

# ESTRUCTURA Y TECNOLOGÍA DE COMPUTADORES

## I.T.I. Gestión, I.T.I. Sistemas e I. Superior

### Examen de teoría – Parte II: cuestiones teórico-prácticas – 9 de septiembre de 2009

Apellidos:	Nombre:
D.N.I.:	
Grupo:    Sistemas I <input type="checkbox"/> Sistemas II <input type="checkbox"/> Gestión I <input type="checkbox"/> Gestión II <input type="checkbox"/> Superior <input type="checkbox"/>	

1. (2 puntos) Diseñe un circuito secuencial síncrono con biestables tipo D disparados por flanco ascendente y usando un autómata de Moore que tenga tres entradas ( $E_2$ ,  $E_1$  y  $E_0$ ) y una salida (S). La salida se activará si el valor de las tres entradas ha coincidido durante los tres últimos ciclos. Es decir, si  $E_i^t$  es el valor de la entrada  $E_i$  al comienzo del ciclo  $t$ , S valdrá 1 en el ciclo  $t$  si y sólo si  $E_0^t = E_1^t = E_2^t$ ,  $E_0^{t-1} = E_1^{t-1} = E_2^{t-1}$  y  $E_0^{t-2} = E_1^{t-2} = E_2^{t-2}$ .

- (0,5 puntos) Dibuje el autómata que modela el comportamiento del circuito, incluyendo para cada estado el valor de la salida.
- (0,5 puntos) Realice la asignación de estados que crea conveniente y obtenga las expresiones minimizadas de las funciones de transición y de salida.
- (0,5 puntos) Dibuje el circuito que implementa el autómata usando solo puertas NAND y biestables tipo D.
- (0,25 puntos) Calcule la frecuencia máxima a la que puede operar el circuito resultante. El retardo de las puertas NAND es de 15 ns y el de los biestables es de 50 ns.
- (0,25 puntos) ¿Se podría implementar la función de transición mediante un PLA? Si se puede: ¿cuáles serían las dimensiones de dicho PLA?. Si no se puede: ¿por qué?

2. (2 puntos) Resuelve las siguientes cuestiones:

- (0,8 puntos) Utilizando la metodología vista en clase para la inclusión de nuevas instrucciones en el esquema de implementación multiciclo, realice las fases de análisis y diseño de la instrucción, "**jal etiqueta**", que salta a la instrucción etiquetada y guarda en el registro \$ra (\$31) el valor de PC + 4

jal etiqueta: \$31 ← PC + 4; PC ← (PC + 4) + (4 \* valor inmediato)

La fase de diseño debe incluir: las modificaciones a realizar al camino de datos (si las hay), los nuevos estados a añadir al autómata y los valores de las señales de control.

- (0,2 puntos) Codificar la instrucción, indicando el valor de cada uno de los campos, si el salto a realizar es 6 instrucciones más adelante a partir de la instrucción jal (invéntese el valor del campo del código de instrucción).
- (0,8 puntos) Considérese, ahora, la instrucción "**jaldir salto**" que realizara un salto pseudodirecto, similar a la instrucción "j salto", la cual utiliza un campo de 26 bits para codificar la dirección del salto absoluto y guarda en el registro \$ra (\$31) el valor de PC + 4. Indicad las modificaciones en el camino de datos y en el autómata para que la instrucción jaldirect realice las operaciones:

