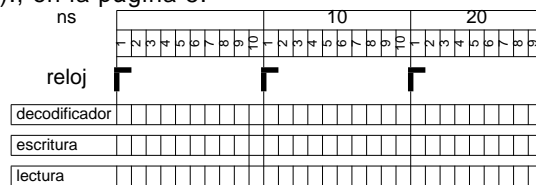


PRACTICA

1 CACHE CON ESCRITURA INMEDIATA

Trabajo 2: Muestre en sendos diagramas temporales de retardos un acceso de lectura y un acceso de escritura. Los retardos deben indicarse en el instante de tiempo más tardío en que pueden producirse para un funcionamiento correcto. Indique los retardos que pueden afectar a un valor, tanto en una posición de almacenamiento como en un puerto de salida. Utilice una señal de reloj cuadrada con periodo de 10 ns como referente. Utilice los valores de retardo especificados en el Apéndice 2.16 (ret_deco_dat, ret_dat_leer, ret_dat_esc), en la página 5.



Trabajo 9: Para el controlador de caché de la Figura 16, construya una tabla de transiciones entre estados (Apéndice 2.3), donde también se especifique la lógica de salida en la segunda subfila de cada estado (activación, desactivación de las señales). En el caso de la lógica de salida indique sólo la activación de la señal (valor 1). Suponga que por defecto las señales no se activan., en la página 15.

Trabajo 13: Analice el programa de prueba. Describa los procesos “productor” y “consumidor” mediante diagramas temporales. Así mismo, describa los procedimientos “Plectura” y “Pescritura”. Para ello, utilice la señal reloj y sus flancos como referente. Céntrese en el protocolo de las interfaces: comunicación entre el procesador y la caché y viceversa., en la página 16.

Trabajo 23: Construya una secuencia con el mínimo número de accesos, que muestre una comprobación incremental del diseño. Para este trabajo utilice una única entrada de cache. Muestre en una tabla la información actualizada en cada acceso del programa de prueba., en la página 17.

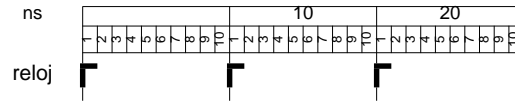
Acceso			Memoria			Cache			Camino que se comprueba		
ciclo	tipo (load, store)	dirección	trans (Pt, PtE)	variable	valor	contenedor	etiqueta	variable	valor	estado	

Trabajo 24: Una vez esté completamente implementado el controlador de cache, prepare una secuencia de accesos a memoria que compruebe el funcionamiento de forma exhaustiva (comprobar entradas distintas, entrelazar sin accesos consecutivos, accesos a la misma y distinta dirección, conflictos en cache)., en la página 17.

Trabajo 25: En la interface procesador/cache mostrada en la Figura 18, utilizada en el diseño RTL (Apéndice 2.4, Figura 45), todas las señales son entrada del multiplexor y del registro. Además, las señales sólo se almacenan en el registro si se cumple la función lógica “pc_val and pc_listo”. Esto es, si no hay petición y “pc_listo = ‘1’” no se actualiza el registro. Desde el punto de vista de la lógica se utilizan tantos multiplexores y registros como señales (pc_dir, pc_dato, pc_esc y pc_val, Figura 17). Rediseñe la interface, dibujando un esquema de circuito, de forma que se utilice el menor número de multiplexores y registros. Considere cada señal pc_dir, pc_dato, pc_esc y pc_val como un todo. Esto, si se utiliza un multiplexor o un registro contabilice una unidad., en la página 17.

Trabajo 26: Represente en un diagrama temporal de retardos los retardos de los componentes en un acierto de lectura. Suponemos que el último ciclo del productor se solapa con el estado DES en el controlador de cache. Los retardos correspondientes al productor deben indicarse en el instante más tardío en el cual es factible, teniendo en cuenta el periodo del reloj. El retardo de la comparación de etiquetas y la puerta “and” de este resultado con la lectura del campo estado tiene un retardo de 0 ns. Recuerde que las señales X_acc y X_esc están incluidas en la lógica de salida del autómata., en la página 18.

