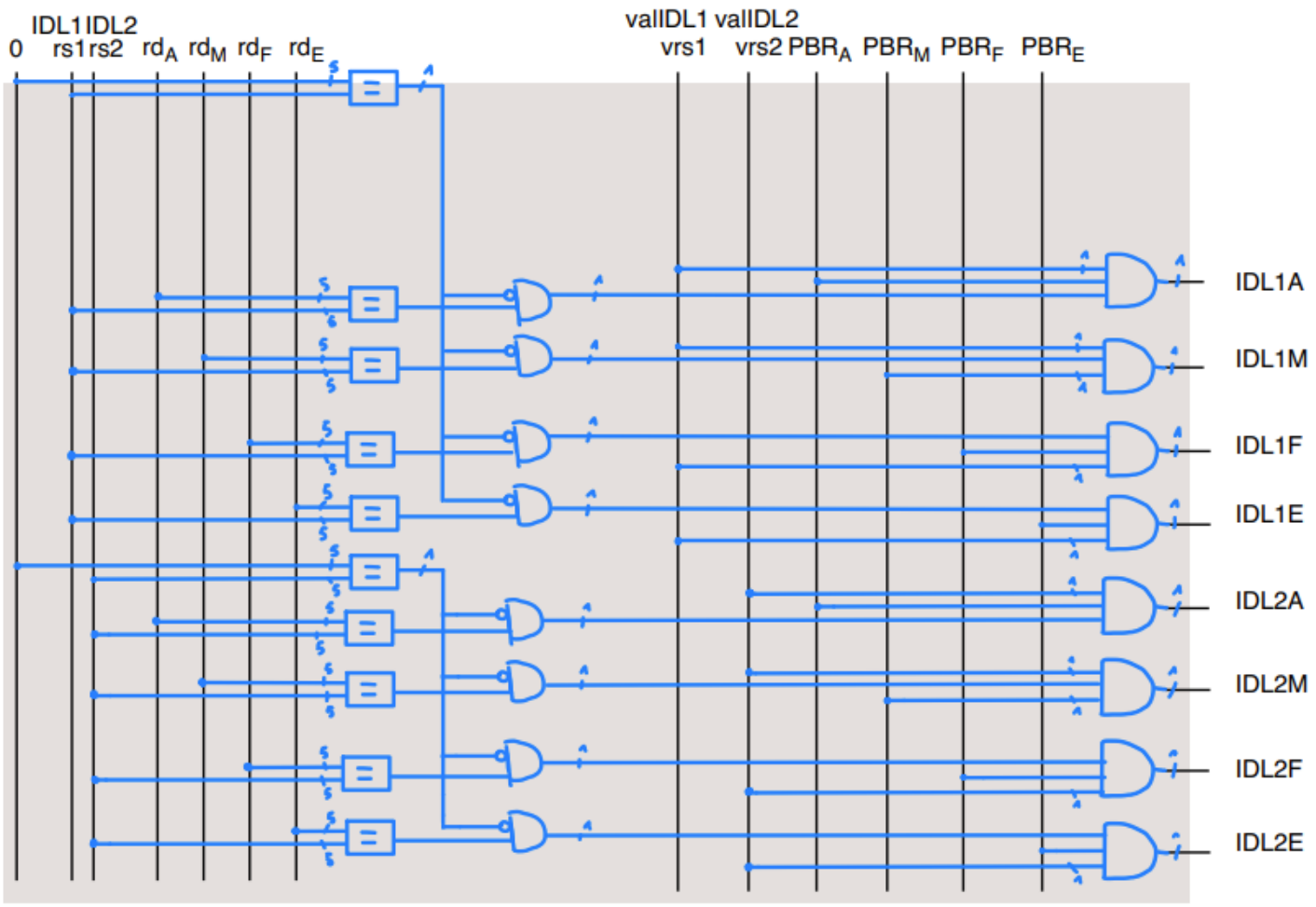
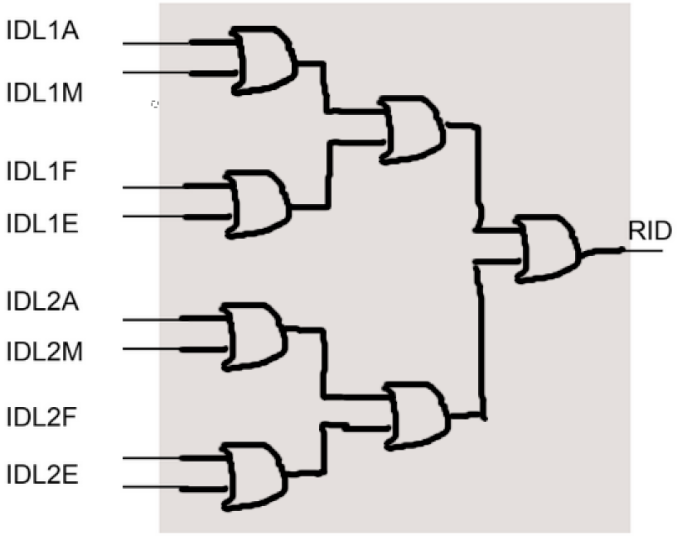
David Cañadas López y Roger Ortega Castilló

