CS 2200 Spring 2011 Test 1

Prism ID:____________

Name:__Kishore__________________ GTID#: 9____________

Problem	Points	Lost	Gained	Running Total	TA
1	1 (1 min)				
2	16 (15 min)				
3	19 (15 min)				
4	15 (10 min)				
5	18 (15 min)				
6	15 (15 min)				
7	16 (10 min)				
Total	100 (81 min)				

- **You may ask for clarification but you are ultimately responsible for the answer you write on the paper.**
- **Illegible answers are wrong answers.**
- **Please look through the entire test before starting. WE MEAN IT!!!**

Good luck!

1. (1 point, 1 min) (select one)

This question pertains to the art work on the cover page of your textbook

- Shows the layout for some weird video game _____
- Anatomical allusion of what is inside the box to what is inside the human body __X____
- Ponytail, bagpipe, walnuts, springs, etc. _____
- A random collection of artifacts from Mayan culture _____
- Duh! This is a human physiology textbook _______

CS 2200 Spring 2011 Test 1

Prism ID:____________

Name:______________________________ GTID#: 9____________

Processor design

2. (16 points, 15 mins)
Given the software convention for programmer visible registers:

a0-a2:	parameter passing
s0-s2:	callee saves if need be
t0-t2:	caller saves if need be
v0:	return value
ra:	return address
at:	target address
sp:	stack pointer

A. (8 points)
Which addressing mode is used for the underlined operand in each of these instructions?

ADD **$a0**, $zero, $s0 Mode: Register

ADDI $a0, $t2, **12** Mode: Immediate

ADDI $a0, **$zero**, 12 Mode: Register

LW **$a0**, 5($a1) Mode: Register

SW $v0, **5($sp)** Mode: Base + offset

JALR $at, **$ra** Mode: Register

JALR **$at**, $ra Mode: Register

BEQ $a0, $a1, **label** Mode: PC-relative

Choose from these modes: **Register, PC-relative, Base + Offset, Immediate**

B. (3 points)
The following is the code to push the return address onto the stack:

```
ADDI  $sp, $sp, -1
SW    $ra,0($sp)
```

Answer the following questions about this machine?

(i) The stack grows toward (circle one of the following)

a) Higher addresses [✓ b) Lower addresses] c) Cannot be determined

(ii) After executing the above two instructions the stack pointer points to the (circle one of the following)

a) first free location on the stack
[✓ b) the last used location on the stack]
c) Cannot be determined

(iii) The machine is (circle one of the following)

a) Big endian b) Little endian [✓ c) Cannot be determined]

C. (5 points)
Inside the **called** procedure, we need a register to store a temporary result. For each of the following registers indicate whether it can be used for this purpose and, if it can be used, what has to be done before it is used.

(i) **$ra**
a) Cannot be used
[✓ b) Can be used, but first we must] Save on stack

(ii) **$s1**
a) Cannot be used
[✓ b) Can be used, but first we must] Save on Stack

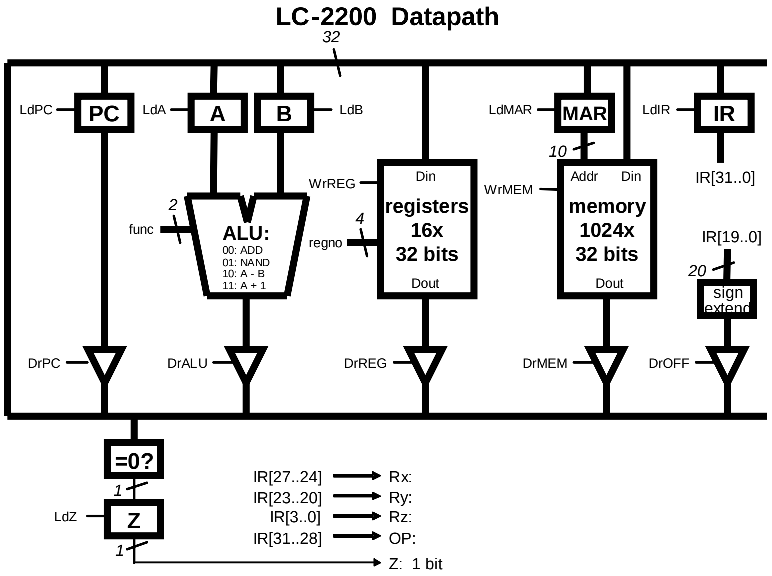
(iii) **$v0**
a) Cannot be used
[✓ b) Can be used, but first we must] do nothing, result register
Caller saves if need be

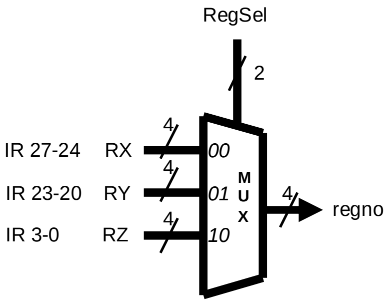
(iv) **$k0**
[✓ a) Cannot be used]
b) Can be used, but first we must ________________________________

(v) **$sp**
[✓ a) Cannot be used]
b) Can be used, but first we must ________________________________

Datapath and control

3. (19 points, 15 min)
You are given below a datapath similar to what we have discussed in class.





