

# **ESTRUCTURA Y TECNOLOGIA DE COMPUTADORES II**

## **TEMA 3: *Unidad de Entrada Salida***

**SOLUCION A LOS PROBLEMAS  
PROPUESTOS EN EXAMEN**

**Curso 2008-2009  
Jose Manuel Díaz Martínez  
Tutor de la asignatura ETC-II**

**CONTENIDO**

SOLUCION PROBLEMA 3.1 .....	3
SOLUCION PROBLEMA 3.2 .....	3
SOLUCION PROBLEMA 3.3 .....	4
SOLUCION PROBLEMA 3.4 .....	4
SOLUCION PROBLEMA 3.5 .....	5
SOLUCION PROBLEMA 3.6 .....	5
SOLUCION PROBLEMA 3.7 .....	6
SOLUCION PROBLEMA 3.8 .....	7
SOLUCION PROBLEMA 3.9 .....	8
SOLUCION PROBLEMA 3.10 .....	9
SOLUCION PROBLEMA 3.12 .....	9
SOLUCION PROBLEMA 3.13 .....	9
SOLUCION PROBLEMA 3.14 .....	10
SOLUCION PROBLEMA 3.15 .....	11
SOLUCION PROBLEMA 3.16 .....	11
SOLUCION PROBLEMA 3.17 .....	11
SOLUCION PROBLEMA 3.18 .....	12
SOLUCION PROBLEMA 3.19 .....	12
SOLUCION PROBLEMA 3.20 .....	13
SOLUCION PROBLEMA 3.21 .....	13
SOLUCION PROBLEMA 3.22 .....	14
SOLUCION PROBLEMA 3.23 .....	14
SOLUCION PROBLEMA 3.24 .....	15
SOLUCION PROBLEMA 3.25 .....	15
SOLUCION PROBLEMA 3.26 .....	16
SOLUCION PROBLEMA 3.27 .....	16
SOLUCION PROBLEMA 3.28 .....	17
SOLUCION PROBLEMA 3.29 .....	17
SOLUCION PROBLEMA 3.30 .....	17
SOLUCION PROBLEMA 3.31 .....	18
SOLUCION PROBLEMA 3.32 .....	19
SOLUCION PROBLEMA 3.33 .....	20
SOLUCION PROBLEMA 3.34 .....	20
SOLUCION PROBLEMA 3.35 .....	20
SOLUCION PROBLEMA 3.36 .....	20
SOLUCION PROBLEMA 3.37 .....	21
SOLUCION PROBLEMA 3.38 .....	21
SOLUCION PROBLEMA 3.39 .....	22
SOLUCION PROBLEMA 3.40 .....	22
SOLUCION PROBLEMA 3.41 .....	23
SOLUCION PROBLEMA 3.42 .....	23
SOLUCION PROBLEMA 3.43 .....	24
SOLUCION PROBLEMA 3.44 .....	25
SOLUCION PROBLEMA 3.45 .....	25
SOLUCION PROBLEMA 3.46 .....	25
SOLUCION PROBLEMA 3.47 .....	26
SOLUCION PROBLEMA 3.48 .....	26
SOLUCION PROBLEMA 3.49 .....	27
SOLUCION PROBLEMA 3.50 .....	28
SOLUCION PROBLEMA 3.51 .....	29
SOLUCION PROBLEMA 3.52 .....	30

**SOLUCION PROBLEMA 3.1****DATOS**

- Canal multiplexor
- 2 unidades de disco => $v_{TD}=500$  Kbytes/seg
- 2 unidades de cinta => $v_{TC}=150$  Kbytes/seg

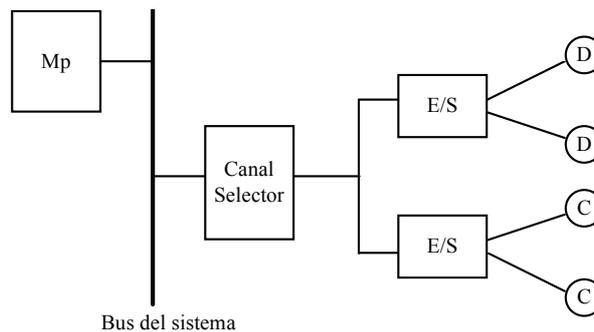
En un canal multiplexor la velocidad máxima de transferencia viene dada por la suma de las velocidades de cada uno de los dispositivos que controla. Luego

$$v_{tmax} = v_{Tcanal} = 2 \cdot v_{TD} + 2 \cdot v_{TC} = 2 \cdot (500) + 2 \cdot (150) = 1300(\text{Kbytes} / \text{seg})$$

**SOLUCION PROBLEMA 3.2****DATOS**

- Canal selector
- 2 unidades de disco => $v_{TD}=10^6$  bytes/seg
- 2 unidades de cinta => $v_{TC}=10^5$  bytes/seg
- $t_{bus}=10$ nseg

Se tiene una configuración como la que se muestra en la figura :



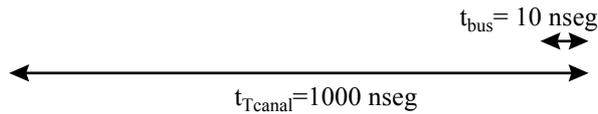
En primer lugar se va a obtener la velocidad de transferencia del canal selector  $v_{Tcanal}$ , que viene dada por la velocidad del dispositivo más rápido que controla, es decir, la velocidad de transferencia de una unidad de disco.

$$v_{Tcanal} = v_{TD} = 10^6(\text{bytes} / \text{seg})$$

La velocidad de transferencia del canal,  $v_{Tcanal}$ , indica el número de bytes que se pueden transmitir entre un periférico y la memoria principal en un segundo. A partir de  $v_{Tcanal}$  se puede obtener el tiempo necesario para transmitir un byte entre un periférico y  $M_p$ .

$$t_{Tcanal} = \frac{1}{v_{Tcanal}} = \frac{1}{10^6} = 10^{-6} \text{seg} = 1000 \text{nseg}$$

El enunciado del problema nos dice que el tiempo de utilización del bus del sistema  $t_{bus}$  en la transferencia de 1 byte entre un periférico y memoria principal  $M_p$  es  $t_{bus}=10$  nseg. En la figura se representan esquemáticamente  $t_{Tcanal}$  y  $t_{bus}$ .



Luego P es :

$$P = \frac{t_{bus}}{t_{Tcanal}} \cdot 100 = \frac{10}{1000} \cdot 100 = 1\%$$

### SOLUCION PROBLEMA 3.3

#### DATOS

- frecuencia del computador es  $f=10^8$  ciclos/seg
- 10 ciclos/instrucción
- 5 ciclos/pal
- DMA con estrategia de *transferencia por ráfagas*

En este tipo de estrategia el controlador del DMA toma el control del bus del sistema, una vez que la CPU ha ejecutado una instrucción y no lo cede hasta que ha transferido todo el bloque de palabras. Nos piden calcular la máxima velocidad de transferencia de datos  $v_{tmax}$  (pal/seg). Dicha velocidad vendrá dada por el cociente entre la velocidad del computador y el número de ciclos que se necesitan para escribir o leer una palabra.

$$v_{tmax} = \frac{f(\text{ciclos / seg})}{5(\text{ciclos / pal})} = \frac{10^8}{5} = 2 \cdot 10^7 \text{ pal / seg}$$

### SOLUCION PROBLEMA 3.4

#### DATOS

- $n=13$  líneas de dirección.
- Memoria con  $2^{12}$  palabras
- Utiliza E/S localizada en memoria.

El computador utiliza E/S localizada en memoria, eso significa que el controlador de E/S y la memoria comparten las líneas de control (lectura y escritura), por lo que se cumple el número total de direcciones posibles  $N_D$  estará formado por las direcciones de palabras de memoria  $N_{DM}$  y las direcciones de palabras en periféricos  $N_{DP}$ :

$$N_D = N_{DM} + N_{DP}$$

El número total de direcciones  $N_D$  viene definido por el número  $n$  de líneas de dirección.

$$N_D = 2^{13} \text{ direcciones}$$

El número de direcciones para acceder a palabras de memoria es

$$N_{DM} = 2^{12} \text{ direcciones}$$

Luego el número de direcciones para acceder a palabras en los periféricos vendrá dado por :

$$N_{DP} = N_D - N_{DM} = 2^{13} - 2^{12} = 2^{12} \text{ direcciones}$$

- a) Nos dicen que cada periférico ocupa una dirección =>  $R = 1$  dirección/periférico. Luego el número máximo de periféricos  $N_{P_{max}}$  es :

$$N_{P_{max}} = \frac{N_{DP}}{R} = \frac{2^{12} (dir)}{1 (dir / per)} = 2^{12} \text{ periféricos}$$

- b) Nos dicen que cada periférico ocupa 8 direcciones =>  $R = 8$  dir/peri. Luego el número máximo de periféricos  $N_{P_{max}}$  es :

$$N_{P_{max}} = \frac{N_{DP}}{R} = \frac{2^{12} (dir)}{2^3 (dir / per)} = 2^9 \text{ periféricos}$$

### SOLUCION PROBLEMA 3.5

#### DATOS

- 4 líneas de interrupción  $PI_3, PI_2, PI_1$  y  $PI_0$
- Son enmascarables las líneas  $PI_3, PI_2,$  y  $PI_1$
- Una petición de interrupción  $PI_i$  enmascara las líneas  $PI_j$  con  $i < j$  e  $i \neq 0$ .
- Se producen 4 peticiones de interrupción en este orden:  $PI_3, PI_2, PI_0$  y  $PI_1$
- Las tres últimas peticiones se producen mientras se está ejecutando el programa de servicio de  $PI_3$

De acuerdo con el enunciado la prioridad de las diferentes líneas es:

$$\text{prior}(PI_0) > \text{prior}(PI_3) > \text{prior}(PI_2) > \text{prior}(PI_1)$$

Analicemos la secuencia de petición de interrupciones.

- 1) Llega  $PI_3$ , se enmascaran  $PI_2$  y  $PI_1$ , y se comienza a ejecutar el programa de servicio de la interrupción  $PI_3$ .
- 2) Llega  $PI_2$ , como está enmascarada no se atiende.
- 3) Llega  $PI_0$ , como tiene una prioridad superior a  $PI_3$ , se detiene la ejecución del programa de servicio de  $PI_3$  y se ejecuta completo el programa de servicio de  $PI_0$ .
- 4) Se continua con el programa de servicio de  $PI_3$ . Llega  $PI_1$  como está enmascarada no se atiende.
- 5) Terminado el programa de servicio de  $PI_3$  se ejecuta el programa de servicio de  $PI_2$ , dejando enmascarada  $PI_1$ .
- 6) Terminado el programa de servicio de  $PI_2$  se ejecuta el programa de servicio de  $PI_1$ , ahora no hay ninguna línea enmascarada.
- 7) Se termina el programa de ejecución de  $PI_1$ .

