

Contenido

Práctica 6 Procesador: segmentación lineal con cortocircuitos 405

Incremento del rendimiento	405
Procesador segmentado con cortocircuitos	406
Cortocircuitos: Unidad de cálculo	407
Cortocircuitos: Unidad de secuenciamiento	408
Riesgos de datos debidos a registros	410
Riesgo de secuenciamiento	412
Interacción entre riesgos	413
Lógica de cortocircuitos e interbloques	415
Camino de datos completo	418
Tiempo de ciclo	419
Señal de reloj, señal de inicio y elementos de almacenamiento	422
Simulación	422
Preparación de la simulación	423
Simulación con ModelSim	423
Simulación de un programa concreto	423
Evolución de las señales del camino de datos	424
Información textual de la simulación	428
Modificación de la unidad de secuenciamiento	430
Modificación de la segmentación de la USE	431
Reducción de la latencia en la instrucción de secuenciamiento "jal"	432
Predicción de sentido en instrucciones de secuenciamiento condicional	435
Predicción fija de seguir en secuencia.	436
Predicción fija de modificar el secuenciamiento.	440
Predicción de sentido en función del signo del literal	443
Apéndice6.1: Retardos	447
Apéndice6.2: Organización de los ficheros: árbol de directorios	449
Apéndice6.3: Documentación	451
Apéndice6.4: Implementación de la activación de cortocircuitos y gestión de riesgos de datos	453
Caracterización de registros fuente y destino	454
Cortocircuitos disponibles y riesgos de datos	455
Apéndice6.5: Simulación: modificación de la segmentación de la USE	459
Preparación de la simulación	459
Simulación con ModelSim	459
Simulación de un programa concreto	460
Información textual de la simulación	460

Apéndice6.6: Organización de los ficheros: modificación de la segmentación de la USE	461
Apéndice6.7: Documentación: modificación de la segmentación de la USE	463
Apéndice6.8: Reducción de la latencia al ejecutar la instrucción "jal": control del encaminamiento por instrucción	465
Apéndice6.9: Reducción de la latencia al ejecutar la instrucción "jal": señales de control del secuenciamiento	467
Apéndice6.10: Simulación: reducción de la latencia al ejecutar la instrucción "jal"	469
Preparación de la simulación	469
Simulación con ModelSim	469
Simulación de un programa concreto	470
Información textual de la simulación	471
Apéndice6.11: Organización de los ficheros: reducción de la latencia al ejecutar la instrucción "jal"	473
Apéndice6.12: Documentación: reducción de la latencia al ejecutar la instrucción "jal"	475
Apéndice6.13: Predicción de seguir en secuencia: señales de control del secuenciamiento condicional	477
Apéndice6.14: Simulación: predicción de seguir en secuencia	479
Preparación de la simulación	479
Simulación con ModelSim	479
Simulación de un programa concreto	480
Información textual de la simulación	480
Apéndice6.15: Organización de los ficheros: predicción de seguir en secuencia	483
Apéndice6.16: Documentación: predicción de seguir en secuencia	485
Apéndice6.17: Predicción de modificar el secuenciamiento: control del encaminamiento por instrucción	487
Apéndice6.18: Predicción de modificar el secuenciamiento: señales de control del secuenciamiento condicional	489
Apéndice6.19: Simulación: predicción de modificar el secuenciamiento	491
Preparación de la simulación	491
Simulación con ModelSim	491
Simulación de un programa concreto	491
Información textual de la simulación	492
Apéndice6.20: Organización de los ficheros: predicción de modificar el secuenciamiento	495
Apéndice6.21: Documentación: predicción de modificar el secuenciamiento	497
Apéndice6.22: Predicción en función del signo del literal: control del encaminamiento por instrucción	499
Apéndice6.23: Predicción en función del signo del literal: señales de control del secuenciamiento condicional	501
Apéndice6.24: Simulación: predicción en función del signo del literal	503
Preparación de la simulación	503
Simulación con ModelSim	503
Simulación de un programa concreto	503
Información textual de la simulación	505
Apéndice6.25: Organización de los ficheros: predicción en función del signo del literal	

507

Apéndice6.26: Documentación: predicción en función del signo del literal 509



Práctica 6

Procesador: segmentación lineal con cortocircuitos

.....

En esta sesión se trata de consolidar los conocimientos adquiridos sobre cortocircuitos en un procesador con el camino de datos segmentado y el control del mismo.

Para ello partiremos del procesador segmentado de la Práctica 5. Los cortocircuitos tienen su destino en la etapa DL o en la etapa ALU.

Un objetivo de la práctica es identificar las posibles fuente de cortocircuito y la determinación de la etapa destino teniendo en cuenta posibles restricciones en el tiempo de ciclo. Así mismo, se analiza cuál de las etapas determina el tiempo de ciclo y la duración de la señal de reloj en cada uno de los dos niveles lógicos. Para ello es necesario determinar el tiempo requerido por cada etapa, el cual es función del retardo de los componentes utilizados por cada tipo de instrucción y el conexionado de estos componentes.

Otros objetivos son: a) reducir la latencia al ejecutar instrucciones de secuenciamiento incondicional con direccionamiento relativo y b) utilizar técnicas de predicción de sentido en instrucciones de secuenciamiento condicional.

Incremento del rendimiento

En el procesador segmentado desarrollado en la Práctica 5 se pierden ciclos por riesgos de datos debidos al banco de registros. En concreto, se pueden llegar a perder cuatro ciclos entre la interpretación de una instrucción productora y una instrucción consumidora.

Un análisis detallado de la segmentación indica que los ciclos perdidos, en la mayoría de los casos, son debidos a la latencia de actualización del banco de registros y no a la falta de disponibilidad del valor que se quiere utilizar como operando. Por ejemplo, el resultado de una operación en la ALU no se escribe en el banco de registros hasta tres ciclos después de disponer del valor y este valor no se puede leer del banco de registros hasta cuatro ciclos después de ser calculado en la ALU. En el caso de una instrucción de lectura en memoria, el valor no se puede leer del banco de registros hasta tres ciclos

después. Finalmente, en el caso de una instrucción de secuenciamiento, que almacena una dirección en el banco de registros, son necesarios seis ciclos para poder leer el valor del banco de registros.

Para disponer del valor y poder utilizarlo antes de que el banco de registros esté actualizado se pueden añadir interconexiones en el camino de datos, denominadas cortocircuitos.

Otra fuente de ciclos perdidos es la latencia en la actualización del registro CP en instrucciones de secuenciamiento. Un análisis de las instrucciones de secuenciamiento muestra que el cálculo de la dirección destino, cuando es relativo a la dirección de la instrucción, puede efectuarse en la etapa DL. Esto es, un ciclo antes que en el diseño de la Práctica 5. Para efectuar el cálculo en la etapa DL hay que rediseñar el camino de datos de la USE.

Por otro lado, dada una instrucción de secuenciamiento condicional, se puede predecir¹ en la etapa DL el resultado de evaluar la condición. Cuando la predicción es correcta la latencia efectiva de ejecución de estas instrucciones se reduce. Para efectuar la predicción y la acción de recuperación en el caso de error de predicción es necesario modificar el control de la USE y, dependiendo de la información que se utiliza para efectuar la predicción, el camino de datos de la USE.

Procesador segmentado con cortocircuitos

El procesador segmentado con cortocircuitos tarda el mismo número de ciclos en interpretar una instrucción (7 ciclos) y utiliza las mismas etapas que el procesador segmentado. Por completitud las etapas utilizadas se muestran en la Figura 6.1.

ciclos	1	2	3	4	5	6	7
	CP	BUS	DL	ALU	M	FMTL	ES

Figura 6.1 Etapas del procesador segmentado lineal con cortocircuitos.

La funcionalidad de las etapas es idéntica a la descrita en la Práctica 5 del procesador segmentado. Por completitud, en la tabla de la Figura 6.2 se describe la funcionalidad de las etapas.

1. Efectuar una hipótesis.

ETAPA	FUNCIONALIDAD
CP	Determinación de la dirección de la instrucción.
BUS	Búsqueda de la instrucción.
DL	Decodificación y lectura de operandos en registros del banco de registros.
ALU	Operaciones aritméticas y lógicas. Cálculo de la dirección efectiva en instrucciones de acceso a memoria. Adicionalmente, cálculo de la dirección destino de salto, evaluación de la condición en instrucciones de secuenciamiento condicional y actualización de la entrada del registro CP. Además, formateo de la escritura en memoria y activación de las señales de escritura en memoria.
M	Acceso a la memoria de datos
FMTL	Formateo de la lectura de memoria.
ES	Escritura en el banco de registros.

Figura 6.2 Descripción funcional de las etapas en el procesador segmentado con cortocircuitos.

Cortocircuitos: Unidad de cálculo

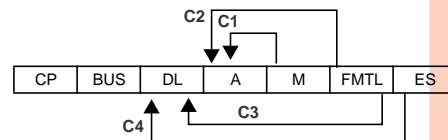
Un cortocircuito requiere el encaminamiento de la información mediante un bus y un multiplexor para seleccionar entre posibles caminos, por lo cuales puede transmitirse el valor que se quiere utilizar. Estos multiplexores se ubican entre la salida del banco de registros y la unidad funcional ALU.

En el diseño que se está efectuando, la fuente de un cortocircuito puede ser el valor calculado en la ALU, el valor disponible al final de la etapa FMTL y el valor en la etapa ES (Figura 6.3). Recordemos que, con el tiempo de ciclo mínimo, no es posible escribir y leer un mismo registro en el mismo ciclo.

En el diseño que se está efectuando supondremos que el tiempo de ciclo está determinado por la lectura del banco de registros. Por tanto, los multiplexores se distribuyen entre la etapa DL y la etapa ALU.

dir.	instrucción	ciclos											
		1	2	3	4	5	6	7	8	9	10	11	
0	addu x6, x5, x2	CP	BUS	D/L	ALU	M	FMTL	ES					
4	addiu x7, x6, 0x0400		CP	BUS	D/L	ALU	M	FMTL	ES				
8	subu x8, x7, x6			CP	BUS	D/L	ALU	M	FMTL	ES			
12	ori x9, x6, 0x5555				CP	BUS	D/L	ALU	M	FMTL	ES		
16	and x10, x6, x3					CP	BUS	D/L	ALU	M	FMTL	ES	
20	xor x12, x6, x14						CP	BUS	D/L	ALU	M	FMTL	ES
cortocircuito (C)						1	2, 3	4					

Figura 6.3 Conjuntos de cortocircuitos.



En la etapa DL es necesario disponer de un cortocircuito, por puerto de lectura, para disponer del valor que se está utilizando para actualizar un registro del banco de registros, si éste es necesario. Entonces, la fuente de estos cortocircuitos es la etapa ES y el destino es la etapa DL (C4)². Otros cortocircuitos en la etapa DL tienen como fuente el valor disponible al final de la etapa FMTL (C3).

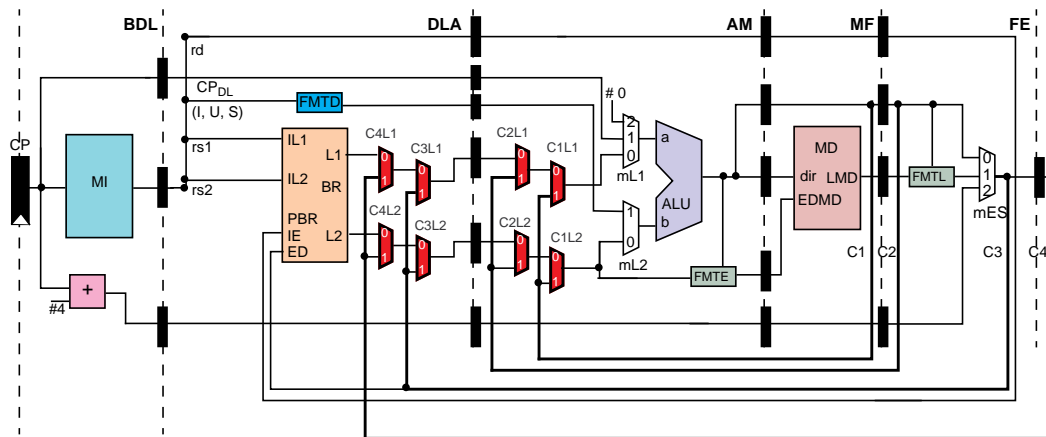


Figura 6.4 Unidad de cálculo con cortocircuitos.

Los otros cortocircuitos tienen como destino la etapa ALU. Un conjunto de ellos suministra el valor que se ha calculado en el ciclo anterior en la etapa ALU (C1, etapa fuente M). Otro conjunto suministra el valor, calculado en la ALU, que está en la etapa FMTL (C2). En la Figura 6.4 se muestra la unidad de cálculo con los ocho cortocircuitos. En el Apéndice 6.4 se detalla un análisis de todos los posibles cortocircuitos.

Cortocircuitos: Unidad de secuenciamiento

Por ejecución de una instrucción de secuenciamiento entendemos el establecimiento de la dirección de la instrucción que debe interpretarse después de la instrucción de secuenciamiento.

Algunas instrucciones de secuenciamiento utilizan el contenido de registros para evaluar la condición. En consecuencia, si se quieren eliminar algunos ciclos perdidos por riesgos de datos, debidos al banco de registros, hay que evaluar la condición en la etapa ALU.

2. Notemos que en la Figura 6.4 se utiliza un registro de desacople después de la etapa FMTL para disponer del valor en la etapa ES.

En estas condiciones, no se modifican las ubicaciones de la lógica USE, en las etapas, respecto del procesador segmentado. Todas las instrucciones de secuenciamiento establecen el secuenciamiento en la etapa ALU (Figura 6.5).

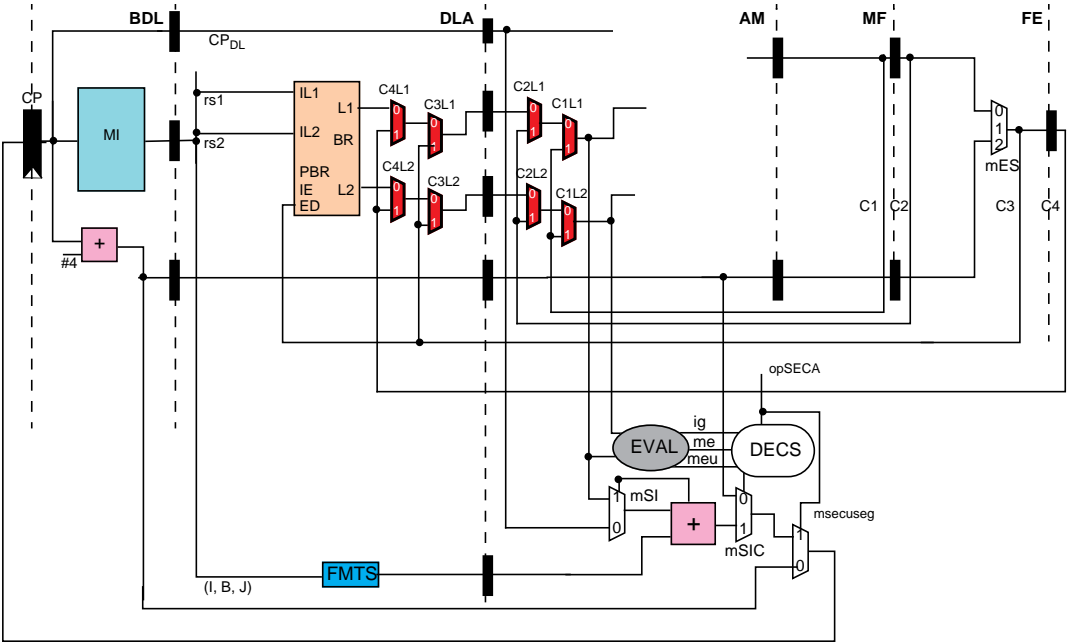


Figura 6.5 Cortocircuitos y unidad de secuenciamiento.

Riesgos de datos debidos a registros

En la Figura 6.6 se muestra un esquema del camino de datos donde se observan las etapas desde DL hasta ES dispuestas en una columna. El bus ED transporta el valor con el cual se actualiza el banco de registros en el flanco ascendente de la señal de reloj.

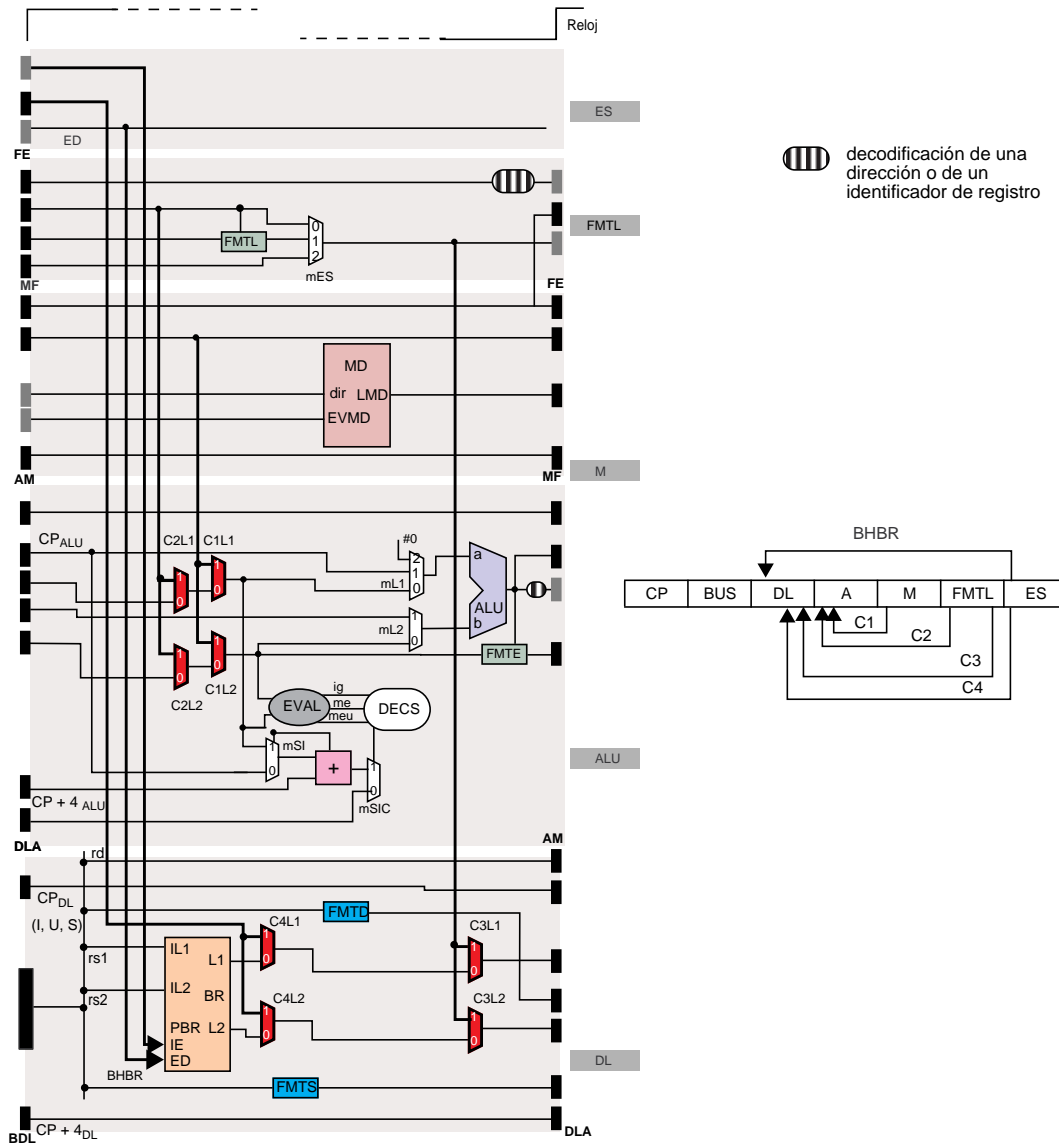


Figura 6.6 Bucle hardware utilizado en la actualización del banco de registros y bucles hardware que implementan los cortocircuitos. El simbolismo de los registros sigue las pautas descritas en la Práctica 5.

La actualización del banco de registros utiliza un bucle hardware (BHBR) desde la etapa ES hasta la etapa DL y la latencia del bucle es 5 ciclos (Figura 6.6).

Los bucles hardware C1L1 y C1L2 permiten disponer, en las entradas de la ALU, del valor generado en la ALU en el ciclo previo. Los bucles hardware C2L1, C2L2 permiten disponer del valor calculado en la ALU dos ciclos antes.

Los bucles hardware C3L1 y C3L2 permiten disponer en la etapa DL del valor disponible al final de la etapa FMTL. Este valor se ha generado dos ciclos antes en la etapa ALU, es un valor leído de la MD y procesado en la lógica FMTL o es la dirección que se almacena al interpretar una instrucción de secuenciamiento con enlace (salida de mES)³. Los bucles hardware C4L1 y C4L2 permiten disponer en la etapa DL del valor utilizado para actualizar el banco de registros (etapa ES).

La latencia de los bucles hardware C1L1 y C1L2 es un ciclo. La latencia de los bucles hardware C2L1 y C2L2 son dos ciclos. La latencia de los bucles hardware C3L1 y C3L2 son tres ciclos. Por último, la latencia de los bucles hardware C4L1 y C4L2 es 4 ciclos.

Los cortocircuitos descritos no son suficientes para eliminar todos los riesgos de datos debidos a registros. Una instrucción load de un byte (lb) tienen una latencia de ejecución de tres ciclos (etapas ALU, M y FMTL). Sin embargo, la latencia de inicio de interpretación de instrucciones es un ciclo. Por tanto, no se dispone del valor cuando la instrucción que sigue a la instrucción load es consumidora (Figura 6.7). Tampoco se dispone del valor en el siguiente ciclo.

dir.	instrucción	ciclos											
		1	2	3	4	5	6	7	8	9	10	11	12
0	lb x25, 4(x30)	CP	BUS	D/L	ALU	M	FMTL	ES					
4	addiu x7, x25, 0x0400		CP	BUS	D/L	nop	nop	nop	nop				
						D/L	nop	nop	nop	nop			
							D/L	ALU	M	FMTL	ES		
8	and x10, x25, x3			CP	BUS	BUS	BUS	D/L	ALU	M	FMTL	ES	
12	xor x12, x6, x14				CP	CP	CP	BUS	D/L	ALU	M	FMTL	ES
cortocircuito (C)							C3L1	C4L1					

Figura 6.7 Riesgo de datos debido a registros.

En el diseño, del procesador segmentado con cortocircuitos, efectuado no se distingue entre tipos de load (lw, lh, lb) al decidir los cortocircuitos implementados (Apéndice 6.4). Por otro lado, el destino de los cortocircuitos son las etapas DL o ALU. En consecuencia, hay que emular un funcionamiento serie cuando la instrucción productora es una instrucción load y una instrucción a distancia 1 o 2 es consumidora. El número de ciclos perdidos es dos o uno respectivamente.

3. Notemos que, en el diseño efectuado, este último valor no está disponible para una instrucción consumidora hasta este ciclo, aunque el valor es conocido cuando la instrucción de secuenciamiento está en la etapa DL.

Riesgo de secuenciamiento

412 · Práctica 6

El bucle hardware BHS1 de latencia uno, entre la etapa BUS y la etapa CP, se utiliza para efectuar el secuenciamiento implícito. Como la latencia del bucle es igual a la latencia de inicio de interpretación de instrucciones, se respeta la semántica del lenguaje máquina.

El bucle hardware BHS2⁴ de latencia tres, entre la etapa ALU y la etapa CP, se utiliza para el secuenciamiento explícito. La semántica de todas las instrucciones de secuenciamiento indica que la siguiente instrucción que debe interpretarse es la que determina la instrucción de secuenciamiento. Entonces, como la latencia de ejecución de una instrucción de secuenciamiento son tres ciclos (efectividad en la modificación o no del secuenciamiento) no se respeta la semántica del lenguaje máquina.

Para adecuar la semántica se emula un funcionamiento serie del procesador. Esta emulación requiere que se descarten las instrucciones buscadas después de la instrucción de secuenciamiento.

