

## PROBLEMAS TEMA 1: Estructuras de interconexión de un computador

### Problemas propuestos en examen

#### PROBLEMA 1.

Una CPU que emplea un tamaño de palabra de 16 bits tiene un repertorio de 16 instrucciones con un formato único compuesto por un campo de 5 bits, para el código de operación, y otro campo de 11 bits para los operandos. Todos los 16 códigos de operación comienzan por un 0 y, por tanto, van del 00000 al 01111. Indicar si las siguientes afirmaciones son verdaderas o falsas

- I. Si en una dirección de memoria nos encontramos el valor 0011010110001010, necesariamente es una instrucción máquina.
- II. Si en una dirección de memoria nos encontramos el valor 1011010110001010, necesariamente es un dato.

#### RESPUESTA:

##### DATOS

- Tamaño de palabra de 16 bits.
- Repertorio de 16 instrucciones.
- Formato de instrucción :
  - Campo de 5 bits para el código de operación.
  - Campo de 11 bits para el campo de operandos.
- Los 16 códigos de operación comienzan por un 0 (00000-01111)

#### Afirmación I

-De acuerdo con los principios de un computador con una arquitectura de Von Neuman, en la memoria principal se pueden almacenar tanto datos como instrucciones.

- Luego la afirmación I, es **falsa**, ya que si el contenido de una dirección de memoria empieza por 0, puede ser un dato o una instrucción máquina dependiendo de la fase de ejecución de la instrucción en que nos encontremos.

#### Afirmación II

- Del enunciado se sabe que para que el contenido de una posición de memoria sea considerado una instrucción su bit más significativo debe ser un 0, puesto que en este caso comienza por 1 necesariamente debe ser un dato. Luego la afirmación II es **verdadera**.

## PROBLEMA 2

La memoria de un ordenador consta de 10 módulos independientes conectados a un bus de memoria común. En una operación de escritura cada uno de estos módulos únicamente ocupa el bus del sistema al comienzo de la operación y durante un 20% del tiempo total del ciclo de escritura. El 80% del tiempo de ciclo de escritura restante el módulo lo emplea para almacenar el dato internamente. Si las operaciones de escritura de dichos módulos de memoria pueden solaparse, ¿en cuánto se ve incrementada la velocidad máxima de almacenamiento de la memoria del ordenador en relación a la velocidad de un único módulo?

## SOLUCIÓN PROBLEMA 2

### DATOS

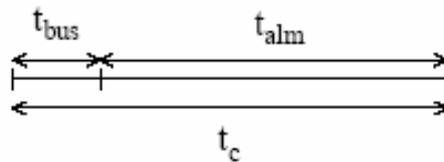
- Memoria de 10 módulos independientes conectados a un mismo bus de memoria.
- Cada bus ocupa el bus durante el 20% del ciclo de escritura.
- El 80 % del tiempo de ciclo restante el módulo almacena el dato internamente.
- Las operaciones de estos módulos pueden solaparse.

Supongamos que la memoria del computador consta de un único módulo, la velocidad máxima de almacenamiento sería:

