Last Name: _____ First Name: _____ CruzID: _____

Final Exam

CMPE 012: Computer Systems and Assembly Language

University of California, Santa Cruz

Summer 2018

This exam is closed book and closed notes. Calculators are not permitted. Answers must be marked on the Scantron form to be graded.

Keep your student or government issued ID on your desk. Brimmed hats must be removed or turned around backwards. Only unmarked water bottles are permitted. Backpacks must be placed at the front of the room. Your cell phone must be on a setting where it will not make noise or vibrate.

All questions are multiple choice. Assume the first answer is 'A', the second is 'B' and so on. Some questions have more than one correct answer. You must mark all correct answers to receive credit for a question. Some questions refer to figures that are displayed on the supplemental sheet.

You have 90 minutes to complete this exam.

Note

An underscore followed by a number indicates the base system. Unless otherwise noted, a number followed by no underscore is in decimal.

Final Exam

1 Which label will the following MIPS code branch or jump to? 0 points

```
li   $t0 -10
sra  $t0 $t0  1
ble  $t0 -10  label1
bge  $t0  10  label2
beq  $t0 -10  label3
bne  $t0 -5   label4
j    label5
```

- ◯ label1
- ◯ label2
- ◯ label3
- ◯ label4
- ◯ label5

2 Which registers in MIPS contain the return values after calling a proper function? 0 points

- ◯ $t0-$t9
- ◯ $a0-$a3
- ◯ $v0-$v1
- ◯ $s0-$s7
- ◯ $k0-$k1

3 Assume the following register contents: $s5 = 0xAAAA0000, $s6 = 0x12345678. What is the value of $s7 for the following sequence of instructions? 0 points

```
srl  $s7 $s5 4
and  $s7 $s7 $s6
```

- ◯ 0x12340228
- ◯ 0x00000228
- ◯ 0x12200000
- ◯ 0x02200000
- ◯ None of the other answers

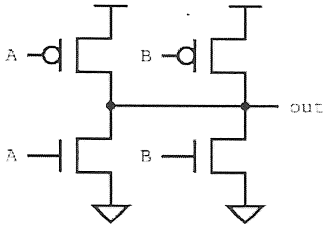
FORM A

4 Which label will the following MIPS code branch or jump to? 0 points

```
li   $t0  10
sra  $t0 $t0  1
ble  $t0   1  label1
bge  $t0  10  label2
beq  $t0  -5  label3
bne  $t0  -5  label4
j    label5
```

- ◯ label1
- ◯ label2
- ◯ label3
- ◯ label4
- ◯ label5

5 What is wrong with the following CMOS circuit? 0 points



- ◯ If A and B are 1, the output will be unknown.
- ◯ If A and B are 1, the power supply will be shorted to ground.
- ◯ None of the other answers.
- ◯ If A and B are 0, the power supply will be shorted to ground.
- ◯ If A is 0 and B is 1, the power supply will be shorted to ground.

6 Assume $t0 is initialized to 0. What is the final value in $t0? 0 points

```
        xor   $s0 $s0 $s0
        addi  $s1 $s0  5
        li    $s2  10
Loop:   ble   $s0 $s1  L1
        addi  $t0 $t0  5
L1:     addi  $t0 $t0  1
        addi  $s0 $s0  1
        bne   $s0 $s2  Loop
```

- ◯ 15
- ◯ 26
- ◯ 30
- ◯ 35
- ◯ None of the other answers.

7 A portion of memory is illustrated below. What is printed to the console after the following instructions? 0 points

```
ADDRESS     VALUE (+0)  VALUE (+4)  VALUE (+8)
0x10010000  0x44434241  0x48474645  0x00000000
```

```
li  $t0  0x10010000
lb  $a0  4($t0)
li  $v0  11
syscall              # print character
```

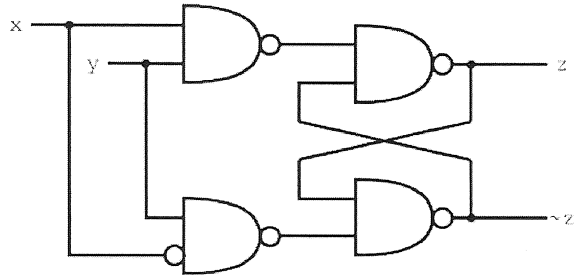
- ◯ H
- ◯ EFGH
- ◯ D
- ◯ None of the other answers
- ◯ E

