CS 2200 Spring 2009 Test 1

Prism ID:_____

Name:_____Kishore_____ GTID#: 9_____

Problem	Points	Lost	Gained	Running Total	TA
1	1				
2	10				
3	9				
4	10				
5	15				
6	20				
7	10				
8	10				
9	15				
Total	100				

- **You may ask for clarification but you are ultimately responsible for the answer you write on the paper.**
- **Illegible answers are wrong answers.**
- **Please do not discuss this test by any means (until 5 pm today)**
- **Please look through the entire test before starting. WE MEAN IT!!!**

Good luck!

1. (1 point, 1 min) (circle one – you get a point regardless of your choice)

Paper or plastic?:

(a) Paper
(b) Plastic
(c) No preference,
(d) I don't care about politics

None of the above! BRING YOUR OWN REUSABLE BAG! ☺

CS 2200 Spring 2009 Test 1

Prism ID:_____

Name:_____Kishore_____ GTID#: 9_____

Processor design

2. (10 points, 10 mins)
Given the software convention for 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
```

Recall that JAL instruction of LC-2200 has the following semantics:

```
      JAL at, ra;       ra <- PC_incremented (return address)
                ;       PC <- at (entry point of procedure)
```

The state of the stack is as shown below. To help you out, we have put down the action corresponding to saving s registers on the stack. Fill out the actions similarly for the other entries on the stack (who is responsible for the action caller/callee, and what is the action).

Your answer:

Stack

Stack Pointer→	Local Variables	**Callee allocates any space needed for local variables**
	Saved s Registers	Callee saves any s registers it plans to use in the procedure
	Prev Return Address	**Caller saves the current return address before executing JAL**
	Add'l Return Values	**Caller allocates space for additional return values beyond v0**
	Add'l parameters	**Caller places additional parameters beyond a0-a2 on the stack**
	Saved t registers	**Caller saves any t registers whose values it needs upon return**

CS 2200 Spring 2009 Test 1

Prism ID:_____

Name:_____Kishore_____ GTID#: 9_____

3. (9 points, 5 mins)
What is the program counter used for? Consider both storing a value into it and saving what is in the PC somewhere else.… Be brief.

