

Conceptos de Organización de Computadoras

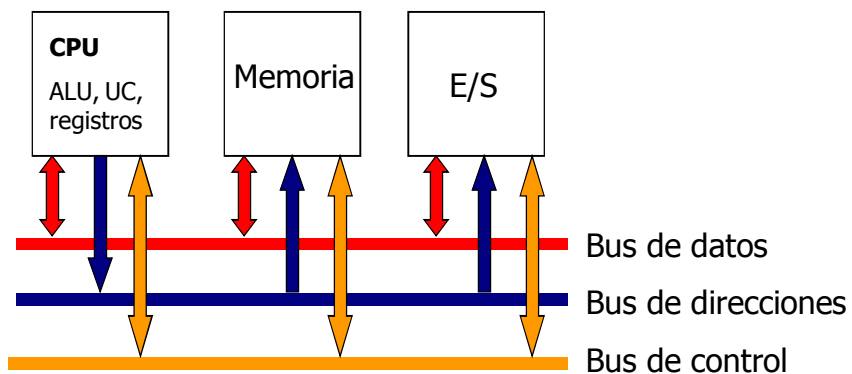
Curso de Ingreso 2014

Clase 1

Prof. Jorge M. Runco

1

Sistema de Cómputo



2

Sistema de Cómputo

- El sistema está formado por tres bloques principales:
- CPU: Unidad Central de Procesamiento
- Memoria: donde se almacenan los datos y programas.
- E/S: entrada/salida que involucra a todos los demás dispositivos (monitor, teclado, etc.)

3

Interconexión: buses

- Los distintos bloques se comunican a través de un medio de comunicación compartido que se llama bus.
- Es compartido porque todos dispositivos están conectados a él.
- Los dispositivos intercambian información por el *bus de datos*. Por él viajan los datos e instrucciones del programa.
- Si todos los dispositivos están conectados al mismo lugar, ¿cómo hace la CPU para comunicarse con uno de ellos en particular?

4

Bus de direcciones

- Cada dispositivo se distingue uno de otro en su *dirección*. Una dirección es un conjunto de bits (unos y ceros) característicos de cada dispositivo.
- Actúan como el "nombre" para cada uno de ellos.
- No puede haber dos dispositivos con la misma dirección.
- Las direcciones viajan por el bus de direcciones.

5

Bus de datos y control

- La CPU envía bits por el bus de direcciones para comunicarse con un dispositivo.
- Por el *bus de datos* se intercambia la información (bits de datos y programas).
- Falta un mecanismo para controlar todas las operaciones: *bus de control*.
- Por el bus de control viajan los bits que controlan las distintas operaciones.

6

Resumiendo

- La CPU envía un conjunto de bits por el bus de direcciones para comunicarse con un dispositivo, por el bus de datos viaja la información (bits) desde/a la cpu, a/desde la memoria ó E/S y por el bus de control viajan los bits que controlan la operación.

7

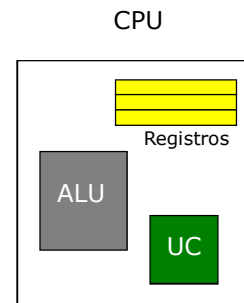
En la guía.....

- La UCP está conectada con el resto de los componentes del sistema a través de 3 buses distintos: direcciones, datos y control.
- Independientemente de la implementación de cada procesador, la información que viaja por el bus de direcciones (las *direcciones*) tiene como objetivo "identificar" otro componente con el cual la UCP quiere comunicarse, todos los componentes están conectados al bus de direcciones, pero aquél que "reconoce" su dirección queda conectado a la UCP y el resto es como si no estuvieran.
- Una vez comunicada la UCP con otro componente, puede enviar ó recibir información (denominada *dato*) a través del bus de datos.

