



# Estructura y Tecnología de Computadores

## Examen final

9 de septiembre de 2011

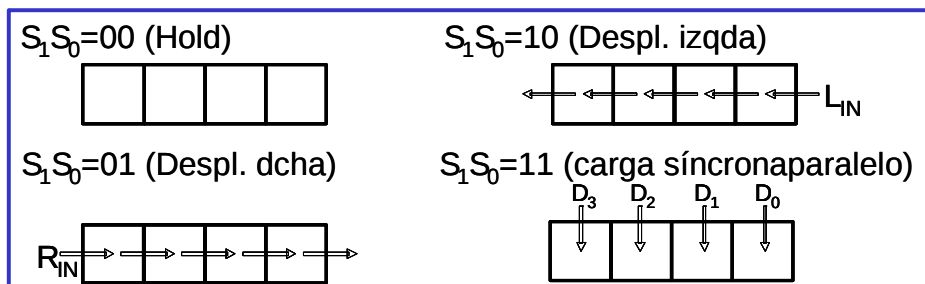
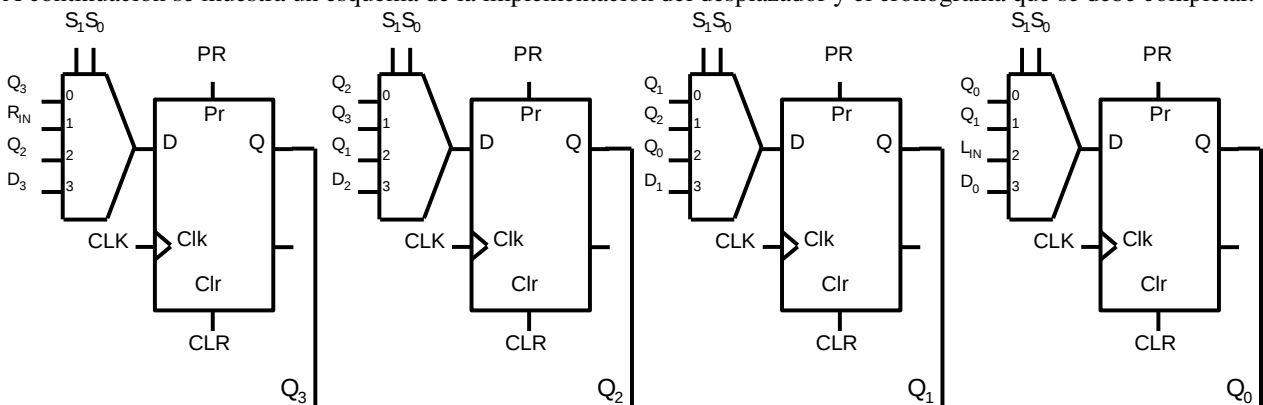
Grupo:	DNI:
Nombre:	Apellidos:

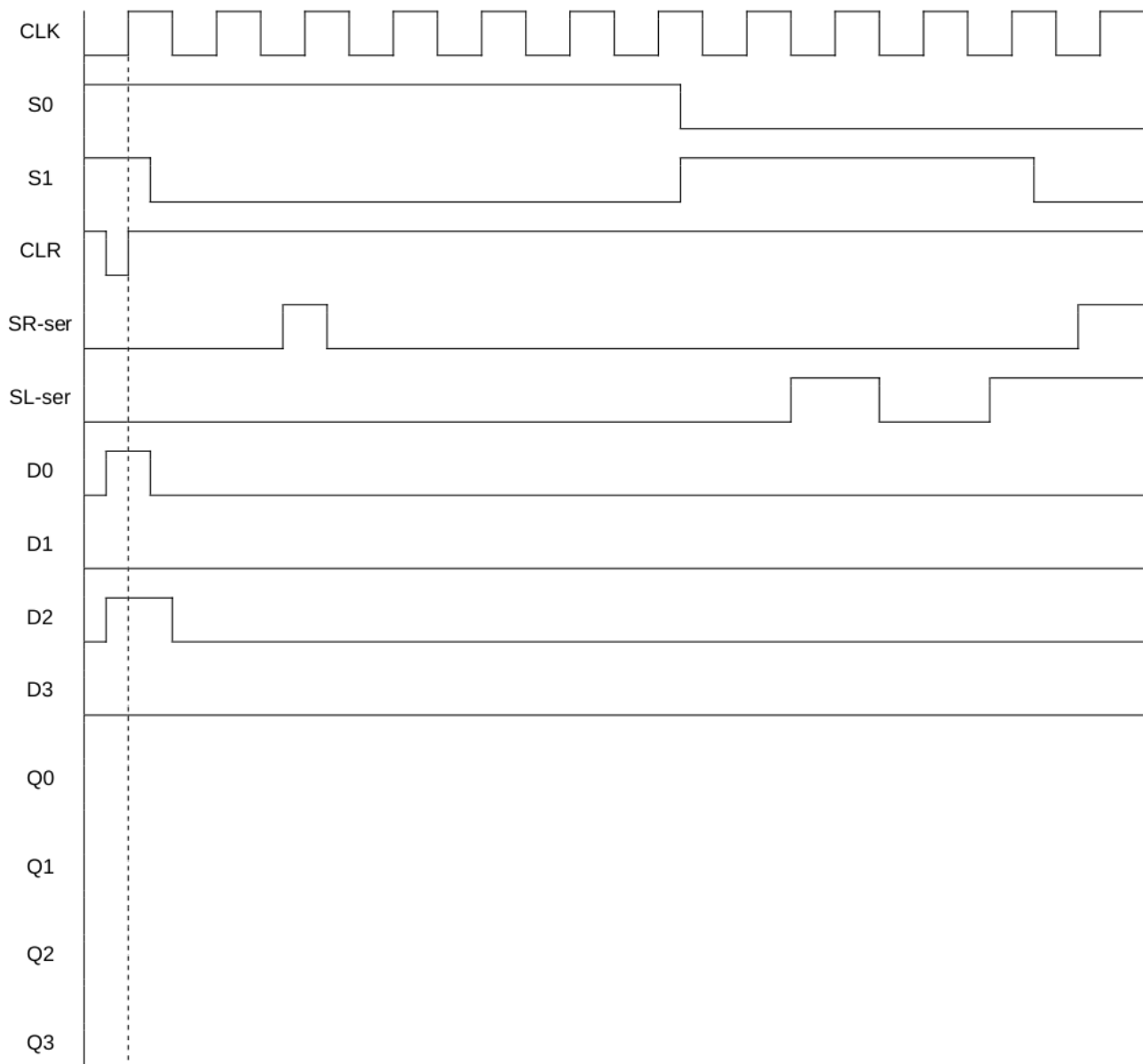
### Instrucciones:

