

EXAMINATIONS – 2016

TRIMESTER 2

NWEN242

Computer Organisation

Time allowed: TWO HOURS

CLOSED BOOK

Permitted materials: Only silent non-programmable calculators or silent programmable calculators with their memories cleared are permitted in this examination.

Paper foreign to English language dictionaries are allowed.

Instructions: Answer all questions.

This exam consists of 100 marks in total.

There are FIVE questions, each worth 20 marks.

Question 1: Basic Concepts

Question 2: MIPS Assembly Language

Question 3: Combinational and Sequential Logic

Question 4: Pipelined Data Path

Question 5: Cache and Memory

THIS PAGE IS INTENTIONALLY LEFT BLANK

Question 1: Basic Concepts

[20 MARKS]

- a) [2 marks] List four of the five major components of a computer.
- b) [2 marks] What is the clock period? What is the clock rate?
- c) [2 marks] MIPS processors are considered RISC processors. Explain the difference between RISC and CISC processors.
- d) [2 marks] Convert the 8 bit number 0000 0110 to a negative two's complement number and then reverse it again. Show your working.
- e) [2 marks] The register file is considered very fast memory. Why is it not used for all memory purposes? Justify your answer.
- f) [5 marks] Provide the formats of R-type and I-type instructions. Explain how **opcode** is used. Specify the source and destination registers for R-type instructions.
- g) [5 marks] Explain what are direct-mapped, set-associative and full-associative caches.

- a) [10 marks] Write a MIPS program to calculate the absolute value of two numbers.

$$C = |A - B|$$

A is at memory address: 0001 0010 0011 0100 0101 0000 0000 0000 _{base2}

B is at memory address: 0001 0010 0011 0100 0101 0000 0000 0100 _{base2}

C is at memory address: 0001 0010 0011 0100 0101 0000 0000 1000 _{base2}

Feel free to refer to the MIPS aide memoir provided below, you may use other MIPS instructions as well, as this list is not exhaustive.

MIPS aide memoir

add \$xx, \$xx, \$xx	Add
sub \$xx, \$xx, \$xx	Subtract
addi \$xx, \$xx, n	Add immediate
subi \$xx, \$xx, n	Subtract immediate
and \$xx, \$xx, \$xx	And
or \$xx, \$xx, \$xx	Or
sll \$xx, \$xx, n	Shift left logical
srl \$xx, \$xx, n	Shift right logical
slt \$xx, \$xx, \$xx	Set less than
sltu \$xx, \$xx, \$xx	Set less than, unsigned
j n	Jump
jal n	Jump and link
jr \$xx	Jump register
beq \$xx, \$xx, n	Branch if equal
lui \$xx, n	Load upper immediate
lb \$xx, n(\$xx)	Load byte
lw \$xx, n(\$xx)	Load word
sb \$xx, n(\$xx)	Store byte
sw \$xx, n(\$xx)	Store byte
ll \$xx, n(\$xx)	Load linked
sc \$xx, n(\$xx)	Store conditional
bgt \$xx, \$xx, n	Pseudo instruction, Branch greater than
bge \$xx, \$xx, n	Pseudo instruction, Branch greater than or equal to

- b) [5 marks] The function below increments the value held at \$a0 by 10

```
myfunc:    lw $s0, 0($a0)
           add $s0, $s0, 10
```

If this represents the desired task, briefly outline the missing steps needed to fulfil this function call and allow it to be used from elsewhere in the code. If you are referring to registers, name them. If there are any pre-conditions, state them.

