Universitat Politècnica de Catalunya

Arquitectura de Computadores de Altas Prestaciones

Práctica 6

Procesador: Segmentación lineal con

cortocircuitos

Alumnos:

Manuel Velastegui

Carlos Andres Rodríguez Torres

**Grupo 5**

Cuatrimestre Primavera 2024-2025



**ÍNDICE**

[**Pregunta 1. 4**](#_heading=h.gjdgxs)

[1.1 Módulo LATPROH 4](#_heading=h.dfoq5pkudcp8)

[1.2 Justificación 4](#_heading=h.tmuhttjp2bax)

[**Pregunta 2. 4**](#_heading=h.30j0zll)

[2.1 Módulo LDL 5](#_heading=h.r249g56ai0e2)

[**Pregunta 3. 5**](#_heading=h.1fob9te)

[3.1 Módulo LDRD 5](#_heading=h.iyamv6i492kr)

[**Pregunta 4. 6**](#_heading=h.3znysh7)

[4.2. Módulo LATPROH 7](#_heading=h.tyjcwt)

[4.3. Módulo LDRD 7](#_heading=h.3dy6vkm)

[**Pregunta 5. 7**](#_heading=h.1t3h5sf)

[**Pregunta 6. 8**](#_heading=h.4d34og8)

[**Pregunta 7. 9**](#_heading=h.2s8eyo1)

[**Pregunta 8. 9**](#_heading=h.17dp8vu)

[**Pregunta 9. 10**](#_heading=h.3rdcrjn)

[9.1 Módulo LDRS 10](#_heading=h.6zzhs6f7wnkr)

[9.2 Justificación 10](#_heading=h.rxwc05ccs12m)

[9.3 Esquema RTL 10](#_heading=h.31xnf83h2vgf)

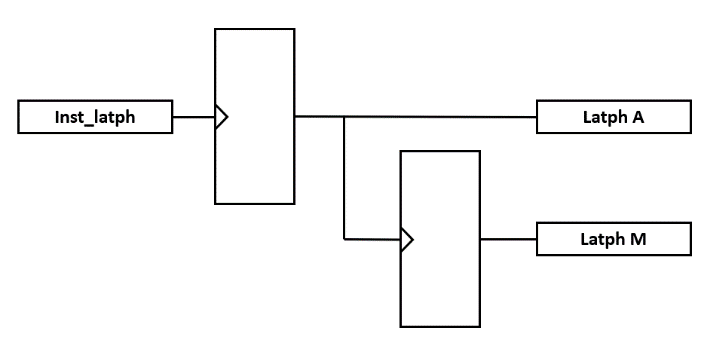
[**Pregunta 10. 11**](#_heading=h.26in1rg)

[10.1 Solución Propuesta 12](#_heading=h.clcyss2jywyl)

# **Pregunta 1.**

Diseñe el módulo LATPROH utilizando el menor número posible de registros y puertas lógicas, limitando el número de entradas de las puertas a 2. Justifique el diseño de forma sucinta y sistemática.

## **1.1 Módulo LATPROH**



*Figura 1. Módulo LATPROH*

## **1.2 Justificación**

El módulo LATPROH, se define como el encargado de la propagación de la información identificando el submódulo inst\_latpto a las diferentes etapas. Este genera señales para indicar (durante la interpretación del código) si hay alguna instrucción que puede producir un riesgo de datos debido a registros.