Pregunta 1: Diseñe el módulo LDD utilizando el menor número posible de puertas lógicas y comparadores (“Diseño de la lógica de interbloqueos“ en la página 368). Justifique el diseño de forma sucinta y sistemática.



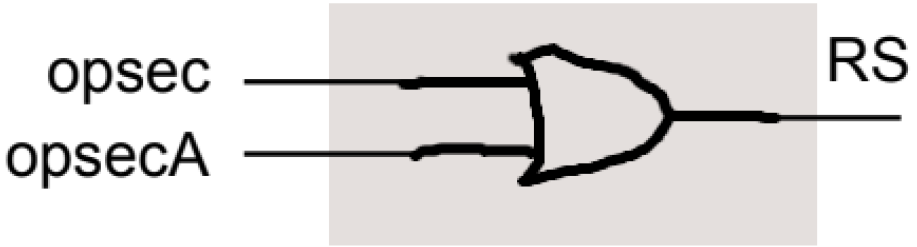
Justificación: En todos los casos hay que mirar que los registros usados (rs1, rs2) no sean el registro 0 ya que este nunca causará riesgo de datos (no se puede escribir en este registro). Además hay que mirar si los registros rs1 y rs2 son válidos y si tienen permisos de escritura según la etapa (ALU, M, FMT, ES).

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



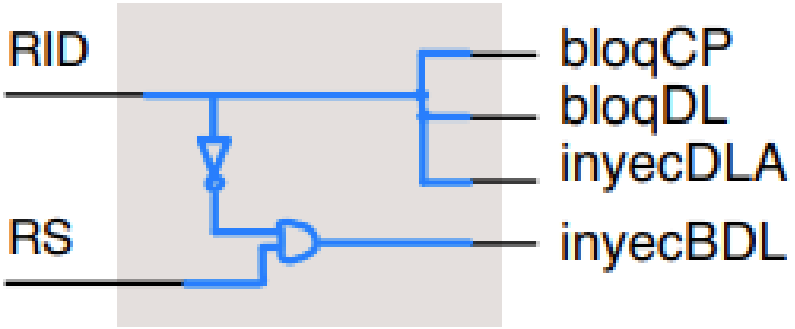
Justificación: Habrá un riesgo de datos mientras cualquiera de los registros que se pretende leer se esté procesando en otra etapa y no se haya escrito aún.

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



Justificación: Habrá un riesgo de secuenciamiento si el bit más alto (bit de válido) de al menos una de las señales de entrada sea 1.

Pregunta 4: Diseñe el módulo LGR utilizando el menor número posible de puertas lógicas, limitando el número de entradas a 2. Justifique el diseño de forma sucinta y sistemática.

