TE WHARE WĀNANGA O TE ŪPOKO O TE IKA A MĀUI
VICTORIA
UNIVERSITY OF WELLINGTON

EXAMINATIONS — 2010

END OF YEAR

NWEN 242
COMPUTER ORGANIZATION

Time Allowed: 3 Hours (180 minutes)

Instructions: Answer all questions.
Make sure your answers are clear and to the point.
Calculators and paper foreign language dictionaries are allowed.
No reference material is allowed.
There are 180 possible marks on the exam.
Every box with a heavy outline requires an answer.
There is an appendix listing MIPS instructions.

	Topic	Marks
1	Input-Output, Performance, and Multi Core	*45 marks*
2	Converting C into MIPS Assembly	*20 marks*
3	MIPS Addressing Modes	*15 marks*
4	MIPS Assembly Programming	*10 marks*
5	Single Cycle Data Path	*30 marks*
6	Pipelined Data Path	*30 marks*
7	Memory Hierarchy	*30 marks*

Note: Marks are shown for each question as a whole and also for their parts.

StudentId_____

Question 1. Input-Output, Performance, and Multi-Core [45 Marks]

For each of the following statements, say whether it is true or false, 2 marks each question.

a) DMA directly transfers data between an I/O device and memory.

ANSWER

b) A DMA controller does not need to inform the processor when the data transfer has completed.

ANSWER

c) Peripheral devices are considered to be part of the physical address space.

ANSWER

d) Memory buses are a common way of connecting I/O devices to processors driven by handshaking.

ANSWER

e) A simple way to ensure that a user program does not read or write a peripheral device, is to ensure that the user program's virtual memory space is set up so that no page corresponds to physical addresses set aside for specific devices.

ANSWER

f) DMA operations can be triggered at bus cycle boundary while interrupts can only be triggered at instruction boundary.

ANSWER

g) Most DMA controllers have a Status Register. This register supplies information to the CPU.

ANSWER

h) Multi-core processors are MIMD: Different cores execute different threads (Multiple Instructions), operating on different parts of memory (Multiple Data).

ANSWER

i) In a multi-core multiprocessor, each core has its own memory.

ANSWER

j) In a MIPS processor, when one thread is waiting for a floating point operation to complete, another thread can use the integer units.

ANSWER

k) There are two kinds of buses: Synchronous buses are more flexible; asynchronous buses are faster.

ANSWER

l) The operating system plays a major role in handling I/O, acting as an interface between program and hardware.

ANSWER

m) The processor always uses different buses to address memory and I/O devices.

ANSWER

n) In burst transfer, a DMA controller steals a cycle before every CPU instruction.

ANSWER

o) The operating system usually prevents user programs from interacting directly with I/O devices.

ANSWER

For the next five questions, choose ONE correct answer. 3 Marks each question.

p) A disadvantage of DMA is that:

A. Computer system performance is improved by direct transfer of data between memory and I/O devices, bypassing the CPU.
B. CPU is free to perform operations that do not use system buses.
C. In case of burst mode data transfer, the CPU is rendered inactive for relatively long periods of time.
D. None of above.

ANSWER

q) To speed up the execution of a program, you can:

A. Use fewer instructions
B. Use fewer cycles per instruction
C. Use faster clock
D. All of above

ANSWER

r) Synchronous buses:

A. Suffer overhead of handshaking
B. Are proof against clock skew (there is no clock)
C. Are clocked: All devices connected must be run at the same clock rate
D. Make longer buses possible

ANSWER

s) Asynchronous buses:

A. Work with fast and slow devices
B. Are inexpensive and fast
C. Use a simpler protocol than synchronous buses
D. All of above

ANSWER

t) The time required to transmit a signal on a bus in the computer depends on

A. Minimum block size
B. Total delay time of all the devices on the bus
C. Time to get control of the bus
D. All of above