8

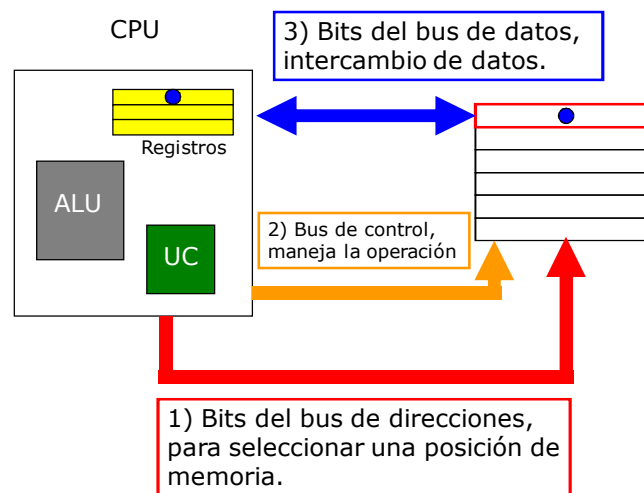
CPU

- Podemos distinguir básicamente 3 elementos:
- Registros: como "memoria" pero en la CPU. Lugar de almacenamiento temporario.
- ALU: unidad aritmético lógica. Realiza las operaciones aritméticas y lógicas.
- UC: unidad de control. "La verdadera CPU", lleva el control de todas las operaciones.



9

Datos desde/hacia la CPU



10

Sistemas de numeración

- Un sistema de numeración es un conjunto de símbolos y reglas, que nos permiten construir todos los números válidos para el sistema.
- Los sistemas de numeración se clasifican en: posicionales y no posicionales.
- En los sistemas posicionales: un número depende no sólo del símbolo utilizado sino también de la posición que ocupa dentro del número.

11

Sistemas posicionales

- Si el sistema posicional tiene *base* b , quiere decir que hay sólo b símbolos distintos para formar todos los números del sistema.
- Ejemplos
 - Base 10: hay 10 símbolos distintos 0...9
 - Base 2: hay 2 símbolos distintos 0 y 1.
 - Base 8: hay 8 símbolos distintos 0...7
- Cualquier número escrito en base 2, sólo puede tener 1 (unos) y 0 (ceros). (No aparecen 2...9)
- Cualquier número escrito en base 8, sólo puede tener símbolos 0...7. (No aparecen 8 y 9).

12

Sistema posicional: ejemplo (1)

- Supongamos trabajar en base 10.
 - Escribamos el número cuatro: 4
 - Escribamos el número cuarenta: 40
 - Escribamos el número cuatrocientos: 400
- Cada nuevo número es escrito con el mismo símbolo (4) desplazado a la izquierda ocupando una posición distinta:
 - Unidad
 - Decena
 - Centena

13

Sistema posicional: ejemplo (2)

- Lo mismo podemos escribirlo de la siguiente manera:

4	→	4×10^0
4 0	→	4×10^1
4 0 0	→	4×10^2
<div style="display: flex; justify-content: space-around; width: 100px;"> <div style="text-align: center;">↓</div> <div style="text-align: center;">↓</div> <div style="text-align: center;">↓</div> </div>		
<div style="display: flex; justify-content: space-around; width: 100px;"> 10^2 10^1 10^0 </div>		

14

Sistema posicional: ejemplo (3)

- En general: $\text{dígito} \times (\text{base})^{\text{posición del dígito}}$
- En general podemos decir que en un sistema posicional cada número se genera usando los distintos dígitos de la base y cada uno de ellos resulta multiplicado por una potencia de la base, de acuerdo a la posición que ocupan.
- $1895 = 1 \times 10^3 + 8 \times 10^2 + 9 \times 10^1 + 5 \times 10^0 = 1000 + 800 + 90 + 5 = 1895$

15

¿Cuántos números podemos escribir?

- En base 10:
- 1 dígito $\Rightarrow 0 \dots 9 \Rightarrow 10^1 = 10$ números
- 2 dígitos $\Rightarrow 0 \dots 99 \Rightarrow 10^2 = 100$ "
- 3 dígitos $\Rightarrow 0 \dots 999 \Rightarrow 10^3 = 1000$ "
- En general: la *cantidad de números distintos* = $(\text{base})^{\text{cantidad de dígitos}}$
- Recordar: siempre. Para cualquier base.

