CS 2200 Spring 2017 Test 1 - A

Prism ID:________________________

Name:_______Kishore______________

Problem	(Points, Minutes)	Lost	Gained	Running Total	TA
1	10, 5				
2	26, 10				
3	6, 3				
4	18, 10				
5	18, 9				
6	10, 5				
7	12, 8				
Total	100, 50				

- **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!

0. (no score, 0 min)
"Transistor density doubles on the chip every 2 years" (attributed to?)
   (a) Bill Gates
   (b) Donald Trump
   (c) Larry Ellison
   (d) Sergei Brin
   (e) Gordon Moore
   (f) Al Gore
   (g) Gordon Bell
   (h) Gene Amdahl
   (i) Vladimir Putin

CS 2200 Spring 2017 Test 1 - A

Prism ID:____________________________

Name:________Kishore________________

Processor design

1. (10 points, 5 minutes) Given the following data declarations:

```
short s; /* occupies 2 bytes; s1, s0 */
int   a; /* occupies 4 bytes; a3, a2, a1, a0 */
char  b; /* occupies 1 byte */
char  c; /* occupies 1 byte */
int   d; /* occupies 4 bytes; d3, d2, d1, d0 */
char  e; /* occupies 1 byte */
short f; /* occupies 2 bytes; f1, f0 */
```

Consider a **32-bit big-endian** architecture; the architecture supports **L/S** instructions at **byte**, **half-word**, and **word** granularities. The architecture expects address alignment of memory operands commensurate with the granularity.
How will the compiler lay out the above data declarations in memory starting at **memory address 100** to achieve the **best space/time tradeoff**? [**Note: The declarations cannot be reordered.]**

In the following memory picture each row represents a memory word comprising of 4 bytes, and each cell represents a byte. Fill it in with your answer.

Byte address ------->

WORD ADDRESS	+0	+1	+2	+3
100	s_1	s_0		
104	a_3	a_2	a_1	a_0
108	b	c		
112	d_3	d_2	d_1	d_0
116	e		f_1	f_0
120				
124				
128				

(-3 for not getting endianness right)
(-1 for not getting data alignment right for each variable)
(NO DOUBLE JEOPARDY)

CS 2200 Spring 2017 Test 1 - A

Prism ID:_________________________

Name:_______Kishore______________

Datapath and control

2. (Total points 26, 10 min)

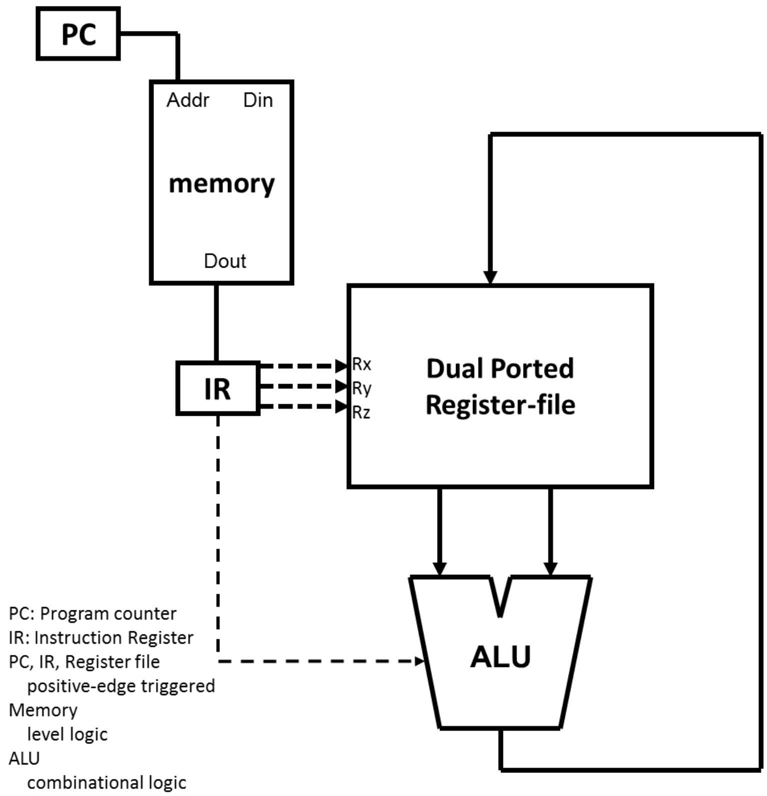
(a) (8 points) Given the following datapath