ANSWER

Question 2. Converting C into MIPS Assembly [20 Marks]

a) [10 marks] Rewrite the following program into MIPS Assembly

```
if ( i > j ) A[3] = i + j ;
else A[4] = A[i] – j;
```

ANSWER

b) [10 marks] Rewrite the following program into MIPS Assembly

```
sum = 0;
for (i=1;i<n;i++){
     sum=sum+1;}
```

ANSWER

Question 3. MIPS Addressing Modes [15 Marks]

Briefly explain each of the following MIPS addressing modes, including how the addressing mode works and any limitations it has. For each mode list two commands which could use the addressing mode.

a) Register addressing

b) PC-relative addressing

c) Base addressing

ANSWER

SPARE PAGE FOR EXTRA ANSWERS

Cross out the rough working that you do not want marked.
Specify the question number for work you do want marked.

Question 4. MIPS Assembly Programming[10 Marks]

Give the console outputs of the MIPS program below, for each of the following input sets:
a) [3 Marks] 25.
b) [3 Marks] -7.
c) [4 Marks] 5, and then 1, 2, 4, 5, 9.

The MIPS program uses several instructions that were not covered in lectures. You will need to read the brief reference material provided after the program in order to understand the program. The same reference material is also given in the Appendix to this exam.

```
.data
str1: .asciiz " Please give an integer from 1 to 20: "
errormsg: .asciiz "Out of range (1-20). \n"
nline: .asciiz "\n"                          # new line
.text
error:
li $v0, 4
la $a0, errormsg
syscall       #print error msg
j get
.globl main
main:
addi $s0, $zero,21  #s0=21
get:
li $v0, 4
la $a0, str1
syscall           #print string1
li $v0, 5
syscall           #read int
slt $s1, $v0, $s0
beq $s1, $zero, error
blez $v0, error
move $t0, $v0
add $t1, $0, $0
loop:
addi $t1, $t1, 1
li $v0, 1
move $a0, $t1
syscall           #print int
li $v0, 4
la $a0, nline
syscall           #print nline
slt $t2, $t1, $t0
bnez $t2, loop
li $v0, 10       #exit program
syscall
```

ANSWER

MIPS Reference

```
la register, variable         Load address of variable into
register
li register, value            Load 32-bit value into register
blez  register, target        branch to target if register is less
than or equal to zero
bnez  register, target        branch to target if register is not
equal to zero
move register1, register2     copies contents of register2 to
register 1
syscall             invokes a system call, for example, for reading
or printing to the console.
```

Usage of syscall:

```
Step 1. Load the service number in register $v0.
Step 2. Load argument values, if any, in $a0, $a1, $a2, or $f12 as
specified.
Step 3. Issue the SYSCALL instruction.
Step 4. Retrieve return values, if any, from result registers as
specified.
```

Service numbers and registers

```
*print integer:     Code 1 in $v0: $a0 = value to print
*print  string:     Code 4 in $v0: $a0 = address of null-
terminated string to print
*read integer:      Code 5 in $v0: $v0  willcontain integer read
```

Example: display the value stored in $t0 on the console

```
    li  $v0, 1            # service 1 is print integer
    add $a0, $t0, $zero   # load desired value into argument
register $a0
    syscall
```

SPARE PAGE FOR EXTRA ANSWERS

Cross out the rough working that you do not want marked.
Specify the question number for work you do want marked.

Question 5. Single Cycle Data Path [30 marks]

a) [10 marks] Consider the data_path of a MIPS processor. Describe in **one or two** sentences what is each of the following data_path components used for:

i) [2 marks] Program counter (PC).

ANSWER

ii) [2 marks] Register File.

ANSWER

iii) [2 marks] ALU.

ANSWER

iv) [2 marks] Control Unit.

ANSWER

v) [2 marks] Multiplexer.