jal dirección: \$31 ← PC + 4; PC ← 4 msb's de PC | 26 bits de dirección | 00

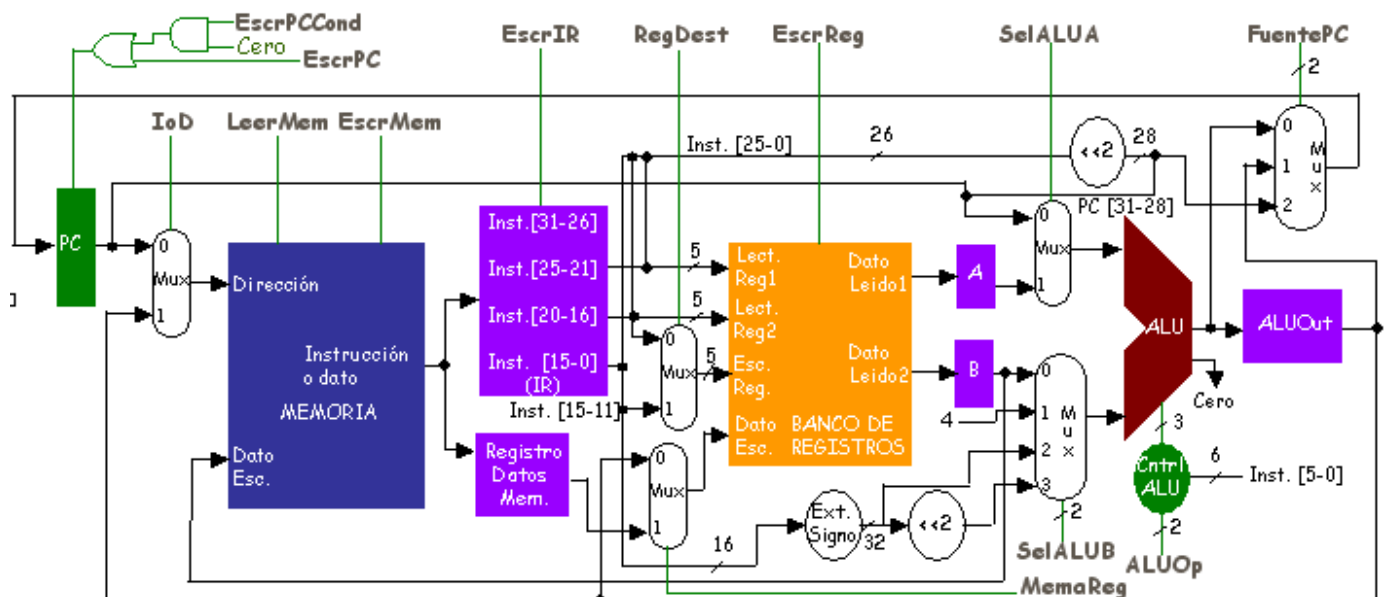
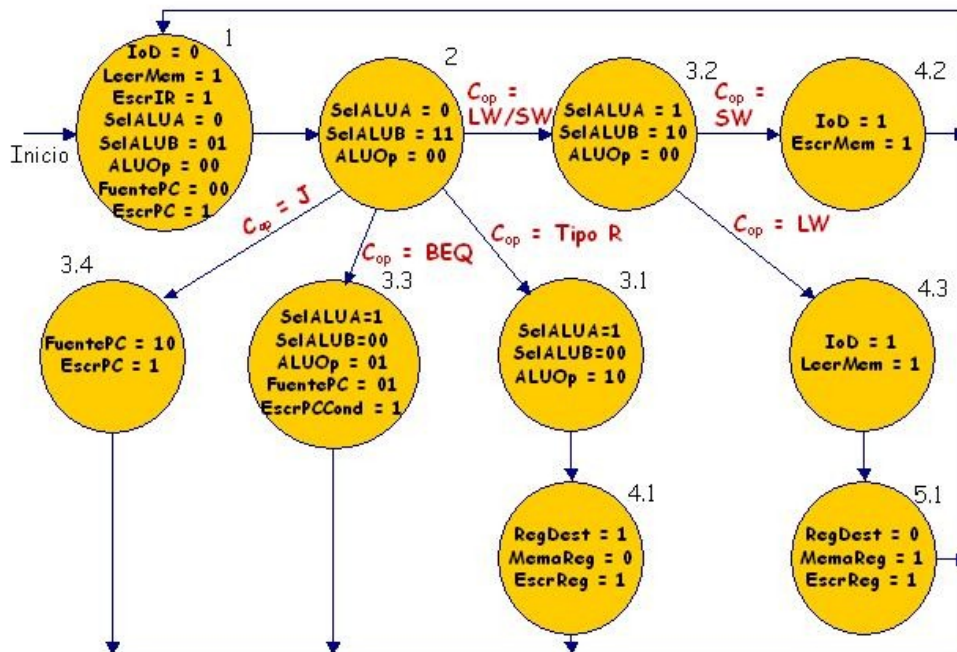
- (0,2 puntos) Codificar la instrucción, indicando el valor de cada uno de los campos, si el salto a realizar es a la dirección 0x12345678 cuando el valor actual del PC es 0x12345600 (invéntese el valor del campo del código de instrucción).

Las soluciones se pueden dar sobre las figuras que hay en el reverso, pero aun así habrá que justificar en papel las modificaciones realizadas. Obviamente, también será necesario entregar las hojas del examen con el nombre y los apellidos escritos en el lugar correspondiente).

3. (2 puntos) La máquina M posee un sistema de Memoria Virtual con las siguientes características:

- Tabla de páginas de 256 entradas.
- Memoria física de 256 bytes.
- Tamaño de página de 16 bytes.
- TLB asociativo con 4 conjuntos de 2 vías (2 bloques por conjunto). El TLB sigue una estrategia de reemplazo LRU y una política de escritura de postescritura.

- (0,5 puntos) Especificar detalladamente el formato de la dirección física y de la dirección virtual. Dibujar un esquema de la tabla de páginas y calcular su tamaño incluyendo los bits de control necesarios (justificando su inclusión). Se parte de una situación inicial en la que las páginas virtuales 1, 3, 5 y 7 se encuentran cargadas en memoria física en las páginas físicas 0, 1, 2 y 3, respectivamente.
- (0,3 puntos) Dibujar un esquema detallado del TLB y calcular su tamaño incluyendo todos los bits de control necesarios (justificando su inclusión). El TLB se encuentra inicialmente vacío.
- (0,6 puntos) Si durante la ejecución de un programa el procesador genera una lectura a la dirección virtual:  $(52)_{10}$ , y una escritura a la dirección  $(190)_{10}$ , indicar cuáles son sus correspondientes NPV (números de página virtual), si habrá acierto o fallo en la Tabla de Páginas, cuáles son sus NPF (números de página física) y la dirección física resultante. Suponer en este apartado que nuestra máquina no dispone de TLB. (Se sabe que la página física 4 se encuentra libre).
- (0,6 puntos) Repetir el apartado anterior, pero ahora teniendo en cuenta que nuestra máquina dispone de TLB.



