

**NATIONAL UNIVERSITY OF SINGAPORE**

**CS2100 – COMPUTER ORGANISATION**

(Semester 1: AY2015/16)

Time Allowed: 2 Hours

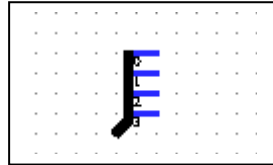
---

**INSTRUCTIONS TO CANDIDATES**

1. Please write your **Student Number** on odd-numbered pages of the **ANSWER BOOKLET** provided. Do not write your name.
2. This assessment paper consists of **TWELVE (12)** questions and comprises **TEN (10)** printed pages.
3. Answer all questions and write your answers in the **ANSWER BOOKLET** provided.
4. This is a **CLOSED BOOK** examination. One handwritten A4 reference sheet is allowed.
5. Calculators are allowed.
6. You may use pencil to write your answers.
7. A partial MIPS Reference Data Sheet is given in Appendix A on page 7.
8. The last three pages are for your rough work. They contain blank K-maps, state table, truth table and timing charts for your use.
9. You are to submit only the **ANSWER BOOKLET** and no other document.

**Questions 1 - 6:** Each question has only one correct answer. Write your answers in the boxes provided in the Answer Booklet. One mark is awarded for a correct answer and no penalty for wrong answer.

1. When you need multi-bit values in Logisim, you use this tool as shown in the picture below. What is this tool called?



- A. Splitter
- B. Zipper
- C. Comb
- D. Brush
- E. There isn't such a thing.

A

2. Which of the following is a **pseudo instruction** in MIPS?

- A. **sub** (subtract)
- B. **slti** (set less than immediate)
- C. **bgt** (branch on greater than)
- D. **xor** (bitwise exclusive-OR)
- E. None of the above.

C

3. A program containing 5000 instructions is run on a machine with a clock frequency of 2 GHz. The table below shows the number of cycles for each instruction class and their frequencies in the program.

Instruction class	A	B	C	D
CPI	3	5	4	6
Frequency	40%	20%	20%	20%

How long does the program take to run on this machine?

- A. 2.38 microseconds
- B. 10.5 microseconds
- C. 11.25 microseconds
- D. 21.0 microseconds
- E. 42.0 microseconds

21000 cycles

B

4. Consider a **4-way set associative cache** with a full data capacity of 8192 bytes. Each cache block consists of 4 words, and each word is 4 bytes long. What are the number of bits in the set-index field and the number of bits in the offset field of the memory address?

- A. Set-index = 4 bits; Offset = 2 bits
- B. Set-index = 6 bits; Offset = 2 bits
- C. Set-index = 6 bits; Offset = 4 bits
- D. Set-index = 7 bits; Offset = 4 bits
- E. Set-index = 8 bits; Offset = 4 bits

offset 4 bits  
1 block  $2^4$  bytes  
1 set  $2^6$  bytes

C

5. The five stages of a certain pipeline take 2 ns, 3 ns, 4 ns, 5 ns, and 2 ns. If there are 20 instructions, what is the maximum speedup in the execution time of a pipeline implementation compared to a single-cycle implementation?

- A. 2.50
- B. 2.67
- C. 2.88
- D. 3.20
- E. 5.00

single cycle 20 cycles of 16ns  
pipeline 24 cycles of 5ns

B

6. A certain machine has 3 types of instructions: *A*, *B* and *C*. Type-*A* instructions have opcode of 4 bits, type-*B* 6 bits, and type-*C* 8 bits. Assuming that each type must have at least one instruction, and the encoding space for opcode is completely utilized, what is the maximum number of type-*C* instructions you can have using an expanding opcode scheme?

- A. 220
- B. 227
- C. 236
- D. 254
- E. 256

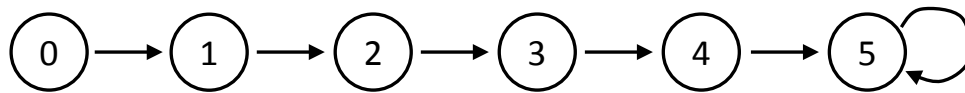
$(16-1)*4-1)*4$

236

256-16-4

## 7. [8 marks]

A sequential circuit goes through the following states, whose state values are shown in decimal, as shown below:



The states are represented by 3-bit values  $ABC$ . Implement the sequential circuit using a  $JK$  flip-flop for  $A$ , a  $T$  flip-flop for  $B$ , and a  $D$  flip-flop for  $C$ .