ANSWER

b) **[7 marks]** Consider the diagram of the Single Cycle Data Path in Figure 1 on the facing page. On the diagram, highlight only those lines and components that will be used to execute a MIPS **sw** instruction. Trace the lines and circle the components using a coloured pen. Do not pay attention to control signals when answering this question.

c) **[3 marks]** Consider the diagram of the Single Cycle Data Path in Figure 1 on the facing page again. Assume a sw instruction is taken. On the diagram, highlight only those control signals that will be asserted during the execution of the MIPS **sw** instruction. Trace these control signals using a coloured pen.

d) **[10 marks]** Opcode of the MIPS bne (branch **not** equal) instruction is 000101_2. Opcode of the MIPS beq (branch **on** equal) instruction is 000100_2. Opcode of the MIPS j (jump) instruction is 000010_2. Design combinational logic blocks that will act as parts of the Single Cycle Data_Path Control Unit and generate Zero&Branch and Jump control signals in Figure 1.

ANSWER

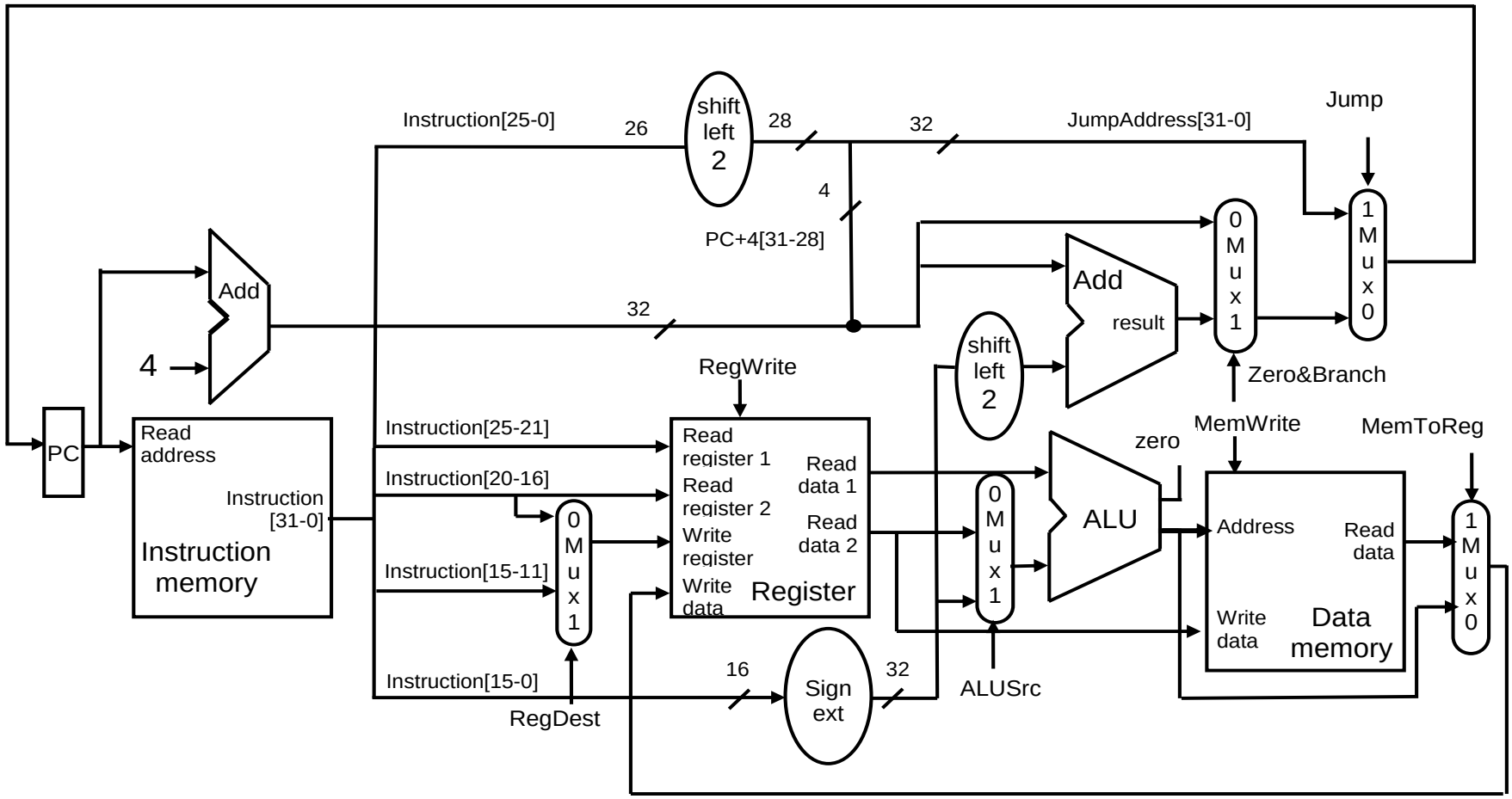
Instruction[25-0]
26
shift left 2
28
32
JumpAddress[31-0]
Jump
4
PC+4[31-28]
Add
4
32
Add
result
0 M u x 1
1 M u x 0
shift left 2
RegWrite
Zero&Branch
PC
Read address
Instruction [31-0]
Instruction memory
Instruction[25-21]
Instruction[20-16]
Instruction[15-11]
Instruction[15-0]
0 M u x 1
Read register 1
Read register 2
Write register
Write data
Read data 1
Read data 2
Register
RegDest
16
Sign ext
32
ALUSrc
zero
MemWrite
MemToReg
ALU
Address
Read data
Write data
Data memory

