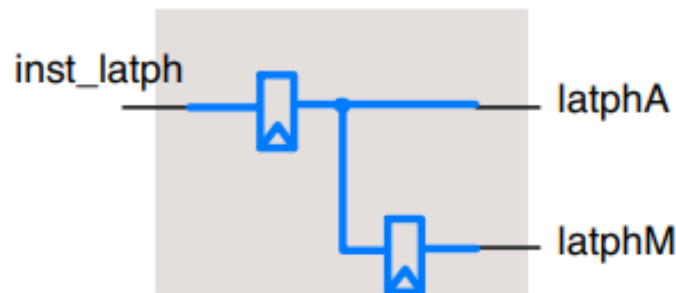
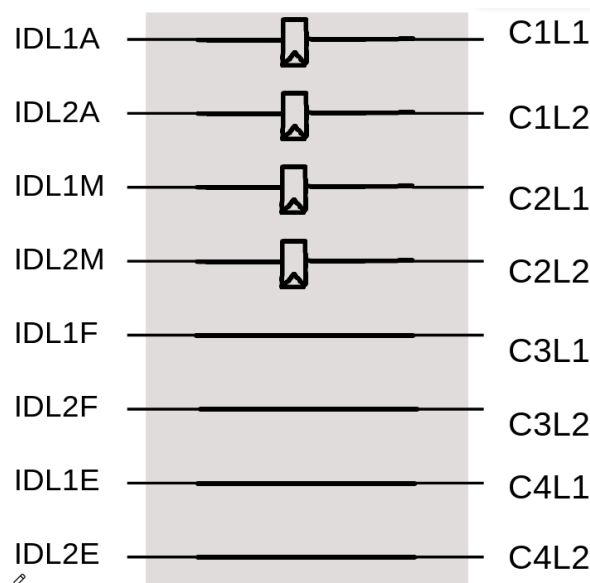


**Pregunta 1:** Diseñe el módulo LATPROH utilizando el menor número posible de registros y puertas lógicas, limitando el número de entradas de las puertas a 2. Justifique el diseño de forma sucinta y sistemática.



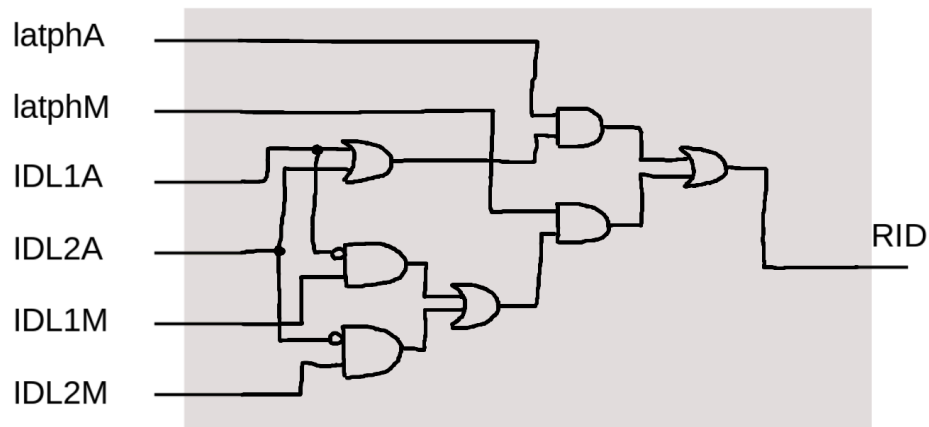
El módulo LATPROH debe emular la propagación de la señal `inst_latph` (generada en la etapa D/L) por las etapas ALU y M, por lo que utilizamos registros para ello.

**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 415). Justifique el diseño de forma sucinta y sistemática.



Si el dato que provoca el riesgo de datos está en las etapas de ALU o M, se deberá esperar un ciclo antes de enviar la señal de control a los multiplexores C1Lx y C2Lx debido a que están colocados al principio de la siguiente etapa. En caso de estar en FMTE o ES, como los multiplexores están al final de la etapa D/L, se puede enviar la señal de control en el mismo ciclo.

