

CS 2200 Spring 2017 Test 1 - A

Problem	(Points, Minutes)	Lost	Gained	Running Total	TA
1	10, 5	0	10	10	DR
2	26, 10	0	26	36	IGY
3	6, 3	0	6	42	NL
4	18, 10	0	18	60	YW
5	18, 9	2	16	76	NL
6	10, 5	0	10	86	IGY
7	12, 8	0	12	98	WX
Total	100, 50			98	WX

- 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!

CS 2200 Spring 2017 Test 1 - A

Prism ID: _____
Name: _____

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 ----->

	+0	+1	+2	+3
W 100	s ₁	s ₀	1111	1111
O 104	a ₃	a ₂	a ₁	a ₀
R	b	c	1111	1111
D 108	d ₃	d ₂	d ₁	d ₀
	1111	1111	1111	1111
A 112	e	1111	f ₁	f ₀
D 116				
D 120				
R				
E 124				
S 128				
S				

CS 2200 Spring 2017 Test 1 - A

Prism ID: _____

Name: _____

Datapath and control

2. (Total points 26, 10 min)

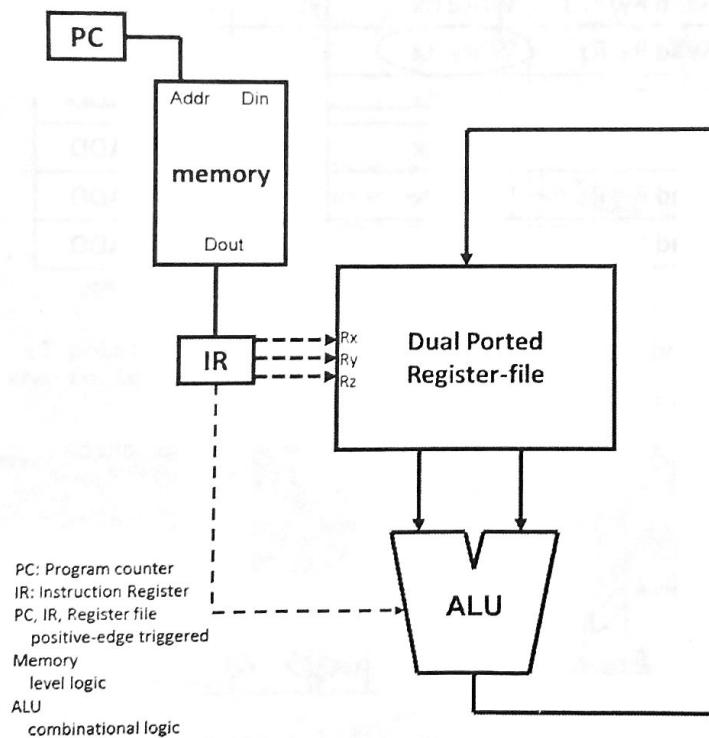
- (a) (8 points) Given the following datapath

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

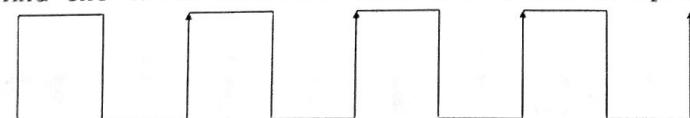
ADD Rx, Ry, Rz; $Rx \leftarrow Ry + Rz$.

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

Given the datapath:



And the clock waveform that drives the datapath:



<-----W1----->E1<-----W2----->E2<-----W3----->E3<-----W4----->E4
E: Positive Edge W: Pulse Width

CS 2200 Spring 2017 Test 1 - A

Prism ID: _____

Name: _____

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

- (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?