- Write out the **simplified SOP expressions** for all the flip-flop inputs. Note that the simplified expression for  $KA$  has been done for you ( $KA = 0$ ). [3 marks]
- Complete the logic diagram on the answer booklet, by adding one inverter and a minimum number of logic gates of another type. [2 marks]
- Complete the given state diagram on the answer booklet, by indicating the next state for each of the two unused states. [2 marks]
- Is the circuit self-correcting? Explain your answer. (No mark will be awarded if there is no explanation or the explanation is wrong.) [1 mark]

## 8. [5 marks]

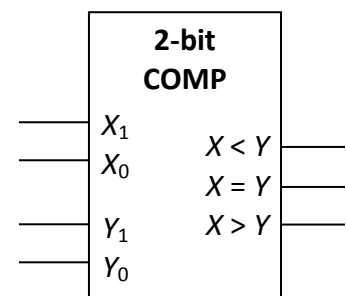
Given the following Boolean function:

$$F(A,B,C,D) = \sum m(5, 6, 9, 10)$$

You are to implement  $F$  using at most two 2-bit magnitude comparators and one two-input logic gate. Note that complemented literals are not available.

No marks will be given if the above conditions are not met.

The block diagram of a 2-bit magnitude comparator is shown on the right.  $X = X_1X_0$  and  $Y = Y_1Y_0$  are unsigned binary values.

9. [6 marks] Answer the following parts about the **addi** (add immediate) instruction.

- What are the values of the control signals **RegDst** and **ALUSrc** for **addi**? [2 marks]

- Given that **\$s1** contains the value 4, **\$t1** contains the value 8, and the data in some memory are shown in the table on the right.

The **addi** instruction below is to be executed. Suppose due to some hardware fault, the value 1 is erroneously generated for the control signal **MemtoReg**. What is the final value in **\$s1** after the **addi** instruction is executed? Explain clearly. [4 marks]

**addi \$s1, \$t1, 12**

Address	Data
0	57
4	-43
8	100
12	3
16	98
20	62
24	-31

## 10. [10 marks]

Study the partial MIPS program below.

The input **\$a1** contains 30 bits of data padded with two zeroes at the end (right-most two bits). The 30 bits are data collected in a half-hour period. Each data bit indicates whether the light is *off* (0) or *on* (1) during a period of one minute. The state of the light (*off* or *on*) may only change at the beginning of a one-minute period. For instance, if **\$a1** contains the following data (only the first 8 bits are shown) 01110010... it means that the light is *off* in the first minute, *on* in the next three minutes, *off* in the next two minutes, *on* in the next minute, and so on.

The partial MIPS program below computes the number of times the light changed from *off* to *on* in that 30-minute period, that is, the number of times "01" appears in **\$a1**.

- (a) Write the instruction encoding in hexadecimal for the **add \$a1, \$v0, \$0** and **srl \$a1, \$a1, 2** instructions. [2 marks]

- (b) Complete the program on the Answer Booklet using not more than 10 MIPS instructions. You are NOT to change or add any instruction before the "Loop" label. [8 marks]

```
# register $a1 contains a 32-bit value to be read
# register $a2 is the answer: the number of "01" in $a1

main: li    $v0, 5          # code 5: read_int call
      syscall              # syscall to read int
      add   $a1, $v0, $0    # transfer int read into $a1

      srl   $a1, $a1, 2     # shift right $a1 by 2 bits
      andi  $t1, $a1, 1     # extract last bit of $a1 to $t1

      add   $a2, $0, $0     # initialise the answer to 0
      addi  $t9, $0, 30     # initialise loop counter to 30

Loop:
```

## 11. [7 marks]

In the MIPS code below, register **\$a0** stores the starting address of an integer array **A**. Assume a 5-stage MIPS pipeline processor.

	<b>addi</b>	<b>\$t1, \$a0, 12</b>	<b># I1</b>
	<b>lw</b>	<b>\$t0, 12(\$a0)</b>	<b># I2</b>
<b>Loop:</b>	<b>lw</b>	<b>\$t2, -4(\$t1)</b>	<b># I3</b>
	<b>sw</b>	<b>\$t2, 0(\$t1)</b>	<b># I4</b>
	<b>addi</b>	<b>\$t1, \$t1, -4</b>	<b># I5</b>
	<b>bne</b>	<b>\$t1, \$a0, Loop</b>	<b># I6</b>
	<b>sw</b>	<b>\$t0, 0(\$t1)</b>	<b># I7</b>

- (a) What does the code do? [2 marks]
- (b) If the first instruction executed is instruction **I1**, which is the seventh instruction executed? [1 mark]
- (c) Fill in the timing chart assuming no data forwarding, and branch resolution is at stage 4. You need to fill in only the first 7 instructions executed. Also, calculate the total number of cycles taken by the code after all iterations. [2 marks]
- (d) Fill in the timing chart assuming data forwarding, branch resolution is shifted to stage 2 and branch prediction is used with the assumption that the branch is to be taken. You need to fill in only the first 7 instructions executed. Also, calculate the total number of cycles taken by the code after all iterations. [2 marks]

## 12. [8 marks]

Study the code below. The code accesses 3 integer arrays **A**, **B**, and **C**, each with 32 elements. Each element takes up 32 bits, which is one word in the MIPS architecture.

```
int sum = 0;
for (int i=0; i<32; i++) {
    sum = sum + (A[i] * B[i]) - C[i];
}
```

The starting addresses of the arrays are shown below:

- Array **A**: starting address at 0x00000080
- Array **B**: starting address at 0xFFFF0040
- Array **C**: starting address at 0x12345688

Given a **direct-mapped cache** with 8 blocks, each block containing 4 words.