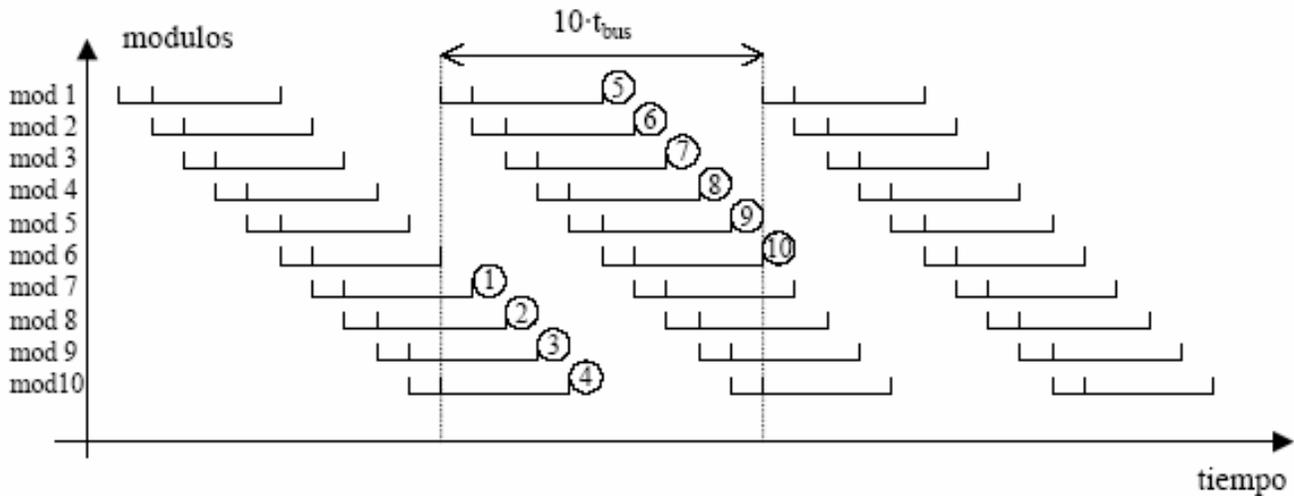
$$v_{max} = \frac{1}{t_c} \text{ (pal / seg)}$$

De acuerdo con el enunciado del problema el tiempo de ciclo  $t_c$ , se puede considerar como la suma de dos componentes:

- tiempo de utilización del bus  $t_{bus}=0.2t_c$
- tiempo de almacenamiento interno del dato  $t_{alm}=0.8t_c$



Utilizando los 10 módulos disponibles, y solapando al máximo los ciclos de escritura para intentar que en todo momento uno de los módulos esté accediendo al bus, se llega a la situación representada por la siguiente figura



El módulo 1 inicia una operación de escritura haciendo uso del bus en exclusiva durante  $t_{bus}$ . A continuación el módulo 2 puede iniciar su operación de escritura y así sucesivamente hasta llegar al módulo 10, al que le sigue de nuevo el módulo 1 repitiéndose el ciclo.

La parte central de la Figura delimitada por dos líneas punteadas se repite de forma continua y, por tanto cada  $10 \cdot t_{bus}$  empiezan o finalizan 10 operaciones de escritura. Encerradas en un círculo se enumeran estas 10 operaciones. La velocidad de almacenamiento definida como el número de palabras escritas en memoria por unidad de tiempo, es por tanto

$$v'_{max} = \frac{10(\text{pal})}{10 \cdot t_{bus} (\text{seg})} = \frac{10}{10 \cdot 0.2 \cdot t_c} = \frac{v_{max}}{0.2} = 5 \cdot v_{max} (\text{pal} / \text{seg})$$

Luego la velocidad de almacenamiento se ve incrementada **5 veces**.

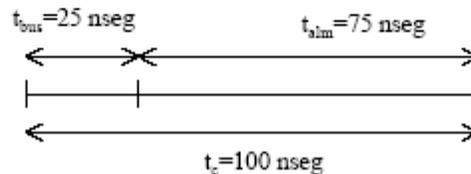
### PROBLEMA 3

La memoria de un ordenador consta de 20 módulos independientes conectados a un bus de memoria común. En una operación de escritura cada uno de estos módulos únicamente ocupa el bus del sistema durante 25 nseg. Al comienzo de la operación. Durante los 75 nseg. siguientes completa el ciclo de escritura almacenando el dato internamente. Si las operaciones de escritura de dichos módulos de memoria pueden solaparse, ¿cuál es la velocidad máxima de almacenamiento de la memoria del ordenador ?