En el procesador diseñado hay que descartar dos instrucciones. Una forma de descartarlas es inyectar una instrucción nop⁵ desde la etapa BUS cuando la instrucción de secuenciamiento ocupa la etapa DL o la etapa ALU. Por tanto, se pierden dos ciclos al ejecutar una instrucción de secuenciamiento (Figura 6.9).

		ciclos										
dir.	instrucción	1	2	3	4	5	6	7	8	9	10	11
8	bne x3, x8, 1\$	CP	BUS	D/L	ALU	M	FMTL	ES				
12	add x26, x0, x4		CP	BUS	nop	nop	nop	nop				
16	or x29, x2, x6			CP	BUS	nop	nop	nop	nop			
1\$: 100	slt x20, x5, x6				CP	BUS	D/L	ALU	M	FMTL	ES	
104	and x9, x9, x7					CP	BUS	D/L	ALU	M	FMTL	ES

recursos + + USE +

Figura 6.9 Riesgo de secuenciamiento y gestión del mismo.

Interacción entre riesgos

El diseño de la gestión de riesgos de datos y de secuenciamiento ha sido efectuado de forma independiente. Esto es, suponiendo que en un ciclo determinado solo se puede detectar un riesgo o que mientras se gestiona un tipo de riesgo no se detecta otro tipo de riesgo. En esta apartado se analizan las interacciones entre los tipos de riesgos.

- En este bucle solo se muestra un trazo más grueso en la señal de salida del multiplexor mSIC. Ahora bien, cualquier otra señal que determina este valor en el mismo ciclo pertenece al bucle hardware.
- Instrucción addi x0, x0, #0, cuya codificación es 0x00000013.

Detección de un riesgo de datos en una instrucción de secuenciamiento. En la Figura 6.10 se muestra una secuencia de instrucciones donde la instrucción de secuenciamiento detecta un riesgo de datos. En primer lugar se gestiona el riesgo de datos y posteriormente el riesgo de secuenciamiento.

		ciclos												
dir.	instrucción	1	2	3	4	5	6	7	8	9	10	11	12	13
0	lw x3 , 4(x30)	CP	BUS	D/L	ALU	M	FMTL	ES						
4	bne x3, x2, 1\$		CP	BUS	D/L	nop	nop	nop	nop					
						D/L	nop	nop	nop	nop				
							D/L	ALU	M	FMTL	ES			
8	and x10 , x25, x3			CP	BUS	BUS	BUS	nop	nop	nop	nop	nop		
12	xor x12 , x6, x14				CP	CP	CP	BUS	nop	nop	nop	nop	nop	
1\$:	100 slt x20 , x5, x6							CP	BUS	D/L	ALU	M	FMTL	ES
cortocircuito (C) y riesgos					RD	RD	RS	RS						
							C3L1							

Figura 6.10 Instrucción de secuenciamiento que detecta un riesgo de datos. Suponemos que se modifica el secuenciamiento.

Detección de un riesgo de datos mientras se está gestionando un riesgo de secuenciamiento. En la Figura 6.11 se muestra una secuencia de instrucciones donde la instrucción, la cual determina la instrucción de secuenciamiento, que debe interpretarse depende de una instrucción load (latencia máxima de ejecución, que es igual a 3)⁶. Durante los ciclos que se gestiona el riesgo de secuenciamiento no se puede detectar ningún otro riesgo, ya que se inyectan instrucciones nop.

		ciclos												
dir.	instrucción	1	2	3	4	5	6	7	8	9	10	11	12	13
0	lw x12 , 4(x30)	CP	BUS	D/L	ALU	M	FMTL	ES						
4	bne x3, x2, 1\$		CP	BUS	D/L	ALU	M	FMTL	ES					
8	and x10 , x25, x12			CP	BUS	nop	nop	nop	nop	nop				
12	xor x12 , x6, x14				CP	BUS	nop	nop	nop	nop	nop			
1\$:	100 slt x20 , x5, x12					CP	BUS	D/L	ALU	M	FMTL	ES		
cortocircuito (C) y riesgos					RS	RS		C4L2						

Figura 6.11 Durante los ciclos de gestión de un riesgo de secuenciamiento no se detectan riesgos.

6. Si la instrucción de secuenciamiento es una instrucción de secuenciamiento incondicional o indexado y ella es fuente de la dependencia tampoco se detecta un riesgo de datos.

Lógica de cortocircuitos e interbloqueos

En la Figura 6.12 se muestran los conjuntos de registros de desacoplo del camino de datos y las funcionalidades de que deben disponer para efectuar el control de riesgos. La actuación en cada tipo de riesgo es la misma que en la Práctica 5.

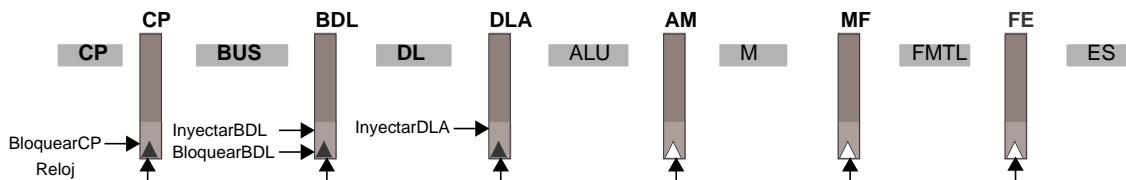


Figura 6.12 Funcionalidades de los conjuntos de registros de desacoplo en el camino de datos.

La detección de riesgos de datos y la decisión de utilizar cortocircuitos se efectúa cuando una instrucción está en la etapa DL. Una vez detectada una dependencia se decide si produce riesgo o puede utilizarse un cortocircuito. La actuación se efectúa en el ciclo correspondiente, que puede ser el mismo ciclo o posterior al de la detección o decisión.

La Lógica de Cortocircuitos e InterBloqueos (LCIB) se divide en varios módulos y submódulos (Figura 6.13). En el módulo LDC no se indican las señales reloj y pzero.

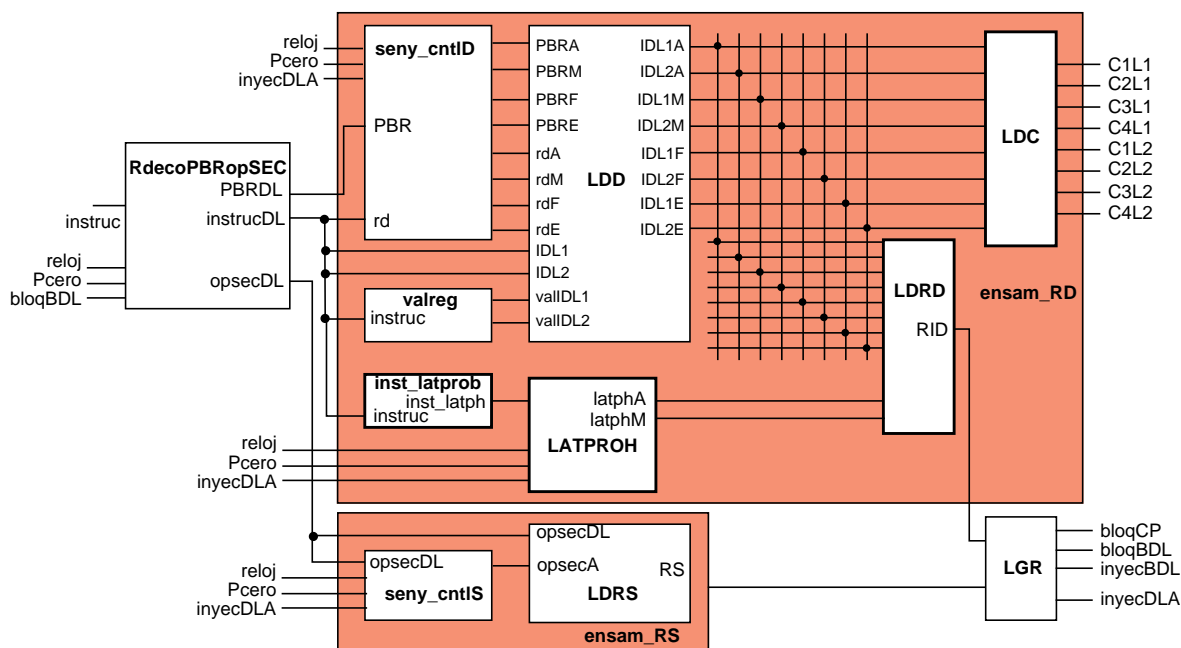


Figura 6.13 Módulos utilizados en la lógica de cortocircuitos e interbloqueos (LCIB). Los subíndices en las señales se refieren a etapa (A para ALU).

Los módulos adicionales o modificados, respecto al procesador segmentado, se resaltan con trazo grueso. Algunos módulos y submódulos recrean localmente, en la lógica LCIB, señales del camino de datos. Estos elementos son los mismos que en el procesador segmentado (Figura 6.14)⁷. En concreto, las señales que se generan en estos elementos son las utilizadas para detectar dependencias de datos y riesgos de secuenciamiento. La descripción de los módulos ha sido efectuada en la Práctica 5.

Módulos/submódulos	RdecoPBRopSEC
	seny_cntlD
	seny_cntlS

Figura 6.14 Módulos y submódulos que recrean señales del camino de datos.

El resto de módulos y submódulos se utiliza para detectar dependencias de datos y riesgos de secuenciamiento y decidir la actuación que se lleva a término⁸.

Los submódulos utilizados en la detección de riesgos y control de los cortocircuitos están incluidos en los módulos:

- ensam_RD: Lógica de detección de riesgo de datos y control de los cortocircuitos.
- ensam_RS: Lógica de detección de riesgo de secuenciamiento.

El módulo de actuación o de gestión de riesgos se denomina “lógica de gestión de riesgos” (LGR).

El módulo ensam_RD, excluyendo los submódulos que recrean señales, incluye los siguiente submódulos:

- valreg: Lógica de validación. El módulo es idéntico al utilizado en la Práctica 5 y su funcionalidad ha sido descrita en la misma.
- LDD: Lógica de detección de Dependencias de Datos debido a registros. El módulo es idéntico al utilizado en la Práctica 5 y su funcionalidad ha sido descrita en la misma.
- LDC: Lógica de Decisión para el control de los Cortocircuitos. La generación de las señales para controlar los cortocircuitos se efectúa en la etapa DL⁹. Ahora bien, su utilización puede ser en la etapa DL o en la etapa ALU.

7. Por tanto, hay que efectuar las mismas acciones que en el camino de datos cuando se gestionan riesgos.

8. Recordemos que se emula un funcionamiento serie cuando no se puede utilizar un cortocircuito en una dependencia de datos.

9. La señal de control de los cortocircuitos se determina independientemente de si hay un riesgo de datos.

- Inst_latprob: Teniendo en cuenta los cortocircuitos implementados en el camino de datos, esta lógica identifica en la etapa DL las instrucciones que pueden producir riesgos de datos debidos a registros¹⁰.

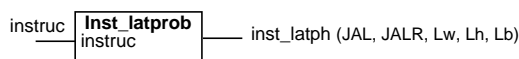


Figura 6.15 Módulo *Inst_latprob*. Señal que indica si la instrucción que ocupa la etapa DL puede producir riesgos de datos.

- LATPROH: propagación de la información identificada en el submódulo “Inst_latprob” por las etapas. Generación de las señales que indican que, en curso de interpretación, hay una instrucción que puede producir un riesgo de datos debido a registros¹¹ (etapa ALU, latphA y etapa M, latphM).
- LDRD: Lógica de Detección de Riesgos de Datos debidos a registros.

El módulo ensam_RS, excluyendo el módulo que recrea una señal, incluye el siguiente submódulo:

- LDRS: Lógica que genera la señal de riesgo de secuenciamiento. El módulo es idéntico al utilizado en la Práctica 5 y su funcionalidad ha sido descrita en la misma.

El módulo LGR, a partir de las señales de riesgo de datos y riesgo de secuenciamiento, genera las señales de control para los registros de desacoplo (CP, BDL y DLA, Figura 6.12). El módulo es idéntico al utilizado en la Práctica 5 y su funcionalidad ha sido descrita en la misma.

Módulo LCIB. El módulo LCIB comprueba, en cada ciclo de reloj, si los operandos fuente de la instrucción que ocupa la etapa DL hay que obtenerlos utilizando cortocircuitos o si aún no están disponibles. Así mismo, determina si se produce un riesgo de secuenciamiento.

En la Figura 6.13 se muestra un esquema de los módulos utilizados en la lógica de cortocircuitos e interbloqueos y las señales de entrada y salida. La señal Cx tiene que tomar el valor uno cuando se quiere obtener el valor del cortocircuito. La señal RID tiene que tomar el valor uno cuando hay que gestionar un riesgo de datos. La señal RS debe tomar el valor uno cuando se quiere gestionar un riesgo de secuenciamiento.

10. En el diseño de esta lógica no se tiene en cuenta que, debido al diseño utilizado en la lógica USE, un riesgo de secuenciamiento inhabilita la detección de un riesgo de datos. Ahora bien, en otro diseño de la lógica USE podría detectarse un riesgo de datos (interacción entre riesgos). Esta decisión no afecta al rendimiento, ya que si no se puede producir un riesgo, este no se detecta.
11. En este módulo hay que tener en cuenta la posibilidad de que se inyecte una instrucción nop desde la etapa DL cuando se produce un riesgo de datos.

Camino de datos completo

En la Figura 6.16 se muestra el esquema completo del camino de datos del procesador, en el cual se han especificado todas las señales menos la señal de reloj y la señal de inicialización. tampoco se muestra el control de parte de los registros de desacoplo.

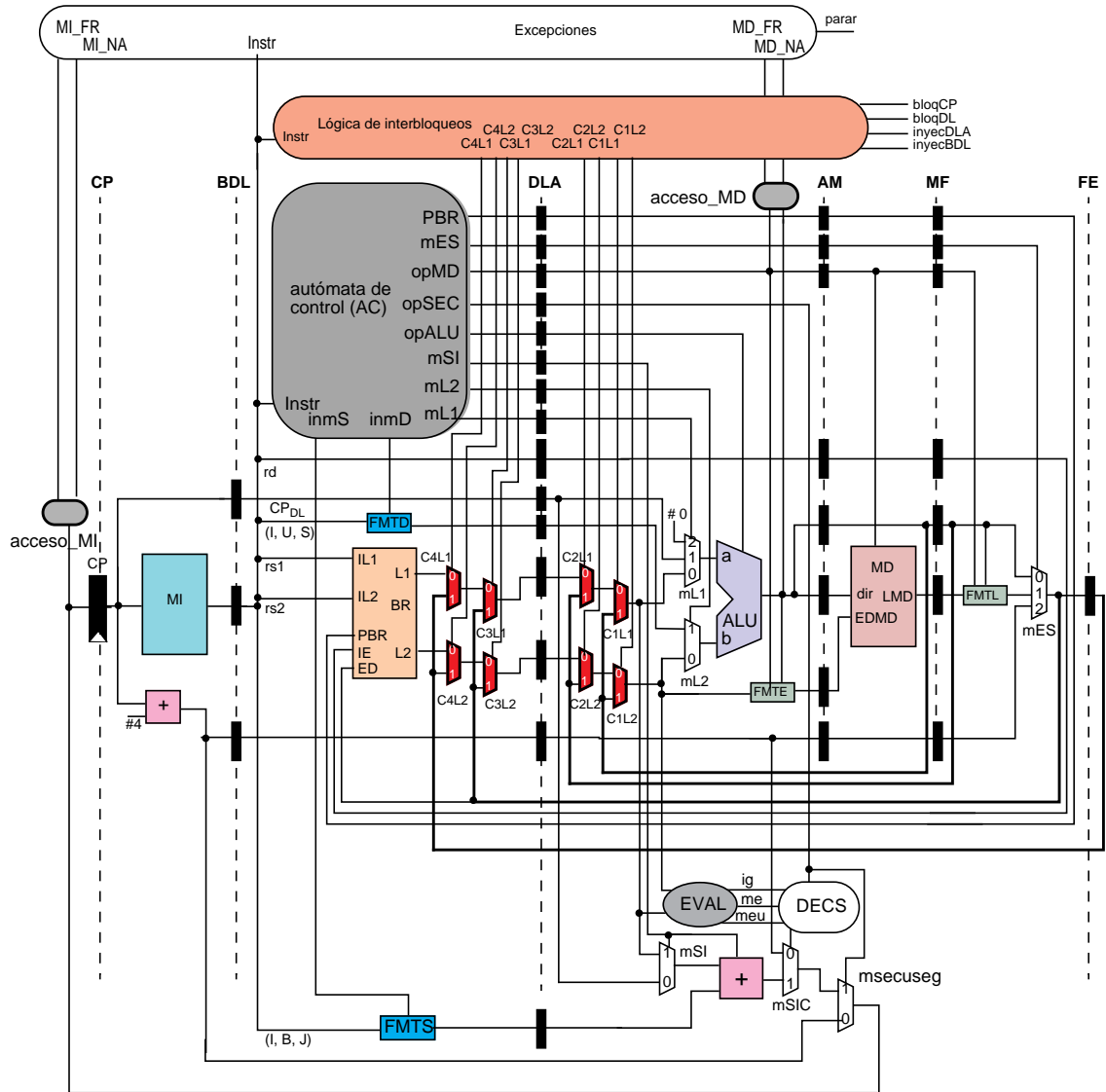


Figura 6.16 Camino de datos del procesador segmentado con cortocircuitos.

Tiempo de ciclo

Para ayudar en el cálculo del tiempo de ciclo, en la parte derecha de la Figura 6.17 se muestra una disposición de los elementos del camino de datos que ayuda a relacionar los retardos con el periodo de la señal de reloj. Las etapas se han dibujado en columna. Los registros de desacoplo en la salida de una etapa (parte derecha de cada etapa) son los registros de desacoplo en la entrada de la etapa representada encima de ella (parte izquierda de cada etapa). En la Figura 6.17 cada conjunto de registros de desacoplo se ha etiquetado con el mismo acrónimo que en la Figura 6.16.

Encima de las etapas se representa un ciclo de la señal de reloj, en el cual no se ha representado a escala el intervalo en cada nivel lógico (Figura 6.17). Los registros de desacoplo almacenan, en el flanco ascendente de la señal de reloj, los valores que hay en sus entradas. Estos valores, después de un retardo, son entradas estables de los elementos del camino de datos utilizados en las etapas. Después del retardo de estos elementos, se dispone de nuevos valores estables en la entrada de los registros de desacoplo. En el siguiente flanco ascendente de la señal de reloj los nuevos valores se almacenan en los registros de desacoplo.

En la Figura 6.17 los caminos etiquetados como BHBR, BHS1 y BHS2 son bucles hardware del camino de datos¹². Los cortocircuitos también son bucles hardware.

La salida del decodificador de instrucciones (DEC) es un conjunto de señales. Estas señales se propagan por las etapas para controlar los elementos del camino de datos y el encaminamiento. Para simplificar el dibujo, estas señales no se representan, pero hay que tener en cuenta el retardo del decodificador en la etapa DL y el retardo de los registros de desacoplo al propagar las señales (Figura 6.16).

12. En este bucle solo se muestra un trazo más grueso en la señal de salida del multiplexor mSIC. Ahora bien, cualquier otra señal que determina este valor, en el mismo ciclo, pertenece al bucle hardware.

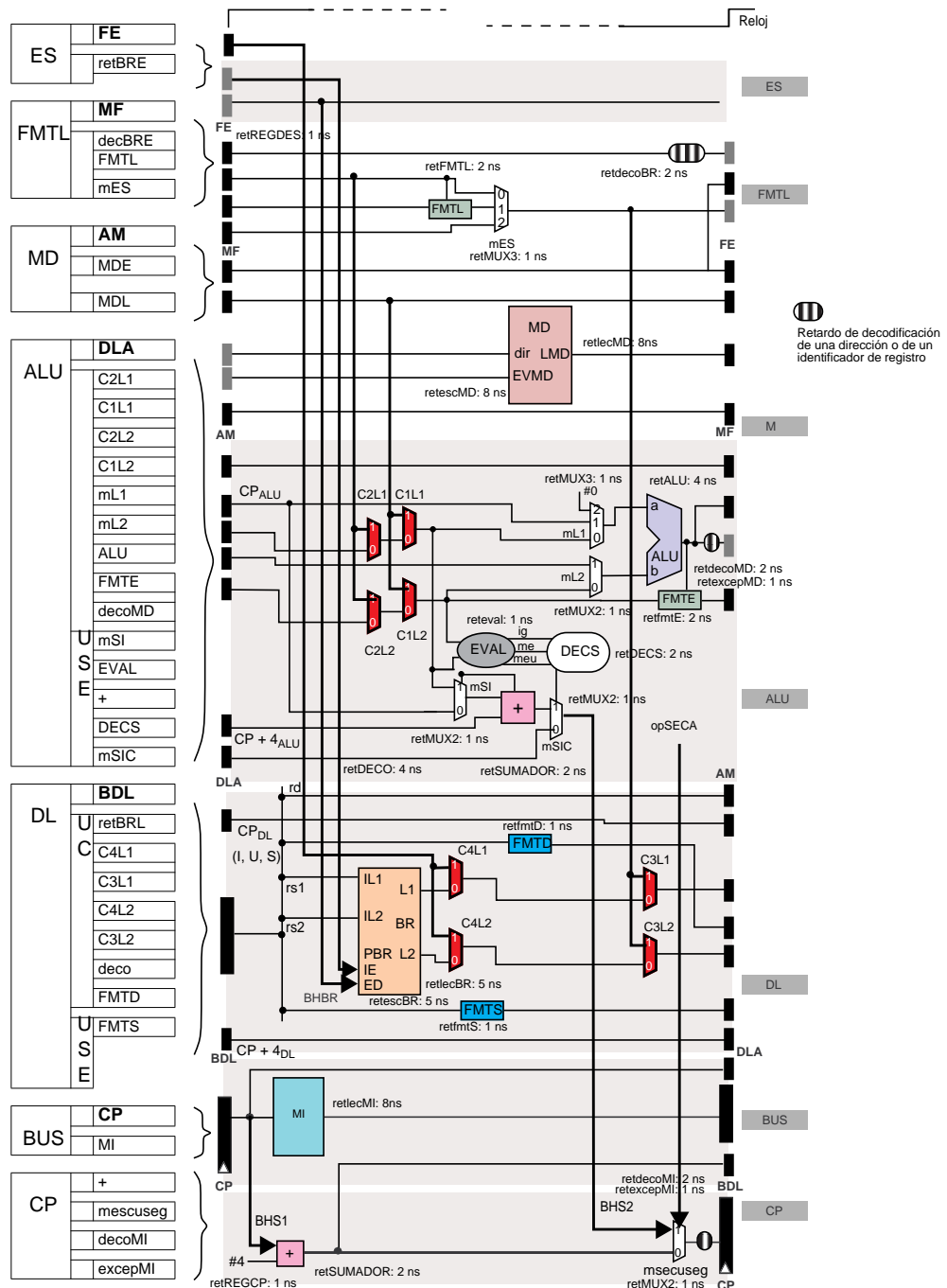


Figura 6.17 Etapas del camino de datos con cortocircuitos y señales en el cronograma de retardos. No se muestra el decodificador ni la LIB.

Para facilitar el cálculo del tiempo de ciclo se utiliza un cronograma donde se representa el retardo de los elementos del camino de datos¹³. En el cronograma se han tenido en cuenta las etapas para agrupar los componentes del camino de datos. En la Figura 6.18 se muestran los acrónimos utilizados para denominar componentes del camino de datos.

Todos deben interpretarse como el retardo del componente. Algunos componentes están incluidos dentro de un módulo que se visualiza. Por ejemplo, el decodificador del identificador del registro cuando se escribe en un registro del banco de registros.