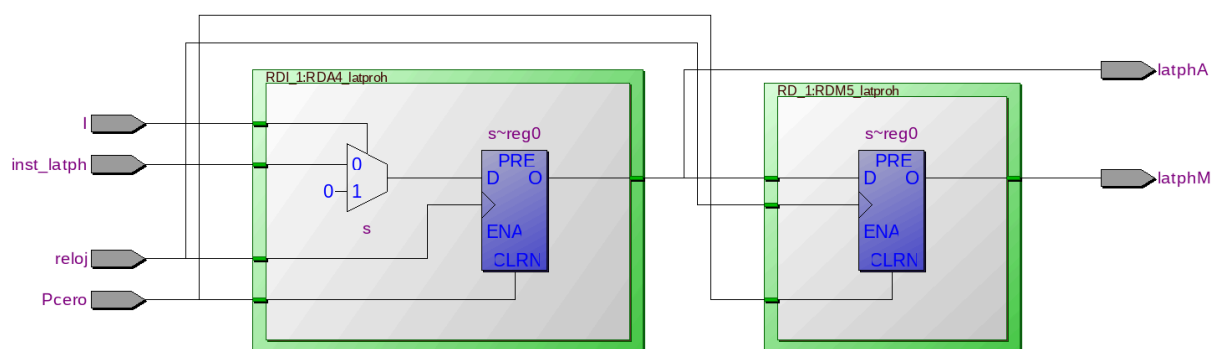
**Pregunta 3:** Diseñe el módulo LDRD utilizando el menor número posible de puertas lógicas, limitando el número de entradas de las puertas a 2. Justifique el diseño de forma sucinta y sistemática.



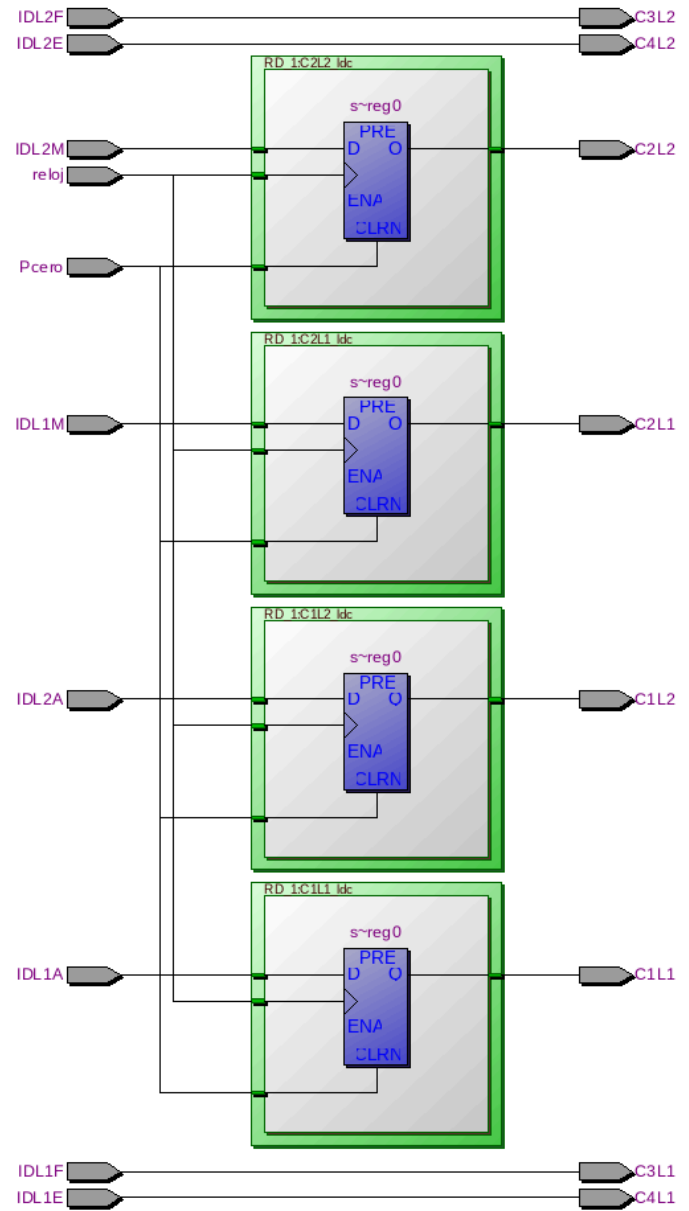
Para que se genere la señal del riesgo de datos debe haber al menos un registro fuente que se esté calculando en la misma etapa que se esté ejecutando una de las instrucciones que puede provocar riesgo de datos, sea en la etapa ALU o en M. Sin embargo, las dependencias en la etapa ALU para un registro fuente tienen más precedencia que las de la etapa M, por lo que se les da más prioridad.

