CS 2200 Fall 2013 Test 1

Prism ID:_____

Name:_____ GTID#: 9_____

Problem	Points	Lost	Gained	Running Total	TA
1	1				
2	10				
3	15				
4	25				
5	14				
6	10				
7	10				
8	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 with anyone in the class**
- **Please look through the entire test before starting. WE MEAN IT!!!**

Good luck!

1. (1 point, 0 min) (This is a freebie, you get 1 point regardless)
There are multiple schools (or departments in the College of Computing). Circle all the right choices:

- School of Basket Weaving
- School of Computer Science
- School of Electrical and Computer Engineering
- School of Nursing
- School of Interactive Computing
- School of Web Design
- School of Computational Science and Engineering
- School of Chris Klaus
- School of Broken Hardware and Legacy Software

CS 2200 Fall 2013 Test 1

Prism ID:_____

Name:_____ GTID#: 9_____

Processor design

2. (10 points, 5 mins)

(a) (circle the correct choice) Saving and restoring of registers on a procedure call…

- Is always done by the caller
- Is always done by the callee
- Is never done since hardware magically takes care of it
- Is done on a need basis partly by the caller and partly by the callee

(b) (circle the correct choice) Local variables in a procedure…

- Are usually allocated on the stack
- Are usually kept in processor registers
- Are usually kept in a special hardware
- Are usually allocated in the heap space of the program
- Are usually allocated in the static (global) data space of the program

(c) (circle the correct choice) Frame pointer…

- Another name for stack pointer
- Changes every time items are pushed and popped on the stack
- Changes every time local variables for a procedure are allocated on the stack
- Is a fixed harness into the activation record of a currently executing procedure

(d) (circle the correct choice) Addressing mode…

- Refers to the kinds of opcodes supported in an architecture
- Refers to the way the operands are specified in an instruction
- Refers to the granularity of the memory element that can be addressed in an instruction
- Refers to the datapath width

(e) (circle the correct choice) An instruction set…

- Serves as a level of abstraction between software and hardware
- Provides the details of the machine implementation
- Deals with the datapath and control of the processor
- Specifies how the microcode is laid out in the control ROM

CS 2200 Fall 2013 Test 1

Prism ID:_____

Name:_____ GTID#: 9_____

3.

(a) (10 points, 5 min)
Given the software convention for registers:

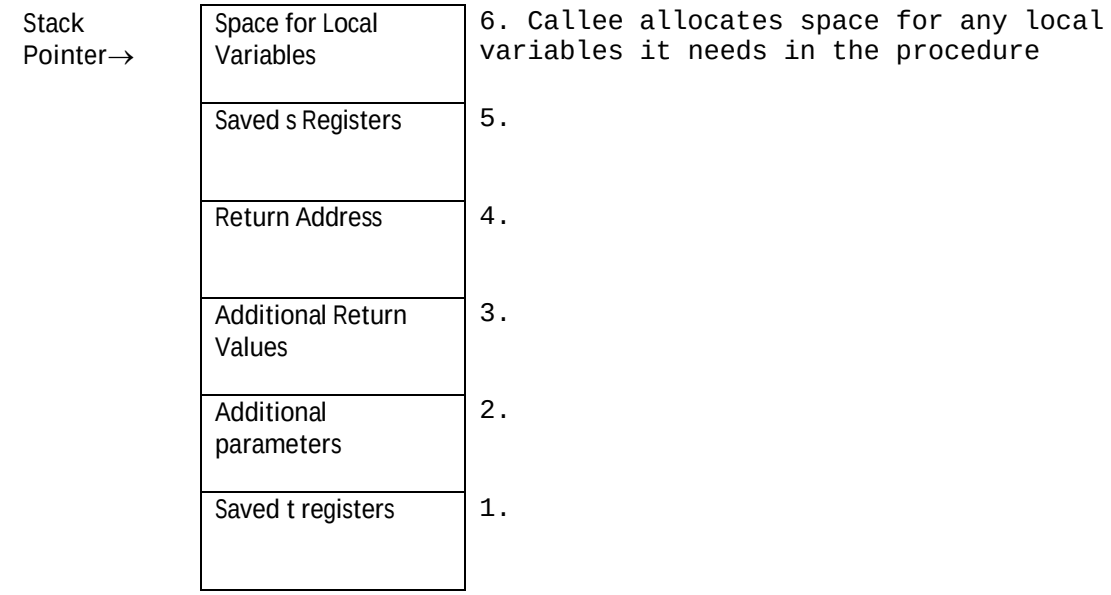
```
a0-a2:  parameter passing
s0-s2:  callee saves if need be
t0-t2:  caller saves if need be
v0:     return value
ra:     return address
at:     target address
sp:     stack pointer
```