Luego el orden de ejecución de las interrupciones es:

$$PI_3, PI_0, PI_3, PI_2 \text{ y } PI_1$$

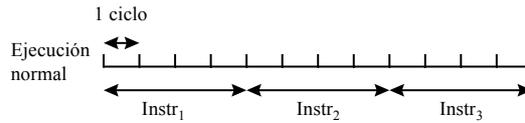
### SOLUCION PROBLEMA 3.6

#### DATOS

- frecuencia del reloj del computador es  $f = 200 \cdot 10^6$  ciclos/seg
- 4 ciclos/instrucción
- 2 ciclos/instrucción no requiere el uso del bus
- DMA con estrategia de *robo de ciclos*

Para resolver este problema, primero hay que calcular la velocidad de ejecución  $v_{ejec1}$  (instrucciones/segundo) cuando no hay DMA y a continuación calcular la velocidad de ejecución  $v_{ejec2}$  (instrucciones/segundo) cuando hay DMA con estrategia de robo de ciclos. La velocidad de ejecución  $v_{ejec}$  se define como el número de instrucciones que el computador es capaz de ejecutar en un segundo.

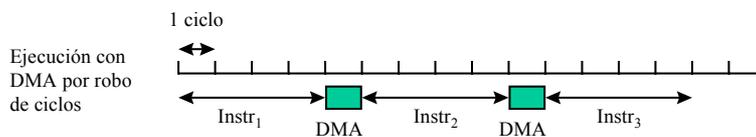
1) Calcular  $v_{ejec1}$ .



Sino hay DMA el computador tarda en promedio 4 ciclos en ejecutar cada instrucción. Luego la velocidad de ejecución es:

$$v_{ejec1} = \frac{f(\text{ciclos / seg})}{4(\text{ciclos / inst})} = \frac{200 \times 10^6}{4} = 5 \cdot 10^7 (\text{instr / seg})$$

2) Calcular  $v_{ejec2}$ .



Ahora cada vez que se ejecuta una instrucción, a continuación se cede el bus del sistema al controlador de DMA para que realice la transferencia de una palabra. Por lo que ahora transcurren 5 ciclos entre instrucción e instrucción. Luego la velocidad de ejecución es:

$$v_{ejec1} = \frac{f(\text{ciclos / seg})}{5(\text{ciclos / inst})} = \frac{200 \times 10^6}{5} = 4 \cdot 10^7 (\text{instr / seg})$$

3) El número de instrucciones que dejen de ejecutarse por segundo viene dado por :

$$v_{ejec1} - v_{ejec2} = 5 \cdot 10^7 - 4 \cdot 10^7 = 10^7 (\text{instr / seg})$$

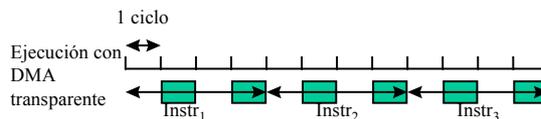
-----

### SOLUCION PROBLEMA 3.7

#### DATOS

- frecuencia del reloj del computador es  $f=10^8$  ciclos/seg
- 4 ciclos/instrucción
- 2 ciclos/instrucción no requiere el uso del bus
- 1 ciclo/palabra

1) Afirmación: Con DMA transparente  $v_{\text{max}}=5 \cdot 10^7$  (pal/seg)



Se ha de recordar que en la estrategia de DMA transparente, el DMA se realiza en aquellas fases de la ejecución de una instrucción en que la CPU no hace uso del bus. El enunciado del problema nos dice que en promedio el computador necesita 4 ciclos para ejecutar una instrucción y que sólo en dos de ellos no requiere el uso del bus. Además nos dicen que el controlador de DMA únicamente necesita de 1 ciclo para transmitir una palabra. Luego el número de palabras que se transmiten por DMA en la ejecución de una instrucción (R), vendría dado por:

$$R = \frac{2(\text{ciclos} / \text{instr})}{1(\text{ciclo} / \text{pal})} = 2(\text{pal} / \text{instr})$$

La velocidad de ejecución es:

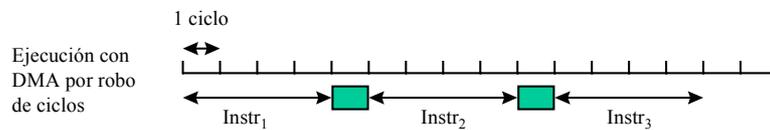
$$v_{ejec} = \frac{f(\text{ciclos} / \text{seg})}{4(\text{ciclos} / \text{inst})} = \frac{10^8}{4} = 25 \cdot 10^6 (\text{instr} / \text{seg})$$

La velocidad de transferencia máxima es:

$$v_{tmax} = R \cdot v_{ejec} = 2(\text{pal} / \text{inst}) \cdot 25 \cdot 10^6 (\text{inst} / \text{seg}) = 5 \cdot 10^7 (\text{pal} / \text{seg})$$

Luego la afirmación I es **verdadera**.

II) Con DMA por robo de ciclos  $v_{tmax} = 2 \cdot 10^7$  (pal/seg).



La velocidad de ejecución es:

$$v_{ejec} = \frac{f(\text{ciclos} / \text{seg})}{5(\text{ciclos} / \text{inst})} = \frac{10^8}{5} = 2 \cdot 10^7 (\text{instr} / \text{seg})$$

La velocidad de transferencia máxima, puesto que  $R=1\text{pal}/\text{inst}$

$$v_{tmax} = R \cdot v_{ejec} = 1(\text{pal} / \text{inst}) \cdot 2 \cdot 10^7 (\text{inst} / \text{seg}) = 2 \cdot 10^7 (\text{pal} / \text{seg})$$

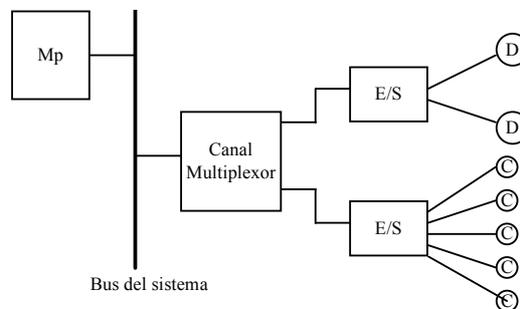
La afirmación II, es **verdadera**.

### SOLUCION PROBLEMA 3.8

**DATOS**

- Canal multiplexor
- 2 unidades de disco =>  $v_{TD} = 10^6$  bytes/seg
- 5 unidades de cinta =>  $v_{TC} = 2 \cdot 10^5$  bytes/seg
- $t_{bus} = 100\text{nseg}$

Se tiene una configuración como la que se muestra en la figura:



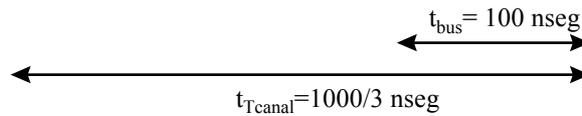
En primer lugar se va a obtener la velocidad de transferencia del canal Multiplexor  $v_{Tcanal}$ , que viene dada por la suma de las velocidades de cada uno de los dispositivos que controla.

$$v_{Tcanal} = 2 \cdot v_{TD} + 5 \cdot v_{TC} = 2 \cdot 10^6 + 5 \cdot 2 \cdot 10^5 = 3 \cdot 10^6 \text{ (bytes / seg)}$$

La velocidad de transferencia del canal,  $v_{Tcanal}$ , indica el número de bytes que se pueden transmitir entre un periférico y la memoria principal en un segundo. A partir de  $v_{Tcanal}$  se puede obtener el tiempo necesario para transmitir un byte entre un periférico y  $M_p$ .

$$t_{Tcanal} = \frac{1}{v_{Tcanal}} = \frac{1}{3 \cdot 10^6} = \frac{1000}{3} \text{ nseg}$$

El enunciado del problema nos dice que el tiempo de utilización del bus del sistema  $t_{bus}$  en la transferencia de 1 byte entre un periférico y memoria principal  $M_p$  es  $t_{bus}=100$  nseg. En la figura se representan esquemáticamente  $t_{Tcanal}$  y  $t_{bus}$ .



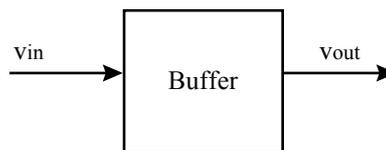
Luego P es :

$$P = \frac{t_{bus}}{t_{Tcanal}} \cdot 100 = \frac{100}{\frac{1000}{3}} \cdot 100 = 30\%$$

### SOLUCION PROBLEMA 3.9

**DATOS**

- Controlador de E/S con buffer de capacidad  $C=256$  Kb.
- Buffer inicialmente vacío.
- Operación de E/S con una impresora :
  - Recibe datos por el bus del sistema a  $v_{in}= 1$  Mb/seg.
  - Transmite datos a la impresora a  $v_{out}= 256$  Kb/seg.
  - Recibe y transmite de forma continua simultáneamente.



De los datos del problema se deduce que el buffer recibe datos a una velocidad más elevada de lo que transmite. Habrá que calcular la velocidad de llenado  $v$ , que será:

$$v = v_{in} - v_{out} = 2^{20} \text{ (bytes / seg)} - 2^{18} \text{ (bytes / seg)} = (2^2 - 1) \cdot 2^{18} = 3 \cdot 2^{18} \text{ (bytes / seg)}$$

El tiempo de llenado se calcula como el cociente entre la capacidad del buffer y la velocidad de llenado.

$$t_{llenado} = \frac{C}{v} = \frac{2^{18} \text{ (bytes)}}{3 \cdot 2^{18} \text{ (bytes / seg)}} = 0.33 \text{ seg}$$

### SOLUCION PROBLEMA 3.10

**DATOS**

- VP velocidad máxima de transferencia empleando E/S controlada por programa.
- VI velocidad máxima de transferencia empleando E/S controlada con interrupciones.
- VD velocidad máxima de transferencia empleando E/S controlada con DMA.

La relación que se cumple es :  $VD > VI > VP$   
 La respuesta correcta es la B.

### SOLUCION PROBLEMA 3.12

**DATOS**

- $n= 32$  líneas de dirección.
- Memoria con 32 Mb
- Utiliza E/S aislada de memoria.
- $R=16$  direcciones/periférico

El computador utiliza E/S aislada de memoria, eso significa que el controlador de E/S y la memoria disponen de líneas de control (lectura y escritura) autónomas, por lo que el mapa de direcciones de palabras de memoria es independiente del mapa de direcciones de palabras en periféricos.