- Cada pregunta se calificará con una nota entre 0 y 10. **Sólo en dos de las cuatro preguntas se podrá obtener una nota inferior a 3.** Si no se cumple esta condición, el examen estará suspenso.
- Las preguntas tienen pesos distintos en la nota final de teoría, la cuál se calculará como la media ponderada de las notas obtenidas. Estos pesos, para las preguntas 1, 2, 3 y 4, son, respectivamente, 30 %, 25 %, 30 % y 15 %.
- La nota obtenida en el examen 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.
- Responda a las preguntas en bolígrafo.
- **Sólo para los alumnos de ETC del plan antiguo:** estos alumnos tienen que resolver los problemas 1, 3 y 4, valiendo cada uno de ellos 2 puntos.

1. (10 puntos) Determinar el estado en cada pulso de reloj de un registro de desplazamiento universal bidireccional de 4 bits en el que se aplican las señales de entrada y de control mostradas. Suponer que solo CLR es asíncrona y que las señales de control  $S_0$ - $S_1$  codifican el siguiente comportamiento:  $S_0=S_1=1$ , carga paralela ( $D_0$ - $D_3$ );  $S_0=S_1=0$ , no hay cambios;  $S_0=0$  y  $S_1=1$ , desplazamiento a la izquierda; y  $S_0=1$  y  $S_1=0$ , desplazamiento a la derecha. SR-ser y SL-ser son las entradas de datos serie derecha e izquierda, respectivamente.

A continuación se muestra un esquema de la implementación del desplazador y el cronograma que se debe completar.





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.

Para que la solución del ejercicio se considere correcta, es importante que incluya claramente todos los pasos correspondientes a 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) Tenemos una máquina cuya jerarquía de memoria tiene las siguientes características:

- Tabla de páginas de 128 entradas.
- 14 bits de tamaño de direcciones físicas.
- 1 KB de tamaño de página.
- TLB completamente asociativo de 4 entradas, común para datos e instrucciones y con política de reemplazo aleatoria y política de postescritura para actualizar la tabla de páginas.
- Memoria caché de instrucciones de 1 KB, correspondencia directa y bloques de 4 palabras (palabras de 32 bits).
- Memoria caché de datos de 1 KB, asociativa por conjuntos de dos vías, y bloques de 4 palabras (palabras de 32 bits). La política de reemplazo es LRU y la actualización de la memoria se hace por postescritura.