ES	FE	
	retBRE	Retardo en la actualización de un registro del banco de registros
FMTL	MF	Registro de desacoplo de entrada de la etapa FMTL
	decBRE	Decodificador del puerto de escritura del banco de registros. No está representado en las figuras
	FMTL	Módulo para formatear el dato con el cual se actualiza el banco de registros
	mES	Multiplexor en la etapa ES
MD	AM	Registro de desacoplo de entrada de la etapa M
	MDE	Retardo de escritura en memoria
	MDL	Retardo de lectura en memoria
ALU	DLA	Registro de desacoplo de entrada de la etapa ALU
	C2L1/C2L2	Multiplexores de cortocircuito
	C1L1/C1L2	Multiplexores de cortocircuito
	mL1	Multiplexor para seleccionar el operando a que utiliza la etapa ALU
	mL2	Multiplexor para seleccionar el operando b que utiliza la etapa ALU
	ALU	Módulo ALU
	FMTE	Módulo para formatear el dato con el cual se actualiza memoria
	excepMD	Lógica de excepción
	decoMD	Decodificador de la dirección de memoria
	USE	
	mSI	Multiplexor
	EVAL	Unidad de evaluación de la condición
	+	Sumador
	DECS	Módulo de secuenciamiento condicional
	mSIC	Multiplexor
DL	BDL	Registro de desacoplo de entrada de la etapa DL
	UretBRL	Retardo en la lectura de un registro del banco de registros
	C4L1/C4L2	Multiplexores de cortocircuito
	C3L1/C3L2	Multiplexores de cortocircuito
	deco	Retardo del decodificador
	FMTD	Formateador del campo literal en la UC
	FMTS	Formateador del campo literal en la USE
	USE	
	LDD	Dependencias de Datos debido a registros
	LDC	Cortocircuitos
CIB	RD	Riesgo de Datos
	RS	Riesgos de Secuenciamiento
	LGR	Lógica de Gestión
BUS	CP	Registro CP
	MI	Retardo de lectura de la memoria de instrucciones
CP	+	Sumador
	mescuseg	Multiplexor cuya salida es entrada en el registro CP
	decoMI	Decodificador de la dirección de memoria
	excepMI	Lógica de excepción

Figura 6.18 Acrónimos utilizados en los diagramas de retardos.

13. Esta información es necesaria para responder alguna de las preguntas.

Los retardos de los componentes utilizados se detallan en el Apéndice 6.1. Estos retardos no son representativos de un diseño. Solo son de utilidad para efectuar los cálculos de retardo que se soliciten.

Señal de reloj, señal de inicio y elementos de almacenamiento

El registro CP y los registros de desacoplo se actualizan por defecto en el flanco ascendente de la señal de reloj (Figura 6.19). Al activar la señal inicializar (señal PCERO) el registro CP y los registros de desacoplo toman el valor cero. En consecuencia se lee la instrucción que está en la dirección cero de la memoria de instrucciones (MI).

El banco de registros se actualiza en el flanco ascendente de la señal de reloj si el permiso de escritura (PBR) está activado (Figura 6.19). La lectura es asíncrona.

Un banco de la memoria de datos se actualiza cuando el permiso de escritura está activado (PMD), el banco ha sido seleccionado (SCx) y la señal de reloj está en un flanco ascendente (Figura 6.19).

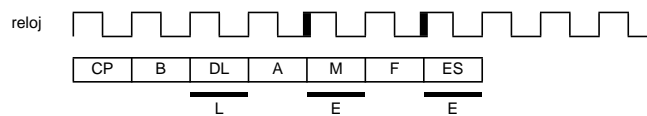


Figura 6.19 Señal de reloj y actualización de los elementos de memorización. El retardo de actualización no está a escala.

Un registro de desacoplo no se actualiza si está activada la señal bloquear (Figura 5.22). De forma síncrona, la salida de un registro de desacoplo toma el valor cero, o la codificación de una nop canónica, cuando la señal inyectar está activada en el flanco ascendente de la señal de reloj.

Simulación

Los ficheros que utiliza Quartus (Rproc_MD_MI_C.qsf y Rproc_MD_MI_C.qpf) se encuentran en el directorio LAB6/PROC_SEG_CORTOS/ENSAMBLADO/QUARTUS.

En los subdirectorios incluidos en LAB6/PROC_SEG_CORTOS solo se incluyen los ficheros modificados respecto a los diseños serie y segmentado (LAB4 y LAB5). En el fichero Rproc_MD_MI_C.qsf se puede observar la ubicación de los ficheros que se utilizan. El árbol de directorios de las prácticas previas del procesador debe ser accesible. Por tanto, los directorios LAB4, LAB5 y LAB6 deben estar en el mismo nivel.

Una vez ha sido abierto el proyecto con Quartus, hay que dar la orden “Processing _> Start -> Start Analysis & Elaboration”. Esta orden elabora el diseño. Una vez elaborado el diseño sin errores, mediante la orden “Tools _> Netlist Viewers -> RTL Viewer” se puede comprobar la elaboración efectuada¹⁴.

El fichero Rproc_MD_MI_C.vhd contiene la descripción en VHDL del procesador descrito. El esquema RTL se muestra en la Figura 6.20. Este fichero está ubicado en el directorio LAB6/PROC_SEG_CORTOS/ENSAMBLADO/CODIGO.

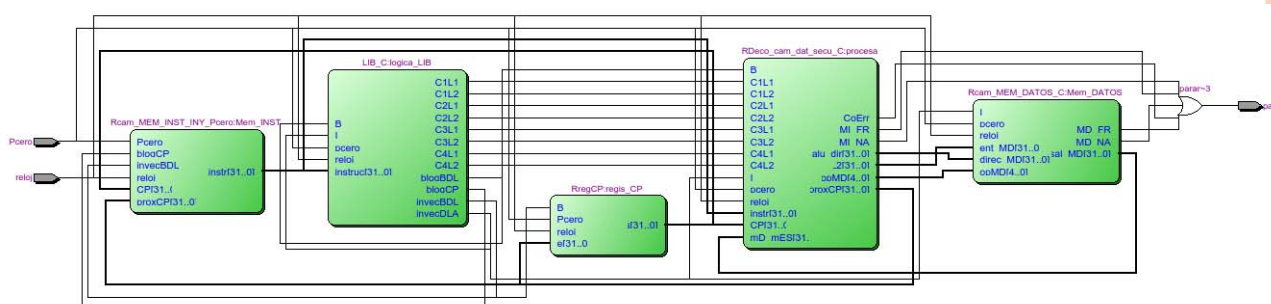


Figura 6.20 Esquema RTL del procesador segmentado con cortocircuitos.

Preparación de la simulación

El fichero de comprobación, denominado prueba_Rproc_MD_MI_C.vhd, se encuentra en el directorio LAB6/PROC_SEG_CORTOS/ENSAMBLADO/PRUEBAS. En este directorio también se encuentra el fichero para formatear las señales que se visualizan en la ventana temporal.

El fichero prueba_Rproc_MD_MI_C.vhd utiliza los mismos genéricos que el fichero parajo de la Práctica 4.

Simulación con ModelSim

Para efectuar una simulación efectúe los mismos pasos descritos en la Práctica 4 (página 305).

Simulación de un programa concreto

Los programas se encuentran en la misma ubicación que en la Práctica 4 (página 306).

14. En los ficheros que se suministra de la lógica LCIB hay que efectuar una descripción de la lógica que se diseña en esta parte de la práctica (Apéndice 6.2). Los ficheros que se suministran contienen la declaración de la interface y la especificación de una arquitectura que no es correcta.

Evolución de las señales del camino de datos

La ventana de tiempo de que dispone ModelSim es útil para visualizar la propagación de una instrucción por las etapas y las interacciones entre ellas.

En la ventana de tiempo se muestran algunas de las señales del camino de datos, las cuales están ordenadas para facilitar la observación.

Descripción funcional de la ventana temporal. En la ventana de tiempos la propagación de una instrucción no se puede observar en una fila, ya que una fila muestra una señal. Por ello, hay que observar distintas filas.

Las señales se han dispuesto de forma que señales contiguas corresponden a la misma etapa o a etapas contiguas. Las señales correspondientes a la etapa CP están en la parte inferior de la ventana de tiempo y las señales correspondientes a la etapa ES están en la parte superior de la ventana de tiempo (parte derecha de la Figura 6.21). En estas condiciones, en un ciclo determinado las señales en la parte inferior de la ventana de tiempo se corresponden con la instrucción más joven y las ubicadas en la parte superior de la ventana de tiempo se corresponden con la instrucción más vieja. Entonces, para observar la propagación de una instrucción hay que moverse hacia la derecha (incrementa el tiempo) y hacia arriba (siguiente etapa). Esto es, en diagonal.

Respecto a los bucles hardware se observan de arriba a abajo en la ventana de tiempo. Notemos que la instrucción más vieja se observa en la parte superior.

En la Figura 6.21 se muestra un ejemplo. En la parte izquierda de la figura se muestra la secuencia de instrucciones. En la parte derecha se muestra la evolución temporal. Para identificar una instrucción se utiliza la dirección de la instrucción como subíndice en cada una de las etapas. Se produce un riesgo de datos y un riesgo de secuenciamiento. También se utiliza un cortocircuito.

		ciclos													
dir.	instrucción	ETAPAS	1	2	3	4	5	6	7	8	9	10	11	12	13
8	lw x3 , 0 (x17)	ES							ES ₈	nop	nop	ES ₁₂	nop	nop	ES ₁₀₀
12	bne x3, x2, 1\$	FMTL						FMTL ₈	nop	nop	FMTL ₁₂	nop	nop	FMTL ₁₀₀	
16	add x26 , x0, x4	M					M ₈	nop	nop	M ₁₂	nop	nop	M ₁₀₀		
20	and x9 , x9, x7	ALU				ALU ₈	nop	nop	ALU ₁₂	nop	nop	ALU ₁₀₀			
24	sub x7 , x21, x31	DL			DL ₈	DL ₁₂	DL ₁₂	DL ₁₂	nop	nop	DL ₁₀₀				
...		BUS		BUS ₈	BUS ₁₂	BUS ₁₆	BUS ₁₆	BUS ₁₆	BUS ₂₀	BUS ₁₀₀					
1\$: 100	slt x20 , x5, x6	CP	CP ₈	CP ₁₂	CP ₁₆	CP ₂₀	CP ₂₀	CP ₂₀	CP ₁₀₀						
Riesgo						RD	RD		RS	RS					

Figura 6.21 Procesador segmentado con cortocircuitos. Observación de la propagación de instrucciones en la ventana de tiempo. En la parte izquierda se muestra una secuencia de instrucciones. En la parte derecha se muestra un esquema de la evolución temporal al interpretar las instrucciones.

Ventana temporal de Modelsim. En la ventana temporal de ModelSim se muestran etiquetas que están asociadas a señales del camino de datos. Estas etiquetas están ordenadas para asociarlas a etapas de forma simple. La asociación entre señales y etapas se muestra en la Figura 6.22.

BR		Banco de registros
FMTL	PE_FMTL	Permiso de escritura en el BR de la instrucción que ocupa la etapa FMTL
	IDE_FMTL	Identificador de registro destino de la instrucción que ocupa la etapa FMTL
M	PE_M	Permiso de escritura en el BR de la instrucción que ocupa la etapa M
	IDE_M	Identificador de registro destino de la instrucción que ocupa la etapa M
	PE_MD_M	Permiso de escritura en MD de la instrucción que ocupa la etapa M
ALU	PE_ALU	Permiso de escritura en el BR de la instrucción que ocupa la etapa ALU
	IDE_ALU	Identificador de registro destino de la instrucción que ocupa la etapa ALU
	opALU	Control de la ALU
	opSEC	Control del secuenciamiento
	opMD	Control de la MD
	me_meu_ig ^a	Salida del circuito que evalúa la condición
	sel_muxSIC	Selección del multiplexor muxSIC (secuenciamiento)
	muxSIC	Salida del multiplexor muxSIC (secuenciamiento)
	selmxsecseg	Selección del multiplexor mxsecseg (secuenciamiento)
	mxsecseg	Próximo valor del registro CP
	C2C1_L1 ^b	Cortocircuitos correspondientes al campo identificador de registro rs1 (IDL1) (C2L1 y C1L1)
	C2C1_L2	Cortocircuitos correspondientes al campo identificador de registro rs2 (IDL2) (C2L2 y C1L2)
DL	PE_DL	Permiso de escritura en el BR de la instrucción que ocupa la etapa DL
	IDE_DL	Identificador de registro destino de la instrucción que ocupa la etapa DL
	C4C3_L1	Cortocircuitos correspondiente al campo identificador de registro rs1 (IDL1) (C4L1 y C3L1)
	C4C3_L2	Cortocircuitos correspondiente al campo identificador de registro rs2 (IDL2) (C4L2 y C3L2)
	val_IDL1	Validez del campo rs1 (IDL1) de la instrucción que ocupa la etapa DL
	IDL1	Campo rs1 (IDL1) de la instrucción que ocupa la etapa DL
	val_IDL2	Validez del campo rs2 (IDL2) de la instrucción que ocupa la etapa DL
	IDL2	Campo rs2 (IDL2) de la instrucción que ocupa la etapa DL
	instruc_DL	Instrucción en la etapa DL. Salida del registro BDL
BUS	instruc_BUS	Salida de MI
	CP	Salida del registro CP
CP	prox_CP	Entrada del registro CP
	ciclo	Contador de ciclos
	reloj	Reloj

a. Las señales están ordenadas, de izquierda a derecha, en el sentido en que se muestran.

b. Las señales C2L1 y C1L1, del camino de datos, que transportan un bit, han sido agrupadas en la señal C2C1_L1 en la ventana de tiempo. El bit más significativo de la señal C2C1_L1 corresponde a la señal C2L1 y el bit menos significativo a la señal C1L1. Para las señales CXC_Y_LZ se ha utilizado la misma codificación.

Figura 6.22 Etiquetas de las señales mostradas en la ventana de tiempo.

En la Figura 6.23 se relacionan otras señales que se muestran en la ventana temporal de Modelsim. Estas señales indican la detección y necesidad de gestión de riesgos de datos y de secuenciamiento. Adicionalmente se relacionan las señales que se utilizan en los registros de desacoplo, en los que son necesarias, para gestionar cada uno de los riesgos.

ALU (Control de riesgos)	bloqCP	Señal de bloqueo del registro CP
	bloqDL	Señal de bloqueo del registro BDL
	inyecBDL	Señal de inyección de nop en el registro BDL
	inyecDLA	Señal de inyección de nop en el registro DLA
Riesgos	RS	Señal de actuación en un riesgo de secuenciamiento
	RD	Señal de actuación en un riesgo de datos

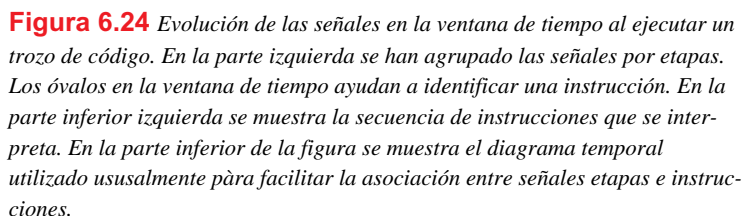
Figura 6.23 Otras señales en la ventana temporal.

En la Figura 6.24 se muestra, en la ventana temporal, la evolución de señales en el camino de datos cuando el procesador interpreta una secuencia de instrucciones. En la parte izquierda de la Figura 6.24 se observan las etiquetas de las señales. Para facilitar la comprensión, ha sido replicado la señal de salida del multiplexor mxsegsec, mostrada en la etapa A_4, en la etapa CP_1. En esta última etapa se denomina prox_CP.

Debajo del diagrama de la ventana de tiempo se muestra el diagrama temporal que se utiliza usualmente al mostrar la interpretación de instrucciones en un procesador segmentado con cortocircuitos. Notemos que en una columna se identifican las etapas en el mismo orden que en la ventana de tiempos. Sin embargo, en el diagrama temporal usual, una instrucción se observa en una fila y en la ventana de tiempo se observa en diagonal (tramas).

En el ciclo 13 (Figura 6.24) se observa la dirección “x00000118” en la salida del registro CP. En la memoria MI se lee la posición de almacenamiento que indica el registro CP. Esta información se indica como instruc_BUS en la ventana temporal.

En el ciclo 14 la instrucción leída de MI, en el ciclo previo, “x00D70733” (add x14, x14, x13) se decodifica. En este ciclo se activa el cortocircuito C3L1 (C4C3_L1 = 01), para obtener el valor que se almacenará en x14. Por otro lado se detecta un riesgo de datos. La interpretación de instrucciones se suspende desde la etapa DL hasta la etapa CP. Además, se inyecta una instrucción nop desde la etapa DL hacia la etapa ALU.



En el ciclo 15 se activan los cortocircuitos C4L1 (C4C3_L1 = 10) y C3L2 (C4C3_L2 = 01), para capturar respectivamente los valores que se almacenarán en los registros x14 y x13. En el banco de registros (BR) se está actualizando el registro x14. En el ciclo previo (14), el identificador de registro destino en la etapa FMTL_6 es IDE_FMTL = 14 y la señal PE_FMTL tiene el valor 1. En el ciclo 15 el identificador de registro destino en la etapa FMTL_6 es IDE_FMTL = 13 y la señal PE_FMTL tiene el valor 1. El identificador de registro destino en la etapa M_5 es IDE_M = 15 y la señal PE_M tiene el valor 1. En la etapa A_4 la señal PE_ALU tiene el valor 0. Los identificadores de registros fuente de la instrucción que ocupa la etapa DL_3 son IDL1 = 14 e IDL2 = 13, siendo los dos valores válidos (val_IDL1 = val_IDL2 = 1).

En el ciclo 16 la etapa DL está ocupada por una instrucción de secuenciamiento condicional (xFEC79AE3, bne x15, x12, xFFA). La instrucción obtiene el operando x15 mediante el cortocircuito C3L1 (C4C3_L1 = 01). En este ciclo se detecta un riesgo de secuenciamiento. La actuación para gestionar el riesgo es inyectar una instrucción nop desde la etapa BUS en este ciclo y en el siguiente ciclo (17).

En el ciclo 17 se establece el secuenciamiento determinado por la instrucción de secuenciamiento condicional (prox_CP = x00000110).

En el ciclo 18 se lee de MI la instrucción "x0007A683" (lw x13, x000(x15)), almacenada en la posición de almacenamiento "x00000110".

En el ciclo 18 finaliza la instrucción nop inyectada desde la etapa DL en el ciclo 14. Notemos que en el ciclo 17, la señal PE_FMTL tiene el valor cero y en el ciclo 16 la señal PE_MD_M también tiene el valor cero.

En los ciclos 21 y 22 finalizan las instrucciones nop inyectadas desde la etapa BUS en los ciclos 16 y 17. Notemos que opALU = 10000, opSEC = 0000 y opMD = 00000 en los ciclos 18 y 19. Además, IDE_ALU es igual a cero (x0).

Información textual de la simulación

En la representación textual se distinguen grupos de información compuestos por dos filas. En la primera fila se representan los cortocircuitos. En la segunda fila se representa información en cada etapa de la segmentación, además de la detección de riesgos. La representación utilizada es idéntica a la utilizada en la Práctica 5.

Representación de cortocircuitos. En la primera fila de un grupo de información se representan los cortocircuitos. Los cortocircuitos están alineados con la columna de la etapa donde se utilizan (destinos, DL y ALU). Las fuentes de los cortocircuitos son las etapas M, FMTL y E.

El orden en el cual se representan los cortocircuitos en cada etapa se muestra en la Figura 6.25 (señal de selección en el multiplexor correspondiente).

Cortocircuitos	C4L2 C3L2	C4L1 C3L1	C2L2 C1L2	C2L1 C1L1
destinos	DL		ALU	

Figura 6.25 Representación del destino de los cortocircuitos en la información textual.

En una etapa destino de un cortocircuito, la codificación es 1 o 0 en función de si se activa o no el cortocircuito. Las fuentes de los cortocircuitos se marcan en las etapas correspondientes (Figura 6.26).

Cortocircuitos	1	2	3	4
fuentes	M	FMTL	E	

Figura 6.26 Representación, en la información textual, de las fuentes de cortocircuito.

En la Figura 6.27 se muestra una representación gráfica. Utilizando flechas con la punta indicando la etapa destino.

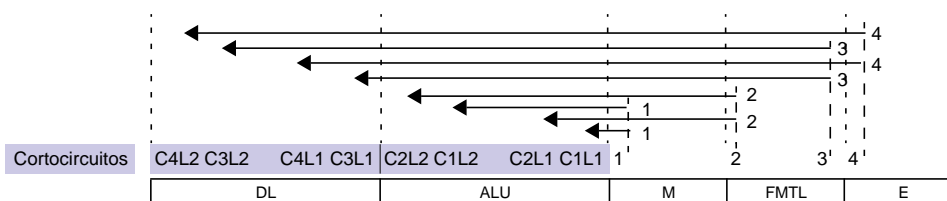


Figura 6.27 Representación gráfica de los cortocircuitos.

En la Figura 6.28 se muestra una secuencia de instrucciones. La instrucción “add x14,...” detecta un riesgo de datos en el ciclo 14. Desde la etapa DL se inyecta una instrucción nop (DLB). En este ciclo ha sido activado el cortocircuito C3L1.

En el ciclo 15 no se detecta ningún riesgo. Se activan los cortocircuitos C3L2 y C4L1.

En el ciclo 17, la instrucción de secuenciamiento “bne ...” establece el secuenciamiento. En el siguiente ciclo la salida del registro CP es el valor determinado por la instrucción de secuenciamiento.

Ciclo	prox_CP	BUS	DL (L.E.)	L2	L1	L2	L1	1	2	3	4	RD	RS	
				c43 c43	c21 c21	c43 c43	c21 c21							
13	0000011C	00D70733	addi x15, x15, x004	00 10 DL	00 00 A	00 10 R	00 00 R	4	E	RD: 0	RS: 0			
14	00000120	FEC79AE3	add x14, x14, x13	00 01 DLB	00 00 A	00 01 M	00 00 R	3	E	RD: 1	RS: 0			
15	00000120	FEC79AE3	add x14, x14, x13	01 10 DL	00 10 nop	00 10 R	00 10 F	3	4	E	RD: 0	RS: 0		C3L2, C4L1. Fuentes: FMTL y E Destino: DL
16	00000124	00000013	bne x15, x12, xFFA	00 01 DL	00 00 A	00 01 nop	00 00 R	3	E	RD:	RS: 1			C3L1. Fuente: FMTL. Destino: DL Riesgo de secuenciamiento
17	00000110	00000013	addi x0, x0, x000	00 00 DLN	00 00 A	00 00 R	00 00 nop	E	RD: 0	RS: 1				RS: Nop en la etapa DL Establece el secuenciamiento
18	00000114	0007A683	addi x0, x0, x000	00 00 DLN	00 00 AN	00 00 -	00 00 R	nop	RD: 0	RS: 0				RS: Nop en la etapa DL
19	00000118	00478793	lw x13, x000(x15)	00 00 DL	00 00 AN	00 00 RN	00 00 -	E	RD: 0	RS: 0				
20	0000011C	FEC79AE3	addi xx15, x15, x004	00 00 DL	00 00 A	00 00 RN	00 00 RN	-	RD: 0	RS: 0				
21	00000120	FEC79AE3	add x14, x14, x13	00 00 DLB	00 00 A	00 00 M	00 00 RN	EN	RD: 1	RS: 0				
22	00000120	FEC79AE3	add x14, x14, x13	01 00 DL	00 00 nop	00 00 R	00 00 F	3	EN	RD: 0	RS: 0			C3L2. Fuente: FMTL. Destino: DL

Figura 6.28 Representación textual de la interpretación de una secuencia de instrucciones.

Modificación de la unidad de secuenciamiento

Por ejecución de una instrucción de secuenciamiento entendemos el establecimiento de la dirección de la instrucción que debe interpretarse después de la instrucción de secuenciamiento. En la tabla de la Figura 6.29 se muestra el conjunto de instrucciones de secuenciamiento.

FI	Nemotécnico y especificación	Descripción	Especificación semántica
I	jalr rd, rs1, imm	Incondicional indexado y almacenamiento de la dirección de la instrucción mas cuatro Incondicional relativo y almacenamiento de la dirección de la instrucción mas cuatro	$CP^V = (rs1^V + (I-Imm))_{[31:1]}$ & '0' y $rd^V = CP^V + 4$
J	jal rd, imm	Incondicional relativo y almacenamiento de la dirección de la instrucción mas cuatro	$CP^V = CP^V + (J-Imm \ll 1)$ y $rd^V = CP^V + 4$
B	BRANCH	condicional	if $(rs1^V \text{COND } rs2^V)$ then $CP^V = CP^V + (B-Imm \ll 1)$; else $CP^V = CP^V + 4$

$J-Imm = \text{ExtSig}(\text{inst}(31), \text{inst}(19:12), \text{inst}(20), \text{inst}(30:21))$
 $I-Imm = \text{ExtSig}(\text{inst}(31:20))$

$\ll 1$: el imm es múltiplo de 2
&: indica concatenación
COND: condición especificada en la instrucción

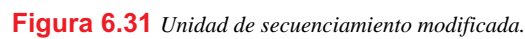
Figura 6.29 Instrucciones de secuenciamiento.