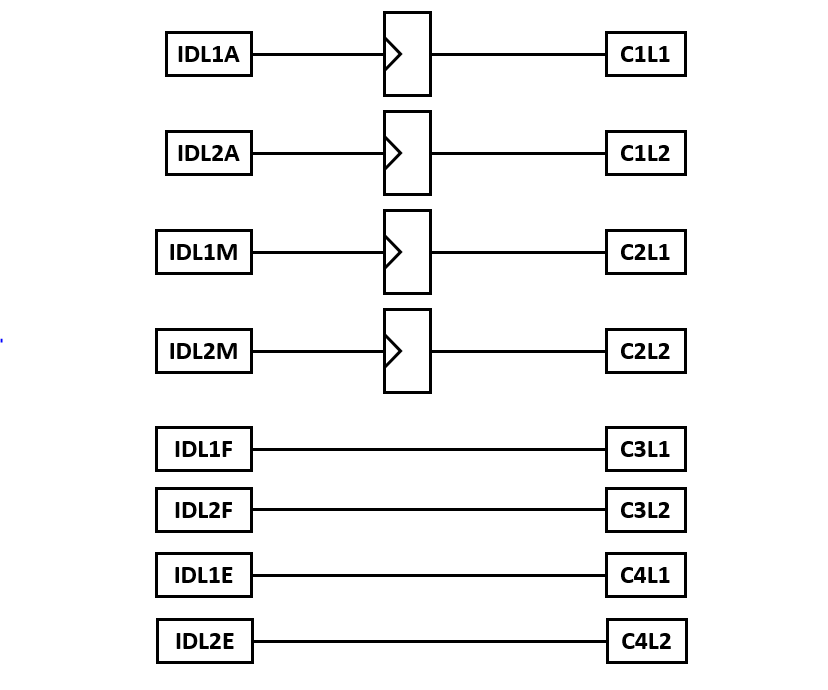
Para la etapa ALU encontramos la salida latphA. Mientras que para la etapa M encontramos la salida latphM. Se necesitan dos registros por la señal que indica posibles riesgos de datos se produce en la etapa DL y, por lo tanto, para latphA se introduce 1 registro ya que se espera 1 ciclo (DL) y para latphM se introducen 2 registros ya se espera 2 ciclos (DL y ALU).

# **Pregunta 2.**

Diseñe el módulo LDC utilizando el menor número posible de registros y puertas lógicas, limitando el número de entradas de las puertas a 2 (Lógica de cortocircuitos e interbloqueos en la página 375). Justifique el diseño de forma sucinta y sistemática.

El módulo LDC, es el encargado de habilitad los cortocircuitos en caso necesario. Las señales se generan en la etapa DL y, también, en la etapa ALU. Los registros se colocan para indicar el cortocircuito y de esta manera se retrasa 1 ciclo la salida de manera que estén en la etapa correcta en el siguiente ciclo.