Responda a las siguientes cuestiones suponiendo que inicialmente tanto las cachés como el TLB y la tabla de páginas están vacíos:

- a) (0.5 puntos) Especifique detalladamente el formato de la dirección física y de la dirección virtual.
- b) (0.5 puntos) ¿Cuál es el tamaño máximo de la memoria física?
- c) (0.5 puntos) ¿Cuál es el tamaño máximo de la memoria virtual (es decir, el espacio máximo direccionable por un proceso)?
- d) (1 punto) Dibuje un esquema de la tabla de páginas y calcule su tamaño, incluyendo los bits de control necesarios (justificando su inclusión).
- e) (1 punto) Dibuje un esquema detallado del TLB y calcule su tamaño incluyendo todos los bits de control necesarios (justificando su inclusión).
- f) (1 punto) Calcule el tamaño de cada caché incluyendo todos los bits de control necesarios (justificando su inclusión).
- g) (5 puntos) Haga un seguimiento del estado del TLB, la tabla de páginas y las cachés para los accesos mostrados en la tabla 1, rellenando las celdas vacías. Suponga que inicialmente las cachés, el TLB y la tabla de páginas están vacías. Las únicas páginas físicas libres (no asignadas a otro proceso) son las 5, 6, 7, 8 y 9.  
La columna “Tipo” indica si se trata de un acceso de lectura a datos (*L*), escritura a datos (*E*) o lectura de instrucción (*I*). En las tres columnas anteriores a la última debe indicar el valor de todos los bits de control de la estructura correspondiente después de que se haya realizado el acceso. Por ejemplo, si una estructura tiene los 5 bits de control *X*, *Y*, *Z*, *X'* e *Y'*, en la celda correspondiente debe aparecer algo parecido a “*X* = 0, *Y* = 1, *Z* = 0, *X'* = 1, *Y'* = 0”. En la columna “Etiqu. reemplazado” debe indicar la etiqueta del bloque reemplazado cuando se produzca algún reemplazo de caché.
- h) (0.5 puntos) ¿Cuáles son las tasas de acierto del TLB, la tabla de páginas y las dos cachés?
4. (10 puntos) Suponga un sistema de memoria formado por una caché con un tamaño de 64 KB y bloques de 8 palabras (1 palabra = 32 bits). El bus que conecta la caché y la memoria principal (MP) utiliza un reloj de 500 MHz, está multiplexado y tiene un ancho de 32 bits. La transmisión de una dirección requiere un ciclo de bus, mientras que el acceso a la MP tiene una latencia de 4 ciclos. El bus dispone de un modo ráfaga capaz de transmitir hasta un máximo de 4 palabras a la velocidad de una palabra cada dos ciclos.
- a) Indique los ciclos que requiere una operación de lectura y una operación de escritura de un bloque (8 palabras) en MP teniendo en cuenta que se realizan dos accesos consecutivos al bus para transferir un bloque completo. Además, en el caso de las escrituras también se envía por el bus un código de CRC (1 ciclo adicional), de forma que las escrituras se realizan como sigue: envía dirección, envía 4 palabras en modo ráfaga, espera la latencia y envía el CRC.
- b) Si la caché usa una política de post-escritura y hay un 40 % de bloques modificados, ¿cuántos ciclos de bus requiere una instrucción de carga que falla en caché? ¿Y una instrucción de almacenamiento que falla en la caché?
- c) Sabemos que el tráfico que circula por el bus proviene de un 70 % de instrucciones de carga y un 30 % de almacenamientos, ¿cuál será el máximo ancho de banda efectivo?
- d) Si usáramos una política de escritura directa, ¿cuál será el máximo ancho de banda efectivo para la misma distribución de cargas/almacenamientos que el apartado c)?
- e) Supongamos que la MP está paginada en páginas de 8 KB y que definimos una operación `flush p` que saca de la MP al proceso `p` pero que antes transfiere aquellas páginas físicas con el bit de modificado activo al disco duro. Si el proceso `p` ocupa un total de 20 páginas físicas estando el 75 % de ellas modificadas, ¿cuántos ciclos tardaremos en realizar un `flush p` si usamos un mecanismo por interrupciones que transfiere los datos en bloques de 16 palabras con una sobrecarga de 1000 ciclos por interrupción? ¿y por DMA si cada transferencia es de 1 página completa y requiere 500 ciclos para inicializar el DMA y 1500 ciclos para la interrupción de terminación?