El número total de direcciones  $N_D$  viene definido por el número  $n$  de líneas de dirección.

$$N_D = 2^{32} \text{ direcciones}$$

El número de direcciones para acceder a palabras en los periféricos  $N_{DP}$  será igual al número total de direcciones disponibles:

$$N_{DP} = N_D = 2^{32} \text{ direcciones}$$

Nos dicen que cada periférico ocupa 16 direcciones  $\Rightarrow R= 16$  dir/perí. Luego el número máximo de periféricos  $N_{Pmax}$  es :

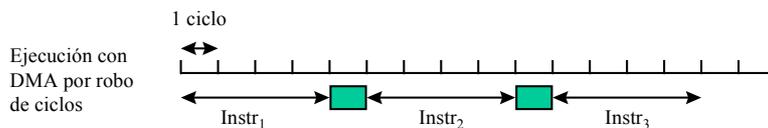
$$N_{Pmax} = \frac{N_{DP}}{R} = \frac{2^{32} (dir)}{2^4 (dir / per)} = 2^{28} \text{ periféricos}$$

### SOLUCION PROBLEMA 3.13

**DATOS**

- Frecuencia del reloj del computador es  $f=200 \cdot 10^6$  ciclos/seg
- 4 ciclos/instrucción
- 1 ciclo/instrucción no requiere el uso del bus
- 1 ciclo/palabra

- Si se realiza DMA por robo de ciclos:



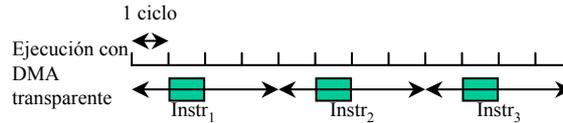
La velocidad de ejecución es:

$$v_{ejec} = \frac{f(\text{ciclos/seg})}{5(\text{ciclos/inst})} = \frac{200 \cdot 10^6}{5} = 40 \cdot 10^6 = 4 \cdot 10^7 (\text{instr/seg})$$

Puesto que el sistema utiliza 1 ciclo en transferir 1 palabra y como tras ejecutar una instrucción se cede 1 ciclo para realizar DMA entonces el número de palabras que se transmiten tras ejecutar una instrucción es  $R=1\text{pal/inst}$ . Así la velocidad de transferencia máxima es:

$$v_{t\max} = R \cdot v_{ejec} = 1(\text{pal/inst}) \cdot 4 \cdot 10^7 (\text{instr/seg}) = 4 \cdot 10^7 (\text{pal/seg})$$

- Si se realiza DMA transparente:



La velocidad de ejecución es:

$$v_{ejec} = \frac{f(\text{ciclos/seg})}{4(\text{ciclos/inst})} = \frac{200 \cdot 10^6}{4} = 50 \cdot 10^6 = 5 \cdot 10^7 (\text{instr/seg})$$

En la estrategia de DMA transparente, el DMA se realiza en aquellas fases de la ejecución de una instrucción en que la CPU no hace uso del bus. El enunciado del problema nos dice que en promedio el computador necesita 4 ciclos para ejecutar una instrucción y que sólo en uno de ellos no requiere el uso del bus. Además nos dicen que el controlador de DMA únicamente necesita 1 ciclo para transmitir una palabra. Luego el número de palabras  $R$  que se transmiten en la ejecución de una instrucción, vendría dado por:

$$R = \frac{1(\text{ciclos/inst})}{1(\text{ciclo/pal})} = 1(\text{pal/inst})$$

La velocidad de transferencia máxima es:

$$v_{t\max} = R \cdot v_{ejec} = 1(\text{pal/inst}) \cdot 5 \cdot 10^7 (\text{instr/seg}) = 5 \cdot 10^7 (\text{pal/seg})$$

De acuerdo con estos resultados las afirmaciones son: Afirmación I: **FALSA** Afirmación II: **VERDADERA**

## SOLUCION PROBLEMA 3.14

### DATOS

- Canal multiplexor
- 3 unidades de disco  $\Rightarrow v_{TD} = 5 \cdot 10^6$  bytes/seg
- 5 unidades de cinta  $\Rightarrow v_{TC} = 10^5$  bytes/seg
- $t_{bus} = 50$  nseg

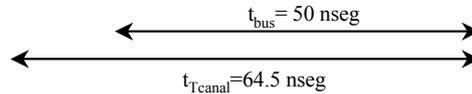
En primer lugar se va a obtener la velocidad de transferencia del canal multiplexor  $v_{T\text{canal}}$ , que viene dada por la suma de las velocidades de cada uno de los dispositivos que controla.

$$v_{T\text{canal}} = 3 \cdot v_{TD} + 5 \cdot v_{TC} = 3 \cdot 5 \cdot 10^6 + 5 \cdot 10^5 = 15.5 \cdot 10^6 (\text{bytes/seg})$$

La velocidad de transferencia del canal,  $v_{T\text{canal}}$ , indica el número de bytes que se pueden transmitir entre un periférico y la memoria principal en un segundo. A partir de  $v_{T\text{canal}}$  se puede obtener el tiempo necesario para transmitir un byte entre un periférico y  $M_p$ .

$$t_{Tcanal} = \frac{1}{v_{Tcanal}} = \frac{1}{15.5 \cdot 10^6} = \frac{10^{-9}}{15.5 \cdot 10^6 \cdot 10^{-9}} = \frac{1000}{15.5} nseg = 64.5 nseg$$

El enunciado del problema nos dice que el tiempo de utilización del bus del sistema  $t_{bus}$  en la transferencia de 1 byte entre un periférico y memoria principal  $M_p$  es  $t_{bus}=50$  nseg. En la figura se representan esquemáticamente  $t_{Tcanal}$  y  $t_{bus}$ .



Luego el porcentaje máximo de tiempo  $P$  que estaría ocupado el bus del sistema en una transferencia de datos a través de este canal multiplexor sería :

$$P = \frac{t_{bus}}{t_{Tcanal}} \cdot 100 = \frac{50}{64.5} \cdot 100 = 77.5\%$$

### SOLUCION PROBLEMA 3.15

Afirmación I: Es **falsa** ya que la comprobación de la interrupción es por hardware (línea RI) y no por software eso corresponde al tipo encuesta.

Afirmación II: Es **falsa**, la CPU recibe el vector de interrupciones  $v_{int}$  exclusivamente por el bus de datos.

Afirmación III: Es **falsa**, se tratan de conceptos independientes, ya que se está hablando por un lado de identificación de la interrupción I y por otro de niveles de interrupción I.

Luego la respuesta correcta es la **D**.

### SOLUCION PROBLEMA 3.16

#### DATOS

- Canal multiplexor
- 2 unidades de disco  $\Rightarrow v_{TD}=10^6$  bytes/seg
- 10 unidades de cinta  $\Rightarrow v_{TC}= 10^3$  bytes/seg

La velocidad de transferencia del canal multiplexor  $v_{Tcanal}$  viene dada por la suma de las velocidades de cada uno de los dispositivos que controla.

$$v_{Tcanal} = 2 \cdot v_{TD} + 10 \cdot v_{TC} = 2 \cdot 10^6 + 10 \cdot 10^3 = 2 \cdot 10^6 + 10^4 = 2.01 \cdot 10^6 \text{ (bytes / seg)}$$

### SOLUCION PROBLEMA 3.17

La solución a este problema se encuentra en el archivo `probT3_17.pdf`

**SOLUCION PROBLEMA 3.18****DATOS**

- 4 líneas de interrupción  $PI_1$ ,  $PI_2$ ,  $PI_3$  y  $PI_4$
- $\text{prior}(PI_4) > \text{prior}(PI_3) > \text{prior}(PI_2) > \text{prior}(PI_1)$
- El sistema no permite anidamiento.
- Se producen 4 peticiones de interrupción en este orden:  $PI_3$ ,  $PI_1$ ,  $PI_4$  y  $PI_2$
- Las tres últimas peticiones se producen mientras se está ejecutando el programa de servicio de  $PI_3$

Conviene darse cuenta de que el sistema no permite anidamiento, en consecuencia se trata de un sistema con un único nivel de interrupciones, de tal forma que cuando comienza a ejecutarse el programa de gestión de una interrupción, éste se completará antes de atender a otra petición de interrupción.

Analicemos la secuencia de petición de interrupciones :

- 1) Llega  $PI_3$  y comienza a ejecutarse el programa de gestión de  $PI_3$ .
- 2) Llega  $PI_1$ , continua ejecutándose el programa de gestión de  $PI_3$ .
- 3) Llega  $PI_4$ , continua ejecutándose el programa de gestión de  $PI_3$ .
- 4) Llega  $PI_2$ , continua ejecutándose el programa de gestión de  $PI_3$ .
- 5) Terminado el programa de gestión de  $PI_3$ , se procede a ejecutar el programa de gestión de  $PI_4$  que es la interrupción recibida con mayor prioridad.
- 6) Terminado el programa de gestión de  $PI_4$ , se procede a ejecutar el programa de gestión de  $PI_2$  que es la interrupción recibida con mayor prioridad.
- 7) Terminado el programa de gestión de  $PI_1$ , se procede a ejecutar el programa de gestión de  $PI_4$  que es la interrupción recibida con mayor prioridad.

Luego el orden de ejecución de las interrupciones es:

$PI_3, PI_4, PI_2$  y  $PI_1$

**SOLUCION PROBLEMA 3.19****DATOS**

- Canal multiplexor
- 2 unidades de disco  $\Rightarrow v_{TD} = 10^6$  bytes/seg
- 10 unidades de cinta  $\Rightarrow v_{TC} = 10^3$  bytes/seg
- $t_{bus} = 100$  nseg

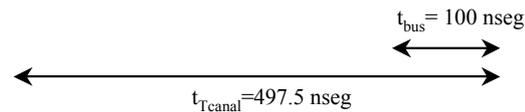
La velocidad de transferencia del canal multiplexor  $v_{Tcanal}$  viene dada por la suma de las velocidades de cada uno de los dispositivos que controla.

$$v_{Tcanal} = 2 \cdot v_{TD} + 10 \cdot v_{TC} = 2 \cdot 10^6 + 10 \cdot 10^3 = 2 \cdot 10^6 + 10^4 = 2.01 \cdot 10^6 \text{ (bytes/seg)}$$

La velocidad de transferencia del canal,  $v_{Tcanal}$ , indica el número de bytes que se pueden transmitir entre un periférico y la memoria principal en un segundo. A partir de  $v_{Tcanal}$  se puede obtener el tiempo necesario para transmitir un byte entre un periférico y  $M_p$ .

$$t_{Tcanal} = \frac{1}{v_{Tcanal}} = \frac{1}{2.01 \cdot 10^6} = \frac{10^{-9}}{2.01 \cdot 10^6 \cdot 10^{-9}} = 497.5 \text{ nseg}$$

El enunciado del problema nos dice que el tiempo de utilización del bus del sistema  $t_{bus}$  en la transferencia de 1 byte entre un periférico y memoria principal  $M_p$  es  $t_{bus} = 100$  nseg. En la Figura se representan esquemáticamente  $t_{Tcanal}$  y  $t_{bus}$ .