PC has address of current instruction, which is an ADD instruction with the semantic:

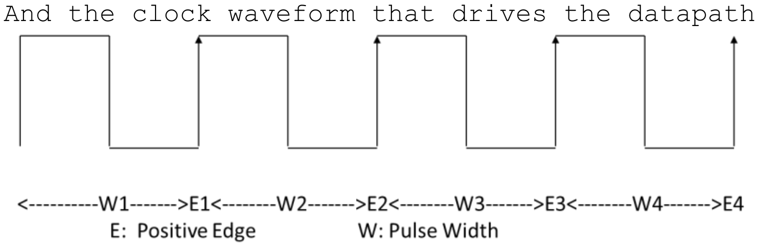
ADD Rx, Ry, Rz; Rx <- Ry + Rz.

Intention is to fetch the instruction from memory and execute the above semantic.

Given the datapath:



And the clock waveform that drives the datapath:



CS 2200 Spring 2017 Test 1 - A

Prism ID:________________________

Name:_______Kishore______________

Show what happens in each clock cycle at the clock **edge(E)** and/or clock pulse **width(W)** in the table below by **circling ONLY** the datapath elements that are involved in the datapath action. (Note: Wrong selection will result in negative points).

Clock	PC		IR		REGFILE		MEMORY		ALU
W1	R	W	R	W	Read Ry, Rz	Write Rx	R	W	ADD
E1	R	W	R	W	Read Ry, Rz	Write Rx	R	W	ADD
W2	R	W	R	W	Read Ry, Rz	Write Rx	R	W	ADD
E2	R	W	R	W	Read Ry, Rz	Write Rx	R	W	ADD
W3	R	W	R	W	Read Ry, Rz	Write Rx	R	W	ADD
E3	R	W	R	W	Read Ry, Rz	Write Rx	R	W	ADD
W4	R	W	R	W	Read Ry, Rz	Write Rx	R	W	ADD
E4	R	W	R	W	Read Ry, Rz	Write Rx	R	W	ADD

R: Read

W: Write

(+1 for register write on edges)
(+1 for each correct selection)
(-1 for each wrong selection)

(b) (2 points) Why is it that we should not hardcode the register number to be read/written from the register file in the microinstruction?

The registers are specified dynamically in the instruction. Hardcoding implies statically specifying the register number(s) to be read/written in the microinstruction which does not make sense.

(All or nothing)

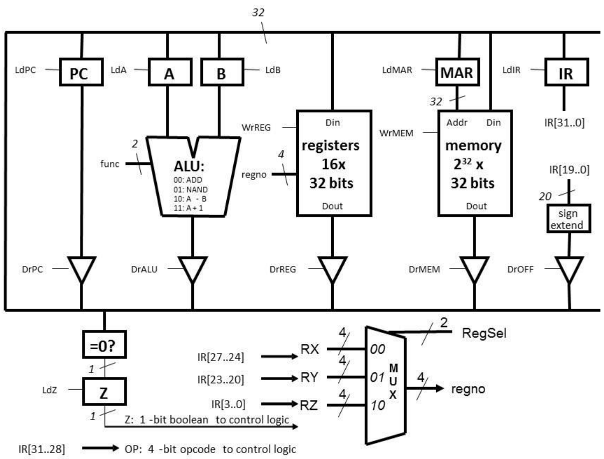
CS 2200 Spring 2017 Test 1 - A

Prism ID:___________________________

Name:________Kishore________________

(c) All the questions are in relation to the datapath shown below((

For all the parts of this question:
(-0.5 for each wrong entry)



(i) (2 points) Place a 1 for the control signals that need to be enabled to do the following operation: [Grab your reader's attention with a great quote from the document or use this space to emphasize a key point. To place this text box anywhere on the page, just drag it.]

