` for these three questions according to their address type in memory.

`name1` _____

`*name_list` _____

`&(name2[1])` _____

Solution: 1. `static`
2. `heap`
3. `stack`

5. **Superscalar**

[2] (a) Both VLIW and out-of-order superscalar processors exploit instruction-level parallelism. Which one adds more complexity to the hardware and which one adds more complexity to the compiler?

Hardware: _____

Compiler: _____

Solution:
Hardware: out-of-order superscalar
Compiler: VLIW

[4] (b) Are the concepts of superscalar processing and out-of-order execution independent of each other? Why or why not? Explain and justify in no more than 20 words.

_____

_____

Solution:
Yes. A superscalar processor can be built that executes in-order. The same goes for out-of-order execution.

6. **Number Representation**

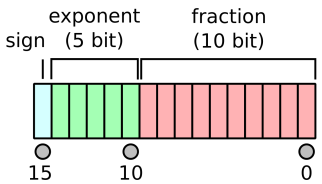
[4] (a) Consider this 8-bit binary pattern 0b11001010, please write down this number if we are using the following representations:

Unsigned binary _____ Sign-Magnitude binary _____

Two's complement binary _____ Hexadecimal _____

Solution:
202 -74 -54 0xCA

[4] (b) Suppose we are using half-precision floating-point (16-bit) format (like on NVIDIA GeForce FX). The layout for the 16-bit floating point is:



Everything else follows the IEEE 754 standard for floating point, except bias. Answer the following questions.

What is the bias? _____

What is the smallest positive denorm? _____

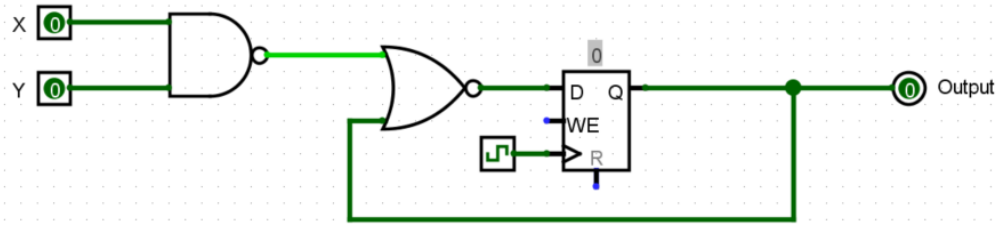
Convert -10.8125 to 16-bit floating point. Write in hexadecimal. _____