Luego el porcentaje máximo de tiempo  $P$  que estaría ocupado el bus del sistema en una transferencia de datos a través de este canal multiplexor sería :

$$P = \frac{t_{bus}}{t_{Tcanal}} \cdot 100 = \frac{100}{497.5} \cdot 100 = 20.1\%$$

## SOLUCION PROBLEMA 3.20

### DATOS

- 5 líneas de interrupción  $PI_4, PI_3, PI_2, PI_1, PI_0$
- Enmascarables mediante 5 bits del Registro de estado de la CPU.
- Una petición  $PI_i$  desactiva las líneas  $PI_i$  con  $i < j$
- $prior(PI_4) > prior(PI_3) > prior(PI_2) > prior(PI_1) > prior(PI_0)$
- Interrupciones anidables. Sistema multinivel.
- Se producen 4 peticiones de interrupción es este orden:  $PI_2, PI_4, PI_3, PI_1$
- Antes de finalizar  $PI_2$ , llegan simultáneamente ( $PI_4, PI_3, PI_1$ )

Analicemos la secuencia de petición de interrupciones :

- 1) Llega  $PI_2$  y comienza a ejecutarse el programa de servicio de  $PI_2$ .
- 2) Llegan ( $PI_4, PI_3, PI_1$ ) simultáneamente, puesto que  $PI_4$  es la interrupción más prioritaria se comienza a ejecutar el programa de servicio de  $PI_4$  hasta completarlo. Quedan pendientes de atender  $PI_3, PI_2$  y  $PI_1$
- 3) Finalizado el programa de servicio de  $PI_4$ , comienza a ejecutar el programa de servicio de  $PI_3$ , que es la interrupción pendiente de mayor prioridad, hasta completarlo. Quedan pendientes de atender  $PI_2$  y  $PI_1$ .
- 4) Finalizado el programa de servicio de  $PI_3$ , continúa con la ejecución del programa de servicio de  $PI_2$ , que es la interrupción pendiente de mayor prioridad, justo en el punto en que fue interrumpido en el paso 2) hasta completarlo. Queda pendiente de atender  $PI_1$ .
- 5) Se atiende a  $PI_1$

Luego el orden en que se tratan las interrupciones es:

$$PI_2, PI_4, PI_3, PI_2, PI_1$$

## SOLUCION PROBLEMA 3.21

### DATOS

- Canal selector
- 3 unidades de disco  $\Rightarrow v_{TD} = 5 \cdot 10^6$  bytes/seg
- 5 unidades de cinta  $\Rightarrow v_{TC} = 10^5$  bytes/seg
- $t_{bus} = 50$  nseg

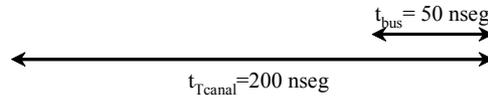
La velocidad de transferencia de un canal selector  $v_{Tcanal}$  es:

$$v_{Tcanal} = \max\{v_{TD}, v_{TC}\} = 5 \cdot 10^6 \text{ (bytes / seg)}$$

La velocidad de transferencia del canal,  $v_{Tcanal}$ , indica el número de bytes que se pueden transmitir entre un periférico y la memoria principal en un segundo. A partir de  $v_{Tcanal}$  se puede obtener el tiempo necesario para transmitir un byte entre un periférico y  $M_p$ .

$$t_{Tcanal} = \frac{1}{v_{Tcanal}} = \frac{1}{5 \cdot 10^6} = \frac{10^{-9}}{5 \cdot 10^6 \cdot 10^{-9}} = \frac{1000}{5} nseg = 200 nseg$$

El enunciado del problema nos dice que el tiempo de utilización del bus del sistema  $t_{bus}$  en la transferencia de 1 byte entre un periférico y memoria principal  $M_p$  es  $t_{bus}=50$  nseg. En la Figura se representan esquemáticamente  $t_{Tcanal}$  y  $t_{bus}$ .



Luego el porcentaje máximo de tiempo  $P$  que estaría ocupado el bus del sistema en una transferencia de datos a través de este canal multiplexor sería :

$$P = \frac{t_{bus}}{t_{Tcanal}} \cdot 100 = \frac{50}{200} \cdot 100 = 25\%$$

## SOLUCION PROBLEMA 3.22

### DATOS

- El mecanismo de la figura es E/S localizada en memoria.

La E/S localizada en memoria se caracteriza porque el espacio de direcciones disponible está compartido por la memoria y por los elementos periféricos. De tal forma que dada una dirección se puede acceder a una posición de memoria o un periférico. Luego en consecuencia las afirmaciones I y II son **verdaderas**.

## SOLUCION PROBLEMA 3.23

### DATOS

- 4 líneas de interrupción  $PI_1$ ,  $PI_2$ ,  $PI_3$  y  $PI_4$
- $prior(PI_4) > prior(PI_3) > prior(PI_2) > prior(PI_1)$
- El sistema permite anidamiento.
- Se producen 4 peticiones de interrupción es este orden:  $PI_3$ ,  $PI_1$ ,  $PI_4$  y  $PI_2$
- Las tres últimas peticiones se producen mientras se está ejecutando el programa de servicio de  $PI_3$

Conviene darse cuenta de que el sistema permite **anidamiento**, en consecuencia se trata de un **sistema multinivel** de interrupciones, de tal forma que cuando comienza a ejecutarse el programa de gestión de una interrupción, si llega una interrupción de prioridad superior pasa a ejecutar la rutina de servicio de dicha interrupción.

Analicemos la secuencia de petición de interrupciones :

- 1) Está ejecutando el programa de gestión de  $PI_3$ .
- 2) Llega  $PI_1$ , puesto que  $prior(PI_1) < prior(PI_3)$  ignora  $PI_1$  colocándola en una cola a la espera de ser atendida, continua con el programa de gestión de  $PI_3$ .
- 3) Llega  $PI_4$ , puesto que  $prior(PI_4) > prior(PI_3)$  detiene la ejecución del programa de gestión de  $PI_3$  y comienza a ejecutar el programa de servicio de  $PI_4$ .
- 4) Llega  $PI_2$ , puesto que  $prior(PI_2) < prior(PI_4)$  ignora  $PI_2$  colocándola en una cola a la espera de ser atendida, continua ejecutando el programa de gestión de  $PI_4$ ,
- 5) Terminado el programa de gestión de  $PI_4$ , se procede a continuar con el programa de gestión de  $PI_3$  hasta su finalización.
- 6) Terminado el programa de gestión de  $PI_3$ , se procede a ejecutar el programa de gestión de  $PI_2$  que es la interrupción colocada en la cola de espera con mayor prioridad.
- 7) Terminado el programa de gestión de  $PI_2$ , se procede a ejecutar el programa de gestión de  $PI_1$ .

Luego el orden de ejecución de las interrupciones es:

PI<sub>3</sub>, PI<sub>4</sub>, PI<sub>3</sub>, PI<sub>2</sub>, PI<sub>1</sub>

### SOLUCION PROBLEMA 3.24

#### DATOS

- Canal multiplexor
- 2 unidades de disco =>  $v_{TD}=10^6$  bytes/seg
- 10 unidades de cinta =>  $v_{TC}= 10^3$  bytes/seg

La velocidad de transferencia del canal multiplexor  $v_{Tcanal}$  viene dada por la suma de las velocidades de cada uno de los dispositivos que controla.

$$v_{Tcanal} = 2 \cdot v_{TD} + 10 \cdot v_{TC} = 2 \cdot 10^6 + 10 \cdot 10^3 = 2 \cdot 10^6 + 10^4 = 2.01 \cdot 10^6 \text{ (bytes / seg)}$$

### SOLUCION PROBLEMA 3.25

#### DATOS

- Canal multiplexor
- 5 unidades de disco =>  $v_{TD}=10^6$  bytes/seg
- 10 unidades de cinta =>  $v_{TC}= 2 \cdot 10^5$  bytes/seg
- $t_{bus}=200$  nseg

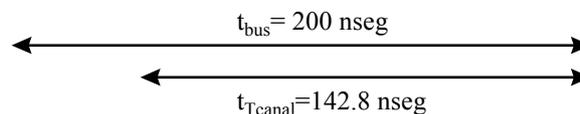
En primer lugar se va a obtener la velocidad de transferencia del canal Multiplexor  $v_{Tcanal}$ , que viene dada por la suma de las velocidades de cada uno de los dispositivos que controla.

$$v_{Tcanal} = 5 \cdot v_{TD} + 10 \cdot v_{TC} = 5 \cdot 10^6 + 10 \cdot 2 \cdot 10^5 = 7 \cdot 10^6 \text{ (bytes / seg)}$$

La velocidad de transferencia del canal,  $v_{Tcanal}$ , indica el número de bytes que se pueden transmitir entre un periférico y la memoria principal en un segundo. A partir de  $v_{Tcanal}$  se puede obtener el tiempo necesario para transmitir un byte entre un periférico y  $M_p$ .

$$t_{Tcanal} = \frac{1}{v_{Tcanal}} = \frac{1}{7 \cdot 10^6} = \frac{1000}{7} = 142.8 \text{ nseg}$$

El enunciado del problema nos dice que el tiempo de utilización del bus del sistema  $t_{bus}$  en la transferencia de 1 byte entre un periférico y memoria principal  $M_p$  es  $t_{bus}=100$  nseg. En la figura se representan esquemáticamente  $t_{Tcanal}$  y  $t_{bus}$ .



De la figura se deduce que se tiene un sistema cuyo canal puede transmitir un byte entre uno de sus periféricos y memoria, más rápido de lo que ese byte tarda en transmitirse por el bus del sistema. El tiempo de transferencia viene limitado por el tiempo de utilización del bus del sistema, ya que  $t_{bus} > t_{Tcanal}$ . Por lo tanto el tiempo de transferencia es :

$$t_T = t_{bus}$$

Luego P es :

$$P = \frac{t_{bus}}{t_T} \cdot 100 = \frac{t_{bus}}{t_{bus}} \cdot 100 = \frac{200}{200} \cdot 100 = 100\%$$

Es decir, el bus estaría saturado.

## SOLUCION PROBLEMA 3.26

### DATOS

- Módulo de gestión de prioridades
- Módulo de enmascaramiento

El orden de colocación de los módulos es:

Primero el módulo de enmascaramiento y a continuación el módulo de gestión de prioridades.

La razón de esta colocación es obvia, ya que si se colocasen al revés podría darse la circunstancia de que ante una petición múltiple de interrupciones el módulo de gestión de prioridades marcara como más prioritaria una interrupción que al pasar por el módulo de enmascaramiento resultase estar enmascarada. Con lo que no se atendería ninguna.

