

# Estructura y Tecnología de Computadores

## Examen final

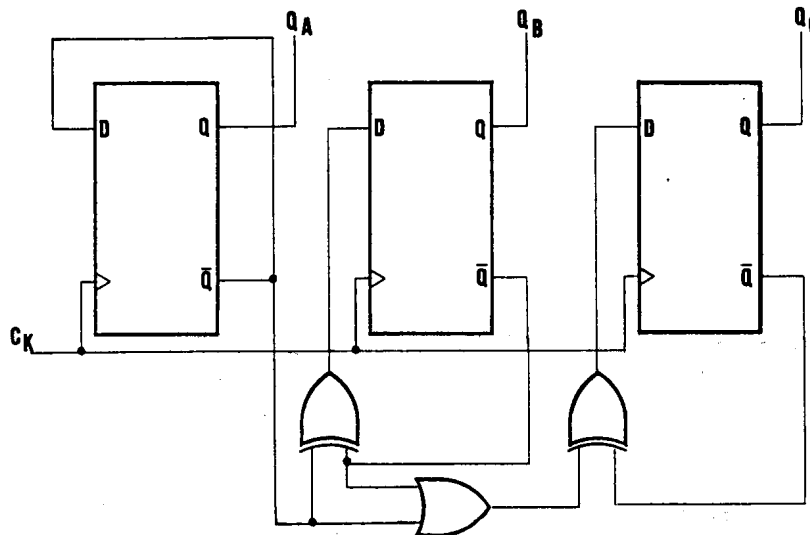
2 de febrero de 2011

Grupo:	DNI:
Nombre:	Apellidos:

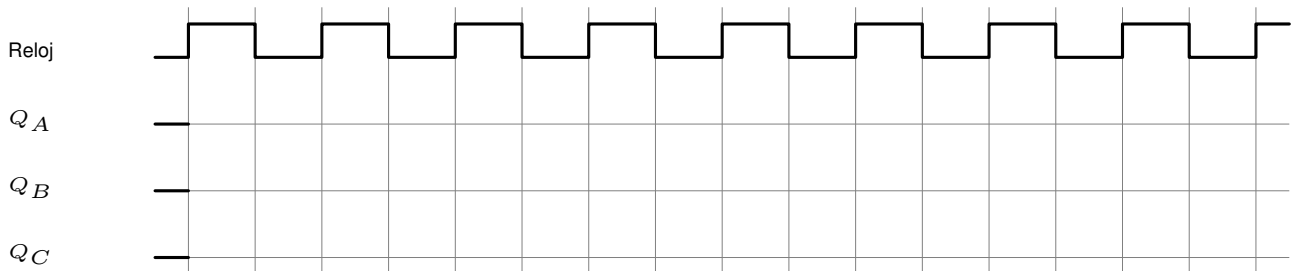
### Instrucciones:

- **Sólo en una de las preguntas se podrá obtener una nota inferior a 3.** Si no se cumple esta condición, el examen estará suspenso.
- La nota obtenida en el examen, una vez se divida por 4, representará el 50% de la nota final de la asignatura, siendo necesario obtener una calificación mayor o igual que 5 para eliminar esta parte.
- No se dará por válida ninguna respuesta que no contenga el desarrollo de la solución paso a paso.
- El peso de cada ejercicio aparece entre paréntesis.
- Responda a las preguntas en bolígrafo.

1. (10 puntos) Dibujar el cronograma y obtener la secuencia de cuenta del contador de la figura:



¿Cuál es la frecuencia máxima de funcionamiento si el retraso de un biestable es de 40 ns, el de una puerta AND u OR es de 20 ns y el de una XOR 25 ns?



2. (10 puntos) Utilizando la metodología descrita en los apuntes para la inclusión de nuevas instrucciones en el esquema de implementación multiciclo, realice las fases de análisis y diseño de la siguiente instrucción:

**pushm (\$a):** Apila el valor contenido en la dirección de memoria contenida en el registro \$a. Nótese que la dirección de memoria no se puede especificar con desplazamiento.

Para que la solución del ejercicio se considere correcta, es importante que incluya claramente todos los pasos correspondientes de las fases de análisis y diseño. Las modificaciones al camino de datos y la unidad de control deberán estar explicadas por escrito. De forma complementaria, se pueden realizar modificaciones y anotaciones sobre los diagramas originales para facilitar la explicación.