Convert 0xCA20 into decimal. _____

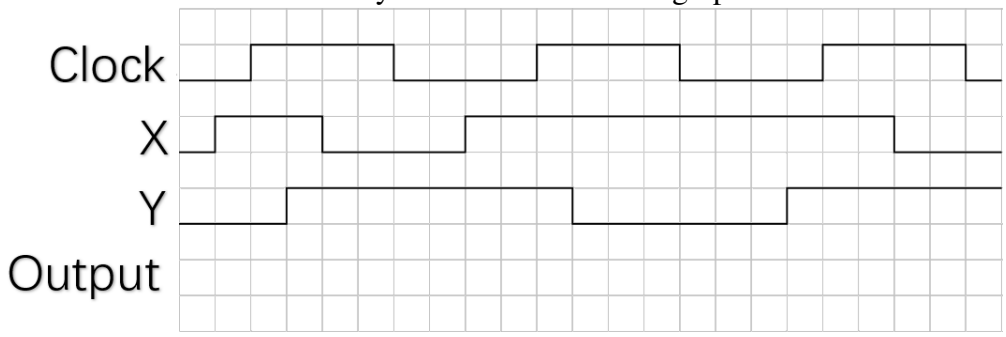
Solution:
15 2^{-24} 0xC968 -12.25

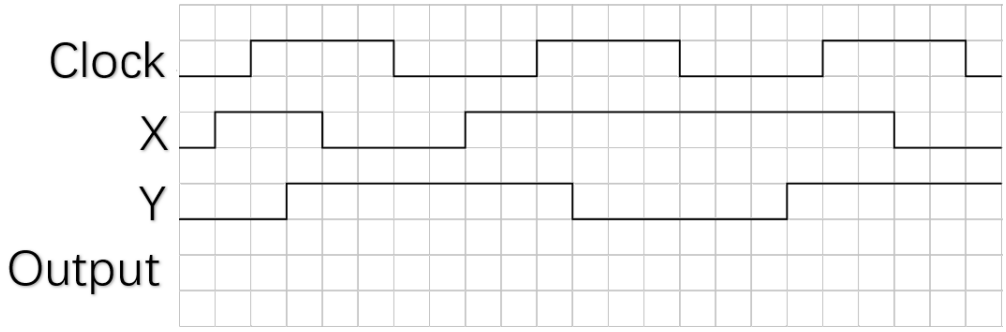
7. **SDS**

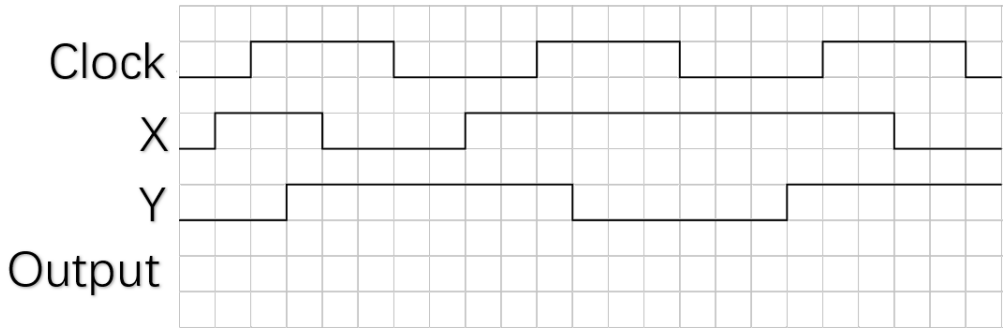
4 (a) Draw the Timing Diagram for the circuit below. The delay for the gates are 10 ns, the clock-to-q delay for a register is 20 ns, each clock cycle is 80 ns, each grid in the following diagram is a unit of 10 ns. The output is initially given in the graph.



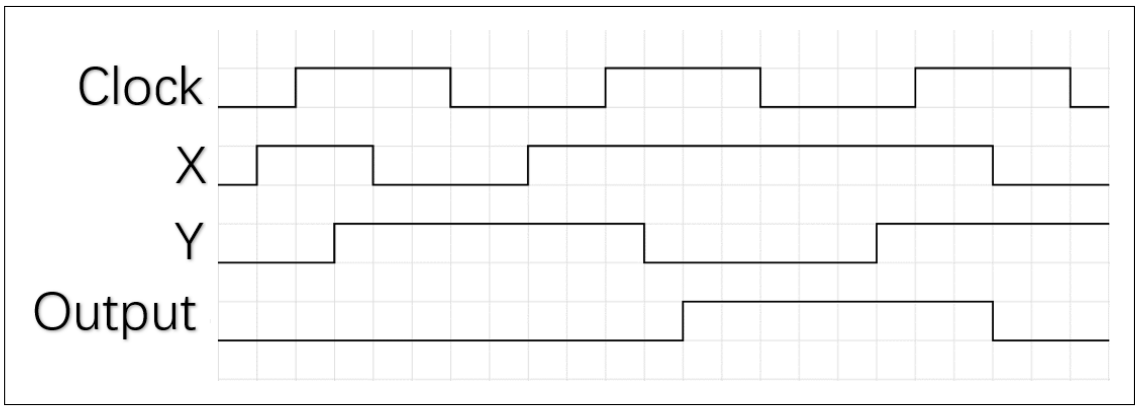
Use any of those graphs to put in your answer (so you can re-do it). Clearly mark your final answer if you use more than one graph!