## **2.1 Módulo LDL**

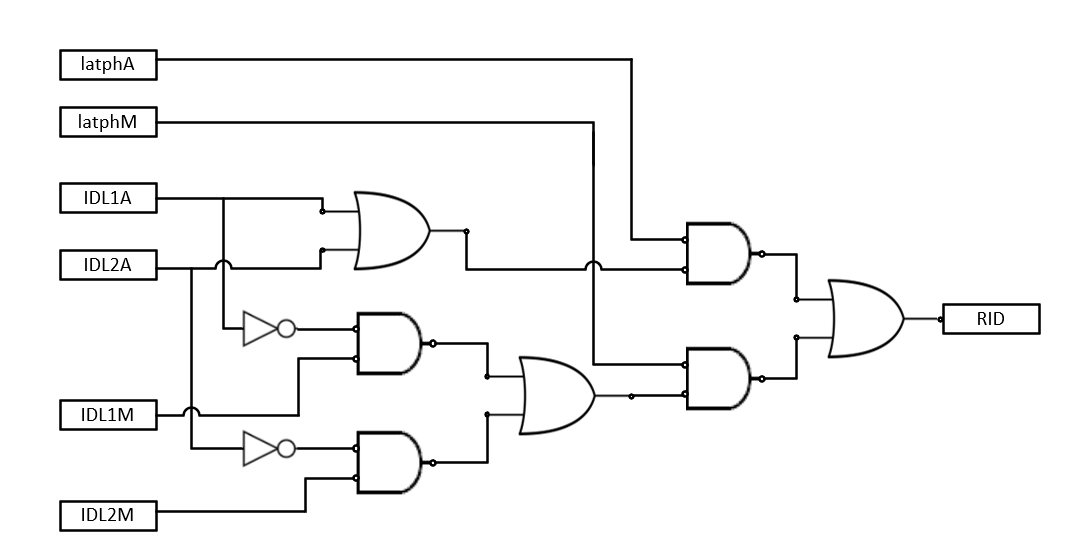


*Figura 2. Módulo LDL*

# **Pregunta 3.**

Diseñe el módulo LDRD utilizando el menor número posible de puertas lógicas, limitando el número de entradas de las puertas a 2. Justifique el diseño de forma sucinta y sistemática.

## **3.1 Módulo LDRD**



*Figura 3. Módulo LDRD*

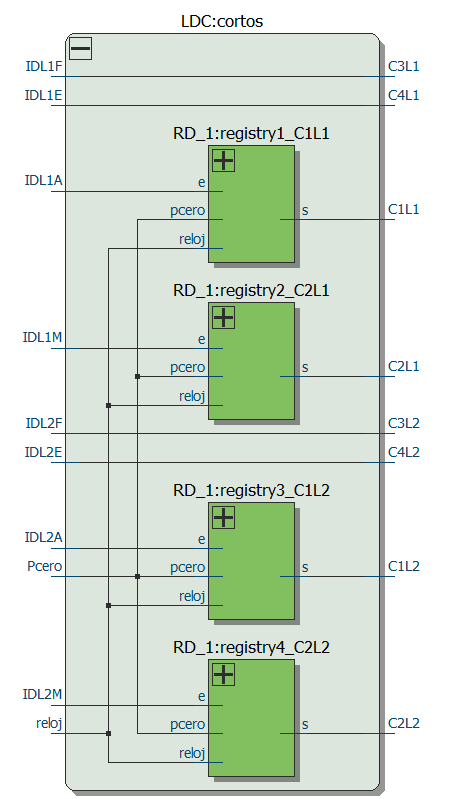
Como se especifica en el enunciado, el módulo LDRD es el encargado de la lógica de detección de riesgos de datos debido a registros. Por lo tanto, existirá riesgo en el caso de que el identificador del registro fuente coincida con los de la etapa M y no con los de la etapa ALU.

# **Pregunta 4.**

En el subdirectorio LIB (Organización de los ficheros: árbol de directorios en la página 401) se encuentran los ficheros asociados al diseño de la lógica de interbloqueos (proyecto quartus LIB\_C.qpf). Describa en VHDL los 3 módulos anteriores (LDC.vhd, latproh.vhd y LDRD.vhd), utilizando un modelo estructural. Entregue los esquemas RTL de los módulos elaborados por Quartus. Compruebe el diseño la Lógica de Cortocircuitos e Interbloqueos . El programa de prueba suministrado

(prueba\_LIB\_C.vhd) compara a cada ciclo las salidas de los 3 módulos diseñados con los respectivos modelos de referencia correctos1.