3. (10 puntos) Se desea diseñar una memoria virtual para un procesador que cumpla los siguientes requisitos:

- Direcciones físicas de 24 bits y memoria direccionable por bytes.
- La consulta de la tabla de páginas debe realizarse en un único acceso a memoria. Además, la tabla de páginas no puede ocupar más del 6,25 % (la dieciseisava parte) de la capacidad máxima de la memoria física. Cada entrada de la tabla de páginas tiene un tamaño de 2 bytes.
- Para los fallos de página se empleará una política de reemplazo pseudo-LRU (un sólo bit de uso) y una estrategia de post-escritura.

Se pide:

- a) Según las especificaciones anteriores, diseñar el formato óptimo de las direcciones virtuales y físicas y de la tabla de páginas si se desea implantar una memoria virtual con la mayor capacidad posible. Especificar detalladamente el formato de la dirección física y de la dirección virtual, dibujar un esquema de la tabla de páginas detallando los bits de control necesarios (justificando su inclusión).
  - b) Se desea implantar también un TLB 2-asociativo de dos conjuntos (4 bloques en total) con estrategia de reemplazo LRU y una política de post-escritura. Dibujar un esquema detallado del TLB y calcular el tamaño exacto incluyendo los bits de control necesarios (justificando su inclusión).
  - c) Mostrar de forma exacta cómo evoluciona el contenido del TLB y de la tabla de páginas después de cada una de las siguientes referencias virtuales generadas por el procesador expresadas en **hexadecimal**: lectura 810, lectura 1010, escritura 800 y escritura 2000. Obtener también la dirección física resultante. (Nota: para los fallos de página, suponga que las primeras direcciones bajas de memoria están libres.)
4. (10 puntos) Un disco duro se encuentra conectado a un ordenador a través de un bus de entrada/salida síncrono que posee 32 líneas para direcciones y 32 líneas para datos. Transmitir una dirección o un dato consume un ciclo de bus. Leer o escribir en memoria supone una latencia de 3 ciclos. A partir del quinto ciclo, el sistema puede leer o escribir hasta 4 palabras de 32 bits a razón de una palabra por ciclo de bus. Además, para las escrituras se necesitan dos ciclos adicionales para escribir un código de corrección de errores. Si sabemos que tanto el bus como el procesador tienen una frecuencia de funcionamiento de 500 Mhz y que el 70 % de operaciones son de lectura y el 30 % de escritura, se pide:
- a) Porcentaje de uso de la CPU si se usan interrupciones en la gestión de la entrada/salida. Sabemos que en cada interrupción se transfieren 8 palabras y supone un gasto de 10 ciclos.
  - b) Porcentaje de uso de la CPU si se usa DMA en la gestión de la entrada/salida. Sabemos que inicializar el DMA consume 200 ciclos y 100 ciclos el tratamiento de la interrupción. En cada operación se transfiere un bloque de 10 KB.
  - c) Otra opción para gestionar la entrada/salida es mediante sondeo. Se establece como requisito que en esta tarea nunca se utilice más de un 49 % del tiempo del procesador. Además, sabemos que para realizar cada sondeo se necesitan un total de 20 ciclos. Sabiendo esto, ¿qué número mínimo de palabras se necesitará transmitir en cada sondeo para que se cumpla el requisito anteriormente dado?
  - d) Queremos transferir a memoria un archivo de 30 GB que se encuentra en disco mediante el método de DMA. ¿Qué tiempo se necesitará para realizar esta tarea? Nota: tenga en cuenta que mientras se programa y se trata la interrupción del DMA no se transfiere ningún dato.

# Soluciones

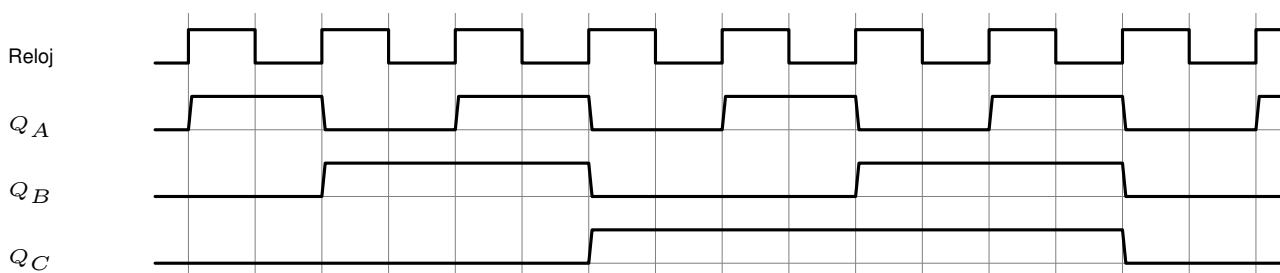
## Ejercicio 1

Del circuito obtenemos las siguientes ecuaciones del estado siguiente:

$$Q_A^* = \overline{Q_A}$$

$$Q_B^* = \overline{Q_A} \oplus \overline{Q_B} = Q_A \oplus Q_B$$

$$Q_C^* = (\overline{Q_A} + \overline{Q_B}) \oplus \overline{Q_B} = Q_A Q_B \overline{Q_C} + \overline{Q_A} Q_C + \overline{Q_B} Q_C$$



Se trata, por tanto, de un contador binario de 3 bits cuya cuenta es: 0-1-2-3-4-5-6-7.