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)
```

At the point of the call, the state of the stack is as shown below. To help you out, we have put down the action corresponding to step 6. Fill out the actions similarly for the other steps before the called procedure can start executing (who is responsible for the action caller/callee, and what is the action).

Your answer:



(b) (5 points, 2 min) Why is it not a good idea to make the caller or the callee to be solely responsible for saving/restoring of registers in the procedure call convention?

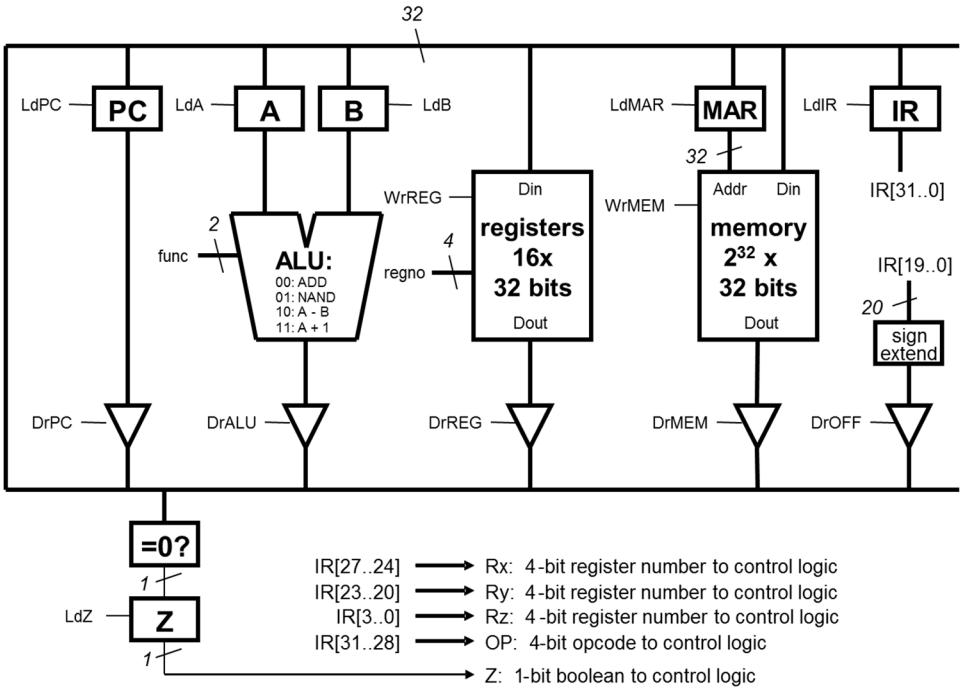
CS 2200 Fall 2013 Test 1

Prism ID:_____

Name:_____ GTID#: 9_____

Datapath and control

4. All the questions are in relation to the datapath shown
A. (10 points, 5 min)



(a) (circle the correct choice)For the FSM-based implementation of BEQ instruction,

- The branch target address calculation has to be done always
- The branch target address calculation has to be done only if the operands are equal
- The branch target address calculation has to be done only if the operands are unequal
- The branch target address is already in PC

(b) (circle the correct choice)The duration of a micro state in the FSM-based implementation of LC-2200 is

- One clock cycle
- One macro state
- As many clock cycles as needed by the macro state
- Three clock cycles

(c) (circle the correct choice) With a single-ported register-file, getting two register contents (Ry and Rz) needed for an ADD instruction requires

- One clock cycle
- Two clock cycles
- Three clock cycles
- Sixty four clock cycles

(d) (circle the correct choice)Implementing the FETCH macro state would require

- One micro state
- Two micro states
- Three micro states
- Four micro states

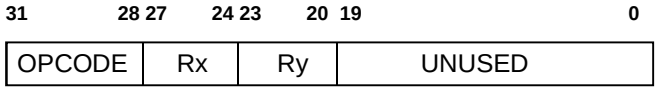
(e) (circle the correct choice)The Z register is used for

- Effecting a multi-way branch at the end of FETCH macro state
- Generating a constant value during the execution of ADDI instruction
- Effecting a 2-way branch during the execution of BEQ instruction
- Saving the return address during the execution of JALR instruction

B. (15 points, 10 min)