We are introducing a new instruction LWI that uses a register content as the address of the address of the operand (i.e., the register content is an "indirect address" to the operand). The syntax and semantics of the LWI instruction is as follows:

```
      LWI Rx, @(Ry);    Rx ← MEM[MEM[Ry]];
```

The instruction format is as shown below:

31 28	27 24	23 20	19 0
OPCODE	Rx	Ry	unused

Write the microcode sequence for the EXECUTE macro state of the LWI instruction (**you don't need to write the fetch sequence for the instruction**). For each microstate, show the datapath action (in register transfer format such as A <- Rx) along with the control signals you need to enable for the datapath action (such as DrREG).

Write your answer on the next page.

(additional workspace for 6)

LWI 1: Ry → MAR (+4)

[7points] Control signals needed:

Reg Sel = 01

DrREG (+1 for each)

Ld MAR

LWI 2: Data from memory → MAR (+4)

[6 points]

Control signals needed:

Dr MEM (+1 for each)

Ld MAR

LWI 3: Data from Memory → Rx (+3)

[6points]

Control signals needed:

DrMEM

Reg Sel=00 (+1 for each)

WrREG

CS 2200 Spring 2011 Test 1

Prism ID:____________

Name:____________________________ GTID#: 9____________

Interrupts

4. (15 points, 10 mins)

A. (5 points)

The following actions happen in the INT macro state of the processor (**Select ALL that apply: +1 for correct choice; -1 for incorrect choice**):

[✓ (1)] Current PC has to be saved in some known place
(2) The registers have to be saved in some known place
[✓ (3)] Interrupts have to be disabled
(4) Interrupts have to be enabled
[✓ (5)] Acknowledge interrupt
[✓ (6)] Retrieve handler address from a known place and place it in PC
(7) Change to user mode
[✓ (8)] Change to kernel mode
(9) Retrieve mode bit from the system stack
(10) Load PC from a general purpose register

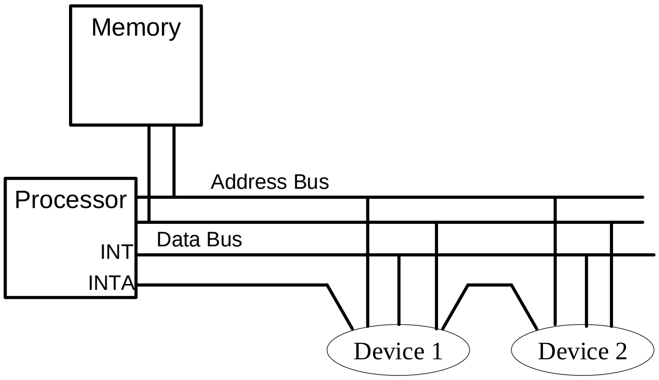
B. (5 points) An interrupt handler differs from a normal procedure in several important ways (fill in the blanks).

(i) First, a normal procedure (i.e., the callee) that modifies registers $s0, $v0, and $t2 must save $S0, while an interrupt handler that modifies the same registers must save All registers.

(ii) Second, the callee can return to the caller using the return address in $ra, while the return address for an interrupt handler is in $k0.

(iii) Third, the starting address for a procedure is specified by the caller. On the other hand, the starting address for an interrupt handler is obtained from an area of memory called the Interrupt Vector Table (IVT)

C. (5 points) This question pertain to the interrupt hardware. Circle the correct choices.



(i) When Device 1 wants to interrupt the processor, it signals its intent by asserting

[✓ a) the INT line]
b) the INTA line
c) the Address Bus
d) the Data Bus

(ii) The processor

a) Aborts
[✓ b) Completes]
c) Saves
d) Restarts

the instruction it is currently executing, and then

(iii) asserts the

a) the INT line
[✓ b) the INTA line]
c) the Address Bus
d) the Data Bus

(iv) When Device 1 sees this, it places a number (called the interrupt vector) on the

a) the INT line
b) the INTA line
c) the Address Bus
[✓ d) the Data Bus]

(iv) The processor gets this number, adds it to the base address of the interrupt vector table, and reads the contents of the memory at this address. The contents of the memory at this location is then

a) placed in the a0 register of the processor
b) placed in the k0 register of the processor
c) placed in the sp register of the processor
[✓ d) placed in the PC register of the processor]

Performance

5. (18 points, 15 mins)

A. (8 points)

A processor with a clock frequency of 1GHZ (1,000,000,000 cycles per second) is running a program that has 2 billion instructions. Twenty percent of all instructions are branches, fifteen percent are loads, and ten percent are stores. A branch takes 6 cycles to execute, a load 8 cycles, a store 7 cycles, and all other instructions take 5 cycles.
What is the execution time of the program?