16

¿Cómo se generan los distintos números? (Base 10)

- Comenzamos con un dígito, desde el 0....9, el siguiente número se genera agregando un dígito a la izquierda 1 (es el que sigue al anterior) y repetimos la columna de la derecha de 0....9.
 - Cuando llegamos al final 19, aumentamos en 1 la columna de la izquierda y comenzamos otra vez, 20.....29.
 - Cuando llegamos al último número (99), agregamos otra columna a la izquierda y repetimos el proceso.
- | | | | | | |
|---|----|----|------|----|-----|
| 0 | 10 | 20 | | 90 | 100 |
| 1 | 11 | 21 | | 91 | |
| 2 | 12 | 22 | | 92 | |
| 3 | 13 | 23 | | 93 | |
| 4 | 14 | 24 | | 94 | |
| 5 | 15 | 25 | | 95 | |
| 6 | 16 | 26 | | 96 | |
| 7 | 17 | 27 | | 97 | |
| 8 | 18 | 28 | | 98 | |
| 9 | 19 | 29 | | 99 | |

17

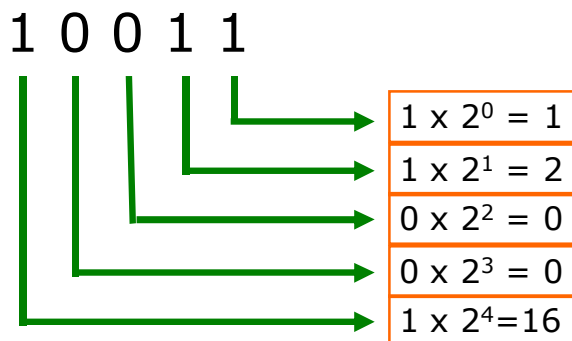
Sistema binario – Base 2

- Veamos estas ideas pero para base=2.
- Hay sólo 2 símbolos distintos: 0 y 1.
- Cada uno de estos símbolos ó dígitos binarios se llama *bit*.
- Cualquier número estará formado por unos (1) y ceros (0). No hay otro símbolo.
- Números distintos = $(2)^{\text{cantidad de dígitos}}$
- Aquí las posiciones también son potencias de la base, pero ahora la base es 2.

18

Ejemplo en base 2

- Dado el número 10011_2 ¿qué número en base 10 representa?



19

- Hicimos lo mismo que en base 10, multiplicamos cada dígito (0,1) por una potencia de la base (2) que depende de la posición del dígito.
- Finalmente el número equivalente en base 10 será:
- $16 + 0 + 0 + 2 + 1 = 19_{10}$ ←
- $10011_2 \Rightarrow 19_{10}$ son números equivalentes escritos en distinta base.

20

Generalizando

- El resultado recién alcanzado, se expresa en forma general:

$$N_{10} = \sum_{i=-m}^{i=n} d_i \times b^i = d_{-m} \times b^{-m} + d_{-m+1} \times b^{-m+1} + \dots + d_0 \times b^0 + d_1 \times b^1 + d_2 \times b^2 + \dots + d_n \times b^n$$

- Se conoce como el *Teorema fundamental de la numeración*. N siempre quedará expresado en base 10, independientemente de la base b utilizada en el lado derecho de la igualdad.

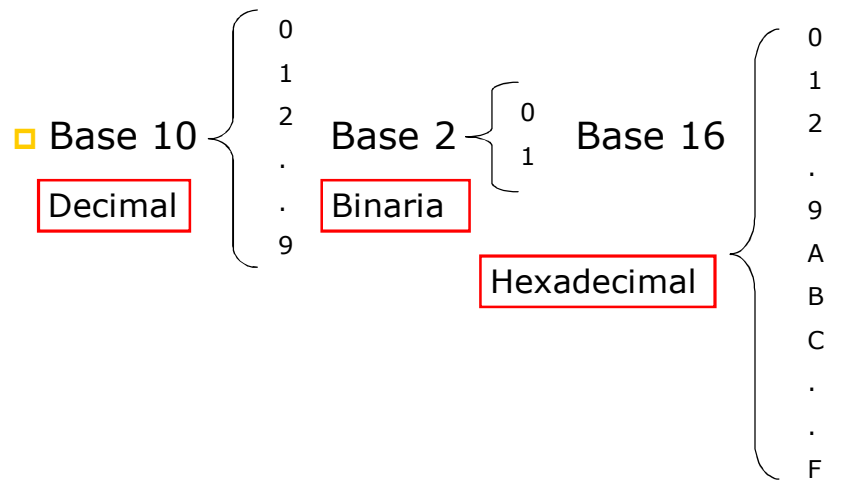
21

Base mayor que 10 (Ej. 16)

- La base nos dice la cantidad de dígitos distintos que debe tener el sistema.
- Si $b=16$, los símbolos serán 0,1,2,...,9 hay 10 símbolos distintos. El símbolo que sigue al 9 no puede ser el 10 porque está formado por el 1 y 0 que ya fueron usados.
- Usamos letras A, B, C, D, E y F. Hasta la F porque ya tenemos los 16 símbolos distintos. $A_{16}=10_{10}$ $B_{16}=11_{10}$ $F_{16}=15_{10}$.