## SOLUCIONES

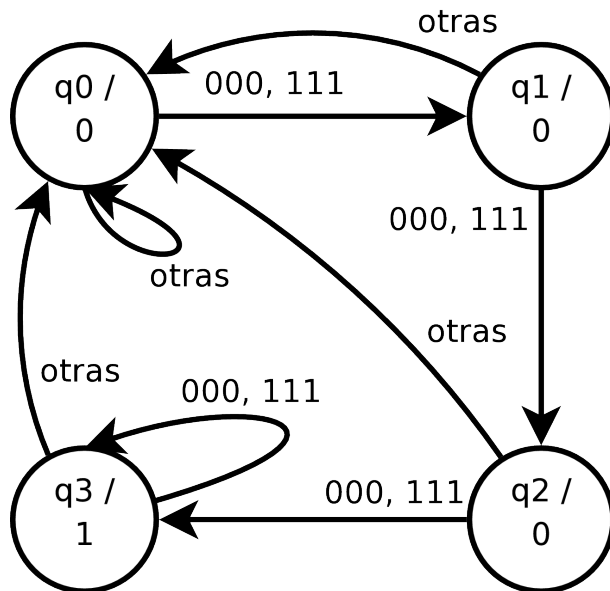
### Solución al ejercicio 1

#### Apartado a

Necesitaremos 4 estados para representar la siguiente información:

Estado	Significado
q0	No ha habido coincidencia en el ciclo anterior.
q1	Ha habido coincidencia en el ciclo anterior pero no en el previo.
q2	Ha habido coincidencia en los dos ciclos anteriores pero no en el previo.
q3	Ha habido coincidencia en los tres ciclos anteriores.

Con estos estados, el autómata para el circuito sería el siguiente:



Obsérvese que lo único importante de las entradas es si coinciden o no, por lo que podríamos implementar un autómata con una sola entrada si realizáramos una comparación antes de calcular la función de transición, de forma que la entrada del autómata fuera 0 para entradas no coincidentes y 1 para entradas coincidentes (o viceversa).

#### Apartado b

Para codificar los 4 estados, necesitaremos 2 flip-flops que llamaremos  $D_0$  y  $D_1$ . Como codificación de estados usaremos la siguiente:

Estado	$D_1$	$D_0$
q0	0	0
q1	0	1
q2	1	0
q3	1	1

Por tanto, las función de salida será:

$D_1$	$D_0$	$S$
0	0	0
0	1	0
1	0	0
1	1	1

Teniendo en cuenta el autómata anterior, la tabla de transición es:

$D_1$	$D_0$	$E_2$	$E_1$	$E_0$	$D_1^*$	$D_0^*$
0	0	0	0	0	0	1
0	0	1	1	1	0	1
0	1	0	0	0	1	0
0	1	1	1	1	1	0
1	0	0	0	0	1	1
1	0	1	1	1	1	1
1	1	0	0	0	1	1
1	1	1	1	1	1	1
<i>cualquier otro caso</i>					0	0

La expresión simplificada de la función de salida es la siguiente:

$$S = D_1 D_0$$

A continuación mostramos la simplificación de  $D_1^*$  y  $D_0^*$  usando mapas de Karnaugh.

$D_1^*$ :

$D_1 D_0 \setminus E_2 E_1 E_0$     000   001   011   010    100   101   111   110

00	0 0	4 0	12 0	8 0	16 0	20 0	28 0	24 0
01	1 1	5 0	13 0	9 0	17 0	21 0	29 1	25 0
11	3 1	7 0	15 0	8 0	19 0	23 0	31 1	27 0
10	2 1	6 0	14 0	9 0	18 0	22 0	30 1	26 0

$$D_1^* = D_1 \bar{E}_2 \bar{E}_1 \bar{E}_0 + D_0 \bar{E}_2 \bar{E}_1 \bar{E}_0 + D_1 E_2 E_1 E_0 + D_0 E_2 E_1 E_0$$

$D_0^*$ :

$D_1 D_0 \setminus E_2 E_1 E_0$	000	001	011	010	100	101	111	110
00	0 1	4 0	12 0	8 0	16 0	20 0	28 1	24 0
01	1 0	5 0	13 0	9 0	17 0	21 0	29 0	25 0
11	3 1	7 0	15 0	8 0	19 0	23 0	31 1	27 0
10	2 1	6 0	14 0	9 0	18 0	22 0	30 1	26 0

$$D_0^* = D_1 \bar{E}_2 \bar{E}_1 \bar{E}_0 + \bar{D}_0 \bar{E}_2 \bar{E}_1 \bar{E}_0 + D_1 E_2 E_1 E_0 + \bar{D}_0 E_2 E_1 E_0$$

### Apartado c

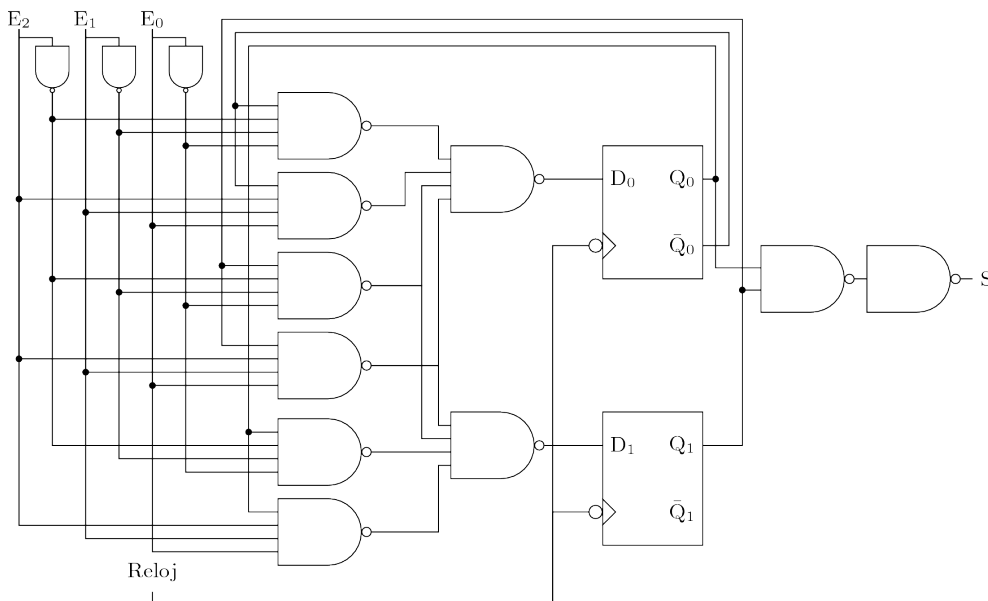
Transformamos las expresiones de las funciones de salida y transición usando las leyes de De Morgan:

$$S = D_1 D_0 = \overline{\bar{D}_1 \bar{D}_0}$$

$$\begin{aligned} D_1^* &= D_1 \bar{E}_2 \bar{E}_1 \bar{E}_0 + D_0 \bar{E}_2 \bar{E}_1 \bar{E}_0 + D_1 E_2 E_1 E_0 + D_0 E_2 E_1 E_0 = \\ &= \overline{\overline{D_1 \bar{E}_2 \bar{E}_1 \bar{E}_0 + D_0 \bar{E}_2 \bar{E}_1 \bar{E}_0 + D_1 E_2 E_1 E_0 + D_0 E_2 E_1 E_0}} = \\ &= \overline{D_1 \bar{E}_2 \bar{E}_1 \bar{E}_0 \cdot D_0 \bar{E}_2 \bar{E}_1 \bar{E}_0 \cdot D_1 E_2 E_1 E_0 \cdot D_0 E_2 E_1 E_0} \end{aligned}$$

$$\begin{aligned} D_0^* &= D_1 \bar{E}_2 \bar{E}_1 \bar{E}_0 + \bar{D}_0 \bar{E}_2 \bar{E}_1 \bar{E}_0 + D_1 E_2 E_1 E_0 + \bar{D}_0 E_2 E_1 E_0 = \\ &= \overline{\overline{D_1 \bar{E}_2 \bar{E}_1 \bar{E}_0 + \bar{D}_0 \bar{E}_2 \bar{E}_1 \bar{E}_0 + D_1 E_2 E_1 E_0 + \bar{D}_0 E_2 E_1 E_0}} = \\ &= \overline{D_1 \bar{E}_2 \bar{E}_1 \bar{E}_0 \cdot \bar{D}_0 \bar{E}_2 \bar{E}_1 \bar{E}_0 \cdot D_1 E_2 E_1 E_0 \cdot \bar{D}_0 E_2 E_1 E_0} \end{aligned}$$

Con lo que el circuito resultante sería:



Obsérvese que se aprovecha que dos de los minterminos de las funciones de transición son comunes para ahorrar dos puertas NAND.

### Apartado d

Dado que la función de transición tiene dos niveles de puertas NAND, el tiempo mínimo de ciclo será:

$$50ns + 2 \times 15ns = 80ns$$

Por tanto, la frecuencia máxima será:

$$\frac{1}{80 \text{ ns}} = 12.5 \text{ MHz}$$

### Apartado e

Sí, se podría implementar con un PLA de  $5 \times 6 \times 2$ .

### Solución al ejercicio 2

#### Instrucción multiciclo jal proc

Guarda en el registro \$31 el valor de PC + 4 y carga en el PC el valor de la dirección de salto, al igual que las instrucciones de salto condicional.

Análisis:

$$\text{jal etiqueta: } \$31 \leftarrow PC + 4; PC \leftarrow (PC + 4) + (4 * \text{valor inmediato})$$

En primer lugar, identificaremos el uso que se realiza de cada uno de los elementos básicos del camino de datos:

- **Banco de registros:** Escribir \$31
- **ALU:** Sumar  $(PC + 4) + (\text{ext\_sign}(\text{imm16}) \ll 2)$
- **Memoria:** Nada

Las dependencias entre las operaciones determinan el orden en el que se han de realizar:

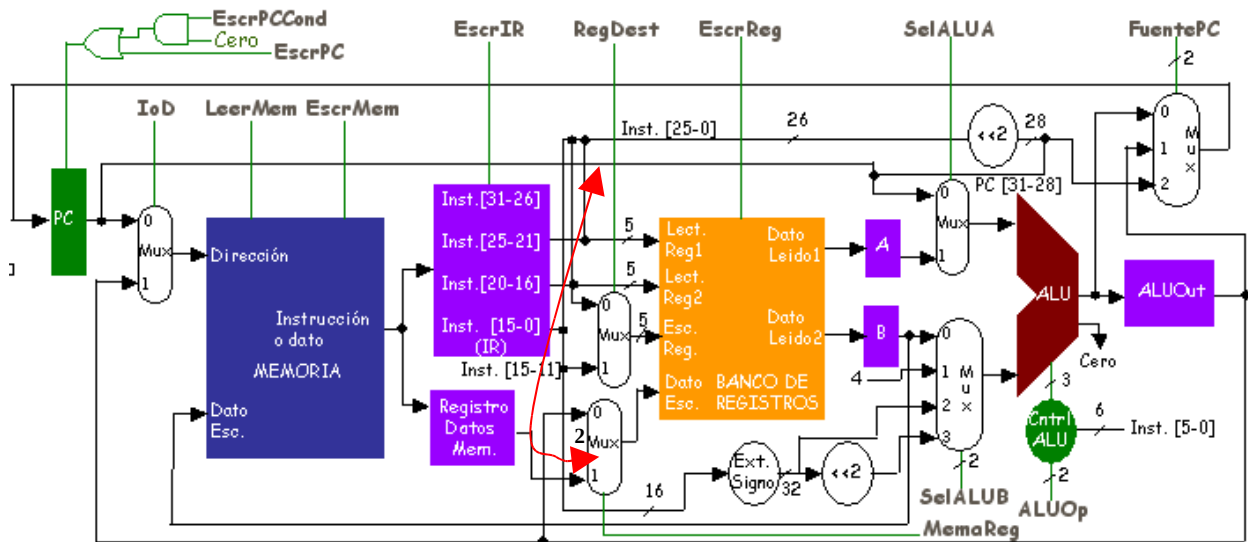
Ciclo 1	Ciclo 2	Ciclo 3
$PC \leftarrow (PC + 4)$	$\rightarrow$ $ALUOut \leftarrow PC + \text{Inmediato}$	$\rightarrow$ Escribir \$31