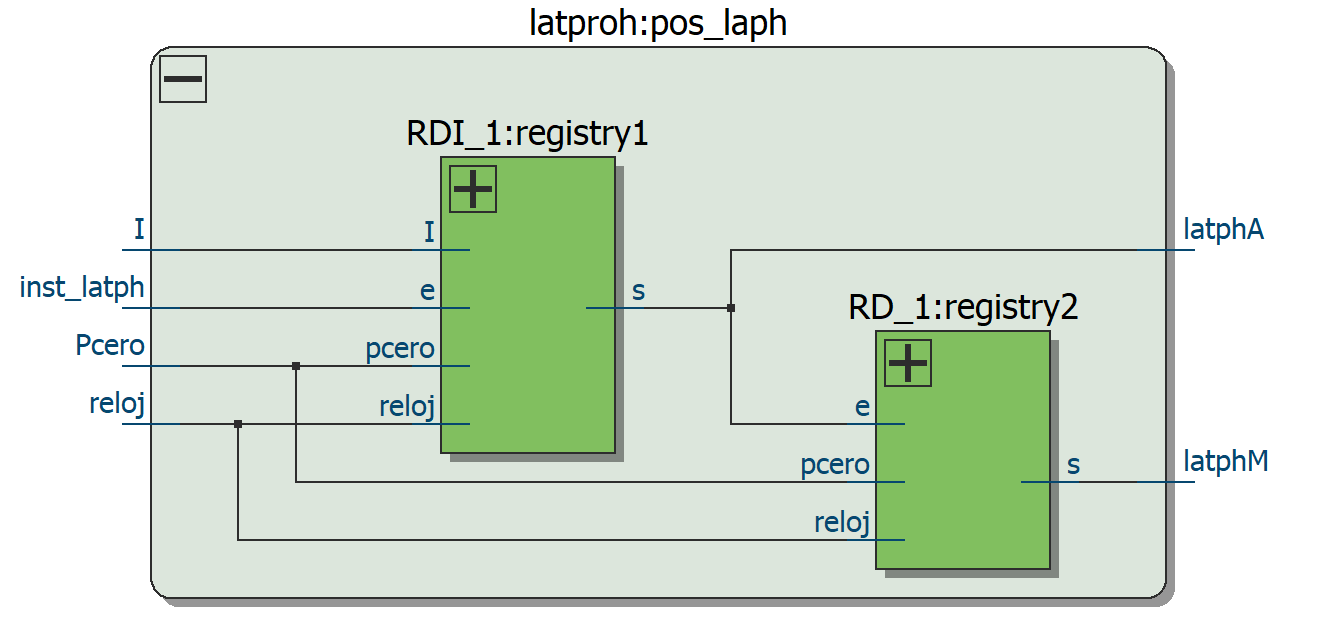
**4.1. Módulo LDC**



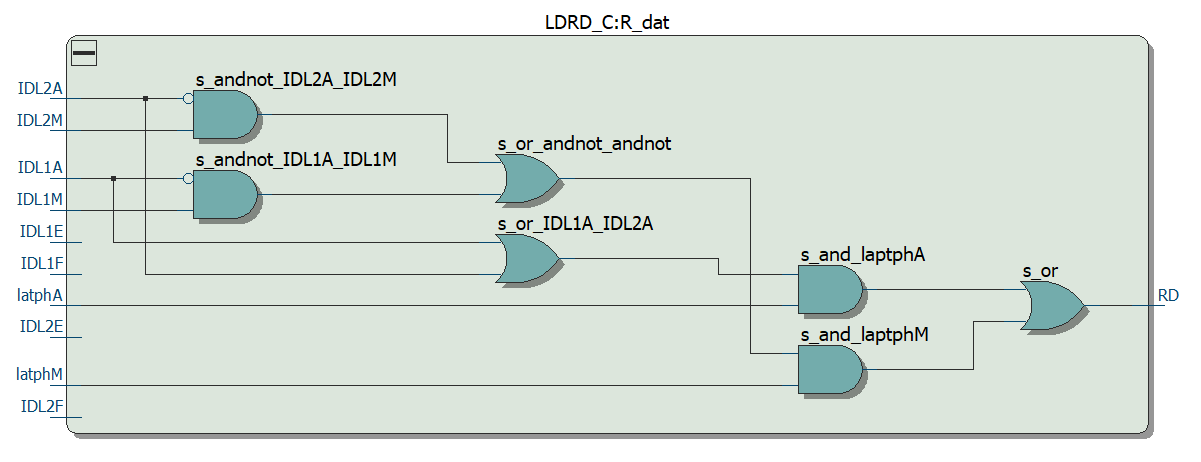
*Figura 3. Módulo LDC*

## 

## **4.2. Módulo LATPROH**

*Figura 5. Módulo LATPROH*  


## **4.3. Módulo LDRD**



*Figura 6. Módulo LDRD*

# 

# **Pregunta 5.**

# La siguiente tabla relaciona las entradas y las salidas de la lógica de interbloqueos LIBC durante 8 ciclos consecutivos. Suponga que en el ciclo 1 la etapa DL está ocupada por la instrucción “lw x3, 0(x1)” y que las etapas posteriores procesan datos inválidos. Deduzca una posible secuencia de las instrucciones que ocupan la etapa DL durante los siguientes 7 ciclos.