8 Processing an instruction requires the following steps: write back result (W), execute operation (X), decode instruction (D), memory read / write (M), fetch instruction (F). What is the correct ordering for the steps? 0 points

- ◯ FDXMW
- ◯ DFXWM
- ◯ DFXMW
- ◯ FDMXW
- ◯ FDXWM

9

What storage element does this entire circuit represent? 0 points



- ◯ edge-triggered D flip flop
- ◯ S-R latch, active high
- ◯ level-triggered D latch
- ◯ S-R latch, active low
- ◯ D-R latch

10

Which registers in MIPS must be preserved when writing a proper function? 0 points

- ◯ $t0-$t9
- ◯ $a0-$a3
- ◯ $v0-$v1
- ◯ $s0-$s7

11

What is Rebecca's rabbit's name? 0 points

- ◯ Nybble
- ◯ Bit
- ◯ Byte
- ◯ Flux
- ◯ Minus

12

Which IEEE 754 Single Precision floating point numbers are additive inverses of each other? (Select two) 0 points

- ☐ 0xC1200000
- ☐ 0x37700000
- ☐ 0x0000000A
- ☐ 0x37DFFFFF
- ☐ 0x41200000

13 A portion of data memory is illustrated below. Assuming little endian memory storage, what is in $t7 after the following instructions? 0 points

```
ADDRESS      DATA    |   addi $t0 $zero 0x10011010
0x10011014   0x90    |   lh   $t7 2($t0)
0x10011013   0x78    |   sw   $t7  ($t0)
0x10011012   0x56    |   lw   $t7  ($t0)
0x10011011   0x34    |
0x10011010   0x12    |
```

- ◯ Undefined. There will be an alignment error.
- ◯ 0x00008765
- ◯ 0x00007856
- ◯ 0x78560000
- ◯ 0xFFFF8765

14 The following expression is in post-fix notation. What is the expression in in-fix notation? 0 points

z y + w v * t + *

- ◯ + * + z y * w v t
- ◯ * + t * v w + y z
- ◯ (z + y) * (w * v) + t
- ◯ None of the other answers.
- ◯ (z + y) * ((w * v) + t)

15 If $t0 = 0x10, which of the following answers are equivalent to the value in $t0? Select all that apply. 0 points

- ☐ 10
- ☐ 0x10
- ☐ 00010000_2
- ☐ 16

16 Assume $t0 = 0x10010001. Which of the following instructions would generate a memory alignment error? (Select all that apply) 0 points

- ☐ sw $t5 ($t0)
- ☐ sw $t5 4($t0)
- ☐ sw $t5 1($t0)
- ☐ sw $t5 2($t0)
- ☐ sw $t5 3($t0)

17 Assume the following register contents: $s5 = 0x0000AAAA, $s6 = 0x12345678 What is the value of $s7 for the following sequence of instructions? 0 points

```
sll  $s7 $s5 4
or   $s7 $s7 $s6
```

- ◯ 0x123EFEF8
- ◯ 0xBBBE5678
- ◯ 0x00000220
- ◯ 0x02200000
- ◯ None of the other answers

18 Convert 2110_4 to octal. Assume all answers are in octal. 0 points

- ◯ 712
- ◯ 123
- ◯ 321
- ◯ 224
- ◯ 422

19 A portion of data memory is illustrated below. Assuming big endian memory storage, what is in $t7 after the following instructions? 0 points

```
ADDRESS      DATA    |   addi $t0 $zero 0x10011010
0x10011014   0x90    |   lh   $t7 2($t0)
0x10011013   0x78    |   sw   $t7  ($t0)
0x10011012   0x56    |   lw   $t7  ($t0)
0x10011011   0x34    |
0x10011010   0x12    |
```

- ◯ 0x00007856
- ◯ 0x00005678
- ◯ 0xFFFF8765
- ◯ 0x78560000
- ◯ Undefined. There will be an alignment error.