## SOLUCION PROBLEMA 3.27

### DATOS

- n= 13 líneas de dirección.
- Memoria con  $3 \times 2^{11}$  palabras
- Utiliza E/S localizada en memoria.
- R=16 direcciones/periférico

El computador utiliza E/S localizada en memoria, eso significa que el controlador de E/S y la memoria comparten las líneas de control (lectura y escritura), por lo que se cumple el número total de direcciones posibles  $N_{TD}$  estará formado por las direcciones de palabras de memoria  $M_{DM}$  y las direcciones de palabras en periféricos  $N_{DP}$ :

$$N_D = N_{DM} + N_{DP}$$

El número total de direcciones  $N_D$  viene definido por el número n de líneas de dirección.

$$N_D = 2^{13} \text{ direcciones}$$

El número de direcciones para acceder a palabras de memoria es

$$N_{DM} = 3 \cdot 2^{11} \text{ direcciones}$$

Luego el número de direcciones para acceder a palabras en los periféricos vendrá dado por :

$$N_{DP} = N_D - N_{DM} = 2^{13} - 3 \cdot 2^{11} = (4 - 3) \cdot 2^{11} = 2^{11} \text{ direcciones}$$

Nos dicen que cada periférico ocupa una dirección  $\Rightarrow R = 16$  dirección/periférico. Luego el número máximo de periféricos  $N_{Pmax}$  es :

$$N_{Pmax} = \frac{N_{DP}}{R} = \frac{2^{11} (dir)}{2^4 (dir / per)} = 2^7 \text{ periféricos}$$

### SOLUCION PROBLEMA 3.28

**DATOS**

- Canal multiplexor
- 5 unidades de disco =>v<sub>TD</sub>=10<sup>6</sup> bytes/seg
- 10 unidades de cinta =>v<sub>TC</sub>= 2·10<sup>5</sup> bytes/seg
- t<sub>bus</sub>=200 nseg

Este problema es completamente análogo al problema 3.25, salvo que nos piden la velocidad de transferencia agregada a este sistema. Puesto que el sistema está saturado P=100%. La velocidad máxima de transferencia viene dada por la inversa del tiempo de utilización del bus:

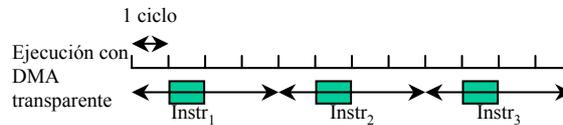
$$v_{T \max} = \frac{1}{\max(\bar{t}_{bus}, t_{Tcanal})} = \frac{1}{t_{bus}} = \frac{1 \text{ byte}}{200 \text{ nseg}} = \frac{10^7}{2} = 5 \cdot 10^6 \text{ (bytes / seg)}$$

### SOLUCION PROBLEMA 3.29

**DATOS**

- Estrategia DMA transparente
- Frecuencia del reloj del computador es f=200·10<sup>6</sup> ciclos/seg
- 4 ciclos/instrucción
- 1 ciclo/instrucción no requiere el uso del bus
- 1 ciclo/palabra

El esquema de DMA transparente de este problema se representa en la siguiente figura



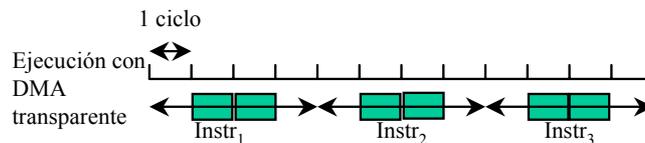
La *estrategia de DMA transparente* es la única que no produce ninguna reducción o interferencia en la velocidad de ejecución de la CPU, ya que el DMA se realiza justamente en aquellos fases de la ejecución de una instrucción en que la CPU no requiere hacer uso del bus del sistema.

### SOLUCION PROBLEMA 3.30

**DATOS**

- Frecuencia del reloj del computador es f=200·10<sup>6</sup> ciclos/seg
- 4 ciclos/instrucción
- 2 ciclos/instrucción consecutivos no requiere el uso del bus
- 2 ciclos/palabra
- Se desea mover un bloque de datos compuesto de 100·10<sup>6</sup> palabras

El esquema de DMA transparente de este problema se representa en la siguiente figura



La velocidad de ejecución de la CPU es:

$$v_{ejec} = \frac{f(\text{ciclos} / \text{seg})}{4(\text{ciclos} / \text{instr})} = \frac{200 \cdot 10^6}{4} = 50 \cdot 10^6 = 5 \cdot 10^7 (\text{instr} / \text{seg})$$

En la estrategia de DMA transparente, el DMA se realiza en aquellas fases de la ejecución de una instrucción en que la CPU no hace uso del bus. El enunciado del problema nos dice que en promedio el computador necesita 4 ciclos para ejecutar una instrucción y que sólo en dos de ellos consecutivos no requiere el uso del bus. Además nos dicen que el controlador de DMA necesita 2 ciclos para transmitir una palabra. Luego el número de palabras R que se transmiten en la ejecución de una instrucción, vendría dado por:

$$R = \frac{2(\text{ciclos} / \text{instr})}{2(\text{ciclos} / \text{pal})} = 1(\text{pal} / \text{instr})$$

La velocidad de transferencia máxima es:

$$v_{t\max} = R \cdot v_{ejec} = 1(\text{pal} / \text{instr}) \cdot 5 \cdot 10^7 (\text{instr} / \text{seg}) = 5 \cdot 10^7 (\text{pal} / \text{seg})$$

Luego, el tiempo que tardará el controlador de DMA en mover un bloque de datos de  $N=10 \cdot 10^7$  palabras es:

$$t = \frac{N(\text{pal})}{v_{t\max} (\text{pal} / \text{seg})} = \frac{10 \cdot 10^7}{5 \cdot 10^7} = 2 \text{ seg}$$

### SOLUCION PROBLEMA 3.31

#### DATOS

- Periféricos conectados: 2 discos duros, 1 CDROM, 1 impresora.
- $PI_1$  petición de interrupción disco duro.
- $PI_2$  petición de interrupción CDROM.
- $PI_3$  petición de interrupción impresora.
- Interrupciones anidables con  $\text{prio}(PI_1) > \text{prio}(PI_2) > \text{prio}(PI_3)$
- Tiempo de servicio de las rutinas de servicio en unidades de tiempo (u.t)
  - $t_{s1}=10$  u.t (disco duro)
  - $t_{s2}=30$  u.t (CDROM)
  - $t_{s3}=15$  u.t (impresora)
- En  $t=0$  u.t petición de interrupción del primer disco duro.
- En  $t=3$  u.t petición de interrupción de la impresora.
- En  $t=5$  u.t petición de interrupción del CDROM.
- En  $t=14$  u.t petición de interrupción del segundo disco duro.

Conviene darse cuenta de que el sistema permite **anidamiento**, en consecuencia se trata de un **sistema multinivel** de interrupciones, de tal forma que cuando comienza a ejecutarse el programa de gestión de una interrupción, si llega una interrupción de prioridad superior pasa a ejecutar la rutina de servicio de dicha interrupción.

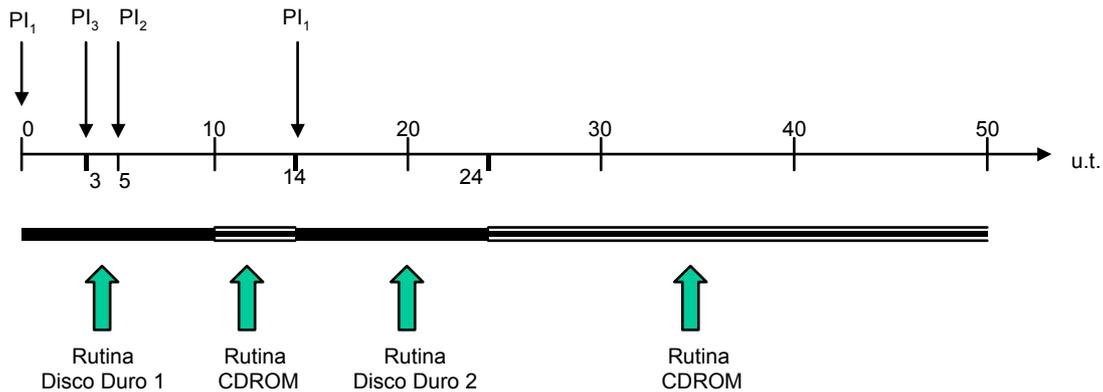


Figura 3.31.1

De acuerdo con el enunciado la forma de atender a las interrupciones (ver Figura 3.31.1) es la siguiente.

- 1) En  $t=0$  u.t comienza a ejecutar la rutina de servicio del primer disco duro, que requiere un tiempo total de ejecución de  $t_{s1}=10$  u.t.
- 2) En  $t=3$  u.t la CPU está ejecutando todavía la rutina de servicio del primer disco duro llega la petición de interrupción de la impresora  $PI_3$ . Puesto que  $prio(PI_3)<prio(PI_1)$  la ignora y la coloca en una cola a la espera de ser atendida.
- 3) En  $t=5$  u.t la CPU está ejecutando todavía la rutina de servicio del primer disco duro llega la petición de interrupción del CD-ROM  $PI_2$ . Puesto que  $prio(PI_2)<prio(PI_1)$  la ignora y la coloca en una cola a la espera de ser atendida.
- 4) En  $t=10$  u.t la CPU termina de ejecutar la rutina de servicio del primer disco duro y comienza a ejecutar la rutina de servicio del CD-ROM, que es la interrupción en cola de espera de mayor prioridad. La ejecución de esta rutina de servicio requiere un tiempo total  $t_{s2}=30$  seg.
- 5) En  $t=14$  u.t la CPU está ejecutando todavía la rutina de servicio del CDROM llega la petición de interrupción del segundo disco duro  $PI_1$ . Puesto que  $prio(PI_1)>prio(PI_2)$  detiene la ejecución de la rutina de servicio del CDROM, cuando aún le restan  $t_{s2}-4=26$  u.t para terminarla, y comienza a ejecutar la rutina de servicio del segundo disco duro que requiere un tiempo total en ser atendida de  $t_{s1}=10$  u.t
- 6) En  $t=24$  u.t termina la ejecución de la rutina de servicio del segundo disco duro. Retoma la CPU ejecución de la rutina de servicio del CDROM del que le faltaban 26 u.t por ejecutar.
- 7) En  $t=50$  u.t finaliza la rutina de servicio del CDROM y comienza la CPU a ejecutar la rutina de servicio de la impresora que se encontraba en la cola de espera, que requiere un tiempo total de  $t_{s3}=15$  u.t
- 8) En  $t=65$  u.t finaliza la ejecución de la rutina de servicio de la impresora.