**Pregunta 4:** En el subdirectorio LIB (Organización de los ficheros: árbol de directorios en la página 449) se encuentran los ficheros asociados al diseño de la lógica de interbloques (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 de la Lógica de Cortocircuitos e Interbloques. 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 correctos.

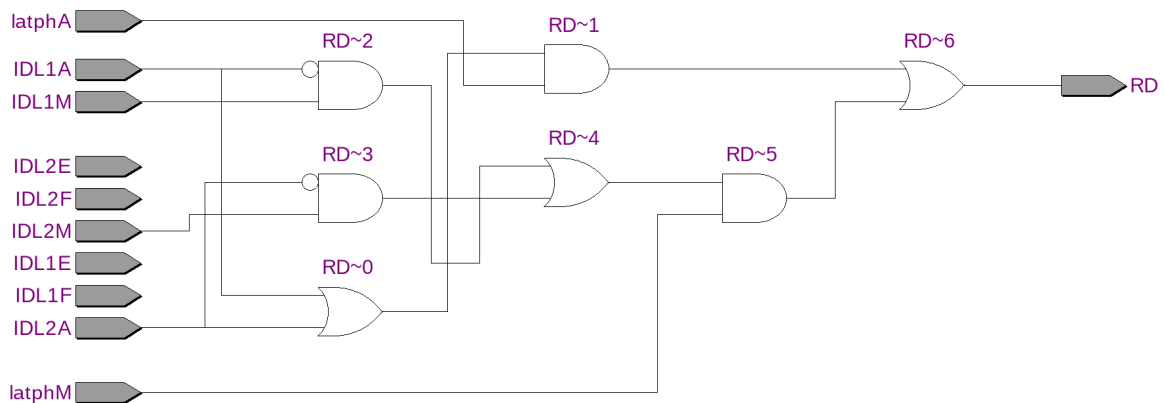
Módulo LATPROH:



## Módulo LDC:

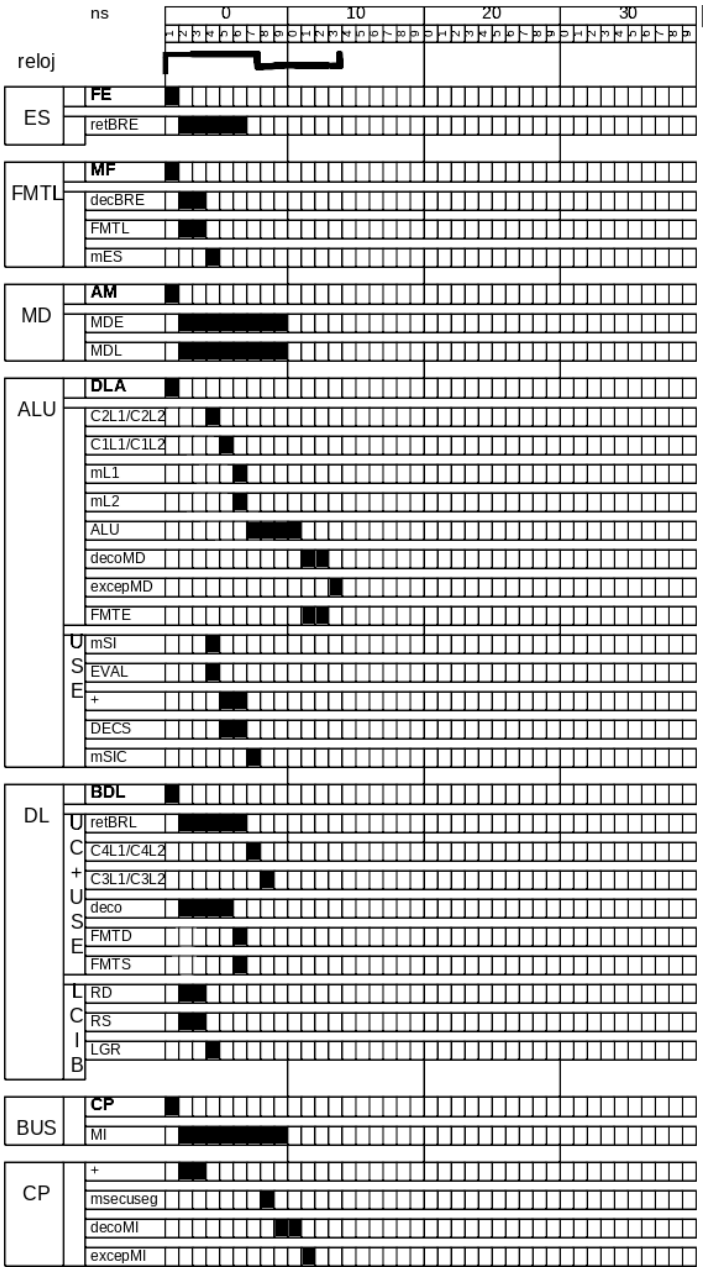


## Módulo LDRD:



Al hacer la simulación de los módulos con el programa de prueba que se proporciona hemos comprobado que las salidas son las esperadas.

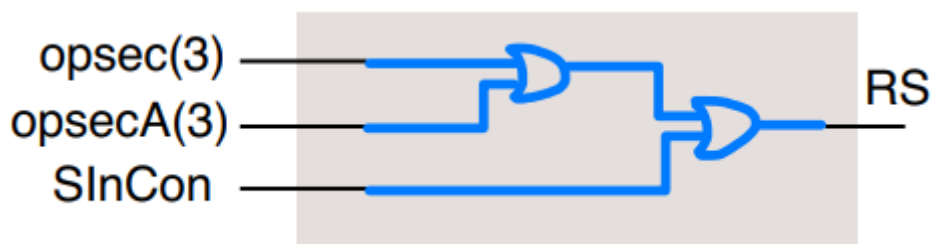




**Pregunta 7:** Utilice el programa char\_sort (Simulación en la página 422). Añada un proceso al programa de prueba (ENSAMBLADO/PRUEBAS/prueba\_Rproc\_MD\_MI\_C.vhd) para obtener las métricas indicadas en la tabla. Compare el rendimiento del procesador con cortocircuitos respecto del procesador segmentado sin cortocircuitos.

	Segmentado + cortocircuitos	Segmentado
Ciclos perdidos por RD	308	1940
Ciclos perdidos por RS	972	1929
Ciclos de ejecución	2308	3940
Tiempo de ciclo (ns)	80	80
Ganancia	1.71	

**Pregunta 8:** 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.



Habrá riesgo de secuenciamiento en caso de que haya un *jal* en DL u otra instrucción de secuenciamiento en DL/ALU.