1	<input type="text" value="mux_pet"/>	multiplexor de la interface procesador/cache	10	<input type="text" value="m_EST"/>	acceso al campo estado
2	<input type="text" value="reg_pet"/>	registro de la interface procesador/cache	11	<input type="text" value="m_DAT"/>	acceso al campo datos
3	<input type="text" value="estado"/>	estado del CC	12	<input type="text" value="reg_cache_bus"/>	registro de acceso de la cache al bus
4	<input type="text" value="prx_estado"/>	próximo estado del CC	13	<input type="text" value="reg_bus_mem"/>	registro de acceso del bus a memoria
5	<input type="text" value="logi_salida"/>	lógica de salida del CC	14	<input type="text" value="deco_mem"/>	decodificador de memoria
6	<input type="text" value="deco_ET"/>	decodificador del campo etiquetad	15	<input type="text" value="memoria"/>	memoria
7	<input type="text" value="deco_EST"/>	decodificador del campo estado	16	<input type="text" value="reg_mem_bus"/>	registro de acceso de la memoria al bus
8	<input type="text" value="deco_DAT"/>	decodificador del campo datos	17	<input type="text" value="reg_bus_cache"/>	registro de acceso del bus a la cache
9	<input type="text" value="m_ET"/>	acceso al campo etiquetas			



Trabajo 35: Finalmente construya una tabla de transiciones entre estados (Apéndice 2.7, Figura 68) donde, además, se elimine el estado LEC en una transacción load que acierta en cache. Modifique la descripción VHDL del controlador de cache de forma oportuna. Elabore el controlador de cache con Quartus., en la página 20.

Trabajo 40: Construya la tabla de transiciones entre estados del controlador de cache, donde también se especifique la lógica de salida en la segunda subfila de cada estado (activación, desactivación de las señales) (Apéndice 2.10, Figura 72). Denomine mxL a la señal que controla el multiplexor muxL (Figura 26)., en la página 22.

Trabajo 48: Analice la influencia del multiplexor muxL en el tiempo de ciclo., en la página 23.

Trabajo 53: Para este proyecto, analice la necesidad de disponer de los estados ESB y ESCP., en la página 25.

Trabajo 56: Construya la tabla de transiciones entre estados del controlador de cache, donde también se especifique la lógica de salida en la segunda subfila de cada estado (activación, desactivación de las señales) (Apéndice 2.13, Figura 76)., en la página 26.

Trabajo 63: En la interface procesador/cache que se utiliza, todos los registros se actualizan cuando pc_listo = '1'. Rediseñe la interface, dibujando un esquema de circuito, de forma que las señales de entrada, que sea posible, se almacenen en el registro sólo si hay una petición pendiente (esquema de circuito)., en la página 27.

Trabajo 64: Represente en un diagrama temporal de retardos los retardos de los componentes en dos acierto de lectura consecutivos a direcciones distintas. El productor tarda 1 ciclo en producir accesos. Los retardos correspondientes al productor deben indicarse en el instante más tardío en el cual es factible, teniendo en cuenta el periodo del reloj. El retardo de la comparación de etiquetas y la puerta "and" de este resultado con la lectura del campo estado tiene un retardo de 0 ns., en la página 27.

1	<input type="text" value="mux_pet"/>	multiplexor de la interface procesador/cache	10	<input type="text" value="m_EST"/>	acceso al campo estado
2	<input type="text" value="reg_pet"/>	registro de la interface procesador/cache	11	<input type="text" value="m_DAT"/>	acceso al campo datos
3	<input type="text" value="estado"/>	estado del CC	12	<input type="text" value="reg_ET"/>	registro de entrada del comparador (Figura 29)
4	<input type="text" value="prx_estado"/>	próximo estado del CC			
5	<input type="text" value="logi_salida"/>	lógica de salida del CC			
6	<input type="text" value="deco_ET"/>	decodificador del campo etiquetad			
7	<input type="text" value="deco_EST"/>	decodificador del campo estado			
8	<input type="text" value="deco_DAT"/>	decodificador del campo datos			
9	<input type="text" value="m_ET"/>	acceso al campo etiquetas			

PRACTICA

■
■
■
■
■