We have decided to add another addressing mode **autoincrement** to LC-2200. This mode comes in handy for LW/SW instructions. The semantics of this addressing mode with LW instruction is as follows:

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

The instruction format is as shown below:



Write the sequence for implementing the LW instruction with this addressing mode (you need to write the sequence for the execution macro state of the instruction). For each microstate, show the datapath action (in register transfer format such as A ← Ry) along with the control signals you need to enable for the datapath action (such as DrPC).

Prism ID:_____

Name:_____ GTID#: 9_____

Additional space for 4.B

Interrupts

5. (14 points, 7 mins)

a) (circle the correct choice) An interrupt handler saves and restores the entire register-file since

(i) we may never return to the original program from the handler
(ii) it does not know what registers may have been in use at the time of the interrupt in the original program
(iii) it is a good software engineering practice
(iv) interrupts could be nested

Prism ID:_____

Name:_____ GTID#: 9_____

b) (circle the correct choice) The interrupt vector table holds the starting addresses of the interrupt handlers for all the known sources of program discontinuities during program execution. This table is set up

(i) by the OS every time a program makes a system call
(ii) by the OS at boot time (i.e., every time we re-start the OS)
(iii) preset by the hardware at manufacturing time of the computer
(iv) by each application when it is started up on your computer

c) (circle the correct choice) We need a "return from interrupt" (RTI) instruction since

(i) we need to enable interrupts in addition to returning to the original program
(ii) the handler does not know where to return to in the original program
(iii) it is more intuitive as to what we want to accomplish than a simple jump instruction
(iv) we need to store the handler address back in the interrupt vector table in addition to returning to the original program

d) (circle the correct choice) Upon executing an RTI instruction the processor always goes back to
