Full Name:_____

CAS CS210 Computer Systems, Fall 2015

Practice Final Exam

Instructions:

- Make sure that your exam is not missing any sheets, then write your full name on the front.
- Write your answers in the space provided below the problem. If you make a mess, clearly indicate your final answer.
- Do your rough work in a work book provided. You do not need to hand in your work books.
- You may find it convient to unstaple the exam. However please restaple in the correct order before handing it in.
- You may use your 1 page of notes that you brought with you.
- The exam has a maximum score of 57 points.
- You have 120 minutes to answer all questions.

  Good luck!

1 (18):	
2 (9):	
3 (6):	
4 (6):	
5 (12):	
6 (6):	
TOTAL (57):	

Problem 1: 18 Points

The following problem concerns the way virtual addresses are translated into physical addresses.

- The memory is byte addressable.
- Memory accesses are to **1-byte words** (not 4-byte words).
- Virtual addresses are 16 bits wide.
- Physical addresses are 14 bits wide.
- The page size is 1024 bytes.
- The TLB is 4-way set associative with 16 total entries.

In the following tables, **all numbers are given in hexadecimal**. The contents of the TLB and the page table for the first 32 pages are as follows:

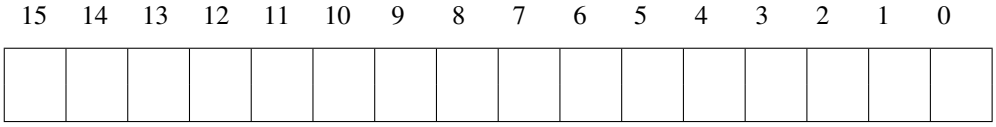
TLB			
Index	Tag	PPN	Valid
0	8	7	1
	F	6	1
	0	3	0
	1	F	1
1	1	E	1
	2	7	0
	7	3	0
	B	1	1
2	0	0	0
	C	1	0
	F	8	1
	7	6	1
3	8	4	0
	3	5	0
	0	D	1
	2	9	0

Page Table					
VPN	PPN	Valid	VPN	PPN	Valid
00	2	0	10	1	1
01	5	1	11	3	0
02	7	1	12	9	0
03	9	0	13	7	1
04	F	1	14	D	1
05	3	1	15	5	0
06	B	0	16	E	1
07	D	1	17	6	0
08	7	1	18	1	0
09	C	0	19	0	1
0A	3	0	1A	8	1
0B	1	1	1B	C	0
0C	0	1	1C	0	0
0D	D	0	1D	2	1
0E	0	0	1E	7	0
0F	1	0	1F	3	0

Part 1

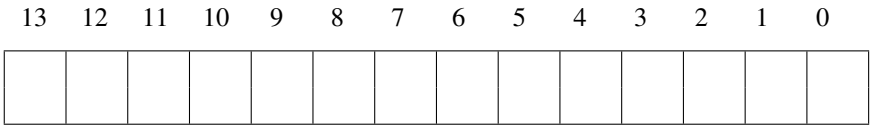
A. The box below shows the format of a virtual address. Indicate (by labeling the diagram) the fields (if they exist) that would be used to determine the following: (If a field doesn't exist, don't draw it on the diagram.)

VPO The virtual page offset
VPN The virtual page number
TLBI The TLB index
TLBT The TLB tag



B. The box below shows the format of a physical address. Indicate (by labeling the diagram) the fields that would be used to determine the following:

PPO The physical page offset
PPN The physical page number



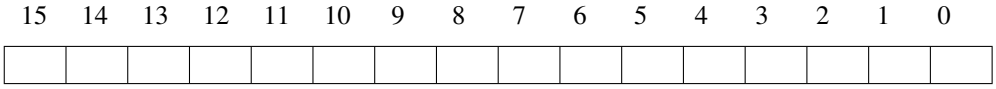
Part 2

For the given virtual addresses, indicate the TLB entry accessed and the physical address. Indicate whether the TLB misses and whether a page fault occurs.

If there is a page fault, enter "-" for "PPN" and leave part C blank.