22

Resumiendo



23



Ejemplo en base 16

- Dado el número A3F₁₆ ¿qué número en base 10 representa? Base

□ A 3 F

$$\begin{aligned}
 &F \times (16)^0 = 15 \times 1 = 15 \\
 &3 \times (16)^1 = 3 \times 16 = 48 \\
 &A \times (16)^2 = 10 \times 256 = 2560
 \end{aligned}$$

24

- Hicimos lo mismo que en base 10, multiplicamos cada dígito (0,1,2....F) por una potencia de la base (16) que depende de la posición del dígito.
- Finalmente el número equivalente en base 10 será:
- $15 + 48 + 2560 = 2623_{10}$ 
- $A3F_{16}$  2623_{10} son números equivalentes escritos en distinta base.

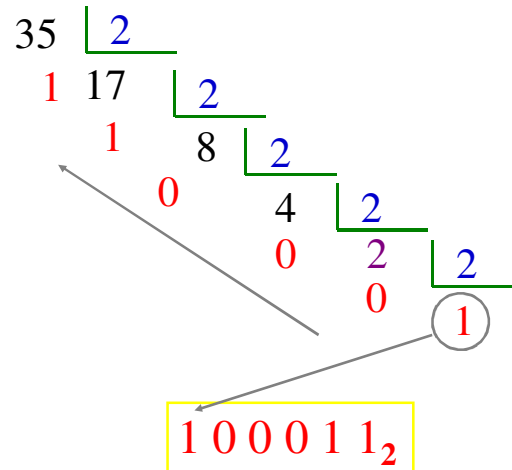
25

Cambio de base

- Para pasar un número de base 10 a cualquier otra base:
 1. Dividir el número decimal por la nueva base. Guardar cociente y resto.
 2. Tomar el cociente anterior y repetir el paso 1 hasta que el cociente sea menor que la base.
 3. Escribir (concatenar) el último cociente y los restos empezando por el último.

26

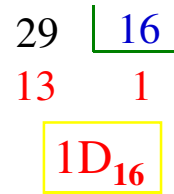
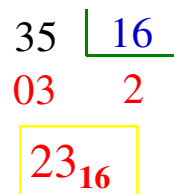
Ej. $35_{10} \longrightarrow ?_2$



27

Ej. $35_{10} \longrightarrow ?_{16}$

$29_{10} \longrightarrow ?_{16}$



Por el teorema fundamental de la numeración ,podemos escribir:

$$23_{16} = 2 \times 16^1 + 3 \times 16^0 = 32 + 3 = 35_{10}$$

$$1D_{16} = 1 \times 16^1 + 13 \times 16^0 = 16 + 13 = 29_{10}$$

28

De base 2 a base 16

- Dado un número en base 2, para pasarlo a base 16 debemos formar "grupos" de 4 bits partiendo de la derecha y luego reemplazando cada grupo por el número hexadecimal correspondiente.
- Ej.

$$\begin{array}{ccccccc} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ \underbrace{\hspace{1.5cm}} & \underbrace{\hspace{1.5cm}} & \underbrace{\hspace{1.5cm}} & & & & & & & \\ 3 & & A & & & & 6 & & & \end{array} = 3A6_{16}$$

29

De base 2 a base 16 ¿Base 8?

- Formamos grupos de 4 bits porque la relación que hay entre las bases es

$$2^{\textcircled{4}} = 16$$

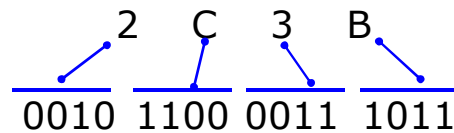
Cantidad de bits
del grupo

Si la base

30

De base 16 a base 2

- Dado un número en base 16, para pasarlo a base 2 debemos reemplazar cada dígito hexadecimal por los 4 bits de su número binario equivalente.
- Ej. $2C3B_{16} = ?_2$
-



