

# Estructura de Computadores

## Curso 2018/2019 – Grupo: C

### Ejercicios de mapas de memoria

Nombre y apellidos: \_\_\_\_\_Vladyslav Mazurkevych\_\_\_\_\_

Fecha: \_\_\_\_\_04/01/19\_\_\_\_\_

#### Ejercicio 1

Considere el diseño de un sistema de memoria que se va a conectar a un procesador con un ancho de palabra de 32 bits y un espacio de direccionamiento de 4GB. La relación de los módulos que integra este sistema de memoria y las características de cada módulo se muestra en la siguiente tabla:

Módulo	Capacidad	Dirección Inicio
RAM1	512 MB	0x00000000
RAM2	1 GB	0x40000000
RAM3	512MB	0x80000000
VIDEO	512MB	0xC0000000

- a) Realiza el mapa de memoria de este sistema, indicando la dirección de inicio y de fin de cada módulo y los huecos si los hubiera, indicando en este caso su tamaño.

Módulo	Capacidad	A <sub>31</sub>	A <sub>30</sub>	A <sub>29</sub>	A <sub>28</sub>	A <sub>27</sub>	A <sub>26</sub> ...	A <sub>0</sub>	Dirección Inicial-Final (hex)
RAM1	512MB			000 000000000...0					0x00000000-0x1FFFFFFF
				000 111111111...1					
BLANCO1	512MB			001 000000000...0					0x20000000-0x3FFFFFFF
				001 111111111...1					
RAM2	1GB			01 000000000...0					0x40000000-0x7FFFFFFF
				01 111111111...1					
RAM3	512MB			100 000000000...0					0x80000000-0x9FFFFFFF
				100 111111111...1					
BLANCO2	512MB			101 000000000...0					0xA0000000-0xBFFFFFFF
				101 111111111...1					
VIDEO	512MB			110 000000000...0					0xC0000000-0xDFFFFFFF
				110 111111111...1					
BLANCO3	512MB			111 000000000...0					0xE0000000-0xFFFFFFF
				111 111111111...1					

- b) Especifica las funciones de selección de cada módulo en lógica negativa (activas a nivel bajo).

Â = A negada.

RAM1: 000 >>  $\hat{A}_{31} * \hat{A}_{30} * \hat{A}_{29} = A_{31} + A_{30} + A_{29} >$  Lógica negada >  $\hat{A}_{31} + \hat{A}_{30} + \hat{A}_{29}$

RAM2: 01 >>  $\hat{A}_{31} * A_{30} = A_{31} + \hat{A}_{30} >$  Lógica negada >  $\hat{A}_{31} + \hat{A}_{30}$

RAM3: 100 >>  $A_{31} * \hat{A}_{30} * \hat{A}_{29} = \hat{A}_{31} + A_{30} + A_{29} >$  Lógica negada >  $A_{31} + \hat{A}_{30} + \hat{A}_{29}$

VIDEO: 110 >>  $A_{31} * A_{30} * \hat{A}_{29} = \hat{A}_{31} + \hat{A}_{30} + A_{29} >$  Lógica negada >  $A_{31} + A_{30} + \hat{A}_{29}$

c) Suponiendo que el módulo RAM1 (512MB) está constituido por una fila de chips SDRAM de 4 bits de ancho de palabra, indíquese

i) El tamaño LxW de cada chip y el número de chips necesarios para construir el módulo.

W= 32b, w= 4b

Numero de chips = W/w = 32/4 = **8chips**

Tamaño de cada chip: 512MB / 8chips = **64 MB/chip**

ii) El nombre de las líneas del bus de direcciones que se utilizarían para el direccionamiento del módulo (palabras y bytes dentro de la palabra)

Tenemos ancho de palabra de 32b, por eso el nombre de las líneas del bus será:

A <sub>31</sub> .....A <sub>0</sub>	BE* <sub>3</sub> BE* <sub>2</sub> BE* <sub>1</sub> BE* <sub>0</sub>
-------------------------------------	---------------------------------------------------------------------

## Ejercicio 2

El sistema de memoria de un procesador de 32 bits de datos y 32 bits de direcciones dispone de tres módulos de memoria: M1, M2 y M3. Las funciones de selección de dichos módulos son, en lógica negativa, las siguientes:

$$\overline{f_{M1}} = A_{31} + A_{30} + A_{29} + A_{28}$$

$$\overline{f_{M2}} = A_{31} + A_{30} + \overline{A_{29}} + A_{28}$$

$$\overline{f_{M3}} = A_{31} + A_{30} + \overline{A_{29}} + \overline{A_{28}} + A_{27}$$

Rellene la siguiente tabla indicando la capacidad, dirección inicial y dirección final de cada módulo, así como de los espacios libres que hay en el mapa.

Módulo	Capacidad	A <sub>31</sub>	A <sub>30</sub>	A <sub>29</sub>	A <sub>28</sub>	A <sub>27</sub>	A <sub>26</sub> ...	A <sub>0</sub>	Dirección Inicial-Final (hex)
FM1	256MB	0000 0000000...0 0000 1111111...1							0x00000000- 0x0FFFFFFF
BLANCO	512MB	0001 0000000...0 0010 1111111...1							0x10000000- 0x2FFFFFFF
FM3	1GB	00110 0000000...0 00110 1111111...1							0x30000000- 0x37FFFFFF
BLANCO	2GB	00111 0000000...0 10111 1111111...1							0x38000000- 0xBFFFFFFF
FM2	256MB	1100 0000000...0 1100 1111111...1							0xC0000000- 0xCFFFFFFF