

Organización y Arquitectura de Computadoras

Práctica 9: Ensamblador

Sandra del Mar Soto Corderi
Edgar Quiroz Castañeda

13 de junio del 2019

1. Ejercicios

1. Clasifica las instrucciones de la sección 4.2 según los tipos descritos en 4.3.

a) **lw** rs dir
Es de tipo **I**

b) **lh** rs dir
Es de tipo **I**

c) **lb** rs dir
Es de tipo **I**

d) **sw** rs dir
Es de tipo **I**

e) **sh** rs dir
Es de tipo **I**

f) **sb** rs dir
Es de tipo **I**

g) **add** rd rs rt
Es de tipo **R**

h) **addi** rd rs imm
Es de tipo **I**

i) **sub** rd rs rt
Es de tipo **R**

j) **subi** rd rs imm
Es de tipo **I**

k) **and** rd rs rt
Es de tipo **R**

l) **andi** rd rs imm
Es de tipo **I**

m) **or** rd rs rt
Es de tipo **R**

n) **ori** rd rs imm
Es de tipo **I**

ñ) **beq** rs rt label
Es de tipo **I**

o) **bgt** rs rt label
Es de tipo **I**

p) **j** label
Es de tipo **J**

q) **jr** rd
Es de tipo **J**

2. Asigna cuidadosamente los códigos de operación a cada una de las instrucciones.

a) **lw** rs dir
Su código de operación es 000101

b) **lh** rs dir
Su código de operación es 000110

c) **lb** rs dir
Su código de operación es 000111

d) **sw** rs dir
Su código de operación es 001000

e) **sh** rs dir
Su código de operación es 001001

f) **sb** rs dir
Su código de operación es 001010

g) **add** rd rs rt
Su código de operación es 00000000001

h) **addi** rd rs imm
Su código de operación es 000001

i) **sub** rd rs rt
Su código de operación es 00000000010

j) **subi** rd rs imm
Su código de operación es 000010

k) **and** rd rs rt
Su código de operación es 00000000011

l) **andi** rd rs imm
Su código de operación es 000011

m) **or** rd rs rt
Su código de operación es 00000000100

n) **ori** rd rs imm
Su código de operación es 000100

ñ) **beq** rs rt label
Su código de operación es 000101

o) **bgt** rs rt label
Su código de operación es 000110

p) **j** label
Su código de operación es 001011

q) **jr** rd
Su código de operación es 00110

3. Define el tamaño de la memoria y las direcciones de inicio de las secciones de memoria para el área de texto y el área de datos.

De acuerdo a la documentación de MIPS, los procesadores modernos compatibles con x86 son capaces de direccionar hasta 2^{32} bytes de memoria y las direcciones de memoria son de 32 bits de ancho, así que en este caso le daremos un tamaño estándar de 32 de registros, donde los registros guardados serán para los datos y los registros locales para el área de texto.

2. Preguntas

1. Una pseudoinstrucción es una instrucción de lenguaje ensamblador sin implementación directa en el hardware, su función es simplificar la programación sin complicar el hardware, por ejemplo:

Propón 3 pseudoinstrucciones para el lenguaje ensamblador y su traducción.

a) **blt** \$8, \$9, 4 se traduce a:

slt \$1, \$8, \$9
bne \$1, \$0, 4

b) **abs** \$1, \$2 se traduce a:

addu \$1, \$2, \$0
bgez \$2, 8 (offset=8 \rightarrow operación 'sub')
sub \$1, \$2, \$0

c) **li** \$8, 0x3BF20 se traduce a:

lui \$at, 0x0003
ori \$8, \$at, 0xBF20

2. Sólo se implementaron dos modos de direccionamiento, propón cómo podría simular el ensamblador los siguientes modos

- label \pm imm

Se puede traducir como una instrucción tipo I, donde la fuente es \$00 y el inmediato es

$$(linea_actual - linea_label) \pm imm$$

- label \pm imm(reg)

Se puede traducir como una instrucción tipo I, donde la fuente es reg y el inmediato es

$$(linea_actual - linea_label) \pm imm$$

3. Especificaciones del programa

El programa no funciona con datos estáticos y para ejecutarlo, se necesita ejecutar el archivo automata.c

Referencias

- [1] *MIPS® Architecture For Programmers Volume II-A: The MIPS64® Instruction Set Reference Manual [Online]*. Disponible: <https://s3-eu-west-1.amazonaws.com/downloads-mips/documents/MD00087-2B-MIPS64BIS-AFP-6.06.pdf>. [Consultado: 13-Junio-2019].
- [2] *MIPS Assembly/Pseudoinstructions [Online]*. Disponible: https://en.wikibooks.org/wiki/MIPS_Assembly/Pseudoinstructions. [Consultado: 13-Junio-2019].