Computer Architecture I Mid-term Exam 2

Chinese Name: ______________________________

Pinyin Name: ______________________________

Student ID: ______________________________

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

Question	Points	Score
1	1	
2	14	
3	12	
4	8	
5	18	
6	6	
7	30	
8	4	
Total:	93	

- This test contains 16 numbered pages, including the cover page, printed on both sides of the sheet.
- We will use blackboard 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 16 times).

14 2. **MISC [14 points]**

(1) We define `#define MAG0(x,y)sqrt(x*x+y*y)`, what does `MAG0(2+1,3+1)` evaluates to? (Hint: `sqrt` performs square root operation)____[1 point]

A. 5 B. $\sqrt{32}$

C. $\sqrt{13}$ D .$\sqrt{12}$

Solution: D

(2) For `jal` instruction, what is written back to the destination register `rd`?____[1 point]

A. PC B. PC+4

C. PC+immediate

Solution: B

(3) Assume we define a pointer variable, pointing to a function. This variable is stored in `x5` register. Which of the following RISC-V instruction(s) can we use to call the function directly?______ [1 point]

A. `jal`. B. `bne`.

C. `jalr`. D. `beq`.

E. None of above.

Solution: C

(4) Rounding 0.5 to an integer using "round-to-nearest-ties-to-even", it evaluates to _______ [1 point]

A. 0 B. -1

C. 1 D. 2

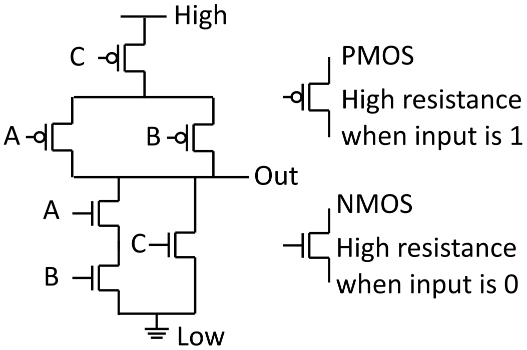
Solution: A

(5) What is the logic expression for the following circuit? (Hint: you can use the truth table to obtain the logic expression.)____[2 points]

A. Out $= \mathrm{AB} + \mathrm{C}$ B. Out $= (\bar{\mathrm{A}} + \bar{\mathrm{B}})\bar{\mathrm{C}}$

C. Out $= (\mathrm{A} + \mathrm{B})\mathrm{C}$ D. Out $= \bar{\mathrm{A}}\bar{\mathrm{B}} + \bar{\mathrm{C}}$

Solution: B



(6) **(True of Flase)** In a Moore FSM, the output depends only on the current state; in a Mealy FSM, the output depends on both the current state and the inputs._____ [1 point] (Note that an FSM can have both Moore and Mealy outputs.)

Solution: True

(7) **(True of Flase)** In a pipelined CPU, each instruction consumes less time to complete than its single-cycle version. _____ [2 points]

Solution: False

(8) **(True of Flase)** Using separate instruction and data caches helps to eliminate structural hazards in the classic 5-stage pipeline. _____ [2 points]

Solution: True

(9) **(True of Flase)** As CPU cache is a fast but small buffer for main memory in computer architecture, to keep duplicate copies of the same cache line across multi-level cache hierarchy is a waste of such valuable hardware resource and not supported. _____ [2 points]

Solution: False

(10) **(True of Flase)** Given a last-level CPU cache that follows the write-back policy, a cache replacement that happens to a dirty cache line in it would not write the cache line to main memory, as it does not use the write-through policy. _____ [2 points]

Solution: False

(11) Memory management. Define a struct

```
typedef struct Node{
int val;
struct Node *next;
} node;
```

and we build a function

```
void push_node(node ** head, int val){
node * new_node;
new_node = (node *) malloc (sizeof (node));
new_node -> val = val;
new_node -> next = *head;
*head = new_node;
}
```

When we run `push_node`, the pointer variable `new_node` is the most likely in _____. `new_node -> val` is in _____. `new_node -> next` is in ______.[3 points]