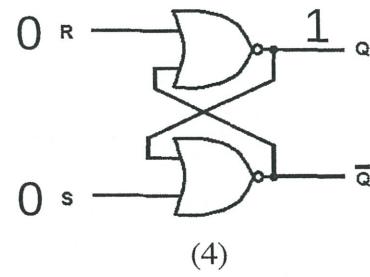
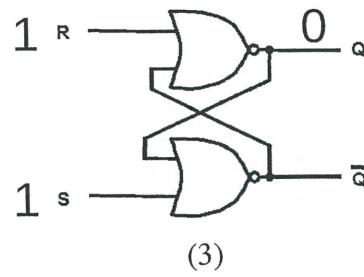
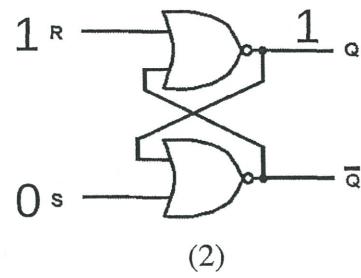
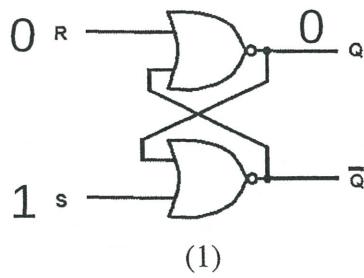
- c) [5 marks] Briefly explain the use of the following two commands **ll** and **sc**. Illustrate with a code sample that demonstrates your explanation.

ll	\$xx, n(\$xx)	Load linked
sc	\$xx, n(\$xx)	Store conditional

a) The SR-latch forms the basis of fast register memory.

(i) [1 mark] In a sentence or two describe what the SR-latch does.

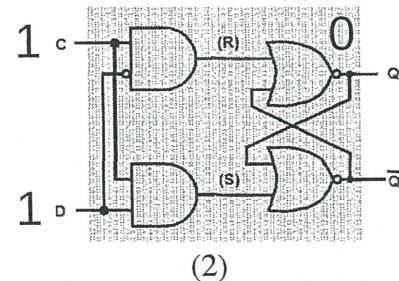
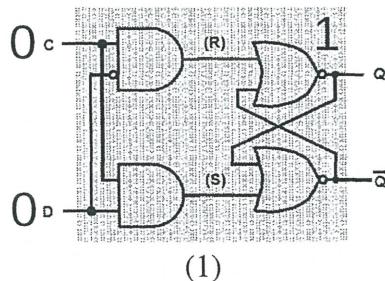
(ii) [4 marks] State the new value of Q after applying the inputs in each of the following four SR-latches and reaching a stable state. If the inputs do not generate a valid Q output, explain why. The existing value of Q is indicated in the figures below.

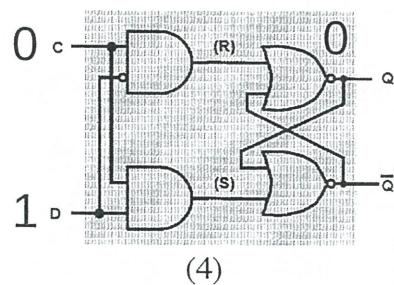
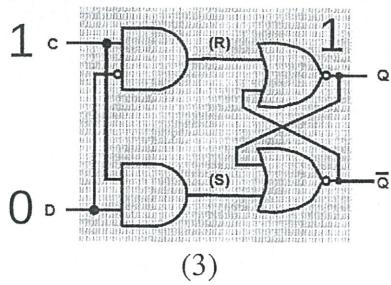


b) The D-latch incorporates the SR-latch.

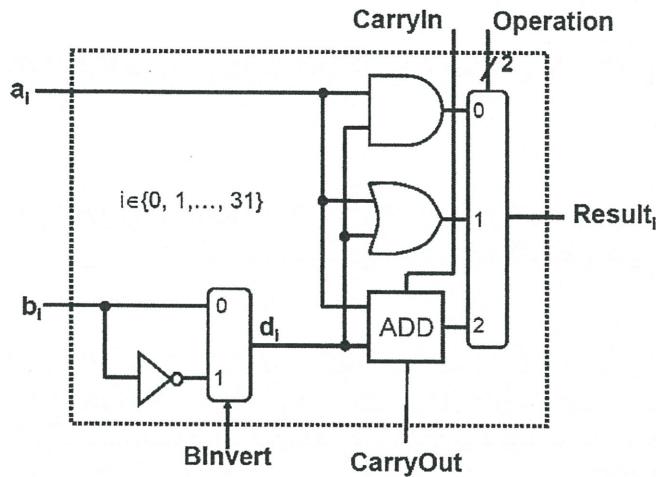
(i) [1 mark] In a sentence or two describe what the D-latch does.