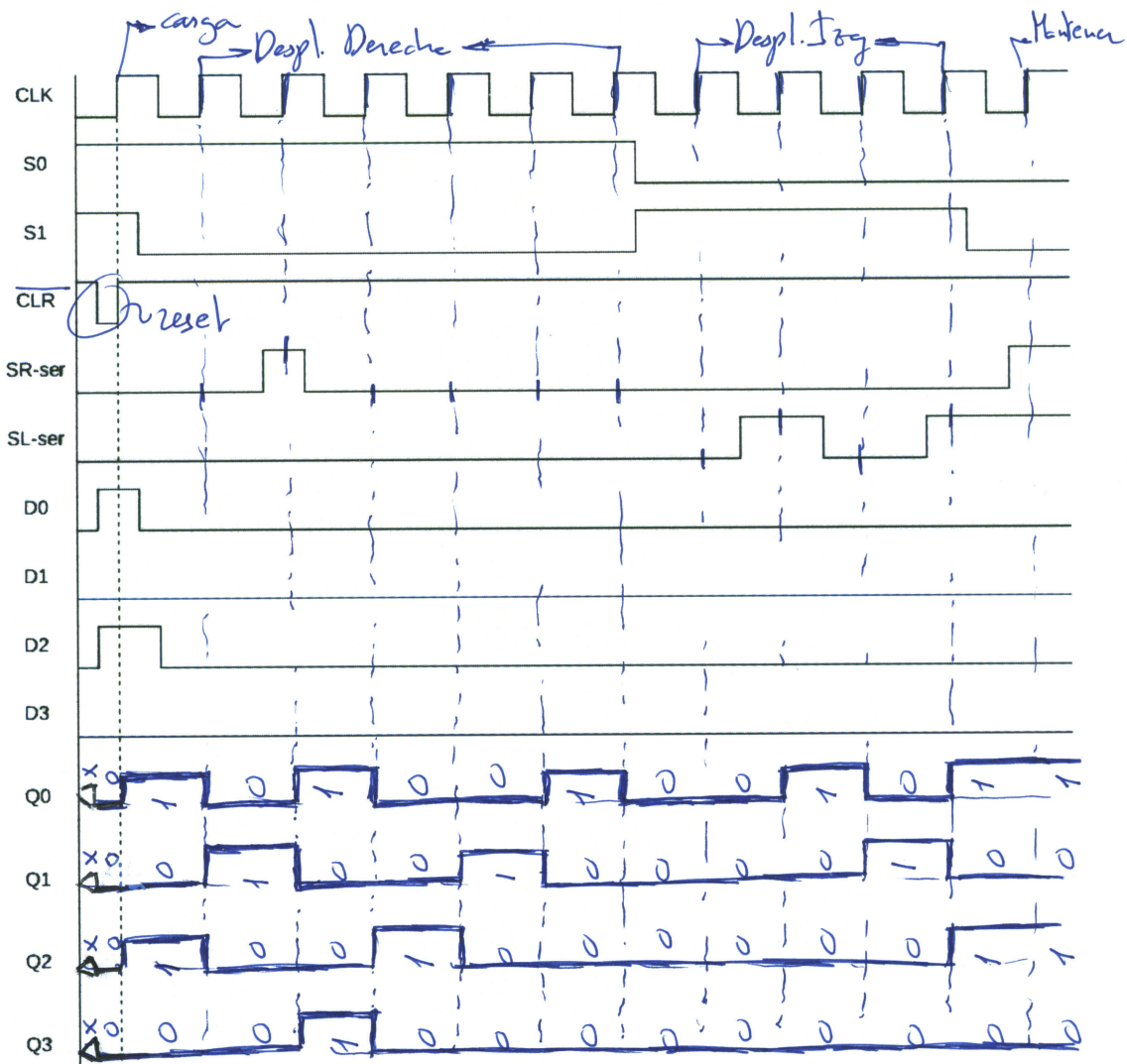
tipo	Dir. virtual	NPV	Desp.	¿Acierto en TLB?	¿Acierto en TP?	NPF	Dir. física	Conj. de caché	Etiq. de caché	¿Acierto en caché?	Control TLB	Control TP	Control caché	Etiq. reemplazado
I	256													
L	4176													
I	260													
I	264													
I	268													
E	9296													
I	272													
I	276													
I	256													
L	16464													
I	260													
I	264													
I	268													
E	4180													
I	272													
I	276													

Tabla 1: Tabla de accesos a memoria para el apartado g del ejercicio 2.

# Soluciones

## Ejercicio 1

Teniendo en cuenta el comportamiento de registro desplazador, las entradas y que la señal CLR es activa en baja, el cronograma resultante sería:



## Ejercicio 2

### ■ Análisis:

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

La semántica de "pushm (\$a)" es:

$$\$29 \leftarrow \$29 - 4$$

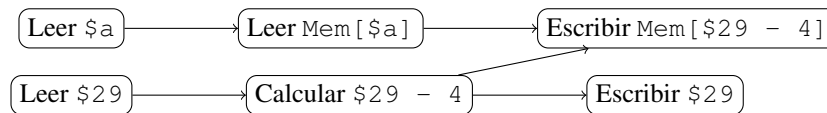
$$\text{Mem}[\$29] \leftarrow \text{Mem}[\$a]$$

(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].

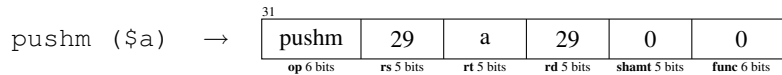
3. Establecimiento del orden de precedencia entre las distintas tareas a realizar:



■ Diseño:

1. **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 ← Reg[rs]	ALUOut ← A - 4	Memoria[ALUOut] ← MDR
PC ← PC + 4	B ← Reg[rt]	MDR ← Memoria[B]	Reg[rd] ← 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 tomará el valor 0 en los estados preexistentes del autómata de control.