$$\text{Ex time} = \sum_{i=1}^{N}(n_i * CPI_i) * \text{Cycle time}$$

← +2 for this equation

$$\text{Total clock Cycles} = (0.2*6+0.15*8+0.1*7+0.55*5)*2*10^9$$

+1 for each correct term

$$\text{Ex time} = \text{Total clock cycles} * \text{Cycle time}$$

$$= 11.7*10^9 * 10^{-9} \text{ secs}$$

$$= 11.7 \text{ Secs}$$

+2 for final answer

B. (10 points)

An architect determines that he can reduce branch execution to 4 cycles with no change to the CPIs of the other instructions but with an increase in the clock cycle time of the processor. What is the maximum permissible increase in clock cycle time that will make this architectural change still worthwhile? (This is a continuation of part A)

Total clock cycles of modified machine

$$= (0.2*4+0.15*8+0.1*7+0.55*5)$$

+1 for each correct term

$$= 5.45$$

$$\text{Ex time\_modified} = 5.45*2*10^9 * x$$

+2 for this equation

For change to be worthwhile

$$\text{Ex time\_modified} < \text{Ex time\_original}$$

+2 for this inequality

$$5.45*2*10^9 * x < 11.7$$

$$x < \frac{11.7}{10.9} * 10^{-9} \text{ Secs}$$

Max permissible increase in $x = 0.07*10^{-9}$ secs

+2 for final answer

CS 2200 Spring 2011 Test 1

Prism ID:____________

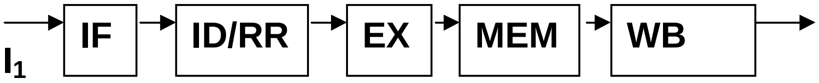
Name:______________________________ GTID#: 9____________

Pipelining

+1 for each correct row 4−10
+2 for each correct arrow

6. (15 points, 10 mins)

I_1: ADD R1,R1,R3
I_2: ADD R4,R4,R2
I_3: LW R5,0(R1)
I_4: LW R6,0(R5)
I_5: ADD R1,R5,R6
I_6: ADD R2,R1,R0



−5 if too many stalls
−5 if there are no stalls

Given the sequence of instructions I_1 through I_6, show the passage of these instructions through the pipeline until all 6 instructions have been completed and retired from the pipeline. We have started you off by showing the state of the pipeline in the first three clock cycles. **Assume that data forwarding is used whenever it is possible. Indicate each use of forwarding by drawing an arrow from the stage that supplies the forwarded value to the stage that it is forwarded to.**

Cycle Number	IF	ID/RR	EX	MEM	WB
1	I_1	-	-	-	-
2	I_2	I_1	-	-	-
3	I_3	I_2	I_1		
4	I_4	I_3	I_2	I_1	
5	I_5	I_4	I_3	I_2	I_1
6	I_5	I_4	NOP	I_3	I_2
7	I_6	I_5	I_4	NOP	I_3
8	I_6	I_5	NOP	I_4	NOP
9		I_6	I_5	NOP	I_4
10			I_6	I_5	NOP
11				I_6	I_5
12					I_6
13					
14					
15					

CS 2200 Spring 2011 Test 1

Prism ID:____________

Name:______________________________ GTID#: 9____________

7. (15 points, 10 mins) (**SHORT ANSWERS WE MEAN IT!!!!**)

A. What is the most important consideration in designing the stages of a pipelined processor?

Ensuring that the amount of work in each stage of the pipe is roughly the same so that no single stage will become a bottleneck to performance.

All or nothing

B. Explain the difference between *latency* and *throughput* in the context of a pipelined processor implementation. What are the metrics used for each?

Latency: The amount of time it takes to execute a single instruction usually expressed as CPI
Throughput: The number of instructions executed per unit time by the processor usually expressed as IPC.

+2 for each

CS 2200 Spring 2011 Test 1

Prism ID:____________

Name:_______________________________ GTID#: 9_____________

```
C. What gives the independence between the stages in the sandwich assembly
   line that is discussed in the book?  Identify the corresponding entities
   in the instruction pipeline that give the same independence.
```

Three things:

1. **No resource contention among the stages (each worker has all he needs to do the work for his stage)**
2. **Order form fully specifies the work that a given worker has to do for that specific sandwich**
3. **The partially assembled sandwich on which he has to do additional work**

+1 for each point

Corresponding entities in the instruction pipeline

1. **Dedicated hardware resources for each stage of the pipeline**
2. **The pipeline buffer serves the role of the order form and the partially assembled sandwich**

+1 for this

```
D. Explain the process by which the size and content of the pipeline
   registers are decided in the instruction pipeline.
```

1. **For each instruction, work out the details of the execution in each stage of the pipeline**
2. **For each instruction, work out what partial state has to be passed from one stage to the next**
3. **The union of the partial states thus compiled for each instruction at a given stage helps us to decide the size and content of the pipeline buffer sitting at its output**

-2 for not being general