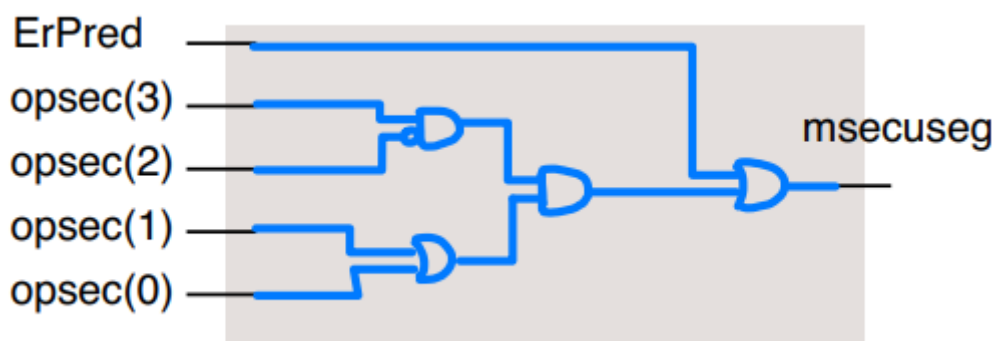
Incluya el diseño en el fichero LDRS\_ModSecu\_rell.vhd (Organización de los ficheros: reducción de la latencia al ejecutar la instrucción “jal” en la página 473). El proyecto quartus está definido en el subdirectorío PROC\_SEG\_ModSecu\_rell.

Indique, para el programa fact\_rekurs, 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 469). Para ello, añada un proceso al programa de prueba (prueba\_Rproc\_MD\_MI\_C\_ModSecu\_rell.vhd) que cuente los ciclos perdidos debidos a riesgos de secuenciamiento.

Programa	Antes de reducir la latencia	Una vez reducida la latencia	Reducción(%)
fact_rekurs	232	223	3,88%

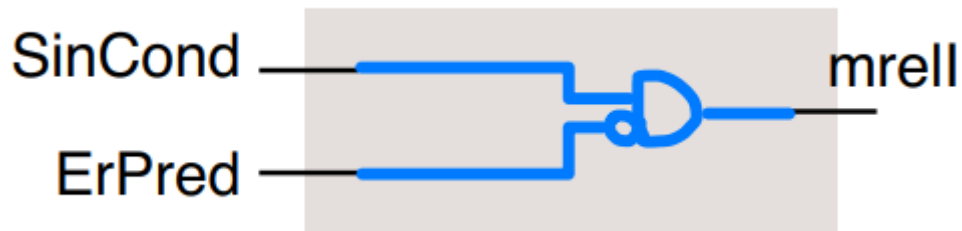
Las siguientes preguntas corresponden a la USE con Predicción de seguir en secuencia en instrucciones de secuenciamiento condicional (página 436).

**Pregunta 9:** Diseñe el módulo Ersec 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.



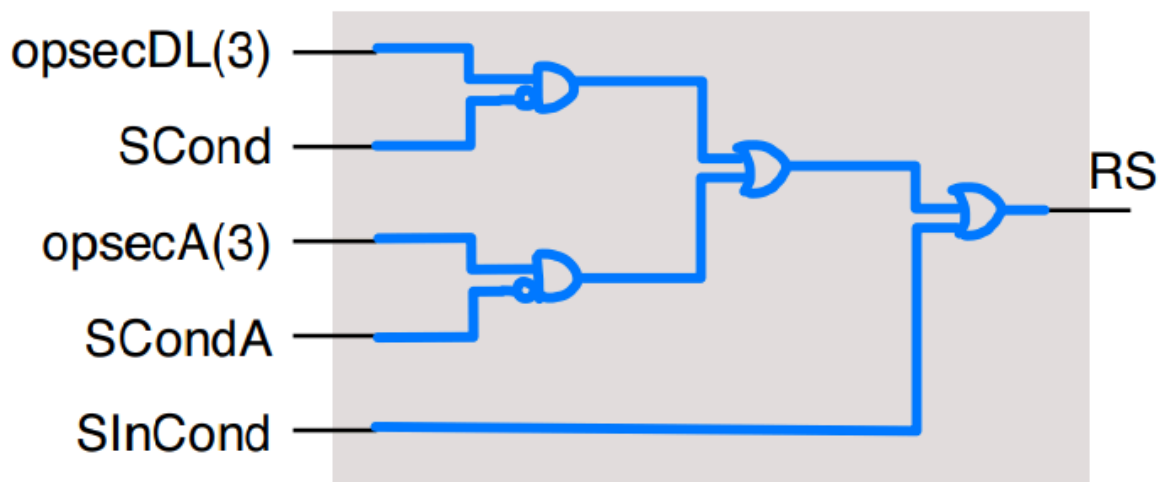
En este módulo miramos que en las instrucciones de secuenciamiento condicional solo se modifica el secuenciamiento cuando se produce un error de predicción.