Modificación de la segmentación de la USE

En este diseño, dada una instrucción de secuenciamiento, se pierden dos ciclos debidos al riesgo de secuenciamiento.

16. Notemos que este cálculo también se utiliza en las instrucciones de secuenciamiento condicional.

El control de los multiplexores mSI, mSIC y msecuseg no se modifica. Por otro lado, el sumador ubicado en la entrada “1” del multiplexor mSI se especializa (simplifica), ya que no es necesaria la señal de control.



Reducción de la latencia en la instrucción de secuenciamiento “jal”

432 Práctica 6

[illegible]

En la Figura 6.33 se muestra una secuencia de instrucciones con una instrucción “jal”. El número de ciclos perdidos por riesgo de secuenciamiento es uno¹⁸.

18. Notemos que en una instrucción “jal” no se puede detectar un riesgo de datos. Por otro lado, en la instrucción que se descarta no se puede detectar ningún tipo de riesgo.

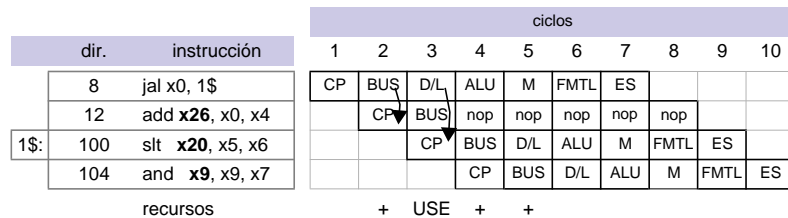


Figura 6.33 Diagrama temporal cuando se reduce la latencia en instrucciones “jal”.

La lógica de cortocircuitos e interbloqueos (LCIB) también debe modificarse. En la Figura 6.34 se resaltan con trazo grueso los módulos modificados y la señal añadida (SInConDL).

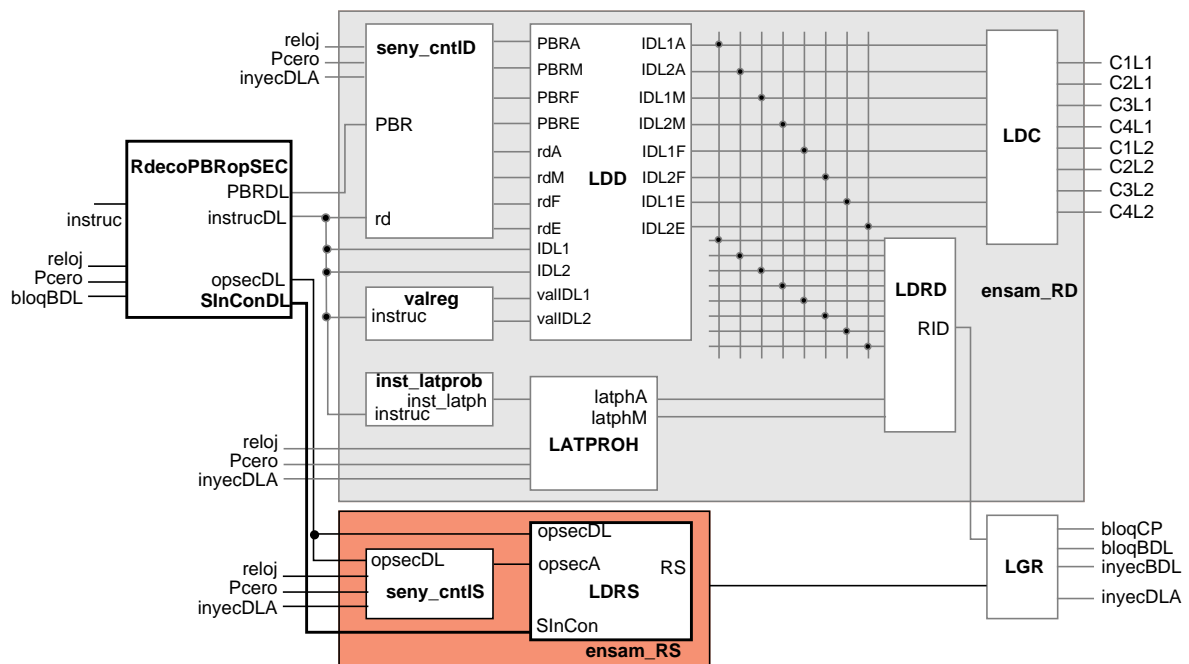


Figura 6.34 Modificación en la LCIB. Una instrucción “jal” modifica el secuenciamiento en la etapa DL.

En la Figura 6.35 se muestra la modificación en el módulo “RdecoPBRopSEC”. Cuando la etapa DL está ocupada por una instrucción “jal” se activa la señal SInConDL. Esta señal se utiliza en el módulo LDRS para indicar riesgo de secuenciamiento.

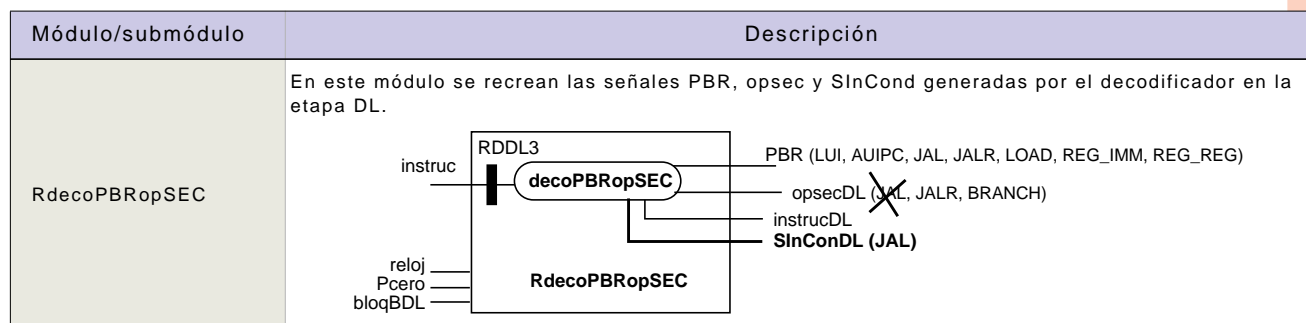


Figura 6.35 Reducción de la latencia de la instrucción “jal”. Módulo *RdecoPBRopSEC*. En negrita se muestra la señal añadida y tachada la instrucción eliminada en la codificación de *opsecDL*.

Simulación. En el Apéndice 6.10 y el Apéndice 6.11 se indican los ficheros utilizados para la simulación y la organización en directorios.

Predicción de sentido en instrucciones de secuenciamiento condicional

En una instrucción de secuenciamiento condicional (BRANCH, Figura 6.29) se evalúa una condición y se calcula una dirección destino. El cálculo de la dirección destino se efectúa en la etapa DL y la condición se evalúa en la etapa ALU (Figura 6.32). El número de ciclos perdidos por riesgo de secuenciamiento es dos.

Para reducir los ciclos perdidos, una alternativa es utilizar predicción en el resultado de la evaluación de la condición (predicción de sentido). Posteriormente hay que comprobar si la predicción es correcta y en caso contrario, hay que recuperar el flujo correcto de interpretación, lo cual determina que hay que anular las instrucciones predichas (instrucciones que se interpretan en función de la predicción)¹⁹.

Seguidamente se analizan dos diseños donde se utiliza predicción fija para todas las instrucciones de secuenciamiento condicional. En el primer diseño se predice seguir en secuencia y en el segundo diseño se predice modificar el secuenciamiento.

Posteriormente se analiza un tercer diseño donde la predicción también es fija, pero dada una instrucción de secuenciamiento, la predicción depende del signo del literal. Cuando el literal es positivo se predice seguir en secuencia. En caso contrario se predice modificar el secuenciamiento²⁰.

19. La debilidad del mecanismo es que, para efectuar la recuperación, puede que sean necesarios más ciclos que cuando no se predice. Por tanto, la predicción debe ser correcta un número muy significativo de veces.

Predicción fija de seguir en secuencia.

No es necesario modificar el camino de datos, ya que por defecto la USE genera direcciones en secuencia mientras no se establezca, mediante una instrucción de secuenciamiento, un valor en la entrada del registro CP.

En la Figura 6.36 se muestra el caso en el cual la predicción es correcta. La etapa ALU se etiqueta como CPre para indicar que se comprueba la predicción.

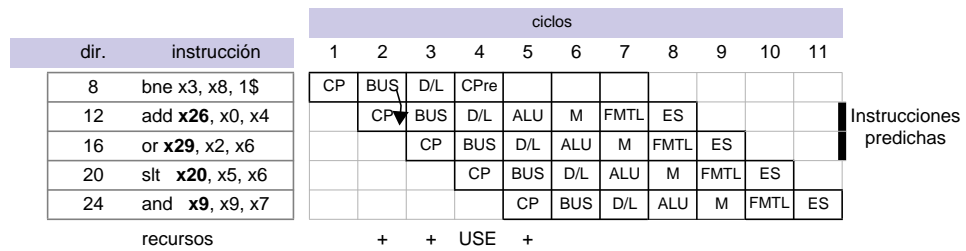


Figura 6.36 Predicción de seguir en secuencia. Acierto en la predicción.

En la Figura 6.37 se muestra el caso en el cual la predicción es errónea. Hay que anular las dos instrucciones predichas. En este diseño es factible, ya que ninguna de ellas ha modificado el estado del procesador (excepto el CP, el cual se modifica al recuperarse) o la memoria.

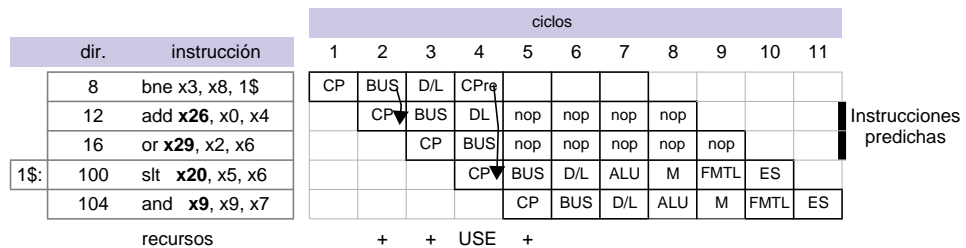


Figura 6.37 Predicción de seguir en secuencia. Error en la predicción.

La anulación de las instrucciones mal predichas requiere inyectar instrucciones nop desde las etapas DL y BUS en el ciclo de recuperación.

En resumen, cuando la predicción es correcta no se pierden ciclos y cuando es errónea se pierden dos ciclos.

Interacciones. Al interpretar una instrucción de secuenciamiento condicional se puede detectar un riesgo de datos (Figura 6.10). No se produce interacción con una modificación del secuenciamiento, ya que, al interpretar la instrucción de secuenciamiento condicional, se predice seguir en secuencia.

20. Cuando el literal es negativo se infiere que estamos en una secuencia repetitiva de instrucciones (bucle).

Por otro lado, en el ciclo de recuperación de un error de predicción se pueden producir interacciones con otros riesgos o predicciones en las instrucciones predichas. Es factible que la primera instrucción predicha (ocupa la etapa DL en el ciclo de recuperación):

- Detecte un riesgo de datos en el mismo ciclo. Este riesgo debe ignorarse. Por tanto hay que modificar la gestión de riesgos.
- Sea una instrucción “jal” de secuenciamiento incondicional. Por tanto, es necesario actuar en el control del multiplexor mrell, ya que no hay que tenerla en cuenta.
- Sea una instrucción de secuenciamiento condicional y se predice seguir en secuencia. Por tanto, se debe ignorar la predicción.

Control de la USE. Para añadir, en el diseño de la Figura 6.32, la predicción de seguir en secuencia es necesario modificar el control del camino de datos de la USE (Figura 6.39) y la LCIB.

La lógica de control del camino de datos que debe modificarse es el módulo DECS (Apéndice 6.13). En este módulo debe tenerse en cuenta que, en una instrucción de secuenciamiento condicional, se ha predicho seguir en secuencia y hay que activar una señal de error de predicción (ErPred, Figura 6.38), si ésta es errónea²¹. La señal ErPred es utilizada para modificar el control del multiplexor “msecuseg” (módulo Ersec, Figura 6.39) y el control del multiplexor mrell (módulo Errell).

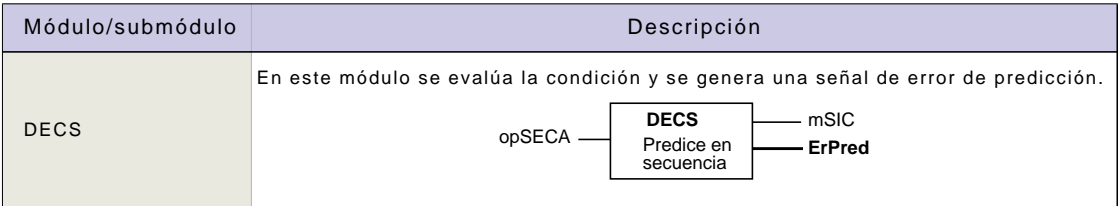


Figura 6.38 Predicción de seguir en secuencia. Modificación del módulo DECS.

En el módulo Ersec hay que discernir entre instrucciones de secuenciamiento condicional y la instrucción “jalr”. En las primeras solo se modifica el secuenciamiento cuando se produce un error de predicción²².

En el módulo Errell hay que anular el secuenciamiento que pueda establecer una instrucción “jal” mal predicha (primera instrucción predicha en Figura 6.37).

21. Esta señal solo se activa si la instrucción es de secuenciamiento condicional y se ha producido un error en la predicción. Esto es, está validada.
 22. Recordemos que la señal ErPred está validada.

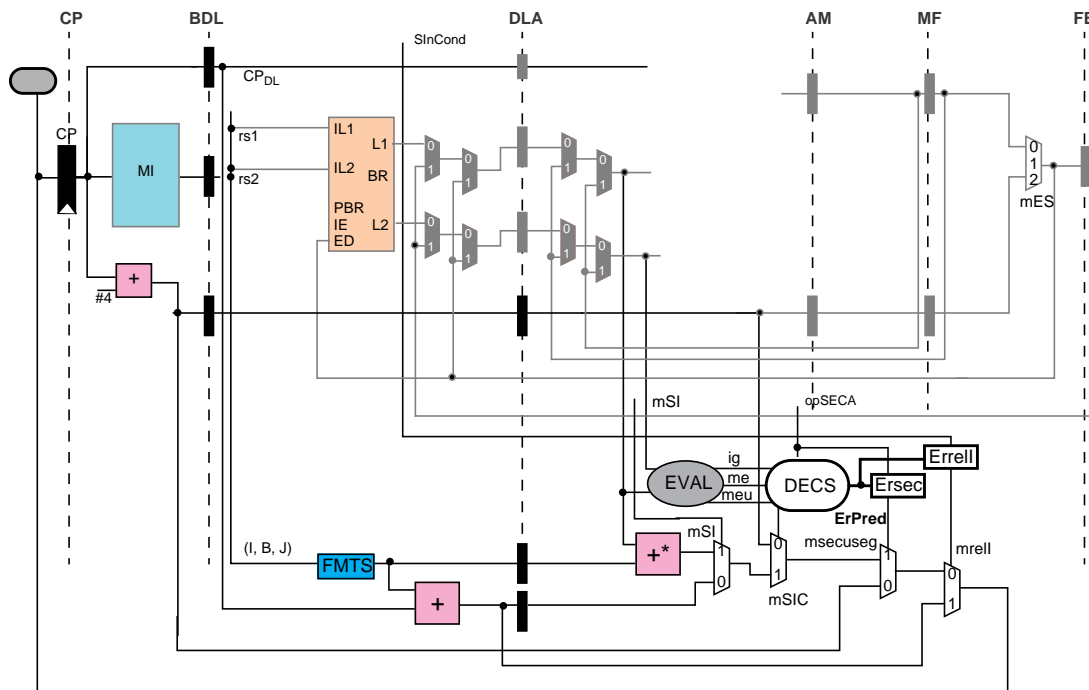


Figura 6.39 Predicción de seguir en secuencia. Modificación del control del camino de datos. Los módulos modificados o añadidos, así como las señales, se muestran en trazo grueso (respecto al esquema de la Figura 6.32).

Respecto del módulo LCIB, la señal ErPred debe ser entrada de éste módulo. Además, en el módulo “RdecoPBRopSEC” debe generarse una señal que indique que se está interpretando una instrucción de secuenciamiento condicional (SConDL, Figura 6.40).

Módulo/submódulo	Descripción
RdecoPBRopSEC	<p>En este módulo se recrean las señales PBR, opsec y SInCond generadas por el decodificador en la etapa DL. Adicionalmente se genera una señal que indica que la etapa DL está ocupada por una instrucción de secuenciamiento condicional.</p>

Figura 6.40 Predicción de seguir en secuencia. Modificación del módulo RdecoPBRopSEC. En negrita se muestra la señal añadida.

En la Figura 6.42 se resaltan con trazo grueso los módulos modificados y las señales añadidas en la LCIB.

El módulo “seny_cntS” genera, a partir de la señal SConDL, la señal SConA, la cual indica que una instrucción de secuenciamiento ocupa la etapa ALU (Figura 6.41).

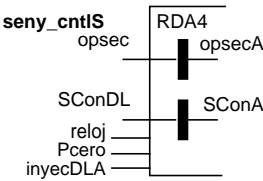
Módulo/submódulo	Descripción
seny_cntIS	<p>En este submódulo se recrea la propagación de las señales opsec y SConDL hacia la etapa ALU (opsecA, SConA), la cual indica si la instrucción que ocupa la etapa ALU es de secuenciamiento y si es una instrucción de secuenciamiento condicional.</p> 

Figura 6.41 Módulo que recrea señales del camino de datos y genera señales específicas para la predicción en instrucciones de secuenciamiento condicional.

El módulo LDRS solo debe activar la señal RS cuando exista un riesgo de secuenciamiento. Notemos que, en las instrucciones de secuenciamiento condicional, al predecir seguir en secuencia no existe riesgo de secuenciamiento como tal.

El módulo LGR tiene como entrada la señal ErPred. Por tanto, además de los riesgos de datos y de secuenciamiento, debe de gestionar los errores de predicción.

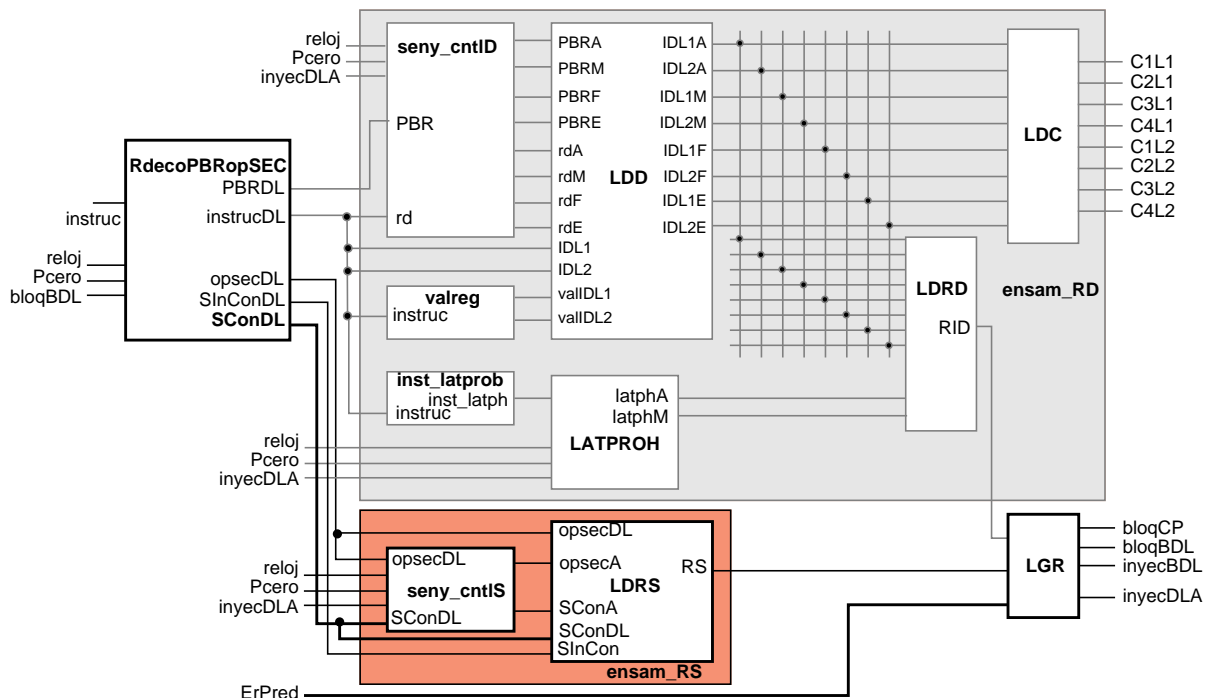


Figura 6.42 Modificación en la LCIB. En una instrucción de secuenciación condicional se predice seguir en secuencia.

Simulación. En el Apéndice 6.14 y el Apéndice 6.15 se indican los ficheros utilizados para la simulación y la organización en directorios.

Predicción fija de modificar el secuenciamiento.

No es necesario modificar el camino de datos, ya que por defecto la USE calcula, en instrucciones de secuenciación condicional, la dirección destino en la etapa DL y ésta puede utilizarse para actualizar el CP (Figura 6.32).

En la Figura 6.43 se muestra el caso en el cual la predicción es correcta. La etapa ALU se etiqueta como CPre para indicar que se comprueba la predicción. Debido al retardo en el cálculo de la dirección destino, hay que descartar la siguiente instrucción que se busca en secuencia²³. Para ello, se inyecta una instrucción nop desde la etapa BUS, en el ciclo que se efectúa la predicción (etapa DL). Esto es, se produce una penalización de un ciclo.

23. Se denomina retardo de búsqueda, ya que no se dispone de la dirección destino de salto.

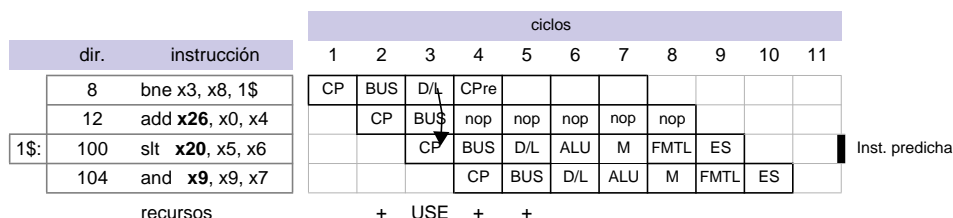


Figura 6.43 Predicción de modificar el secuenciamiento. Acierto en la predicción.

En la Figura 6.44 se muestra el caso en el cual la predicción es errónea. Hay que anular la instrucción predicha. En este diseño es factible, ya que no ha modificado el estado del procesador (excepto el CP, el cual se modifica al recuperarse) o la memoria.

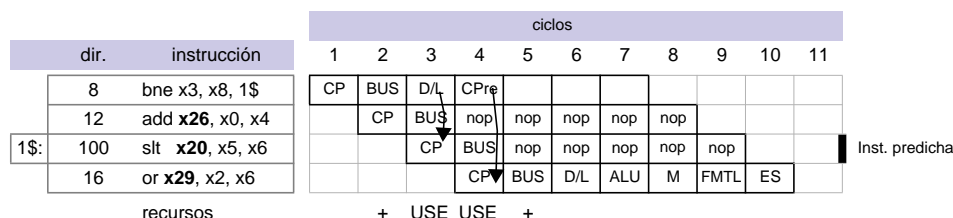


Figura 6.44 Predicción de modificar el secuenciamiento. Error en la predicción.