# 

|  |  |  | | | | | | | | Salidas LGR | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Salidas control Cortocircuitos | | | | | | | | CP | BDL | | DLA |
| ciclo | instrucción DL | C1L1 | C2L1 | C3L1 | C4L1 | C1L2 | C2L2 | C3L2 | C4L2 | bloq | bloq | inyec | inyec |
| 1  2  3  4  5  6  7  8 | lw x3, 0(x1) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Jarl x4, 0 (x3) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| Jarl x4, 0 (x3) | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| Jarl x4, 0 (x3) | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| NOP | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| NOP | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| add x2, x4, x3 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| add x4, x5, x4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |

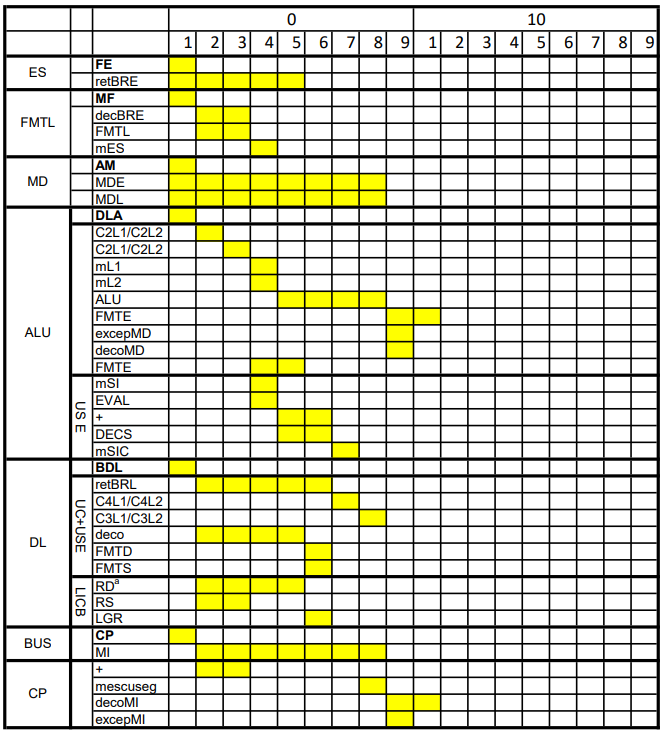
*Tabla 1. Tabla de entradas y salidas de la lógica de interbloqueos*

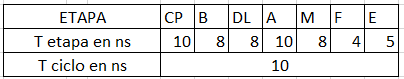
Para poder sacar las instrucciones, se han analizado los valores indicados en la instrucción anterior para poder obtener la siguiente instrucción. En primer lugar las 3 primeras instrucciones son de secuenciamiento incondicional, son 3 debido a que en las instrucciones anteriores el CP y DBL se bloquean, por ese motivo se arrastra la instrucción. En el ciclo 4, vemos que la salida BDL indica cómo se inyectan NOPs, por ese motivo los ciclos 5 y 6 son NOPs. Finalmente, teniendo en cuenta que registros coger para tener en cuenta los riesgos y cortocircuitos añadimos dos operaciones aritmético-lógicas.

# 

# **Pregunta 6.**

Para facilitar el cálculo del tiempo de ciclo utilizaremos un cronograma donde se representa el retardo de cada elemento del camino de datos. En el cronograma se han tenido en cuenta las etapas para agrupar los componentes del camino de datos.