Luego la respuesta al problema se obtiene en el paso 7, es decir, en  $t=50$  u.t finaliza la ejecución de la rutina de servicio de la interrupción del CDROM. Otra forma de calcular este valor es darse cuenta de que es la suma de los tiempos  $2 \cdot t_{s1} + t_{s2} = 2 \cdot 10 + 30 = 50$  u.t correspondientes a ejecutar tres rutinas de servicio : la del primer disco, la del segundo disco y la del CDROM.

## SOLUCION PROBLEMA 3.32

AFIRMACION I : **Verdadera**. Efectivamente el controlador de DMA necesita tener el control del bus del sistema para la realización de operaciones de transmisión de datos.

AFIRMACION II: **Falsa**. De hecho la estrategia de transmisión por ráfagas es la que produce una mayor interferencia con la CPU, es decir, una mayor reducción de la velocidad de ejecución de la CPU.

### SOLUCION PROBLEMA 3.33

En sistemas provistos de PE/S, la CPU es se encarga de iniciar la transferencia de E/S, y el PE/S ejecuta las instrucciones de E/S que se encuentran almacenadas en memoria principal.

AFIRMACION I : **Falsa**.

AFIRMACION II: **Falsa**.

### SOLUCION PROBLEMA 3.34

En el procesador Motorola 68000, las peticiones de interrupción de nivel 7 (IPL=7) se atienden siempre. Luego con los bits ( $I_2$   $I_1$   $I_0$ ) del registro de estado no se pueden enmascarar las interrupciones de nivel 7 ya que son **interrupciones no enmascarables**.

### SOLUCION PROBLEMA 3.35

La solución a este problema se encuentra en el archivo `probT3_35.pdf`

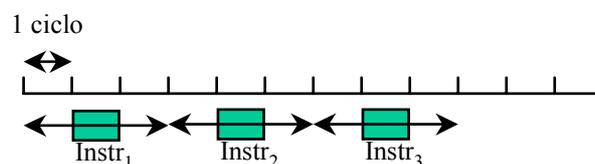
### SOLUCION PROBLEMA 3.36

#### DATOS

- Si CPU dispone en exclusiva del bus
  - Las instrucciones emplean 3 ciclos de reloj en ser ejecutadas.
  - En 1 de esos ciclos, el segundo, no requiere el acceso al bus.
- El controlador de DMA tarda dos ciclos de reloj consecutivos en transmitir una palabra.

AFIRMACION I : Es **verdadera**, ya que en la estrategia de DMA por ráfagas el controlador de DMA hace uso del bus cuando se ha finalizado la ejecución de una instrucción y no lo libera hasta haber finalizado de enviar todas las palabras.

AFIRMACION II: Es **falsa**., en la estrategia de DMA transparente, el controlador de DMA hace uso del bus en aquellos ciclos durante la ejecución de una instrucción en que la CPU no necesita utilizarlo. Puesto que el controlador de DMA requiere dos ciclos consecutivos para enviar una palabra y por otra parte la CPU únicamente no utiliza el bus, en el segundo de los tres que requiere para ejecutar una instrucción (ver Figura), nunca hay dos ciclos consecutivos en que el bus este libre y por lo tanto no se puede implementar el DMA con esta estrategia.



**SOLUCION PROBLEMA 3.37****DATOS**

- Canal multiplexor
- 5 unidades de disco =>  $v_{TD}=10^6$  bytes/seg
- 5 unidades de cinta =>  $v_{TC}=10^5$  bytes/seg
- $t_{bus}=50$  nseg

En primer lugar se va a obtener la velocidad de transferencia del canal multiplexor  $v_{Tcanal}$ , que viene dada por la suma de las velocidades de cada uno de los dispositivos que controla.

$$v_{Tcanal} = 5 \cdot v_{TD} + 5 \cdot v_{TC} = 5 \cdot 10^6 + 5 \cdot 10^5 = 5.5 \cdot 10^6 \text{ (bytes / seg)}$$

La velocidad de transferencia del canal,  $v_{Tcanal}$ , indica el número de bytes que se pueden transmitir entre un periférico y la memoria principal en un segundo. A partir de  $v_{Tcanal}$  se puede obtener el tiempo necesario para transmitir un byte entre un periférico y  $M_p$ .

$$t_{Tcanal} = \frac{1}{v_{Tcanal}} = \frac{1}{5.5 \cdot 10^6} = \frac{10^{-9}}{5.5 \cdot 10^6 \cdot 10^{-9}} = \frac{1000}{5.5} \text{ nseg} = 181.81 \text{ nseg}$$

La velocidad máxima de transferencia agregada a este sistema es:

$$v_{Tmax} = \frac{1}{\max(\bar{t}_{bus}, t_{Tcanal})} = \frac{1}{t_{Tcanal}} = v_{Tcanal} = 5.5 \cdot 10^6 \text{ (pal / seg)}$$

**SOLUCION PROBLEMA 3.38****DATOS**

- $n=24$  líneas de dirección.
- Memoria con  $2^{20}$  palabras
- Utiliza E/S aislada de memoria.
- $R=1$  dirección/periférico

El computador utiliza E/S aislada de memoria, eso significa que el controlador de E/S y la memoria disponen de líneas de control (lectura y escritura) autónomas, por lo que el mapa de direcciones de palabras de memoria es independiente del mapa de direcciones de palabras en periféricos.

El número total de direcciones  $N_D$  viene definido por el número  $n$  de líneas de dirección.

$$N_D = 2^n = 2^{24} \text{ direcciones}$$

El número de direcciones para acceder a palabras en los periféricos  $N_{DP}$  será igual al número total de direcciones disponibles:

$$N_{DP} = N_D = 2^{24} \text{ direcciones}$$

Nos dicen que cada periférico ocupa 1 dirección =>  $R=1$  dirección/periférico. Luego el número máximo de periféricos  $N_{Pmax}$  es :

$$N_{Pmax} = \frac{N_{DP}}{R} = \frac{2^{24} \text{ (direcciones)}}{1 \text{ (dirección / periférico)}} = 2^{24} \text{ perifericos}$$

### SOLUCION PROBLEMA 3.39

**Afirmación A:** VERDADERA. Efectivamente un PE/S de tipo canal selector controla múltiples dispositivos de alta velocidad.

**Afirmación B:** VERDADERA. Efectivamente un PE/S de tipo canal multiplexor puede controlar de forma simultánea operaciones de E/S con múltiples dispositivos.

**Afirmación C:** VERDADERA. En E/S controlada por programa la CPU realiza entre otras operaciones, la comprobación del estado del periférico.

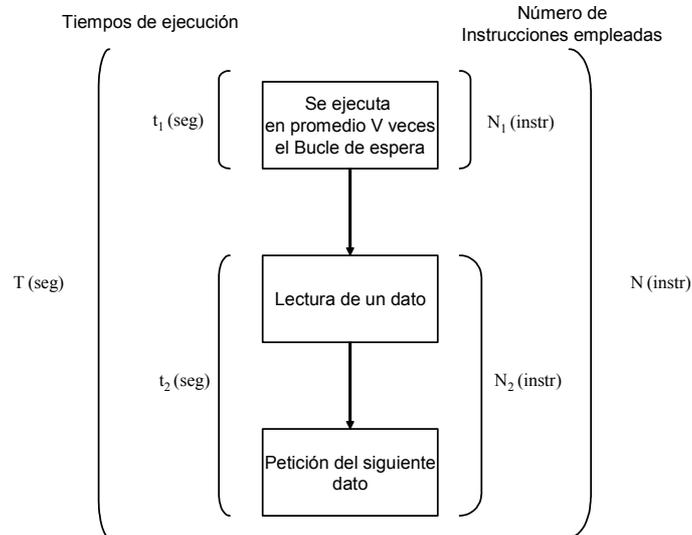
**Afirmación D:** FALSA. Las afirmaciones A, B y C son verdaderas.

### SOLUCION PROBLEMA 3.40

**DATOS**

- E/S controlada por programa.
- El dispositivo de E/S tarda 6 mseg en tener el dato solicitado. →  $t_1=6$  mseg
- Se supone que:
  - - El computador sólo se dedica a E/S.
  - - El bucle de espera se implementa con 1 instrucción. →  $N_{BE}=1$  instr
- - La lectura de un dato y la solicitud del siguiente dato por la CPU requiere 10 instrucciones. →  $N_2=10$  instrucciones
- - Cada instrucción de la CPU se ejecuta en  $t_i=200$   $\mu$ seg.

La E/S controlada por programa se puede considerar formada por tres etapas, tal y como se muestra en la siguiente figura:



El tiempo total  $T$  que se tarda en ejecutar una E/S controlada por programa, será la suma del tiempo  $t_1$  que permanece la CPU en el bucle de espera y del tiempo  $t_2$  que emplea la CPU en leer un dato y solicitar el siguiente:

$$T = t_1 + t_2$$

Del enunciado se sabe que  $t_1=6$  ms, además se sabe el número de instrucciones que emplea el programa en leer un dato y pedir el siguiente es  $N_2=10$  instr. Como también se conoce el tiempo que se emplea en ejecutar una instrucción  $t_i$ , entonces  $t_2$  se puede calcular de la siguiente forma:

$$t_2(\text{seg}) = N_2(\text{instr}) \cdot t_1(\text{seg} / \text{instr}) = 10(\text{instr}) \cdot 200(\mu\text{seg} / \text{instr}) = 2 \text{mseg}$$

Por lo tanto

$$T = 2 + 6 = 8 \text{mseg}$$

Luego el porcentaje de tiempo P que dedica la CPU al bucle de espera es:

$$P = \frac{t_1}{T} \cdot 100 = \frac{6}{8} \cdot 100 = 75\%$$

#### Comentarios

Aunque en el problema no lo piden con los datos de que disponemos y los que hemos calculado también sería posible calcular los siguientes:

- Número total de instrucciones  $N_T$  que emplea el programa de E/S:

$$N_T(\text{instr}) = \frac{T(\text{seg})}{t_1(\text{seg} / \text{instr})} = \frac{8 \cdot 10^{-3}}{200 \cdot 10^{-6}} = 40 \text{instr}$$

- Número total de instrucciones  $N_1$  que ejecuta el programa de E/S en el bucle de espera

$$N_1(\text{instr}) = \frac{t_1(\text{seg})}{t_1(\text{seg} / \text{instr})} = \frac{6 \cdot 10^{-3}}{200 \cdot 10^{-6}} = 30 \text{instr}$$

También puede calcularse como:

$$N_1 = N_T - N_2 = 40 - 10 = 30 \text{instr}$$

- De acuerdo con el enunciado ejecutar una vez el bucle de espera requiere de  $N_{BE}=1$  instr, luego el número V de veces que se ejecuta el bucle de espera es:

$$V(\text{veces}) = \frac{N_1(\text{instr})}{N_{BE}(\text{instr} / \text{vez})} = \frac{30}{1} = 30 \text{veces}$$

### **SOLUCION PROBLEMA 3.41**

La solución a este problema se encuentra en el archivo `probT3_41.pdf`

### **SOLUCION PROBLEMA 3.42**

#### **DATOS**

- Velocidad de lectura de una unidad de disco  $v_{TD}=2^{21}$  (pal/seg)
- Buffer interno de capacidad  $C_B=2^{20}$  pal.
- DMA por ráfagas.
- 4 ciclos/palabra
- 8 ciclos/instrucción si la CPU dispone en exclusiva del bus.
- $f=500 \times 10^6$  ciclos/s

El sistema sería el siguiente una unidad de disco que envía al computador la información que le solicita a través de un buffer. Es concreto, el disco le envía la información al buffer y cuando éste se llena utiliza DMA por ráfagas para enviar dicha información a la memoria principal.