La anulación de la instrucción predicha requiere inyectar una instrucción “nop” desde la etapa BUS hacia la etapa DL en el ciclo de recuperación²⁴.

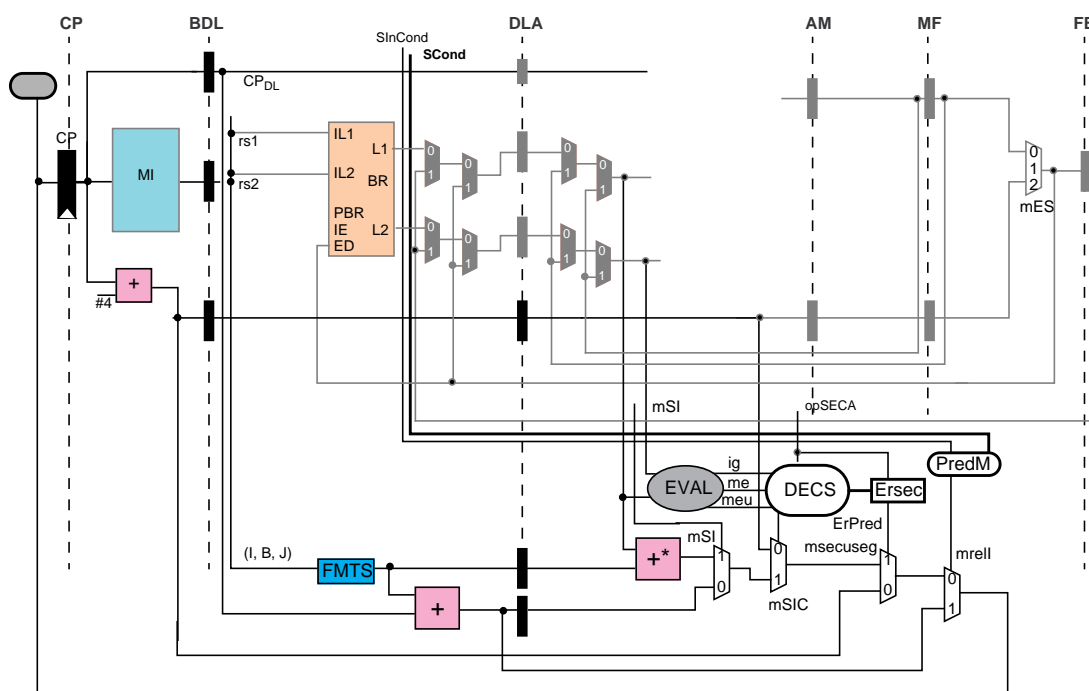
En resumen, cuando la predicción es correcta se pierde un ciclo y cuando es errónea se pierden dos ciclos.

Interacciones. Al interpretarse una instrucción de secuenciamiento condicional se puede detectar un riesgo de datos (Figura 6.10). Como se predice modificar el secuenciamiento, existe una interacción con el riesgo de secuenciamiento. Este último debe ignorarse mientras exista riesgo de datos.

Por otro lado, en el ciclo de recuperación de un error de predicción no se pueden detectar riesgos, ya que la instrucción predicha se anula antes de decodificarse.

24. Notemos que también se puede inyectar una instrucción “nop” desde DL en el mismo ciclo. Si se efectúa esta acción, el control para anular instrucciones, tanto si se predice modificar el secuenciamiento como seguir en secuencia, es el mismo.

El módulo decodificador debe activar una señal al decodificar una instrucción de secuenciamiento condicional (SCond, Apéndice 6.17). Esta señal se utiliza en el control del multiplexor mre11 (módulo PredM, Figura 6.45).



En el módulo DECS debe tenerse en cuenta que, en una instrucción de secuenciación condicional, se ha predicho modificar el secuenciamiento. Entonces, hay que activar una señal de error de predicción (ErPred) si la predicción es errónea. Esta señal se utiliza para modificar el control del multiplexor msecuseq (módulo Ersec, Figura 6.46).

El módulo LDRS solo debe activar la señal RS cuando exista un riesgo de secuenciamiento. Notemos que al predecir modificar el secuenciamiento se produce un riesgo de secuenciamiento, debido al retardo de búsqueda.

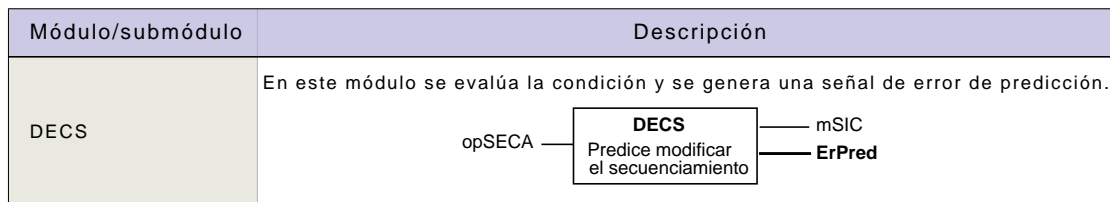


Figura 6.46 Predicción de modificar el secuenciamiento. Modificación del módulo DECS.

Respecto del módulo LCIB, el esquema es el de la Figura 6.42. Hay que diseñar los módulos mostrados en trazo grueso.

Simulación. En el Apéndice 6.19 y el Apéndice 6.20 se indican los ficheros utilizados para la simulación y la organización en directorios.

Predicción de sentido en función del signo del literal

La predicción se efectúa en la etapa DL y el secuenciamiento se establece en función de la predicción²⁵. Cuando el signo del literal es positivo se predice seguir en secuencia²⁶. En caso contrario se predice modificar el secuenciamiento.

Camino de datos de la USE. Es necesario modificar el camino de datos, además de control del mismo, para extraer información de la instrucción. Para ello se utiliza el módulo Predictor, el cual determina el signo del literal²⁷. La salida de este módulo, en el ciclo correspondiente (PredDL, PredA), es entrada de los módulos DECS y PredS (Figura 6.48).

Control de la USE. El diseño para cada una de las posibilidades de predicción ha sido efectuado previamente. En este nuevo diseño deben acomodarse los dos diseños previos. Para ello se utiliza la señal PredDL, que es la salida del predictor (Apéndice 6.22 y Apéndice 6.23).

En el módulo DECS debe tenerse en cuenta que, en una instrucción de secuenciamiento condicional, la predicción está determinada por la señal PredA. Entonces, hay que activar una señal de error de predicción (ErPred) si la predicción es errónea. Esta señal se utiliza para modificar el control del multiplexor msecuseg (módulo Ersec, Figura 6.46).

25. El signo del literal no se conoce hasta la etapa DL.

26. Recordemos que por defecto se utiliza secuenciamiento implícito. En consecuencia, si se predice seguir en secuencia, en la etapa DL, no hay que efectuar ninguna actuación.

27. La señal de salida de este módulo no es necesario que esté validada. La información se valida en los módulos donde se utiliza.

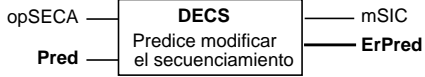
Módulo/submódulo	Descripción
DECS	<p>En este módulo se evalúa la condición y se genera una señal de error de predicción en función de la predicción efectuada.</p> 

Figura 6.47 Predicción en función del sentido del literal. Modificación del módulo DECS.

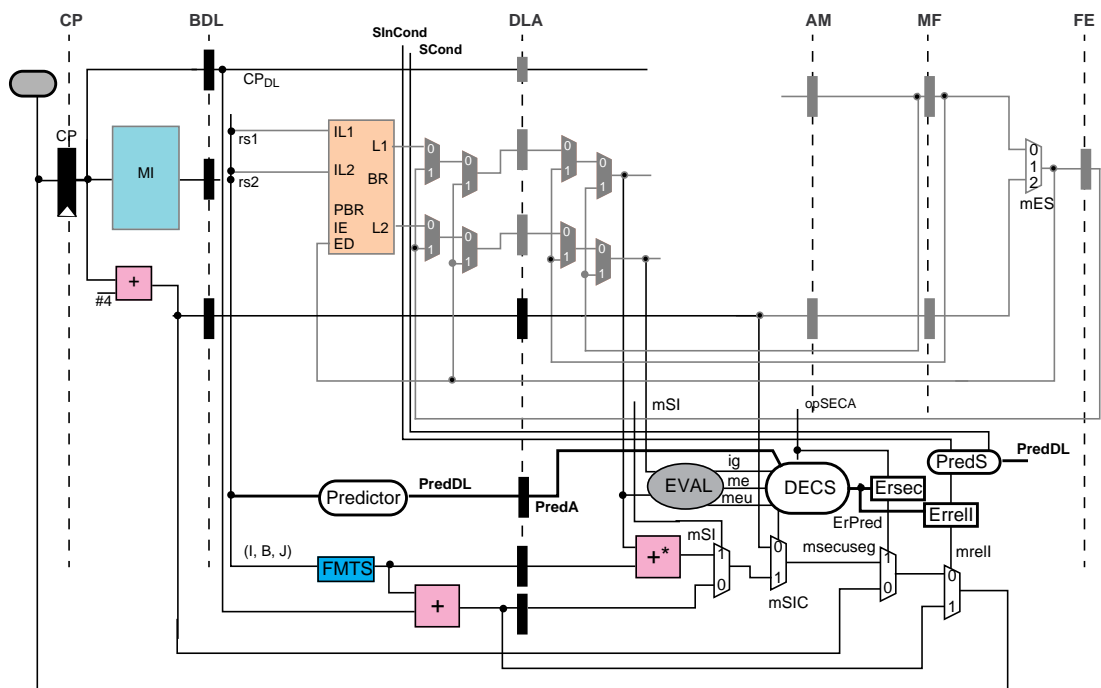


Figura 6.48 Predicción de secuenciamiento en función del signo del literal de la instrucción. Modificación del camino de datos y del control del mismo. Los módulos modificados o añadidos, así como las señales, se muestran en trazo grueso (respecto al esquema de la Figura 6.32).

Respecto del módulo LCIB, el esquema es el de la Figura 6.49. La señal Pred es entrada del módulo LDRS. Esta señal se utiliza para determinar el posible riesgo de secuenciamiento, cuando una instrucción de secuenciamiento condicional ocupa la etapa DL²⁸. En la Figura 6.49 se indica el módulo que hay que modificar, respecto al diseño previo, en trazo grueso.

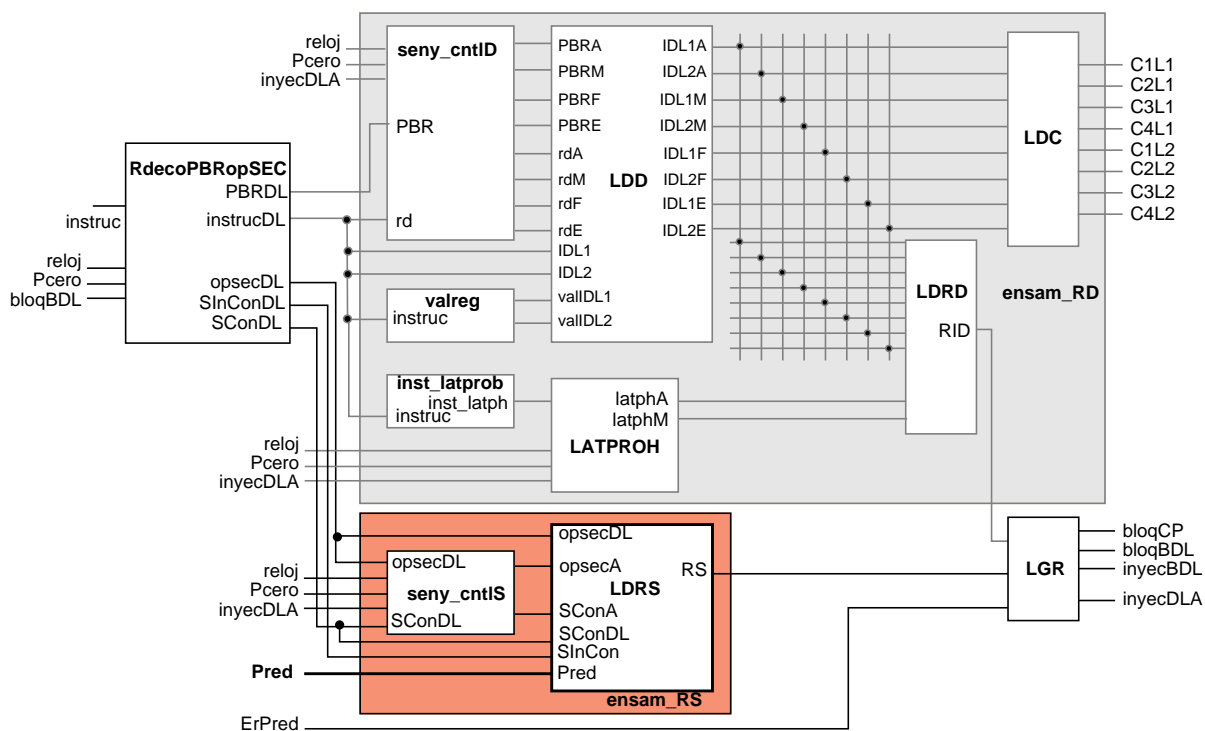


Figura 6.49 Predicción de secuenciamiento en función del signo del literal de la instrucción. Modificación en la LCIB. En una instrucción de secuenciamiento condicional se predice en función del signo del literal.

Simulación. En el Apéndice 6.24 y el Apéndice 6.25 se indican los ficheros utilizados para la simulación y la organización en directorios.

28. La señal Pred indica la predicción efectuada. Recordemos que el módulo LDRS solo debe activar la señal RS cuando exista un riesgo de secuenciamiento.

Apéndice 6.1: Retardos

En el “package retardos_cntl_seg_C” se declaran los retardos de los elementos de la lógica de interbloques del procesador segmentado con cortocircuitos (Figura 6.50).

```
LDC    constant retLDC: time := 1 ns;
LDRD   constant retLDRD_C: time := 2 ns;    cuando se utilizan cortocircuitos
```

Figura 6.50 Retardos de los elementos de la lógica de interbloques.

Los retardos del resto de elementos están indicados en la Figura 5.55.

Apéndice 6.2: Organización de los ficheros: árbol de directorios

El árbol de directorios, desde el directorio raíz, de esta parte de la práctica es mimético al árbol de la práctica de los procesador serie y segmentado (LAB4 y LAB5), excluyendo el directorio que contiene el diseño de los cortocircuitos (CORTOCIRCUITOS). En los subdirectorios miméticos a los de LAB4 y LAB5, incluidos en LAB6, solo se incluyen los directorios que contienen ficheros modificados respecto a los diseño serie y segmentado (LAB4 y LAB5).

En el directorio CONTROL_SEG_CORTOS están incluidos todos los ficheros que se utilizan en la lógica de cortocircuitos e interbloqueos (LCIB, Figura 6.13), excepto los que no se modifican de la Práctica 5 (por ejemplo ENSAM_RS, Apéndice 5.2)²⁹. En el directorio CORTOCIRCUITOS están ubicados los ficheros que contienen las descripciones de los multiplexores que se utilizan para seleccionar entre los cortocircuitos y otros caminos.

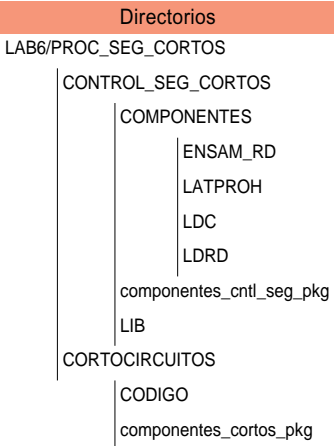


Figura 6.51 Arbol de directorios del control de la segmentación y cortocircuitos.

Edición de ficheros. En los directorios LATPROH, LDC y LDRD (Figura 6.51) se incluyen los ficheros que deben utilizarse para diseñar parte de la lógica LCIB. Los ficheros contienen la declaración de la interface³⁰, la cual se corresponde con la especificada en el fichero “componentes_control_seg_C_pkg.vhd”, ubicado en el directorio “componentes_cntl_seg_pkg”. La instanciación de estos elementos se efectúa en los ficheros incluidos en el directorio ENSAM_RD.

29. Tenga en cuenta que la especificación del módulo LDD es la que ha sido diseñada en la Práctica 5. Por tanto, entre otros, el directorio, de la Práctica 5, donde está ubicado el fichero debe ser accesible.
30. Ninguna de las interfaces debe modificarse.

En la Figura 6.52 se muestra la ubicación de “packages” que se utilizan en el programa de prueba para mostrar la evolución de la simulación. El directorio que se muestra sustituye al mismo directorio de la Práctica 5.

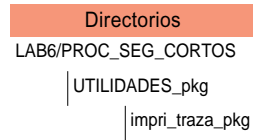


Figura 6.52 Directorio que contiene “packages” utilizados en el programa de prueba.

Apéndice 6.3: Documentación

La documentación ha sido generada utilizando la herramienta Doxygen. El fichero que hay que abrir con un navegador es "index.html" ubicado en el directorio LAB6/PROC_SEG_CORTOS/documentacio/html.

Las pestañas que se muestran son autoexplicativas.

Dado un módulo se muestra un grafo de dependencias jerárquicas con otros módulos. Pulsando en un nodo del grafo se observan el módulo o módulos que agrupa. También se puede acceder al código VHDL que describe al módulo.

Apéndice 6.4: Implementación de la activación de cortocircuitos y gestión de riesgos de datos

En el procesador es necesario, en primer lugar, detectar una dependencia entre instrucciones que se ejecutan concurrentemente o detectar una instrucción de secuenciamiento. Posteriormente hay que actuar. La actuación puede ser: utilizar un cortocircuito, bloquear o suspender la interpretación de instrucciones (riesgo de datos) o descartar instrucciones (riesgo de secuenciamiento).

En el diseño que se utiliza, todas las detecciones de dependencias que afectan a una instrucción se efectúan cuando la instrucción ocupa la etapa DL.

Notemos que la actuación no tiene porqué efectuarse en el mismo ciclo que la detección. El instante de actuación depende de la especificación del lenguaje máquina y de la microarquitectura utilizada. Por ejemplo, en la microarquitectura utilizada la mitad de los cortocircuitos están al principio de la etapa ALU.

En estas condiciones los módulos de detección son los mismos que en la Práctica 5, añadiendo módulos (inst_latprob, LATPROH, Figura 6.13) que detecten la imposibilidad o no disponibilidad de cortocircuito en algunos casos.

El módulo LDC (Figura 6.13) es el encargado de activar los cortocircuitos, si es el caso³¹. Este módulo, a partir de la información generada en un ciclo determinado por el módulo LDD, puede actuar en el ciclo en que se genera la información o en el siguiente ciclo, según las dependencias detectadas. Notemos que hay cortocircuitos en la etapa DL y en la etapa ALU.

El módulo LDRD (Figura 6.13) junto con los módulos inst_latprob y LATPROH, además del módulo LDD, detecta riesgos de datos. El módulo inst_latprob detecta instrucciones cuya latencia de ejecución es mayor que un ciclo o instrucciones cuyo resultado no está disponible después de un ciclo o más ciclos debido al diseño de cortocircuitos utilizado en la microarquitectura (jalr, jal). El módulo LATPROH determina los ciclos en los cuales no está disponible el resultado mediante un cortocircuito, si es el caso.

En el diseño de la lógica de riesgos de datos no se tiene en cuenta que, en algunos de los diseños de la microarquitectura, en particular, de la lógica USE, no se puede producir algún riesgo de datos debido a la gestión efectuada en un riesgo de secuenciamiento. Esta decisión permite utilizar la misma lógica de detección de riesgos de datos en todos los diseños que se realizan. Por otro lado, no afecta al rendimiento, ya que si no se puede producir un riesgo no se detecta.

31. Notemos que la señal de control de los cortocircuitos se determina independientemente de si hay que bloquearse por un riesgo de datos.

Caracterización de registros fuente y destino

Dado el formato de instrucciones “I” desagregaremos tres tipos de instrucciones para la caracterización y enumeración en la utilización de registros del banco de registros. Para identificarlas utilizaremos como sufijo el acrónimo de la instrucción.

- **I_jalr**: Para el cálculo de la dirección de la siguiente instrucción se utiliza el contenido de un registro y un operando especificado en la propia instrucción. Almacena la dirección de la instrucción mas cuatro en un registro especificado en la instrucción.
- **I_lw**: Un operando se especifica en la propia instrucción. El otro operando y el resultado en registros. El resultado son 4 bytes.
- **I_lh_b**: Un operando se especifica en la propia instrucción. El otro operando y el resultado en registros. El resultado son 2 bytes (h) o un byte (b).
- **I_sw_h_b³²**: Un operando se especifica en la propia instrucción. Los otros dos operandos en registros. El resultado, que se almacena en memoria, son 2 bytes (h) o un byte (b).

Para la caracterización de riesgos de datos y cortocircuitos utilizaremos 3 dimensiones ortogonales: a) registros que se actualizan, registros que se leen y distancia entre instrucciones interpretadas. Para especificar el tipo de instrucción utilizaremos el acrónimo del formato de instrucción o los tres casos desagregados, seguido de un punto y el sufijo del registro fuente o destino, según el caso. Cuando en una clasificación no se muestra un caso desagregado (I con sufijo, I_*) se considera incluido en el caso I.

- Registro que se actualiza.

Registro destino						
U.rd	J.rd	I.rd	R.rd	I_jalr.rd	I_lw.rd	I_lh_b.rd

- Registros que se leen.

Registros fuente							
B.rs1	B.rs2	I.rs1	R.rs1	R.rs2	S.rs1	S.rs2	I_sw_h_b.rs2

- Distancia entre instrucciones: 1, 2, 3 y 4.

32. Instrucciones sw_h o sw_b

Cortocircuitos disponibles y riesgos de datos

En la Figura 6.16 se muestra el camino de datos con cortocircuitos. El destino de los cortocircuitos está en las etapas DL y ALU. Las fuentes de los cortocircuitos son las etapas M, FMTL y ES.

En la Figura 6.53 se muestran los cortocircuitos disponibles en un diagrama temporal.

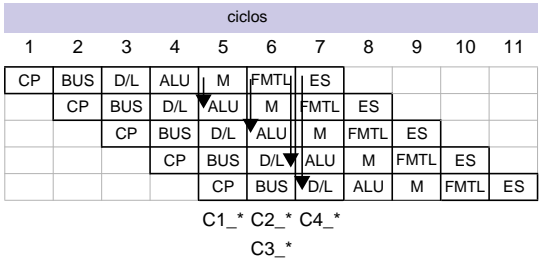


Figura 6.53 Cortocircuitos disponibles en el camino de datos.

En la Figura 6.54 se muestran otros cortocircuitos que, por razones de causalidad, pueden ser factibles en el camino de datos. El acrónimo “a” en la fila inferior indica que la productora es una instrucción I_jalr y la consumidora puede ser cualquier instrucción. El acrónimo “b” indica que la productora es I_jalr o I_w y la consumidora es cualquier instrucción. El acrónimo “c” indica que la productora es I_h_b y la consumidora es I_sw_h_b³³.

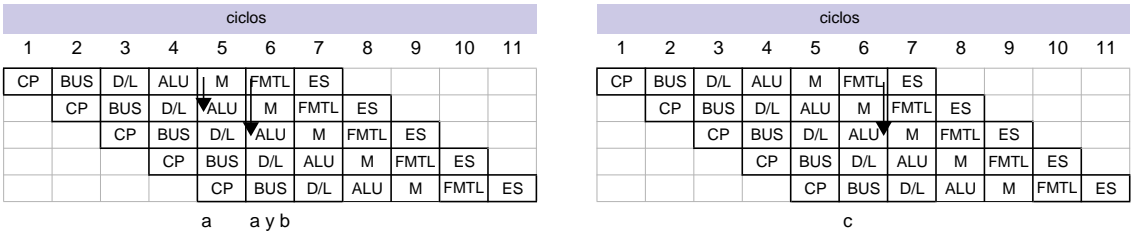


Figura 6.54 Otros cortocircuitos, no disponibles, pero factibles.

En las siguiente tablas se muestran los cortocircuitos disponibles dado un tipo de instrucción productora (fila), un tipo de instrucción consumidora (columna) y la distancia (Figura 6.55, Figura 6.56, Figura 6.57 y Figura 6.58).