*Tabla 2. Retardos* 



Para poder sacar las instrucciones especificadas en la tabla, se han analizado los valores que se indican en la instrucción anterior para poder obtener la siguiente instrucción.En primer lugar las 3 primeras instrucciones son de secuenciamiento incondicional, son 3 ya que como podemos ver se bloquea el CP y el BDL en las instrucciones anteriores y por ese motivo se arrastra. En el ciclo 4 podemos ver como en la salida BDL se indica cómo se inyectan NOPs, por eso motivo los ciclos 5 y 6 corresponden a NOPs, por último, y teniendo en cuenta que registros coger para tener en cuenta los riesgos y los cortocircuitos añadimos dos operaciones aritmético-lógicas.

# **Pregunta 7.**

Utilice el programa char\_sort (Simulación en la página 383). Añada un proceso al programa de prueba (ENSAMBLADO/PRUEBAS/prueba\_Rproc\_MD\_MI\_C.vhd) para obtener las métricas indicadas en la tabla.

|  | Procesador segmentado CON cortocircuitos | | Procesador segmentado SIN cortocircuitos | |
| --- | --- | --- | --- | --- |
| Ciclos perdidos por riesgos de datos | 380 | | 1940 | |
| Ciclos perdidos por riesgos de secuenciamiento | 664 | | 664 | |
| Ciclos de ejecución | 2307 | | 3939 | |
| Tiempo de ciclo (ns) | 10ns | | 8ns | |
| Ganancia | 36,60% | | |