La frecuencia máxima de funcionamiento viene determinada por el camino más largo desde que se produce el flanco activo del reloj (flanco ascendente) hasta que las salidas de los biestables y las entradas de excitación de los mismos están estables. En nuestro caso, el camino más largo es la entrada D del biestable cuya salida es  $Q_C$ .

$$T_{min} = T_{FF} + T_{OR} + T_{XOR} = 40 + 20 + 25 = 85ns$$

$$f_{max} = \frac{1}{T_{min}} = \frac{1000}{85} \times 10^6 Hz = 11.76MHz$$

## Ejercicio 2

### ■ Análisis:

#### 1. Especificación precisa de la semántica de la instrucción:

La semántica de “pushm (\$a)” es:

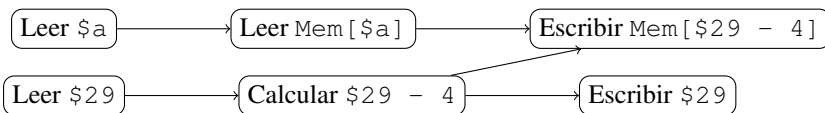
$$\begin{aligned} \$29 &\leftarrow \$29 - 4 \\ \text{Mem}[\$29] &\leftarrow \text{Mem}[\$a] \end{aligned}$$

(suponiendo que ambas operaciones se realizan secuencialmente)

#### 2. Identificación del trabajo a realizar por cada unidad funcional principal:

- Banco de registros:
  - Leer \$a.
  - Leer \$29.
  - Escribir \$29.
- ALU:
  - Calcular  $\$29 - 4$ .
- Memoria:
  - Leer Mem[\$a].
  - Escribir Mem[\$29].

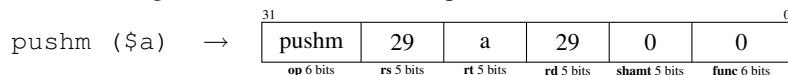
#### 3. Establecimiento del orden de precedencia entre las distintas tareas a realizar:



■ **Diseño:**

- Definición de la codificación de la instrucción:** Aunque la instrucción tiene un operando en memoria, éste se especifica sin desplazamiento. Por tanto, podemos utilizar el formato R para codificarla. Utilizando este formato, podemos usar los campos **rs** y **rt** para codificar el registro 29 en el cual hay que leer y escribir.

Usaremos la siguiente distribución de campos:



2. **División del trabajo en ciclos:**

Ciclo 1	Ciclo 2	Ciclo 3	Ciclo 4
IR ← Memoria[PC]	A ← \$29	ALUOut ← A - 4	Memoria[ALUOut] ← MDR
PC ← PC + 4	B ← \$a	MDR ← Memoria[B]	\$29 ← ALUOut

3. **Extensión del camino de datos:**

Para realizar algunas de las acciones especificadas en la tabla anterior necesitamos modificar el camino de datos:

**MDR ← Memoria[B]:** Añadir una nueva entrada al multiplexor controlado por la señal de control IoD que esté conectada al registro B (en el que tenemos el contenido de \$a). La señal IoD pasará a ser una señal de dos bits en lugar de uno, y la combinación “10” seleccionará la nueva entrada.

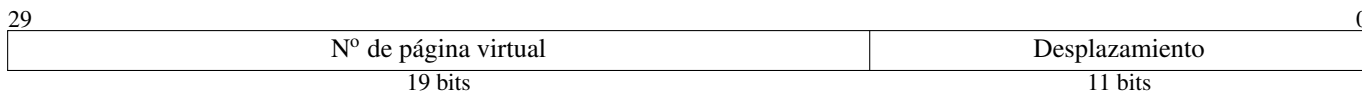
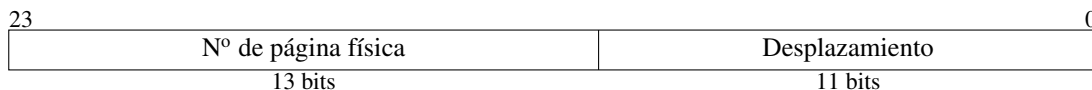
**Memoria[ALUOut] ← MDR:** Añadir un multiplexor a la entrada “Dato a escribir” de la memoria que estará controlado por una nueva señal que llamaremos MemSrc. Esta señal podrá tomar el valor 0 para que el dato que se escriba en la memoria sea el contenido en el registro B o 1 para que el dato a escribir sea el contenido en el registro de datos de memoria (MDR). La nueva señal tomaría el valor 0 en los estados preexistentes del autómata de control.