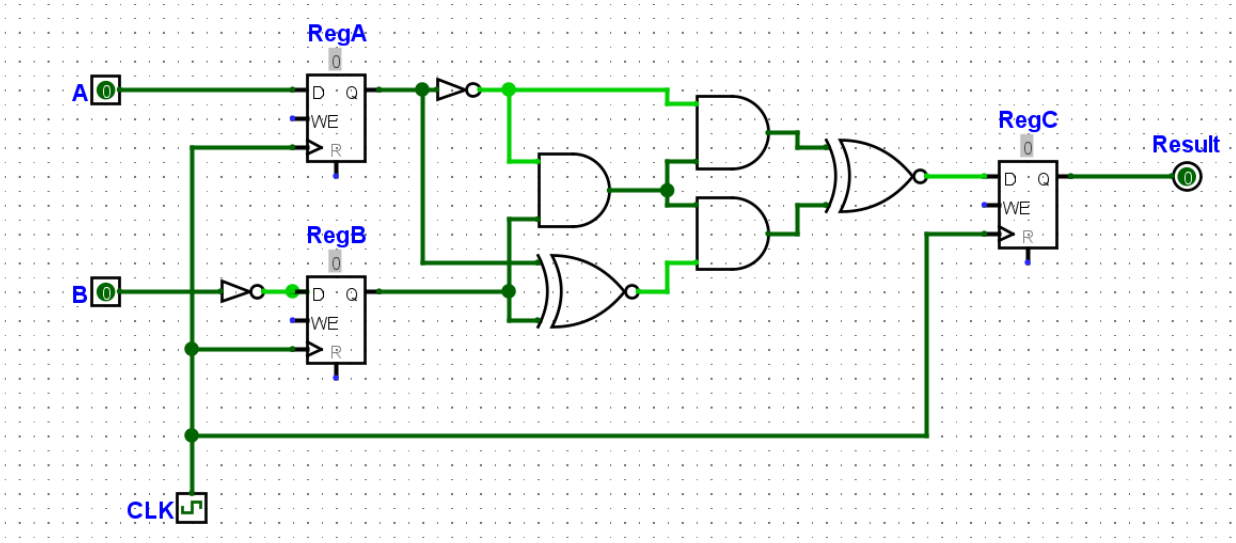




Solution:


Clock
X
Y
Output

8. **Circuit time calculation**
In this circuit below, RegA and RegB have setup, hold and clk-to-q times of 8ns, NOT logic gate has a delay of 1ns, AND logic gate has a delay of 3ns, XNOR logic gate has a delay of 5ns, and RegC has a setup time of 9ns.



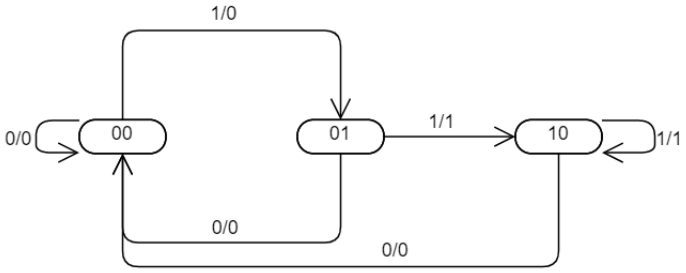
2 (a) What is the minimum acceptable clock cycle time for this circuit, and the clock frequency this corresponds to?

Clock cycle time:

Clock frequency:

Solution:
minimum clock cycle time = 8 + 13 + 9 = 30ns
clock frequency = $\frac{1}{30*10^{-9}} Hz$ = 33.3MHz

9. **FSM and Truth Table**


1/0
0/0
00
01
1/1
10
1/1
0/0
0/0

2 (a) Fill in the truth table for the FSM.

state bit1	state bit0	input	next state bit1	next state bit0	output
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			

Solution:

state bit1	state bit0	Input	next state bit1	next state bit0	Output
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	1
1	0	0	0	0	0
1	0	1	1	0	1

2 (b) Using st1(state bit1), st0(state bit0) and ip(Input) as the input and Output as the output, extract a boolean expression from your table.