20 Convert 10111101_2 to decimal. 0 points

- ◯ 1001
- ◯ 753
- ◯ 368
- ◯ 189
- ◯ 95

21 The von Neumann model contains which of the following components? Select all that apply. 0 points

- [ ] Mill
- [ ] Control Unit
- [ ] Processing Unit
- [ ] Memory
- [ ] Input

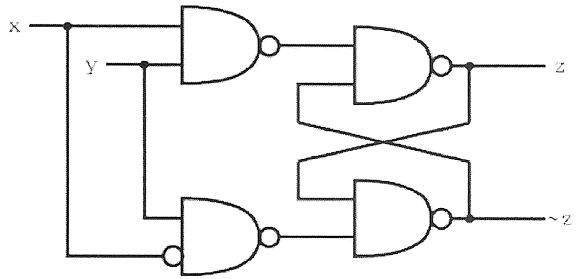
22 Convert 00011011_2 to base 4. Assume all answers are in base 4. 0 points

- ◯ 1100
- ◯ 1023
- ◯ 0123
- ◯ 3322
- ◯ 2111

23 In Memory Mapped IO, peripherals directly modify specific memory addresses. 0 points

- ◯ True
- ◯ False

24 Which timing diagram from Figure 3 corresponds with the circuit below? 0 points



- ◯ A
- ◯ B
- ◯ C
- ◯ D
- ◯ None of the other answers

25 Assume an ISA with 14-bit instructions in the following format. What is the maximum number of instructions this ISA can have? 0 points

opcode	rs	rt	rd
13:9	8:6	5:3	2:0

- ◯ 1
- ◯ 5
- ◯ 16
- ◯ 32
- ◯ 64

26 A portion of memory is illustrated below. Assuming little endian memory storage, what is in $t0 after the following instructions? 0 points

```
ADDRESS      DATA   |  li $t1 0x10011011
0x10011014   0x90   |  lw $t0 ($t1)
0x10011013   0x78   |
0x10011012   0x56   |
0x10011011   0x34   |
0x10011010   0x12   |
```

- ◯ 0x87654321
- ◯ 0x90785634
- ◯ 0x78563412
- ◯ 0x34567890
- ◯ Undefined. There will be an alignment error.

27 What MIPS instruction should be called to go to a subroutine named func? 0 points

- ◯ jr func
- ◯ jal func
- ◯ beq $zero, $zero, func
- ◯ bne $zero, $zero, func

28 Decode the following instruction: 0x20E20010. Select all that apply. 0 points

- ☐ addi $2 $7 16
- ☐ addi $v0 $a3 0x10
- ☐ addi $t2 $t7 0x10
- ☐ addi $v0 $a3 10
- ☐ addi $t2 $t7 16

29

Assume that the register $t6 is initialized to the value 10. What is the final value in $s5 assuming $s5 is initially 0? 0 points

```
Loop: beq   $t6  $zero Done
      subi  $t6  $t6   1
      addi  $s5  $s5   0x10
      j     Loop

Done:
```

- ○ 110
- ○ 100
- ○ 176
- ○ 160
- ○ None of the other answers.

30

Assume $t0 = 0x10010001. Which of the following instructions would generate a memory alignment error? (Select all that apply) 0 points

- ☐ sh $t5 1($t0)
- ☐ sh $t5 3($t0)
- ☐ sh $t5 2($t0)
- ☐ sh $t5 4($t0)
- ☐ sh $t5 ($t0)

31

If a computer has 16-byte addressability and needs 4 bits to access a location in memory, what is the total size of memory in bytes? 0 points

- ○ 256
- ○ 16
- ○ 512
- ○ 64
- ○ 128

32

What would be printed to the console if the following code is run in MARS? 0 points

```
.data

str1: .ascii   "bunny"
str2: .asciiz  "rabbit"
str3: .ascii   "hop"

.text

la $a0 str1
li $v0 4
syscall
```

- ○ bunny rabbit hop
- ○ bunnyrabbithop
- ○ bunny
- ○ bunnyrabbit
- ○ bunny rabbit

33 Translate the following Java statement into MIPS assembly code. Assume that x, y, z, q are stored in registers $s1, $s2, $s3, and $s4, respectively. 0 points

```
x = x + y + z - q
```

```
add $t0 $s1 $s2
add $t0 $t0 $s3
sub $s1 $t0 $s4
```

○ A

```
sub $s3 $s3 $s4
add $s2 $s2 $s3
add $s1 $s2 $s3
```

○ B

```
add $s1 $s1 $s2
sub $s3 $s4 $s3
add $s1 $s3 $s1
```

○ C

```
sub $t0 $s3 $s4
add $t1 $s1 $s2
add $t2 $t0 $t1
```

○ D

```
add $s1 $s1 $s2
sub $s2 $s3 $s4
add $s1 $s2 $s3
```

E

○ E

34 What is the equivalent operation between $t1 and $t2 based on the following MIPS code? 0 points

```
addi  $t3  $t2  0
add   $t2  $t1  $zero
subi  $t1  $t3  0
```