4. **Extensión del control:**

Habrá que añadir 2 nuevos estados al autómata de control original. El resultado se muestra en la figura 1.

### Ejercicio 3

- a) Los 24 bits de las direcciones físicas de memoria indican que disponemos de una memoria física máxima de 16MB ( $2^{24}$  bytes), de los cuales reservamos 1/16 para alojar la tabla de páginas cuyo tamaño será de 1MB. Con ese tamaño tendremos 512K-entradas para la tabla de páginas con dos bytes por entrada. En esos dos bytes se debe codificar el bit de uso (U), el bit de modificación (M) y el bit de validez (V), por lo que quedan 13 bits para el número de página física. El campo desplazamiento serán los 11 bits restantes hasta conseguir los 24 bits totales de la dirección física de memoria. Luego el tamaño de página será de 2KB ( $2^{11}$  bytes). Como tenemos 512K-entradas en la tabla de páginas, necesitamos 19 bits para indicar el número de entrada, es decir, el número de página virtual. Luego el tamaño de la memoria virtual será de 1GB ( $2^{19+11}$  bytes). Los formatos de la dirección física y de la dirección virtual serán, respectivamente:



La tabla de páginas queda de la siguiente manera:

	V	M	U	NPF (13 bits)
0	0	-	-	-
1	0	-	-	-
2	0	-	-	-
...	...	...	...	...
$2^{19} - 2$	0	-	-	-
$2^{19} - 1$	0	-	-	-