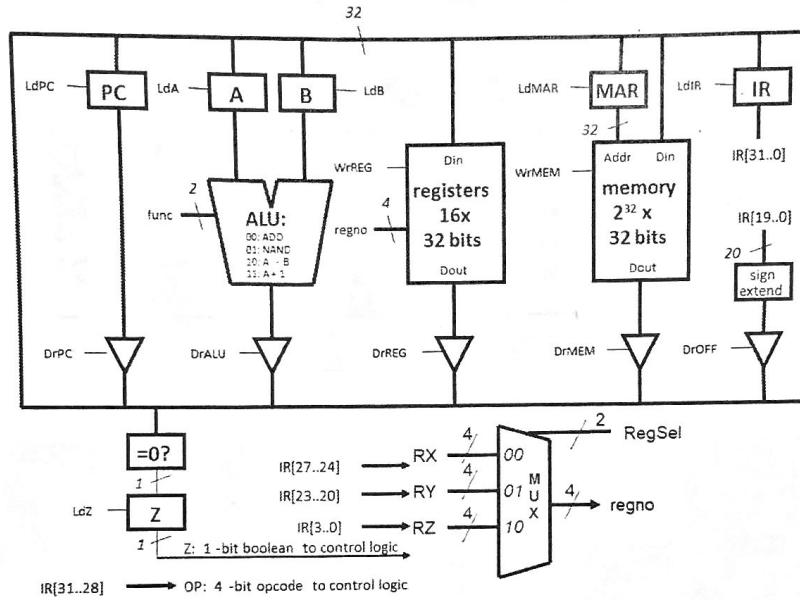
We can't hardcode the register numbers because the registers are set by the user per instruction issued. Instead we read from the 6-bit-ranges in the instruction where the register numbers are stored.

CS 2200 Spring 2017 Test 1 - A

Prism ID: _____

Name: _____

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



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

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

(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
													10	

CS 2200 Spring 2017 Test 1 - A

Prism ID: _____

Name: _____

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

Register Rz \rightarrow B

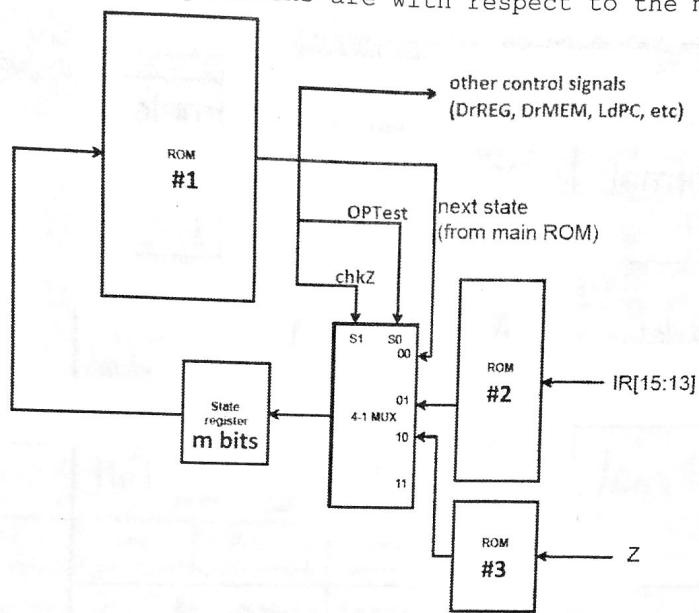
Drive Signals					Load Signals					Write Signals			ALU	Reg File
PC	ALU	Reg	MEM	OFF	PC	A	B	MAR	IR	Z	MEM	REG	Func	RegSel
		1					1							10

CS 2200 Spring 2017 Test 1 - A

Prism ID:

Name:

- (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 has m bits. However, some entries may be zeroed out if there are not exactly 2^m microstates

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

$2^3 = 8$ entries, because there are 8 possible 3-bit patterns for IR[5:13].
Some entries may be zeroed.

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

$2^1 = 2$ (assuming Z is a 1 bit wide), because the address may only be a 0 or a 1

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

Rom 2 is chosen at the end of the fetch state to determine which macrostate to go to based on the opcode. It is chosen when ROM1 is signalling OPTest

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

Rom 3 is chosen when the next microstate is determined by conditional logic, like the BEQ instruction. It is used when Rom1 signals chkZ.