Figure 1.

StudentId_____

Question 6: Pipelined Data Path [30 marks]

a) [2 marks] What is pipelining and what is the expected benefit of pipelining?

ANSWER

b) [2 marks] What is a data hazard?

ANSWER

c) [2 marks] Suppose there is neither hardware detection nor prevention of hazards. How many nops are needed between any two subsequent instructions being involved in a hazard?

ANSWER

d) [8 marks] Assume you run a MIPS assembly program on a processor whose data_path contains forwarding unit as those that we considered in lectures.

i) [3 marks] Consider the following fragment of a MIPS assembly program.

```
add $s0, $t0, $t1
sw  $t2, 0($s2)
sub $s2, $s0, $t3
```

If you think there should be nops inserted between instructions to avoid hazards, then insert the necessary nops and show the altered program fragment in the answer box below. If you think nops are not needed, just copy the program fragment above in the answer box below.

ANSWER

ii) [5 marks] Consider the data_path with forwarding in Figure 2. below and the fragment of the MIPS assembly program in the answer box above. If you think the forwarding unit still has to undertake some actions to prevent hazards, write instructions of the program fragment in the appropriate pipeline stages in Figure 2. Take a coloured pen and trace only those lines and circle the components in Figure 2 that will be used to perform the needed forwarding.

Note: The data_path in Figure 2 is not complete, but contains all components you may need to answer the question.

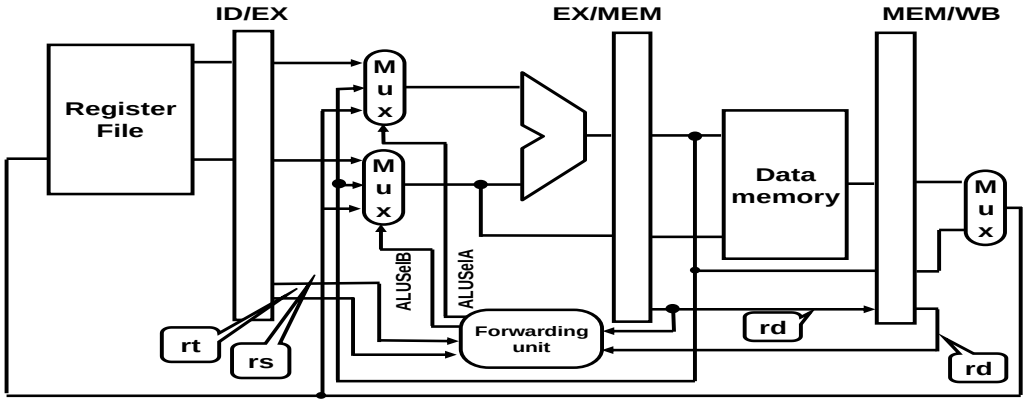

Figure 2.