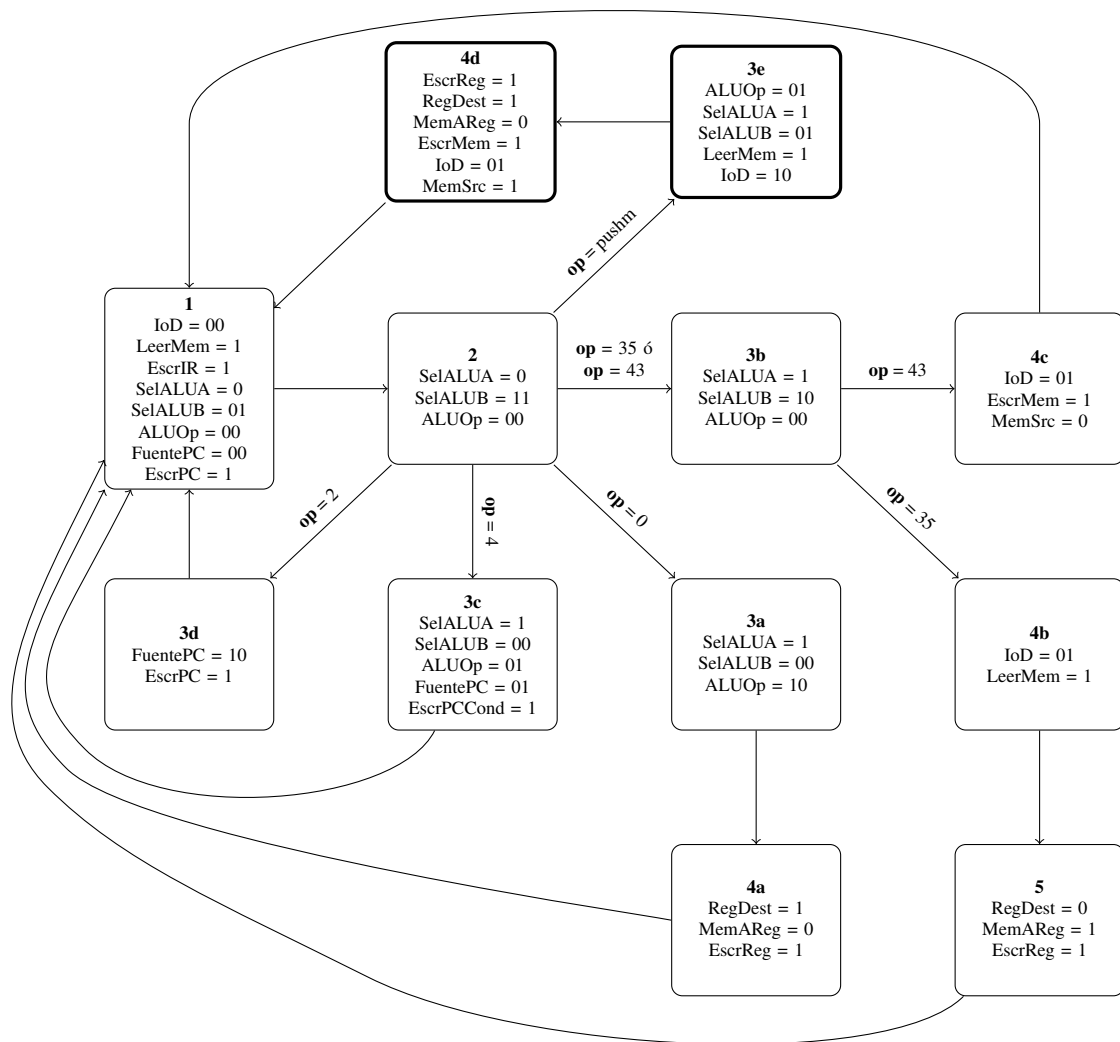


Figura 1: Modificaciones al autómata de control para incluir la instrucción `pushm`.

El bit V (validez) siempre se debe incluir para saber si el contenido de una entrada es válido o no. El bit M (modificación) se debe incluir porque para la tabla de páginas se utiliza una estrategia de post-escritura. Finalmente, el bit U (uso) se debe incluir para implementar un reemplazo pseudo-LRU.

		Vía 0						Vía 1							
		V	M	U	Etiqueta (18 bits)	M'	U'	NPF (13 bits)	V	M	U	Etiqueta (18 bits)	M'	U'	NPF (13 bits)
b)	0	0	-	-	-	-	-	-	0	-	-	-	-	-	-
	1	0	-	-	-	-	-	-	0	-	-	-	-	-	-

Tenemos 2 conjuntos y 2 vías, lo que nos permite almacenar en el TLB hasta 4 entradas distintas de la tabla de páginas. Por lo tanto, de los 19 bits que conforman el NPV, necesitamos 1 bit para el índice y nos quedan 18 bits para la etiqueta.

Tamaño =  $2 * 2 * (3+18+2+13)$  bits = 144 bits

El bit V (validez) siempre se debe incluir para saber si el contenido de una entrada es válido o no. El bit M (modificación) se debe incluir porque para el TLB se utiliza una estrategia de post-escritura. Finalmente, el bit U (uso) se debe incluir para implementar un reemplazo LRU. Generalmente, un único bit de uso no es suficiente para implementar un reemplazo LRU puro. Sin embargo, en este caso es más que suficiente porque cada conjunto tiene sólo dos entradas.

c) Para resolver esta parte, debemos tener en cuenta que las direcciones virtuales son de 30 bits, de los que 19 corresponden al número de página virtual (NPV) y 11 al desplazamiento. En el caso del TLB, de los 19 bits que corresponden al NPV, el bit menos significativo es el índice que selecciona el conjunto del TLB y los 18 bits restantes son la etiqueta.

- Lectura de la dirección  $0x810 = 00\ 0000\ 0000\ 0000\ 0000\ 1000\ 0001\ 0000)_2$ .

**NPV:**  $00\ 0000\ 0000\ 0000\ 0000\ 0000\ 1)_2 = 1$ .

**Desplazamiento:**  $000\ 0001\ 0000)_2$ .

**Acierto/fallo de TLB:** fallo (el TLB está inicialmente vacío).

**Acierto/fallo de página:** fallo (la tabla de páginas está inicialmente vacía). El S.O. lee de disco la página virtual 1 que carga en la página física 0 (la memoria está libre). Se actualiza la tabla de páginas:

	V	M	U	NPF (13 bits)
0	0	-	-	-
1	1	0	0	0
2	0	-	-	-
...	...	...	...	...
$2^{19} - 2$	0	-	-	-
$2^{19} - 1$	0	-	-	-

**Conjunto del TLB:** 1 (el bit menos significativos del NPV) y queda de la siguiente manera:

		Vía 0						Vía 1							
		V	M	U	Etiqueta (18 bits)	M'	U'	NPF (13 bits)	V	M	U	Etiqueta (18 bits)	M'	U'	NPF (13 bits)
0	0	0	-	-	-	-	-	-	0	-	-	-	-	-	-
	1	1	1	1	0	0	1	0	0	-	-	-	-	-	-

Ya que el bit de uso U' se pone a 1, el bit M del TLB debe ponerse también a 1.