4. Extensión del control:

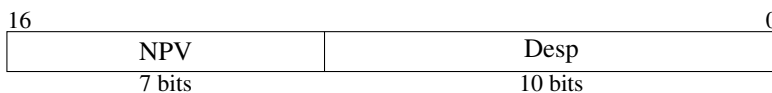
Habrá que añadir 2 nuevos estados al autómata de control original. El resultado se muestra en la figura 1.

## Ejercicio 3

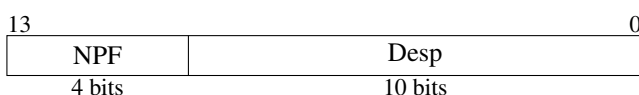
### Apartado a

El campo de desplazamiento de las direcciones físicas y virtuales necesitará  $\log_2 1024 = 10$  bits. Si la tabla de páginas tiene 128 entradas, el campo NPV de la dirección virtual necesitará  $\log_2 128 = 7$  bits. Y como las direcciones físicas tienen 14 bits, el campo NPF de las mismas tendrá  $14 - 10 = 4$  bits.

Por tanto, las direcciones virtuales se descompondrán de la siguiente manera:



Y las direcciones físicas:



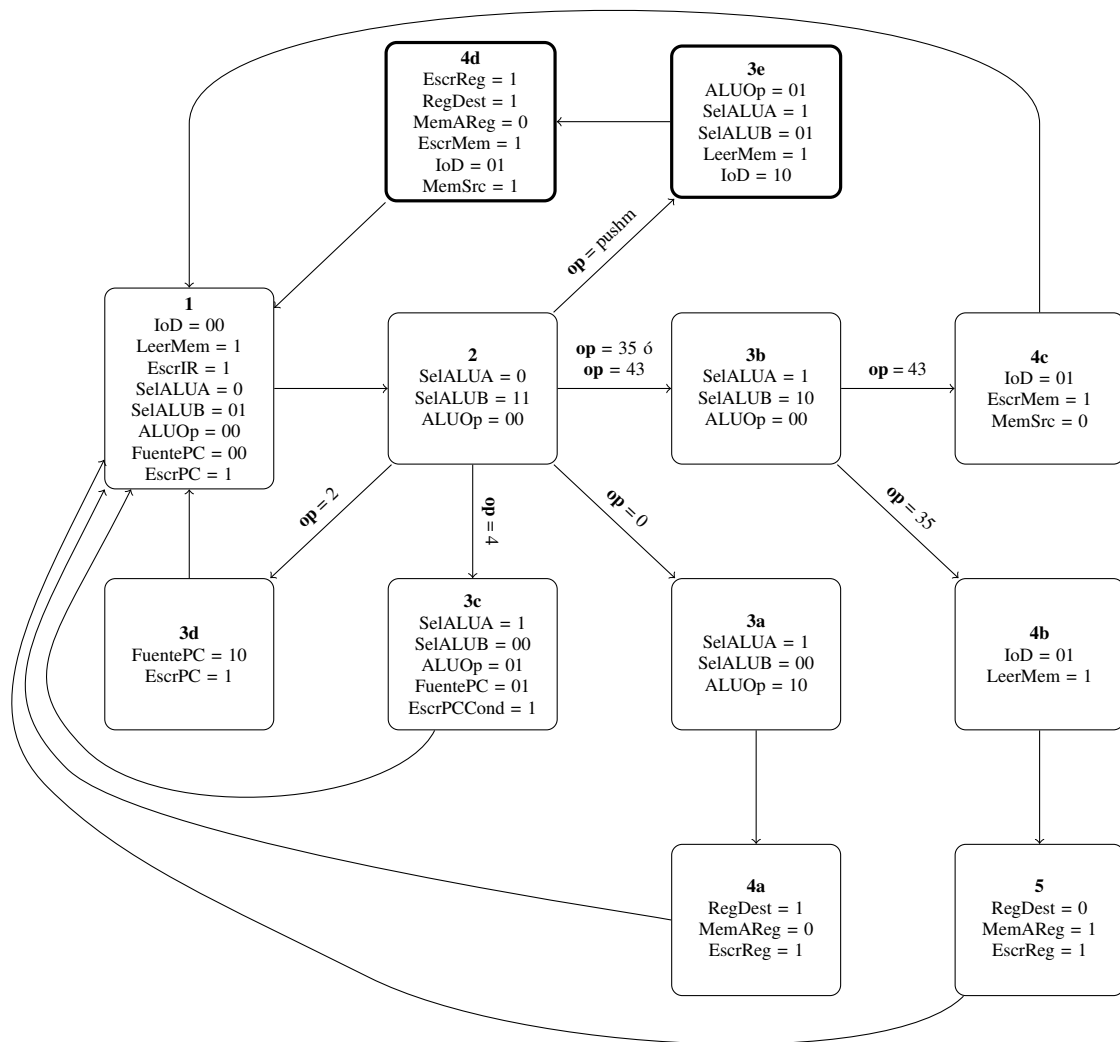


Figura 1: Modificaciones al autómata de control para incluir la instrucción `pushm`.