A. Code/text.
B. Stack.
C. Static/data.
D. Heap.

Solution: B, D, D

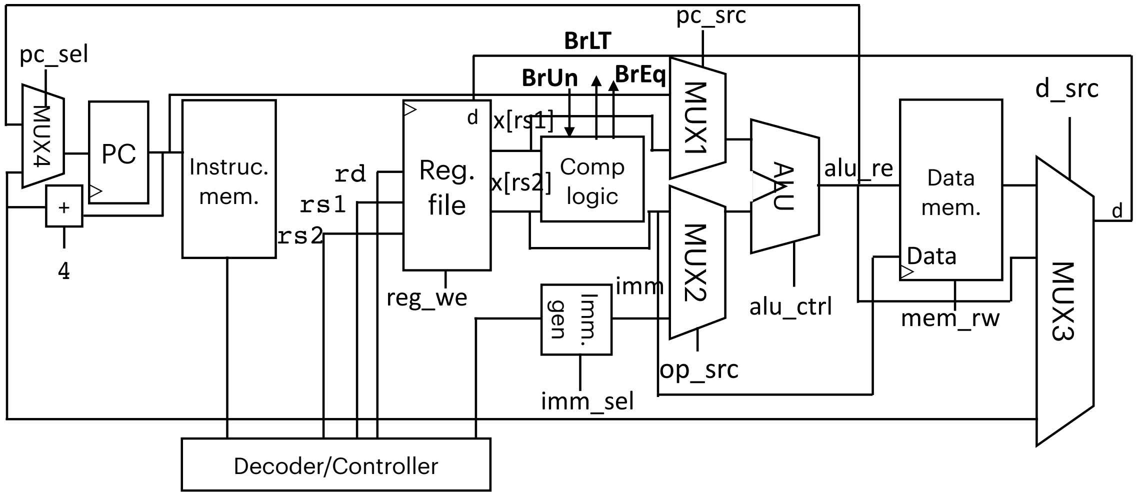
(12) Choose below the fastest (______) and slowest (______) memory components for data access. [2 points]

A. CPU registers.
B. Main memory (DRAM).
C. Cache (SRAM).
D. Hard disk drive.

Solution: A, D

12 3. **Single-cycle CPU datapath [12 points]**

Below is the single-cycle CPU datapath we have learnt in the lectures.



(1) Which of the following type(s) of instructions experience(s) the longest propagation delay?________ [2 points]

A. Conditional branch.
B. Load.
C. I-type arithmetic and logic.
D. Store.

Solution: B

(2) For an immediate (I-) type arithmetic and logic instruction, what is the total propagation delay? ______ (The subscript indicates the component/type of the delay.) [2 points]

A. $t_{\text{clock-to-q}} + t_{\text{MUX}} + t_{\text{PC}} + t_{\text{instruction memory}} + t_{\text{decoder/controller}} + \max\{t_{\text{Reg. file}}, t_{\text{immediate generation}}\} + t_{\text{MUX}} + t_{\text{data memory}} + t_{\text{MUX}} + t_{\text{setup time}}$.

B. $t_{\text{clock-to-q}} + t_{\text{instruction memory}} + t_{\text{decoder/controller}} + \max\{t_{\text{Reg. file}}, t_{\text{immediate generation}}\} + t_{\text{MUX}} + t_{\text{ALU}} + t_{\text{MUX}} + t_{\text{setup time}}$.

C. $t_{\text{clock-to-q}} + t_{\text{MUX}} + t_{\text{adder}} + t_{\text{PC}} + t_{\text{setup time}}$.

D. None of the above.

Solution: B

(3) For an R-type instruction, select from below which piece of data is selected for multiplexers MUX1, MUX2, MUX3 and MUX4? What about U-type `auipc` instruction? [8 points]

R-type instructions:

MUX1_A__

MUX2_A__

MUX3_F_

MUX4__C__

U-type `auipc`:

MUX1__B__

MUX2__D__

MUX3__F__

MUX4__C__

A. Data from the register file.

B. Data from the PC register (value of `PC`).

C. Data from the PC register (value of `PC+4`).

D. Data of the immediate.

E. Data from the data memory.

F. Data from the ALU calculation.

$\boxed{8}$ 4. **Performance** [8 points]

(1) Assume a program of 4×10^6 instructions runs on a processor with a clock rate of 2.5 GHz. The average CPI of the program is 2. Please estimate the runtime of the program. [2 points]

$$\begin{aligned} \text{Runtime} &= 4 \times 10^6 \times 2 \times (1/2.5 \text{ GHz}) \\ &= 3.2\text{ms} \end{aligned} \tag{1}$$

(2) For a program, we can also analyze the portion of each type of the instructions and estimate the total runtime based on the CPI of each type of instruction, i.e.,

$$\frac{\text{Runtime}}{\text{Program}} = \sum_{\text{X-type instruction}} \left(\frac{\text{Number}_{\text{X-type instructions}}}{\text{Program}} \cdot \text{CPI}_{\text{X-type instruction}} \cdot t_{\text{clock cycle}} \right) \tag{2}$$

where X can be R-type, I-type arithmetic and logic, branch, U-type, load, store and unconditional jump, etc.