**NPV:** 0.

**Dirección física resultante:**  $0000\ 0000\ 0000\ 0000\ 0001\ 0000)_2 = 0x10$ .

- Lectura de la dirección  $0x1010 = 00\ 0000\ 0000\ 0000\ 0001\ 0000\ 0001\ 0000)_2$ .

**NPV:**  $00\ 0000\ 0000\ 0000\ 0001\ 0)_2 = 2$ .

**Desplazamiento:**  $000\ 0001\ 0000)_2$ .

**Acierto/fallo de TLB:** fallo; las dos vías del conjunto 0 del TLB están vacías.

**Acierto/fallo de página:** fallo; la entrada 2 de la tabla de páginas está vacía. El S.O. lee de disco la página virtual 2 que almacena en la página física 1. Se actualiza la tabla de páginas:

	V	M	U	NPF (13 bits)
0	0	-	-	-
1	1	0	0	0
2	1	0	0	1
...	...	...	...	...
$2^{19} - 2$	0	-	-	-
$2^{19} - 1$	0	-	-	-

**Conjunto del TLB:** 0 (el bit menos significativos del NPV, quedando la etiqueta con valor 1) y queda de la siguiente manera:

Vía 0							Vía 1							
	V	M	U	Etiqueta (18 bits)	M'	U'	NPF (13 bits)	V	M	U	Etiqueta (18 bits)	M'	U'	NPF (13 bits)
0	1	1	1	1	0	1	1	0	-	-	-	-	-	-
1	1	1	1	0	0	1	0	0	-	-	-	-	-	-

Ya que el bit de uso U' se pone a 1, el bit M del TLB debe ponerse también a 1.

**NPF:** 1.

**Dirección física resultante:**  $0000\ 0000\ 0000\ 1000\ 0001\ 0000)_2 = 0x810$ .

- Escritura en la dirección  $0x800 = 00\ 0000\ 0000\ 0000\ 0000\ 1000\ 0000\ 0000)_2$ .

**NPV:**  $00\ 0000\ 0000\ 0000\ 0000\ 1)_2 = 1$ .

**Desplazamiento:**  $000\ 0000\ 0000)_2$ .

**Acierto/fallo de TLB:** la información para la traducción se encuentra en el TLB ya que en el segundo conjunto del TLB hay una entrada cuya etiqueta coincide con la obtenida del NPV. Si tenemos un acierto de TLB, obligatoriamente también hay un acierto en la tabla de páginas. Lo único que hay que hacer es actualizar el TLB para indicar que la página se ha modificado, pero no hay que actualizar la tabla de páginas pues el TLB tiene política de post-escritura:

Vía 0							Vía 1							
	V	M	U	Etiqueta (18 bits)	M'	U'	NPF (13 bits)	V	M	U	Etiqueta (18 bits)	M'	U'	NPF (13 bits)
0	1	1	1	1	0	1	1	0	-	-	-	-	-	-
1	1	1	1	0	1	1	0	0	-	-	-	-	-	-

**Acierto/fallo de página:** acierto y no se actualiza.

**NPF:** 0.

**Dirección física resultante:**  $0000\ 0000\ 0000\ 0000\ 0000\ 0000)_2 = 0$ .

- Escritura en la dirección  $0x2000 = 00\ 0000\ 0000\ 0000\ 0010\ 0000\ 0000\ 0000)_2$ .

**NPV:**  $00\ 0000\ 0000\ 0000\ 0010\ 0)_2 = 4$ .

**Desplazamiento:**  $000\ 0000\ 0000)_2$ .

**Acierto/fallo de TLB:** se produce un fallo de TLB, pues la entrada en el primer conjunto tiene una etiqueta distinta de 2.

**Acierto/fallo de página:** también se produce un fallo de página al no encontrarse la información en la tabla de páginas. El S.O. lee de disco la página virtual 4 que deja en la página física 2 (la siguiente página libre en memoria). Se actualiza la tabla de páginas:

	V	M	U	NPF (13 bits)
0	0	-	-	-
1	1	0	0	0
2	1	0	0	1
3	0	-	-	-
4	1	0	0	2
5	0	-	-	-
...	...	...	...	...
$2^{19} - 2$	0	-	-	-
$2^{19} - 1$	0	-	-	-

**Conjunto del TLB:** 0 (el bit menos significativos del NPV, quedando la etiqueta con valor 2). Al ser el TLB asociativo de 2 vías y al haber en el primer conjunto todavía un bloque libre, no es necesario expulsar nada del TLB. Lo que sí es necesario es poner a 0 el bit de uso del primer bloque del conjunto 0. Esto nos permite saber qué bloque se utilizó hace más tiempo para implementar así una política de reemplazo LRU. El TLB queda de la siguiente manera:

Vía 0							Vía 1							
	V	M	U	Etiqueta (18 bits)	M'	U'	NPF (13 bits)	V	M	U	Etiqueta (18 bits)	M'	U'	NPF (13 bits)
0	1	1	0	1	0	1	1	1	1	1	2	1	1	2
1	1	1	1	0	1	1	0	0	-	-	-	-	-	-

**NPF:** 2.

**Dirección física resultante:**  $0000\ 0000\ 0001\ 0000\ 0000\ 0000)_2 = 0x1000$ .

## Ejercicio 4

- a) Lo primero que tenemos que hacer es calcular el ancho de banda. Para ello utilizamos la siguiente fórmula:

$$BW = \frac{\text{bytes}}{\text{ciclos}} \times \text{frecuencia}$$

Para las lecturas tenemos la secuencia de ciclos que podemos ver en la figura 2, mientras que para las escrituras la secuencia de ciclos se muestra en la figura 3.

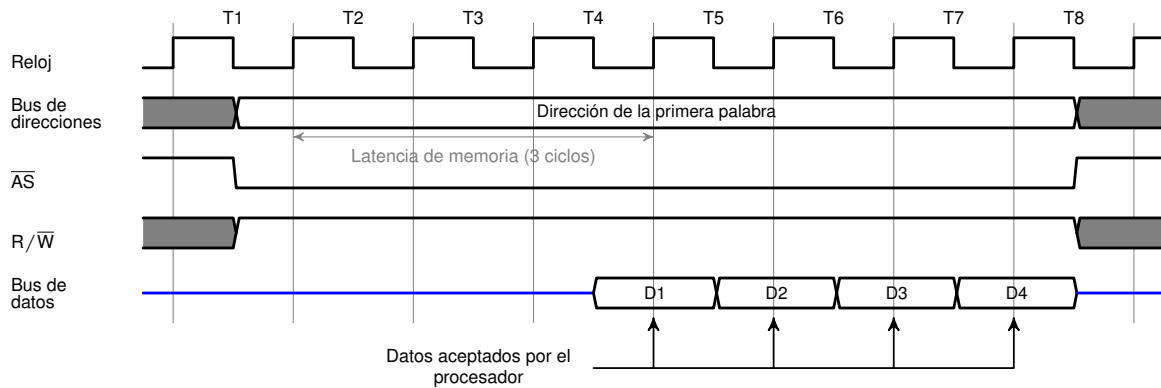


Figura 2: Lectura síncrona de 4 palabras en modo ráfaga.

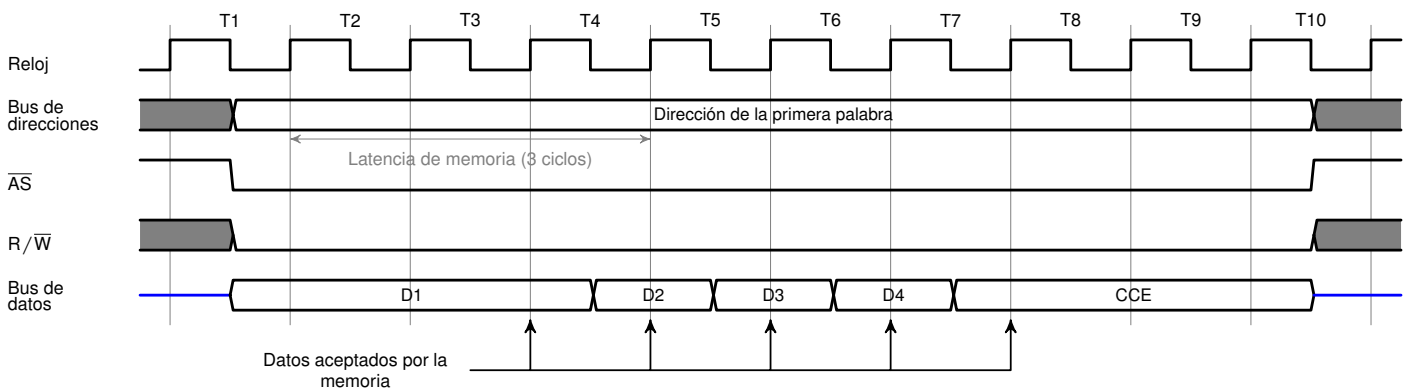


Figura 3: Escritura síncrona de 4 palabras en modo ráfaga con código de corrección de errores final.

$$BW = \frac{4 \text{ palabras} \cdot 4 \text{ bytes/palabra}}{(10 \text{ ciclos} \cdot 0'3 + 8 \text{ ciclos} \cdot 0'7)} \cdot 500 \cdot 10^6 = 930'23 \text{ MB/s} \approx 930 \text{ MB/s}$$