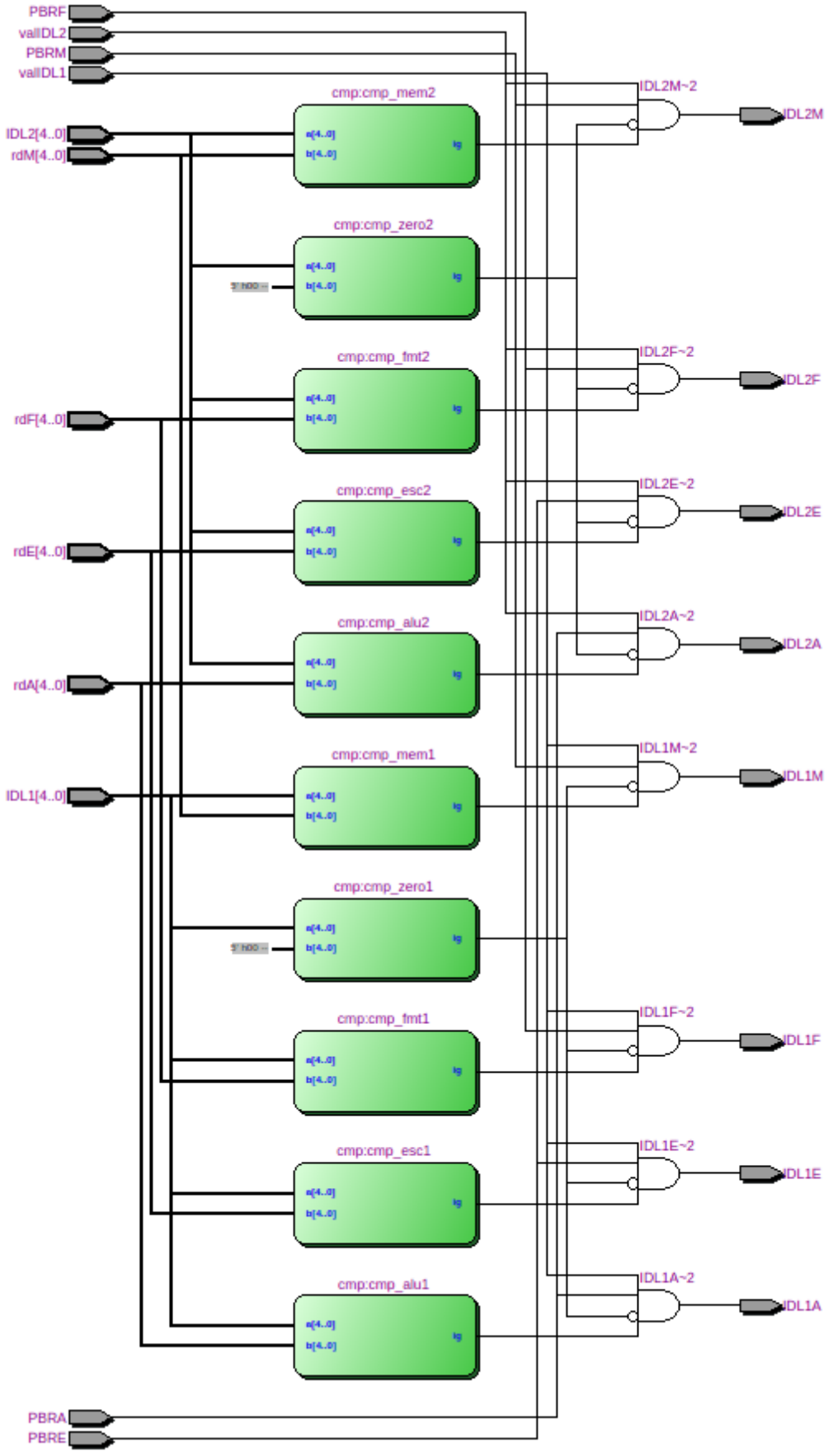


Justificación: En caso de riesgo de datos (RID) se deben detener las etapas posteriores a DL, además de bloquear CP y DL. En caso de riesgo de secuenciamiento (RS) y no de riesgo de datos, se debe detener DL.