Assume an ISA that contains only 4 types of instructions, A-, B-, C-, and D-type. For a processor P, the CPIs of the 4 types of instructions are 2, 2, 1 and 3, respectively. Consider a 10^6-instruction program, Pro, with 10% A-type, 30% B-type, 40% C-type and 20% D-type instructions, what is the average CPI of processor P running Pro? Assume the clock rate of P is 1 GHz. [6 points]

Method 1:

$$CPI = 2 \times 10\% + 2 \times 30\% + 1 \times 40\% + 3 \times 20\% = 1.8 \tag{3}$$

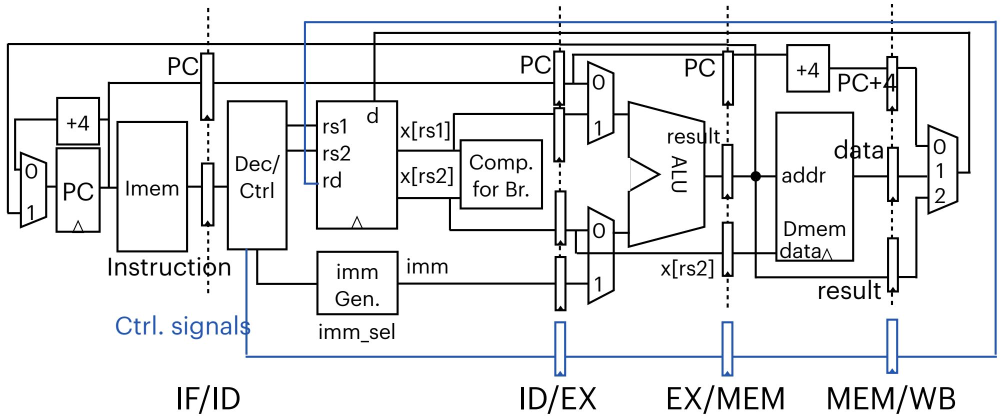
Method 2:

Total runtime = $(2 \times 10^6 \times 10\% + 2 \times 10^6 \times 30\% + 1 \times 10^6 \times 40\% + 3 \times 10^6 \times 20\%) \times (1/1 \text{ GHz}) = 1.8$ ms.

Average CPI = Total runtime/#instruction/Clock cycle = 1.8 ms/10^6/(1/1 GHz) = 1.8

18 5. **Pipeline** [18 points]

Consider the five-stage pipelined processor we covered in the lectures with the following stages: Instruction fetch (IF), instruction decode and register read (ID), execute (EX), memory access (MEM), and write back (WB). Assume that the processor has no forwarding or hazard detection mechanisms implemented as shown below.



The given code is executed on the processor:

```
1 ADD R1, R2, R3
2 SUB R4, R1, R5
3 AND R6, R1, R7
4 OR  R8, R6, R9
```

(1) Identify the hazards, the hazard type, and the registers and/or instructions that cause them in the code. Point out all of the hazards using the following table. [6 points]

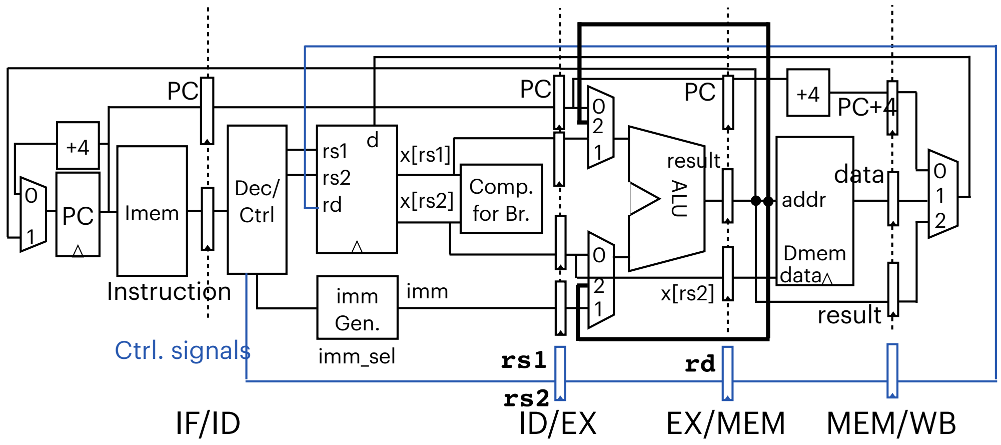
Instructions and/or registers cause the hazard	Hazard's type
Instruction 1/2 R1	Data hazard
Instruction 1/3 R1	Data hazard
Instruction 3/4 R6	Data hazard

(2) Count the number of stall cycles (by inserting `nops`) required to resolve the above hazards.

