

ESTRUCTURA Y TECNOLOGIA DE COMPUTADORES II

TEMA 1: *Estructuras de interconexión de un computador*

SOLUCIÓN A LOS PROBLEMAS PROPUESTOS EN EXAMEN

**Curso 2007-2008
Jose Manuel Díaz Martínez
Tutor de la asignatura ETC-II
Coslada (Madrid)**

CONTENIDO

SOLUCIÓN PROBLEMA 1.1	3
SOLUCIÓN PROBLEMA 1.2	3
SOLUCIÓN PROBLEMA 1.3	4
SOLUCIÓN PROBLEMA 1.4	5
SOLUCIÓN PROBLEMA 1.5	5
SOLUCIÓN PROBLEMA 1.6	6
SOLUCIÓN PROBLEMA 1.7	7
SOLUCIÓN PROBLEMA 1.8	7
SOLUCIÓN PROBLEMA 1.9	8
SOLUCIÓN PROBLEMA 1.10	8
SOLUCIÓN PROBLEMA 1.11	9
SOLUCIÓN PROBLEMA 1.12	11
SOLUCIÓN PROBLEMA 1.13	12
SOLUCIÓN PROBLEMA 1.14	16
SOLUCIÓN PROBLEMA 1.15	17
SOLUCIÓN PROBLEMA 1.16	18
SOLUCIÓN PROBLEMA 1.17	19
SOLUCIÓN PROBLEMA 1.18	19
SOLUCIÓN PROBLEMA 1.19	21
SOLUCIÓN PROBLEMA 1.20	22
SOLUCIÓN PROBLEMA 1.21	22
SOLUCIÓN PROBLEMA 1.22	23
SOLUCIÓN PROBLEMA 1.23	23
SOLUCIÓN PROBLEMA 1.24	25
SOLUCIÓN PROBLEMA 1.25	26

SOLUCIÓN PROBLEMA 1.1

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. Que el contenido de una palabra de memoria sea considerado un dato o una instrucción dependerá del momento del ciclo de instrucción en que se realice el acceso a memoria.
- Si el acceso a memoria se realiza en el ciclo de búsqueda de la instrucción el contenido de la posición de memoria será considerada una instrucción que se almacenará en el registro de instrucción (IR).
- Si el acceso a memoria se realiza en las subfases del ciclo de ejecución (búsqueda de operandos, almacenamiento de resultados, etc) de la instrucción el contenido de la posición de memoria será considerado un dato.
- 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**.

SOLUCIÓN PROBLEMA 1.2

DATOS

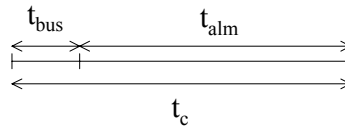
- Memoria de 10 módulos independientes conectados a un mismo bus de memoria.
- Cada bus ocupa el bus durante el 20% del ciclo de escritura.
- El 80 % del tiempo de ciclo restante el módulo almacena el dato internamente.
- Las operaciones de estos módulos pueden solaparse.

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

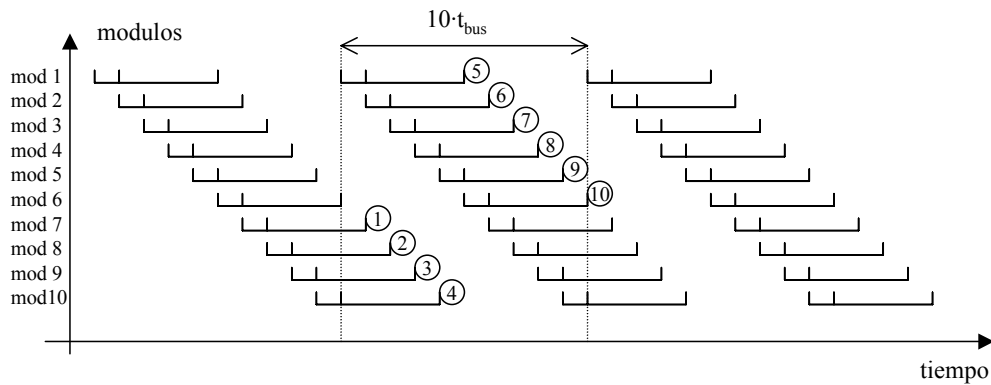
$$v_{max} = \frac{1}{t_c} \text{ (pal/seg)}$$

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

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

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

SOLUCIÓN PROBLEMA 1.3

DATOS

- Computador de Von Neuman
- Longitud de palabra de 16 bits.
- Instrucciones máquina ocupan 16 o 32 bits (1 o 2 palabras)
- Código de operación ocupa un campo de 7 bits.

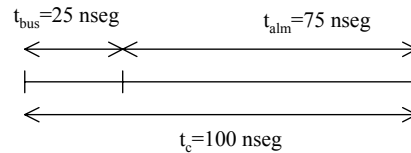
Una de las características de la arquitectura de Von Neumann es precisamente el tener un único espacio de memoria utilizado tanto para almacenar datos como para almacenar instrucciones. La afirmación I es falsa.

Cuando se habla de instrucciones máquina no tiene sentido hablar de señales de control, sino de códigos de operación, formato de instrucciones, etc. Sin embargo, si nos referimos a microinstrucciones, sí tiene sentido. La afirmación II es falsa.

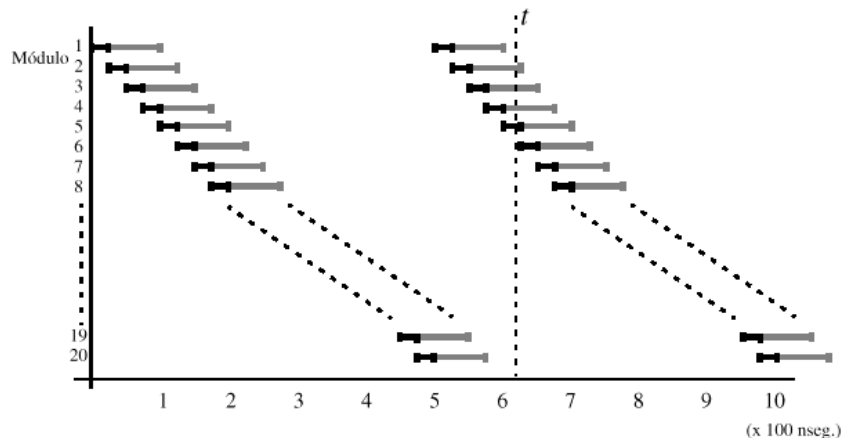
SOLUCIÓN PROBLEMA 1.4

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, como 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:

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

SOLUCIÓN PROBLEMA 1.5

Afirmación I:

- Es **falsa**, ya que en la técnica de multiplexación en el tiempo se comparten las mismas líneas de un bus, para enviar datos o direcciones, por lo que es imposible mantener en paralelo una dirección y un dato.

Afirmación II:

- Es **falsa**, ya que si solo una puerta triestado estuviera en estado de alta impedancia, el resto tendrían sus salidas activas y se produciría un error al cargar la línea del bus con más de una entrada. Lo correcto sería justamente lo contrario, es decir, todas las puertas menos una deben estar en estado de alta impedancia.

SOLUCIÓN PROBLEMA 1.6**DATOS**

- En un Bus se han conectado la memoria principal y cinco discos iguales.
- Tiempo de ciclo de M_p es $t_c=400\text{nseg}$.
- La velocidad de transferencia de cada disco es $v_T=B$ (bits/seg).
- La longitud de una palabra es de 32 bits.
- Cada transmisión sobre el bus necesita 500 nseg para enviar el dato y las diferentes ordenes del protocolo.
- Se desea usar un Bus con temporización síncrona.

