NATIONAL UNIVERSITY OF SINGAPORE

SCHOOL OF COMPUTING

TERM TEST #2
AY2013/14 Semester 2

CS2100 — COMPUTER ORGANISATION

12 April 2014 Time Allowed: **1 hour 15 minutes**

INSTRUCTIONS

1. This question paper contains **NINE(9)** questions and comprises **EIGHT(8)** printed pages.
2. The last two pages are for your reference and rough work. The full "datapath and control" diagram is provided.
3. An **Answer Sheet**, comprising **TWO(2)** printed page, is provided for you.
4. Answer **ALL** questions within the space provided on the **Answer Sheet**.
5. Maximum score is **30 marks**.
6. This is a **CLOSED BOOK** test. However, a handwritten single-sheet double-sided A4 cheat sheet is allowed.
7. Write legibly with a pen or pencil (at least 2B).
8. Calculators and computing devices such as laptops and PDAs are not allowed.
9. Submit only the **Answer Sheet** at the end of the test. You may keep the question paper.
10. Write your **MATRICULATION NUMBER** on the **Answer Sheet** using a **PEN**.

——— END OF INSTRUCTIONS ———

SECTION A (5 Multiple Choice Questions: 10 marks)

Each question has only one correct answer. Write your answer in the space provided on the **Answer Sheet**. Two marks are awarded for each correct answer and no penalty for wrong answer.

1. Given the following **Cycle Per Instruction** for the following instruction types:

Integer	**1 cycle**
Floating Point	**3 cycles**
Memory	**4 cycles**
Branch	**2 cycles**

Which of the following programs has the performance of **500 million instructions per second** on a processor with **1GHz** clock frequency?

A. Program A with 100% Integer instructions.
B. Program B with 50% Integer and 50% Memory instructions.
C. Program C with 25% Floating Point, 25% Branch and 50% Integer instructions.
D. Program D with 25% Memory, 25% Branch and 50% Integer instructions.
E. None of the above

2. Given an **unknown MIPS I-format instruction** and its **encoding:**

XXX $5, $9, 64 encoded as **0x 21 25 00 40**

What is the encoding for the same instruction **XXX $6, $9, 65?**

A. `0x21 26 00 41`
B. `0x21 45 00 41`
C. `0x22 25 00 40`
D. `0x21 46 00 41`
E. None of the above

3. Mr.Noob is pondering about this strange line of MIPS instruction in his program:

```
beq $t3, $t9, 0   #0 is the immediate value
```

Which of the following statements is **TRUE**?

A. That instruction is an "infinite loop".
B. That instruction can be removed from the program with NO impact on execution result.
C. That instruction is equivalent to a branch-lesser-or-equal (**`ble`**).
D. That instruction jumps to the instruction at instruction address 0.
E. None of the above.

4. Suppose instruction types **I**, **J**, **K** and **L** takes up 40%, 30%, 20% and 10% respectively in the total execution time for a program. Which of the following gives the CORRECT conclusion regarding overall speedup?

	Option X	Option Y	Conclusion
A.	Speedup **I** by 4 times.	Speedup **J** by 3 times.	Both options give the same overall speedup.
B.	Speedup **I** by 2 times.	Speedup **L** by 10 times.	Option Y gives better overall speedup.
C.	Speedup **J** by 2 times.	Speedup both **K** and **L** by 2 times.	Option X gives better overall speedup.
D.	Speedup both **J** and **L** by 4 times.	Speedup **I** by 4 times.	Both options give the same overall speedup.

E. None of the above

5. Consider the following jump instruction and the two possible jump targets **T1** and **T2**. The two jump targets are ten instructions away from the jump:

```
T1: <instruction -10>
    … … … … … … … …
    <instruction -2>
    <instruction -1>
    j   #jump instruction
    <instruction 1>
    <instruction 2>
    … … … … … … … …
T2: <instruction 10>
```

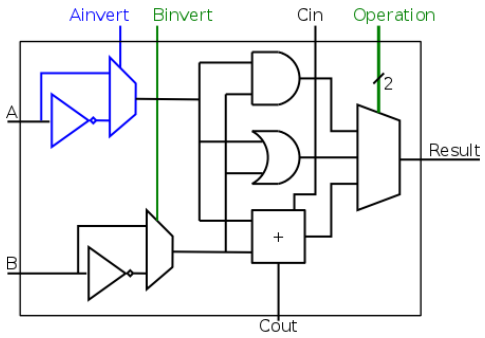
Suppose jump has the instruction address **0xA0 00 00 10**, which of the following is the correct encoding for "**j T1**" and "**j T2**"? The opcode for instruction "**j**" is $\mathbf{2_{16}}$.

	j T1	**j T2**
A.	**0x08 00 00 00**	**0x08 00 00 20**
B.	**0x08 00 00 00**	**0x08 00 00 08**
C.	**0x0B FF FF FA**	**0x08 00 00 0C**
D.	**0x0B FF FF F0**	**0x08 00 00 0C**
E.	None of the above.	