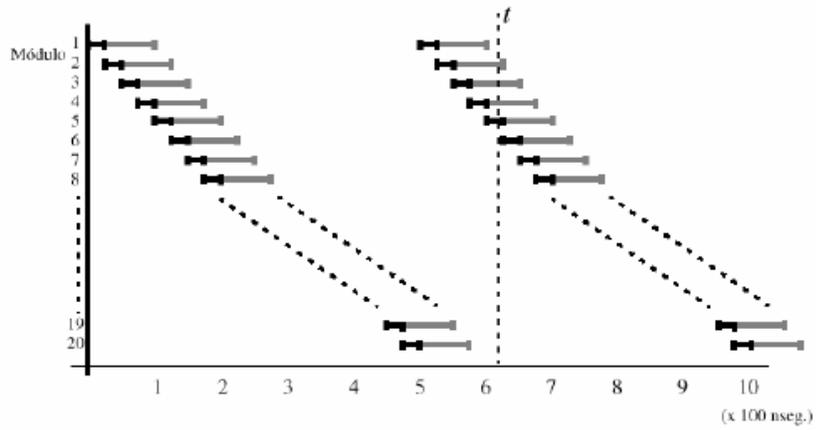
### SOLUCIÓN PROBLEMA 3

#### DATOS

- Memoria de 20 módulos independientes conectados a un mismo bus de memoria.
- Cada bus ocupa el bus durante el 25 nseg del ciclo de escritura.
- Durante los 75 nseg siguientes el módulo almacena el dato internamente.
- Las operaciones de estos módulos pueden solaparse.



De forma inmediata se ve que si cada 100 nseg. el módulo de memoria ocupa 25 nseg., es decir, 1/4 del tiempo, vamos a poder solapar, cómo máximo, 4 operaciones de escritura con 4 módulos de memoria distintos aunque nuestro sistema disponga de 20 módulos. Para aclarar este punto en la Figura se muestra solapamiento máximo que se puede conseguir utilizando todos los módulos de memoria. En el instante de tiempo  $t$  únicamente están realizando parte de una operación de escritura 4 módulos: módulos 2, 3, 4 y 5.



Por tanto, en una operación de escritura continua en este sistema de memoria, se pueden llegar a iniciar y acabar 4 operaciones de escritura cada 100 nseg. La velocidad resultante es:

$$v_{max} = \frac{4(pal)}{100(nseg)} = \frac{4}{10^{-7}} = 4 \cdot 10^7 (pal / seg)$$

### PROBLEMA 1.6

A un bus dedicado a E/S se han conectado la memoria principal y cinco discos iguales. El tiempo de ciclo de la memoria principal es 400 nseg. La velocidad de transferencia de cada disco es B. La longitud de palabra es 32 bits. Cada transmisión sobre el bus necesita 500 nseg para enviar el dato y las diferentes órdenes del protocolo. ¿En cuál de las siguientes situaciones estaría más justificado el uso de un bus con temporización síncrona?

- A)  $B = 10^6$  bits/seg
- B)  $B = 4 \times 10^6$  bits/seg
- C)  $B = 10^7$  bits/seg
- D)  $B = 8 \times 10^7$  bits/seg

### SOLUCIÓN PROBLEMA 1.6

En un bus con *temporización síncrona*, todos los dispositivos conectados a él están funcionando con una frecuencia de reloj fija. Su utilización es conveniente cuando los dispositivos que tiene conectados tienen velocidades de transferencia  $v_T$  muy parecidas ya que si se tienen conectados un grupo heterogéneo de dispositivos, con velocidades muy distintas, el bus debe adecuarse al más lento impidiendo que los más rápidos empleen todo su potencial.

$$v_T = \frac{1}{t_c} = \frac{1}{400 \cdot 10^{-9}} = 25 \cdot 10^5 \text{ (palabras / seg)}$$

$$v_T = 25 \cdot 10^5 \left( \frac{\text{palabras}}{\text{seg}} \right) \cdot 32 \left( \frac{\text{bits}}{\text{palabra}} \right) = 8 \cdot 10^7 \text{ (bits / seg)}$$

$$\boxed{B \approx v_T = 8 \cdot 10^7 \text{ (bits / seg)}}$$

## **PROBLEMA 1.7**

Indique si las siguientes afirmaciones son verdaderas. En un bus con arbitraje distribuido:

- I. La posición de conexión de los dispositivos a la línea de arbitraje no determina la prioridad de aquellos en el uso del bus.
- II. El uso del bus por un dispositivo se interrumpe cuando otro dispositivo con mayor prioridad solicita el uso del bus.