(i) user mode
(ii) system mode
(iii) mode prior to entering this interrupt handler

e) (circle the correct choice) The difference between an external device interrupt and an internal trap or exception is that
(i) there is none
(ii) the vector number for a trap/exception is internally generated by the processor
(iii) the vector number for a trap/exception is provided by the user in one of the general-purpose registers
(iv) the vector number that is put out by a given device, changes each time it interrupts

f) (circle the correct choice) We need an "Enable Interrupt" instruction in the ISA since
(i) each individual instruction is not atomic
(ii) the processor often needs to execute a set of instructions atomically
(iii) we have to allow higher priority interrupts to interrupt a lower priority handler
(iv) the handler needs to execute that instruction before saving the return address in **$k0** on the stack

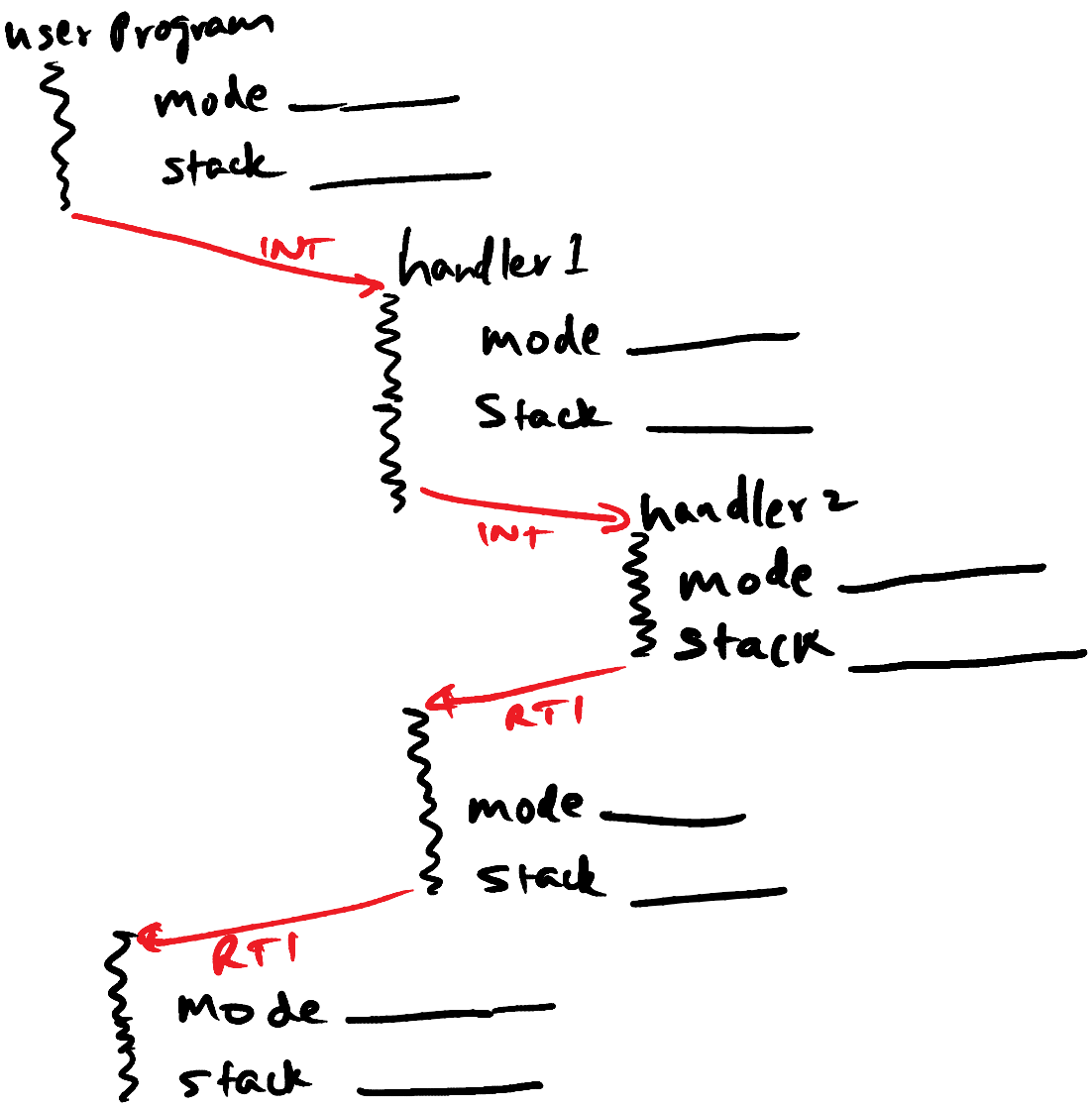
g) (circle the correct choice) Identify which of the following is NOT atomic
(i) An individual instruction execution in LC-2200
(ii) Saving all the registers on the stack
(iii) The processor action in the INT macro state
(iv) The processor action in the FETCH macro state

CS 2200 Fall 2013 Test 1

Prism ID:_____

Name:_____ GTID#: 9_____

6. (10 points, 5 mins)Fill in the blanks below to indicate what is the mode of the processor and what is the stack in use



CS 2200 Fall 2013 Test 1

Prism ID:_____

Name:_____ GTID#: 9_____

Performance

7. (10 mins)
a) (5 points)
A processor spends 20% of its time on add instructions. An engineer proposes to improve the add instruction by 4 times. What is the **speedup** achieved because of the modification?

b) (5 points)
Consider the program shown below that consists of 1000 instructions.

I_1:
I_2:
..
..
..
I_{10}:
I_{11}: ADD
I_{12}:
I_{13}:
I_{14}: COND BR I_{10}
..
..
I_{1000}:

loop

ADD instruction occurs exactly once in the program as shown. Instructions I_{10}-I_{14} constitute a loop that gets executed 900 times. All other instructions execute exactly once. Calculate the **dynamic frequency** of ADD instruction in this program.