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.

En este problema se sabe que el bus tiene conectado la memoria principal M_p y cinco discos duros. De acuerdo con el razonamiento anterior, si se desea utilizar un bus con *temporización síncrona* todos los elementos conectados a él deben tener velocidades de transferencia parecidas. Por ello la velocidad de transferencia de los discos B debe ser lo más cercana posible a la velocidad de transferencia v_T de la M_p que es el dispositivo más rápido conectado. Es decir, se debe cumplir

$$B \approx v_T$$

Por lo tanto la resolución de este problema se reduce a calcular la velocidad de transferencia de la M_p , que viene dada, supuesto que es de acceso aleatorio, por la inversa del tiempo de ciclo:

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

Cada palabra tiene 32 bits por lo tanto, la velocidad de transferencia v_T de la M_p se puede expresar en (bits/seg):

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

Luego

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

SOLUCIÓN PROBLEMA 1.7

DATOS

- Bus con arbitraje distribuido

En un esquema distribuido cada módulo contiene la lógica de control suficiente para poder acceder al bus y todos ellos actúan de forma cooperativa para compartir el recurso.

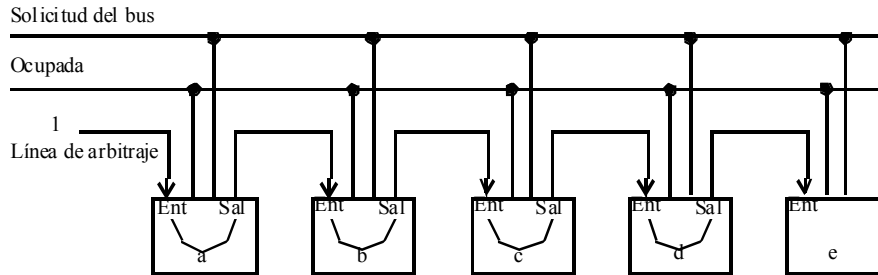


Figura: Bus con arbitraje distribuido

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 módulo, no puede ser utilizado el bus por ningún otro módulo pese a poseer una mayor prioridad.

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.

SOLUCIÓN PROBLEMA 1.9

DATOS

- Bus con arbitraje distribuido

En un esquema distribuido cada módulo contiene la lógica de control suficiente para poder acceder al bus y todos ellos actúan de forma cooperativa para compartir el recurso.

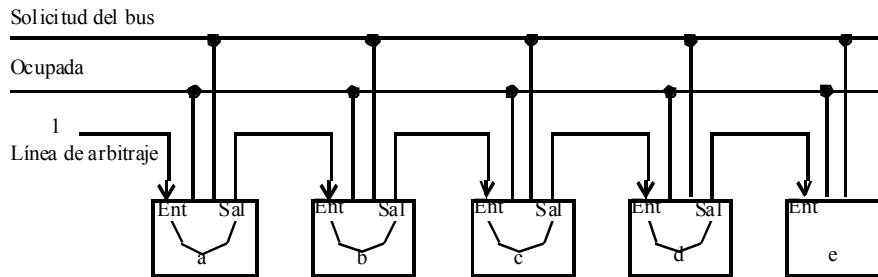


Figura: Bus con arbitraje distribuido

Afirmación I:

- Es **VERDADERA**, ya que en este método de arbitraje el dispositivo más cercano a la línea de arbitraje es aquel que tiene la máxima prioridad de uso del bus.

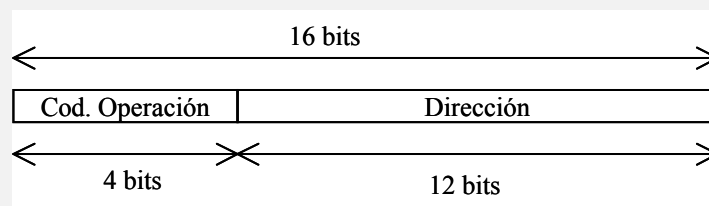
Afirmación II:

- Es **VERDADERA**, ya que en este método de arbitraje cada módulo contiene la lógica de control suficiente para poder acceder al bus y todos ellos actúan de forma cooperativa para compartir el recurso.

SOLUCIÓN PROBLEMA 1.10

DATOS

- Instrucciones y datos tienen una longitud de 16 bits.
- Formato de instrucción



- Códigos de operación:
0011: $Ac \leftarrow [M]$
0101: $M \leftarrow [Ac]$
0110: $Ac \leftarrow [Ac] + [M]$

Para la resolución de este problema se va a utilizar la siguiente nomenclatura:

M indica una dirección de memoria.

[M] hace referencia al contenido de la dirección de memoria M

Ac indica la dirección del acumulador.

[Ac] hace referencia al contenido de la dirección de memoria M

La equivalencia en hexadecimal de los códigos de operación es:

3_{16} $Ac \leftarrow [M]$

5_{16} $M \leftarrow [Ac]$

6_{16} $Ac \leftarrow [Ac] + [M]$

De acuerdo con el enunciado del problema la operación que se desea realizar es sumar el contenido de la dirección de memoria $3A5_{16}$ al contenido de la dirección de memoria $3B9_{16}$ y almacenar el resultado en la dirección de memoria $3A5_{16}$. Es decir, de forma esquemática dicha operación se puede representar de la siguiente forma:

$$3A5_{16} \leftarrow [3A5_{16}] + [3B9_{16}]$$

Esta operación se puede realizar mediante la siguiente secuencia de instrucciones:

Opción 1:

- 1) $33A5_{16}$ que indica la operación $Ac \leftarrow [3A5_{16}]$
- 2) $63B9_{16}$ que indica la operación $Ac \leftarrow [Ac] + [3B9_{16}]$
- 3) $53A5_{16}$ que indica la operación $3A5_{16} \leftarrow [Ac]$

Opción 2:

- 1) $33B9_{16}$ que indica la operación $Ac \leftarrow [3B9_{16}]$
- 2) $63A5_{16}$ que indica la operación $Ac \leftarrow [Ac] + [3A5_{16}]$
- 3) $53A5_{16}$ que indica la operación $3A5_{16} \leftarrow [Ac]$

Luego la respuesta correcta es la C.

SOLUCIÓN PROBLEMA 1.11

DATOS

- Memoria de 4 módulos independientes conectados a un bus de memoria común
- Cada módulo ocupa el bus durante 50 nseg del ciclo de escritura.
- Durante esos 50 nseg y los 200 ns siguientes, el módulo de memoria direccionado ejecuta 1 ciclo aceptando y almacenando el dato.
- Las operaciones de estos módulos pueden solaparse, pero solo puede haber una petición por instante de tiempo.
- t_c es el tiempo de ciclo de escritura.
- ¿Cual es la velocidad máxima de transferencia ?

[Ver problema 1-3 del libro de problemas]. La situación planteada por el problema se ilustra en la Figura 1.11.1, donde los M_i representan los 4 módulos de memoria propuestos.

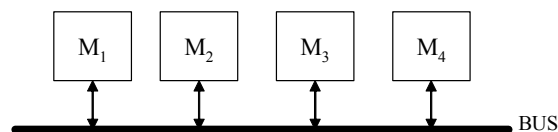


Figura 1.11.1: Esquema de los módulos de memoria.

El ciclo de escritura en los módulos de memoria se muestra gráficamente en la Figura 1.11.2. En la primera parte del ciclo el módulo de memoria utiliza durante $t_{bus}=50\text{ ns}$ el bus en exclusiva, pero en la segunda parte se realiza el almacenamiento del dato $t_{alm}=200\text{ ns}$. En esta segunda parte del ciclo de escritura cualquier otro módulo puede hacer uso del bus.