33. En una implementación habría que analizar con detalle la influencia en el tiempo de ciclo de este cortocircuito.

- Distancia 1. En las casillas de la tabla se muestran los cortocircuitos disponibles.

distancia = 1 Registros fuente (consumidora)								
Reg. destino	B.rs1	B.rs2	I.rs1	R.rs1	R.rs2	S.rs1	S.rs2	I_sw_h_b.rs2
U.rd	C1L1	C1L2	C1L1	C1L1	C1L2	C1L1	C1L2	C1L2
J.rd	*	*	*	*	*	*	*	*
I.rd	C1L1	C1L2	C1L1	C1L1	C1L2	C1L1	C1L2	C1L2
R.rd	C1L1	C1L2	C1L1	C1L1	C1L2	C1L1	C1L2	C1L2
I_jalr.rd	*	*	*	*	*	*	*	*
I_lw.rd	X	X	X	X	X	X	X	X
I_lh_b.rd	X	X	X	X	X	X	X	X

Figura 6.55 Cortocircuitos a distancia 1.

En el caso de una marca X, se pierden ciclos. Cuando la marca es * también se pierden ciclos, pero no se perderían si se hubiera añadido otro cortocircuito, lo cual es factible por causalidad (Figura 6.54).

- Distancia 2. En las casillas de la tabla se muestran los cortocircuitos disponibles.

distancia = 2 Registros fuente (consumidora)								
Reg. destino	B.rs1	B.rs2	I.rs1	R.rs1	R.rs2	S.rs1	S.rs2	I_sw_h_b.rs2
U.rd	C2L1	C2L2	C2L1	C2L1	C2L2	C2L1	C2L2	C2L2
J.rd	*	*	*	*	*	*	*	*
I.rd	C2L1	C2L2	C2L1	C2L1	C2L2	C2L1	C2L2	C2L2
R.rd	C2L1	C2L2	C2L1	C2L1	C2L2	C2L1	C2L2	C2L2
I_jalr.rd	*	*	*	*	*	*	*	*
I_lw.rd	*	*	*	*	*	*	*	*
I_lh_b.rd	X	X	X	X	X	X	X	#

Figura 6.56 Cortocircuitos a distancia 2

En el caso de una marca #, se pierden ciclos. Ahora bien, se podría añadir un cortocircuito, teniendo en cuenta la causalidad, aunque es posible que se incremente el tiempo de ciclo (Figura 6.54).

- Distancia 3. En las casillas de la tabla se muestran los cortocircuitos disponibles.

distancia = 3 Registros fuente (consumidora)							
Reg. destino	B.rs1	B.rs2	I.rs1	R.rs1	R.rs2	S.rs1	S.rs2
U.rd	C3L1	C3L2	C3L1	C3L1	C3L2	C3L1	C3L2
J.rd	C3L1	C3L2	C3L1	C3L1	C3L2	C3L1	C3L2
I.rd	C3L1	C3L2	C3L1	C3L1	C3L2	C3L1	C3L2
R.rd	C3L1	C3L2	C3L1	C3L1	C3L2	C3L1	C3L2

Figura 6.57 Cortocircuitos a distancia 3

- Distancia 4. En las casillas de la tabla se muestran los cortocircuitos disponibles.

distancia = 4		Registros fuente (consumidora)					
Reg. destino	B.rs1	B.rs2	I.rs1	R.rs1	R.rs2	S.rs1	S.rs2
U.rd	C4L1	C4L2	C4L1	C4L1	C4L2	C4L1	C4L2
J.rd	C4L1	C4L2	C4L1	C4L1	C4L2	C4L1	C4L2
I.rd	C4L1	C4L2	C4L1	C4L1	C4L2	C4L1	C4L2
R.rd	C4L1	C4L2	C4L1	C4L1	C4L2	C4L1	C4L2

Figura 6.58 Cortocircuitos a distancia 4.

Apéndice 6.5: Simulación: modificación de la segmentación de la USE

Los ficheros que utiliza Quartus (Rproc_MD_MI_C_ModSecu.qsf y Rproc_MD_MI_C_ModSecu.qpf) se encuentran en el directorio LAB6/PROC_SEG_ModSecu/ENSAMBLADO/QUARTUS.

En los subdirectorios incluidos en LAB6/PROC_SEG_ModSecu solo se incluyen los ficheros modificados respecto a los diseños serie, segmentado y segmentado con cortocircuitos (LAB4, LAB5 y LAB6/PROC_SEG_CORTOS). En el fichero Rproc_MD_MI_C_ModSecu.qsf se puede observar la ubicación de los ficheros que se utilizan. El árbol de directorios de las prácticas previas del procesador debe ser accesible. Entre los ficheros necesarios están los editados para diseñar el control del procesador segmentado sin y con cortocircuitos.

Una vez ha sido abierto el proyecto con Quartus, hay que dar la orden "Processing _> Start -> Start Analysis & Elaboration". Esta orden elabora el diseño. Una vez elaborado el diseño sin errores, mediante la orden "Tools _> Netlist Viewers -> RTL Viewer" se puede comprobar la elaboración efectuada.

El fichero Rproc_MD_MI_C_ModSecu.vhd contiene la descripción en VHDL del procesador descrito. Este fichero está ubicado en el directorio LAB6/PROC_SEG_ModSecu/ENSAMBLADO/CODIGO^{34 35}

Preparación de la simulación

El fichero de comprobación, denominado prueba_Rproc_MD_MI_C_ModSecu.vhd, se encuentra en el directorio LAB6/PROC_SEG_ModSecu/ENSAMBLADO/PRUEBAS. En este directorio también se encuentra el fichero para formatear las señales que se visualizan en la ventana temporal.

El fichero prueba_Rproc_MD_MI_C_ModSecu.vhd utiliza los mismos genéricos que el fichero parejo de la Práctica 4.

Simulación con ModelSim

Para efectuar una simulación realice los mismos pasos descritos en la Práctica 4 (página 305).

34. La organización del proyecto en directorios se muestra en el Apéndice 6.6.

35. En el Apéndice 6.7 se indica una posibilidad para analizar el diseño y su estructura utilizando un navegador.

Simulación de un programa concreto

Los programas se encuentran en la misma ubicación que en la Práctica 4 (página 306).

En la ventana de tiempo se muestran las mismas señales que antes de modificar la segmentación de la USE.

Los ficheros de resultados que se obtengan deben ser idénticos a los que se obtienen cuando se simula con el procesador segmentado con cortocircuitos.

Información textual de la simulación

En la representación textual se muestra la misma información que antes de modificar la segmentación de la USE.

Apéndice 6.6: Organización de los ficheros: modificación de la segmentación de la USE

El árbol de directorios, desde el directorio raíz, de esta parte de la práctica es mimético a los árboles de las prácticas de los procesadores serie, segmentado sin y con cortocircuitos. En este árbol se incluyen los directorios que contienen el diseño de la USE.

En los subdirectorios miméticos a los de los diseños anteriores solo se incluyen los directorios que contienen ficheros modificados respecto a los diseños de los procesadores previos³⁶.

El árbol de directorios que contiene los ficheros VHDL modificados, que describen el procesador con la modificación de la segmentación de la USE, se muestra en la Figura 6.59 y en la Figura 6.60.

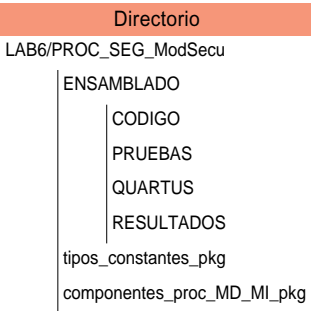


Figura 6.59 *Modificación de la segmentación de la USE. Directorio que contiene los directorios y ficheros que describen el ensamblado del procesador con las memorias.*

36. Tenga en cuenta que se utilizan algunas especificaciones de la logica LCIB diseñada en la Práctica 5 y las diseñadas, hasta llegar a este diseño, en esta práctica. A partir de ahora, cuando una modificación utilice parte de diseños previos se indicará como otros proyectos QUARTUS del procesador.

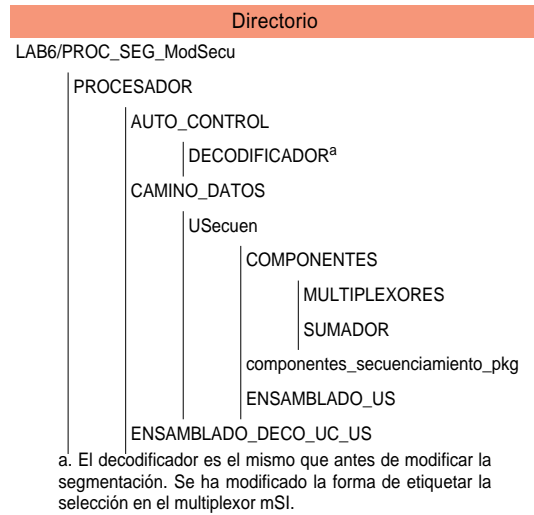


Figura 6.60 *Modificación de la segmentación de la USE. Arbol de directorios del procesador.*

Apéndice 6.7: Documentación: modificación de la segmentación de la USE

La documentación ha sido generada utilizando la herramienta Doxygen. El fichero que hay que abrir con un navegador es "index.html" ubicado en el directorio LAB6/PROC_SEG_ModSecu/documentacio/html.

Las pestañas que se muestran son autoexplicativas.

Dado un módulo se muestra un grafo de dependencias jerárquicas con otros módulos. Pulsando en un nodo del grafo se observan el módulo o módulos que agrupa.

También se puede acceder al código VHDL que describe a un módulo.

Apéndice 6.8: Reducción de la latencia al ejecutar la instrucción “jal”: control del encaminamiento por instrucción

En la tabla de la Figura 6.61 se describen las señales de encaminamiento para cada instrucción (minúsculas) o subconjunto de instrucciones (mayúsculas). Además, se indican las unidades funcionales utilizadas. Esta tabla es una ampliación de la tabla de la Figura 4.90, se ha añadido una columna, correspondiente al multiplexor mrell, y se han modificado las señales mSIC y opSEC.

CoOp	bits 4 . . . 2						PBR	lnmD	mL1	mL2	mES	lnmS	mSI	mSIC	SlmCond	opALU	opMD	opSEC
bits 6 . . . 5	000	001	010	011	100	101												
00	LOAD						si	I	reg	inm	MD			seculmpl	no	si	si	no
					OP_IMM		si	I	reg	inm	ALU			seculmpl	no	si	no	no
						auipc	si	U	CP	inm	ALU			seculmpl	no	si	no	no
01	STORE						no	S	reg	inm				seculmpl	no	si	si	no
					OP		si		reg	reg	ALU			seculmpl	no	si	no	no
						lui	si	U	cero	inm				seculmpl	no	si	no	no
10																		
11	BRANCH						no		reg	reg		B	rel	secu	no	no	no	si
		jalr					si		reg		ret	I	indx	secuMod	no	no	no	si
				jal			si				ret	J	rel	seculmpl	si	no	no	no
					SYSTEM													

Figura 6.61 Reducción de la latencia de la instrucción “jal”. Señales de encaminamiento para las instrucciones e indicación de las unidades funcionales utilizadas.

Cuando en una casilla no hay ningún acrónimo se está indicando que no es importante el valor.

En la Figura 6.62 se muestra la modificación del autómata de control de la Figura 4.49.

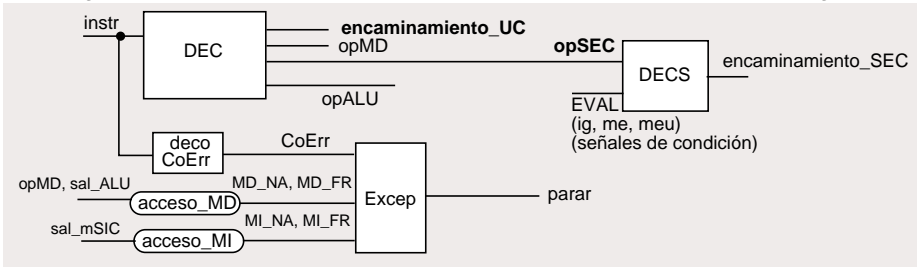


Figura 6.62 Reducción de la latencia de la instrucción “jal”. Modificación en los módulos en el autómata de control. DEC: decodificación y gestión de las señales de error. DECS: segundo nivel de decodificación y control para las instrucciones de secuenciamiento condicional y gestión de las señales de condición. decoCoerr: identificación de secuencias de bits que no pertenecen al lenguaje máquina implementado.

En la Figura 6.63 se muestra la modificación en la organización del decodificador DEC de la Figura 4.50.

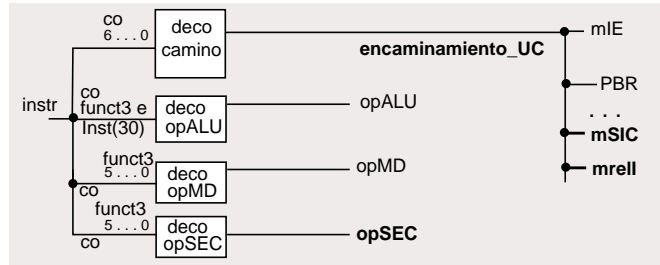


Figura 6.63 Reducción de la latencia de la instrucción “jal”. Modificación en la organización del decodificador DEC. El módulo decocamino utiliza los bits del código de operación de una instrucción para generar las señales de encaminamiento. Los otros módulos utilizan los bits mostrados para generar las señales de control de las unidades funcionales.

Apéndice 6.9: Reducción de la latencia al ejecutar la instrucción “jal”: señales de control del secuenciamiento

En la tabla de la Figura 6.64 se muestran las señales de control del secuenciamiento para cada instrucción (módulo DEC de la Figura 6.62, módulo decoopSEC de la Figura 6.63). Esta tabla es una modificación de la tabla de la Figura 4.110.

Co bits 6 . . . 5	bits 4 . . . 2						opSEC				mSIC
	000	001	010	011	100	101	3	2	1	0	
00	LOAD				OP_IMM	auipc	no	secuImplicto			secuImpl
01	STORE				OP	lui	no	secuImplicto			secuImpl
10											
11	BRANCH						si	secuCond			secu
		jalr					si	secuIncond			secuMod
				jal			no	secuImplicto			
					SYSTEM						

Figura 6.64 Señal opSEC y control del multiplexor de secuenciamiento mSIC. Los acrónimos secuImpl/secuMod se codifican como 0/1. Una casilla en blanco indica que el valor no importa.

La instrucción “jal” se ejecuta cuando está ubicada en la etapa DL. En consecuencia, se modifican las señales mSIC y opSEC (Figura 6.63). Al indicar en opSEC que no es una instrucción de secuenciamiento se selecciona la entrada cero del multiplexor msecuseg (Figura 6.32), lo cual determina secuenciamiento implícito cuando la instrucción ocupa la etapa ALU.

Apéndice 6.10: Simulación: reducción de la latencia al ejecutar la instrucción “jal”

Los ficheros que utiliza Quartus (Rproc_MD_MI_C_ModSecu_rell.qsf y Rproc_MD_MI_C_ModSecu_rell.qpf) se encuentran en el directorio LAB6/PROC_SEG_ModSecu_rell/ENSAMBLADO/QUARTUS

En los subdirectorios incluidos en LAB6/PROC_SEG_ModSecu_rell solo se incluyen los ficheros modificados respecto a todos los diseños previos del procesador (LAB4/PROC_SERIE, LAB5/PROC_SEGMENTADO, LAB6/PROC_SEG_CORTOS y LAB6/PROC_SEG_ModSecu). En el fichero Rproc_MD_MI_C_ModSecu_rell.qsf se puede observar la ubicación de los ficheros que se utilizan.

Una vez ha sido abierto el proyecto con Quartus, hay que dar la orden “Processing _> Start -> Start Analysis & Elaboration”. Esta orden elabora el diseño. Una vez elaborado el diseño sin errores, mediante la orden “Tools _> Netlist Viewers -> RTL Viewer” se puede comprobar la elaboración efectuada³⁷.

El fichero Rproc_MD_MI_C_ModSecu_rell.vhd contiene la descripción en VHDL del procesador descrito. Este fichero está ubicado en el directorio LAB6/PROC_SEG_ModSecu_rell/ENSAMBLADO/CODIGO^{38 39}

Preparación de la simulación

El fichero de comprobación, denominado prueba_Rproc_MD_MI_C_ModSecu_rell.vhd, se encuentra en el directorio LAB6/PROC_SEG_ModSecu_rell/ENSAMBLADO/PRUEBAS. En este directorio también se encuentra el fichero para formatear las señales que se visualizan en la ventana temporal.

El fichero prueba_Rproc_MD_MI_C_ModSecu_rell.vhd utiliza los mismos genéricos que el fichero parejo de la Práctica 4.

Simulación con ModelSim

Para efectuar una simulación realice los mismos pasos descritos en la Práctica 4.

37. En uno de los ficheros que se suministra de la lógica LCIB hay que efectuar una descripción de la lógica que se diseña en esta parte de la práctica (Apéndice 6.11). El fichero que se suministra contiene la declaración de la interface y una especificación de una arquitectura que no es correcta.

38. La organización del proyecto en directorios se muestra en el Apéndice 6.11.

39. En el Apéndice 6.12 se indica una posibilidad para analizar el diseño y su estructura utilizando un navegador.

Simulación de un programa concreto

Los programas se encuentran en la misma ubicación que en la Práctica 4.

En la ventana de tiempo se muestran algunas de las señales del camino de datos, las cuales están ordenadas para facilitar la observación.

Ventana temporal de Modelsim. En la ventana temporal de ModelSim se muestran etiquetas que están asociadas a señales del camino de datos. Estas etiquetas están ordenadas para asociarlas a etapas de forma simple.

En particular, en el grupo de señales asociadas a la etapa DL se ha añadido la señal denominada “S_InCond_jal”, que indica que la etapa DL está ocupada por una instrucción “jal”. En la etapa ALU se ha añadido la salida del multiplexor mrell y la señal de selección de este multiplexor (Figura 6.65).

ALU	PE_ALU	Permiso de escritura en el BR de la instrucción que ocupa la etapa ALU
	IDE_ALU	Identificador de registro destino de la instrucción que ocupa la etapa ALU
	opALU	Control de la ALU
	opSEC	Control del secuenciamiento
	opMD	Control de la MD
	me_meu_ig ^a	Salida del circuito que evalúa la condición
	sel_muxSIC	Selección del multiplexor muxSIC (secuenciamiento)
	muxSIC	Salida del multiplexor muxSIC (secuenciamiento)
	selmxsecseg	Selección del multiplexor mxsecseg (secuenciamiento)
	mxsecseg	Próximo valor del registro CP
	sel_mrell	Selección del multiplexor mrell
	muxrell	Salida del multiplexor mrell
	C2C1_L1 ^b	Cortocircuitos correspondientes al campo identificador de registro rs1 (IDL1) (C2L1 y C1L1)
	C2C1_L2	Cortocircuitos correspondientes al campo identificador de registro rs2 (IDL2) (C2L2 y C1L2)
DL	PE_DL	Permiso de escritura en el BR de la instrucción que ocupa la etapa DL
	IDE_DL	Identificador de registro destino de la instrucción que ocupa la etapa DL
	C4C3_L1	Cortocircuitos correspondiente al campo identificador de registro rs1 (IDL1) (C4L1 y C3L1)
	C4C3_L2	Cortocircuitos correspondiente al campo identificador de registro rs2 (IDL2) (C4L2 y C3L2)
	val_IDL1	Validez del campo rs1 (IDL1) de la instrucción que ocupa la etapa DL
	IDL1	Campo rs1 (IDL1) de la instrucción que ocupa la etapa DL
	val_IDL2	Validez del campo rs2 (IDL2) de la instrucción que ocupa la etapa DL
	IDL2	Campo rs2 (IDL2) de la instrucción que ocupa la etapa DL
	S_Incond_jal	Una instrucción de secuenciamiento “jal” ocupa la etapa DL
	instruc_DL	Instrucción en la etapa DL. Salida del registro BDL

a. Las señales están ordenadas, de izquierda a derecha, en el sentido en que se muestran.

b. Las señales C2L1 y C1L1, del camino de datos, que transportan un bit, han sido agrupadas en la señal C2C1_L1 en la ventana de tiempo. El bit más significativo de la señal C2C1_L1 corresponde a la señal C2L1 y el bit menos significativo a la señal C1L1. Para las señales CXC_Y_LZ se ha utilizado la misma codificación.

Figura 6.65 Reducción de la latencia al ejecutar la instrucción “jal”.

Señales en la ventana temporal de Modelsim.

Información textual de la simulación

En la representación textual es la misma que antes de reducir la latencia de ejecución de la instrucción “jal”, excepto para la instrucción “jal”.

Representación de las instrucciones. La representación de una instrucción “jal” se modifica (Figura 6.66).

Instrucción	Etapas					Comentario
instrucción jal	DL	R	R	R	E	Modifica el secuenciamiento en la etapa DL

Figura 6.66 Acrónimos utilizadas en las etapas al progresar una instrucción “jal”.

Por otro lado, al ejecutar la instrucción “jal” se observa un ciclo de riesgo de secuenciamiento, en lugar de los dos ciclos que se observaban previamente.

Apéndice 6.11: Organización de los ficheros: reducción de la latencia al ejecutar la instrucción “jal”

El árbol de directorios, desde el directorio raíz, de esta parte de la práctica es mimético al árbol de la parte de la práctica donde se modifica la segmentación de la USE.

En los subdirectorios miméticos solo se incluyen los directorios que contienen ficheros modificados respecto al diseño donde se modifica la segmentación de la USE.

El árbol de directorios que contiene los ficheros VHDL modificados, que describen el procesador con reducción de la latencia de ejecución de una instrucción “jal”, se muestra en la Figura 6.67 y en la Figura 6.68

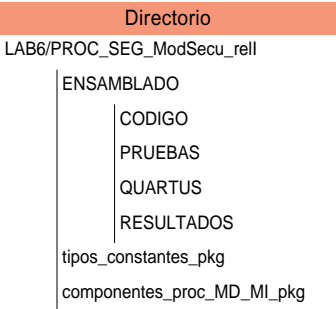


Figura 6.67 Reducción de la latencia de la instrucción “jal”. Directorio que contiene los directorios y ficheros que describen el ensamblado del procesador con las memorias.

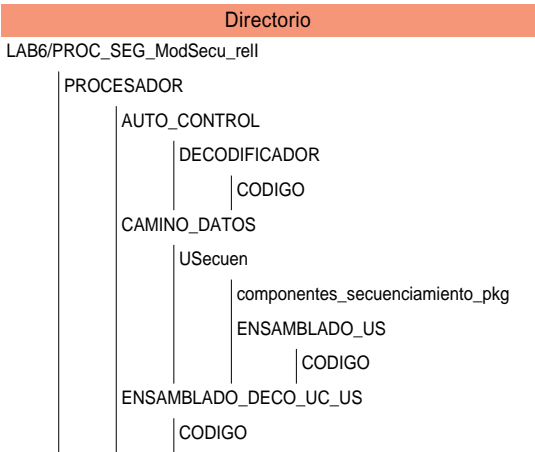


Figura 6.68 Reducción de la latencia de la instrucción “jal”. Arbol de directorios del procesador.

En los directorios CONTROL_SEGMENTACION y CONTROL_SEG_CORTOS están incluidos todos los ficheros que se utilizan en la lógica de interbloqueos, excepto los que no se modifican de las prácticas o diseños previos (LIB, Figura 6.34). En la Figura 6.69 se muestran los directorios donde residen ficheros modificados y el fichero que debe editarse⁴⁰.

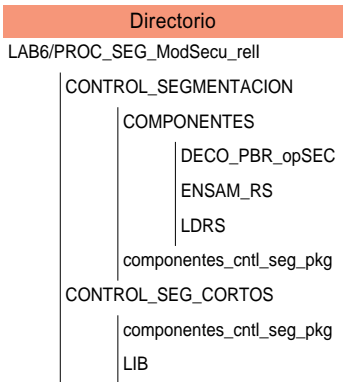


Figura 6.69 Reducción de la latencia de la instrucción “jal”. Arbol de directorio del control de la segmentación y del control de cortocircuitos..