Solution:

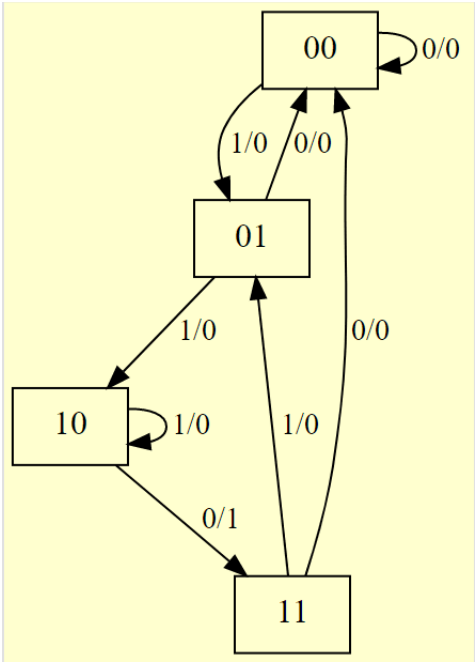
$$output = \bar{st1} \cdot st0 \cdot ip + st1 \cdot \bar{st0} \cdot ip$$

2 (c) What does the given FSM implement (Describe when the FSM will output 1)?

Solution: When the FSM receives two or more successive 1, it will output 1.

3 (d) Extend and modify the given FSM to make it output 1 if and only if the FSM receives input sequence including ”110”
Draw the diagram below:

Solution:


00
0/0
1/0
0/0
01
1/0
0/0
10
1/0
1/0
0/1
11

10. **RISC-V Programming**

Look at the following RISC-V function that sorts an word array in-place, whose start address is given in a0 and length given in a1.

```
1  sort:
2     # prologue ......
3     mv s0, a0
4     0x00259493  # p1
5     addi s1, s1, -4
6     add s1, s1, s0
7     mv t0, s0
8  outer_loop:
9     mv t1, t0
10    0x00032383  # p2
11    mv t3, t1
12 inner_loop:
13    addi t1, t1, 4
14    lw t4, 0(t1)
15    ble t4, t2, mystery_label
16    mv t2, t4
17    mv t3, t1
18 mystery_label:
19    blt t1, s1, inner_loop
20    lw t5, 0(t0)
21    sw t2, 0(t0)
22    sw t5, 0(t3)
23    addi t0, t0, 4
24    blt t0, s1, outer_loop
25    # epilogue ......
26    ret
```

2 (a) Please disassemble machine code marked #p1 and #p2 above to RISC-V instructions. (Please use register names, e.g. s0, s1, etc., **NOT** x8, x9, etc.)

0x00259493 # p1: _____

0x00032383 # p2: _____

Solution: p1: slli s1, a1, 2 p2: lw t2, 0(t1)

2 (b) How many different types of pseudo instructions appeared above? (Instructions with different names are considered different types.) Please list them.

_____

Solution: 3; mv, ble, ret

4 (c) "li x1, 0xDCBAABCD" is also a pseudo instruction. Below it's expanded to 2 normal instructions. Please fill in the blanks and translate each of them to machine code.

lui x1, 0x_____ : 0x_____

addi x1, _____ : 0x_____

Solution: lui x1, 0xDCBAB: 0xDCBAB0B7
addi x1, x1(ra), -1075: 0xBCD08093

`memset()` is a function defined in standard C library, it fills a byte string with a byte value and can be implemented as follows:

```
#include <stddef.h>

void *memset(void *dst, int c, size_t len) {
    char *start = dst, *end = start + len;
    for (char *ptr = dst; ptr < end; ++ptr) {
        *ptr = (unsigned char) c;
    }
    return dst;
}
```

5 (d) Please implement `memset()` function in RISCV assembly. Your `memset()` function should fills the first `len` bytes of the memory area pointed to by `dst` with the constant byte `c` and returns its first argument.

```
memset:
        _____________________________________

        _____________________________________
loop:

        _____________________________________
        ret
continue:

        _____________________________________

        _____________________________________
        j     loop
```

Solution:

```
memset:
        add   a2, a0, a2
        mv    a3, a0
loop:
        bltu  a3, a2, continue
        ret
continue:
        sb    a1, 0(a3)
        addi  a3, a3, 1
        j     loop
```

11. **Cache**

2 (a) The Average Memory Access Time equation (AMAT) has three components: hit time, miss time, and miss penalty. For each of the following cache optimizations, indicate which component of the L1 AMAT equation may be ***improved***. Circle one.