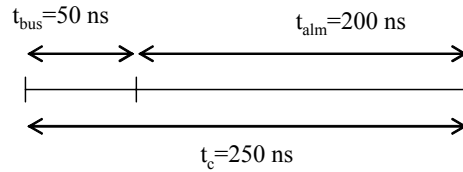


Figura 1.11.2: Ciclo de escritura de los módulos de memoria.

El tiempo de ciclo de escritura correspondiente, t_c , es la suma del tiempo de ocupación del bus más el tiempo de almacenamiento:

$$t_c = t_{bus} + t_{alm} = 50 + 200 = 250\text{ ns}$$

Utilizando los cuatro 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 en la Figura 1.11.3:

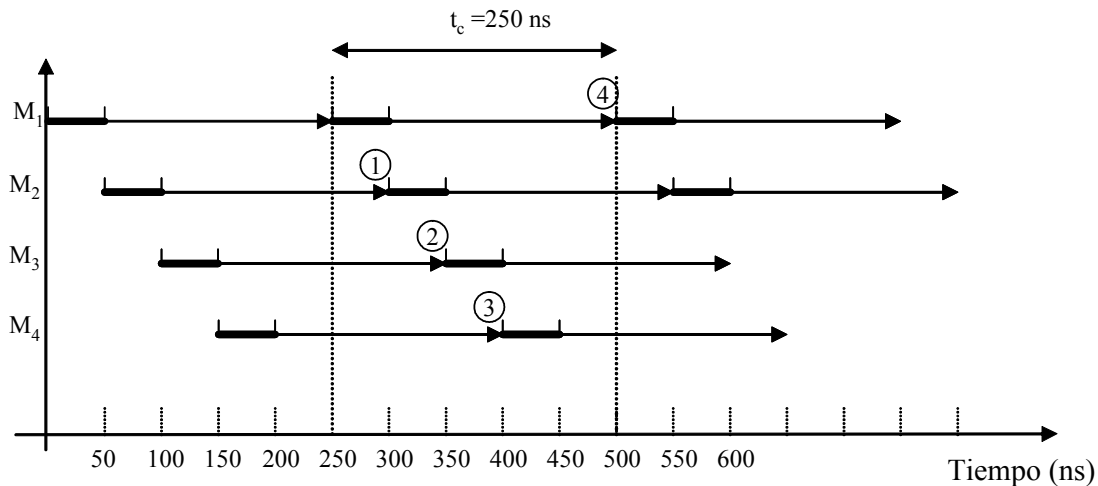


Figura 1.11.3: Máximo solapamiento con $t_c=250\text{ ns}$

Se observa en esta figura como el módulo M_1 ocupa el bus hasta $t=50\text{ ns}$, a continuación lo utilizar M_2 hasta $t=100\text{ ns}$, luego lo utiliza M_3 hasta $t=150\text{ ns}$ y posteriormente lo utiliza M_4 hasta $t=200\text{ ns}$. Obsérvese que en este instante de tiempo el módulo M_1 no ha finalizado la operación de escritura de una palabra, por lo que no puede comenzar otra operación de escritura hasta $t=250\text{ ns}$, que es cuando acaba su ciclo.

Luego en $t=250\text{ ns}$ se empezaría por tanto otra ronda de uso del bus por parte de M_1 que lo ocupa hasta $t=300\text{ ns}$, a continuación lo utilizar M_2 hasta $t=350\text{ ns}$, luego lo utiliza M_3 hasta $t=400\text{ ns}$ y posteriormente lo utiliza M_4 hasta $t=450\text{ ns}$.

En el intervalo de tiempo comprendido entre $t=250\text{ ns}$ y $t=500\text{ ns}$, se han escrito las siguientes palabras en el sistema:

- 1) El módulo M_2 finaliza la escritura de una palabra en $t=300\text{ ns}$.
- 2) El módulo M_3 termina de escribir una palabra en $t=350\text{ ns}$.
- 3) El módulo M_4 termina de escribir una palabra en $t=400\text{ ns}$.
- 4) El módulo M_1 termina de escribir una palabra en $t=500\text{ ns}$.

5)

Luego en un tiempo de ciclo $t_c=250$ ns se escriben **4 palabras**, una en cada módulo. Luego la velocidad máxima de escritura (palabras/s) será:

$$v_{T\max} = \frac{4}{t_c}$$

SOLUCIÓN PROBLEMA 1.12

DATOS

- El computador ejecuta 50 millones de instrucciones por segundo.
- Las instrucciones están almacenadas en la memoria principal.
- En la tabla del enunciado se dan los tipos de instrucciones que soporta el computador y los porcentajes de utilización de cada una de ellos.
- Cada instrucción ocupa 1 palabra de memoria.

Para resolver este problema se debe recordar que a la hora de ejecutar una instrucción el computador realiza dos pasos o fases: la fase de búsqueda y la fase de ejecución.

Fase o ciclo de búsqueda

De acuerdo con el enunciado todas las instrucciones se encuentran en la memoria principal, luego en la fase de búsqueda independientemente del tipo de instrucción, se realizará un acceso a la memoria principal para leer la palabra que contiene la instrucción.

Luego si el computador ejecuta $5 \cdot 10^7$ instrucciones por segundo se realizarán debido a la fase de búsqueda de cada instrucción N_B accesos por segundo.

$$N_B = 5 \cdot 10^7 \left(\frac{\text{instrucciones}}{s} \right) * 1 \left(\frac{\text{acceso}}{\text{instruccion}} \right) = 5 \cdot 10^7 \left(\frac{\text{accesos}}{s} \right)$$

Fase o ciclo de ejecución

Ahora hay que considerar únicamente aquellas instrucciones que utilizan operandos que se encuentran en la memoria principal, es el caso de LOAD $Acum \leftarrow M[dir]$. De la tabla se sabe que el porcentaje de utilización de estas instrucciones es del 30 %.

Además hay que considerar aquellas instrucciones que almacenan el resultado en la memoria principal, este es el caso de STORE $M[dir] \leftarrow Acum$. De la tabla se sabe que el porcentaje de utilización de estas instrucciones es del 10 %.

En ambos casos sólo hay que hacer un acceso a memoria, bien para leer o bien para escribir.

Luego si el computador ejecuta $5 \cdot 10^7$ instrucciones por segundo se realizarán debido a la fase de ejecución de cada instrucción N_E accesos por segundo:

$$N_E = (0.3 \cdot 5 \cdot 10^7 + 0.1 \cdot 5 \cdot 10^7) \cdot \left(\frac{\text{instrucciones}}{s} \right) * 1 \left(\frac{\text{accesos}}{\text{instruccion}} \right) = 0.4 \cdot 5 \cdot 10^7 \left(\frac{\text{accesos}}{s} \right)$$

Numero de accesos a memoria por segundo:

El número total de accesos a memoria por segundo vendrá dado por la suma de N_B y de N_E :

$$N = N_B + N_E = 5 \cdot 10^7 + (0.4 \cdot 10^7) = (1 + 0.4) \cdot 5 \cdot 10^7 = 1.4 \cdot 5 \cdot 10^7 = 70 \cdot 10^6 \cdot \frac{\text{accesos}}{s}$$

SOLUCIÓN PROBLEMA 1.13**DATOS**

- Memoria de N módulos independientes conectados a un bus de memoria común
- Cada módulo ocupa el bus durante 20 nseg del ciclo de escritura.
- Durante esos 20 nseg y los 80 ns siguientes, el módulo de memoria direccionado ejecuta 1 ciclo aceptando y almacenando el dato.
- Las operaciones de estos módulos pueden solaparse, pero solo puede haber una petición por instante de tiempo.
- La velocidad máxima de escritura de datos en memoria es $v_{Tmax}=0.05$ palabras/nseg.
- ¿Cual es el número máximo de módulos N conectados al bus?