SECTION B (Bonus Question: 1 mark)

This is a bonus question. The mark of this question will be added only if the total mark scored is less than 30.

6. What is the "magic" control signal we can use on the following 1-bit ALU to get "**A `xor` B**"?



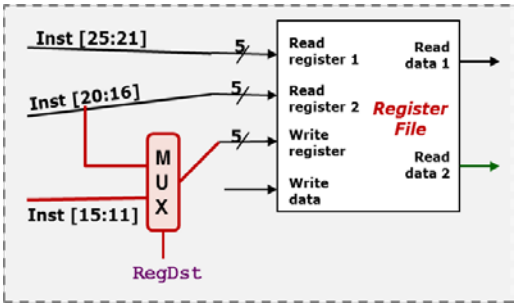
Note the order for the 4 bits control signal is {Ainvert, Binvert, Operation}. Multiplexer ports are labeled from top to bottom, starting at 0 at the top port.

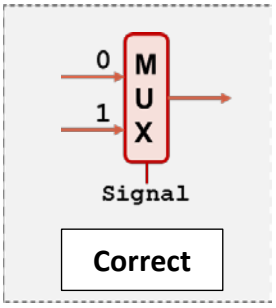
A. 1101
B. 0101
C. 1001
D. 0010
E. None of the above.

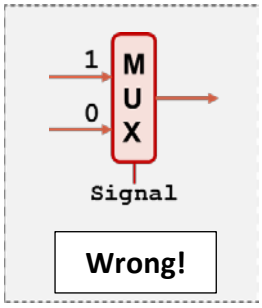
SECTION C (Questions: 20 marks)

Write your answer in the space provided on the **Answer Sheet**. You do not need to show workings, unless otherwise stated.

7. Mr. De Blunder made a ***huge*** mistake while making his own non-pipelined MIPS processor. He accidentally **swapped the two input ports for the Multiplexer(RegDst):**







For each of the following instructions, give:

i. One example where the incorrect processor still gives the **right execution result.**

ii. One example where the incorrect processor gives the **wrong execution result**.

If there is no suitable answer, please circle "**no answer**" in the answer sheet. **[6 marks]**

a) **`add`** (Addition)

b) **`lw`** (Load Word)

c) **`beq`** (branch-if-equal) , provide the branch offset as immediate value.

Note: The full datapath diagram is provided at the end of this paper if you need it.

8. Consider the following MIPS code and the associated memory content:

```
#$s0 is initialized to 200
#$s1 is initialized to 3
#$s2 is initialized to 0

      add  $t2, $s0, $zero
      lw   $t0, 0($t2)
      lw   $t1, 4($t2)
loop: addi $s2, $s2, 1
      beq  $s1, $t1, end
      beq  $t0, $zero, end
      sll  $t2, $t0, 3
      add  $t2, $s0, $t2
      lw   $t0, 0($t2)
      lw   $t1, 4($t2)
      j loop
end:
```

Address	Content
192	4
196	1
200	3
204	1
208	2
212	2
216	0
220	3
224	1
228	0
232	3
236	0
240	4
244	1

Give the **final contents of registers `$s2`, `$t0`, `$t1` and `$t2`** when the execution reaches the "end" label. **[1 mark each for `$s2`, `$t0`; 2 marks each for `$t1`, `$t2`; total = 6 marks]**

9. Let us focus on the pipeline latches ID/EXE and EXE/MEM in the 5-stage pipelined MIPS processor. The MIPS processor is equipped with **full data forwarding paths for handling RAW data hazards.**

   Assume the following for the initial content for registers and memory:

   - Registers have the same content as the register number, i.e. [$0] = 0, [$1] = 1, …, [$31] = 31.
   - Memory locations have the same content as the memory address, i.e. Memory location at address 1000 has the value 1000 etc.

   Given the following MIPS program:

```
add $3, $4, $12
add $1, $3, $2
lw  $5, 4($3)
add $2, $3, $5
```

   Give the following values for the pipeline latches from **cycle 4 to cycle 7**:

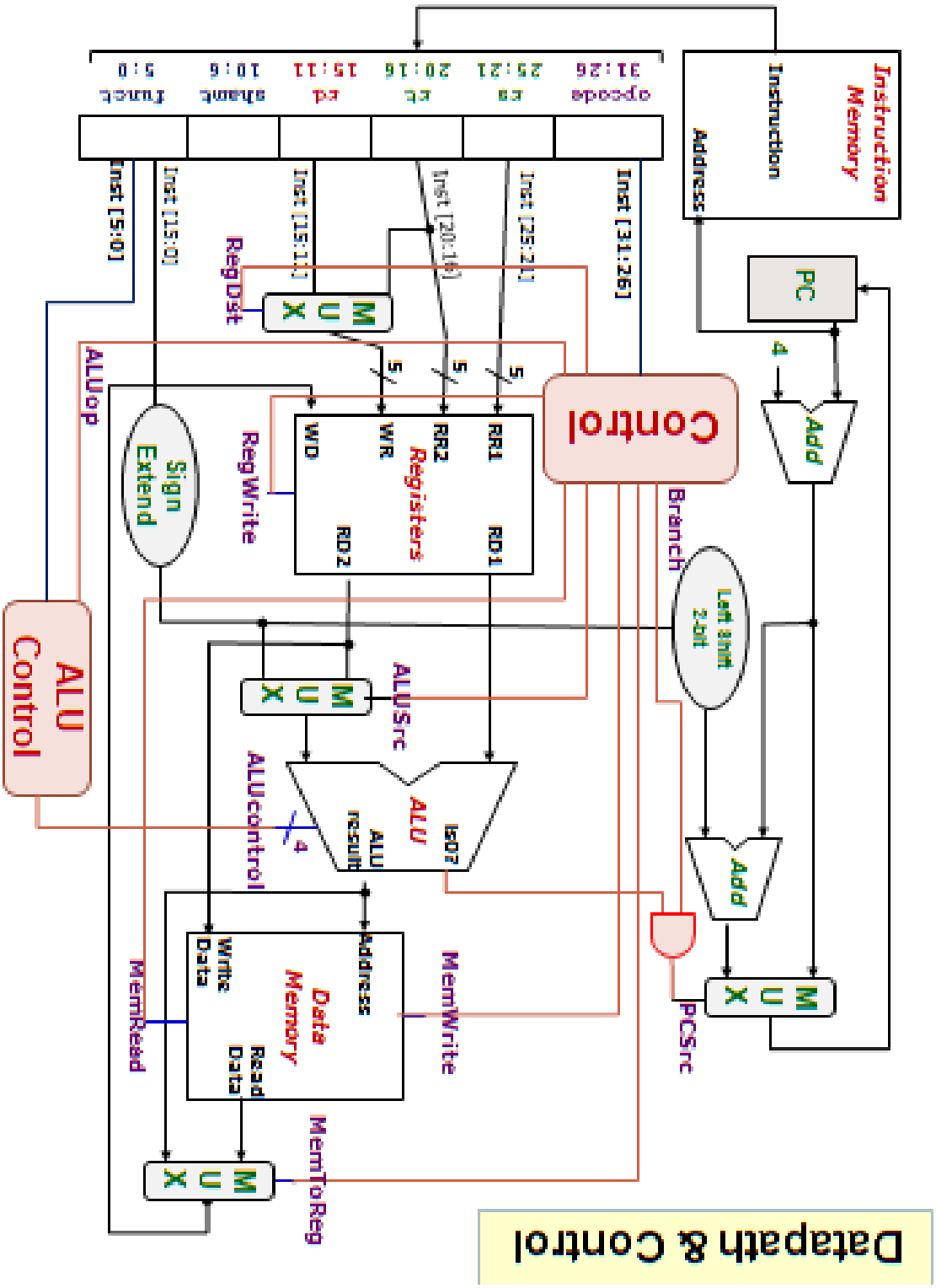
   - ID/EXE: ***Read Data 1*** and ***Read Data 2*** (i.e. the two read registers)
   - Exe/Mem: ***ALU Result***

   If needed, put a "Stall" for stalled pipeline latch(es), put a "X" for irrelevant values. [8 marks]

   Note that the space for pipeline latch MEM/WB is provided only to help your tracing efforts. The MEM/WB latch content **does not carry any weightage and will not be marked.**

Cycle 3 (Given as an example)		
ID/EXE	**EXE/MEM**	**MEM/WB**
Read Data 1: 4	**ALU Result:** X	**Memory Result:** X
Read Data 2: 12		**ALU Result:** X

——— END OF PAPER ———


Instruction Memory
Address
Instruction
PC
4
Add
Control
Inst [31:26]
Inst [25:21]
Inst [20:16]
Inst [15:11]
Inst [15:0]
Inst [5:0]
opcode
rs
rt
rd
shamt
funct
31:26
25:21
20:16
15:11
10:6
5:0
RegDst
M U X
5
WD
WR
RR2
RR1
RD2
RD1
Registers
RegWrite
Sign Extend
ALUop
ALU Control
Branch
Left shift 2-bit
ALUSrc
ALU
ALU result
Zero?
ALUcontrol
4
Add
PCSrc
MemRead
MemWrite
Write Data
Address
Data Memory
Read Data
MemToReg
Datapath & Control

This page is for your rough work.