- (a) What is the cache hit rate of the above code? You may write your answer as a fraction. [2 marks]
- (b) Fill in the content of the cache after executing the above code. [6 marks]

For instance, if you think that block 0 of the cache contains **A[12]**, **A[13]**, **B[24]** and **B[25]**, you may fill in the cache this way:

Block 0	A[12]	A[13]	B[24]	B[25]
---------	-------	-------	-------	-------

**MIPS Reference Data (partial)**

Name	Mnemonic	Format	Operation	Opcode/Funct
Add	add	R	$R[rd] = R[rs] + R[rt]$	0/20 <sub>hex</sub>
Add Immediate	addi	I	$R[rt] = R[rs] + \text{SignExtImm}$	8 <sub>hex</sub>
And	and	R	$R[rd] = R[rs] \& R[rt]$	0/24 <sub>hex</sub>
Branch On Equal	beq	I	If $(R[rs] == R[rt])$ $PC = PC + 4 + \text{BranchAddr}$	4 <sub>hex</sub>
Load Word	lw	I	$R[rt] = M[R[rs] + \text{SignExtImm}]$	23 <sub>hex</sub>
Or	or	R	$R[rd] = R[rs]   R[rt]$	0/25 <sub>hex</sub>
Set Less Than	slt	R	$R[rd] = (R[rs] < R[rt]) ? 1 : 0$	0/2A <sub>hex</sub>
Shift Right Logical	srl	R	$R[rd] = R[rt] \gg \text{shamt}$	0/02 <sub>hex</sub>
Store Word	sw	I	$M[R[rs] + \text{SignExtImm}] = R[rt]$	2B <sub>hex</sub>
Subtract	sub	R	$R[rd] = R[rs] - R[rt]$	0/22 <sub>hex</sub>

**BASIC INSTRUCTION FORMATS**

R	opcode		rs	rt	rd	shamt	funct					
	31	26	25	21	20	16	15	11	10	6	5	0
I	opcode		rs	rt	immediate							
	31	26	25	21	20	16	15	0				
J	opcode		address									

**REGISTER NAMES AND NUMBERS**

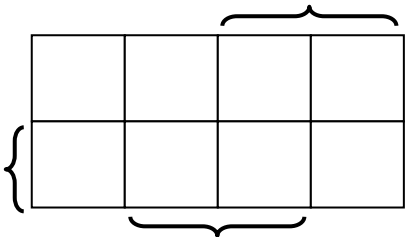
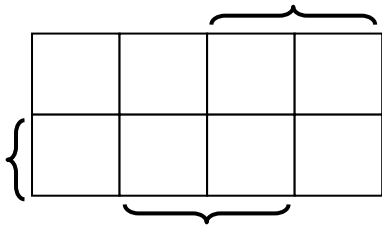
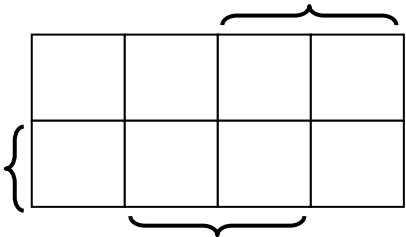
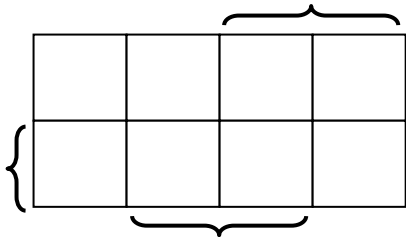
Name	Number	Use
\$zero	0	The constant value 0
\$at	1	Assembler Temporary
\$v0 – \$v1	2 – 3	Values for Function Results and Expression Evaluation
\$a0 – \$a3	4 – 7	Arguments
\$t0 – \$t7	8 – 15	Temporaries
\$s0 – \$s7	16 – 23	Saved Temporaries
\$t8 – \$t9	24 – 25	Temporaries
\$k0 – \$k1	26 – 27	Reserved for OS Kernel
\$gp	28	Global Pointer
\$sp	29	Stack Pointer
\$fp	30	Frame Pointer
\$ra	31	Return Address

~~ END OF PAPER ~~

(Blank K-maps, state table, truth table and timing charts are provided in the next two pages.)



This page is for your rough work.



A	B	C							
0	0	0							
0	0	1							
0	1	0							
0	1	1							
1	0	0							
1	0	1							
1	1	0							
1	1	1							

A	B	C	D	
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
<b>I1</b>																					
<b>I2</b>																					
<b>I3</b>																					
<b>I4</b>																					
<b>I5</b>																					
<b>I6</b>																					

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
<b>I1</b>																					
<b>I2</b>																					
<b>I3</b>																					
<b>I4</b>																					
<b>I5</b>																					
<b>I6</b>																					