Your answer (need three valid reasons for full credit):

1) **Need to know where to return to after a procedure call**

2) **Need to know where to jump to on branches and procedure calls**

3) **Need to know where to resume execution of a program after an interrupt**

4. (10 points, 5 mins)(select the correct choice)
(a) An activation record of a procedure

___**X**___ is usually on the stack

_____ is usually kept in processor registers

_____ is usually kept in a special hardware

_____ is usually allocated in the heap space of the program

_____ is usually allocated in the static (global) data space of the program

_____ all of the above

_____ none of the above

(b) A Frame Pointer

_____ is the same as a stack pointer

___**X**___ is a fixed harness into the activation record for the currently executing procedure

_____ is not a register at all

_____ is implemented in memory

_____ is used for parameter passing during procedure call

_____ all of the above

_____ none of the above

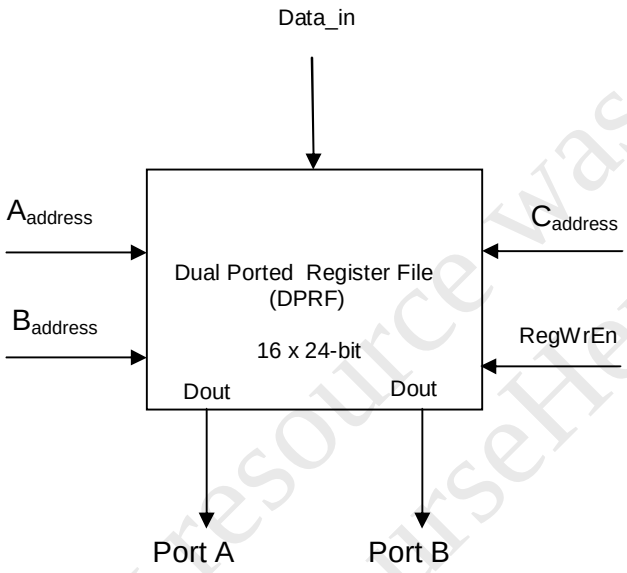
CS 2200 Spring 2009 Test 1

Prism ID:_____

Name:_____Kishore_____ GTID#: 9_____

Datapath elements

5. (15 points, 10 mins)
Shown below is a 16 element dual-ported register file (DPRF). Each register has 24 bits. $A_{address}$ and $B_{address}$ are the register addresses for reading the 24-bit register contents on to Ports A and B, respectively. $C_{address}$ is the register address for writing Data_in into a chosen register in the register file. RegWrEn is the write enable control for writing into the register file.



Answer the following:

(a) Data_in has _____**24**_____ wires

(b) Port A has _____**24**_____ wires

(c) Port B has _____**24**_____ wires

(d) $A_{address}$ has _____**4**_____ wires

(e) $B_{address}$ has _____**4**_____ wires

(f) $C_{address}$ has _____**4**_____ wires

(g) RegWrEn has _____**1**_____ wires

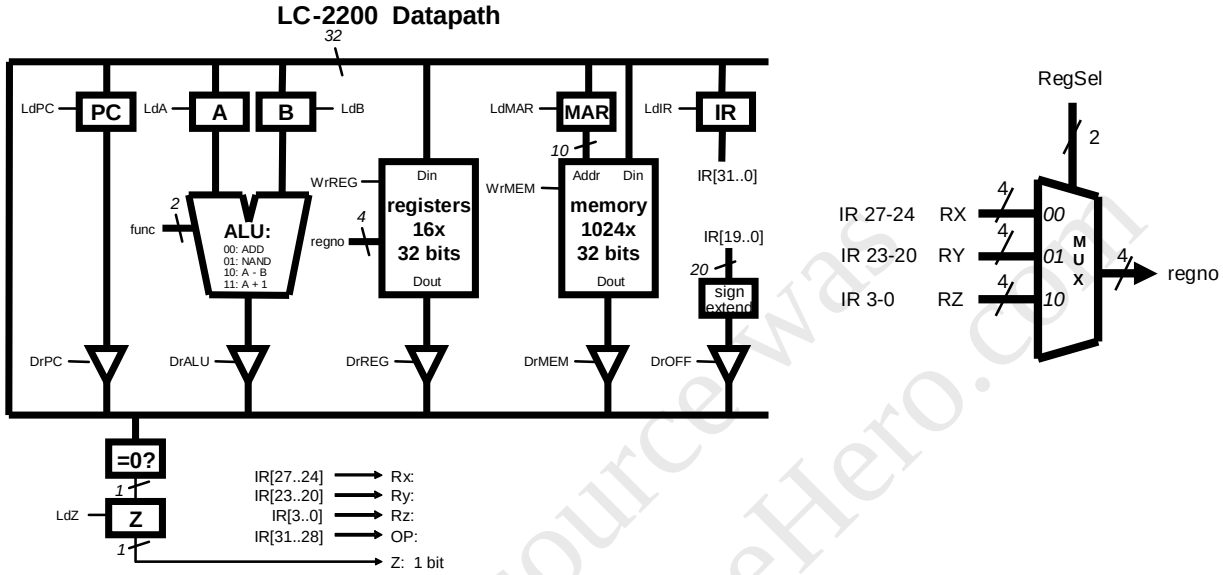
CS 2200 Spring 2009 Test 1

Prism ID:_____

Name:_____Kishore_____ GTID#: 9_____

Control

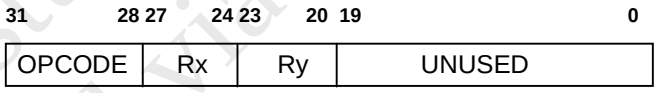
6. (20 points, 15 min)
You are given below a datapath similar to what we have discussed in class.



We have decided to add a complex instruction **POPM** to LC-2200. The semantics of this instruction is as follows:

```
POPM  Rx, (Ry++)  ;    Rx <- MEM[Ry];
                       Ry <- Ry + 1;
```

The instruction format is as shown below:



Write the sequence for implementing the POPM (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.

Prism ID:_____

Name:_____Kishore_____ GTID#: 9_____

6. (Continued)

Popm1:

Ry -> MAR, A
Control signals needed:
RegSel = 01
DrREG
LdMAR
LdA

Popm2:

MEM[MAR] -> Rx
Control signals needed:
DrMEM
RegSel = 00
WrREG

Popm3:

A + 1 -> Ry
Control Signals needed:
func = 11
DrALU
RegSel = 01
WrREG

CS 2200 Spring 2009 Test 1

Prism ID:_____

Name:_____Kishore_____ GTID#: 9_____

Interrupts, exceptions, and traps

7. (10 points, 10 mins)
(Use the following words and phrases exactly once in filling in the blanks. There are more words and phrases than what you need to answer this question.)

Word/phrase list (separated by comma):

FETCH, DECODE, EXECUTE, BUS, EXCEPTION/TRAP REGISTER (ETR), INTERRUPT VECTOR TABLE (IVT), PC, $k0, DEVICE MANUFACTURER, OPERATING SYSTEM, PROCESSOR DESIGNER, HACKER, USER, KERNEL, ANY

(a) Upon an interrupt from a device, the processor enters the INT macro state at the end of the _____ **EXECUTE** _____ macro state.

(b) Upon an internal program discontinuity such as divide by zero, the processor gets the vector number via the _____**ETR**_____.

(c) Upon an interrupt from a device, the processor gets the vector number via the ___**BUS**_____.

(d) The handler address for processing a device interrupt is set in the interrupt vector table (IVT) by the ____**OS**_____.

(e) RETI (return from interrupt) instruction can only be executed in ____**KERNEL**_____ mode of the processor

CS 2200 Spring 2009 Test 1

Prism ID:_____

Name:_____Kishore_____ GTID#: 9_____

Performance

8. (10 points, 10 mins)
Based on typical workloads, HAL engineers figured out the following dynamic instruction frequencies for the three types of instructions:

A – 40%
B – 20%
C – 20%

Two engineering teams independently design processors for the same instruction set and come out with the following designs:

Ma:

Clock cycle time = 1 ns

Type	CPI
A	5
B	3
C	2

Mb:

Clock cycle time = 1.5 ns

Type	CPI
A	4
B	2
C	2

(a) Which machine is faster?

Let E_a and E_b be the normalized execution times of Ma and Mb.

$E_a = (N_A * CPI_A + N_B * CPI_B + N_C * CPI_C)_a * CYCLE\text{-}TIME_a$
$= (0.4 * 5 + 0.2 * 3 + 0.2 * 2) * 1$ ns
$=$ 3 ns

$E_b = (N_A * CPI_A + N_B * CPI_B + N_C * CPI_C)_b * CYCLE\text{-}TIME_b$
$= (0.4 * 4 + 0.2 * 2 + 0.2 * 2) * 1.5$ ns
$= 2.4 * 1.5$ ns
$=$ 3.6 ns

Ma is faster than Mb

(b) what is the speedup of the faster machine over the slower machine

Speedup of Ma over Mb = execution time of Mb/execution time of Ma
= 3.6/3
= 1.2

(c) what is the percentage improvement in the execution time of the faster machine over the slower machine?

Percentage improvement of Ma over Mb
= (difference in execution time/execution time on Mb) * 100
= ((3.6-3)/3.6)* 100 = (0.6/3.6) * 100 = 16.6%

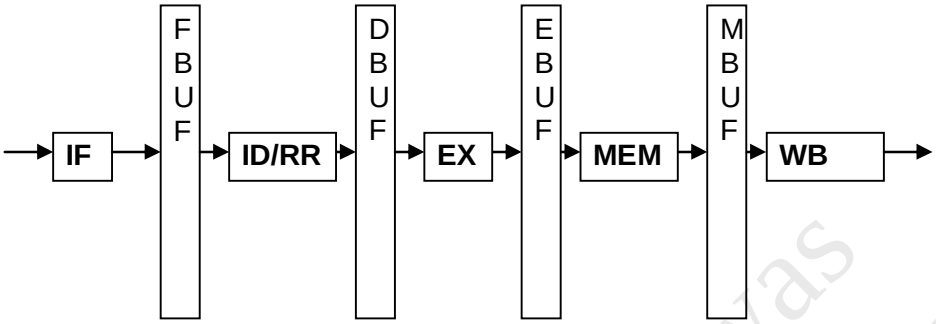
CS 2200 Spring 2009 Test 1

Prism ID:_____

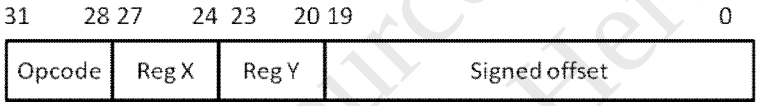
Name:_____Kishore_____ GTID#: 9_____

Pipelining

9. (15 points, 10 mins)
For the LC-2200 instruction set we are considering a pipelined processor design using a 5-stage pipeline as shown below



Assume the instruction going through the pipeline is
SW Rx, Ry, offset; MEM[Ry + signed offset] <- Rx



Considering only the SW instruction, show each item that must be stored in each of the pipeline registers along with its size in bits. You may assume the opcode is in all four of the registers

FBUF (IF:ID/RR)

Same as IR in LC-2200; 32 bits

DBUF (ID/RR:EX)

Opcode = 4 bits; Contents of Ry = 32 bits; Contents of Rx = 32 bits; Signed offset = 20 bits
Total = 88 bits

EBUF (EX:MEM)

Opcode = 4 bits; Result (contents of Ry + signed offset) = 32 bits; Contents of Rx = 32 bits
Total = 68 bits

MBUF (MEM:WB)

Nothing other than Opcode needed for this instruction
Total = 4 bits