Using a second-level cache	hit time	miss rate	miss penalty
Using larger blocks	hit time	miss rate	miss penalty
Using a smaller first-level cache	hit time	miss rate	miss penalty
Using a larger first-level cache	hit time	miss rate	miss penalty

Solution:
miss penalty miss rate hit time miss rate

6 (b) Consider a 32-bit physical memory space and a 32 KiB 2-way set associative cache with LRU replacement. You are told the cache uses 5 bits for the offset field.

1. Write in the number of bits in the tag and index fields.

TAG	Set index	Block offset
		5 bits

Solution: 18 bits 9 bits

2. Given the following C source code,

```
1 int ARRAY_SIZE = 64 * 1024;
2 int arr[ARRAY_SIZE]; // *arr is aligned to a cache block
3
4 /* loop 1 */
5 for (int i = 0; i < ARRAY_SIZE; i += 8) arr[i] = i;
6 /* loop 2 */
7 for (int i = ARRAY_SIZE - 8; i >= 0; i -= 8) arr[i+1] =
     arr[i];
```

What is the hit rate of loop 1? What types of misses (of the 3 Cs), if any, occur as a result of loop 1?

Solution: 0% hit rate, Compulsory Misses

What is the hit rate of loop 2? What types of misses (of the 3 Cs), if any, occur as a result of loop 2?

Solution: 9/16 (56.25%) hit rate, Capacity Misses

3 (c) This section involves T / F questions. Circle the correct answer. **Notice: NO selection will be treated as a wrong choice.**

T / F: The local miss rate of one level of a cache is always greater than the global miss rate of that cache.

T / F: Any cache miss that occurs when the cache is full is a capacity miss.

T / F: The only way to remove capacity miss is to increase the cache capacity.

T / F: For the same cache size and block size, a 4-way set associative cache will have more index bits than a direct-mapped cache.

T / F: The hit rate of a combined cache is usually worse than the two split caches which have the same size in sum with the combined cache.

T / F: The index of a cache block, together with the tag contents of that block, uniquely specifies the memory address of the word contained in the cache block.

Solution: F F T F F T

4 (d) AMAT Calculation

Suppose your system consists of:

- An L1 cache that has a hit time of 5 cycles and has a local miss rate of 20%.
- An L2 cache that has a hit time of 20 cycles and has a local miss rate of 15%.
- An L3 cache that has a hit time of 200 cycles and has a local miss rate of 5%.
- Main memory hits in 1000 cycles.

Notes: You should show your calculation process. Only giving a solution will receive no point.

1. What is the global miss rate?

Solution: Global miss rate $= 20\% \times 15\% \times 5\% = 0.15\%$

2. What is the AMAT of the system?

Solution: AMAT = $5 + 20\% \times (20 + 15\% \times (200 + 5\% \times 1000)) = 16.5$ cycles

6 (e) Consider the following program and cache behaviors.

Suppose a CPU with a write-through, write-allocate cache achieves a CPI of 2. What are the read and write bandwidths (measured by bytes per cycle) between RAM and the cache? (Assume each miss generates a request for one block.)

Data Reads per 1000 Instructions	Data Writes per 1000 Instructions	Instruction Cache Miss Rate	Data Cache Miss Rate	Block Size (bytes)
250	150	0.30%	2%	64

Notes: You should show your calculation process. Only giving a solution will receive no point.

Solution:

When the CPI is 2, there are on average 0.5 instruction accesses per cycle. 0.30% of these instructions accesses cause a cache miss and subsequent memory request. Assuming each miss requests one block, instruction accesses generate an average of $0.5 \times 0.30\% \times 64 =$ **0.096** bytes/cycle.

