Homework 02 (35 Points) – Due Sep 26th 2020

Rubric: 25% Identify instruction format, 50% for step by step process, 25% for the correct answer,

3 points for neatness.

Source used <https://inst.eecs.berkeley.edu/~cs61c/resources/MIPS_help.html>

1. Translate the given four instructions into its 32-bit format and Hex format. (4 points each)
   1. sub $t0, $s1, $s2

Instruction type:

R type : I assume *register* type.

Instruction format:

An R type comes in the format

31-26 bits opcode “tells the ALU what circuit to run Via multiplexer selection bits”

25-21 bits *r*egister *s*ource: first operand “$RS”

20-16 bits *r*egister transfer: second operand “$RT”

15-11 bits *r*egister *d*estination: output register “$RD”

10-6 bits “shamt”: shift register amount // only used in bit shift commands

5-0 bits “funct” extra selection bits for R type “register type” instructions

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |

32 bit Binary:

Hex Format:

Some slots are five or six slots but the higher bits in hex are truncated. So, the top 2 in opcode are always zero.

Opcode, Rs, Rt, Rd, shamt, funct

0x00, 0x11, 0x12,0x08 0x0,0x022

1. and $t0, $s1, $t1

Instruction type:

R type : *register* type.

Instruction format:

An R type comes in the format

31-26 bits opcode “tells the ALU what circuit to run Via multiplexer selection bits”

25-21 bits *r*egister *s*ource: first operand “$RS”

20-16 bits *r*egister transfer: second operand “$RT”

15-11 bits *r*egister *d*estination: output register “$RD”

10-6 bits “shamt”: shift register amount // only used in bit shift commands

5-0 bits “funct” extra selection bits for R type “register type” instructions

Hex Format:

Some slots are five slots or six but the higher bits in hex are truncated.

Opcode, Rs, Rt, Rd, shamt, funct

and $t0, $s1, $t1

0x20,0x10,0x09,0x08,0x0,0x00

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | | 5 | 4 | | 3 | 2 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | 0 | 0 | 0 | 0 | 0 |

1. From below given four hex values encode the MIPS instruction. (4 points each)
   1. 0x0000A812

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |

32 bits Binary:

No opcode

Function code 18

Move from lo

Assumption no truncating //this cost me 3 hours

No shift

Destination register 16+1+4 = 21 $s5

Instruction format:

R type: Register type

MIPS Instruction:

Mflo $s5

* 1. 0x02286025

32 bits Binary:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |

No opcode

Function code 0x25 or 37 or

Register type

Source register 17

Transfer register 8

Destination register 13

No shamt

Instruction format:

R type: Register type

MIPS Instruction:

Or $t5, $s1, $t0

* 1. 0x00125883

32 bits Binary:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

No opcode

Function code 3 shift right arithmetic: sra

Shamt 2 right shift 2 slots fill with 1’s

Destination register: 11: $t3

Transfer register: 16+2 = 18: $s2

Source register: 0 $zero

Instruction format:

R type: register type

MIPS Instruction:

Sra $t3, $s2, 2

* 1. 0xAE4B0008

32 bits Binary:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

0x2b opcode code sw

Offset 8

Transfer Register: 13: $t5

Source Register: 16+2 = 18: $s2

Instruction format:

I type: instruction type

MIPS Instruction:

Sw $t5,8($s2)