Virtual address: `2F09`

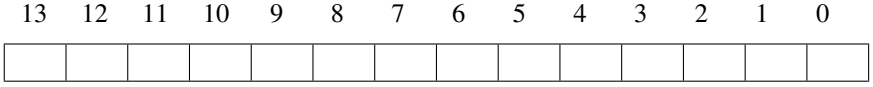
A. Virtual address format (one bit per box)



B. Address translation

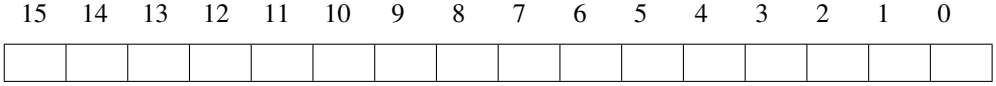
Parameter	Value
VPN	0x
TLB Index	0x
TLB Tag	0x
TLB Hit? (Y/N)	
Page Fault? (Y/N)	
PPN	0x

C. Physical address format (one bit per box)



Virtual address: `0C53`

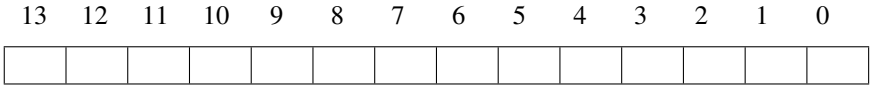
A. Virtual address format (one bit per box)



B. Address translation

Parameter	Value
VPN	0x
TLB Index	0x
TLB Tag	0x
TLB Hit? (Y/N)	
Page Fault? (Y/N)	
PPN	0x

C. Physical address format (one bit per box)



Problem 2: 9 Points

The following problem concerns basic cache lookups.

- The memory is byte addressable.
- Memory accesses are to **1-byte words** (not 4-byte words).
- Physical addresses are 12 bits wide.
- The cache is 4-way set associative, with a 2-byte block size and 32 total lines.

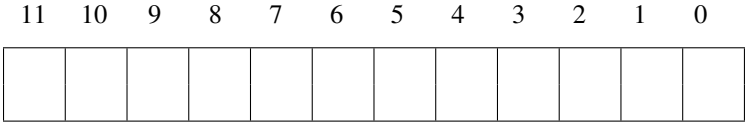
In the following tables, **all numbers are given in hexadecimal**. The contents of the cache are as follows:

4-way Set Associative Cache																
Index	Tag	Valid	Byte 0	Byte 1	Tag	Valid	Byte 0	Byte 1	Tag	Valid	Byte 0	Byte 1	Tag	Valid	Byte 0	Byte 1
0	29	0	34	29	87	0	39	AE	7D	1	68	F2	8B	1	64	38
1	F3	1	0D	8F	3D	1	0C	3A	4A	1	A4	DB	D9	1	A5	3C
2	A7	1	E2	04	AB	1	D2	04	E3	0	3C	A4	01	0	EE	05
3	3B	0	AC	1F	E0	0	B5	70	3B	1	66	95	37	1	49	F3
4	80	1	60	35	2B	0	19	57	49	1	8D	0E	00	0	70	AB
5	EA	1	B4	17	CC	1	67	DB	8A	0	DE	AA	18	1	2C	D3
6	1C	0	3F	A4	01	0	3A	C1	F0	0	20	13	7F	1	DF	05
7	0F	0	00	FF	AF	1	B1	5F	99	0	AC	96	3A	1	22	79

Part 1

The box below shows the format of a physical address. Indicate (by labeling the diagram) the fields that would be used to determine the following:

CO The block offset within the cache line
CI The cache index
CT The cache tag



Part 2

For the given physical address, indicate the cache entry accessed and the cache byte value returned **in hex**. Indicate whether a cache miss occurs.

If there is a cache miss, enter "-" for "Cache Byte returned".

Physical address: `3B6`

A. Physical address format (one bit per box)

11	10	9	8	7	6	5	4	3	2	1	0

B. Physical memory reference

Parameter	Value
Cache Offset (CO)	0x
Cache Index (CI)	0x
Cache Tag (CT)	0x
Cache Hit? (Y/N)	
Cache Byte returned	0x

Problem 3: 6 Points

Consider a direct mapped cache of size 64K with block size of 16 bytes. Furthermore, the cache is write-back and write-allocate. You will calculate the miss rate for the following code using this cache. Remember that `sizeof(int) == 4`. Assume that the cache starts empty and that local variables and computations take place completely within the registers and do not spill onto the stack.