○ AND

○ Initialize to 0

○ Swap

○ Overwrite

35 A portion of memory is illustrated below. What is printed to the console after the following instructions? 0 points

```
ADDRESS      VALUE (+0)   VALUE (+4)
0x10010020   0x52254246   0x00000000
```

```
li  $a0  0x10010020
li  $v0  4
syscall
```

○ FB%R

○ %R$d

○ None of the other answers

○ R%BF

○ d$R%

36 Convert 0x2018 to base 4. Assume all answers are in base 4. 0 points

- ◯ 02000120
- ◯ 22011022
- ◯ 10321122
- ◯ 01230123
- ◯ 22222222

37 Which IEEE 754 Single Precision floating point number is closest to zero? 0 points

- ◯ 0xC1200000
- ◯ 0x442CCCCD
- ◯ 0x385E5D4B
- ◯ 0x2F9FD394
- ◯ 0x782CCCCC

NEXT

Final Exam

Data Path

38

Figure 1 illustrates the MIPS data path. Answer the following questions based on the instruction sw $v0 8($a3). Assume $a3 = 0x10010000 and $v0 = 0xBAADCAFE

What is on wire 8?

0 points

- ◯ 7
- ◯ 0
- ◯ 2
- ◯ 0x10010000
- ◯ 0xBAADCAFE

39

What is on wire 2?

0 points

- ◯ 7
- ◯ 0
- ◯ 2
- ◯ 0x10010000
- ◯ 0xBAADCAFE

40

What is on wire 10?

0 points

- ◯ 0xCAAECAFE
- ◯ 0xBAADCAFE
- ◯ 0x10010000
- ◯ 0x10010008
- ◯ 0xBAADCB06

41

What is on wire 9?

0 points

- ◯ 0x10010008
- ◯ 0x10010000
- ◯ 0x00000008
- ◯ 0xBAADCB06
- ◯ 0xBAADCAFE

BACK NEXT

Final Exam

Register File

42

Figure 2 lists four successive instructions and the state of the $t0-$t3 registers before and after each instruction. The next four questions ask to fill in the omission (indicated by the number <#>) in the instructions and register file.

What is <2>? 0 points

- ◯ LB
- ◯ SH
- ◯ SB
- ◯ LBU
- ◯ LW

43

What is <3>? 0 points

- ◯ SBU
- ◯ LB
- ◯ LBU
- ◯ SB
- ◯ None of the other answers

44

What is <4>? 0 points

- ◯ 0xFFFFFFFE
- ◯ 0x00000000
- ◯ 0x7FFFFFFF
- ◯ 0xFFFFFFFF
- ◯ 0x7FFFFFFE

45

What is <1>? 0 points

- ◯ $t2
- ◯ 4
- ◯ None of the other answers
- ◯ 3
- ◯ $t3

BACK SUBMIT

REG NAME	REG #
$zero	0
$at	1
$v0	2
$v1	3
$a0	4
$a1	5
$a2	6
$a3	7
$t0	8
$t1	9
$t2	10
$t3	11
$t4	12
$t5	13
$t6	14
$t7	15
$s0	16
$s1	17
$s2	18
$s3	19
$s4	20
$s5	21
$s6	22
$s7	23
$t8	24
$t9	25
$k0	26
$k1	27
$gp	28
$sp	29