PC -> MAR, A, B

Drive Signals					Load Signals						Write Signals		ALU	Reg File
PC	ALU	Reg	MEM	OFF	PC	A	B	MAR	IR	Z	MEM	REG	Func	RegSel
1						1	1	1						

(ii) (2 points) Place a 1 for the control signals that need to be enabled to do the following operation:

Perform Arithmetic A-B in the ALU;
Clock Z register with the result of the zero-detect logic of the ALU result

Drive Signals					Load Signals						Write Signals		ALU	Reg File
PC	ALU	Reg	MEM	OFF	PC	A	B	MAR	IR	Z	MEM	REG	Func	RegSel
	1									1			10	

CS 2200 Spring 2017 Test 1 - A

Prism ID:________________________

Name:________Kishore________________

(iii) (2 points) Place a 1 for the control signals that need to be enabled to do the following operation:

Register Rz -> B

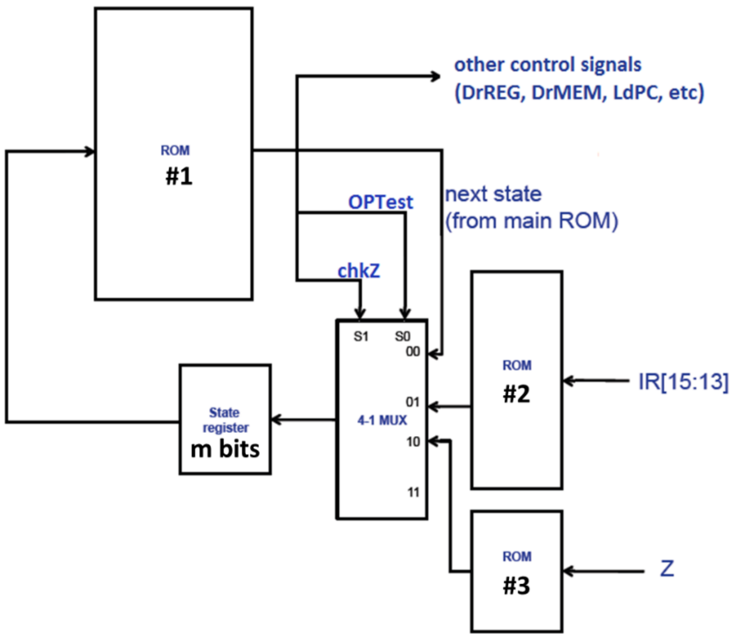
Drive Signals					Load Signals						Write Signals		ALU	Reg File
PC	ALU	Reg	MEM	OFF	PC	A	B	MAR	IR	Z	MEM	REG	Func	RegSel
		1					1							10

CS 2200 Spring 2017 Test 1 - A

Prism ID:___________________________

Name:________Kishore________________

(d) All the questions are with respect to the microsequencer shown below



(i) (2 points) How many entries are there in ROM#1? Why?

2^m entries; since the state register is m bits

(+1 for each number of entries; +1 for reasoning)

(ii) (2 point) How many entries are there in ROM#2? Why?

2^3 entries; since the opcode is using 3-bits IR 15-13.

(+1 for each number of entries; +1 for reasoning)

(iii) (2 point) How many entries are there in ROM#3? Why?

2 entries; based on the Z bit choose one of two place to do a micro-branch

(+1 for each number of entries; +1 for reasoning)

(iv) (2 point) When is ROM#2 chosen as the input to the state register?

At the end of the FETCH macro state.

(All or nothing)

(v) (2 point) When is ROM#3 chosen as the input to the state register?

In the middle of the BEQ sequence, to determine if branch target address computation needs to be done or not based on the result of the equality check.

(All or nothing)

CS 2200 Spring 2017 Test 1 - A

Prism ID:____________________________

Name:________Kishore________________

Interrupts

3. (6 points, 3 min) Fill in the table below

	Sync/ Async	Internal/ External	Intentional Yes/No	Examples
Exception	**Sync**	**Internal**	**No**	**Overflow, Divide by zero**
Trap	**Sync**	**Internal**	**Yes**	**System Call**
Interrupt	**Async**	**External**	**Yes**	**I/O Device Completion**

(+0.5 for each slot in the table)
(Trap/Exception: intentional (Yes) / (Yes and No) bot answers accepted)