## Apartado b

El tamaño de la memoria física viene determinado por el número de bytes distintos que se pueden direccionar con las direcciones físicas. Como el tamaño de las direcciones físicas es de 14 bits, el tamaño máximo será de  $2^{14}$  bytes, o lo que es lo mismo, 16 KB.

## Apartado c

De forma análoga al apartado anterior, el tamaño de las direcciones virtuales determina el tamaño máximo de la memoria virtual (de cada espacio de direcciones). Por tanto, un proceso podrá direccionar como máximo  $2^{17}$  bytes, que son 128 KB.

## Apartado d

Cada entrada de la tabla de páginas necesitará un bit de modificación (M) para permitir la política de postescritura y un bit de uso (U) para permitir la política de reemplazo LRU, además del bit de validez (V). El NPF necesita 4 bits, por lo que cada entrada de la tabla de páginas tiene 7 bits en total. La tabla de páginas queda así:

	V	M	U	NPF (4 bits)
0	1	0	-	-
1	0	0	-	-
2	0	0	-	-
3	0	0	-	-
4	0	0	-	-
5	0	0	-	-
6	0	0	-	-
7	0	0	-	-
8	0	0	-	-
9	0	0	-	-
10	0	0	-	-
11	0	0	-	-
12	0	0	-	-
13	0	0	-	-
...	...	...	...	...
128	0	0	-	-

La tabla de páginas ocuparía  $128 \times 7$  bits si no tenemos en cuenta el alineamiento de las entradas. Si redondeamos el tamaño de cada entrada a un byte, ocuparía 128 bytes.

## Apartado e

Debido a que el TLB es totalmente asociativo, no se utilizará ningún bit del NPV para elegir el conjunto en el que almacenar la información de una entrada de la tabla de páginas (hay un solo conjunto). Los 7 bits del NPV formarán la etiqueta.

El único conjunto tiene 4 vías. Cada vía necesitará un bit de modificación (M) para permitir la política de postescritura, y un bit de validez (V) para saber cuándo la entrada contiene información útil.

Además de esos bits, cada vía contendrá también los bits M y U de la entrada correspondiente de la tabla de páginas (que mostraremos como M' y U', respectivamente) y el campo NPF con el correspondiente número de página física. El TLB queda, pues, de la siguiente forma:

Vía 0						Vía 1					
V	M	Etiqueta (7 bits)	M'	U'	NPF (4 bits)	V	M	Etiqueta (7 bits)	M'	U'	NPF (4 bits)
0	-	-	-	-	-	0	-	-	-	-	-

Vía 2						Vía 3					
V	M	Etiqueta (7 bits)	M'	U'	NPF (4 bits)	V	M	Etiqueta (7 bits)	M'	U'	NPF (4 bits)
0	-	-	-	-	-	0	-	-	-	-	-

Por tanto, el tamaño total del TLB será de  $4 \times (1 + 1 + 7 + 1 + 1 + 4) = 60$  bits.

## Apartado f

Cada bloque de caché almacena 16 bytes (4 palabras de 4 bytes), por lo que cada caché almacena un total de 64 bloques (1024 bytes/16 bytes/bloque).

En el caso de la caché de instrucciones, los 64 bloques se organizarán en 64 conjuntos. Cada bloque necesitará un bit de válido y una etiqueta de  $14 - \log_2 64 - \log_2 16 = 4$  bits. Por tanto, cada entrada ocupará  $16 \times 8 + 1 + 4 = 133$  bits, y la caché completa necesitará  $133 \times 64 = 8512$  bits.

Por otro lado, cada conjunto de la caché de datos tiene 2 vías, por lo que la caché tendrá 32 conjuntos. Cada bloque necesitará un bit de válido, un bit de modificado (para implementar la política de postescritura), un bit usado (para implementar la política LRU) y una etiqueta de  $14 - \log_2 32 - \log_2 16 = 5$  bits. Por tanto, cada entrada ocupará  $16 \times 8 + 1 + 1 + 1 + 5 = 136$  bits, y la caché completa necesitará  $136 \times 32 \times 2 = 8704$  bits.

### Apartado g

La solución se puede ver en la tabla 2.

### Apartado h

- Tasa de acierto de TLB:  $\frac{12}{16} = 75\%$ .
- Tasa de acierto de tabla de páginas: 0% (ó 75% si contamos los aciertos de TLB como aciertos de tabla de páginas, aunque no se accede realmente).
- Tasa de acierto de caché de instrucciones:  $\frac{10}{12} = 83,33\%$ .
- Tasa de acierto de caché de datos: 0%.

## Ejercicio 4

### Apartado a

Siguiendo las indicaciones del ejercicio, una lectura de 4 palabras en modo ráfaga produce el cronograma que se puede ver en la figura 2. Por lo tanto, una lectura de 4 palabras consecutivas requiere 13 ciclos del bus que conecta a la caché con la memoria principal, y un bloque de caché de 8 palabras necesita dos ráfagas, es decir, 26 ciclos en total.