#### Diseño:

Formato (I):

op	rs	rt	inmediato
jal	xxxxx	\$31	Nº instrucciones desde PC+4 a salto

En gris aparecen aquellas operaciones que no podemos realizar con la ruta de datos actual. Será necesario modificar la ruta de datos para hacerlo posible. Es necesario hacer una modificación en el camino de datos que consiste en conectar el registro PC a una entrada nueva en el multiplexor MemAReg, para conseguir escribir directamente el valor del PC en el banco de registros.



Modificaciones en el autómata. La siguiente tabla muestra el valor de las líneas de control existentes en los diferentes estados, así como el valor de las nuevas líneas:

Ciclo 1 (No se añade nada nuevo)	Ciclo 2 (No se añade nada nuevo)	Ciclo 3 (nuevo)
Lectura instrucción e incremento PC (IR = Memoria[PC] y PC = PC + 4)	Cálculo de la dirección destino de salto (ALUOut = PC + (Signo Extendido ( IR[15-0] << 2))	Escribir \$31
IoD = 00 LeerMem EscrIR SelAluA = 0 SelAluB = 01 AluOp = 00 FuentePC = 00 EscrPC	SelALUA = 0 SelALUB = 11 ALUOp = 00 (prepara la dirección de salto)	<b>RegDest = 0</b> (Para seleccionar como registro a escribir el segundo operando de la instr. (\$31)). <b>MemAReg = 10</b> (Para seleccionar como dato para el BR el contenido del PC). <b>EscReg = 1</b> (Para realizar la escritura en el Banco de Registros) <b>EscrPC = 1</b> (da el salto) <b>FuentePC = 01</b> (salta a etiqueta, calculado previamente en ALUOut)

Existen otros planteamientos alternativos en el análisis y diseño, como p.ej.: Podemos no conectar el registro PC a la entrada del Mux que selecciona la entrada de datos al banco de registros, si leemos el registro \$Zero y se lo sumamos al PC de manera que seleccionaríamos ALUOut como entrada de datos al banco de registros. La ejecución tardaría 4 ciclos y aún así necesitaríamos conectar el campo IR[25-21] a un nuevo mux que seleccionaría el registro a leer por el canal B. En la codificación de la instrucción sería necesario codificar el\$0 en, por ejemplo, el campo rt.

### Codificación instrucción jal proc

op	rs	rt	inmediato
Jal proc	xxxxx	11111	000000000000101

### Instrucción multiciclo jaldir salto

Guarda en el registro \$31 el valor de PC + 4 y carga en el PC el valor de la dirección de salto, al igual que la instrucción de salto incondicional j dirección.

### Análisis:

jaldir salto: \$31  $\leftarrow$  PC + 4; PC  $\leftarrow$  4 msb's de PC | 26 bits de salto | 00

En primer lugar, identificaremos el uso que se realiza de cada uno de los elementos básicos del camino de datos:

- **Banco de registros:** Escribir \$31
- **ALU:** Nada
- **Memoria:** Nada

Las dependencias entre las operaciones determinan el orden en el que se han de realizar:

Ciclo 1	Ciclo 2	Ciclo 3
PC $\leftarrow$ (PC + 4)	No se hace nada	Escribir \$31

Hemos dejado el segundo ciclo si tocar (se realizan operaciones que no tienen transcendencia en la ejecución de la instrucción en concreto) pues el diseño de la instrucción j dir sigue estos mismos criterios.

### Diseño:

Formato (J):

