

Formato de Instrucción

Organización del Computador I

~~David Alejandro González Márquez~~

Actual dueño: Maximiliano Geier

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

15.02.2018

Agenda

- Tamaño, Dirección y Unidad direccionable
- Diseño de un Formato de Instrucción
- Codificación de una Instrucción

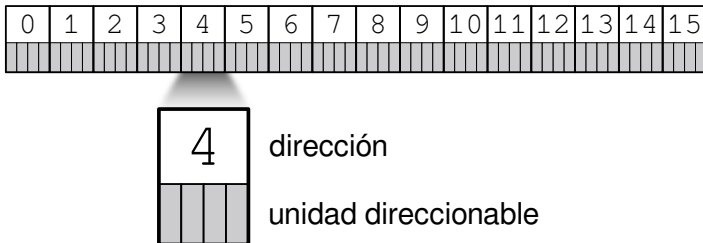
Tamaño, dirección, unidad direccionable

Ejemplos

Ejemplo 1: Memoria de 8 bytes

16 direcciones (4 bits de dirección) y

4 bits de unidad direccionable (1 nibble)



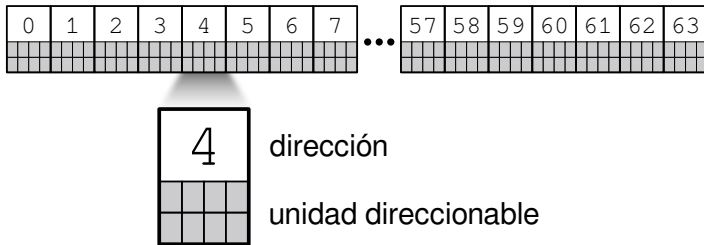
Tamaño, dirección, unidad direccionable

Ejemplos

Ejemplo 2: Memoria de 64 bytes

64 direcciones (6 bits de dirección) y

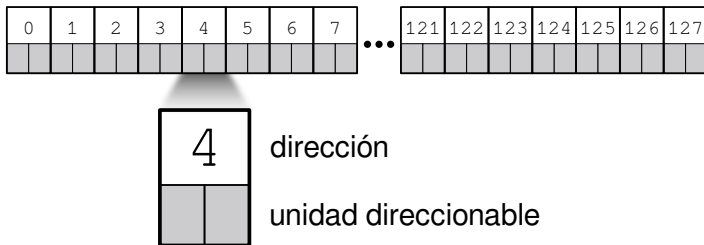
8 bits de unidad direccionable (1 byte)



Tamaño, dirección, unidad direccionable

Ejemplos

Ejemplo 3: Memoria de 32 bytes
128 direcciones (7 bits de dirección) y
2 bits de unidad direccionable



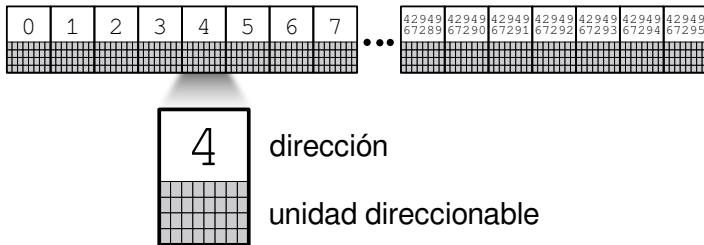
Tamaño, dirección, unidad direccionable

Ejemplos

Ejemplo 4: Memoria de 16 GB

4294967296 direcciones (32 bits de dirección)

32 bits de unidad direccionable (4 bytes)



Tamaño, dirección, unidad direccionable

¿Cuántos bits son necesarios para poder direccionar la memoria?

Tamaño, dirección, unidad direccionable

¿Cuántos bits son necesarios para poder direccionar la memoria?

- 1 Memoria de 2 GB con direccionamiento a *byte*

Tamaño, dirección, unidad direccionable

¿Cuántos bits son necesarios para poder direccionar la memoria?