[Este problema es una variante del **problema 1.11**].

La situación planteada por el problema se ilustra en la Figura 1.13.1, donde los M_i $i=1,...,N$ representan los N módulos de memoria propuestos.

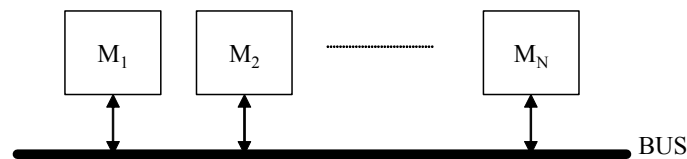


Figura 1.13.1: Esquema de los módulos de memoria.

El ciclo de escritura en los módulos de memoria se muestra gráficamente en la Figura 1.13.2. En la primera parte del ciclo el módulo de memoria utiliza durante $t_{bus}=20ns$ el bus en exclusiva, pero en la segunda parte se realiza el almacenamiento del dato $t_{alm}=80 ns$. En esta segunda parte del ciclo de escritura cualquier otro módulo puede hacer uso del bus.

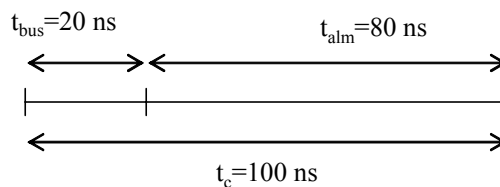


Figura 1.13.2: Ciclo de escritura de los módulos de memoria.

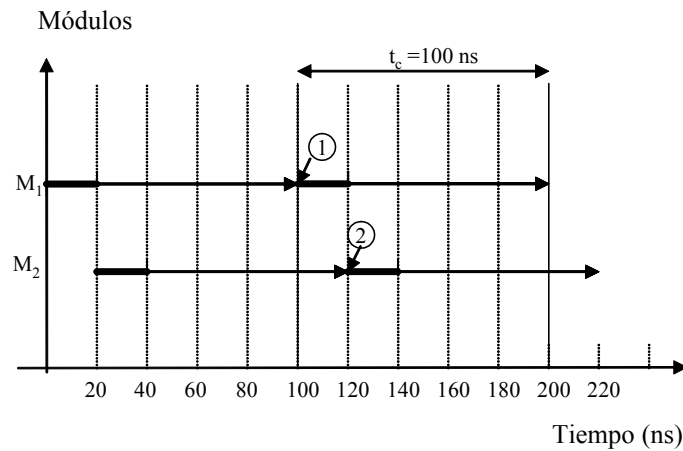
El tiempo de ciclo de escritura correspondiente, t_c , es la suma del tiempo de ocupación del bus más el tiempo de almacenamiento:

$$t_c = t_{bus} + t_{alm} = 20 + 80 = 100 \text{ ns}$$

Este problema puede ser resuelto de dos formas distintas:

FORMA GRAFICA: Se van a solapar al máximo los ciclos de escritura para intentar que en todo momento uno de los módulos esté accediendo al bus.

- Si el número de módulos es $N=2$ se tiene el siguiente esquema:

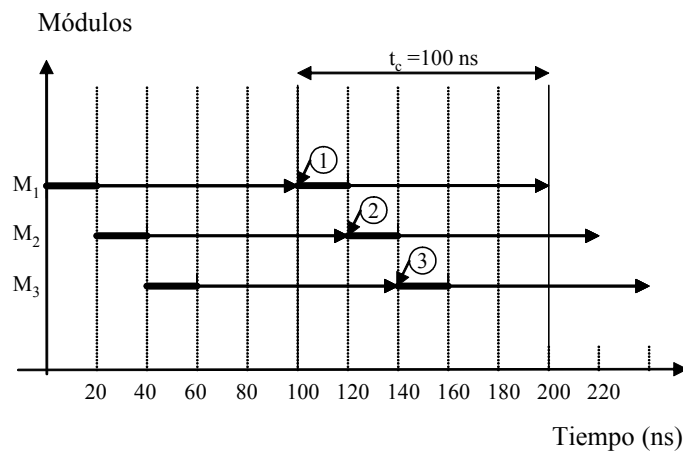


Se observa que en un periodo de tiempo igual a $t_c=100$ ns se escriben dos palabras. Luego la velocidad máxima del sistema es:

$$v_{T\max} = \frac{2}{100} = 0.02 \text{ (palabras/nseg)}$$

Luego con dos módulos no es posible alcanzar la velocidad deseada de 0.05 (palabras/nseg).

- Si el número de módulos es $N=3$ se tiene el siguiente esquema:

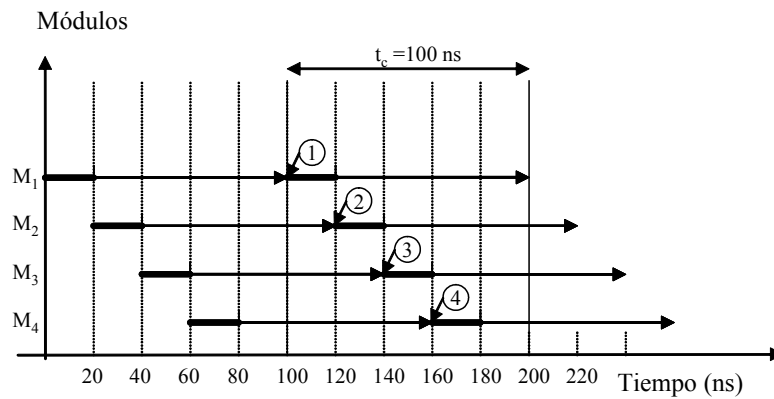


Se observa que en un periodo de tiempo igual a $t_c=100$ ns se escriben tres palabras. Luego la velocidad máxima del sistema es:

$$v_{T\max} = \frac{3}{100} = 0.03 \text{ (palabras/nseg)}$$

Luego con tres módulos no es posible alcanzar la velocidad deseada de 0.05 (palabras/nseg).

- Si el número de módulos es $N=4$ se tiene el siguiente esquema:

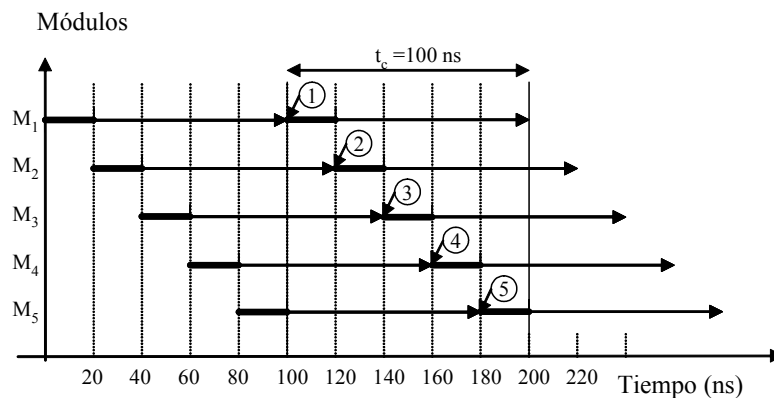


Se observa que en un periodo de tiempo igual a $t_c=100$ ns se escriben cuatro palabras. Luego la velocidad máxima del sistema es:

$$v_{T_{\max}} = \frac{4}{100} = 0.04 \text{ (palabras/nseg)}$$

Luego con cuatro módulos no es posible alcanzar la velocidad deseada de 0.05 (palabras/nseg).