Recuérdese que cuando se utiliza DMA por ráfagas el controlador de DMA toma posesión del BUS y no lo libera ( no puede utilizarlo la CPU) hasta transmitir todas las palabras que necesita. Se pide en el enunciado el número de instrucciones por segundo que deja de ejecutar la CPU cuando el BUS lo toma el controlador de DMA. P

Se resolvería este problema en tres pasos:

1) En primer lugar se debe calcular la velocidad de transferencia del buffer  $v_{TB}$  (pal/seg).

$$v_{TB} = \frac{C_B}{t_{fB}}$$

El tiempo de llenado  $t_{fB}$  del buffer es:

$$t_{fB} = \frac{C_B}{v_{TD}} = \frac{2^{20}}{2^{21}} = \frac{1}{2} = 0.5 \text{ seg}$$

Luego:

$$v_{TB} = \frac{C_B}{t_{fB}} = \frac{2^{20}}{0.5} = 2^{21} (\text{pal} / \text{seg})$$

2) A continuación hay que calcular los (ciclos/seg) que requiere la operación de DMA:

$$v_{TB} \left( \frac{\text{pal}}{\text{seg}} \right) * 4 \left( \frac{\text{ciclos}}{\text{pal}} \right) = 2^{21} * 2^2 = 2^{23} \left( \frac{\text{ciclos}}{\text{seg}} \right)$$

3) Y finalmente se calcularía las (instrucciones/seg) que dejarían de ejecutarse por realizarse la operación de DMA

$$\frac{2^{23} \left( \frac{\text{ciclos}}{\text{seg}} \right)}{8 \left( \frac{\text{ciclos}}{\text{instr}} \right)} = 2^{20} \left( \frac{\text{instr}}{\text{seg}} \right)$$

## SOLUCION PROBLEMA 3.43

### DATOS

- 8 líneas de interrupción  $PI_i$   $i=1, \dots, 8$ .
- Prioridades:  $PI_1 > PI_2 > PI_3 > \dots > PI_8$
- Registro de Estado (RE) de la CPU:  $[RE_1, RE_2, RE_3, RE_4, RE_5, RE_6, RE_7, RE_8]$
- $RE_1$  es el bit más significativo.
- Si  $RE_i=1$  la petición está enmascarada, si  $RE_i=0$  la petición está activa.
- En un determinado instante:
  - El registro de Estado contiene el valor  $6C_{16}$
  - No está siendo atendida ninguna interrupción.
  - Se solicita interrupción por las líneas 2 y 4.

Para resolver este ejercicio se debe expresar en primer lugar el contenido del registro en binario:

$$6C_{16} = 0110 \ 1100_2$$

Se observa que puesto que  $[RE_1, RE_2, RE_3, RE_4, RE_5, RE_6, RE_7, RE_8]$  las únicas interrupciones que no están enmascaradas y por lo tanto pueden ser atendidas son la  $PI_1, PI_4, PI_7$  y  $PI_8$ .

En consecuencia de las dos peticiones de interrupción que llegan ( $PI_2$  y  $PI_4$ ), sólo la  **$PI_4$  será atendida** puesto que es la única que no está enmascarada.

### SOLUCION PROBLEMA 3.44

**Afirmación A:** FALSA. El sentido (lectura/escritura) de la operación de E/S se lo debe indicar la CPU al controlador de DMA a través de las líneas de control adecuadas.

**Afirmación B:** FALSA. La CPU sólo participa al principio y al final de la operación de DMA.

**Afirmación C:** FALSA. Un procesador de E/S se comunica con la CPU mediante el uso de interrupciones.

**Afirmación D:** VERDADERA.

### SOLUCION PROBLEMA 3.45

**Afirmación A:** VERDADERA. Efectivamente el procesador de E/S de tipo *canal selector* se utiliza para controlar la E/S de varios dispositivos de alta velocidad.

**Afirmación B:** FALSA. Ya que se utiliza para controlar la E/S de varios dispositivos de BAJA velocidad.

**Afirmación C:** FALSA. En este tipo de E/S la intervención de la CPU no es mínima.

**Afirmación D:** FALSA.

### SOLUCION PROBLEMA 3.46

#### DATOS

- Circuito de gestión de prioridades
- $P_0 > P_1 > P_2 > P_3$

La salida  $INTR$  se debe activar cuando exista alguna petición de interrupción, es decir, cuando cualquiera de las entradas  $INTR_i$   $i=0,1,2,3$  este activada. Luego la expresión lógica que define  $INTR$  es:

$$INTR = INTR_3 + INTR_2 + INTR_1 + INTR_0$$

Luego la respuesta A) es cierta.

Por otra parte la respuesta B también es cierta, ya que teniendo en cuenta la relación lógica

$$a + \bar{a}xb = a + b$$

se puede demostrar que la función propuesta en B) es equivalente a la función lógica propuesta en A)

$$INTR = INTR_3 + \overline{INTR_3}(INTR_2 + \overline{INTR_2}(INTR_1 + \overline{INTR_1}INTR_0)),$$

$$INTR = INTR_3 + \overline{INTR_3}(INTR_2 + \overline{INTR_2}(INTR_1 + INTR_0))$$

$$INTR = INTR_3 + \overline{INTR_3}(INTR_2 + (INTR_1 + INTR_0))$$

$$INTR = INTR_3 + (INTR_2 + (INTR_1 + INTR_0))$$

$$INTR = INTR_3 + INTR_2 + INTR_1 + INTR_0$$

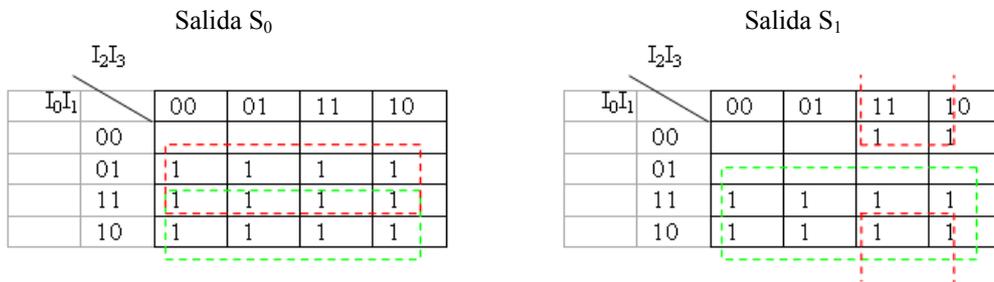
También se puede comprobar generando la tabla de verdad de la función y simplificando mediante mapas de Karnaugh, obteniéndose idéntico resultado. La respuesta C) es falsa ya que si no hay activación de una de las  $INTR_i$  el resto de peticiones quedan enmascaradas y por tanto no se transmite la petición de interrupción a la CPU. Por tanto **la solución final es la D) Dos de las anteriores son ciertas.**

### SOLUCION PROBLEMA 3.47

**DATOS**

- Codificador de prioridad con 4 entradas ( $I_0, I_1, I_2, I_3$ ) y 3 salidas ( $S_0, S_1, PI$ )
- Tabla de funcionamiento de este dispositivo dada en el enunciado

Las X's de la tabla representan que el valor puede ser indistintamente 0 ó 1. Obsérvese que PI se pone a 1 cuando una o más entradas valen 1. Si todas las entradas son 0's entonces  $PI=0$  y las otras salidas del codificador no se usan (por lo que están marcadas en la tabla con X's). En la siguiente figura se muestran los diagramas de Karnaugh que simplifican las salidas.



Luego las funciones lógicas de  $S_0$  y  $S_1$  son:

$$S_0 = I_0 + I_1$$

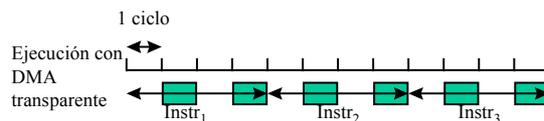
$$S_1 = I_0 + I_2 \bar{I}_1$$

Luego la respuesta correcta es la C.

### SOLUCION PROBLEMA 3.48

**DATOS**

- DMA transparente
- 4 ciclos/instrucción
- 2 ciclos/instrucción no requiere el uso del bus
- 1 ciclo/palabra
- frecuencia del reloj del computador es  $f=10^6$  ciclos/seg
- Se transmiten  $10^6$  palabras.



Se ha de recordar que en la estrategia de DMA transparente, el DMA se realiza en aquellas fases de la ejecución de una instrucción en que la CPU no hace uso del bus. El enunciado del problema nos dice que en promedio el computador necesita 4 ciclos para ejecutar una instrucción y que sólo en dos de ellos no requiere el uso del bus. Además nos dicen que el controlador de DMA únicamente necesita de 1 ciclo para transmitir una palabra. Luego el número de palabras que se transmiten por DMA en la ejecución de una instrucción ( $R$ ), vendría dado por:

$$R = \frac{2(\text{ciclos} / \text{instr})}{1(\text{ciclo} / \text{pal})} = 2(\text{pal} / \text{instr})$$

La velocidad de ejecución es:

$$v_{\text{ejec}} = \frac{f(\text{ciclos} / \text{seg})}{4(\text{ciclos} / \text{inst})} = \frac{10^6}{4} = 2.5 \cdot 10^5 (\text{instr} / \text{seg})$$

La velocidad de transferencia es:

$$v_t = R \cdot v_{\text{ejec}} = 2(\text{pal} / \text{inst}) \cdot 2.5 \cdot 10^5 (\text{inst} / \text{seg}) = 5 \cdot 10^5 (\text{pal} / \text{seg})$$

Por otra parte, la velocidad de transferencia se define como

$$v_T = \frac{N}{T}$$

donde  $N$  es el número de palabras que se transmiten y  $T$  el tiempo que tardan en transmitirse. Despejando  $T$  de la expresión anterior

$$T = \frac{N}{v_T}$$

y sustituyendo los valores de  $N$  (dado en el enunciado) y  $v_T$ , se obtiene:

$$T = \frac{10^6}{5 \cdot 10^5} = 2 \text{ s}$$

## SOLUCION PROBLEMA 3.49

### DATOS

- DMA mediante robo de ciclos
- CPU ejecuta 1 MIPS
- La velocidad del procesador debido al DMA se reduce un 0.89%
- 8 bits/palabra
- 2 ciclos/instrucción

La velocidad del procesador sin DMA es un datos del enunciado  $v_{e0}=1 \text{ MIPS}=10^6$  (instrucciones/s).