CS 2200 Spring 2017 Test 1 - A

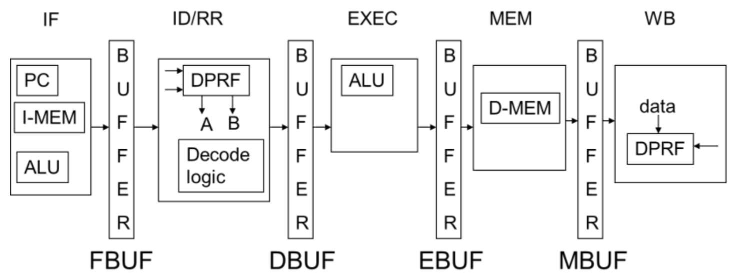
Prism ID:____________________________

Name:________Kishore________________

Pipelining

4. (18 points, 10 mins)

For the LC-2200 instruction set we are considering a pipelined processor design using a 5-stage pipeline as shown below



Recall the semantics of JALR instruction:

JALR, Rx, Ry;   Ry <- PC; PC <- Rx

31 - 28	27 - 24	23 - 20	19 - 0
JALR	Rx	Ry	unused

Show the anatomy of JALR instruction as it goes through the pipeline by showing what work is done in each stage for the instruction and the contents of the buffer after each stage (you have to show what information gets stored in the buffer after each stage and the exact size). Recall that all operands in LC-2200 are 32-bits in length.

Notes:
1) PC gets written with the jump target address in the ALU stage of the pipeline.
2) The purpose of this question is ONLY to know work done in each stage and the contents of the buffers.
3) Do NOT worry about other instructions that may be following the JALR instruction for the purposes of this question.
4) By the same token, you do not have to worry about any other instruction in the buffer design for this problem.

(a) IF Stage **(4 points total)**

Work done: **(+1 for work done)**

PC -> I-MEM -> FBUF; Fetch instruction from memory into FBUF
PC + 1 -> PC; return address
PC -> FBUF; Put the return address into FBUF

FBUF contents: **(-0.5 for each missing entry shown in red below; net score >= 0)**

4 bits	4 bits	4 bits	32 bits
JALR opcode	Rx Specifier	Ry Specifier	PC contents (return address)

Prism ID:___________________________

Name:________Kishore_______________

(b) ID/RR stage **(4 points total)**

Work done: **(+1 for work done)**

```
DPRF[FBUF[Rx Specifier]] -> DBUF
FBUF[PC contents] -> DBUF
FBUF[Ry Specifier] -> DBUF
FBUF[JALR opcode] -> DBUF
```

DBUF contents: **(-0.5 for each missing entry shown in red below; net score >= 0)**

4 bits	32 bits	4 bits	32 bits
JALR opcode	Contents of Rx	Ry Specifier	PC contents (return address)

(b) EX stage **(4 points total)**

Work done: **(+1 for work done)**

```
DBUF[PC contents] -> EBUF
DBUF[Ry Specifier] -> EBUF
DBUF[JALR opcode] -> EBUF
DBUF[Contents of Rx] -> PC; Target address of branch stored in PC
```

EBUF contents: **(-0.5 for each missing entry shown in red below; net score >= 0)**

4 bits	4 bits	32 bits
JALR opcode	Ry Specifier	PC contents (return address)

(b) MEM stage **(4 points total)**

Work done: **(+1 for work done)**

```
EBUF[PC contents] -> MBUF
EBUF[Ry Specifier] -> MBUF
EBUF[JALR opcode] -> MBUF
```

MBUF contents: **(-0.5 for each missing entry shown in red below; net score >= 0)**

4 bits	4 bits	32 bits
JALR opcode	Ry Specifier	PC contents (return address)

(b) WB stage **(2 points total)**

Work done: **(ALL or nothing)**
MBUF[PC Contents] -> DPRF[MBUF[Ry Specifier]];
return address saved in link register

Potpourri

5. (Total points 18, 9 min)

(a) (2 points) Addressing mode refers to the following (choose one right answer)
1. Refers to the kinds of opcodes supported in an architecture
2. Refers to the way the operands are specified in an instruction
3. Refers to the granularity of the memory element that can be addressed in an instruction
4. Refers to the datapath width

(b) (2 points) In LC-2200, Saving and restoring of registers on a procedure call (choose one right answer)
1. Is always done by the caller
2. Is always done by the callee
3. Is never done since hardware magically takes care of it
4. Is done on a need basis partly by the caller and partly by the callee