Edición de ficheros. En el directorio LDRS/CODIGO (Figura 6.69) se incluye el fichero que debe utilizarse para diseñar parte de la lógica LCIB (LDRS_ModSecu_rell.vhd). El fichero contiene la declaración de la interface⁴¹, la cual se corresponde con la especificada en el fichero “componentes_control_seg_ModSecu_rell_pkg.vhd”, ubicado en el directorio “componentes_cntl_seg_pkg”. La instanciación de este elemento se efectúa en el fichero incluido en el directorio ENSAM_RS.

40. Tenga en cuenta que los ficheros que se utilizan de otros proyectos QUARTUS del procesador deben ser accesibles. .
41. Ninguna de las interfaces debe modificarse.

Apéndice 6.12: Documentación: reducción de la latencia al ejecutar la instrucción “jal”

La documentación ha sido generada utilizando la herramienta Doxygen. El fichero que hay que abrir con un navegador es “index.html” ubicado en el directorio LAB6/PROC_SEG_ModSecu_rell/documentacio/html.

Las pestañas que se muestran son autoexplicativas.

Dado un módulo se muestra un grafo de dependencias jerárquicas con otros módulos. Pulsando en un nodo del grafo se observan el módulo o módulos que agrupa.

También se puede acceder al código VHDL que describe a un módulo.

Apéndice 6.13: Predicción de seguir en secuencia: señales de control del secuenciamiento condicional

En la tabla de la Figura 6.70 se muestran las condiciones que se evalúan en las instrucciones BRANCH, el control del multiplexor mSIC (acrónimo secu) y la señal ErPred. Esta tabla es una modificación de la tabla de la Figura 4.111 al tener en cuenta la predicción de seguir en secuencia.

funct3	bits 13 . . . 12				opSEC				Salidas de EVAL	mSIC (secu)		ErPred	
bit 14	00	01	10	11	3	2	1	0	ig, me, meu	cumple	no cumple	cumple	no cumple
0	beq				si	func	t3		ig	secuMod	secuImpl	error	acierto
		bne							not ig	secuMod	secuImpl	error	acierto
1	blt								me	secuMod	secuImpl	error	acierto
		bge							not me or ig	secuMod	secuImpl	error	acierto
			bltu						meu	secuMod	secuImpl	error	acierto
				bgeu					not meu or ig	secuMod	secuImpl	error	acierto

Figura 6.70 Predicción de seguir en secuencia. Instrucciones BRANCH. Condición que se evalúa, control del multiplexor mSIC y señal ErPred. Igualdad: ig, menor utilizando aritmética entera (signed): me, menor utilizando aritmética con naturales (unsigned): meu. Los acrónimos secuImpl/secuMod se codifican como 0/1.

La señal ErPred se genera en el módulo DECS, cuando se conoce el resultado de la evaluación. En la Figura 6.71 se muestra la modificación del autómata de control de la Figura 6.62.

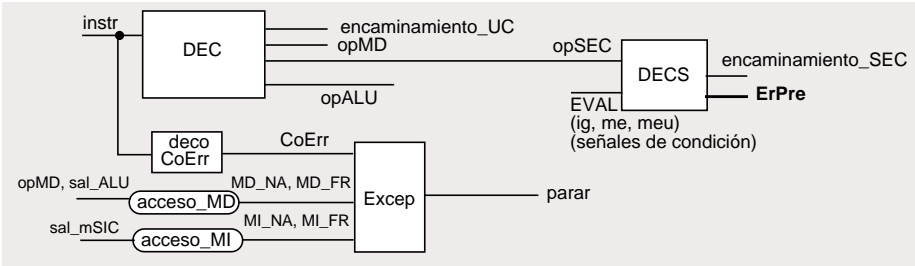


Figura 6.71 Predicción de seguir en secuencia- Modificación en los módulos del autómata de control. DEC: decodificación y gestión de las señales de error. DECS: segundo nivel de decodificación y control para las instrucciones de secuenciamiento condicional y gestión de las señales de condición. decoCoerr: identificación de secuencias de bits que no pertenecen al lenguaje máquina implementado.

Apéndice 6.14: Simulación: predicción de seguir en secuencia

Los ficheros que utiliza Quartus (Rproc_MD_MI_C_ModSecu_rell_PreSecu.qsf y Rproc_MD_MI_C_ModSecu_rell_PreSecu.qpf) se encuentran en el directorio LAB6/PROC_SEG_ModSecu_rell_PreSecu/ENSAMBLADO/QUARTUS

En los subdirectorios incluidos en LAB6/PROC_SEG_ModSecu_rell_PreSecu solo se incluyen los ficheros modificados respecto a los proyectos QUARTUS previos del procesador. En el fichero Rproc_MD_MI_C_ModSecu_rell_PreSecu.qsf se puede observar la ubicación de los ficheros que se utilizan.

Una vez ha sido abierto el proyecto con Quartus, hay que dar la orden "Processing _> Start -> Start Analysis & Elaboration". Esta orden elabora el diseño. Una vez elaborado el diseño sin errores, mediante la orden "Tools _> Netlist Viewers -> RTL Viewer" se puede comprobar la elaboración efectuada⁴².

El fichero Rproc_MD_MI_C_ModSecu_rell_PreSecu.vhd contiene la descripción en VHDL del procesador descrito.

Preparación de la simulación

El fichero de comprobación, denominado prueba_Rproc_MD_MI_C_ModSecu_rell_PreSecu.vhd se encuentra en el directorio LAB6/PROC_SEG_ModSecu_rell_PreSecu/ENSAMBLADO/PRUEBAS. En este directorio también se encuentra el fichero para formatear las señales que se visualizan en la ventana temporal.

El fichero prueba_Rproc_MD_MI_C_ModSecu_rell_PreSecu.vhd utiliza los mismos genéricos que el fichero parejo de la Práctica 4.

Simulación con ModelSim

Para efectuar una simulación efectúe los mismos pasos descritos en la Práctica 4.

42. En alguno de los ficheros que se suministra hay que efectuar una descripción de la lógica que se diseña en esta parte de la práctica (Apéndice 6.15). Los ficheros que se suministran contienen la declaración de la interface y la especificación de una arquitectura que no es correcta.

Simulación de un programa concreto

Los programas se encuentran en la misma ubicación que en la Práctica 4.

En la ventana de tiempo se muestran algunas de las señales del camino de datos, las cuales están ordenadas para facilitar la observación.

Ventana temporal de Modelsim. En la ventana temporal de ModelSim se muestran etiquetas que están asociadas a señales del camino de datos. Estas etiquetas están ordenadas para asociarlas a etapas de forma simple.

En particular, en el grupo de señales asociadas a la etapa ALU se ha añadido la señal ErPre, que indica si la predicción de sentido efectuada en la etapa DL, al ser ocupada por una instrucción de secuenciamiento condicional en el ciclo previo, es errónea.

ALU	PE_ALU	Permiso de escritura en el BR de la instrucción que ocupa la etapa ALU
	IDE_ALU	Identificador de registro destino de la instrucción que ocupa la etapa ALU
	opALU	Control de la ALU
	opSEC	Control del secuenciamiento
	opMD	Control de la MD
	me_meu_ig ^a	Salida del circuito que evalúa la condición
	sel_muxSIC	Selección del multiplexor muxSIC (secuenciamiento)
	muxSIC	Salida del multiplexor muxSIC (secuenciamiento)
	selmxsecseg	Selección del multiplexor mxsecseg (secuenciamiento)
	mxsecseg	Próximo valor del registro CP
	sel_mrell	Selección del multiplexor mrell
	muxrell	Salida del multiplexor mrell
	C2C1_L1 ^b	Cortocircuitos correspondientes al campo identificador de registro rs1 (IDL1) (C2L1 y C1L1)
	C2C1_L2	Cortocircuitos correspondientes al campo identificador de registro rs2 (IDL2) (C2L2 y C1L2)
	ErPre	Error de predicción de sentido en una instrucción de secuenciamiento condicional

a. Las señales están ordenadas, de izquierda a derecha, en el sentido en que se muestran.

b. Las señales C2L1 y C1L1, del camino de datos, que transportan un bit, han sido agrupadas en la señal C2C1_L1 en la ventana de tiempo. El bit más significativo de la señal C2C1_L1 corresponde a la señal C2L1 y el bit menos significativo a la señal C1L1. Para las señales CXC_Y_LZ se ha utilizado la misma codificación.

Figura 6.72 Predicción de seguir en secuencia. Señales en la ventana temporal de Modelsim.

Información textual de la simulación

En la representación textual se distinguen grupos de información compuestos de dos filas. En la primera fila se representan los cortocircuitos, información sobre la predicción cuando una instrucción de secuenciamiento condicional ocupa la etapa DL e información sobre la comprobación de la predicción cuando una instrucción de secuenciamiento condi-

cional ocupa la etapa ALU. En la segunda fila se representa información en cada etapa de la segmentación, además de la detección de riesgos. La representación utilizada es idéntica a la utilizada en la Práctica 5.

Representación de la predicción. En la primera fila, encima del acrónimo RD de la segunda fila, se utiliza el acrónimo Pre para indicar predicción. El tipo de predicción es seguir en secuencia. Para indicarlo se utiliza el acrónimo SE.

Representación de la comprobación de la predicción. En la primera fila, encima del acrónimo RS de la segunda fila, se utiliza el acrónimo CPr para indicar la comprobación de la predicción. Para indicar el resultado de la comprobación se utiliza el acrónimo E para error en la predicción y C para el caso contrario. .

Representación de las instrucciones. La representación de una instrucción de secuenciamiento condicional se modifica (Figura 6.73).

Instrucción	Etapas					Comentario
secuenciamiento condicional	DL	CPre	-	-	-	CPre: comprobación de la predicción

Figura 6.73 Acrónimos utilizadas en las etapas al progresar una instrucción de secuenciamiento condicional.

Representación de la inyección de una instrucción nop desde la etapa DL debido a un error de predicción. Se utiliza el acrónimo “nop” desde la etapa ALU. (Figura 6.74).

Instrucción	Etapas				
nop inyectada	DL	nop	nop	nop	nop

Figura 6.74 Acrónimos utilizados en una instrucción nop inyectada desde la etapa DL debido a un error de predicción.

En la Figura 6.75 se muestra una secuencia de instrucciones. En el ciclo 43, al ocupar la instrucción “beq x15, ...” la etapa DL, se predice seguir en secuencia. En el siguiente ciclo se comprueba la predicción y se determina que es correcta. Las instrucciones que siguen en secuencia se siguen interpretando.

En el ciclo 47, cuando la etapa DL está ocupada por la instrucción “bne x15, ...”, se predice seguir en secuencia. En el siguiente ciclo se comprueba que la predicción es errónea. En este ciclo se actualiza el registro CP. Además, se inyectan instrucciones “nop” desde la etapa DL y BUS hacia las etapas ALU y DL respectivamente. Esta acción se observa en el ciclo 49.

				L2	L1	L2	L1						
				c43	c43	c21	c21	1	2	3	4	Pre	CPr
Ciclo	prox_CP	BUS	DL (L.E.)	DL	A	M	FMTL	E	RD	RS			
42	0000011C	00E78A63	addi x13, x0, x000	00 00 DL	00 00 A	M	F	E	RD: 0	RS: 0			
43	00000120	02E7F263	beq x15, x14, x00A	01 10 DL	00 00 A	R	F	E	Pre: SE RD: 0	RS: 0			Predicción de seguir en secuencia
44	00000124	40F70733	bgeu x15, x14, x012	10 00 DL	00 00 CPr	R	R	E	Pre: SE RD: 0	CPr: C RS: 0			Predicción de seguir en secuencia Acierto en la predicción de "beq x15, "
45	00000128	00100693	sub x14, x14, x15	00 00 DL	00 00 CPr	-	R	E	RD:	CPr: C RS: 0			Acierto en la predicción de "bgeu x15, "
46	0000012C	FEE79AE3	addi x13, x0, x001	00 00 DL	00 00 A	-	-	E	RD: 0	RS: 0			
47	00000130	02069063	bne x15, x14, xFFA	00 00 DL	00 00 A	R	-	-	Pre: SE RD: 0	RS: 0			Predicción de seguir en secuencia
48	0000011C	00000013	bne x13, x0, x010	00 00 DL	00 00 CPr	R	R	-	Pre: SE RD: 0	CPr: E RS: 0			Predicción de seguir en secuencia Error en la predicción de "bne x15, ."
49	00000120	02E7F263	addi x0, x0, x000	00 00 DLN	00 10 nop	-	R	<e	RD: 0	RS: 0			
50	00000124	40F70733	bgeu x15, x14, x012	00 00 DL	00 00 AN	nop	-	E	Pre: SE RD: 0	RS: 0			Predicción de seguir en secuencia
51	00000128	00100693	sub x14, x14, x15	00 00 DL	00 00 CPr	RN	nop	-	RD: 0	CPr: C RS: 0			Acierto en la predicción de "bgeu x15, "
52	0000012C	FEE79AE3	addi x13, x0, x001	00 00 DL	00 00 A	-	RN	nop	RD: 0	RS: 0			

Figura 6.75 Predicción de seguir en secuencia. Representación textual de la interpretación de una secuencia de instrucciones.

Apéndice 6.15: Organización de los ficheros: predicción de seguir en secuencia

El árbol de directorios, desde el directorio raíz (LAB6/PROC_SEG_ModSecu_rell_PreSecu) de esta parte de la práctica, es mimético al árbol de los proyectos QUARTUS previos del procesador.

En los subdirectorios miméticos solo se incluyen los directorios que contienen ficheros modificados respecto a los proyectos previos.

El árbol de directorios, que contiene los ficheros VHDL modificados, que describen el procesador, en el cual se predice seguir en secuencia en instrucciones de secuenciamiento condicional, se muestra en la Figura 6.76 y en la Figura 6.77

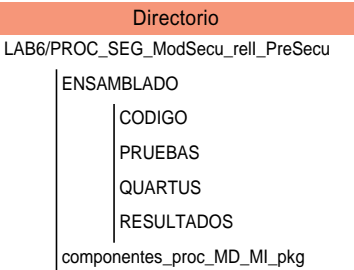


Figura 6.76 Predicción de seguir en secuencia en instrucciones de secuenciamiento condicional. Directorio que contiene los directorios y ficheros que describen el ensamblado del procesador con las memorias.

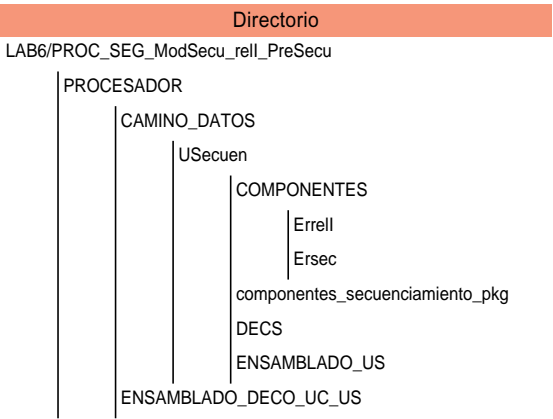


Figura 6.77 Predicción de seguir en secuencia en instrucciones de secuenciamiento condicional. Árbol de directorios del procesador.

En los directorios CONTROL_SEGMENTACION y CONTROL_SEG_CORTOS están incluidos todos los ficheros que se utilizan en la lógica de interbloqueos, excepto los que no se modifican de las prácticas o diseños previos (LIB, Figura 6.42). En la Figura 6.78 se muestran los directorios donde residen ficheros modificados y el fichero que debe editarse⁴³.

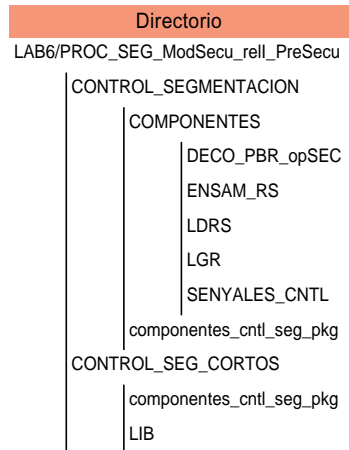


Figura 6.78 Predicción de seguir en secuencia en instrucciones de secuenciación condicional. Árbol de directorios del control de la segmentación y del control de los cortocircuitos.

Edición de ficheros. En los directorios Ersec y Erell (Figura 6.77) se incluyen los ficheros que deben utilizarse para diseñar el control de los multiplexores msecuseg y mrell. Cada fichero contiene la declaración de la interface⁴⁴, las cuales se corresponden con las especificadas en el fichero “componentes_secuenciamiento_PreSecu_pkg.vhd”, ubicado en el directorio “componentes_secuenciamiento_pkg”. La instanciación de estos elementos se efectúa en el fichero incluido en el directorio ENSAMBLADO_US.

En los directorios LDRS y LGR (Figura 6.78) se incluyen los ficheros que deben utilizarse para diseñar parte de la lógica LCIB. Cada fichero contiene las declaraciones de las interfaces, las cuales se corresponden con las especificadas en el fichero “componentes_control_seg_ModSecu_rell_PreSecu_pkg.vhd”, ubicado en el directorio “componentes_cntl_seg_pkg”. La instanciación de estos elementos se efectúa en el fichero incluido en el directorio ENSAM_RS.

43. Tenga en cuenta que los ficheros que se utilizan de otros proyectos QUARTUS del procesador deben ser accesibles.

44. Ninguna de las interfaces de los diseños que se solicitan debe modificarse.

Apéndice 6.16: Documentación: predicción de seguir en secuencia

La documentación ha sido generada utilizando la herramienta Doxygen. El fichero que hay que abrir con un navegador es "index.html" ubicado en el directorio LAB6/PROC_SEG_ModSecu_rell_PreSecu/documentacio/hmtl.

Las pestañas que se muestran son autoexplicativas.

Dado un módulo se muestra un grafo de dependencias jerárquicas con otros módulos. Pulsando en un nodo del grafo se observan el módulo o módulos que agrupa.

También se puede acceder al código VHDL que describe a un módulo.

Apéndice 6.17: Predicción de modificar el secuenciamiento: control del encaminamiento por instrucción

En la tabla de la Figura 6.79 se describen las señales de encaminamiento para cada instrucción (minúsculas) o subconjunto de instrucciones (mayúsculas). Además, se indican las unidades funcionales utilizadas. Esta tabla es una ampliación de la tabla Figura 6.61, se ha añadido una columna, correspondiente a la señal SCond, la cual se utiliza para controlar el multiplexor mrell. La señal SCond se genera en el módulo decoopSEC del decodificador.

CoOp	bits 4 . . . 2						PBR	inmD	mL1	mL2	mES	inmS	mSI	mSIC	SInCond	SCond	opALU	opMD	opSEC
bits 6 . . . 5	000	001	010	011	100	101													
00	LOAD						si	I	reg	inm	MD			secuImpl	no	no	si	si	no
					OP_IMM		si	I	reg	inm	ALU			secuImpl	no	no	si	no	no
						auipc	si	U	CP	inm	ALU			secuImpl	no	no	si	no	no
01	STORE						no	S	reg	inm				secuImpl	no	no	si	si	no
					OP		si		reg	reg	ALU			secuImpl	no	no	si	no	no
						lui	si	U	cero	inm				secuImpl	no	no	si	no	no
10																			
11	BRANCH						no		reg	reg		B	rel	secu	no	si	no	no	si
		jalr					si		reg		ret	I	indx	secuMod	no	no	no	no	si
				jal			si				ret	J	rel		si	no	no	no	si
					SYSTEM														

Figura 6.79 Predicción de modificar el secuenciamiento. Señales de encaminamiento para las instrucciones e indicación de las unidades funcionales utilizadas.

Cuando en una casilla no hay ningún acrónimo se está indicando que no es importante el valor.

En la Figura 6.80 se muestra la modificación del autómata de control de la Figura 6.62.

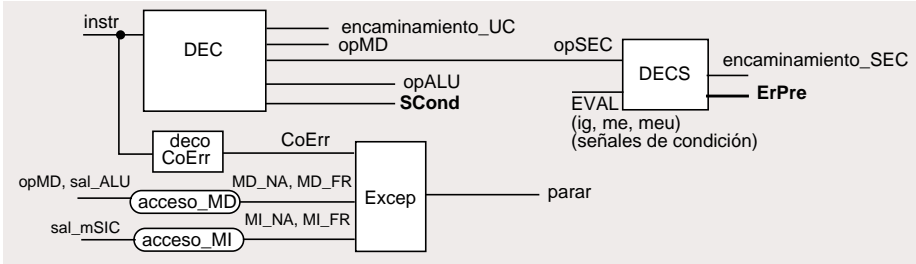


Figura 6.80 Predicción de modificar el secuenciamiento. Modificación en los módulos del autómata de control. DEC: decodificación y gestión de las señales de error. DECS: segundo nivel de decodificación y control para las instrucciones de secuenciamiento condicional y gestión de las señales de condición. decoCoerr: identificación de secuencias de bits que no pertenecen al lenguaje máquina implementado.

En la Figura 6.81 se muestra la modificación en la organización del decodificador DEC de la Figura 6.63.

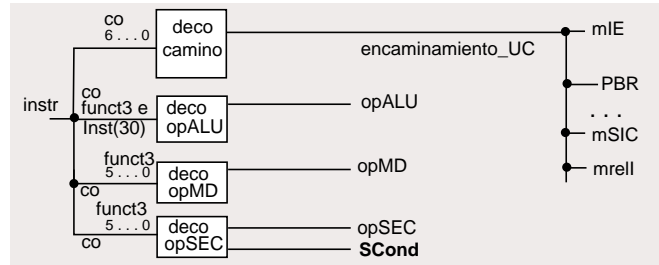


Figura 6.81 Predicción de modificar el secuenciamiento. Modificación en la organización del decodificador DEC. El módulo decocamino utiliza los bits del código de operación de una instrucción para generar las señales de encaminamiento. Los otros módulos utilizan los bits mostrados para generar las señales de control de las unidades funcionales.

Apéndice 6.18: Predicción de modificar el secuenciamiento: señales de control del secuenciamiento condicional

En la tabla de la Figura 6.82 se muestran las condiciones que se evalúan en las instrucciones BRANCH, el control del multiplexor mSIC (acrónimo secu) y la señal ErPred. Esta tabla es una modificación de la tabla de la Figura 4.111 al tener en cuenta la predicción de modificar el secuenciamiento.

funct3	bits 13 . . . 12				opSEC				Salidas de EVAL	mSIC (secu)		ErPred	
bit 14	00	01	10	11	3	2	1	0	ig, me, meu	cumple	no cumple	cumple	no cumple
0	beq				si	funct3			ig	secuMod	secuImpl	acierto	error
		bne							not ig	secuMod	secuImpl	acierto	error
1	blt								me	secuMod	secuImpl	acierto	error
		bge							not me or ig	secuMod	secuImpl	acierto	error
			bltu						meu	secuMod	secuImpl	acierto	error
				bgeu					not meu or ig	secuMod	secuImpl	acierto	error

Figura 6.82 Predicción de modificar el secuenciamiento. Instrucciones BRANCH. Condición que se evalúa, control del multiplexor mSIC y señal ErPred. Igualdad: ig, menor utilizando aritmética entera (signed): me, menor utilizando aritmética con naturales (unsigned): meu. Los acrónimos secuImpl/secuMod se codifican como 0/1.

La señal ErPred se genera en el módulo DECS (Figura 6.80), cuando se conoce el resultado de la evaluación.

Apéndice 6.19: Simulación: predicción de modificar el secuenciamiento

Los ficheros que utiliza Quartus (Rproc_MD_MI_C_ModSecu_rell_PreSal.qsf y Rproc_MD_MI_C_ModSecu_rell_PreSal.qpf) se encuentran en el directorio LAB6/PROC_SEG_ModSecu_rell_PreSal/ENSAMBLADO/QUARTUS

En los subdirectorios incluidos en LAB6/PROC_SEG_ModSecu_rell_PreSal solo se incluyen los ficheros modificados respecto a los proyectos QUARTUS previos del procesador. En el fichero Rproc_MD_MI_C_ModSecu_rell_PreSal.qsf se puede observar la ubicación de los ficheros que se utilizan.