CS 2200 Fall 2013 Test 1

Prism ID:_____

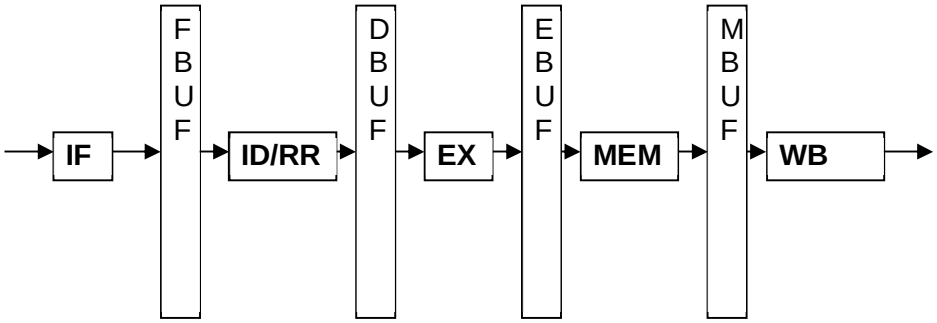
Name:_____ GTID#: 9_____

Pipelining

8. (10 mins)

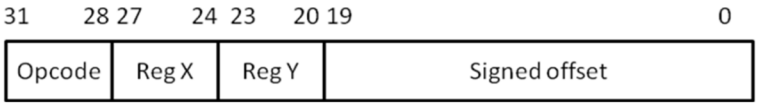
a) (2 points) In Bill's sandwich shop employing 5 workers to form an assembly line, the latency incurred by each customer from the time she places her order is _____ time units, and the steady-state throughput achieved is _____ sandwiches per time unit.

b) (13 points) 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

LW Rx, Ry, offset; Rx <- MEM[Ry + signed offset]



Considering only the LW 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

CS 2200 Fall 2013 Test 1

Prism ID:_____

Name:_____ GTID#: 9_____

FBUF (IF:ID/RR)

DBUF (ID/RR:EX)

EBUF (EX:MEM)

MBUF (MEM:WB)

CS 2200 Fall 2013 Test 1

Prism ID:_____

Name:_____ GTID#: 9_____

Metrics cheat sheet:

Name	Notation	Units	Comment
Memory footprint	-	Bytes	Total space occupied by the program in memory
Execution time	$(\sum CPI_j) * \text{clock cycle time}$, where $1 \le j \le n$	Seconds	Running time of the program that executes *n* instructions
Arithmetic mean	$(E_1+E_2+\ldots+E_p)/p$	Seconds	Average of execution times of constituent *p* benchmark programs
Weighted Arithmetic mean	$(f_1*E_1+f_2*E_2+\ldots+f_p*E_p)$	Seconds	Weighted average of execution times of constituent *p* benchmark programs
Geometric mean	p^{th} root $(E_1*E_2*\ldots*E_p)$	Seconds	p^{th} root of the product of execution times of *p* programs that constitute the benchmark
Harmonic mean	$1/(((1/E_1)+(1/E_2)+\ldots+(1/E_p)))/p)$	Seconds	Arithmetic mean of the reciprocals of the execution times of the constituent *p* benchmark programs
Static instruction frequency		%	Occurrence of instruction *i* in compiled code
Dynamic instruction frequency		%	Occurrence of instruction *i* in executed code
Speedup (M_A over M_B)	E_B/E_A	Number	Speedup of Machine A over B
Speedup (improvement)	E_{Before}/E_{After}	Number	Speedup After improvement
Improvement in Exec time	$(E_{old}-E_{new})/E_{old}$	Number	New Vs. old
Amdahl's law	$Time_{after} = Time_{unaffected} + Time_{affected}/x$	Seconds	*x* is amount of improvement