31

DECIMAL	BINARIO	HEXADECIMAL	OCTAL
00	0000	0	00
01	0001	1	01
02	0010	2	02
03	0011	3	03
04	0100	4	04
05	0101	5	05
06	0110	6	06
07	0111	7	07
08	1000	8	10
09	1001	9	11
10	1010	A	12
11	1011	B	13
12	1100	C	14
13	1101	D	15
14	1110	E	16
15	1111	F	17

32

Modelo de Memoria

- Suponga un modelo de memoria que está formado por "cajitas" que pueden guardar información. Dentro de esa cajita se escribirá el dato a almacenar (DATO) y la identificaremos con un nombre, una etiqueta llamada DIRECCION.
- Tenga en cuenta que una dirección es un número que identifica un lugar, en este caso una cajita. El número es un valor binario que deberá ser expresado como combinación de unos y ceros.
- Cada dirección deberá ser única, a fin de que, cuando la UCP quiera trabajar con una cajita, lo haga con una a la vez.

33

DIRECCIÓN

DATO

Si se tienen solamente 2 cajitas, con una dirección de un solo bit se podrá identificar cada una de ellas, sin posibilidad de error. Por ejemplo tendremos:

DIRECCIÓN = 0_2

DATO1

DIRECCIÓN = 1_2

DATO2

34

Si tengo 4 cajitas se necesitarán direcciones con 2 bits para poder identificar cada una de ellas. Notar que utilizamos el subíndice 2 en las direcciones para identificarlas como números binarios.

DIRECCIÓN = 00_2	DATO1
DIRECCIÓN = 01_2	DATO2
DIRECCIÓN = 10_2	DATO3
DIRECCIÓN = 11_2	DATO4

35

En general para identificar **N** diferentes posiciones de memoria (cajitas) se necesitará que **n** (el número de bits del bus de direcciones) sea tal que se cumpla

$$N \leq 2^n$$

Ejemplo 1. ¿Cuántos bits deberán tener las direcciones para identificar 250 posiciones de memoria diferentes?

$N = 256 \leq 2^n$ si $n=8$, $2^8 = 256$ y se cumple la desigualdad. Respuesta: 8 bits.

Ejemplo 2. Escriba en binario la dirección más pequeña y la más grande para el bus de direcciones del ej. anterior. ¿Puede determinar su valor en decimal?.

Dirección mas pequeña = $00000000_2 = 0_{10}$

Dirección mas grande = $11111111_2 = 255_{10}$

36

En los ejemplos del apartado anterior identificamos cada posición de memoria (celda o cajita) con un número binario llamado dirección. La cantidad de bits almacenados en ella (y que llamamos DATO) se conoce como "unidad mínima direccionable". Si la cajita puede contener 8 bits (llamado byte) de información decimos que la unidad mínima direccionable es el byte.

Entonces podemos decir que, el byte llamado DATO1 se encuentra en la dirección 00_2 , el byte DATO2 en la dirección 01_2 , etc..

37

Supongamos organizar las cuatro cajitas del dibujo anterior de la siguiente manera:

DIRECCIÓN = 0_2	DATO1	DATO2
DIRECCIÓN = 1_2	DATO3	DATO4

La dirección 0_2 identifica a un valor almacenado en 2 cajitas "pegadas" DATO1 y DATO2. O sea tenemos 4 cajitas pero no "puedo distinguir" como antes (con 2 direcciones distintas) a DATO1 de DATO2, sólo puedo leer o escribir ambas cajitas a la vez. Esto implica que la mínima unidad direccionable es una caja con la capacidad de las 2 cajitas anteriores, por lo tanto necesito menos bits en el bus de direcciones, uno en este caso, pero el bus de datos debe tener el doble de bits que el caso anterior.

38

Tamaño palabra	Número	Abreviatura
8 bits	$2^8 = 256$	
10 bits	$2^{10} = 1024$	1 Kb
16 bits	$2^{16} = 65\,536$	64 Kb
20 bits	$2^{20} = 1\,048\,576$	1 Mb
28 bits	$2^{28} = 268\,435\,456$	256 Mb
30 bits	$2^{30} = 1\,073\,741\,820$	1 Gb

39

P2)Ej3)