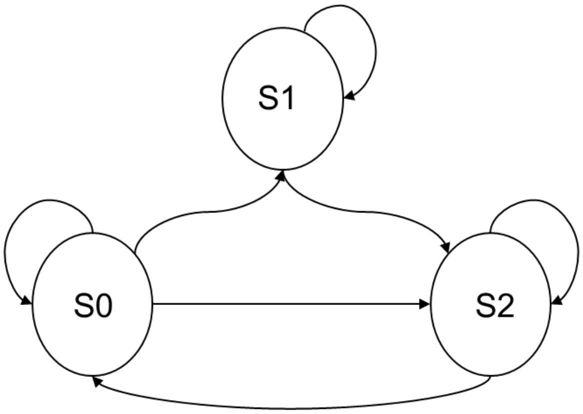
(c) (2 points) Local variables in a procedure (choose one right answer)
1. Are usually allocated on the stack
2. Are usually kept in a special hardware
3. Are usually allocated in the heap space of the program
4. Are usually allocated in the static (global) data space of the program

(d) (2 points) Frame pointer (choose one right answer)
1. Points to the top of the stack
2. Changes every time items are pushed and popped on the stack
3. Changes every time local variables for a procedure are allocated on the stack
4. Is a fixed harness into the activation record of a currently executing procedure

(e) (2 points) Choice of instructions to have in the ISA of a processor (choose one right answer)
1. Is decided entirely by the amount of real estate available in the silicon
2. Refers to how the various fields of an instruction are laid out in memory
3. Is largely influenced by high level language constructs
4. Is entirely influenced by the power consumption budget for the processor

(f) (2 points) A micro state is:
1. Datapath actions in one clock cycle
2. The width of a microinstruction in the control ROM
3. Datapath actions in FETCH macro state
4. Datapath actions needed in any macro state

(g) (2 points) Given the FSM below:



The number of rows in the state transition table is (Circle one correct choice):

1. Three
2. Four
3. Five
4. Six
5. Seven

(h) (4 points) Characterize the actions below interruptible/non-interruptible

Action	Interruptible/non-interruptible
Datapath actions in one clock cycle	**Non-interruptible**
Datapath actions in one macro state	**Non-interruptible**
Individual instruction execution	**Non-interruptible**
Execution of a set of instructions	**Interruptible**