25% of instructions generate a read request, and 2% of these generate a cache miss. So read misses generate an average of $0.5 \times \frac{250}{1000} \times 2\% \times 64 =$ **0.16** bytes/cycle of read traffic.

10% of instructions generate a write request, and 2% of these generate a cache miss. Because the cache is a write-through cache, only one word (8 bytes) must be written back to memory; but every write is written through to memory (not just the cache misses). Thus, write misses generate an average of $0.5 \times \frac{100}{1000} \times 8 = 0.4$ bytes/cycle of write traffic. Because the cache is a write-allocate cache, a write miss also makes a read request to RAM. Thus, write misses require an average of $0.5 \times \frac{100}{1000} \times 2\% \times 64 =$ **0.064** bytes/cycle of read traffic.

The total read bandwidth $= 0.096 + 0.16 + 0.064 =$ **0.32** bytes/cycle
The total write bandwidth is **0.4** bytes/cycle.

12. **RISC-V pipelining**

2 (a) please **circle** the correct answer. **Notice: NO selection will be treated as a wrong choice.**

T / F: Pipelining the CPU datapath results in instructions being executed with higher latency and throughput

T / F: Without forwarding, data hazards will usually result in 3 stalls

T / F: All data hazards can be resolved with forwarding

T / F: Control hazards are caused byjump and branch instructions

Solution: TTFT

The delays of circuit elements of a datapath are given as follows:

Element	Register clk-to-q	Register Setup	MUX	ALU	Mem Read	Mem Write	RegFile Read	RegFile Setup	branch comp
Parameter	$t_{clk-to-q}$	t_{setup}	t_{mux}	t_{ALU}	$t_{MEMread}$	$t_{MEMwrite}$	t_{RFread}	$t_{RFsetup}$	t_{Bcomp}
Delay(ps)	30	20	25	200	150	125	130	20	75

Answer the following questions.

2 (b) What was the clock time and frequency of a single cycle CPU ?

Solution: ~~750ps 1.33GHz~~ 730

2 (c) What is the clock time and frequency of a pipelined CPU?

Solution: ~~300ps 3.33GHz~~ 275

2 (d) What is the speed-up? Why is it less than five?

Solution: 2.5 This is because pipeline stages are not balanced evenly and there is overhead from pipeline registers

Consider the following 3 datapaths:

The execution time of these datapaths are listed below:

	Stage1	Stage2	Stage3	Stage4	Stage5	Stage6
Datapath1	IF	ID	EXE	MEM	WB	-
Datapath2	IF	ID	EXE1	EXE2	MEM	WB
Datapath3	IF/ID	EXE1	EXE2	MEM	WB	-

	Stage1	Stage2	Stage3	Stage4	Stage5	Stage6
Datapath1	200ps	150ps	350ps	170ps	130ps	-
Datapath2	220ps	180ps	100ps	200ps	250ps	150ps
Datapath3	400ps	200ps	200ps	250ps	150ps	-

[1] (e) Which datapath is the same as RISC-V datapath as learned in class?

Solution: Datapath1

[1] (f) **Without** pipelining, what's the maximum clock rate of **Datapath3**?

Solution: $fs_{max} = 1/(400 + 200 + 200 + 250 + 150)ps = 0.875GHz$

[1] (g) What method can you use to improve performance?

Solution: Pipeline.

[3] (h) **With** pipelining, what's the maximum clock rate of each datapath?

Solution:

- Datapath1: $fs_{max} = 1/\max(200, 150, 350, 170, 130)ps = 2.86GHz$
- Datapath2: $fs_{max} = 1/\max(220, 180, 100, 200, 250, 150)ps = 4GHz$
- Datapath3: $fs_{max} = 1/\max(400, 200, 200, 250, 150)ps = 2.5GHz$

2 (i) In this question, you'll only need to consider **Datapath1**. Which of the following instruction(s) exercise the critical path?

A. `add`
B. `lw`
C. `mul`

Solution: B.

No question here!