**Pregunta 10:** Diseñe el módulo Errell 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.



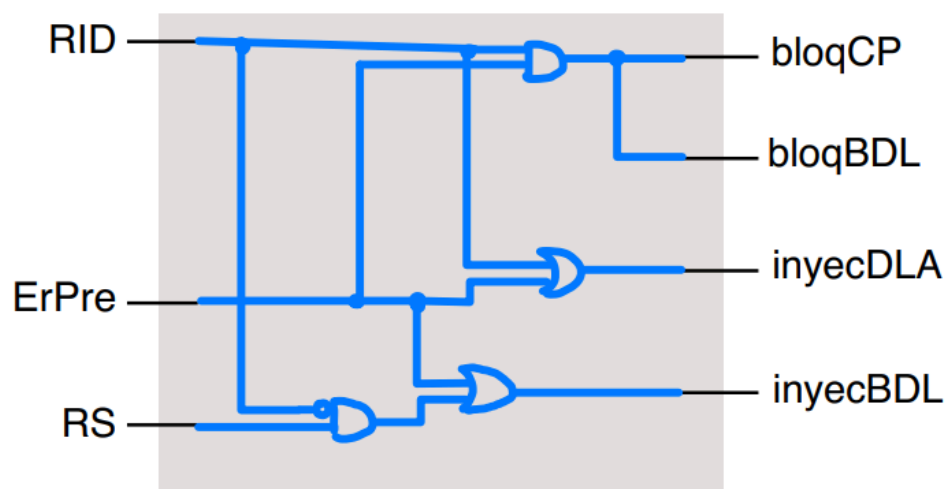
Hay que anular el secuenciamiento si el *jal* está mal predicho.

**Pregunta 11:** Diseñe el módulo LDRS utilizando el menor número posible puertas lógicas, limitando el número de entradas de las puertas a 2. Justifique el diseño de forma sucinta y sistemática.



Habrà riesgo de secuenciamiento si hay una instrucción incondicional o si hay una instrucción de salto condicional en DL o ALU y no siguen en secuencia.

**Pregunta 12:** Diseñe el módulo LGR utilizando el menor número posible puertas lógicas, limitando el número de entradas de las puertas a 2. Justifique el diseño de forma sucinta y sistemática.



Bloquearemos CP, B y DL cuando haya RID y error en predicción. Inyectamos en DL y ALU cuando hay riesgo de datos y/o error en predicción e inyectamos en B y DL cuando hay solo riesgo de secuenciamiento y/o error en predicción.

**Pregunta 13:** Incluya los diseños anteriores en los ficheros correspondientes (Organización de los ficheros: predicción de seguir en secuencia en la página 483). El proyecto quartus está definido en el subdirectorio PROC\_SEG\_ModSecu\_rell\_PreSecu.

Indique para los siguientes programas la reducción en ciclos perdidos por riesgos de secuenciamiento cuando se predice seguir en secuencia (Simulación: predicción de seguir en secuencia en la página 479). Para ello modifique el programa de pruebas. Indique también el número de predicciones y el número de errores de predicción.

Programa	Latencia reducida (jal)	Latencia reducida + predicción	Reducción	Predicciones	Errores
Euclides	58	8	50	33	12
Sort	672	14	658	510	246

**Pregunta 14:** Observe que los dos mecanismos de predicción fija descritos utilizan la misma unidad de secuenciamiento, solo cambia el control de los multiplexores de encaminamiento de direcciones (página 438, página 442). Indique, para cada mecanismo de predicción, qué multiplexores de la USE se podrían eliminar.

Predicción	MUX que se pueden eliminar	Justificación
Fija seguir en secuencia	mSIC	En caso de fallo de predicción el único valor que importa escoger es la dirección calculada.
Fija modificar el secuenciamiento	mSI	En caso de fallo de predicción el valor que importa escoger es el CP de la secuencia original, o bien la dirección correcta.