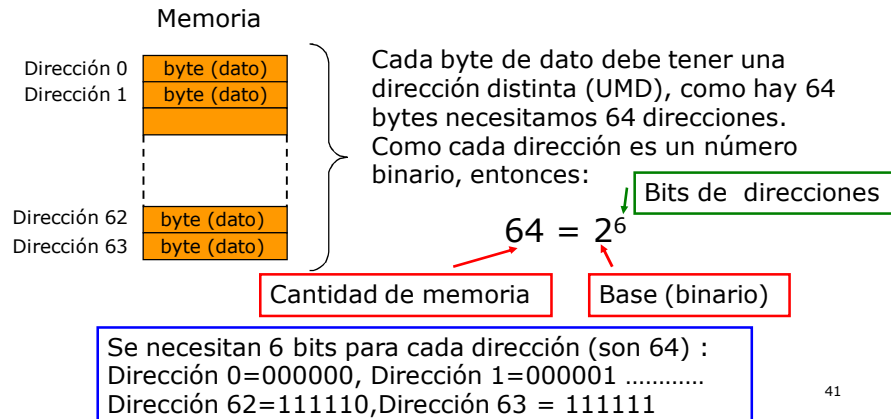
➤ Suponga tener tres procesadores de 8, 16 y 32 bits del bus de datos:

- a) ¿Cuántos bits tienen las direcciones de cada procesador, considerando una memoria que almacena 64 bytes y la unidad mínima direccionable el byte?.
- b) Idem a) pero considere que la unidad mínima direccionable es de 8, 16 y 32 bits respectivamente.

40

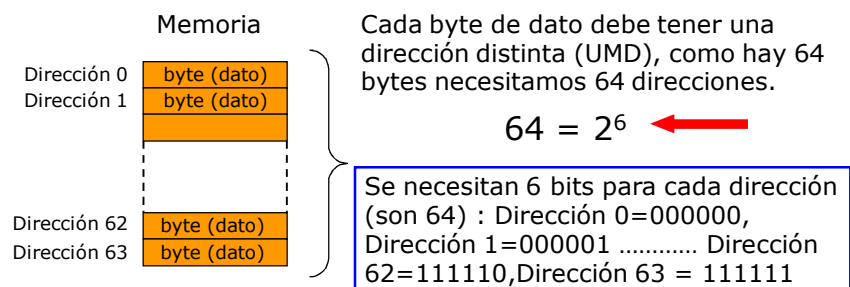
a) Bus de datos = 8 bits

- Unidad mínima direccionable (UMD) : 1 byte. A cada byte le corresponde una dirección.



a) Bus de datos = 16 bits

- Unidad mínima direccionable (UMD) : 1 byte. A cada byte le corresponde una dirección.

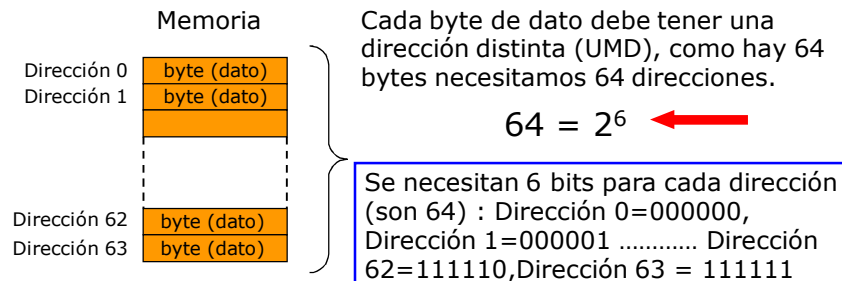


El resultado es el mismo que el inciso anterior, pues lo determina la UMD. Por tener un bus de datos de 16 bits (2 bytes), el procesador puede leer/escribir 1 ó 2 datos en memoria (1 ó 2 bytes).

42

a) Bus de datos = 32 bits

- Unidad mínima direccionable (UMD) : 1 byte. A cada byte le corresponde una dirección.

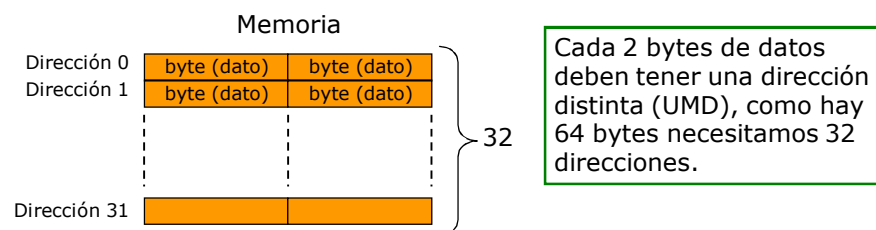


El resultado es el mismo que el inciso anterior, pues lo determina la UMD. Por tener un bus de datos de 32 bits (4 bytes), el procesador puede leer/escribir 1,2 ó 4 datos en memoria (1, 2 ó 4 bytes).