(ii) [4 marks] State the new value of Q after applying the inputs C and D in each of the following four D-latches. If the inputs do not generate a valid Q output, explain why. The existing value of Q is indicated in the figures below.





- c) [5 marks] Below is the circuit for a simple one bit ALU.



For each row 1 to 10 in the table below, provide in your **answer sheet** the two outputs: Carryout and Result, for example,

- 1) x, y
- 2) x, y
- ...
- 10) x, y.

In this example **x** is the Carryout and **y** is the Result.

	a_i	b_i	CarryIn	Binvert	Operation	Carryout	Result
1)	1	0	0	1	00		
2)	1	0	1	0	00		
3)	1	1	1	0	01		
4)	0	1	0	1	01		
5)	0	1	1	1	10		
6)	1	0	0	1	10		
7)	0	1	1	0	10		
8)	1	1	1	0	10		
9)	0	1	0	0	10		
10)	1	0	1	1	10		

- d) [5 marks] Use AND, OR and NOT gates to create a logic block that produces the carry-out of the 1-bit adder. Clearly label the inputs and the output.

Question 4: Pipelined Datapath

[20 MARKS]

Consider the following sequence of instructions.

```

xx: add $3, $1, $2
xx: lw   $4, 8($3)
xx: add $6, $4, $3
xx: beq $6, $7, 5
40: sub $9, $8, $7
...
xx: sw   $t4, 4($t4)

```

- [2 marks] What is the target address if the branch is taken, upon executing the **beq** instruction?
- [4 marks] Assuming that the pipeline has no forwarding support, and branches execute in the EX stage, insert nops to ensure correct execution.
- [8 marks] Assume that the pipeline has full forwarding support and the extra hardware to move the branch decision to the ID stage. Also assume that the **store (sw)** instruction is the target if the branch is taken, and the branch prediction is **Not Taken**. However, the prediction is **incorrect**.

The table below is an **incomplete** multi-cycle view of the pipeline:

Cycle	1	2	3	4	5	6	7	8	9
add \$3, \$1, \$2	ID	EX	MEM	WB					
lw \$4, 8(\$3)									
add \$6, \$4, \$5									
beq \$6, \$7, 5									
sub \$9, \$8, \$7									
sw \$4, 4(\$4)									

In your **answer sheet**, complete the table (identify the stages of the instructions at each of the 9 cycles). You may write your answer in the format below:

add	ID	EX	MEM	WB
lw	xx	xx	xx	xx
add		xx	xx	xx
...				

- [6 marks] Assume that the pipeline has full forwarding and hazard detection support. Explain whether there are hazards between **the first and the second add instructions**, and between **the load (lw) instruction and the second add instruction**. Give the conditions for detecting the hazards (you need to specify the register number and whether it is Rs/Rt/Rd, e.g., \$3/Rs).

Question 5: Cache and Memory

[20 MARKS]

- a) For a direct-mapped cache with a 32-bit address, consider a cache with 256 blocks and a block size of 8 bytes.
- (i) [3 marks] Identify the Tag, Index and Offset bits of a 32-bit address. Show your working.
- (ii) [3 marks] What are the values of the Tag, Index and Offset fields for byte address 4807 (use decimal notation for the values)? Show your working.
- b) Consider an 8-way set-associative cache with 512 blocks and a block size of 8 bytes.
- (i) [3 marks] Determine the total number of Tag bits for the cache.
- (ii) [3 marks] What are the values of the Tag, Index and Offset fields for byte address 3200 (use decimal notation for the values)? Show your working.
- c) [4 marks] In a two-level cache system where the L1 cache access time is **1** cycle, the L2 cache access time is **10** cycles, and the main memory access time is **100** cycles, assume that the L1 cache has a miss rate of **4%**. Determine the required miss rate for the L2 cache so that the system may achieve an effective CPI of **3**.
- d) [4 marks] In a virtual memory system, the page size is 4KB and a PTE (Page Table Entry) is 4 bytes. Assume that a virtual memory address is 32 bits. Determine the size of the page table.