David Cañadas López y Roger Ortega Castilló

Pregunta 5: En el subdirectorio LIB (Apéndice 5.2) se encuentran los ficheros asociados al diseño de la Lógica de InterBloqueos (proyecto quartus LIB.qpf). Describa en VHDL los 4 módulos anteriores (ficheros LDD.vhd, LDRD.vhd, LDRS.vhd y LGR.vhd), utilizando un modelo estructural. Los ficheros contienen la declaración de la interface. El módulo LDD tiene una estructura regular. Utilice sentencias generate en la descripción VHDL. Entregue el esquema RTL del módulo LDD elaborado por Quartus. Compruebe conjuntamente el diseño de los 4 módulos anteriores. El programa de prueba suministrado (prueba_LIB.vhd) compara en cada ciclo las salidas de los módulos diseñados con los respectivos modelos de referencia correctos.

David Cañadas López y Roger Ortega Castilló



David Cañadas López y Roger Ortega Castilló

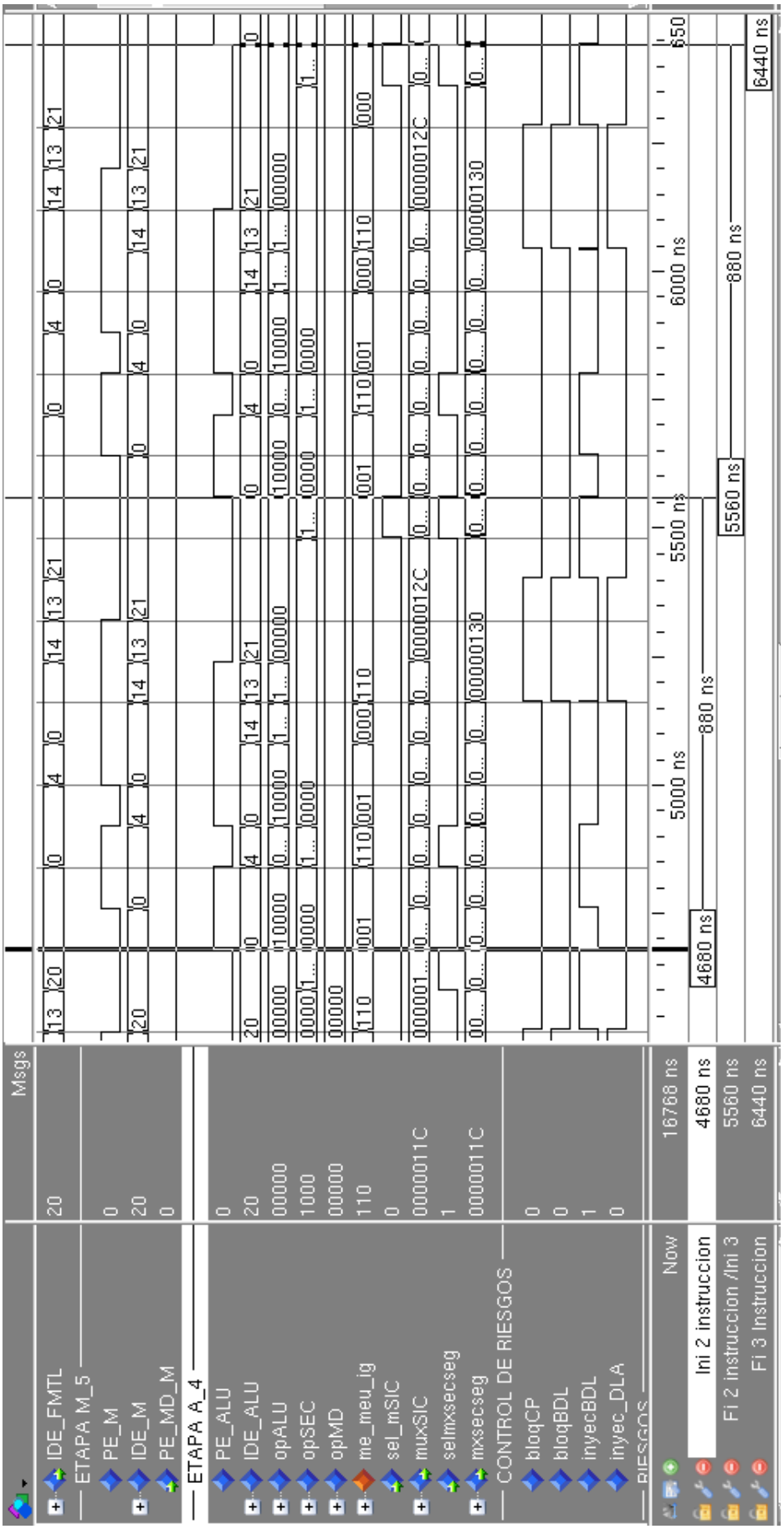
Pregunta 6: Utilice el programa euclides para comprobar el funcionamiento del procesador segmentado (“Simulación“ en la página 380). Añada un proceso al programa de prueba (ENSAMBLADO/PRUEBAS/prueba_Rproc_MD_MI.vhd) para obtener las métricas indicadas en la tabla. Calcule la Ganancia respecto del procesador serie.