## **SOLUCIÓN PROBLEMA 1.7**

### **Afirmación I:**

Es **FALSA**, ya que en este tipo de arbitraje el dispositivo más cercano a la línea de arbitraje es aquel que tiene la máxima prioridad.

### **Afirmación II:**

Es **FALSA**, ya que mientras la línea de “bus ocupado” esté activada por un modulo, no puede ser utilizado el bus por ningún otro módulo pese a poseer una mayor prioridad.

### **PROBLEMA 1.8**

Indique cuál de las siguientes afirmaciones sobre los buses es cierta:

- A) Los buses transportan únicamente datos y direcciones.
- B) Los buses pueden ser compartidos por dispositivos de muy distintas velocidades.
- C) No es necesario un método de arbitraje para evitar que varias unidades vuelquen simultáneamente datos al bus.
- D) Ninguna de las anteriores.

### **SOLUCIÓN PROBLEMA 1.8**

**Afirmación A:** Es **FALSA**. Un bus puede transportar datos, direcciones y también señales de control.

**Afirmación B:** Es **VERDADERA**. Por ejemplo, en el bus del sistema se encuentran conectados la CPU, la memoria principal y los controladores de E/S, cada uno de estos elementos poseen velocidades distintas.

**Afirmación C:** Es **FALSA**, es necesario un método de arbitraje (centralizado o distribuido) para gestionar el uso del bus por parte de los distintos elementos conectados al mismo.

**Afirmación D:** Es **FALSA**, ya que es verdadera la afirmación B.

### **PROBLEMA 1.9**

Indique si las siguientes afirmaciones son verdaderas. En un bus con **arbitraje distribuido**:

- I. La posición de conexión de un dispositivo a la línea de arbitraje determina su prioridad en el uso del bus.
- II. No se requiere un árbitro.

### PROBLEMA 1.17

Un bus con arbitraje centralizado por encadenamiento (*daisy-chaining*) y con un único nivel de prioridad tiene conectados 4 dispositivos (d1, d2, d3, d4). El dispositivo d1 es el que se encuentra conectado más próximo al árbitro, a continuación está conectado d2, seguidamente d3 y finalmente d4. La fracción de la capacidad utilizada por los dispositivos d1, d2, d3 y d4 es 0.2, 0.35, 0.1 y 0.25 respectivamente. Indique cuál es la *capacidad sobrante* del dispositivo d3. (Nota: Se define la *capacidad sobrante* de un dispositivo como 1 menos la suma de las fracciones de las capacidades utilizadas por todos los dispositivos que tienen una prioridad superior).

### SOLUCIÓN PROBLEMA 1.17

En Bus con arbitraje centralizado por encadenamiento (*daisy-chain*) y con único nivel de prioridad, el dispositivo de mayor prioridad es el conectado más cerca del árbitro, en este caso d1. El siguiente dispositivo más prioritario sería d2, luego d3 y finalmente d4.

La *capacidad sobrante* de un dispositivo se define como 1 menos la suma de las fracciones de las capacidades utilizadas por todos los dispositivos que tienen una prioridad superior. Se pide la capacidad sobrante del dispositivo d3, que se va a denotar con  $Cs_3$ , y se sabe que los dispositivos d1 y d2 son más prioritarios que éste, luego aplicando la definición se obtiene:

$$Cs_3 = 1 - (2.0 + 35.0) = 45.0$$

### **PROBLEMA 1.24**

En un bus con arbitraje centralizado

- A) Cualquier dispositivo puede decidir en qué momento toma el control del bus.
- B) La temporización debe ser asíncrona.
- C) Todos los dispositivos actúan de forma cooperativa para compartir el bus.
- D) **Todas las afirmaciones anteriores son falsas.**

## PROBLEMA 1.25

El formato de instrucción de un procesador hipotético es el siguiente: los 4 bits más significativos almacenan el código de operación y los restantes 12 bits contienen la dirección de memoria.