Una vez ha sido abierto el proyecto con Quartus, hay que dar la orden "Processing _> Start -> Start Analysis & Elaboration". Esta orden elabora el diseño. Una vez elaborado el diseño sin errores, mediante la orden "Tools _> Netlist Viewers -> RTL Viewer" se puede comprobar la elaboración efectuada⁴⁵.

El fichero Rproc_MD_MI_C_ModSecu_rell_PreSal.vhd contiene la descripción en VHDL del procesador descrito.

Preparación de la simulación

El fichero de comprobación, denominado prueba_Rproc_MD_MI_C_ModSecu_rell_PreSal.vhd se encuentra en el directorio LAB6/PROC_SEG_ModSecu_rell_PreSal/ENSAMBLADO/PRUEBAS. En este directorio también se encuentra el fichero para formatear las señales que se visualizan en la ventana temporal.

El fichero prueba_Rproc_MD_MI_C_ModSecu_rell_PreSal.vhd utiliza los mismos genéricos que el fichero parejo de la Práctica 4.

Simulación con ModelSim

Para efectuar una simulación efectúe los mismos pasos descritos en la Práctica 4.

Simulación de un programa concreto

Los programas se encuentran en la misma ubicación que en la Práctica 4.

45. En alguno de los ficheros que se suministra hay que efectuar una descripción de la lógica que se diseña en esta parte de la práctica (Apéndice 6.20). Los ficheros que se suministran contienen la declaración de la interface y la especificación de una arquitectura que no es correcta.

En la ventana de tiempo se muestran algunas de las señales del camino de datos, las cuales están ordenadas para facilitar la observación.

Ventana temporal de Modelsim. En la ventana temporal de ModelSim se muestran etiquetas que están asociadas a señales del camino de datos. Estas etiquetas están ordenadas para asociarlas a etapas de forma simple.

En particular, en el grupo de señales asociadas a la etapa DL, se ha añadido la señal SCond, que indica que la etapa DL está ocupada por una instrucción de secuenciamiento condicional.

DL	PE_DL	Permiso de escritura en el BR de la instrucción que ocupa la etapa DL
	IDE_DL	Identificador de registro destino de la instrucción que ocupa la etapa DL
	C4C3_L1	Cortocircuitos correspondiente al campo identificador de registro rs1 (IDL1) (C4L1 y C3L1)
	C4C3_L2	Cortocircuitos correspondiente al campo identificador de registro rs2 (IDL2) (C4L2 y C3L2)
	val_IDL1	Validez del campo rs1 (IDL1) de la instrucción que ocupa la etapa DL
	IDL1	Campo rs1 (IDL1) de la instrucción que ocupa la etapa DL
	val_IDL2	Validez del campo rs2 (IDL2) de la instrucción que ocupa la etapa DL
	IDL2	Campo rs2 (IDL2) de la instrucción que ocupa la etapa DL
	SCond	Una instrucción de secuenciamiento condicional ocupa la etapa DL
	S_Incond_jal	Una instrucción de secuenciamiento "jal" ocupa la etapa DL
	instruc_DL	Instrucción en la etapa DL. Salida del registro BDL

Figura 6.83 Predicción de modificar el secuenciamiento en una instrucción de secuenciamiento condicional. Señales en la ventana temporal de Modelsim.

Información textual de la simulación

En la representación textual se distinguen grupos de información compuestos de dos filas. En la primera fila se representan los cortocircuitos, información sobre la predicción cuando una instrucción de secuenciamiento condicional ocupa la etapa DL e información sobre la comprobación de la predicción cuando una instrucción de secuenciamiento condicional ocupa la etapa ALU. En la segunda fila se representa información en cada etapa de la segmentación, además de la detección de riesgos. La representación utilizada es idéntica a la utilizada en la Práctica 5.

Representación de la predicción. En la primera fila, encima del acrónimo RD de la segunda fila, se utiliza el acrónimo Pre para indicar predicción. El tipo de predicción es modificar el secuenciamiento. Para indicarlo se utiliza el acrónimo SA (saltar).

En la Figura 6.84 se muestra una secuencia de instrucciones. En el ciclo 43, al ocupar la instrucción "beq x15,..." la etapa DL, se predice modificar el secuenciamiento, estableciéndose un valor en el registro CP. Debido al retardo de búsqueda se produce un riesgo

de secuenciamiento. En el siguiente ciclo se comprueba la predicción y se determina que es errónea. Se inyectan instrucciones “nop” desde las etapas BUS y DL hacia las etapas DL y ALU respectivamente..

En el ciclo 51, cuando la etapa DL está ocupada por la instrucción “bne x15, ...”, se predice modificar el secuenciamiento, estableciendo un valor en el registro CP. Debido al retardo de búsqueda se produce un riesgo de secuenciamiento. En el siguiente ciclo se comprueba que la predicción es correcta.

Ciclo	prox_CP	BUS	DL (L.E.)	L2 L1		1	2 3		4	Pre	CPr	RD	RS	
				c43 c43	c21 c21									
42	0000011C	00E78A63	addi x13, x0, x000	DL	A	M	F	E		RD: 0	RS: 0			
43	0000012C	00000013	beq x15, x14, x00A	DL	A	R	F	E	3	Pre: SA	RD: 0	RS: 1		Predicción de modificar el secuenciamiento
44	0000011C	00000013	addi x0, x0, x000	DLN	CPr	R	R	E		RD: 0	RS: 0			Error de predicción en “beq x15, ”
45	00000120	02E7F263	addi x0, x0, x000	DLN	nop	-	R	E		RD: 0	RS: 0			
46	00000140	00000013	bgeu x15, x14, x012	DL	AN	nop	-	E		Pre: SA	RD: 0	RS: 1		Predicción de modificar el secuenciamiento
47	00000120	00000013	addi x0, x0, x000	DLN	CPr	RN	nop	-		RD: 0	RS: 0			Error de predicción en “bgeu x15, ”
48	00000124	40F70733	addi x0, x0, x000	DLN	nop	-	RN	nop		RD: 0	RS: 0			
49	00000128	00100693	sub x14, x14, x15	DL	AN	nop	-	EN		RD: 0	RS: 0			
50	0000012C	FEE79AE3	addi x13, x0, x001	DL	A	RN	nop	-		RD: 0	RS: 0			
51	0000011C	00000013	bne x15, x14, xFFA	DL	A	R	RN	nop		Pre: SA	RD: 0	RS: 1		Predicción de modificar el secuenciamiento
52	00000120	FEE79AE3	addi x0, x0, x000	DLN	CPr	R	R	EN	2	RD: 0	RS: 0			Acierto en la predicción de “bne x15, ”
53	00000140	00000013	bgeu x15, x14, x012	DL	AN	-	R	E	4	Pre: SA	RD: 0	RS: 1		Predicción de modificar el secuenciamiento

Figura 6.84 Predicción de seguir en secuencia. Representación textual de la interpretación de una secuencia de instrucciones.

Apéndice 6.20: Organización de los ficheros: predicción de
modificar el secuenciamiento

El árbol de directorios, desde el directorio raíz (LAB6/PROC_SEG_ModSecu_rell_PreSal) de esta parte de la práctica, es mimético al árbol de los proyectos QUARTUS previos del procesador.

En los subdirectorios miméticos solo se incluyen los directorios que contienen ficheros modificados respecto a los proyectos previos.

El árbol de directorios, que contiene los ficheros VHDL modificados, que describen el procesador, en el cual se predice seguir en secuencia en instrucciones de secuenciamiento condicional, se muestra en la Figura 6.85 y en la Figura 6.86.

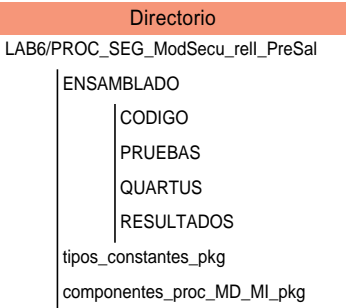


Figura 6.85 Predicción de modificar el secuenciamiento en instrucciones de secuenciamiento condicional. Directorio que contiene los directorios y ficheros que describen el ensamblado del procesador con las memorias.

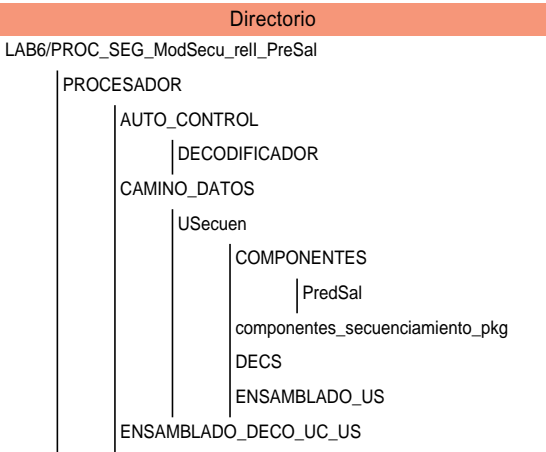


Figura 6.86 Predicción de modificar el secuenciamiento en instrucciones de secuenciamiento condicional. Árbol de directorios del procesador.

En los directorios CONTROL_SEGMENTACION y CONTROL_SEG_CORTOS están incluidos todos los ficheros que se utilizan en la lógica de interbloqueos, excepto los que no se modifican de las prácticas o diseños previos (LIB, Figura 6.42, el esquema de la lógica es el mismo que cuando se predice seguir en secuencia). En la Figura 6.87 y en la se muestran los directorios donde residen ficheros modificados y el fichero que debe editarse⁴⁶.

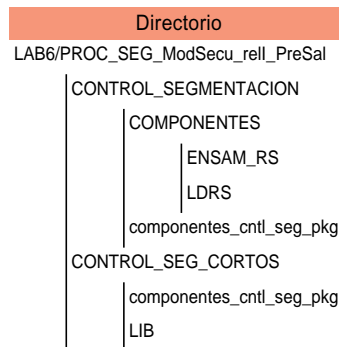


Figura 6.87 Predicción de modificar el secuenciamiento en instrucciones de secuenciamiento condicional. Árbol de directorio del control de la segmentación y del control de los cortocircuitos..

Edición de ficheros. En el directorio PredSal (Figura 6.86) se incluye el fichero que deben utilizarse para diseñar el control del multiplexor mrell. El fichero contiene la declaración de la interface⁴⁷, la cual se corresponde con la especificada en el fichero “componentes_secuenciamiento_PreSal_pkg.vhd”, ubicado en el directorio “componentes_secuenciamiento_pkg”. La instanciación de este elemento se efectúa en el fichero incluido en el directorio ENSAMBLADO_US.

En el directorio LDRS (Figura 6.87) se incluye el fichero que debe utilizarse para diseñar parte de la lógica LCIB. El fichero contiene la declaración de la interfaz, la cual se corresponden con la especificadas en el fichero “componentes_control_seg_ModSecu_rell_PreSal_pkg.vhd”, ubicado en el directorio “componentes_cntl_seg_pkg”. La instanciación de este elemento se efectúa en el fichero incluido en el directorio ENSAM_RS.

46. Tenga en cuenta que los ficheros que se utilizan de otros proyectos QUARTUS del procesador deben ser accesibles. .

47. Ninguna de las interfaces de los diseños que se solicitan debe modificarse.

Apéndice 6.21: Documentación: predicción de modificar el secuenciamiento

La documentación ha sido generada utilizando la herramienta Doxygen. El fichero que hay que abrir con un navegador es "index.html" ubicado en el directorio LAB6/PROC_SEG_ModSecu_rell_PreSal/documentacio/hmtl.

Las pestañas que se muestran son autoexplicativas.

Dado un módulo se muestra un grafo de dependencias jerárquicas con otros módulos. Pulsando en un nodo del grafo se observan el módulo o módulos que agrupa.

También se puede acceder al código VHDL que describe a un módulo.

Apéndice 6.22: Predicción en función del signo del literal: control del encaminamiento por instrucción

El módulo decodificador no se modifica respecto a predecir modificar el secuenciamiento. La tabla de encaminamiento es la mostrada en la Figura 6.79.

La señal de predicción PreDL, salida del módulo Predictor, se utiliza en el módulo PredS (Figura 6.48) para controlar el multiplexor mrell.

Apéndice 6.23: Predicción en función del signo del literal: señales de control del secuenciamiento condicional

En la tabla de la Figura 6.88 se muestran las condiciones que se evalúan en las instrucciones BRANCH, el control del multiplexor mSIC (acrónimo secu) y la señal ErPred. Esta tabla es una modificación de la tabla de la Figura 4.111 al tener en cuenta la predicción en función del signo del literal (señal Pred en la etapa ALU) en una instrucción de secuenciamiento.

funct3	bits 13 . . . 12				opSEC				Salidas de EVAL	mSIC (secu)		ErPred			
												Pre: SE		Pre: SA	
bit 14	00	01	10	11	3	2	1	0	ig, me, meu	cumple	no cumple	C	N.C.	C	N.C.
0	beq				si	funct3			ig	secuMod	secuImpl	E	A	A	E
		bne							not ig	secuMod	secuImpl	E	A	A	E
1	blt								me	secuMod	secuImpl	E	A	A	E
		bge							not me or ig	secuMod	secuImpl	E	A	A	E
			bltu						meu	secuMod	secuImpl	E	A	A	E
				bgeu					not meu or ig	secuMod	secuImpl	E	A	A	E

Figura 6.88 Predicción en función del signo del literal. Instrucciones BRANCH. Condición que se evalúa, control del multiplexor mSIC y señal ErPred. Igualdad: ig, menor utilizando aritmética entera (signed): me, menor utilizando aritmética con naturales (unsigned): meu. Los acrónimos secuImpl/secuMod se codifican como 0/1. Los acrónimos C y N.C. indican cumple y no cumple respectivamente. El acrónimo Pre indica predicción. Los acrónimos SE y SA indican seguir en secuencia y modificar el secuenciamiento. Los acrónimos A y E indican acierto y error.

La señal ErPred se genera en el módulo DECS (Figura 6.80), cuando se conoce el resultado de la evaluación.

Apéndice 6.24: Simulación: predicción en función del signo del literal

Los ficheros que utiliza Quartus (Rproc_MD_MI_C_PreSig.qsf y Rproc_MD_MI_C_ModSecu_rell_PreSig.qpf) se encuentran en el directorio LAB6/PROC_SEG_ModSecu_rell_PreSig/ENSAMBLADO/QUARTUS

En los subdirectorios incluidos en LAB6/PROC_SEG_ModSecu_rell_PreSig solo se incluyen los ficheros modificados respecto a los proyectos QUARTUS previos del procesador. En el fichero Rproc_MD_MI_C_ModSecu_rell_PreSig.qsf se puede observar la ubicación de los ficheros que se utilizan.

Una vez ha sido abierto el proyecto con Quartus, hay que dar la orden "Processing -> Start -> Start Analysis & Elaboration". Esta orden elabora el diseño. Una vez elaborado el diseño sin errores, mediante la orden "Tools -> Netlist Viewers -> RTL Viewer" se puede comprobar la elaboración efectuada⁴⁸.

El fichero Rproc_MD_MI_C_ModSecu_rell_PreSig.vhd contiene la descripción en VHDL del procesador descrito.

Preparación de la simulación

El fichero de comprobación, denominado prueba_Rproc_MD_MI_C_ModSecu_rell_PreSig.vhd se encuentra en el directorio LAB6/PROC_SEG_ModSecu_rell_PreSig/ENSAMBLADO/PRUEBAS. En este directorio también se encuentra el fichero para formatear las señales que se visualizan en la ventana temporal.

El fichero prueba_Rproc_MD_MI_C_ModSecu_rell_PreSig.vhd utiliza los mismos genéricos que el fichero parejo de la Práctica 4.

Simulación con ModelSim

Para efectuar una simulación efectúe los mismos pasos descritos en la Práctica 4.

Simulación de un programa concreto

Los programas se encuentran en la misma ubicación que en la Práctica 4.

48. En alguno de los ficheros que se suministra hay que efectuar una descripción de la lógica que se diseña en esta parte de la práctica (Apéndice 6.25). Los ficheros que se suministran contienen la declaración de la interface y la especificación de una arquitectura que no es correcta.

En la ventana de tiempo se muestran algunas de las señales del camino de datos, las cuales están ordenadas para facilitar la observación.

Ventana temporal de Modelsim. En la ventana temporal de ModelSim se muestran etiquetas que están asociadas a señales del camino de datos. Estas etiquetas están ordenadas para asociarlas a etapas de forma simple (Figura 6.89).

ALU	PE_ALU	Permiso de escritura en el BR de la instrucción que ocupa la etapa ALU
	IDE_ALU	Identificador de registro destino de la instrucción que ocupa la etapa ALU
	opALU	Control de la ALU
	opSEC	Control del secuenciamiento
	opMD	Control de la MD
	me_meu_ig ^a	Salida del circuito que evalúa la condición
	sel_muxSIC	Selección del multiplexor muxSIC (secuenciamiento)
	muxSIC	Salida del multiplexor muxSIC (secuenciamiento)
	selmxsecseg	Selección del multiplexor mxsecseg (secuenciamiento)
	mxsecseg	Próximo valor del registro CP
	sel_mrell	Selección del multiplexor mrell
	muxrell	Salida del multiplexor mrell
	C2C1_L1 ^b	Cortocircuitos correspondientes al campo identificador de registro rs1 (IDL1) (C2L1 y C1L1)
	C2C1_L2	Cortocircuitos correspondientes al campo identificador de registro rs2 (IDL2) (C2L2 y C1L2)
	ErPre	Error de predicción
	Pred_ALU	Predicción en la etapa ALU
DL	PE_DL	Permiso de escritura en el BR de la instrucción que ocupa la etapa DL
	IDE_DL	Identificador de registro destino de la instrucción que ocupa la etapa DL
	C4C3_L1	Cortocircuitos correspondiente al campo identificador de registro rs1 (IDL1) (C4L1 y C3L1)
	C4C3_L2	Cortocircuitos correspondiente al campo identificador de registro rs2 (IDL2) (C4L2 y C3L2)
	val_IDL1	Validez del campo rs1 (IDL1) de la instrucción que ocupa la etapa DL
	IDL1	Campo rs1 (IDL1) de la instrucción que ocupa la etapa DL
	val_IDL2	Validez del campo rs2 (IDL2) de la instrucción que ocupa la etapa DL
	IDL2	Campo rs2 (IDL2) de la instrucción que ocupa la etapa DL
	Pred	Predicción
	SCond	Una instrucción de secuenciamiento condicional ocupa la etapa DL
	S_Incond_jal	Una instrucción de secuenciamiento "jal" ocupa la etapa DL
	instruc_DL	Instrucción en la etapa DL. Salida del registro BDL

a. Las señales están ordenadas, de izquierda a derecha, en el sentido en que se muestran.

b. Las señales C2L1 y C1L1, del camino de datos, que transportan un bit, han sido agrupadas en la señal C2C1_L1 en la ventana de tiempo. El bit más significativo de la señal C2C1_L1 corresponde a la señal C2L1 y el bit menos significativo a la señal C1L1. Para las señales CXC_Y_LZ se ha utilizado la misma codificación.

Figura 6.89 Predicción de modificar el secuenciamiento en una instrucción de secuenciamiento condicional. Señales en la ventana temporal de Modelsim.

En particular, en el grupo de señales asociadas a la etapa DL, se ha añadido la señal Pred que indica la predicción efectuada en la etapa DL⁴⁹ y la propagación de esta señal a la etapa ALU (Pred_ALU).

Información textual de la simulación

La representación textual utilizada es mimética a la descrita en el proyecto QUARTUS previo.

49. Recordemos que, en la salida del predictor, esta señal no está validada.

Procesador: segmentación lineal con cortocircuitos

Apéndice 6.25: Organización de los ficheros: predicción en función del signo del literal

El árbol de directorios, desde el directorio raíz (LAB6/PROC_SEG_ModSecu_rell_PreSig) de esta parte de la práctica, es mimético al árbol de los proyectos QUARTUS previos del procesador.

En los subdirectorios miméticos solo se incluyen los directorios que contienen ficheros modificados respecto a los proyectos previos.

El árbol de directorios, que contiene los ficheros VHDL modificados, que describen el procesador, en el cual se predice seguir en secuencia en instrucciones de secuenciamiento condicional, se muestra en la Figura 6.90 y en la Figura 6.91.

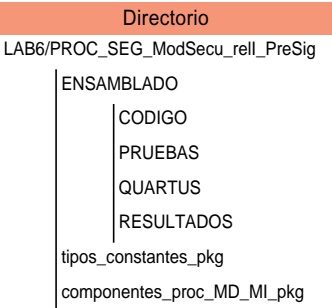


Figura 6.90 Predicción en función del signo del literal en instrucciones de secuenciamiento condicional. Directorio que contiene los directorios y ficheros que describen el ensamblado del procesador con las memorias.

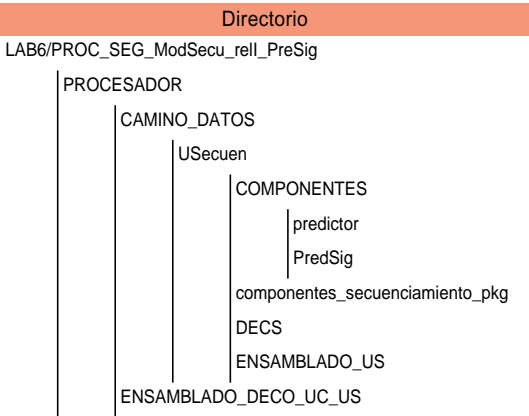


Figura 6.91 Predicción en función del signo del literal en instrucciones de secuenciamiento condicional. Arbol de directorios del procesador.

En los directorios CONTROL_SEGMENTACION y CONTROL_SEG_CORTOS están incluidos todos los ficheros que se utilizan en la lógica de interbloqueos, excepto los que no se modifican de las prácticas o diseños previos (LIB, Figura 6.48). En la Figura 6.92 se muestran los directorios donde residen ficheros modificados y el fichero que debe editarse⁵⁰.

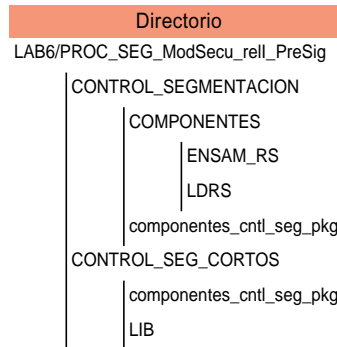


Figura 6.92 Predicción en función del signo del literal en instrucciones de secuenciamiento condicional. Árbol de directorio del control de la segmentación y del control de los cortocircuitos.

Edición de ficheros. En el directorio PredSig (Figura 6.91) se incluye el fichero que deben utilizarse para diseñar el control del multiplexor mrell. El fichero contiene la declaración de la interfaz⁵¹, la cual se corresponde con la especificada en el fichero “componentes_secuenciamiento_PreSig_pkg.vhd”, ubicado en el directorio “componentes_secuenciamiento_pkg”. La instanciación de este elemento se efectúa en el fichero incluido en el directorio ENSAMBLADO_US.

En el directorio LDRS/CODIGO (Figura 6.92) se incluye el fichero que debe utilizarse para diseñar parte de la lógica LCIB. El fichero contiene la declaración de la interfaz, la cual se corresponden con la especificadas en el fichero “componentes_control_seg_ModSecu_rell_PreSal_pkg.vhd”, ubicado en el directorio “componentes_cntl_seg_pkg”. La instanciación de este elemento se efectúa en el fichero incluido en el directorio ENSAM_RS.

50. Tenga en cuenta que los ficheros que se utilizan de otros proyectos QUARTUS del procesador deben ser accesibles.

51. Ninguna de las interfaces de los diseños que se solicitan debe modificarse.

Apéndice 6.26: Documentación: predicción en función del signo del literal

La documentación ha sido generada utilizando la herramienta Doxygen. El fichero que hay que abrir con un navegador es "index.html" ubicado en el directorio LAB6/PROC_SEG_ModSecu_rell_PreSig/documentacio/hmtl.

Las pestañas que se muestran son autoexplicativas.

Dado un módulo se muestra un grafo de dependencias jerárquicas con otros módulos. Pulsando en un nodo del grafo se observan el módulo o módulos que agrupa.

También se puede acceder al código VHDL que describe a un módulo.