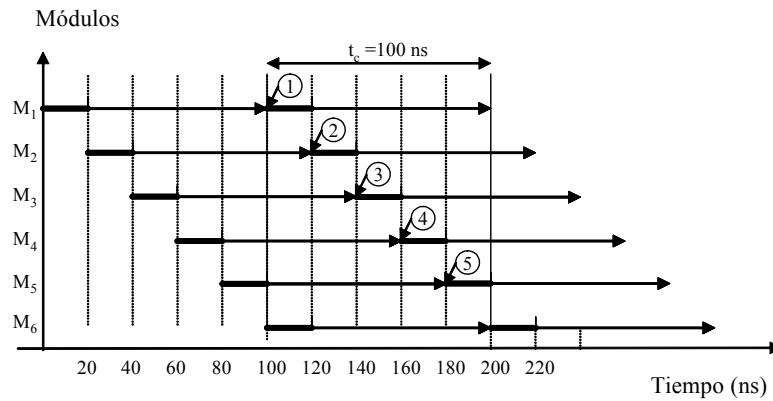
- Si el número de módulos es $N=5$ se tiene el siguiente esquema:



Se observa que en un periodo de tiempo igual a $t_c=100$ ns se escriben cinco palabras. Luego la velocidad máxima del sistema es:

$$v_{T_{\max}} = \frac{5}{100} = 0.05 \text{ (palabras/nseg)}$$

Es decir, a partir de cinco módulos es posible alcanzar la velocidad deseada de 0.05 (palabras/nseg). Vamos a comprobar esta afirmación suponiendo que el número de módulos es $N=6$ se tiene el siguiente esquema:



Se observa que en un periodo de tiempo igual a $t_c = 100$ ns se siguen escribiendo cinco palabras. En conclusión, si $N \geq 5$ se obtendrá la velocidad deseada de 0.05 (palabras/nseg).

FORMA ANALITICA

La velocidad máxima de escritura v_{Tmax} (palabras/nseg) viene dada como el cociente entre el número máximo de palabras N escritas durante un cierto intervalo de tiempo, por ejemplo, el tiempo de ciclo t_c (nseg). Para obtener este valor máximo de palabras escritas se debe solapar al máximo los ciclos de escritura para intentar que en todo momento uno de los módulos esté accediendo al bus, es decir, cada una de las N palabras se debe escribir en un módulo de memoria distinto. Por lo tanto:

$$v_{Tmax} = \frac{N}{t_c}$$

Si se despeja N se obtiene:

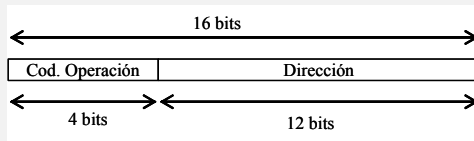
$$N = v_{Tmax} \cdot t_c = 0.05(\text{pal} / \text{nseg}) \cdot 100(\text{nseg}) = 5 \text{ palabras}$$

Por lo tanto, hay que escribir 5 palabras en un tiempo de ciclo para obtener la velocidad máxima de escritura que se indica en el enunciado. Esto sólo es posible si el sistema posee 5 o más módulos de memoria conectados al bus, supuesto el máximo solapamiento.

SOLUCIÓN PROBLEMA 1.14

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:
0001₂ (Ac ← [M])
0010₂ (M ← [Ac])
0110₂ (Ac ← [Ac] - [M])
1111₂ (Ac ← [Ac] + [M])
- Contenido de ciertas posiciones de memoria: [803]₁₆ = 0032₁₆, [804]₁₆ = 1011₁₆
- Averiguar el contenido de la posición de memoria 805₁₆ tras ejecutar las instrucciones 1803₁₆ F804₁₆ 2805₁₆

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

1) La instrucción 1803₁₆ tiene por código de operación 1₁₆ = 0001₂, y como dirección 803₁₆ se ejecuta la siguiente instrucción:

$$Ac \leftarrow [803]_{16}$$

Luego, se estaría cargando en el acumulador el contenido de la dirección de memoria 803₁₆, es decir, 0032₁₆.

2) La instrucción F804₁₆ tiene por código de operación F₁₆ = 1111₂, y como dirección 804₁₆ se ejecuta la siguiente instrucción:

$$Ac \leftarrow [Ac] + [804]_{16}$$

Luego, se estaría cargando en el acumulador el resultado que se obtiene al sumar el contenido del acumulador [Ac] = 0032₁₆ con [804]₁₆ = 1011₁₆. Que da como resultado 1043₁₆

3) La instrucción 2805₁₆ tiene por código de operación 2₁₆ = 0010₂, y como dirección 805₁₆ se ejecuta la siguiente instrucción:

$$805_{16} \leftarrow [Ac]$$

Luego, se almacena en la dirección de memoria 805₁₆ el contenido del acumulador [Ac] = 1043₁₆. En consecuencia la respuesta a este problema es [805]₁₆ = **1043₁₆**.

SOLUCIÓN PROBLEMA 1.15

DATOS

- El computador ejecuta 100 millones de instrucciones por segundo.
- Las instrucciones están almacenadas en la memoria principal.
- En la tabla del enunciado se dan los tipos de instrucciones que soporta el computador y los porcentajes de utilización de cada una de ellos.
- Las instrucciones LOAD y STORE ocupan 2 palabras de memoria.
- El resto de instrucciones sólo ocupa 1 palabra de memoria

[Este problema es similar al **Problema 1.12**]

Fase o ciclo de búsqueda

De acuerdo con la Tabla del enunciado el uso de las instrucciones LOAD y STORE es del 40 % (20+20). Por lo tanto, el número de instrucciones de estos tipos que se ejecutan por segundo será el 40% del total de instrucciones, es decir, $0.4 \cdot 10^8$ instrucciones/s. Además del enunciado se sabe que estos tipos de instrucciones ocupan cada una 2 palabras/instrucción. Luego se realizarán dos accesos a la memoria principal para leer las dos palabras que contienen una instrucción LOAD o STORE.

El número de accesos por segundo $N_B(\text{LOAD \& STORE})$ que se realizarán a la memoria principal en la fase de búsqueda de estas instrucciones es:

$$N_B(\text{LOAD} - \text{STORE}) = 0.4 \cdot 10^8 \left(\frac{\text{instrucciones}}{s} \right) * 2 \left(\frac{\text{acceso}}{\text{instruccion}} \right) = 0.8 \cdot 10^8 \left(\frac{\text{accesos}}{s} \right)$$

Por otra parte el uso de las restantes instrucciones es del 60%(100-40). Luego el número de instrucciones de estos otros tipos que se ejecutan por segundo será $0.6 \cdot 10^8$ instruccione/s. Para leer estas instrucciones en memoria principal sólo hay que realizar un acceso. Entonces, el número de accesos por segundo $N_B(\text{OTRAS})$ que se realizarán a la memoria principal en la fase de búsqueda del resto de instrucciones es:

$$N_B(\text{Otras}) = 0.6 \cdot 10^8 \left(\frac{\text{instrucciones}}{s} \right) * 1 \left(\frac{\text{acceso}}{\text{instruccion}} \right) = 0.6 \cdot 10^8 \left(\frac{\text{accesos}}{s} \right)$$

Ya es posible calcular el número de accesos total por segundo que se realizarán a la memoria principal en la fase de búsqueda de todas las instrucciones:

$$N_B = N_B(\text{LOAD} - \text{STORE}) + N_B(\text{Otras}) = 0.8 \cdot 10^8 + 0.6 \cdot 10^8 = 1.4 \cdot 10^8 \left(\frac{\text{accesos}}{s} \right)$$

Fase o ciclo de ejecución

Ahora hay que considerar únicamente aquellas instrucciones que utilizan operandos que se encuentran en la memoria principal, es el caso de $\text{LOAD } Acum \leftarrow M[dir]$. De la tabla se sabe que el porcentaje de utilización de estas instrucciones es del 20 %.