43

b) Bus de datos = 8 bits

- Unidad mínima direccionable (UMD) : 2 bytes. Cada 2 bytes corresponde una dirección.

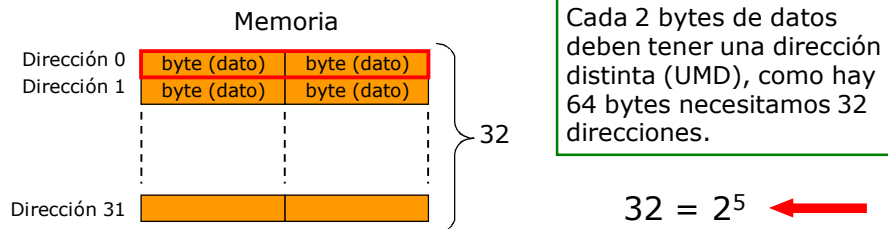


Para este modelo de memoria el procesador debería leer/escribir 16 bits (2 bytes) en forma simultánea. Como el bus de datos es de 8 bits, no puede hacerlo. Por lo tanto no nos sirve para este modelo de memoria.

44

b) Bus de datos = 16 bits

- Unidad mínima direccionable (UMD) : 2 bytes.
Cada 2 bytes corresponde una dirección.

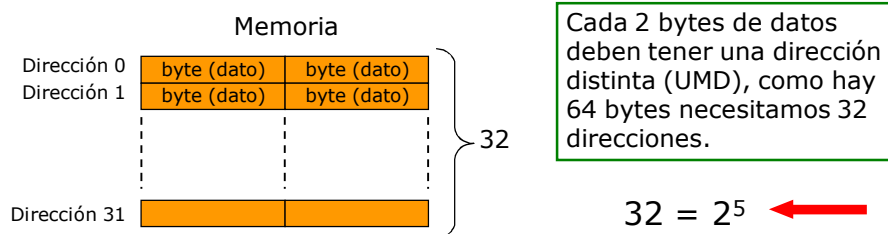


Se necesitan 5 bits para cada dirección (son 32) : Dirección 0=00000, Dirección 1=00001 Dirección 30=11110, Dirección 31 = 11111. Aquí no hay una dirección por byte, la Dirección 0 " nombra " a los 2 bytes "juntos".

45

b) Bus de datos = 32 bits

- Unidad mínima direccionable (UMD) : 2 bytes.
Cada 2 bytes corresponde una dirección.

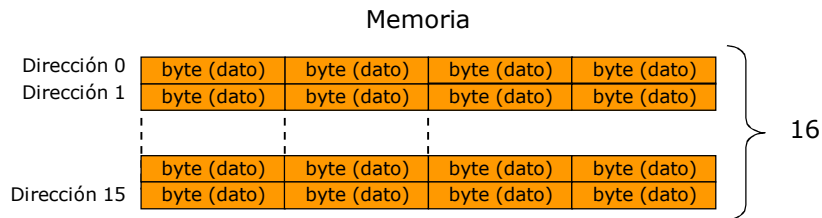


Mismo resultado que el inciso anterior: 5 bits. Pero al ser el procesador de 32 bits, puede leer/escribir 2 ó 4 bytes simultáneos.

46

b) Bus de datos = 8, 16 y 32 bits

- Unidad mínima direccionable (UMD) : 4 bytes.
Cada 4 bytes corresponde una dirección.



$$16 = 2^4 \quad \leftarrow$$




Cada 4 bytes de datos deben tener una dirección distinta (UMD), como hay 64 bytes necesitamos 16 direcciones. Por lo explicado anteriormente, sólo nos sirve el procesador con 32 bits de bus de datos

47

P2)Ej2)

- ¿Cuántos bits se necesitan para direccionar 68719476736 lugares de memoria?

- Cantidad de memoria = $2^{\text{bits de direcciones}}$

- $128 = 2^7$  7 bits
- $256 = 2^8$  8 bits
- $512 = 2^9$  9 bits

48