e) [8 marks] Assume you run a MIPS assembly program on a processor whose data_path contains a forwarding unit as one considered in lectures.

i) [3 marks] Consider the following fragment of a MIPS assembly program.

```
add $s0, $t0, $t1
sw  $t2, 0($s0)
sub $s2, $s2, $t3
```

If you think there should be nops inserted between instructions to avoid hazards, then insert the necessary nops and show the altered program fragment in the answer box below. If you think nops are not needed, just copy the program fragment above in the answer box below.

ANSWER

ii) [5 marks] Consider the data_path with forwarding in Figure 3 below and the fragment of the MIPS assembly program in the answer box above. If you think the forwarding unit still has to undertake some actions to prevent hazards, write instructions of the program fragment in the appropriate pipeline stages in Figure 3, take a coloured pen and trace only those lines and circle the components that will be used to perform the needed forwarding.

Note: The data_path in Figure 3 is not complete, but contains all components you may need to answer the question.

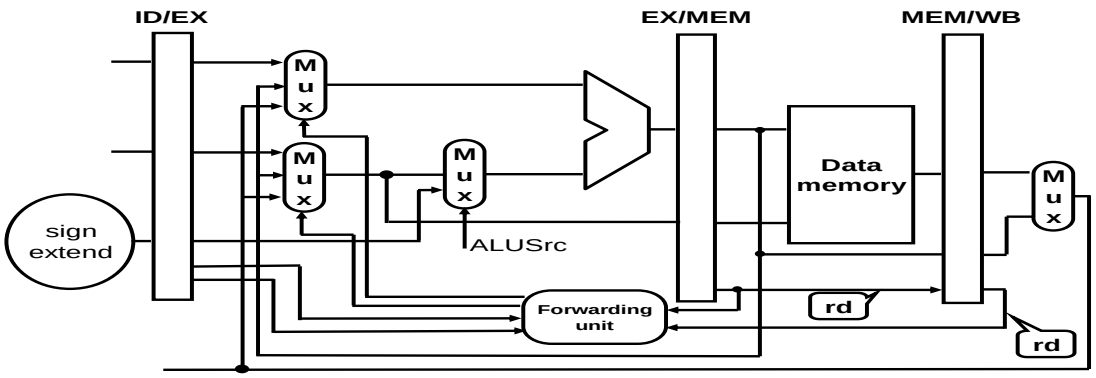

Figure 3.

f) [8 marks] Assume you run a MIPS assembly program on a processor whose data_path contains a forwarding unit as one considered in lectures.

i) [3 marks] Consider the following fragment of a MIPS assembly program.

```
add $s0, $t0, $t1
lw  $t2, 0($s1)
sub $s2, $s1, $t2
```

If you think there should be nops inserted between instructions to avoid hazards, then insert the necessary nops and show the altered program fragment in the answer box below. If you think nops are not needed, just copy the program fragment above in the answer box below.

ANSWER

ii) [5 marks] Consider the data_path with forwarding in Figure 4 below and the fragment of the MIPS assembly program in the answer box above. If you think the forwarding unit still has to undertake some actions to prevent hazards, show instructions of the program fragment in the appropriate pipeline stages in Figure 4, take a coloured pen and trace only those lines and circle the components that will be used to perform the needed forwarding.

Note: The data_path in Figure 4 is not complete, but contains all components you may need to answer the question.