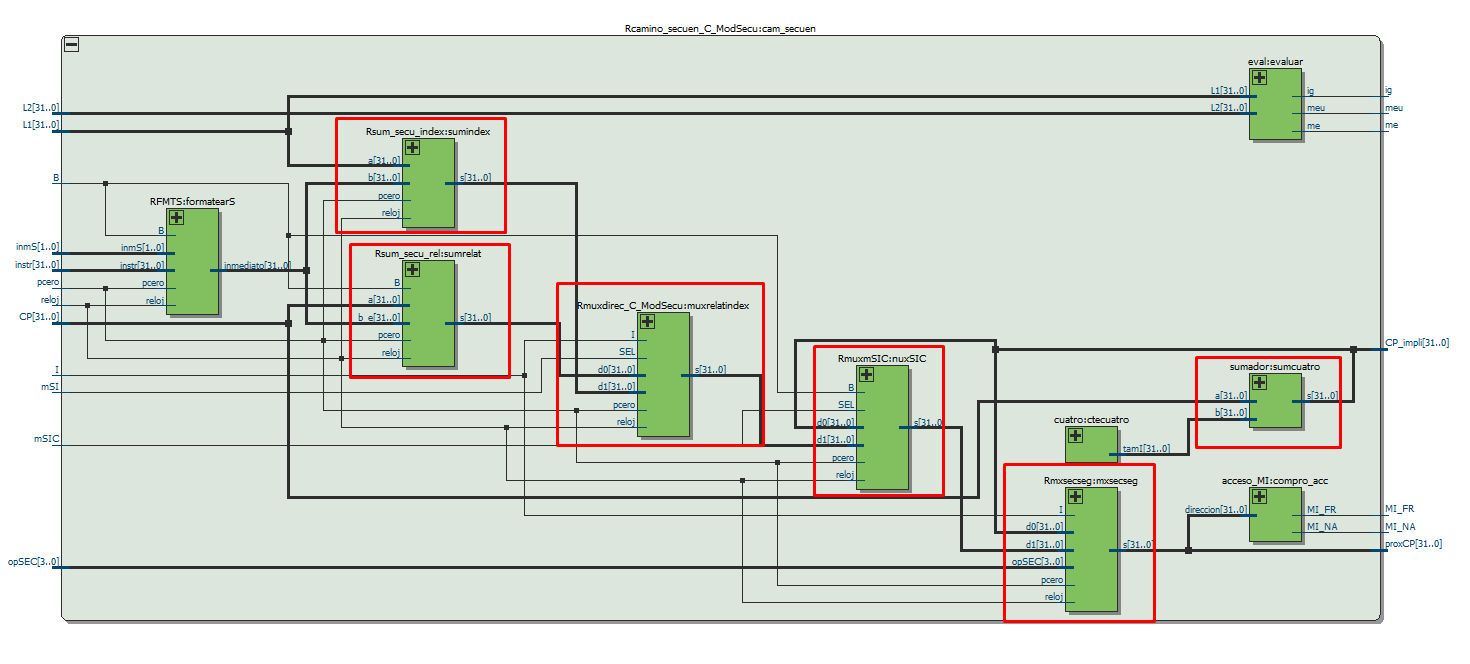
*Tabla 3. Métricas*

Para calcular todos los datos se han usado los códigos de las practicas 4 y 5, con ello sacaremos los datos del apartado.

# 

# **Pregunta 8.**

Dado el proyecto PROC\_SEG\_ModSecu (Apéndice 6.5, Apéndice 6.6 y Apéndice 6.7) entregue la elaboración efectuada por Quartus de la unidad de secuenciamiento. Identifique en ella los sumadores y multiplexores.

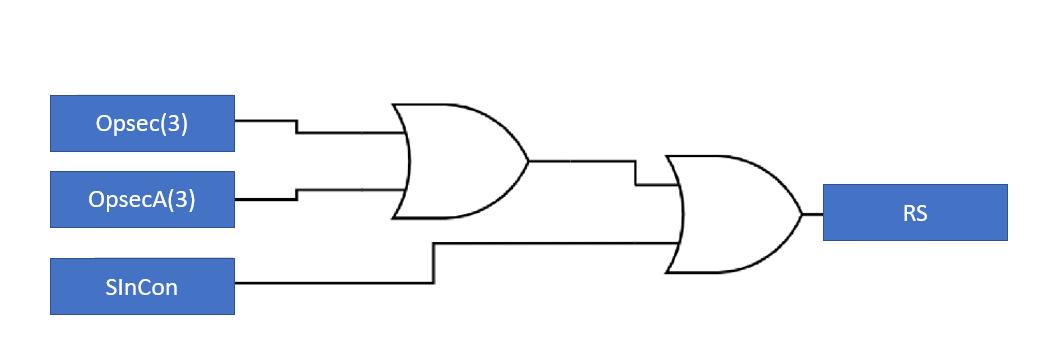
*Tabla. Unidad Secuenciamiento.* 

# En la tabla está marcado con recuadro los sumadores y mux involucrados en el cálculo de la nueva dirección. En función de la operación que se realice.

# **Pregunta 9.**

Considere la unidad de secuenciamiento con reducción de la latencia de la instrucción “jal” (página 432). Diseñe el módulo LDRS utilizando el menor número posible de puertas lógicas, limitando el número de entradas de las puertas a 2. Justifique el diseño de forma sucinta y sistemática.