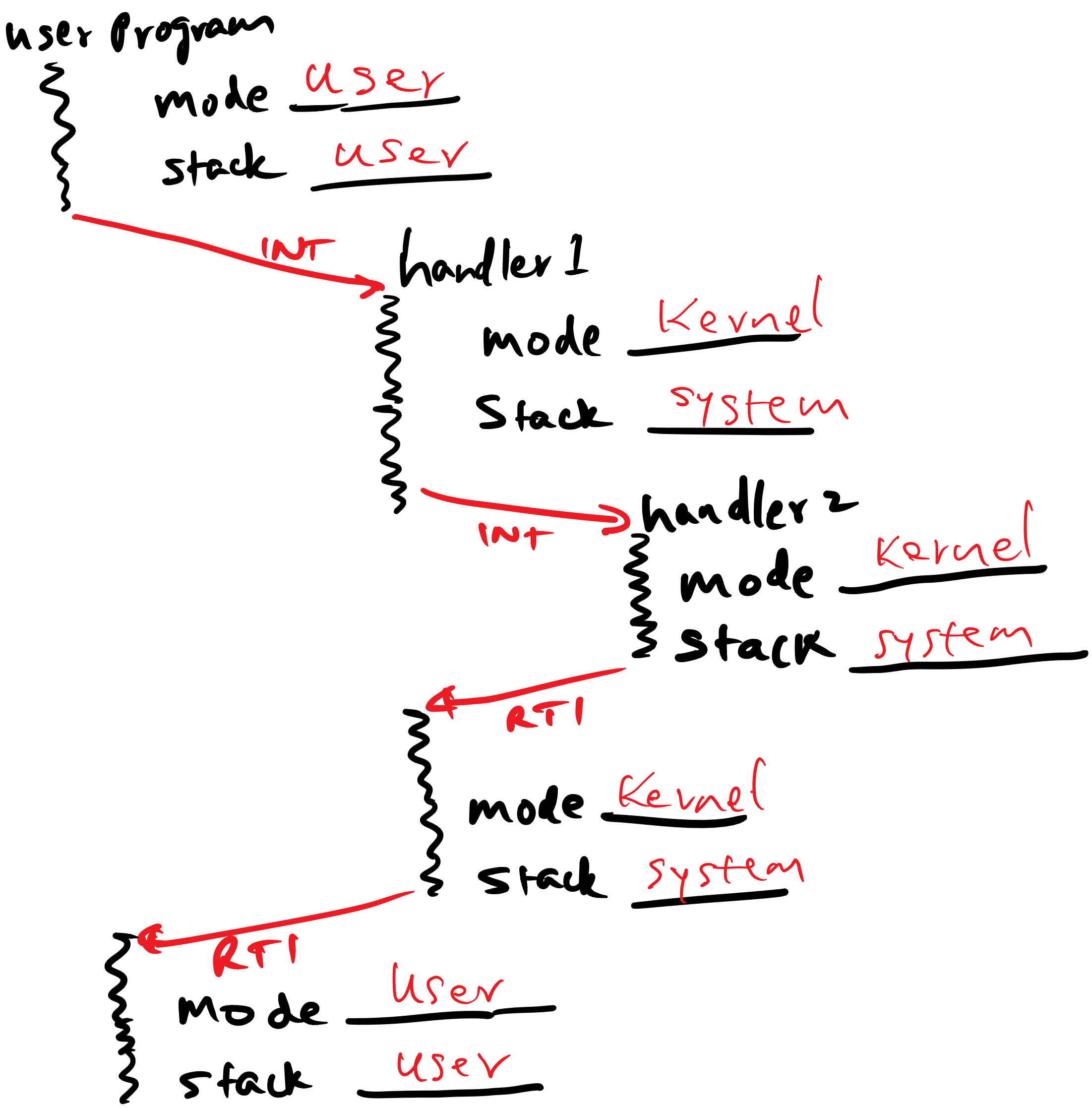
Invariants

CS 2200 Spring 2017 Test 1 - A

Prism ID:_________________________

Name:_______Kishore______________

6. (10 points, 5 min)
   Fill in the blanks below to indicate what is the mode (**user, kernel**) of the processor and what is the stack in use (**user, system**)



CS 2200 Spring 2017 Test 1 - A

Prism ID:_________________________

Name:_______Kishore______________

Performance

7. (Total 12 points, 8 min)

(a) (10 points) An architecture has three types of instructions that have the following CPI:

Type	CPI
A	3
B	6
C	1

An architect determines that he can reduce the CPI for B to 4, with no change to the CPIs of the other two instruction types, but with an increase in the clock cycle time of the processor. What is the maximum permissible percentage increase in clock cycle time that will make this architectural change still worthwhile? Assume that all the workloads that execute on this processor use 20% of A, 20% of B, and 60% of C types of instructions.

(You have to show your work to get any credit)

EX_{old} = Ex time of old machine = $N * (F_A * CPI_{Ao} + F_B * CPI_{Bo} + F_C * CPI_{Co}) * CL_{old}$

$= N * (0.2 * 3 + 0.2 * 6 + 0.6 * 1) * CL_{old}$

$= N * 2.4 * CL_{old}$

EX_{new} = Ex time of new machine = $N * (0.2 * 3 + 0.2 * 4 + 0.6 * 1) * CL_{new}$

$= N * (2.0) * CL_{new}$

For design to be viable,

$EX_{new} < EX_{old}$

$2*CL_{new} < 2.4*CL_{old}$

$CL_{new} < 1.2*CL_{old}$

Max permissible percentage increase in clock speed = 20%

(+4 formula right
+2 all terms for old machine right
+2 all terms for new machine right
+2 inequality right)

b) (2 points) (circle the correct choice) Static instruction frequency…

1. Refers to the type of instructions in the instruction-set
2. Refers to the frequency of occurrence of instructions in compiled code
3. Refers to the frequency of occurrence of instructions that actually get executed
4. Refers to clock frequency of the processor
5. Is the basis for datapath design