Total number of stall cycles: __6__. [2 points]

(3) Now we add forwarding/bypassing to the pipelined CPU as shown below. Complete the following timing table. Fill in the pipeline stage that the instruction is experiencing at a certain clock cycle and make sure the instructions execute correctly. "CC" stands for "clock cycle". (Hint: see `ADD` as an example. If you feel stalls are required, delay the

instruction directly instead of inserting `nop` explicitly. We assume all the sequential components are triggered on the rising clock edge and the rising clock edge appears between adjacent clock cycles.) [6 points]



CC	1	2	3	4	5	6	7	8	9	10
ADD	IF	ID	EX	MEM	WB					
SUB		IF	ID	EX	MEM	WB				
AND					IF	ID	EX	MEM	WB	
OR						IF	ID	EX	MEM	WB

Note: we only add one forwarding path in the diagram, so the hazard between instructions 1 and 3 cannot be avoided, and thus the `AND` instruction does not "ID" until the `ADD` instruction "WB".

The value of the pipeline register at stage __B__ is forwarded. [2 points]

A. IF/ID. B. EX/MEM.

C. ID/EX. D. MEM/WB.

(4) Name at least two other techniques that can be employed to mitigate pipeline hazards and improve the processor's performance. [2 points]

Solution:

Code scheduling

Branch prediction

6. C programming language

6 (a) Considering the following piece of C code:

```c
#include <stdio.h>
#include <stdlib.h>

typedef struct student {
   char *name;
   int age;
} student_t;

student_t *new_student(char *name, int age) {
   student_t *student = malloc(sizeof(student_t));
   student->name = name;
   student->age = age;
   return student;
}

void delete_student(student_t *student) {
   free(student->name);
   free(student);
}

student_t test_student;

int main(int argc, char *argv[]) {
   printf("argc = %d\n", argc);
   printf("argv = %p\n", (void *) argv);
   for (int i = 0; i < argc; ++i) {
      printf("argv[%d] = %s\n", i, argv[i]);
   }
   new_student("John", 20);
   printf("test_student.age = %d\n", test_student.age);
   printf("test_student.name = %s\n", test_student.name);
   delete_student(&test_student);
   return 0;
}
```

Now we compile and run the code above with the following command:

```
gcc -Wall -Wextra -Werror -Wpedantic -O3 -std=c11 main.c &&
   ./a.out 135 qwe
```

Please choose all correct options for the following questions:

(i) Which of the following statements is correct?

A. Compile error

B. Runtime error due to visiting uninitialized memory

C. Runtime error due to invalid free

D. The program exits with code 0 normally

E. None of the above

(ii) We first fix the compile error/runtime error bug (if any) by commenting that line of code. What will Valgrind complain of the code?

A. Everything is fine

B. Invalid read at line 27

C. Memory allocated at line 10 not freed

D. Invalid free at line 17

E. Visiting uninitialized memory at line 31

(iii) Which of the following statements is correct?

A. The string literal "$John$" is stored in the read-only section of the executable

B. "$test\_student$" is a mutable static variable

C. Taking the address of a string literal is legal: $\&$"$John$"

D. $argv[argc]$ is guaranteed to be initialized to $NULL$

E. An undefined behavior occured during execution

Solution:

(i) C

(ii) C: Note that argv[argc] is always NULL

(iii) ABCD

7. **Cache**

10

(a) Suppose a computer has an 8-bit address space. It has a 2-way set-associative cache with LRU replacement policy. Cache size is 64 Bytes and block size is 8 Bytes.

1. What is the length of tag, set index and block offset fields?

____

2. The addresses of memory access are as follows; the sequence is from the top to bottom in the table. Fill in the blanks.

Address	Hit or Miss
0b10001000	Miss
0b10011000	Miss
0b10001001	Hit
0b00001000	Miss
0b00001001	
0b10010001	
0b11001001	
0b00001010	
0b10001001	

3. If we modify the cache to direct mapping and other conditions remain the same (8-bit address space, cache size is 64 Bytes and block size is 8 Bytes). What is the hit rate of the memory accesses in the above table?

____

Solution: 3; 2; 3.