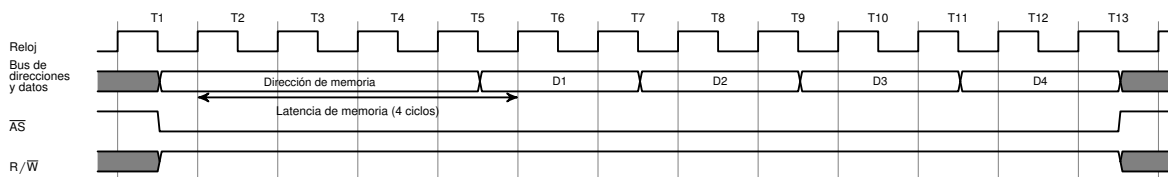


Figura 2: Lectura síncrona de 4 palabras en modo ráfaga.

En el caso de las escrituras, si atendemos estrictamente a lo que nos dice este apartado, se realizarían según el cronograma que podemos ver en la figura 3. En esta caso, hacen falta 15 ciclos cada 4 palabras o, lo que es lo mismo, 30 ciclos para escribir un bloque de caché de 8 palabras.

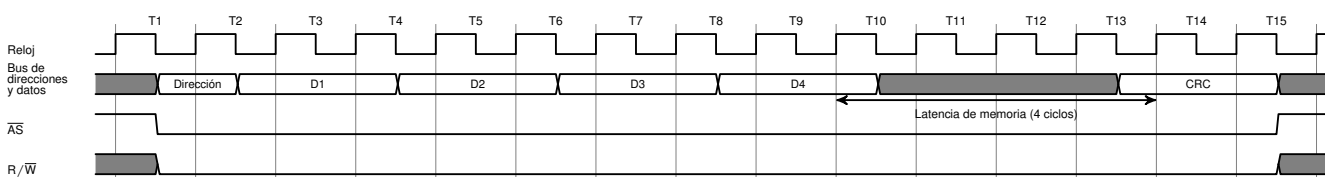


Figura 3: Escritura síncrona de 4 palabras en modo ráfaga.

### Apartado b

En el caso de un fallo de caché durante una lectura, será necesario leer un bloque de 8 palabras lo que requiere, como hemos visto, 26 ciclos. Además, el 40% de las veces que se produzca un fallo de caché en lectura será necesario escribir en memoria un bloque de caché de 8 palabras modificado, lo que supone 30 ciclos más. En promedio, una instrucción de carga que produce fallo de caché necesitará  $26 + 0,40 \cdot 30 = 38$  ciclos.

En el caso de una instrucción de almacenamiento, se necesitan los mismos ciclos, porque, al ser la caché de postescritura, un fallo de caché durante una escritura producirá la lectura de memoria principal del bloque correspondiente y, en el 40% de los casos, la escritura del bloque modificado que se expulsa.

Tipo	Dir. virtual	NPV	Desp.	¿Acuerdo en TLB?	¿Acuerdo en TP?	NPF	Dir. física	Conj. de caché	Etiqu. de caché	¿Acuerdo en caché?	Control TLB	Control TP	Control caché	Etiqu. reemplazado
I	256	0	256	no	no	5	5376	16	5	no	V=1, M=1, M'=0, U'=1	V=1, M=0, U=0	V=1	
L	4176	4	80	no	no	6	6224	5	12	no	V=1, M=1, M'=0, U'=1	V=1, M=0, U=0	V=1, M=0, U=1	
I	260	0	260	sí	no accede	5	5380	16	5	sí	V=1, M=1, M'=0, U'=1	V=1, M=0, U=0	V=1	
I	264	0	264	sí	no accede	5	5384	16	5	sí	V=1, M=1, M'=0, U'=1	V=1, M=0, U=0	V=1	
I	268	0	268	sí	no accede	5	5388	16	5	sí	V=1, M=1, M'=0, U'=1	V=1, M=0, U=0	V=1	
E	9296	9	80	no	no	7	7248	5	14	no	V=1, M=1, M'=1, U'=1	V=1, M=0, U=0	V=1, M=1, U=1	
I	272	0	272	sí	no accede	5	5392	17	5	no	V=1, M=1, M'=0, U'=1	V=1, M=0, U=0	V=1	
I	276	0	276	sí	no accede	5	5396	17	5	sí	V=1, M=1, M'=0, U'=1	V=1, M=0, U=0	V=1	
I	256	0	256	sí	no accede	5	5376	16	5	sí	V=1, M=1, M'=0, U'=1	V=1, M=0, U=0	V=1	
L	16464	16	80	sí	no accede	8	8272	5	16	no	V=1, M=1, M'=0, U'=1	V=1, M=0, U=0	V=1, M=0, U=1	12
I	260	0	260	sí	no accede	5	5380	16	5	sí	V=1, M=1, M'=0, U'=1	V=1, M=0, U=0	V=1	
I	264	0	264	sí	no accede	5	5384	16	5	sí	V=1, M=1, M'=0, U'=1	V=1, M=0, U=0	V=1	
I	268	0	268	sí	no accede	5	5388	16	5	sí	V=1, M=1, M'=0, U'=1	V=1, M=0, U=0	V=1	
E	4180	4	84	no	no	6	6228	5	12	no	V=1, M=1, M'=1, U'=1	V=1, M=0, U=0	V=1, M=1, U=1	14
I	272	0	272	sí	no accede	5	5392	17	5	sí	V=1, M=1, M'=0, U'=1	V=1, M=0, U=0	V=1	
I	276	0	276	sí	no accede	5	5396	17	5	sí	V=1, M=1, M'=0, U'=1	V=1, M=0, U=0	V=1	