(a branch-type instruction)

CS 2200 Spring 2017 Test 1 - A

Prism ID: _____

Name: _____

Interrupts

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

	Sync/ Async	Internal/ External	Intentional Yes/No	Example
Exception	Sync	Internal	NO	Divide by 0
Trap	Sync	Internal	YES	System Call
Interrupt	Async	External	Yes	IO device ready

✓

CS 2200 Spring 2017 Test 1 - A

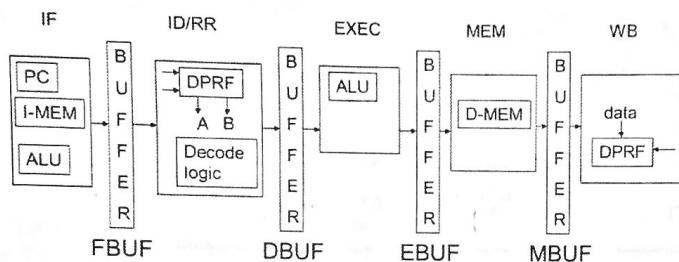
Prism ID: _____

Name: _____

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

Work done:
 $FBUF[Ins] \leftarrow I-MEM[PC]$
 $PC \leftarrow PC+1, FBUF[PC] \leftarrow PC$ (incremented)
 FBUF contents:

OP	RX	RY	unused	PC (incremented)
4	4	4	20	32

CS 2200 Spring 2017 Test 1 - A

Prism ID: _____
 Name: _____

(b) ID/RR stage

Work done:

$$\begin{aligned} DBUF[A] &\leftarrow DPRF[FBUF[RX]] \\ DBUF[OP] &\leftarrow FBUF[OP] \\ DBUF[RY] &\leftarrow FBUF[RY] \quad DBUF[PC] \leftarrow FBUF[PC] \end{aligned}$$

DBUF contents:

OP	RY	A	PC
4	4	32	32

(b) EX stage

Work done:

$$PC \leftarrow DBUF[A], EBUF[OP] \leftarrow DBUF[OP], EBUF[RY] \leftarrow DBUF[RY], EBUF[PC] \leftarrow DBUF[PC]$$

EBUG contents:

OP	RY	PC
4	4	32

(b) MEM stage

Work done:

$$MBUF \leftarrow EBUF$$

MBUF contents:

OP	RY	PC
4	4	32

(b) WB stage

Work done:

$$DPRF[MBUF[RY]] \leftarrow MBUF[PC]$$

CS 2200 Spring 2017 Test 1 - A

Prism ID: _____

Name: _____

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

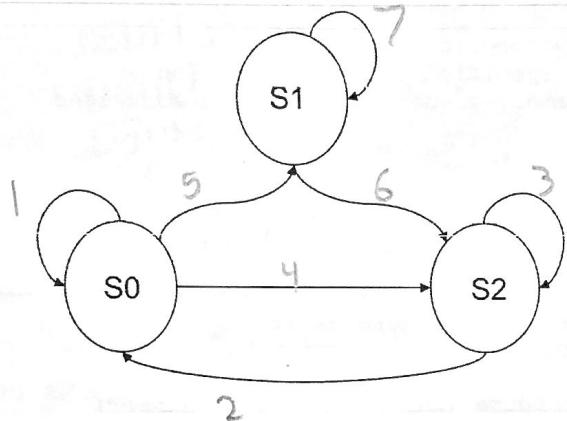
(f) (2 points) A micro state is:

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

CS 2200 Spring 2017 Test 1 - A

Prism ID: _____
Name: _____

(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

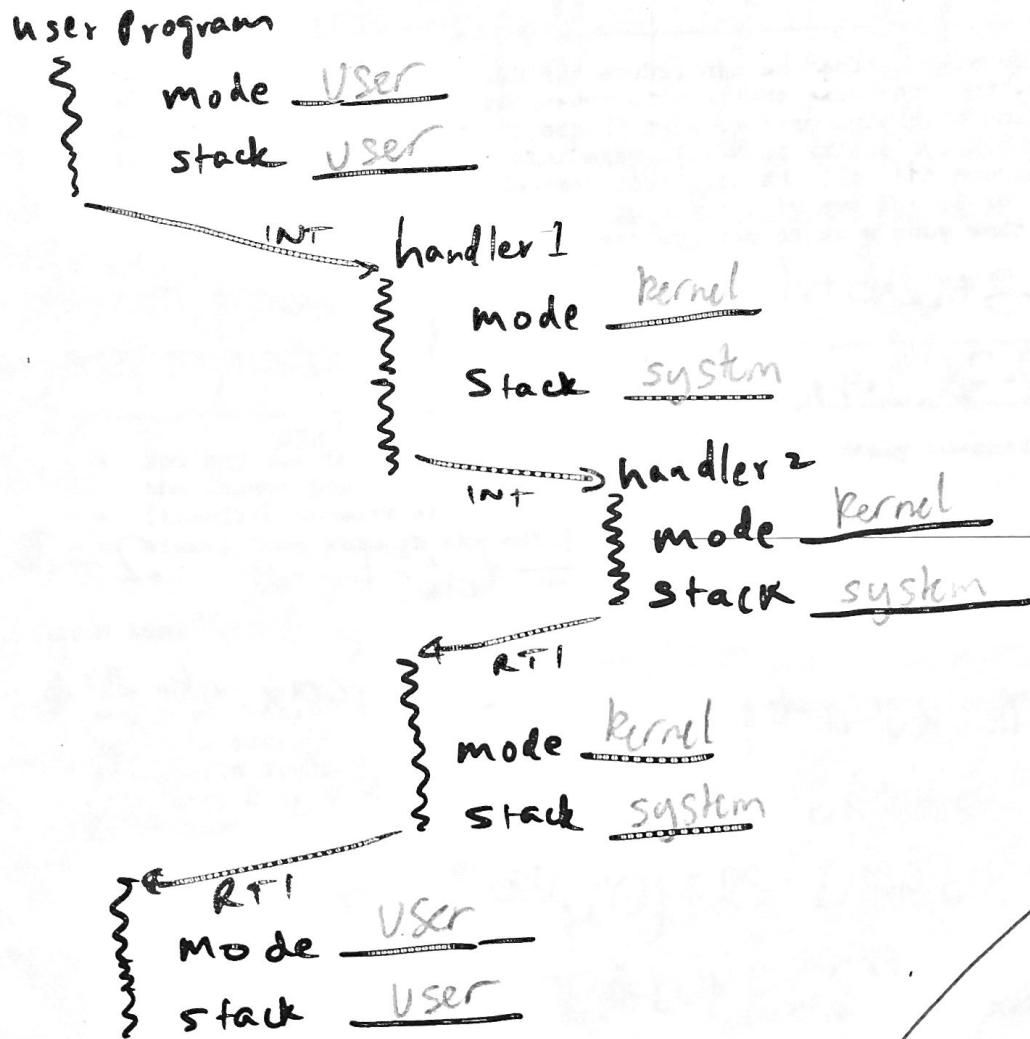
CS 2200 Spring 2017 Test 1 - A

Prism ID: _____
Name: _____

Invariants

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

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)

$$ET_{old} = [2 \cdot 3 + 2 \cdot 6 + 6 \cdot 1] = [6 + 12 + 6] = 2.4t_{old}$$

$$ET_{new} = [2 \cdot 3 + 2 \cdot 4 + 6 \cdot 1] = [6 + 8 + 6] = 2.0t_{new}$$

$$1 = \frac{2.4t_{old}}{2.0t_{new}} \quad t_{new} = \frac{2.4}{2}t_{old} = 1.2t_{old} \quad .2 \rightarrow 20\%$$

The new clock cycle time must be an increase of 20% or less

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