Además hay que considerar aquellas instrucciones que almacenan el resultado en la memoria principal, este es el caso de $\text{STORE } M[dir] \leftarrow Acum$. De la tabla se sabe que el porcentaje de utilización de estas instrucciones es del 20 %.

En ambos casos sólo hay que hacer un acceso a memoria, bien para leer o bien para escribir.

Luego si el computador ejecuta 10^8 instrucciones por segundo se realizarán debido a la fase de ejecución de cada instrucción N_E accesos por segundo:

$$N_E = (0.2 \cdot 10^8 + 0.2 \cdot 10^8) \cdot \left(\frac{\text{instrucciones}}{s} \right) * 1 \left(\frac{\text{accesos}}{\text{instruccion}} \right) = 0.4 \cdot 10^8 \left(\frac{\text{accesos}}{s} \right)$$

Numero de accesos a memoria por segundo:

El número total de accesos a memoria por segundo vendrá dado por la suma de N_B y de N_E

$$N = N_B + N_E = 1.4 \cdot 10^8 + 0.4 \cdot 10^8 = 1.8 \cdot 10^8 = 180 \cdot 10^6 \left(\frac{\text{accesos}}{s} \right)$$

SOLUCIÓN PROBLEMA 1.16

DATOS

- Se emplean 4 ciclos de reloj para las instrucciones de cálculo.
- Se emplean 8 ciclos para el resto de instrucciones.
- La frecuencia del reloj es $f=8 \cdot 10^6$ ciclos/s
- Programa con 1000 instrucciones de las cuales el 20 % son de cálculo

I) En primer lugar se va a determinar el tiempo t_{IC} que se emplea en ejecutar una instrucción de cálculo y el tiempo que se emplea en ejecutar una instrucción de otro tipo t_{IR} .

$$t_{IC} = \frac{4(\text{ciclos} / \text{instr})}{f(\text{ciclos} / s)} = \frac{4}{8 \cdot 10^6} (s / \text{instr}) = 0.5 \cdot 10^{-6} (s / \text{instr}) = 0.5 (\mu s / \text{instr})$$

$$t_{IR} = \frac{8(\text{ciclos} / \text{instr})}{f(\text{ciclos} / s)} = \frac{8}{8 \cdot 10^6} (s / \text{instr}) = 10^{-6} (s / \text{instr}) = 1 (\mu s / \text{instr})$$

A continuación, en el programa de $N_I=1000$ instrucciones se van a determinar el número de instrucciones de cálculo N_{IC} y el número de instrucciones que hay de otro tipo N_{IR} :

$$N_{IC} = 0.2 \cdot N_I = 0.2 \cdot 1000 = 200 (\text{instr})$$

$$N_{IR} = N_I - N_{IC} = 1000 - 200 = 800 (\text{instr})$$

Luego el tiempo total que tarda en ejecutarse este programa será

$$T = N_{IC} \cdot t_{IC} + N_{IR} \cdot t_{IR} = 200 \cdot 0.5 + 800 \cdot 1 = 100 + 800 = 900 \mu s$$

II) Para calcular cuantos millones de instrucciones por segundo se pueden ejecutar como máximo hay que suponer el caso en que todas las instrucciones de un programa son del tipo que requiere menos ciclos de reloj para ejecutarse, en este caso las de cálculo. En ese caso, el número máximo de instrucciones se puede calcular como:

$$n_{I \max} = \frac{f(\text{ciclos} / s)}{4(\text{ciclos} / \text{instr})} = \frac{8 \cdot 10^6}{4} = 2 \cdot 10^6 \text{ instr} / s = 2 \text{ MIPS}$$

SOLUCIÓN PROBLEMA 1.17

DATOS

- Bus con arbitraje centralizado por encadenamiento (daisy-chain) y con único nivel de prioridad.
- Hay conectados al bus 4 dispositivos (d1, d2, d3, d4).
- El dispositivo d1 está conectado más cerca del árbitro, seguido de d2, de d3 y luego 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
- ¿Capacidad sobrante del dispositivo d3?

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:

$$Cs_3 = 1 - (0.2 + 0.35) = 0.45$$

SOLUCIÓN PROBLEMA 1.18

DATOS

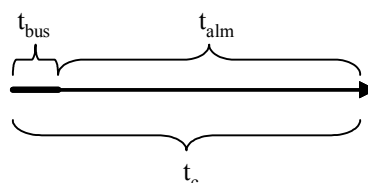
- Memoria de 10 módulos independientes conectados a un mismo bus de memoria.
- Cada bus ocupa el bus durante el 12.5% del ciclo de escritura.
- El 87.5 % del tiempo de ciclo restante el módulo almacena el dato internamente.
- Las operaciones de estos módulos pueden solaparse.

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

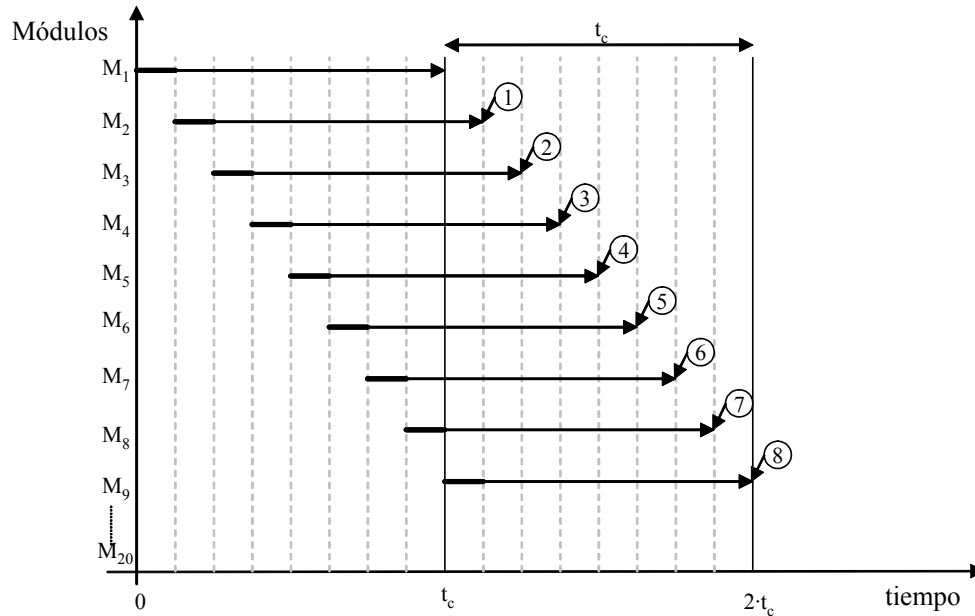
$$v_{\max} = \frac{1}{t_c} (\text{pal} / \text{seg})$$

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 $t_{\text{bus}} = 0.125 \cdot t_c$
- tiempo de almacenamiento interno del dato $t_{\text{alm}} = 0.875 \cdot t_c$



Utilizando los 20 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



Se observa que en un periodo de tiempo igual a t_c se escriben ocho palabras. Luego la velocidad máxima del sistema es:

$$v_{T\max} = \frac{8}{t_c} \text{ (palabras/seg)}$$

Que se puede expresar equivalentemente como:

$$v_{T\max} = 8 \cdot v_{\max}$$

Luego la velocidad de almacenamiento se ve incrementada **8 veces**.

