

ALDAVERA GALLAGA IVÁN	INSTITUTO POLITÉCNICO NACIONAL	ESCUELA SUPERIOR DE CÓMPUTO
TAREA 05 (SEGUNDO PARCIAL)	DEFINICIÓN DE CONCEPTOS	FECHA DE ENTREGA: <b>15 DE ABRIL DEL 2019</b>
GRUPO (3CM2)	ARQUITECTURA DE COMPUTADORAS	

INCISO	CONCEPTO	DEFINICIÓN
A	<i>Index register</i>	Es el registro en las instrucciones de transferencia de datos se inventó originalmente para mantener un índice de una matriz con el desplazamiento utilizado para la dirección de inicio de una matriz.
B	<i>Load Instruction</i>	El formato de la instrucción de carga es el nombre de la operación seguida por el registro a cargar, luego una constante y registro se usa para memoria de acceso.
C	<i>store instruction</i>	Es una arquitectura del conjunto de instrucciones que divide las instrucciones en dos categorías: acceso a la memoria (carga y almacenamiento entre la memoria y los registros) y operaciones ALU (que solo ocurren entre los registros).
D	<i>alignment restriction</i>	Un requisito en el que los datos están alineados en la memoria en los límites naturales.
E	<i>Explicar la instrucción [ lw \$t0,8(\$s3) ]</i>	<p>Supongamos que A es una matriz de 100 palabras y que el compilador ha asociado las variables <b>g</b> y <b>h</b> con los registros <b>\$s1</b> y <b>\$s2</b> como antes. Supongamos también que la dirección de inicio, o la dirección base, de la matriz está en <b>\$s3</b>. Compila esta sentencia de asignación de C: <b>g = h + A [8];</b></p> <p>Aunque hay una sola operación en esta declaración de asignación, uno de los operandos está en la memoria, por lo que primero debemos transferir <b>A[8]</b> a un registro. La dirección de este elemento de la matriz es la suma de la base de la matriz A, que se encuentra en el registro <b>\$s3</b>, más el número para seleccionar el elemento 8. Los datos deben colocarse en un registro temporal para su uso en la siguiente instrucción: <b>lw \$t0,8(\$s3) # Temporary reg \$t0 gets A[8]</b></p> <p>(Haremos un pequeño ajuste a esta instrucción, pero usaremos esta versión simplificada por ahora.) La siguiente instrucción puede operar en el valor en <b>\$t0</b> (que es igual a <b>A[8]</b>) ya que está en un registro. La instrucción debe agregar <b>h</b> (contenida en <b>\$s2</b>) a <b>A[8]</b> (contenida en <b>\$t0</b>) y poner la suma en el registro correspondiente a <b>g</b> (asociada con <b>\$s1</b>)</p> <p><b>add \$s1,\$s2,\$t0 # g = h + A[8]</b></p>
F	<i>Starting address, base register, and index register</i>	“rs” es el registro base que es sumado a “address” para formar la dirección de memoria. Para una operación “store” “rt” es el registro fuente cuyo valor será almacenado en memoria.
G	<i>Convención para los registros en el</i>	La convención del modelo MIPS es usar nombres de dos caracteres después de un signo de dólar para representar un registro.

	<b>procesador del modelo MIPS</b>	Por ahora, usaremos \$s0, \$s1,... para registros que corresponden a variables en programas C y Java y \$t0, \$t1,... para los registros temporales necesarios para compilar el programa en instrucciones MIPS.
<b>H</b>	<b>machine language</b>	Una representación binaria de instrucciones de la máquina.
	<b>machine code</b>	Una secuencia de instrucciones de lenguaje máquina.
<b>I</b>	<b>Big-Endian</b>	Aquellas computadoras que usan la dirección del byte más a la izquierda o el "extremo más grande" como la word address
	<b>Little-Endian</b>	Aquellas computadoras que usan la dirección del byte más a la derecha o el "extremo más pequeño" como la word address
<b>J</b>	<b>Conditional Branch</b>	Instrucciones de decisión tipo if-equal e if-not equal. Instrucciones que diferencian a una computadora de una calculadora, las cuales se encargan de tomar decisiones.
	<b>beq</b>	branch on equal
	<b>bnq</b>	branch on not equal
	<b>blt</b>	branch on less than
	<b>ble</b>	branch less than or equal
	<b>bgt</b>	branch greater than
	<b>bge</b>	branch greater than or equal