Now consider the following code to copy one matrix to another. Assume that the `src` matrix starts at address 0 and that the `dest` matrix follows immediately follows it.

```
void copy_matrix(int dest[ROWS][COLS], int src[ROWS][COLS])
{
    int i, j;

    for (i=0; i<ROWS; i++) {
        for (j=0; j<COLS; j++) {
            dest[i][j] = src[i][j];
        }
    }
}
```

1. What is the cache miss rate if `ROWS = 128` and `COLS = 128`?
   Miss rate = _____%
2. What is the cache miss rate if `ROWS = 128` and `COLS = 192`?
   Miss rate = _____%
3. What is the cache miss rate if `ROWS = 128` and `COLS = 256`?
   Miss rate = _____%

Problem 4: 6 Points

BURISCV disassembly of memory

```
0000000000010000       ADDI    a3,zero,000000000000de00
0000000000010004       JAL     ra,0000000000000024
0000000000010008       ADD     a3,a3,t6
000000000001000c       SBREAK
....
0000000000010028       ADDI    a3,zero,00000000000000ac
000000000001002c       ADDI    t5,zero,0000000000000001
0000000000010030       ADDI    a3,t5,00000000000000ac
0000000000010034       JALR    zero,ra,0000000000000000
```

Initial Register state

```
x0(zero):0000000000000000 x16(a6):0000000000000000
x1(ra):0000000000000000   x17(a7):0000000000000000
x2(sp):0000000000000000   x18(s2):0000000000000000
x3(gp):0000000000000000   x19(s3):0000000000000000
x4(tp):0000000000000000   x20(s4):0000000000000000
x5(t0):0000000000000000   x21(s5):0000000000000000
x6(t1):0000000000000000   x22(s6):0000000000000000
x7(t2):0000000000000000   x23(s7):0000000000000000
x8(s0):0000000000000000   x24(s8):0000000000000000
x9(s1):0000000000000000   x25(s9):0000000000000000
x10(a0):0000000000000000  x26(s10):0000000000000000
x11(a1):0000000000000000  x27(s11):0000000000000000
x12(a2):0000000000000000  x28(t3):0000000000000000
x13(a3):0000000000000000  x29(t4):0000000000000000
x14(a4):0000000000000000  x30(t5):0000000000000000
x15(a5):0000000000000000  x31(t6):fffffffffffffffe
pc(pc):0000000000010000
```

Given the disassembly of buriscv memory and the following initial register state on the previous page fill in the final state of the register file below. You may use the extract from the riscv manual on the next page as needed. You may also leave unmodified register blank.

x0		x16	
x1	0000000000010008	x17	
x2		x18	
x3		x19	
x4		x20	
x5		x21	
x6		x22	
x7		x23	
x8		x24	
x9		x25	
x10		x26	
x11		x27	
x12		x28	
x13	00000000000000ab	x29	
x14		x30	0000000000000001
x15		x31	
pc			

31 27	26 25	24 20	19 15	14 12	11 7	6 0	
funct7		rs2	rs1	funct3	rd	opcode	R-type
imm[11:0]			rs1	funct3	rd	opcode	I-type
imm[11:5]		rs2	rs1	funct3	imm[4:0]	opcode	S-type
imm[12|10:5]		rs2	rs1	funct3	imm[4:1|11]	opcode	SB-type
imm[31:12]					rd	opcode	U-type
imm[20|10:1|11|19:12]					rd	opcode	UJ-type