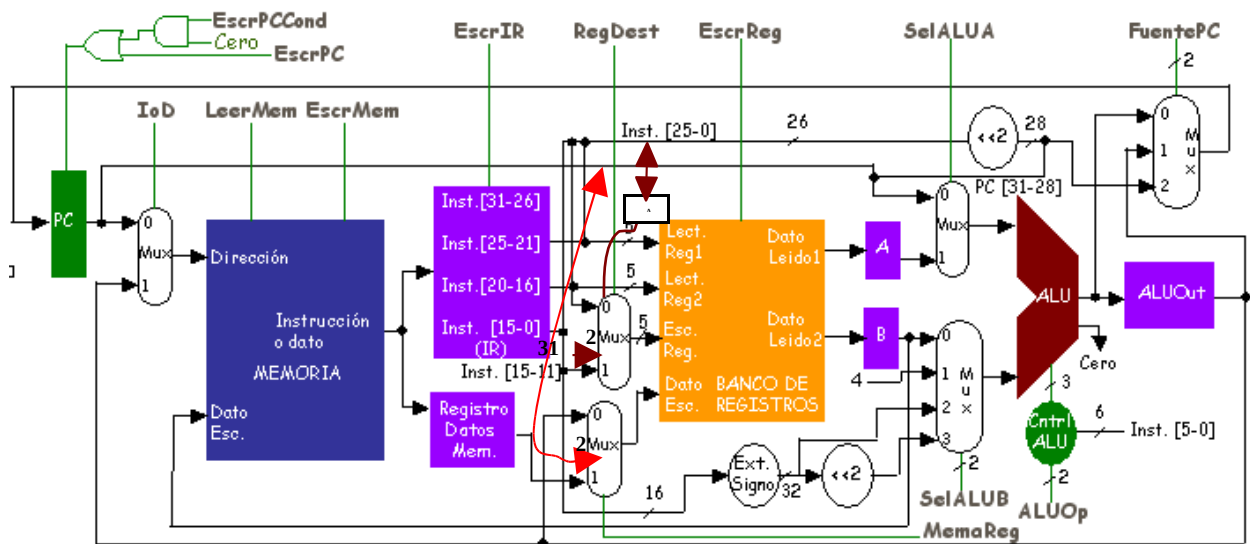
op	inmediato
jaldir	26 bits de salto

En gris aparecen aquellas operaciones que no podemos realizar con la ruta de datos actual. Es necesario hacer la misma modificación anterior en el camino de datos que consistía en conectar el registro PC a una entrada nueva en el multiplexor MemAReg, para conseguir escribir directamente el valor del PC en el banco de registros.

Es necesario aumentar en una entrada más el multiplexor RegDest, pues precisamos activar el registro \$31 para escritura cuando se ejecuta esta instrucción y dejar sin modificarlo para el resto de instrucciones. Este valor no lo podemos tomar de la instrucción, ya que ésta sólo tiene campos para el código de operación y el valor del salto directo. Una solución a este problema pasa por utilizar el valor del código de operación (exclusivo para esta instrucción) y con un sencillo módulo combinacional generar una señal que active la nueva entrada al multiplexor RegDest, que estará conectada al valor 31 (en binario son 5 unos).

Una puerta AND tiene conectada cada una de sus seis entradas al correspondiente bit del código de operación de IR, de manera que si el bit correspondiente del valor concreto con que codificamos la instrucción es un 1, lo conectamos directamente a la puerta AND, y si se trata de un cero, interponemos una puerta NOT. Así si el código de operación coincide con el de la instrucción, a la salida de la puerta AND tendremos un 1 y en caso contrario un cero. Esta señal puede estar conectada al bit más significativo de la entrada de control del multiplexor RegDest.

Otra solución alternativa es crear una nueva salida del sistema secuencial síncrono de la unidad de control que se activa en el estado correspondiente de esta instrucción,



conectada igualmente al msb del mux RegDest. Elegir una opción u otra pasa por hacer un análisis de coste de puertas lógicas añadidas o de complejidad en el diseño lógico.

Modificaciones en el autómata. La siguiente tabla muestra el valor de las líneas de control existentes en los diferentes estados, así como el valor de las nuevas líneas:

Ciclo 1 (No se añade nada nuevo)	Ciclo 2 (No se añade nada nuevo)	Ciclo 3 (nuevo)
Lectura instrucción e incremento PC (IR = Memoria[PC] y PC = PC + 4)	No se hace nada (son operaciones cuyas señales no se utilizarán)	Escribir \$31
IoD = 00 LeerMem EscrIR SelAluA = 0 SelAluB = 01 AluOp = 00 FuentePC = 00 EscrPC		<b>RegDest = 10</b> (Para seleccionar como registro a escribir el valor 31 (nueva entrada al Mux)). El 1 más significativo de RegDest viene de la AND nueva o de la unidad de control (ver antes). <b>MemAReg = 10</b> (Para seleccionar como dato para el BR el contenido del PC). <b>EscrReg = 1</b> (Para realizar la escritura en el Banco de Registros) <b>EscrPC = 1</b> (da el salto) <b>FuentePC = 10</b> (igual que una instrucción de salto pseudodirecto)

### Codificación instrucción Jaldir salto

op	inmediato
Jaldir	00 1000 1110 0010 0010 0110 1001 1110

### Solución al ejercicio 3

#### a) (0'5 puntos)

256 páginas virtuales =  $2^8 \rightarrow 8$  bits para el número de página virtual  
 Memoria física de 256 Bytes =  $2^8 \rightarrow 8$  bits para la dirección física  
 Tamaño de página de 16 bytes =  $2^4 \rightarrow 4$  bits para el desplazamiento

Con todo ello tenemos:

#### Dirección virtual: 12 bits

Número página virtual: 8 bits	Desplazamiento: 4 bits
-------------------------------	------------------------

#### Dirección física: 8 bits

Número página física: 4 bits	Desplazamiento: 4 bits
------------------------------	------------------------

#### Tabla de páginas:

Tamaño	1	1	1	4 bits	
	V	M	U	Número	Página Física
0	0	0	0	-	-
1	1	0	0	0	0
2	0	0	0	-	-
3	1	0	0	1	1
4	0	0	0	-	-
5	1	0	0	2	2
6	0	0	0	-	-
7	1	0	0	3	3
8	0	0	0	-	-
...	0	0	0	-	-
$2^8-2$	0	0	0	-	-
$2^8-1$	0	0	0	-	-

Tamaño =  $2^8 * (3 + 4)$  bits = **1792 Bits**

Tamaño (ajustando el tamaño a 1 byte por cada entrada) =  $2^8 * 1$  byte = **256 Bytes**

#### b) (0'3 puntos)

	V	U	M	Etí	M'	U'	NPF
0	1	1	1	6	1	1	4
1							
2							
3							

	V	U	M	Etí	M'	U'	NPF
0	1	1	1	6	1	1	4
1							
2							
3							

Tenemos 4 conjuntos. Por lo tanto, de los 8 bits que conforman el NPV, necesitamos para el índice 2 bits y nos quedan 6 bits para la etiqueta.