MNEMONIC	MEANING	TYPE	OPCODE	FUNCT
sll	Logical Shift Left	R	0x00	0x00
srl	Logical Shift Right (0-extended)	R	0x00	0x02
sra	Arithmetic Shift Right (sign-extended)	R	0x00	0x03
jr	Jump to Address in Register	R	0x00	0x08
mfhi	Move from HI Register	R	0x00	0x10
mflo	Move from LO Register	R	0x00	0x12
mult	Multiply	R	0x00	0x18
multu	Unsigned Multiply	R	0x00	0x19
div	Divide	R	0x00	0x1A
divu	Unsigned Divide	R	0x00	0x1B
add	Add	R	0x00	0x20
addu	Add Unsigned	R	0x00	0x21
sub	Subtract	R	0x00	0x22
subu	Unsigned Subtract	R	0x00	0x23
and	Bitwise AND	R	0x00	0x24
or	Bitwise OR	R	0x00	0x25
xor	Bitwise XOR (Exclusive-OR)	R	0x00	0x26
nor	Bitwise NOR (NOT-OR)	R	0x00	0x27
slt	Set to 1 if Less Than	R	0x00	0x2A
sltu	Set to 1 if Less Than Unsigned	R	0x00	0x2B
j	Jump to Address	J	0x02	NA
jal	Jump and Link	J	0x03	NA
beq	Branch if Equal	I	0x04	NA
bne	Branch if Not Equal	I	0x05	NA
blez	Branch if Less Than or Equal to Zero	I	0x06	NA
addi	Add Immediate	I	0x08	NA
addiu	Add Unsigned Immediate	I	0x09	NA
slti	Set to 1 if Less Than Immediate	I	0x0A	NA
sltiu	Set to 1 if Less Than Unsigned Immediate	I	0x0B	NA
andi	Bitwise AND Immediate	I	0x0C	NA
ori	Bitwise OR Immediate	I	0x0D	NA
lui	Load Upper Immediate	I	0x0F	NA
mfc0	Move from Coprocessor 0	R	0x10	NA
lb	Load Byte	I	0x20	NA
lh	Load Halfword	I	0x21	NA
lw	Load Word	I	0x23	NA
lbu	Load Byte Unsigned	I	0x24	NA
lhu	Load Halfword Unsigned	I	0x25	NA
sb	Store Byte	I	0x28	NA
sh	Store Halfword	I	0x29	NA
sw	Store Word	I	0x2B	NA

MNEMONIC	MEANING	TYPE	OPCODE	FUNCT
add	Add	R	0x00	0x20
addi	Add Immediate	I	0x08	NA
addiu	Add Unsigned Immediate	I	0x09	NA
addu	Add Unsigned	R	0x00	0x21
and	Bitwise AND	R	0x00	0x24
andi	Bitwise AND Immediate	I	0x0C	NA
beq	Branch if Equal	I	0x04	NA
blez	Branch if Less Than or Equal to Zero	I	0x06	NA
bne	Branch if Not Equal	I	0x05	NA
div	Divide	R	0x00	0x1A
divu	Unsigned Divide	R	0x00	0x1B
j	Jump to Address	J	0x02	NA
jal	Jump and Link	J	0x03	NA
jr	Jump to Address in Register	R	0x00	0x08
lb	Load Byte	I	0x20	NA
lbu	Load Byte Unsigned	I	0x24	NA
lh	Load Halfword	I	0x21	NA
lhu	Load Halfword Unsigned	I	0x25	NA
lui	Load Upper Immediate	I	0x0F	NA
lw	Load Word	I	0x23	NA
mfc0	Move from Coprocessor 0	R	0x10	NA
mfhi	Move from HI Register	R	0x00	0x10
mflo	Move from LO Register	R	0x00	0x12
mult	Multiply	R	0x00	0x18
multu	Unsigned Multiply	R	0x00	0x19
nor	Bitwise NOR (NOT-OR)	R	0x00	0x27
or	Bitwise OR	R	0x00	0x25
ori	Bitwise OR Immediate	I	0x0D	NA
sb	Store Byte	I	0x28	NA
sh	Store Halfword	I	0x29	NA
sll	Logical Shift Left	R	0x00	0x00
slt	Set to 1 if Less Than	R	0x00	0x2A
slti	Set to 1 if Less Than Immediate	I	0x0A	NA
sltiu	Set to 1 if Less Than Unsigned Immediate	I	0x0B	NA
sltu	Set to 1 if Less Than Unsigned	R	0x00	0x2B
sra	Arithmetic Shift Right (sign-extended)	R	0x00	0x03
srl	Logical Shift Right (0-extended)	R	0x00	0x02
sub	Subtract	R	0x00	0x22
subu	Unsigned Subtract	R	0x00	0x23
sw	Store Word	I	0x2B	NA
xor	Bitwise XOR (Exclusive-OR)	R	0x00	0x26