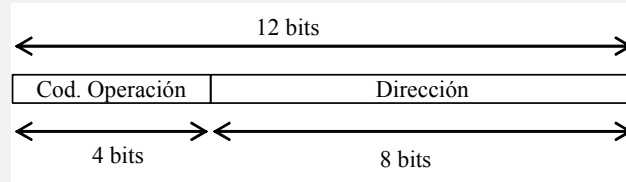
Nota: A este mismo resultado se hubiera podido llegar sin realizar la representación teniendo en cuenta el hecho de que para las hipótesis del enunciado se tiene que

$$\frac{t_c}{t_{bus}} = \frac{t_c}{0.125 \cdot t_c} = 8$$

Es decir en un ciclo de escritura es posible comenzar ocho operaciones de escritura.

SOLUCIÓN PROBLEMA 1.19**DATOS**

- Instrucciones y datos tienen una longitud de 12 bits.
- Formato de instrucción



- Códigos de operación:
- 0111: $Ac \leftarrow [M]$
 1101: $M \leftarrow [Ac]$
 1110: $Ac \leftarrow [Ac] - [M]$

Para la resolución de este problema se va a utilizar la siguiente nomenclatura:

M indica una dirección de memoria.

[M] hace referencia al contenido de la dirección de memoria M

Ac indica la dirección del acumulador.

[Ac] hace referencia al contenido de la dirección de memoria M

La equivalencia en hexadecimal de los códigos de operación es:

7_{16} $Ac \leftarrow [M]$

D_{16} $M \leftarrow [Ac]$

E_{16} $Ac \leftarrow [Ac] - [M]$

De acuerdo con el enunciado del problema la operación que se desea realizar es restar al contenido de la dirección de memoria $3B_{16}$ al contenido de la dirección de memoria $3A_{16}$ y almacenar el resultado en la dirección de memoria $3A_{16}$. Es decir, de forma esquemática dicha operación se puede representar de la siguiente forma:

$$3A_{16} \leftarrow [3B_{16}] - [3A_{16}]$$

Esta operación se puede realizar mediante la siguiente secuencia de instrucciones:

- 1) $73B_{16}$ que indica la operación $Ac \leftarrow [3B_{16}]$
- 2) $E3A_{16}$ que indica la operación $Ac \leftarrow [Ac] - [3A_{16}]$
- 3) $D3A_{16}$ que indica la operación $3A_{16} \leftarrow [Ac]$

Luego la respuesta correcta es la **A**.

SOLUCIÓN PROBLEMA 1.20

Afirmación I:

- En una transferencia de datos en un bus que utiliza la técnica de *multiplexación en el tiempo* de direcciones y datos se utiliza el mismo conjunto de líneas para transmitir direcciones y datos. En conclusión la afirmación es **FALSA**.

Afirmación II:

- Cuando las salidas de varias puertas triestado se encuentran conectadas a una misma línea de un bus todas las puertas menos una, aquella cuya entrada se desea volcar al bus, se deben encontrar en estado de alta impedancia. En conclusión la afirmación es **FALSA**.

SOLUCIÓN PROBLEMA 1.21

DATOS

- Bus con arbitraje distribuido

En un esquema distribuido cada módulo contiene la lógica de control suficiente para poder acceder al bus y todos ellos actúan de forma cooperativa para compartir el recurso.

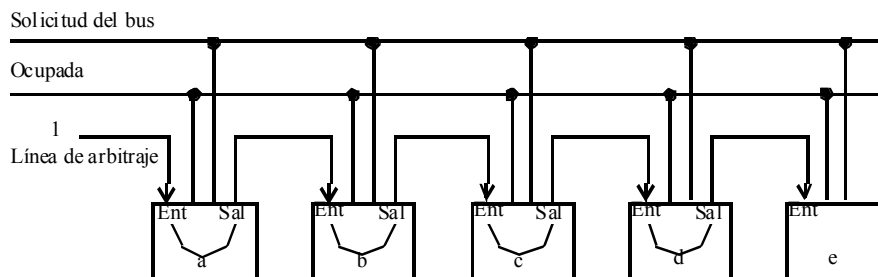


Figura: Bus con arbitraje distribuido

Afirmación I:

- Es **FALSA**, ya que en este método de arbitraje el dispositivo más cercano a la línea de arbitraje es aquel que tiene la máxima prioridad de uso del bus.

Afirmación II:

- Es **FALSA**, ya que un dispositivo aunque tenga mayor prioridad no puede hacer uso del bus hasta que el dispositivo que lo está utilizando deja de hacerlo.

SOLUCIÓN PROBLEMA 1.22

DATOS

- CPU con
 - Bus de direcciones de 16 bits.
 - Bus de datos de 8 bits
 - Un registro de 8 bits conectado al bus de datos y la unidad de control

De acuerdo con los datos aportados en el enunciado en este computador las direcciones son de 16 bits y los datos o instrucciones de 8 bits. Se van analizar cada uno de los registros propuestos en las soluciones:

- A) El PC contador de programa contiene la dirección de la próxima instrucción que se va a ejecutar luego su tamaño debe ser de 16 bits.
- B) El IR registro de instrucción contiene la instrucción actual que se va a ejecutar, luego su tamaño es de 8 bits.
- C) El MAR contiene la dirección de una posición de memoria principal, luego su tamaño es de 16 bits.

Por lo tanto de todos los registros propuestos el único que cumple la restricción de ser de 8 bits es el IR. Luego la respuesta correcta es la **B**.

SOLUCIÓN PROBLEMA 1.23

DATOS

- Memoria de N módulos independientes conectados a un bus de memoria común
- Cada módulo ocupa el bus durante 20 nseg del ciclo de escritura.
- Durante esos 20 nseg y los 80 ns siguientes, el módulo de memoria direccionado ejecuta 1 ciclo aceptando y almacenando el dato.
- Las operaciones de estos módulos pueden solaparse, pero solo puede haber una petición por instante de tiempo.

[Este problema es una variante del problema 1.13]

La situación planteada por el problema se ilustra en la Figura 1.23.1, donde los M_i $i=1,...,N$ representan los N módulos de memoria propuestos.

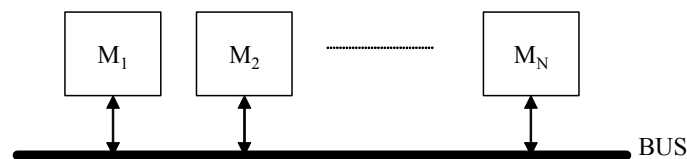


Figura 1.13.1: Esquema de los módulos de memoria.

El ciclo de escritura en los módulos de memoria se muestra gráficamente en la Figura 1.23.2. En la primera parte del ciclo el módulo de memoria utiliza durante $t_{bus}=20ns$ el bus en exclusiva, pero en la segunda parte se realiza el almacenamiento del dato $t_{alm}=80 ns$. En esta segunda parte del ciclo de escritura cualquier otro módulo puede hacer uso del bus.

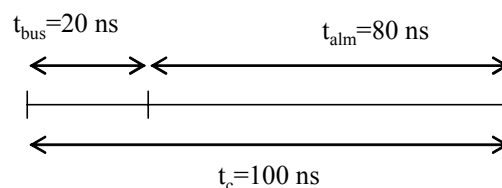


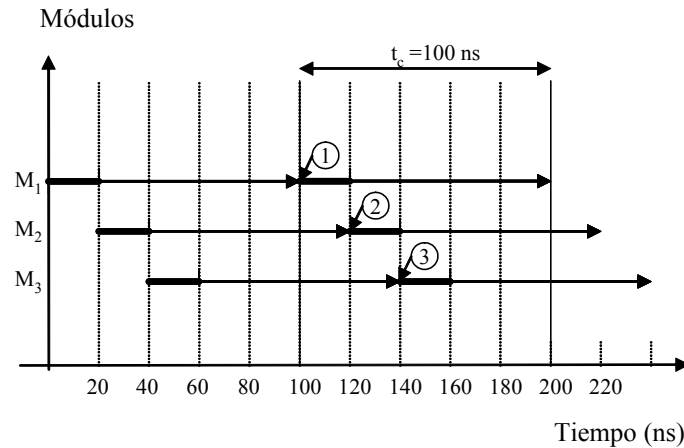
Figura 1.23.2: Ciclo de escritura de los módulos de memoria.

