Quiz 1: Ten minutes

1. Please give an example from the MIPS architecture for the design principle, “Simplicity favors regularity.”
2. Convert the instruction below into MIPS machine code:
   sw $t0, 12($t3)
   Register $t0 is register number 8.

Instruction Syntax

Encoding	Syntax	Template
Register	ArithLog	f $d, $s, $t
	DivMult	f $s, $t
	Shift	f $d, $t, a
	ShiftV	f $d, $t, $s
	JumpR	f $s
	MoveFrom	f $d
	MoveTo	f $s
Immediate	ArithLogI	o $t, $s, i
	LoadI	o $t, immed32
	Branch	o $s, $t, label
	BranchZ	o $s, label
	LoadStore	o $t, i($s)
Jump	Jump	o label
	Trap	o i

Instruction Encoding Formats

Register	000000ss ssstttttt dddddaaa aafffffff
Immediate	oooooooss ssstttttt iiiiiiii iiiiiiii
Jump	oooooooii iiiiiiii iiiiiiii iiiiiiii

Memory Instructions

Instruction	Operation	Opcode or Function	Syntax	Comments
lb $t, i($s)	$t = MEM[$s + i]	100000	LoadStore	Sign-extends the loaded byte
lbu $t, i($s)	$t = MEM[$s + i]	100100	LoadStore	Zero-extends the loaded byte
lh $t, i($s)	$t = MEM[$s + i]	100001	LoadStore	Sign-extends the loaded bytes
lhu $t, i($s)	$t = MEM[$s + i]	100101	LoadStore	Zero-extends the loaded bytes
lw $t, i($s)	$t = MEM[$s + i]	100011	LoadStore	
sb $t, i($s)	MEM[$s + i] = $t	101000	LoadStore	Lowest order byte is stored
sh $t, i($s)	MEM[$s + i] = $t	101001	LoadStore	2 lowest order bytes are stored
sw $t, i($s)	MEM[$s + i] = $t	101011	LoadStore	

Quiz 1: Ten minutes

1. Please give an example from the MIPS architecture for the design principle, “Good design demands good compromises.”
2. Convert the instruction below into MIPS machine code:

   andi $t1, $t3, -7

   Register $t0 is register number 8.

Instruction Syntax

Encoding	Syntax	Template
Register	ArithLog	f $d, $s, $t
	DivMult	f $s, $t
	Shift	f $d, $t, a
	ShiftV	f $d, $t, $s
	JumpR	f $s
	MoveFrom	f $d
	MoveTo	f $s
Immediate	ArithLogI	o $t, $s, i
	LoadI	o $t, immed32
	Branch	o $s, $t, label
	BranchZ	o $s, label
	LoadStore	o $t, i($s)
Jump	Jump	o label
	Trap	o i

Instruction Encoding Formats

Register	000000ss sssttttt dddddaaa aaffffff
Immediate	ooooooss sssttttt iiiiiiii iiiiiiii
Jump	ooooooii iiiiiiii iiiiiiii iiiiiiii

Arithmetic and Logical Instructions

Instruction	Operation	Opcode or Function	Syntax	Comments
add $d, $s, $t	$d = $s + $t	100000	ArithLog	
addu $d, $s, $t	$d = $s + $t	100001	ArithLog	
addi $t, $s, i	$t = $s + i	001000	ArithLogI	i is sign-extended
addiu $t, $s, i	$t = $s + i	001001	ArithLogI	i is sign-extended
and $d, $s, $t	$d = $s & $t	100100	ArithLog	
andi $t, $s, i	$t = $s & i	001100	ArithLogI	i is zero-extended

Quiz 1: Ten minutes

1. Please give an example from the MIPS architecture for the design principle, “Make the common case fast.”
2. Convert the instruction below into MIPS machine code:
   lw $t2, 0x20($t3)
   Register $t0 is register number 8.

Instruction Syntax

Encoding	Syntax	Template
Register	ArithLog	f $d, $s, $t
	DivMult	f $s, $t
	Shift	f $d, $t, a
	ShiftV	f $d, $t, $s
	JumpR	f $s
	MoveFrom	f $d
	MoveTo	f $s
Immediate	ArithLogI	o $t, $s, i
	LoadI	o $t, immed32
	Branch	o $s, $t, label
	BranchZ	o $s, label
	LoadStore	o $t, i($s)
Jump	Jump	o label
	Trap	o i

Instruction Encoding Formats

Register	000000ss sssttttt dddddaaa aaffffff
Immediate	ooooooss sssttttt iiiiiiii iiiiiiii
Jump	oooooooi iiiiiiii iiiiiiii iiiiiiii

Memory Instructions

Instruction	Operation	Opcode or Function	Syntax	Comments
lb $t, i($s)	$t = MEM[$s + i]	100000	LoadStore	Sign-extends the loaded byte
lbu $t, i($s)	$t = MEM[$s + i]	100100	LoadStore	Zero-extends the loaded byte
lh $t, i($s)	$t = MEM[$s + i]	100001	LoadStore	Sign-extends the loaded bytes
lhu $t, i($s)	$t = MEM[$s + i]	100101	LoadStore	Zero-extends the loaded bytes
lw $t, i($s)	$t = MEM[$s + i]	100011	LoadStore	
sb $t, i($s)	MEM[$s + i] = $t	101000	LoadStore	Lowest order byte is stored
sh $t, i($s)	MEM[$s + i] = $t	101001	LoadStore	2 lowest order bytes are stored
sw $t, i($s)	MEM[$s + i] = $t	101011	LoadStore	