ASCII CODE						ASCII CODE				
BIN	OCT	DEC	HEX	CHARACTER		BIN	OCT	DEC	HEX	CHARACTER
010 0000	40	32	20	space		100 1110	116	78	4E	N
010 0001	41	33	21	!		100 1111	117	79	4F	O
010 0010	42	34	22	"		101 0000	120	80	50	P
010 0011	43	35	23	#		101 0001	121	81	51	Q
010 0100	44	36	24	$		101 0010	122	82	52	R
010 0101	45	37	25	%		101 0011	123	83	53	S
010 0110	46	38	26	&		101 0100	124	84	54	T
010 0111	47	39	27	'		101 0101	125	85	55	U
010 1000	50	40	28	(		101 0110	126	86	56	V
010 1001	51	41	29	)		101 0111	127	87	57	W
010 1010	52	42	2A	*		101 1000	130	88	58	X
010 1011	53	43	2B	+		101 1001	131	89	59	Y
010 1100	54	44	2C	,		101 1010	132	90	5A	Z
010 1101	55	45	2D	-		101 1011	133	91	5B	[
010 1110	56	46	2E	.		101 1100	134	92	5C	\
010 1111	57	47	2F	/		101 1101	135	93	5D	]
011 0000	60	48	30	0		101 1110	136	94	5E	^
011 0001	61	49	31	1		101 1111	137	95	5F	_
011 0010	62	50	32	2		110 0000	140	96	60	`
011 0011	63	51	33	3		110 0001	141	97	61	a
011 0100	64	52	34	4		110 0010	142	98	62	b
011 0101	65	53	35	5		110 0011	143	99	63	c
011 0110	66	54	36	6		110 0100	144	100	64	d
011 0111	67	55	37	7		110 0101	145	101	65	e
011 1000	70	56	38	8		110 0110	146	102	66	f
011 1001	71	57	39	9		110 0111	147	103	67	g
011 1010	72	58	3A	:		110 1000	150	104	68	h
011 1011	73	59	3B	;		110 1001	151	105	69	i
011 1100	74	60	3C	<		110 1010	152	106	6A	j
011 1101	75	61	3D	=		110 1011	153	107	6B	k
011 1110	76	62	3E	>		110 1100	154	108	6C	l
011 1111	77	63	3F	?		110 1101	155	109	6D	m
100 0000	100	64	40	@		110 1110	156	110	6E	n
100 0001	101	65	41	A		110 1111	157	111	6F	o
100 0010	102	66	42	B		111 0000	160	112	70	p
100 0011	103	67	43	C		111 0001	161	113	71	q
100 0100	104	68	44	D		111 0010	162	114	72	r
100 0101	105	69	45	E		111 0011	163	115	73	s
100 0110	106	70	46	F		111 0100	164	116	74	t
100 0111	107	71	47	G		111 0101	165	117	75	u
100 1000	110	72	48	H		111 0110	166	118	76	v
100 1001	111	73	49	I		111 0111	167	119	77	w
100 1010	112	74	4A	J		111 1000	170	120	78	x
100 1011	113	75	4B	K		111 1001	171	121	79	y
100 1100	114	76	4C	L		111 1010	172	122	7A	z
100 1101	115	77	4D	M						