Instrucciones ejecutadas	99
Instrucciones de secuenciamiento	32
Dependencias de datos 4 ciclos de bloqueo	3
Dependencias de datos 3 ciclos de bloqueo	9
Dependencias de datos 2 ciclos de bloqueo	1
Dependencias de datos 1 ciclos de bloqueo	0
Ciclos perdidos por riesgos de datos	41
Ciclos perdidos por riesgos de secuenciamiento	64
CPI	210/99 = 2.12
Ganancia	0.49

Texe proc. secuencial (lab4) = 8291 ns
Texe proc. segmentado (lab5) = 16681 ns

Pregunta 7: Entregue una copia de la ventana de tiempo correspondiente a la ejecución de la segunda y tercera iteración del programa de prueba euclides. Tenga en cuenta la forma de representar las instrucciones (“Evolución de las señales del camino de datos“ en la página 382) para explicar la ventana temporal. Identifique claramente los ciclos perdidos por riesgos de datos. Para ello, marque los ciclos en los cuales la etapa E está procesando nops inyectadas por la lógica de interbloqueos.

David Cañadas López y Roger Ortega Castilló



David Cañadas López y Roger Ortega Castilló

Pregunta 8: Suponga que se modifica el periodo de la señal de reloj para permitir leer en un mismo ciclo el valor con el cual se está actualizando un registro del banco de registros. Deduzca cuál debería ser el tiempo el tiempo de ciclo mínimo.

Tiempo de ciclo mínimo: ?

Utilizando los resultados de la pregunta 6 para el programa euclides, cuantifique si el rendimiento de esta opción sería mejor.

Instrucciones ejecutadas	99
Instrucciones de secuenciamiento	32
Dependencias de datos 4 ciclos de bloqueo	0
Dependencias de datos 3 ciclos de bloqueo	3
Dependencias de datos 2 ciclos de bloqueo	9
Dependencias de datos 1 ciclos de bloqueo	1
Ciclos perdidos por riesgos de datos	28
Ciclos perdidos por riesgos de secuenciamiento	64
CPI	197/99 = 1.98
Ganancia/Pérdida	?

Lo que pasará es que como las instrucciones que dependen de datos pueden leer un ciclo antes, las dep. de datos con 4 ciclos de bloqueo pasarán a ser de 3 ciclos de bloqueo, las de 3 de 2, las de 2 de 1 y las de 1 de 0. Habrá 3+9+1 ciclos menos perdidos por riesgos de datos, pero como la frecuencia habrá aumentado para permitir esta funcionalidad puede que la ganancia no sea tan inmensa.