**[Básicas]**

nombreLabel: Label. Es para ponerle un nombre a algo y después usarlo.

! Comentario

arrayA: 1 Declarar una etiqueta de un array para cargarlo

2

3

4

ba goto

call llamar a una función (label)

jmpl como un “return”

sethi 0x00000A, %r1 establece los 22 bits más representativos de %r1

a 0x00000A y el resto de los bits se hacen 0

**[Directivas]**

.begin inicio del programa

x .equ #5 asignación de la constante x = 0x5

.org 2048 dirección de inicio del programa

.end fin de lo que se tiene que ensamblar

.dwb 25 reservar bloque sde memoria en bloques de 4 B

.global Y etiqueta global

.extern Z etiqueta externa

.macro M a ba definir un macro

**[Memoria]**

ld [x], %r1 cargar la dirección x en el registro %r1

st %r2, [y] guardar en en la dirección y lo que hay en el registro %r2

stb para 8 bits

ldsb para 8 bits, con signo (el resto completa con 1)

ldub para 8 bits, sin signo (el resto completa con 0)

sth para 16 bits

ldsh para 16 bits, con signo

lduh para 16 bits, sin signo

**[Aritmético Lógicas]**

and and bit a bit

andcc

or or bit a bit

orcc

andn nand

andncc

orn nor

orncc

xor

xnor

xorcc

xnorcc

add %r1, %r2, %r3 sumar (1 y 2 operandos, 3 resultados)

sub %r1, %r2, %r3 restar

addcc %r1, %r2, %r3

subcc %r1, %r2, %r3

sra shift a derecha aritmético (división por 2 con signo)

srl shift a derecha (división por 2)

sll shift a izquierda (multiplicación por 2)

**[Condicionales]**

be if z == 1

bne if z == 0

bneg if n == 1

bpos if n == 0

bcs if c == 1

bcc if c == 0

bvs if v == 1

bvc if v == 0

bg if >

bgu if > (sin signo)

bge if >= (bgeu no existe)

bl if < (blu no existe)

ble if <=

bleu if <= (sin signo)

**[Registros]**

%r0 siempre está en 0 (no puede cambiar, se usa como /dev/null)

%sp %r14, reservado para la cabeza de la pila

%link %r15, reservado para la dirección de retorno de control

**Observaciones**

- Cuando hacemos un load o un store: se hace de a un byte, pero el resultado es de 32 bits (4 bytes)

- Case sensitive

- Números en base 10: por defecto

- Números en base 16: empieza con "0x" o finaliza con "h"

- tiene que tener una línea en blanco al final para que el emulador ejecute

- el u es de unsigned

- el cc quiere decir que impacta los flags

- Directiva: del emulador (empieza con .)

- Set de instrucciones: específico del procesador

%psr es el que tiene los flags [NO ACCESIBLE]

%pc puntero a la dirección de memoria de la instrucción en ejecución [NO ACCESIBLE]

%ir registro de la instrucción en ejecución [NO ACCESIBLE]