RV32I Base Instruction Set

imm[31:12]					rd	0110111	LUI rd,imm
imm[31:12]					rd	0010111	AUIPC rd,imm
imm[20|10:1|11|19:12]					rd	1101111	JAL rd,imm
imm[11:0]			rs1	000	rd	1100111	JALR rd,rs1,imm
imm[12|10:5]		rs2	rs1	000	imm[4:1|11]	1100011	BEQ rs1,rs2,imm
imm[12|10:5]		rs2	rs1	001	imm[4:1|11]	1100011	BNE rs1,rs2,imm
imm[12|10:5]		rs2	rs1	100	imm[4:1|11]	1100011	BLT rs1,rs2,imm
imm[12|10:5]		rs2	rs1	101	imm[4:1|11]	1100011	BGE rs1,rs2,imm
imm[12|10:5]		rs2	rs1	110	imm[4:1|11]	1100011	BLTU rs1,rs2,imm
imm[12|10:5]		rs2	rs1	111	imm[4:1|11]	1100011	BGEU rs1,rs2,imm
imm[11:0]			rs1	000	rd	0000011	LB rd,rs1,imm
imm[11:0]			rs1	001	rd	0000011	LH rd,rs1,imm
imm[11:0]			rs1	010	rd	0000011	LW rd,rs1,imm
imm[11:0]			rs1	100	rd	0000011	LBU rd,rs1,imm
imm[11:0]			rs1	101	rd	0000011	LHU rd,rs1,imm
imm[11:5]		rs2	rs1	000	imm[4:0]	0100011	SB rs1,rs2,imm
imm[11:5]		rs2	rs1	001	imm[4:0]	0100011	SH rs1,rs2,imm
imm[11:5]		rs2	rs1	010	imm[4:0]	0100011	SW rs1,rs2,imm
imm[11:0]			rs1	000	rd	0010011	ADDI rd,rs1,imm
imm[11:0]			rs1	010	rd	0010011	SLTI rd,rs1,imm
imm[11:0]			rs1	011	rd	0010011	SLTIU rd,rs1,imm
imm[11:0]			rs1	100	rd	0010011	XORI rd,rs1,imm
imm[11:0]			rs1	110	rd	0010011	ORI rd,rs1,imm
imm[11:0]			rs1	111	rd	0010011	ANDI rd,rs1,imm
0000000		shamt	rs1	001	rd	0010011	SLLI rd,rs1,shamt
0000000		shamt	rs1	101	rd	0010011	SRLI rd,rs1,shamt
0100000		shamt	rs1	101	rd	0010011	SRAI rd,rs1,shamt
0000000		rs2	rs1	000	rd	0110011	ADD rd,rs1,rs2
0100000		rs2	rs1	000	rd	0110011	SUB rd,rs1,rs2
0000000		rs2	rs1	001	rd	0110011	SLL rd,rs1,rs2
0000000		rs2	rs1	010	rd	0110011	SLT rd,rs1,rs2
0000000		rs2	rs1	011	rd	0110011	SLTU rd,rs1,rs2
0000000		rs2	rs1	100	rd	0110011	XOR rd,rs1,rs2
0000000		rs2	rs1	101	rd	0110011	SRL rd,rs1,rs2
0100000		rs2	rs1	101	rd	0110011	SRA rd,rs1,rs2
0000000		rs2	rs1	110	rd	0110011	OR rd,rs1,rs2
0000000		rs2	rs1	111	rd	0110011	AND rd,rs1,rs2
0000	pred	succ	00000	000	00000	0001111	FENCE
0000	0000	0000	00000	001	00000	0001111	FENCE.I
000000000000			00000	000	00000	1110011	SCALL
000000000001			00000	000	00000	1110011	SBREAK
110000000000			00000	010	rd	1110011	RDCYCLE rd
110010000000			00000	010	rd	1110011	RDCYCLEH rd
110000000001			00000	010	rd	1110011	RDTIME rd
110010000001			00000	010	rd	1110011	RDTIMEH rd
110000000010			00000	010	rd	1110011	RDINSTRET rd
110010000010			00000	010	rd	1110011	RDINSTRETH rd

Problem 5: 12 Points

Please answer each of the following questions. Be clear, concise and complete.

A. (a) What is 2's complement?

(b) Given a w bit word size what is the largest and smallest 2's complement numbers that can be represented?

(c) What is the significance of the value corresponding to all w bits being 1 in 2's complement?

B. (a) What is endianness?

(b) What is signed extension?

(c) What is overflow with respect to data representation?

C. (a) Where are local variables located?

(b) What is the value of an uninitialized local variable?

(c) What determines when the memory of a local variable is no longer considered valid?

D. (a) What is a spatial locality?

(b) What is code motion?

(c) What is the purpose of the Unix fork call?

Problem 6: 6 Points

The following table gives the parameters for a number of different caches, where m is the number of physical address bits, C is the cache size (number of data bytes), B is the block size in bytes, and E is the number of lines per set. For each cache, determine the number of cache sets (S), tag bits (t), set index bits (s), and block offset bits (b).

Cache	m	C	B	E	S	t	s	b
1.	32	256	2	2				
2.	24	1024	4	64				
3.	32	512	8	1				
4.	16	1024	8	8				
5.	64	4096	128	4				
6.	44	4096	64	32				