El tiempo de ciclo de escritura correspondiente, t_c , es la suma del tiempo de ocupación del bus más el tiempo de almacenamiento:

$$t_c = t_{\text{bus}} + t_{\text{alm}} = 20 + 80 = 100 \text{ ns}$$

Afirmación I:

Si el número de módulos es $N=3$ se tiene el siguiente esquema:



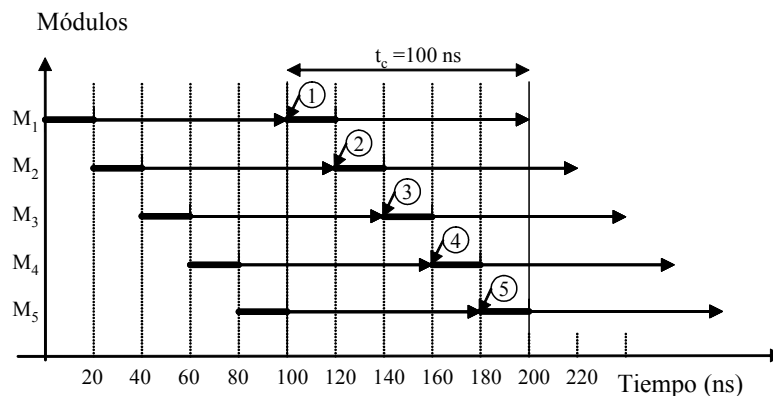
Se observa que en un periodo de tiempo igual a $t_c = 100 \text{ ns}$ se escriben tres palabras. Luego la velocidad máxima del sistema es:

$$v_{T_{\text{max}}} = \frac{3}{100} = 0.03 \text{ (palabras/nseg)}$$

Luego la afirmación es **VERDADERA**.

Afirmación II:

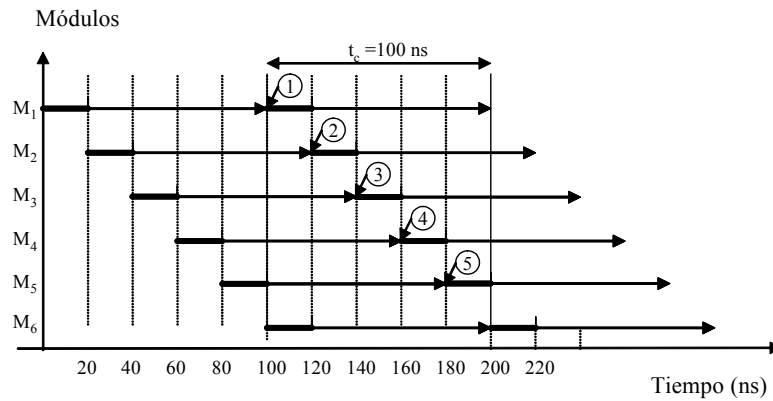
Supóngase que el número de módulos es $N=5$ se tiene el siguiente esquema:



Se observa que en un periodo de tiempo igual a $t_c = 100 \text{ ns}$ se escriben cinco palabras. Luego la velocidad máxima del sistema es:

$$v_{T_{\text{max}}} = \frac{5}{100} = 0.05 \text{ (palabras/nseg)}$$

Si $N=6$ se tiene el siguiente esquema:



Se observa que en un periodo de tiempo igual a $t_c=100$ ns se siguen escribiendo cinco palabras.

En conclusión, si $N>4$ se la velocidad máxima de transferencia es de 0.05 (palabras/nseg) y no de 0.06 (palabras/nseg) como se indica en la afirmación. Luego la afirmación es **FALSA**.

SOLUCIÓN PROBLEMA 1.24

DATOS

- Bus con arbitraje centralizado

La afirmación de la respuesta A es falsa, porque en un bus con arbitraje centralizado la decisión de qué módulo toma el control del bus corre a cargo del dispositivo denominado controlador del bus o árbitro.

La afirmación de la respuesta B es falsa ya que el método de arbitraje no condiciona el tipo de temporización.

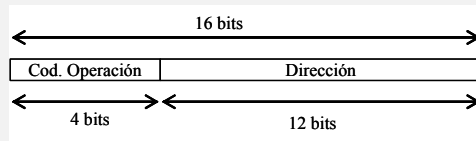
La afirmación de la respuesta C es falsa, ya que ese requisito sólo se da en los buses con arbitraje distribuido.

En conclusión la respuesta correcta es la **D**.

SOLUCIÓN PROBLEMA 1.25

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_2 ($Ac \leftarrow [M]$)
 0010_2 ($M \leftarrow [Ac]$)
 1000_2 ($Ac \leftarrow [Ac] - [M]$)
 1001_2 ($Ac \leftarrow [Ac] + [M]$)
- Contenido de ciertas posiciones de memoria: $[A0F_{16}] = X$, $[3A0_{16}] = Y$
- Averiguar el contenido de la posición de memoria $3A0_{16}$ tras ejecutar las instrucciones $13A0_{16}$ $9A0F_{16}$ $93A0_{16}$ $2A0F_{16}$

[Este problema es similar al **Problema 1.14**]

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

1) La instrucción $13A0_{16}$ tiene por código de operación $1_{16} = 0001_2$, y como dirección $3A0_{16}$ se ejecuta la siguiente instrucción:

$$Ac \leftarrow [3A0_{16}]$$

Luego, se estaría cargando en el acumulador el contenido de la dirección de memoria $3A0_{16}$, es decir, Y.

2) La instrucción $9A0F_{16}$ tiene por código de operación $9_{16} = 1001_2$, y como dirección $A0F_{16}$ se ejecuta la siguiente instrucción:

$$Ac \leftarrow [Ac] + [A0F_{16}]$$

Luego, se estaría cargando en el acumulador el resultado que se obtiene al sumar el contenido del acumulador $[Ac] = Y$ con $[A0F_{16}] = X$. Que da como resultado $Y + X$

3) La instrucción $93A0_{16}$ tiene por código de operación $9_{16} = 1001_2$, y como dirección $3A0_{16}$ se ejecuta la siguiente instrucción:

$$Ac \leftarrow [Ac] + [3A0_{16}]$$

Luego, se estaría cargando en el acumulador el resultado que se obtiene al sumar el contenido del acumulador $[Ac] = Y + X$ con $[3A0_{16}] = Y$. Que da como resultado $2Y + X$

4) La instrucción $2A0F_{16}$ tiene por código de operación $2_{16}=0010_2$, y como dirección $A0F_{16}$ se ejecuta la siguiente instrucción:

$$A0F_{16} \leftarrow [Ac]$$

Luego, se almacena en la dirección de memoria $A0F_{16}$ el contenido del acumulador $[Ac]=2Y+X$.

Este programa por tanto no modifica el contenido inicial de la posición de memoria $3A0_{16}$. Luego la respuesta al problema es que el contenido de $3A0_{16}$ tras ejecutar este programa sigue siendo el mismo, es decir,

$$[3A0_{16}] = Y$$