Address	Hit or Miss
0b10001000	Miss
0b10011000	Miss
0b10001001	Hit
0b00001000	Miss
0b00001001	Hit
0b10010001	Miss
0b11001001	Miss
0b00001010	Hit
0b10001001	Miss

Solution: $\frac{2}{9}$

6 (b) The following code describes two loops that calculate the sum of a 32 by 32 matrix of 4-byte integers. The array is stored contiguously in memory in row-major order.

```
int array[1024];
int sum = 0;
void loopA(){
   sum = 0;
   for (int i = 0; i < 32; i++){
      for (int j = 0; j < 32; j++){
         sum += array[i][j];
      }
   }
}
void loopB(){
   sum = 0;
   for (int j = 0; j < 32; j++){
      for (int i = 0; i < 32; i++){
         sum += array[i][j];
      }
   }
}
```

1. Assume we have a computer with a 2KB direct-mapped L1 data cache and 32-bit address space. The cache block size is 16 Bytes.

   The number of cache misses when running loop A: ______________________

The number of cache misses when running loop B: ____________________

Assuming we run loop A **an infinite number of times**, what number will the hit rate converge to?

____

2. Assume we have a computer with an 8KB direct-mapped L1 data cache and 32-bit address space. The cache block size is 16 Bytes.

   Which type(s) of miss occur(s) when running loop A? ____________________

   The hit rate when running loop A: ____________________

   The hit rate when running loop B: ____________________

Solution:
256; 1024; 0.75
Compulsory/Cold Miss; 0.75; 0.75

4 (a) **True or False**

i. Local miss rate of L2 cache is smaller than the global miss rate

ii. In a RISC-V CPU with a L1 data cache, a L1 instruction cache and a general L2 cache, only store and load instructions will cause L2 cache misses

iii. AMAT of L1 cache equals to (L1_hit_time+L1_local_miss_rate*AMAT_of_L2)

iv. multi level cache can take use of the locality on misses from a cache

i	ii	iii	iv

Solution:

i	ii	iii	iv
F	F	T	T

3 (b) Suppose you have the following system that consists of:

- L1 cache with a hit time of 2 cycles
- L2 cache with a hit time of 10 cycles
- DRAM with an access time of 100cycles

After an amount of memory accesses, we get the following data:

- L1 cache is accessed by 100 times
- L2 cache is accessed by 50 times
- DRAM is accessed by 20 times

Please answer the questions:
Global miss rate:____________________
L2 cache local hit rate:____________________
AMAT of L1 cache:______________________

Solution: Global miss rate: 20%
L2 cache local hit rate: 60%
AMAT of L1 cache: 27 cycles

2 (c) If we want to improve AMAT of L1 cache to make it not greater than 17 cycles, the maximum local miss rate of L2 cache is:

Solution:

$$AMAT = 2 + 0.5 * (10 + x * 100) = 7 + 50x \leq 17$$
$$x \leq 0.2$$

5 (d) Suppose the local hit rate of each level cache is $\frac{1}{2}$, and the hit time of n-level cache is 4^n cycles. For example, L2 cache has a hit time of $4^2 = 16$ cycles and L3 cache has a hit time of $4^3 = 64$ cycles. The access time of DRAM is 1024 cycles. If we want the AMAT of L1 cache to be 124 cycles, how many level caches do we need?

Solution:

$$\begin{aligned}AMAT =& 4^1 + \frac{1}{2}(4^2 + \frac{1}{2}(4^3 + ... + \frac{1}{2}(4^n + \frac{1}{2} * 1024)))\\ =& 2^2 + 2^3 + 2^4 + ... + \frac{1}{2^{n-1}} * 4^n + \frac{1}{2^n} * 1024\\ =& 2^2 + 2^3 + 2^4 + ... + 2^{n+1} + \frac{1}{2^n} * 1024\\ =& 4\frac{1 - 2^n}{1 - 2} + \frac{1024}{2^n}\\ =& 4 * 2^n + \frac{1024}{2^n} - 4\\ \geq& 2\sqrt{4 * 1024} - 4 = 124\end{aligned}$$

Thus, we need 4 level caches.

4 8. **SIMD** [4 points]

Can the following for-loop be unrolled?

If yes, loop-unroll the following program using the template provided below.

If not, explain your reason.

```
for (i=2; i<10; i++)
{
   a[i] = a[i-1] + a[i-2];
}
```

Loop-unrolling:

```
for (j=2; j<10; j=j+2)
{
   ______________
   ______________
   ______________
   ______________
   ______________
   ______________
   ______________
}
```

Solution:

```
for (j=2; j<10; j=j+2)
{
   a[i] = a[i-1] + 1[i-2];
   a[i+1] = a[i] + a[i-1];
}
```