El formato de almacenamiento de los números enteros es el siguiente: el bit más significativo contiene el signo (0: positivo; 1: negativo), y los 15 bits restantes la magnitud.

La CPU contiene tres registros internos: el contador de programa (PC), el registro de instrucción (IR) y el acumulador (AC).

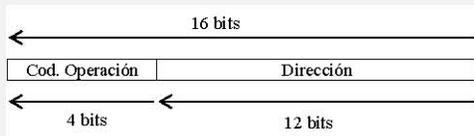
La lista de los códigos de operación incluye los siguientes: 0001<sub>2</sub> (cargar AC desde memoria), 0010<sub>2</sub> (almacenar AC en memoria), 1000<sub>2</sub> (restar al AC de memoria), 1001<sub>2</sub> (sumar al AC de memoria).

Si las posiciones de memoria A0F<sub>16</sub> y 3A0<sub>16</sub> contienen respectivamente: X e Y, determinar cuál es el contenido de la posición de memoria 3A0<sub>16</sub> una vez ejecutadas las cuatro instrucciones siguientes: 13A0<sub>16</sub>, 9A0F<sub>16</sub>, 93A0<sub>16</sub>, 2A0F<sub>16</sub>.

## SOLUCIÓN PROBLEMA

### DATOS

- Formato de instrucción:



- El formato de almacenamiento de los números enteros es el siguiente:  
Bit más significativo: Signo (0 positivo, 1 negativo).  
15 bits restantes: Magnitud
- Registros de la CPU: Contador del programa (PC), registro de instrucción (IR) y acumulador (AC).
- Lista parcial de códigos de operación:  
0001<sub>2</sub> (Ac ← [M])  
0010<sub>2</sub> (M ← [Ac])  
1000<sub>2</sub> (Ac ← [Ac] - [M])  
1001<sub>2</sub> (Ac ← [Ac] + [M])
- Contenido de ciertas posiciones de memoria: [A0F<sub>16</sub>] = X, [3A0<sub>16</sub>] = Y
- Averiguar el contenido de la posición de memoria 3A0<sub>16</sub> tras ejecutar las instrucciones 13A0<sub>16</sub>, 9A0F<sub>16</sub>, 93A0<sub>16</sub>, 2A0F<sub>16</sub>

Para resolver este problema se va ir analizando cada una de las instrucciones que se ejecutan:

1) La instrucción 13A016 tiene por código de operación  $1_{16}=0001_2$ , y como dirección 3A016 se ejecuta la siguiente instrucción:  $Ac \leftarrow [3A016]$

Luego, se estaría cargando en el acumulador el contenido de la dirección de memoria 3A016, es decir, Y.

2) La instrucción 9A0F16 tiene por código de operación  $9_{16}=1001_2$ , y como dirección A0F16 se ejecuta la siguiente instrucción:  $Ac \leftarrow [Ac] + [A0F16]$

Luego, se estaría cargando en el acumulador el resultado que se obtiene al sumar el contenido del acumulador  $[Ac]=Y$  con  $[A0F16]=X$ . Que da como resultado  $Y+X$

3) La instrucción 93A016 tiene por código de operación  $9_{16}=1001_2$ , y como dirección 3A016 se ejecuta la siguiente instrucción:  $Ac \leftarrow [Ac] + [3A016]$

Luego, se estaría cargando en el acumulador el resultado que se obtiene al sumar el contenido del acumulador  $[Ac]=Y+X$  con  $[3A016]=Y$ . Que da como resultado  $2Y+X$

4) La instrucción 2A0F16 tiene por código de operación  $2_{16}=0010_2$ , y como dirección A0F16 se ejecuta la siguiente instrucción:  $A0F16 \leftarrow [Ac]$

Luego, se almacena en la dirección de memoria A0F16 el contenido del acumulador  $[Ac]=2Y+X$ .

Este programa por tanto no modifica el contenido inicial de la posición de memoria 3A016. Luego la respuesta al problema es que el contenido de 3A016 tras ejecutar este programa sigue siendo el mismo, es decir,  $[3A016]=Y$