Una vez que hemos calculado el ancho de banda, podemos pasar a calcular el porcentaje de uso de la CPU. En primer lugar, tenemos que calcular el número de interrupciones que se producirán en un segundo:

$$\text{N}^\circ \text{ interrupciones} = \frac{930 \text{ MB/s}}{8 \text{ palabras} \cdot 4 \text{ bytes}} = \frac{930 \cdot 10^6 \text{ bytes/s}}{32 \text{ bytes}} = 29062500 \text{ interrupciones/s}$$

Ahora calculamos el total de ciclos que se consumirán en la realización de las interrupciones:

$$\text{Ciclos totales en interrupciones por segundo} = 29062500 \text{ int/s} \cdot 10 \text{ ciclos} = 290625000 \text{ ciclos/s}$$

Por último, calculamos el porcentaje de uso de la CPU:

$$\% \text{ CPU} = \frac{290625000}{500 \cdot 10^6} \cdot 100 = 58'13 \%$$

b) Para este apartado se seguirán pasos parecidos a los dados en el apartado a). En primer lugar vamos, a calcular el número de operaciones de DMA que se realizarán en un segundo:

$$\text{N}^\circ \text{ op. DMA} = \frac{930 \text{ MB/s}}{10 \text{ KB}} = \frac{930 \cdot 10^3 \text{ KB/s}}{10 \text{ KB}} = 93000 \text{ operaciones/s}$$

Ahora calculamos el total de ciclos que se consumirán en el uso del DMA:

Ciclos totales en DMA por segundo =  $93000 \text{ op. DMA/s} \cdot (200 + 100) \text{ ciclos} = 27900000 \text{ ciclos/s}$

Por último, calculamos el porcentaje de uso de la CPU:

$$\% \text{ CPU} = \frac{27900000}{500 \cdot 10^6} \cdot 100 = 5'58 \%$$

- c) El primer paso es averiguar qué número de ciclos podemos dedicar como máximo a gestionar la entrada/salida mediante sondeo. Si en un segundo el procesador es capaz de ejecutar  $500 \cdot 10^6$  ciclos, y para sondeo sólo podemos dedicar un máximo del 49 % del tiempo del procesador, entonces se podrán dedicar un total de:

$$500 \cdot 10^6 \cdot 49 \% = 245 \cdot 10^6 \text{ ciclos/segundo}$$

Ahora vamos a calcular el número de sondeos que se pueden realizar en un segundo:

$$\text{N}^\circ \text{ de sondeos} = \frac{245 \cdot 10^6 \text{ ciclos/segundo}}{20 \text{ ciclos}} = 12250000 \text{ sondeos/segundo}$$

Una vez que sabemos el número de sondeos, sólo nos queda calcular el número de palabras que hay que enviar en cada sondeo:

$$\text{N}^\circ \text{ de palabras} = \frac{\frac{930 \cdot 10^6 \text{ bytes/s}}{4 \text{ bytes/pal.}}}{12250000 \text{ sondeos/s}} = 18'98 \rightarrow 19 \text{ palabras/sondeo}$$

- d) En primer lugar vamos a ver lo que se tardaría en transmitir los 30GB a memoria sin tener en cuenta el tiempo usado en la programación del DMA:

$$\text{Tiempo} = \frac{30 \cdot 10^3 \text{ MB}}{930 \text{ MB/s}} = 32'26 \text{ segundos}$$

Ahora vamos a calcular el tiempo usado en la programación del DMA. Para ello vamos a calcular cuantas veces se usa el DMA en la transferencia del fichero:

$$\text{N}^\circ \text{ op. DMA} = \frac{30 \cdot 10^6 \text{ KB}}{10 \text{ KB}} = 3 \cdot 10^6 \text{ operaciones}$$

El siguiente paso es calcular el número de ciclos gastados en realizar todas estas operaciones:

$$\text{Ciclos totales en DMA} = 3 \cdot 10^6 \text{ op. DMA} \cdot (200 + 100) \text{ ciclos} = 900 \cdot 10^6 \text{ ciclos}$$

El último paso es calcular cuánto tiempo dedica el procesador en ejecutar todos esos ciclos:

$$\text{Tiempo} = \frac{900 \cdot 10^6 \text{ ciclos}}{500 \cdot 10^6 \text{ ciclos/s}} = 1'8 \text{ segundos}$$

El tiempo total para la transferencia del fichero será  $32'26 \text{ seg} + 1'8 \text{ seg} = 34'06 \text{ segundos}$ .