Tabla 2: Tabla de accesos a memoria para el apartado g del ejercicio 2 rellena.

## Apartado c

Según el apartado anterior, las instrucciones de carga y de almacenamiento tienen el mismo coste en ciclos cuando se produce un fallo de caché. Este coste se deriva de la lectura de un bloque y, en un 40 % de los casos, de la escritura de otro bloque. Dicho de otra manera, por cada 100 bloques que se leen, se escriben 40. Luego el  $\frac{100}{140} = 0,714 \rightarrow 71,4\%$  de las transferencias son lecturas y el 28,6 % restante son escrituras.

Por lo tanto, el ancho de banda es:

$$AB = \frac{8 \text{ palabras} \cdot 4 \text{ bytes/palabra}}{(26 \text{ ciclos} \cdot 0,714 + 30 \text{ ciclos} \cdot 0,286)} \cdot 500 \cdot 10^6 = 589,45 \text{ MB/s} \approx 589 \text{ MB/s}$$

Se puede llegar al mismo resultado de otra manera y es teniendo en cuenta que cada fallo de caché supone, en promedio, la transferencia de  $8 \text{ palabras} \cdot 4 \text{ bytes/palabra} \cdot (1 + 0,4) = 44,8 \text{ bytes}$  (es decir, la transferencia siempre de un bloque entero de caché y de otro bloque el 40 % de las veces) en 38 ciclos (ver apartado b):

$$AB = \frac{44,8 \text{ bytes}}{38 \text{ ciclos}} \cdot 500 \cdot 10^6 \text{ ciclos/seg} \approx 589 \text{ MB/s}$$

## Apartado d

Con escritura directa, los fallos de caché durante las lecturas no producen escrituras porque cualquier modificación que se haya hecho en la caché también se habrá hecho en memoria principal, es decir, no quedará en la caché ningún bloque modificado que esté pendiente de ser escrito en memoria. Por lo tanto, cada fallo de caché sólo supone la lectura de un bloque de 8 palabras, lo que requiere 26 ciclos.

En el caso de las escrituras, tanto si se produce un acierto como un fallo de caché, siempre se escribe en memoria principal la palabra deseada. El número de ciclos necesarios en este caso, siguiendo las instrucciones para las escrituras del apartado a, es: 1 (envío de la dirección) + 2 (envío de la palabra a escribir) + 4 (latencia) + 1 (CRC) + 1 (finalización escritura) = 9 ciclos.

En ancho de banda en este caso es:

$$AB = \frac{(8 \text{ palabras} \cdot 0,7 + 1 \text{ palabra} \cdot 0,3) \cdot 4 \text{ bytes/palabra}}{(26 \text{ ciclos} \cdot 0,7 + 9 \text{ ciclos} \cdot 0,3)} \cdot 500 \cdot 10^6 = 564,59 \text{ MB/s} \approx 565 \text{ MB/s}$$

## Apartado e

De las 20 páginas que posee el proceso, hay que escribir en disco  $20 \cdot 0,75 = 15$ . Ya que cada página tiene un tamaño de 8 KB, esas 15 páginas suponen  $8 \cdot 15 = 120 \text{ KB}$ .

Utilizando interrupciones, puesto que en cada una se transfieren  $16 \cdot 4 = 64 \text{ bytes}$  y se requieren 1000 ciclos, en total, para escribir las 15 páginas a disco, se necesitarán:

$$\frac{120 \cdot 1024 \text{ bytes}}{64 \text{ bytes/int.}} \cdot 1000 \text{ ciclos/int.} = 1920000 \text{ ciclos}$$

Si en lugar de interrupciones utilizamos DMA, puesto que en cada transferencia de DMA se transfiere una página y se requieren  $500 + 1500 = 2000 \text{ ciclos}$ , en total, para escribir las 15 páginas, ahora se necesitarán:

$$15 \text{ trans. DMA} \cdot 2000 \text{ ciclos/trans} = 30000 \text{ ciclos}$$