Tamaño =  $4 * 2 * (3 + 6 + 2 + 4)$  bits = **120 bits**

#### c) (0'6 puntos)

LECTURA  $52_{10} = 0000\ 0011\ 0100$  (recordar que la dirección virtual tiene 12 bits)

Descomponemos la dirección virtual en: NPV (los 8 bits superiores) + desplazamiento (los 4 bits inferiores)

NPV =  $0000\ 0011 = 3_{10}$

desplazamiento =  $0100 = 4_{10}$

Por lo tanto el NPV es **3**.

¿Hay acierto en la Tabla de Páginas?

	V	M	U	Número Página Física
0	0	0	0	-
1	1	0	0	0
2	0	0	0	-
→ 3	<b>1</b>	<b>0</b>	<b>0</b>	<b>1</b>
4	0	0	0	-
5	1	0	0	2
6	0	0	0	-
7	1	0	0	3
8	0	0	0	-
...	0	0	0	-
$2^8-2$	0	0	0	-
$2^8-1$	0	0	0	-

Como en la posición 3 de la TP el bit de validez está a 1 eso significa que la página virtual 3 sí está cargada en memoria física, con lo que se produce un **ACIERTO DE PÁGINA**.

Número de Página Física, **NPF = 1**

Ya se puede realizar la traducción de dirección virtual a dirección física utilizando el NPF+desplazamiento:

Dirección física: 8 bits	
NPF: 4 bits	Desplazamiento: 4 bits

**Dirección virtual: 0000 0011 0100 = 52)<sub>10</sub> → Dirección física: 0001 0100 = 20)<sub>10</sub>**

Hay que tener en cuenta que tenemos que actualizar el bit de uso (U) de la posición 3 de la tabla de páginas ya que acabamos de hacer uso de la página física que en esta se referencia. El bit de modificación (M) no hay que actualizarlo pues ha sido una lectura.

**ESCRITURA 190)<sub>10</sub> = 0000 1011 1110** (recordar que la dirección virtual tiene 12 bits)

Descomponemos la dirección virtual en: NPV (los 8 bits superiores) + desplazamiento (los 4 bits inferiores)

NPV = 0000 1011 = 11)<sub>10</sub>

desplazamiento = 1110 = 14)<sub>10</sub>

Por lo tanto el NPV es **11**.

¿Hay acierto en la Tabla de Páginas?

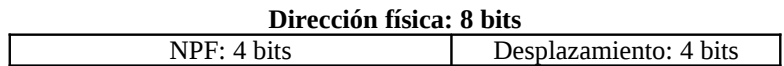
	V	M	U	Número Página Física
0	0	0	0	-
1	1	0	0	0
2	0	0	0	-
3	1	0	1	1
4	0	0	0	-
5	1	0	0	2
6	0	0	0	-
7	1	0	0	3
8	0	0	0	-
...	0	0	0	-
→ 11	<b>0</b>	<b>0</b>	<b>0</b>	-
...	0	0	0	-
$2^8-2$	0	0	0	-
$2^8-1$	0	0	0	-

Como en la posición 11 de la TP el bit de validez está a 0, significa que la página virtual 11 no está cargada en memoria física, con lo que se produce un **FALLO DE PÁGINA**, y por tanto el sistema operativo tendrá que acceder a memoria secundaria y traer la página virtual correspondiente a memoria principal. Y como sabemos que la página física 4 está libre, entonces la página virtual se almacenará en ésta, quedando la tabla de páginas de la siguiente forma:

	V	M	U	Número Página Física
0	0	0	0	-
1	1	0	0	0
2	0	0	0	-
3	1	0	1	1
4	0	0	0	-
5	1	0	0	2
6	0	0	0	-
7	1	0	0	3
8	0	0	0	-
...	0	0	0	-
⇒ 11	1	1	1	4
...	0	0	0	-
2 <sup>8</sup> -2	0	0	0	-
2 <sup>8</sup> -1	0	0	0	-

Número de Página Física, **NPF = 4**

Ya se puede realizar la traducción de dirección virtual a dirección física utilizando el NPF+desplazamiento:



**Dirección virtual: 0000 1011 1110 = 190)<sub>10</sub> → Dirección física: 0100 1110 = 78)<sub>10</sub>**

Notar que los bits de uso y modificación han sido actualizados, ya que se acaba de referenciar esa página, y además para hacer una escritura.

**d) (0'6 puntos)**

**LECTURA 52)<sub>10</sub> = 0000 0011 0100** (recordar que la dirección virtual tiene 12 bits)

Descomponemos la dirección virtual en: NPV (los 8 bits superiores) + desplazamiento (los 4 bits inferiores)

NPV = 0000 0011 = 3)<sub>10</sub>

desplazamiento = 0100 = 4)<sub>10</sub>

Por lo tanto el NPV es **3**.

¿Hay acierto en el TLB?

NPV = 0000 0011 = 3)<sub>10</sub> → Etiqueta → 0000 00 = 0)<sub>10</sub>

Posición en TLB → 11 = 3)<sub>10</sub>

	V	U	M	Etí	M'	U'	NPF
0	0						
1	0						
2	0						
3	0						

	V	U	M	Etí	M'	U'	NPF
0	0						
1	0						
2	0						
3	0						

Inicialmente como el TLB está vacío, concretamente el conjunto 3 tiene los dos bits de validez a 0, por lo que se produce un fallo y tenemos que acudir a la tabla de páginas.