La velocidad del procesador con DMA con robo de ciclos es  $v_{e1}$  y se podría calcular a partir del dato del porcentaje de reducción de la velocidad del procesador debido al DMA:

$$P_{\text{red}} \equiv \frac{v_{e1}}{v_{e0}} \cdot 100 = 0.89$$

Despejando  $v_{e1}$  de la expresión anterior se obtiene:

$$v_{e1} = \left( \frac{P_{\text{red}}}{100} \right) \cdot v_{e0} = 0.89 \cdot 10^6 (\text{instrucciones} / \text{s})$$

La velocidad de transferencia del DMA en (palabras/s) es proporcional a la velocidad de ejecución del sistema con DMA:

$$v_{t1} = R \cdot v_{e1}$$

donde  $R$  es el número de palabras que se transmiten por DMA en la ejecución de una instrucción. Del enunciado se sabe que se usan dos ciclos de CPU por instrucción, suponiendo que el controlador de DMA únicamente necesita de 1 ciclo para transmitir una palabra, lo cual es lo habitual en la estrategia de DMA por robo de ciclos, se tendría:

$$R = \frac{2(\text{ciclos / instrucción})}{1(\text{ciclo / palabra})} = 2(\text{palabras / instrucción})$$

Luego sustituyendo y operando se obtendría que:

$$v_{t1} = 2 \cdot 0.89 \cdot 10^4 = 17800 (\text{palabras / s})$$

Puesto que se conoce del enunciado el número de bits que ocupa una palabra la velocidad de transferencia se puede expresar en (bits/s):

$$v_{t1} = 17800(\text{pal / s}) \cdot 8(\text{bits / pal}) = 142400 (\text{bits / s})$$

### SOLUCION PROBLEMA 3.50

#### DATOS

- Velocidad de transferencia del periférico  $1.5 \cdot 10^6$  (bytes/s).
- El procesador tarda  $t_I=200$  ns en ejecutar una instrucción.
- $N_{\text{inicioDMA}}=10$  instrucciones
- En la transferencia de 1 byte ocupa los buses durante 100 ns.
- Se desean transferir 500 bytes
- Se supone que la CPU está parada mientras el DMA ocupa el bus.

A) El tiempo  $T$  que dedica el procesador a la transferencia de los 500 bytes será la suma de dos componentes:

$$T = T_{\text{inicioDMA}} + T_{\text{cesiónBUS}}$$

donde  $T_{\text{inicioDMA}}$  es el tiempo en gestionar la inicialización del proceso de DMA y  $T_{\text{cesiónBUS}}$  es el tiempo que deja de usar la CPU el bus por cedérselo al controlador de DMA.

$T_{\text{inicioDMA}}$  se puede calcular de la siguiente forma:

$$T_{\text{inicioDMA}} = N_{\text{inicioDMA}}(\text{instrucciones}) \cdot t_I(\text{s / instrucción}) = 10 \cdot 200 \cdot 10^{-9} = 2 \cdot 10^{-6} \text{ s} = 2 \mu\text{s}$$

Asimismo puesto que hay que transmitir 500 bytes y en la transmisión de cada byte se ocupan los buses durante 100 ns, entonces:

$$T_{\text{cesiónBUS}} = 500(\text{bytes}) \cdot 100 \cdot 10^{-9} (\text{ns / byte}) = 50 \cdot 10^{-6} \text{ s} = 50 \mu\text{s}$$

Luego

$$T = T_{\text{inicioDMA}} + T_{\text{cesiónBUS}} = 2 + 50 = 52 \mu\text{s}$$

B) El tiempo de transferencia de un bloque de datos una vez que se ha inicializado el DMA es:

$$T_t = \frac{500}{1.5 \cdot 10^6} \approx 333 \mu s$$

De este tiempo, la CPU podría ejecutar instrucciones solamente si no tiene que ceder el uso del BUS al controlador del DMA. En el apartado anterior se cálculo que  $T_{cesiónBUS} = 50 \mu s$ . Luego el tiempo útil  $T_{útil}$  de que dispone la CPU para ejecutar instrucciones es:

$$T_{útil} = T_t - T_{cesiónBUS} = 333 - 50 = 283 \mu s$$

Dividiendo este tiempo por el tiempo que requiere el procesador en ejecutar una instrucción, se obtendría el número total de instrucciones  $N$  que puede dedicar la CPU a ejecutar otro proceso mientras se está realizando la transferencia por DMA:

$$N = \frac{T_{útil}(s)}{t_I(s / instrucción)} = \frac{283 \cdot 10^{-6}}{200 \cdot 10^{-9}} = 1415 \text{ instrucciones}$$

## SOLUCION PROBLEMA 3.51

### DATOS

- Procesador de 32 bits.
- Velocidad de ejecución del procesador  $v_e = 4 \text{ MIPS} = 4 \cdot 10^6$  (instrucciones/s)
- Tiempo empleado en un cambio de contexto  $T_{cc} = 1.25 \mu s$
- Instrucciones empleadas en un cambio de contexto  $N_{cc} = 5$  instrucciones
- Tamaño bloque de datos de 512 bytes
- Velocidad de transferencia del periférico  $v_T = 1 \text{ Kbyte} = 2^{10}$  bytes.
- El módulo de E/S interrumpe cada vez que se recibe 1 byte.
- 20 instrucciones en ejecutar la rutina de tratamiento de la interrupción de este periférico.

El tiempo de transferencia de un bloque de datos desde el periférico es:

$$T_t = \frac{512(\text{bytes})}{2^{10}(\text{bytes} / s)} = 0.5 s$$

En este tiempo si no se produjese ninguna interrupción la CPU podría ejecutar:

$$N_{TCPU} = T_t \cdot v_e = 0.5 \cdot 4 \cdot 10^6 = 2 \cdot 10^6 \text{ instrucciones}$$

Cada vez que se recibe 1 byte el periférico interrumpe, por lo tanto como un bloque consta de 512 bytes habrá  $K=512$  (interrupciones/bloque).

El número total de instrucciones asociadas a una interrupción es la suma de tres componentes:

$$N_I = N_{CC} + N_{RS} + N_{CC} = 2 \cdot N_{CC} + N_{RS}$$

donde:

- $N_{CC}$  son las instrucciones que se emplean en realizar un cambio de contexto. Debe recordarse que la atención de una interrupción requiere de la realización de dos cambios de contexto, uno en su inicio y otro en su finalización. Del enunciado se sabe que  $N_{CC} = 5$  instrucciones
- $N_{RS}$  es el número de instrucciones que requiere la rutina de tratamiento de la interrupción asociada a este periférico. Del enunciado se sabe que  $N_{RS} = 20$  instrucciones.

Luego

$$N_I = 2 \cdot N_{CC} + N_{RS} = 2 \cdot 5 + 20 = 30 \text{ instrucciones}$$

Puesto que la transmisión del bloque produce 512 interrupciones, el número total de instrucciones que se ejecutan asociadas a dichas interrupciones es:

$$N_{TI} = K \cdot N_I = 512 \cdot 30 = 15360 \text{ (instrucciones)}$$

Por lo tanto, el número de instrucciones  $N_{OP}$  que puede dedicar la CPU a la ejecución de otros procesos mientras se está transmitiendo el bloque de datos es:

$$N_{OP} = N_{TCPU} - N_{TI} = 2 \cdot 10^6 - 15360 = 1984640 \text{ instrucciones}$$

Finalmente, el porcentaje  $P$  de instrucciones que puede dedicar el procesador a otros procesos durante la transferencia de un bloque del periférico es:

$$P = \left( \frac{N_{OP}}{N_{TCPU}} \right) \cdot 100 = \left( \frac{1984640}{2 \cdot 10^6} \right) \cdot 100 = 99.23\%$$

## SOLUCION PROBLEMA 3.52

### DATOS

- Ocho líneas de interrupción ( $PI_0, \dots, PI_7$ )
- Prioridad ( $PI_0$ ) > Prioridad( $PI_1$ ) > ... > Prioridad( $PI_7$ )
- Secuencia de interrupciones: 4, 7, 1, 3, 0, 5, 6, 4, 2, 1
- Durante la gestión de una interrupción pueden llegar dos peticiones de interrupción.
- No se permite anidamiento

Se van a detallar los pasos en la gestión de estas interrupciones:

- 1) En primer lugar se comienza ejecutando la interrupción 4. Durante su ejecución llegan las interrupciones 7 y 1. Luego la cola de interrupciones pendientes sería Cola={7,1}.
- 2) De las interrupciones pendientes Cola={7,1}, se atiende 1 por ser la más prioritaria. Durante su ejecución llegan las interrupciones 3 y 0. Luego la cola de interrupciones pendientes sería Cola={7,3,0}.
- 3) De las interrupciones pendientes Cola={7,3,0}, se atiende 0 por ser la más prioritaria. Durante su ejecución llegan las interrupciones 5 y 6. Luego la cola de interrupciones pendientes sería Cola={7,3,5,6}.
- 4) De las interrupciones pendientes Cola={7,3,5,6}, se atiende 3 por ser la más prioritaria. Durante su ejecución llegan las interrupciones 4 y 2. Luego la cola de interrupciones pendientes sería Cola={7,5,6,4,2}.
- 5) De las interrupciones pendientes Cola={7,5,6,4,2}, se atiende 2 por ser la más prioritaria. Durante su ejecución llega la interrupción 1. Luego la cola de interrupciones pendientes sería Cola={7,5,6,4,1}.
- 6) De las interrupciones pendientes Cola={7,5,6,4,1}, se atiende 1 por ser la más prioritaria. La cola de interrupciones pendientes sería Cola={7,5,6,4}.
- 7) De las interrupciones pendientes Cola={7,5,6,4}, se atiende 4 por ser la más prioritaria. La cola de interrupciones pendientes sería Cola={7,5,6}.
- 8) De las interrupciones pendientes Cola={7,5,6}, se atiende 5 por ser la más prioritaria. La cola de interrupciones pendientes sería Cola={7,6}.
- 9) De las interrupciones pendientes Cola={7,6}, se atiende 6 por ser la más prioritaria. La cola de interrupciones pendientes sería Cola={7}.
- 10) Se atiende la interrupción 7 que es la última que queda pendiente

Luego el orden de atención de las interrupciones es: 4, 1, 0, 3, 2, 1, 4, 5, 6 y 7