

HERNÁNDEZ ESCOBEDO FERNANDO	INSTITUTO POLITÉCNICO NACIONAL	ESCUELA SUPERIOR DE CÓMPUTO
NUMERO DE LA TAREA 5	DEFINICIÓN DE CONCEPTOS	12 DE ABRIL DEL 2019
GRUPO (3CM3)	ARQUITECTURA DE COMPUTADORAS	

## ➔DEFINICIÓN DE LOS SIGUIENTES CONCEPTOS

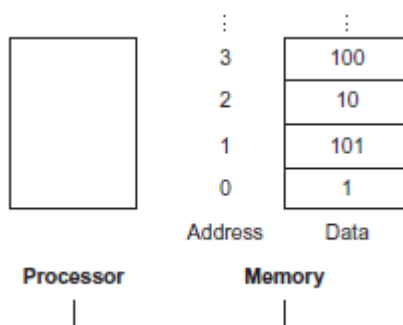
### A).- Index register.

Es un número que actúa como índice de las memorias, que son matrices grandes y unidimensionales con una dirección que actúa como índice de la matriz, comenzando en 0.

Por lo tanto, es;

“Un valor utilizado para delinear la ubicación de un elemento de datos específico dentro de una matriz de memoria.”

Por ejemplo, para acceder a una palabra en la memoria, la instrucción debe proporcionar la dirección de la memoria. Por ejemplo, en la figura siguiente, la dirección del tercer elemento de datos es 2 y el valor de Memoria con índice=2=[2 ] es 10.



El registro en las instrucciones de transferencia de datos se inventó originalmente para contener un índice de una matriz con el desplazamiento utilizado para la dirección de inicio de una matriz. Por lo tanto, el registro base también se llama el registro de índice.

### B).- Load Instruction

Es la instrucción de transferencia de datos que copia los datos de la memoria a un registro.

El formato de la instrucción de carga es el nombre de la operación seguida por el registro a cargar, luego una constante y un registro usado para acceder a la memoria.

El nombre real de MIPS para esta instrucción es lw, que significa palabra de carga.

### C).- Store instruction

La instrucción complementaria a la carga se denomina tradicionalmente store; copia datos de un registro a la memoria.

El formato de un almacenamiento es similar al de una carga: el nombre de la operación, seguido del registro que se almacenará, luego se configura para seleccionar el elemento de la matriz, y finalmente el registro base/índice.

La dirección MIPS se especifica en parte por una constante y en parte por el contenido de un registro. El nombre real de MIPS es sw, que significa store word.

### D).- Alignment restriction

Es un requisito de que los datos estén alineados en la memoria en los límites naturales.

En MIPS, las palabras deben comenzar en direcciones que son múltiplos de 4. Este requisito se denomina restricción de alineación, y muchas arquitecturas lo tienen.

#### E).- Explicar la instrucción [ lw \$t0,8(\$s3) ]

lw \$ t0,8 (\$ s3)

Esta instrucción tiene la tarea de cargar en un registro temporal, en este caso \$t0, al dato(s) que están en el índice 8 de la matriz que tiene como dirección base \$s3. Posteriormente este registro podrá utilizarse en otras instrucciones.

#### F).- starting address, base register, and index register

##### Starting address

Es la dirección de inicio de una matriz, es decir, desde donde comenzará.

\*El registro en las instrucciones de transferencia de datos se inventó originalmente para contener un índice de una matriz con el desplazamiento utilizado para la dirección de inicio de una matriz.

##### Base register

Contiene un puntero a una dirección de memoria, la cual sirve como referencia para ubicar una dirección.

Por ejemplo: el desplazamiento MIPS más el direccionamiento del registro base es una excelente coincidencia con las estructuras y los arreglos, ya que el registro puede apuntar al principio de la estructura y el desplazamiento puede seleccionar el elemento deseado.

##### Index register

Es un valor utilizado para delinear la ubicación de un elemento de datos específico dentro de una matriz de memoria.

#### G).- Conversión para los registros en el procesador del modelo MIPS

Como los registros se mencionan en las instrucciones, debe haber una convención para asignar los nombres de los registros a los números;

En el lenguaje ensamblador MIPS, los registros de \$ s0 a \$ s7 se asignan a los registros 16 a 23, y los registros de \$ t0 a \$ t7 a los registros 8 a 15. Por lo tanto, \$ s0 significa registro 16, \$ s1 significa registro 17, \$ s2 significa registro 18, . . . , \$ t0 significa registro 8, \$ t1 significa registro 9, y así sucesivamente.

#### H).- a).- machine language

Representación binaria utilizada para la comunicación dentro de un sistema informático.

##### b).- machine code

Es un programa de computadora escrito en instrucciones en lenguaje de máquina que pueden ejecutarse directamente por la unidad central de procesamiento (CPU) de una computadora.

I).- El **formato endian** es un atributo de los datos que describe el orden de los bytes. Cuando las aplicaciones intercambian datos, deben conocer el convenio de clasificación para los datos de varios bytes. En caso contrario, los datos podrían malinterpretarse. Los datos pueden tener los formatos de orden de bytes siguientes:

##### a).- Big-Endian

Formato en el que el byte más significativo se almacena en primer lugar. Los demás bytes le siguen en orden de significado descendente. Por ejemplo, en el caso de una palabra de cuatro bytes, el orden de los bytes es 0, 1, 2, 3. En el caso de una palabra de dos bytes, es 0, 1.

Instruction Word 0			
Half Word B		Half Word A	
Byte 0	Byte 1	Byte 2	Byte 3

### Big – endian format

#### b).- Little-Endian

Formato en el que el byte menos significativo se almacena en primer lugar. Los demás bytes le siguen en orden de significado ascendente. Por ejemplo, en el caso de una palabra de cuatro bytes, el orden de los bytes es 3, 2, 1, 0. En el caso de una palabra de dos bytes, es 1, 0.

Instruction Word 0			
Half Word B		Half Word A	
Byte 3	Byte 2	Byte 1	Byte 0
31 to 24	23 to 16	15 to 8	7 to 0

### Little – endian format

**J).- Conditional Branch**, es una instrucción que requiere la comparación de dos valores y que permite una posterior transferencia de control a una nueva dirección en el programa basada en el resultado de la comparación.

Se consideran adecuadamente como la ejecución condicional de la instrucción de branch incondicional.

- a).- beq = branch on equal
- b).- bneq = branch on not equal
- c).- blt = branch less than
- d).- ble = branch less than or equal
- f).- bgt = branch greater than
- g).- bge = branch greater than or equal

#### BIBLIOGRAFÍA.

- Patterson, D.A. & Hennessy, J.L. (2014). *COMPUTER ORGANIZATION AND DESIGN: THE HARDWARE/SOFTWARE INTERFACE*. United States: ELSEVIER.
- Hennessy, John L.; Patterson, David A. *Computer Organization and Design. The Hardware/Software Interface*. Morgan Kaufmann Publishers. ISBN 1-55860-281-X.
- **IBM Knowledge Center** (2019). FORMATO ENDIAN. Recuperado de: [https://www.ibm.com/support/knowledgecenter/es/SSEPEK\\_11.0.0/char/src/tpc/db2z\\_endianness.html](https://www.ibm.com/support/knowledgecenter/es/SSEPEK_11.0.0/char/src/tpc/db2z_endianness.html)