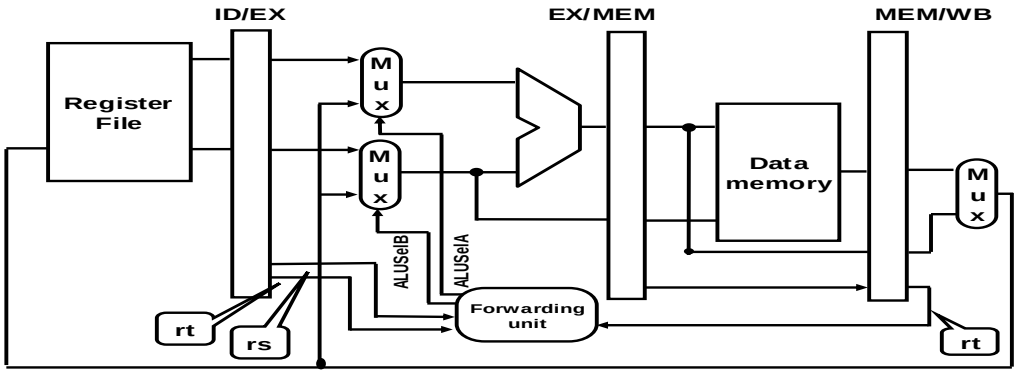

Figure 4.

Question 7. Memory Hierarchy [30 marks]

a) [10 marks] In the table provided for the answer, list the names of the memory hierarchy levels, their typical size (capacity) in bytes or their multiples, a typical time to read or write, and a typical amount of data read or written during one read or right operation in bytes or their multiples.

ANSWER

name	size	r/w time	amount

b) [10 marks] Determining of the word offset length, slot number length, and the tag length are crucial for a correct memory to cache mapping. All these lengths are expressed by the number of memory address bits. Figure 5 below displays a memory address of *a* bits with its tag, slot (or slot set) number, word offset, and byte offset fields.

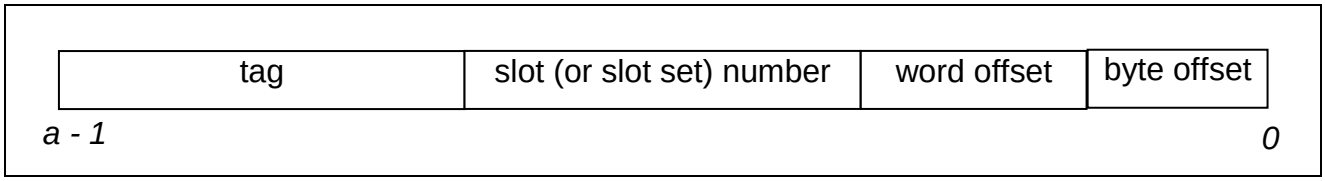

Figure 5.

Let the cache size be $n = 2^s$ slots ($s \geq 0$), the way of cache set associativeness be $m = 2^k$ ($k \geq 0$ and $n \geq m$), and the memory block size be $w = 2^i$ ($i \geq 0$). The length of the byte offset is a constant $b = 2$. Recall, for $m = 1$, the cache is direct mapped, and for $m = n$, the cache is fully associative. Otherwise it is m-way set associative.

i) [5 marks] Express the tag length *t* as a function of *a*, *s*, *k*, *i*, and *b*.

ANSWER

ii) [5 marks] Suppose $a = 32$, $n = 2048$, $m = 8$, and $w = 16$. How much is the tag length t?

ANSWER

c) [10 marks] Consider an 8-way set associative cash having 2048 slots of 16 words per block. Assume the cache is initially empty. The processor issues the following requests:

00000000 11111111 00000000 11000000

00000000 11111111 00000000 11111111

00000000 10101010 01000000 11010101

00000000 10101010 01000000 11101001

i) [8 marks] In the answer box below, show the cache content after satisfying the four processor requests above. In your answer show the content of the set slot number, valid bit, and tag fields of only those slots that are updated when satisfying processor requests.

ANSWER

Slot Set Number	V	Tag	W_0	…	W_{15}

ii) [2 marks] How many hits and misses will produce the processor requests above?

ANSWER

Number of Hits =

Number of Misses =

****