¿Hay acierto en la Tabla de Páginas?

	V	M	U	Número Página Física
0	0	0	0	-
1	1	0	0	0
2	0	0	0	-
⇒ 3	1	0	0	1
4	0	0	0	-
5	1	0	0	2
6	0	0	0	-

7	1	0	0	3
8	0	0	0	-
...	0	0	0	-
$2^8-2$	0	0	0	-
$2^8-1$	0	0	0	-

Como en la posición 3 de la TP el bit de validez está a 1 eso significa que la página virtual 3 sí está cargada en memoria física, con lo que se produce un **ACIERTO DE PÁGINA**.  
Número de Página Física, **NPF = 1**

Ahora lo que hacemos es actualizar el TLB, que quedará de la siguiente manera:

	V	U	M	Etí	M'	U'	NPF
0	0						
1	0						
2	0						
3	1	1	1	0) <sub>10</sub>	0	1	1

	V	U	M	Etí	M'	U'	NPF
0	0						
1	0						
2	0						
3	0						

Ya se puede realizar la traducción de dirección virtual a dirección física utilizando el NPF+desplazamiento:

Dirección física: 8 bits	
NPF: 4 bits	Desplazamiento: 4 bits

**Dirección virtual: 0000 0011 0100 = 52)<sub>10</sub>** → **Dirección física: 0001 0100 = 20)<sub>10</sub>**

En este caso tenemos que actualizar el bit de uso (U) en el TLB pero no en la tabla de páginas, pues el TLB sigue una política de postescritura.

**ESCRITURA 190)<sub>10</sub> = 0000 1011 1110** (recordar que la dirección virtual tiene 12 bits)

Descomponemos la dirección virtual en: NPV (los 8 bits superiores) + desplazamiento (los 4 bits inferiores)

NPV = 0000 1011 = 11)<sub>10</sub>

desplazamiento = 1110 = 14)<sub>10</sub>

Por lo tanto el NPV es **11**.

¿Hay acierto en el TLB?

NPV = 0000 1011 = 11)<sub>10</sub> → Etiqueta → 0000 10 = 2)<sub>10</sub>  
Posición en TLB → 11 = 3)<sub>10</sub>

	V	U	M	Etí	M'	U'	NPF
0	0						
1	0						
2	0						
3	1	1	1	0) <sub>10</sub>	0	1	1

	V	U	M	Etí	M'	U'	NPF
0	0						
1	0						
2	0						
3	0						

Como podemos ver, en una de las entradas del conjunto 3 el bit de validez está puesto a 1, por lo que tenemos que proceder a comprobar si las etiquetas coinciden y por lo tanto se produce un acierto. Esto no es así, ya que la etiqueta en el TLB vale 0 y la etiqueta de la referencia es 2, por lo que tenemos que comprobar la otra entrada. En la otra entrada el bit de validez está puesto a 0, lo que significa que tenemos un fallo de TLB y tenemos que acceder a la tabla de páginas.

¿Hay acierto en la Tabla de Páginas?

	V	M	U	Número Página Física
0	0	0	0	-
1	1	0	0	0
2	0	0	0	-
3	1	0	0	1
4	0	0	0	-

5	1	0	0	2
6	0	0	0	-
7	1	0	0	3
8	0	0	0	-
...	0	0	0	-
⇒ 11	<b>0</b>	<b>0</b>	<b>0</b>	-
...	0	0	0	-
2 <sup>β</sup> -2	0	0	0	-
2 <sup>β</sup> -1	0	0	0	-

Como en la posición 11 de la TP el bit de validez está a 0, significa que la página virtual 11 no está cargada en memoria física, con lo que se produce un **FALLO DE PÁGINA**, y por tanto el sistema operativo tendrá que acceder a memoria secundaria y traer la página virtual a memoria principal. Como sabemos, la página física 4 está libre, por lo que almacenaremos la página virtual correspondiente en ésta, quedando la tabla de páginas de la siguiente forma:

	V	M	U	Número Página Física
0	0	0	0	-
1	1	0	0	0
2	0	0	0	-
3	1	0	0	1
4	0	0	0	-
5	1	0	0	2
6	0	0	0	-
7	1	0	0	3
8	0	0	0	-
...	0	0	0	-
⇒ 11	<b>1</b>	<b>0</b>	<b>0</b>	<b>4</b>
...	0	0	0	-
2 <sup>β</sup> -2	0	0	0	-
2 <sup>β</sup> -1	0	0	0	-

Número de Página Física, **NPF = 4**

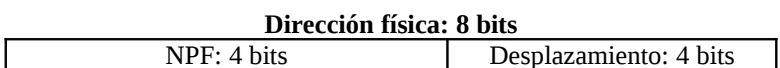
También se debe actualizar el TLB, que quedará de la siguiente manera:

	V	U	M	Eti	M'	U'	NPF
0	0						
1	0						
2	0						
3	1	<b>0</b>	1	0) <sub>10</sub>	0	1	1

	V	U	M	Eti	M'	U'	NPF
0	0						
1	0						
2	0						
3	<b>1</b>	<b>1</b>	<b>1</b>	2) <sub>10</sub>	<b>1</b>	<b>1</b>	<b>4</b>

Como vemos, el bit de U de la primera vía del conjunto 3 se ha puesto a 0 para poder aplicar correctamente una política de reemplazo LRU en caso necesario.

Ya se puede realizar la traducción de dirección virtual a dirección física utilizando el NPF+desplazamiento:



**Dirección virtual: 0000 1011 1110 = 190)<sub>10</sub> → Dirección física: 0100 1110 = 78)<sub>10</sub>**