1 Memoria de 2 GB con direccionamiento a *byte*

$$\blacksquare \# \text{direcciones} = \frac{2 \text{ GB}}{1 \text{ B/dir}} = \frac{2 \times 2^{30} \text{ B}}{1 \text{ B/dir}} = 2^1 \times 2^{30} \text{ dir}$$

Tamaño, dirección, unidad direccionable

¿Cuántos bits son necesarios para poder direccionar la memoria?

1 Memoria de 2 GB con direccionamiento a *byte*

- $\#direcciones = \frac{2 \text{ GB}}{1 \text{ B/dir}} = \frac{2 \times 2^{30} \text{ B}}{1 \text{ B/dir}} = 2^1 \times 2^{30} \text{ dir}$
- $\#bits \text{ de direccionamiento} = \log_2(2^1 \times 2^{30}) = 31 \text{ bits}$

Tamaño, dirección, unidad direccionable

¿Cuántos bits son necesarios para poder direccionar la memoria?

1 Memoria de 2 GB con direccionamiento a *byte*

- $\#direcciones = \frac{2 \text{ GB}}{1 \text{ B/dir}} = \frac{2 \times 2^{30} \text{ B}}{1 \text{ B/dir}} = 2^1 \times 2^{30} \text{ dir}$
- $\#bits \text{ de direccionamiento} = \log_2(2^1 \times 2^{30}) = 31 \text{ bits}$

2 Memoria de 2 GB con direccionamiento a 16 *bits*

Tamaño, dirección, unidad direccionable

¿Cuántos bits son necesarios para poder direccionar la memoria?

1 Memoria de 2 GB con direccionamiento a *byte*

- $\#direcciones = \frac{2 \text{ GB}}{1 \text{ B/dir}} = \frac{2 \times 2^{30} \text{ B}}{1 \text{ B/dir}} = 2^1 \times 2^{30} \text{ dir}$
- $\#bits \text{ de direccionamiento} = \log_2(2^1 \times 2^{30}) = 31 \text{ bits}$

2 Memoria de 2 GB con direccionamiento a 16 *bits*

- $\#direcciones = \frac{2 \text{ GB}}{2 \text{ B/dir}} = 2^{30} \text{ dir}$

Tamaño, dirección, unidad direccionable

¿Cuántos bits son necesarios para poder direccionar la memoria?

1 Memoria de 2 GB con direccionamiento a *byte*

- $\#direcciones = \frac{2 \text{ GB}}{1 \text{ B/dir}} = \frac{2 \times 2^{30} \text{ B}}{1 \text{ B/dir}} = 2^1 \times 2^{30} \text{ dir}$
- $\#bits \text{ de direccionamiento} = \log_2(2^1 \times 2^{30}) = 31 \text{ bits}$

2 Memoria de 2 GB con direccionamiento a 16 *bits*

- $\#direcciones = \frac{2 \text{ GB}}{2 \text{ B/dir}} = 2^{30} \text{ dir}$
- $\#bits \text{ de direccionamiento} = \log_2(2^{30}) = 30 \text{ bits}$

Tamaño, dirección, unidad direccionable

¿Cuántos bits son necesarios para poder direccionar la memoria?

1 Memoria de 2 GB con direccionamiento a *byte*

- $\#direcciones = \frac{2 \text{ GB}}{1 \text{ B/dir}} = \frac{2 \times 2^{30} \text{ B}}{1 \text{ B/dir}} = 2^1 \times 2^{30} \text{ dir}$
- $\#bits \text{ de direccionamiento} = \log_2(2^1 \times 2^{30}) = 31 \text{ bits}$

2 Memoria de 2 GB con direccionamiento a 16 *bits*

- $\#direcciones = \frac{2 \text{ GB}}{2 \text{ B/dir}} = 2^{30} \text{ dir}$
- $\#bits \text{ de direccionamiento} = \log_2(2^{30}) = 30 \text{ bits}$

3 Memoria de 16 MB con direccionamiento a *nibble*(4 bits)

Tamaño, dirección, unidad direccionable

¿Cuántos bits son necesarios para poder direccionar la memoria?