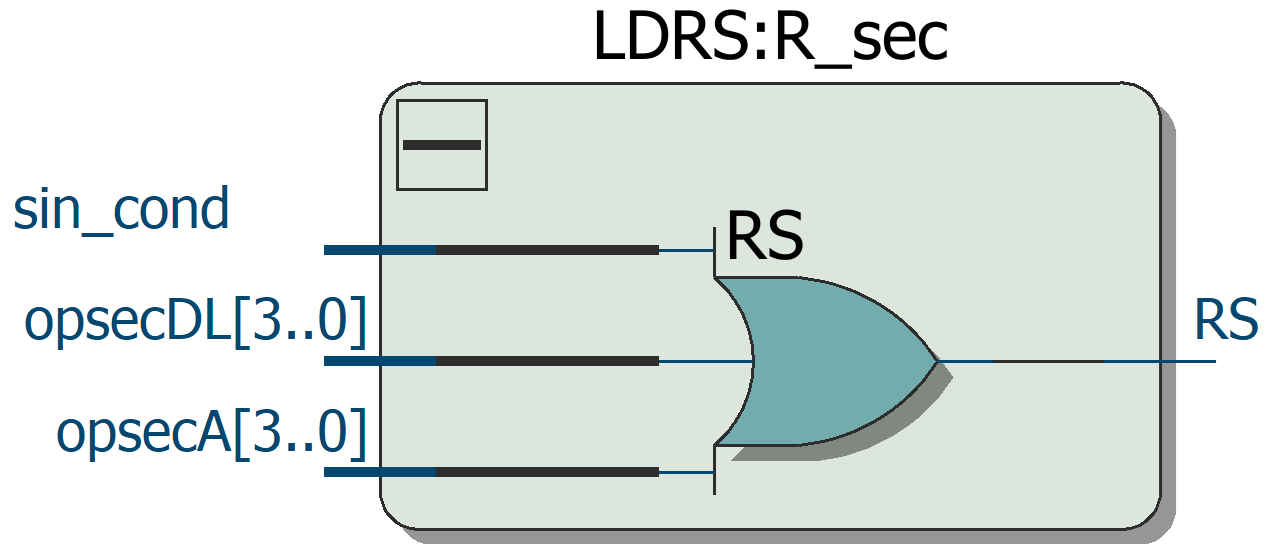
## **9.1 Módulo LDRS**

*Figura 7. Módulo LDRS*

## **9.2 Justificación**

El módulo se encarga de la lógica de detección de riesgos de secuenciamiento. Para poder detectar este riesgo usaremos las entradas Opsec(3), OpsecA(3) y SinCon, y en el caso de una de ellas sea 1, se podrá detectar riesgo de secuenciamiento. Es por eso que si hay una instrucción de secuenciamiento incondicional en DL o en ALU ya se detecta este riesgo.

## **9.3 Esquema RTL**



*Figura 7. Esquema RTL*

Indique, para el programa fact\_recurs, la reducción en ciclos perdidos por riesgos de secuenciamiento cuando se reduce la latencia al ejecutar una instrucción “jal” (Simulación: reducción de la latencia al ejecutar la instrucción “jal” en la página 421). Para ello, añada un proceso al programa de prueba (prueba\_Rproc\_MD\_MI\_C\_ModSecu\_relI.vhd) que cuente los ciclos perdidos debidos a riesgos de secuenciamiento.

|  | Ciclos perdidos por riesgos de secuenciamiento | | |
| --- | --- | --- | --- |
| Programa | Antes de reducir la latencia | Una vez reducida la latencia | Reducción (%) |
| fact\_recurs | 232 | 223 | 3,88% |

Para calcular los ciclos de riesgo de secuenciamiento se ha programado un proceso que ha ejecutado en el programa prueba con tal de contar todos los ciclos perdidos. Para calcular los ciclos perdidos antes de recudir la latencia se ha ejecutado el proyecto PROC\_SEC\_CORTOS, mientras que para conocer los ciclos perdidos una vez reducida la latencia se ha ejecutando en el proyecto PROC\_SEG\_ModSecurell. Para conocer la reducción se ha aplicado la siguiente fórmula:

# 

# **Pregunta 10.**

En el camino de datos segmentado (Práctica 5) se propaga la dirección de la instrucción y la dirección de la instrucción más cuatro. Esta última se utiliza como dirección de retorno en instrucciones de llamada a subrutina.

## **10.1 Solución Propuesta**

## 

*Figura 8. Solución propuesta*

Se eliminan el sumador, cableado de la etapa BUS, registro BDL, DLA y cables que transmite el CP+4. Se aprovecha el CP de la etapa A que está en el Registro DLA, y se añade un sumador (rectángulo parte inferior) para que cumpla la misma función. Finalmente, se conecta a donde iba conectado el cp+4 original.

Nota: Se ha eliminado el hardware que ya no es necesario. En este caso el CP+4 de la etapa BUS y el cableado.