Figure 1

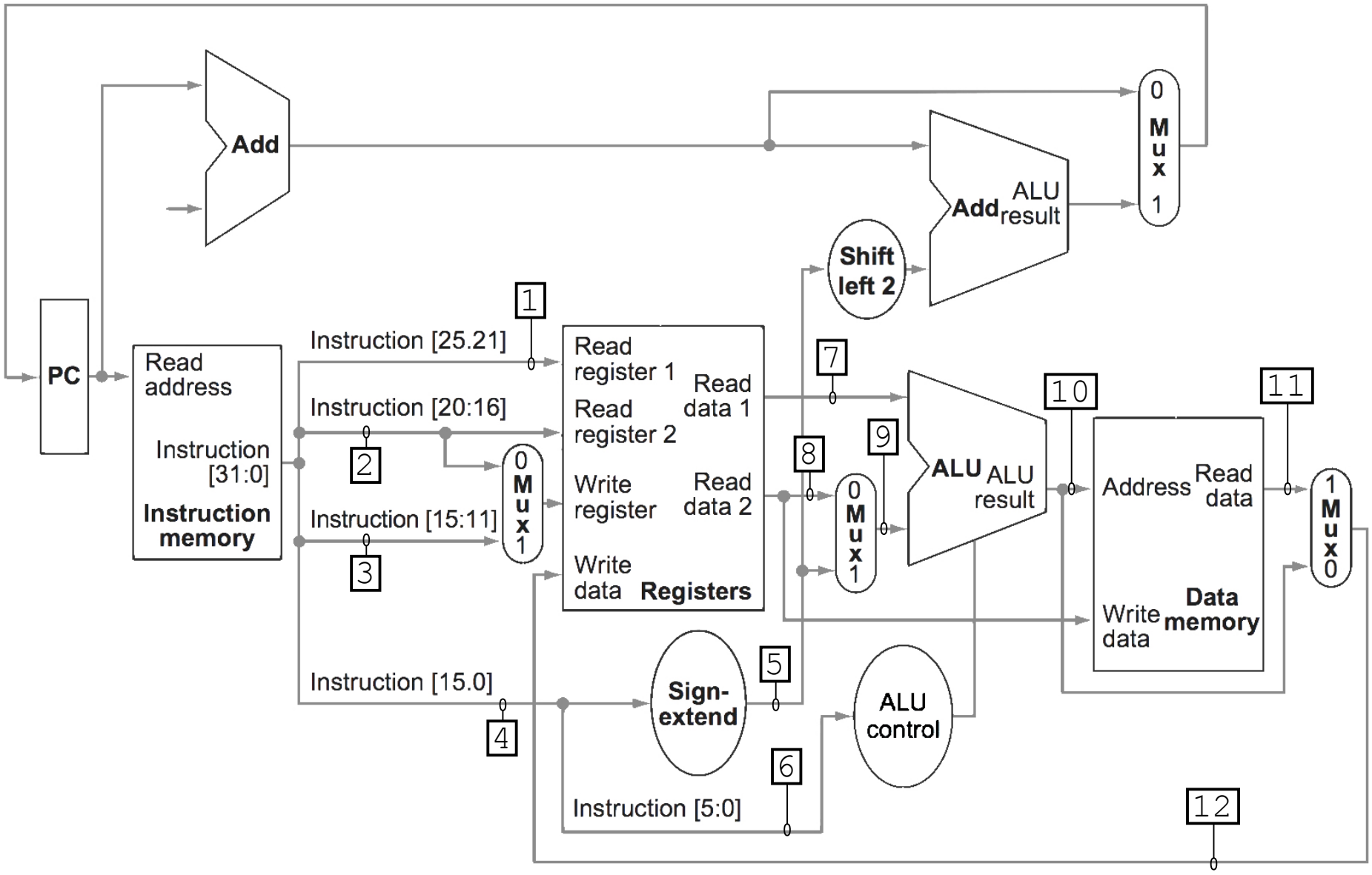

Add
PC
Read address
Instruction [31:0]
Instruction memory
Instruction [25.21]
Instruction [20:16]
Instruction [15:11]
Instruction [15.0]
Instruction [5:0]
Read register 1
Read register 2
Write register
Write data
Read data 1
Read data 2
Registers
0 M u x 1
Shift left 2
ALU Add
ALU result
0 M u x 1
ALU
ALU result
Address
Read data
Write data
Data memory
1 M u x 0
Sign-extend
ALU control
1
2
3
4
5
6
7
8
9
10
11
12

Figure 2

Register File

Reg	Initial	After Inst 1	After Inst 2	After Inst 3	After Inst 4
$t0	0x10010000	0x10010000	0x10010000	0x10010000	0x10010000
$t1	0xC0FFEEEE	0xC0FFEEEE	0xC0FFEEEE	0xC0FFEEEE	0xC0FFEEEE
$t2	0xABCDEF0	0x303FFBBB	0x303FFBBB	0x000000FB	0x000000FB
$t3	0x8000000	0x8000000	0x8000000	0x8000000	<4>

Instructions

```
SRL  $t2   $t1  <1>
<2>  $t2  ($t0)
<3>  $t2 1($t0)
ADDI $t3   $t3  0x7FFFFFFE
```

MIPS Instruction Types

R-type format: instr rd rs rt
(shifts) instr rd rt shamt

opcode	rs	rt	rd	shamt	function
31:26	25:21	20:16	15:11	10:6	5:0

I-type format: inst rt rs immediate
inst rt immediate(rs)

opcode	rs	rt	immediate
31:26	25:21	20:16	15:0

J-type format: j immediate

opcode	immediate
31:26	25:0

Figure 3: Timing Diagrams