1 Memoria de 2 GB con direccionamiento a *byte*

- $\#direcciones = \frac{2 \text{ GB}}{1 \text{ B/dir}} = \frac{2 \times 2^{30} \text{ B}}{1 \text{ B/dir}} = 2^1 \times 2^{30} \text{ dir}$
- $\#bits \text{ de direccionamiento} = \log_2(2^1 \times 2^{30}) = 31 \text{ bits}$

2 Memoria de 2 GB con direccionamiento a 16 *bits*

- $\#direcciones = \frac{2 \text{ GB}}{2 \text{ B/dir}} = 2^{30} \text{ dir}$
- $\#bits \text{ de direccionamiento} = \log_2(2^{30}) = 30 \text{ bits}$

3 Memoria de 16 MB con direccionamiento a *nibble*(4 bits)

- $\#direcciones = \frac{16 \text{ MB}}{4 \text{ b/dir}} = \frac{2^4 \times 2^{20} \text{ B} \times (8 \text{ b/B})}{4 \text{ b/dir}} = 2^5 \times 2^{20} \text{ dir}$

Tamaño, dirección, unidad direccionable

¿Cuántos bits son necesarios para poder direccionar la memoria?

1 Memoria de 2 GB con direccionamiento a *byte*

- $\#direcciones = \frac{2 \text{ GB}}{1 \text{ B/dir}} = \frac{2 \times 2^{30} \text{ B}}{1 \text{ B/dir}} = 2^1 \times 2^{30} \text{ dir}$
- $\#bits \text{ de direccionamiento} = \log_2(2^1 \times 2^{30}) = 31 \text{ bits}$

2 Memoria de 2 GB con direccionamiento a 16 *bits*

- $\#direcciones = \frac{2 \text{ GB}}{2 \text{ B/dir}} = 2^{30} \text{ dir}$
- $\#bits \text{ de direccionamiento} = \log_2(2^{30}) = 30 \text{ bits}$

3 Memoria de 16 MB con direccionamiento a *nibble*(4 bits)

- $\#direcciones = \frac{16 \text{ MB}}{4 \text{ b/dir}} = \frac{2^4 \times 2^{20} \text{ B} \times (8 \text{ b/B})}{4 \text{ b/dir}} = 2^5 \times 2^{20} \text{ dir}$
- $\#bits \text{ de direccionamiento} = \log_2(2^5 \times 2^{20}) = 25 \text{ bits}$

Tamaño, dirección, unidad direccionable

En resumen,

Obtener la cantidad de direcciones de una memoria:

$$\text{cantidad de direcciones} = \frac{\text{tamaño de la memoria}}{\text{unidad de direccionamiento}}$$

(ojo con las unidades)

Obtener la cantidad de bits para direccionar una memoria:

$$\text{cantidad de bits de dirección} = \log_2(\text{cantidad de direcciones})$$

Diseño de un formato de instrucción

Diseño de un formato de instrucción

Diseñaremos una nueva máquina de propósito general, llamada MATETI.

La misma cuenta con direcciones de 6 *bits*, palabras de 8 bits y direccionamiento a *byte*.

Además posee 8 registros de proposito general: $R_0 \dots R_7$ de 8 *bits*.

Diseño de un formato de instrucción

Diseñaremos una nueva máquina de propósito general, llamada MATETI.

La misma cuenta con direcciones de 6 *bits*, palabras de 8 bits y direccionamiento a *byte*.

Además posee 8 registros de proposito general: $R_0 \dots R_7$ de 8 *bits*.

Diseño de un formato de instrucción

El set de instrucciones es el siguiente:

Instrucción	Efecto
MOV R_x, R_y	$R_x \leftarrow R_y$
LOAD d	$R_0 \leftarrow [d]$
STORE d	$[d] \leftarrow R_0$
ADD1 R_x	$R_x \leftarrow R_x + 1$
SUB1 R_x	$R_x \leftarrow R_x - 1$
JNE $cte3$	si $R_0 \neq 0$ se incrementa el PC en $cte3$
WAIT	no hace nada
SET	Pone en 0 el valor de todos los registros
ROUNDR	Pone en $R_0 \leftarrow R_7$ y $R_i \leftarrow R_{i-1}$ para $0 < i < 8$
ROUNDL	Pone en $R_7 \leftarrow R_0$ y $R_i \leftarrow R_{i+1}$ para $0 \leq i < 7$

donde $cte3$ es una constante de 3 bits en complemento a 2, y $x, y \in [0, 7]$

Ejercicios

- 1 ¿Cuál es el tamaño máximo de memoria que podrá tener MATETI ?
¿Cuál debe ser el tamaño (en *bits*) del *program counter* (PC)?
- 2 Diseñar un formato de instrucción de longitud fija (1 byte) para el *set* de instrucciones dado.
- 3 Decidir si es posible agregar más instrucciones sin operandos o no.
En caso de ser posible indicar cuántas.
- 4 Los usuarios de MATETI van a necesitar con frecuencia sumarle a un registro el valor de una constante, pero el *set* de instrucciones no provee la operación “ADD Rx, N” Dado que NO es posible modificar el *set* de instrucciones y que se usará solamente cuando N igual a 2, 3 ó 7. ¿es posible dotar al ensamblador de un *mecanismo* para que el programador pueda hacer uso de esta funcionalidad? ¿Cómo?

- 1 ¿Cuál es el tamaño máximo de memoria que podrá tener MATETI ? ¿Cuál debe ser el tamaño (en *bits*) del *program counter* (PC)?

El tamaño máximo de la memoria será $2^6 \times \text{byte}$, (64 bytes).

Se requieren 6 bits para almacenar una dirección de memoria.

El PC debe almacenar una dirección de memoria, por lo tanto tendrá 6 bits.

- 2 Diseñar un formato de instrucción de longitud fija (1 byte) para el *set* de instrucciones dado.

Posible solución:

MOV	= 00	rrr ₁	rrr ₂
LOAD	= 01	mmm	mmm
STR	= 10	mmm	mmm
ADD1	= 11	000	rrr ₁
SUB1	= 11	001	rrr ₁
JNE	= 11	010	ccc
WAIT	= 11	011	000
SET	= 11	011	001
ROUNDR	= 11	011	010
ROUNDL	= 11	011	011

- 3 Decidir si es posible agregar más instrucciones sin operandos o no. En caso de ser posible indicar cuántas.

Es posible agregar más instrucciones sin operandos.

Faltan usar las tiras del 11 011 100 al 11 011 111 (4 en total), y del 11 100 000 al 11 111 111 ($4 \times 8 = 32$).

Por lo tanto, se pueden agregar 36 instrucciones sin operandos.

- 4 Los usuarios de MATETI van a necesitar con frecuencia sumarle a un registro el valor de una constante, pero el *set* de instrucciones no provee la operación “ADD R_x , N”
 Dado que NO es posible modificar el *set* de instrucciones y que se usará solamente cuando N igual a 2, 3 ó 7. ¿es posible dotar al ensamblador de un *mecanismo* para que el programador pueda hacer uso de esta funcionalidad? ¿Cómo?

Es posible crear mecanismo a nivel de ensamblador. Este debe repetir la instrucción ADD1 R_x 2, 3 ó 7 veces en función de cuánto valga N.

Entonces el ensamblador se encargará de chequear el valor de N y luego agregar la cantidad de ADD1 necesarias para simular el ADD R_x N.

¿Preguntas?



¿Preguntas?



Mumm-Ra