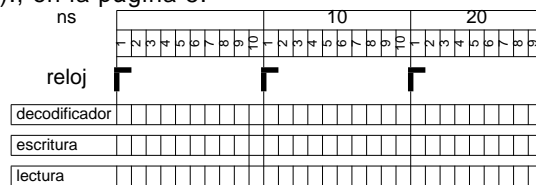


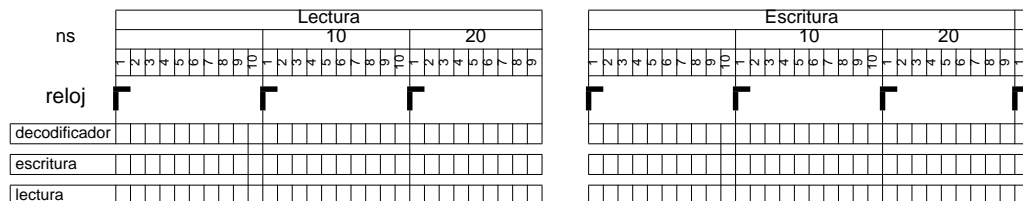
PRACTICA

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.



Tenga en cuenta que tanto la lectura como la escritura están sincronizadas con la señal de reloj. En concreto, con el flanco ascendente. Cualquier señal necesaria en un acceso a un elemento de memorización tiene que estar estable antes del flanco correspondiente. Por ejemplo la decodificación de la dirección.



Trabajo 9: Para el controlador de cache 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.

El diagrama de estados y las transiciones entre ellos se suministran en la práctica. Solo es necesario efectuar una codificación. En el Apéndice 2.3 se suministra la tabla de transiciones entre estados con las acciones que se efectúan en el camino de datos. Note que se indican acciones, no el instante de activación de las señales para que se lleven a término estas acciones.

En cada estado identifique el valor que deben tener las señales de las interfaces. Posteriormente identifique las señales cuyo valor es necesario evaluar. Finalmente identifique las señales del camino de datos en las cuales debe establecerse un valor distinto del valor por defecto.

Recuerde la pregunta previa. En un acceso a un elemento de memorización las señales de interés en el acceso deben ser estables antes del flanco ascendente del reloj. Eso es, hay que establecer el valor en el ciclo previo en el cual se quiere acceder a memoria.

Recuerde especificar las señales, generadas por el controlador, de la interface con el procesador tanto cuando actúa como productor como consumidor (pc_listo, rc_val, Figura 6). Si es el caso, estas señales deben especificarse las últimas en la lista de señales correspondiente a un estado.

Recomendación. Es recomendable codificar y comprobar el funcionamiento de forma incremental. Por ejemplo, empezar con las transiciones entre estados de la transacción inicio y posteriormente seguir con las transiciones entre estados de una instrucción load que acierta en cache. Mediante estas dos transacciones se comprueba un funcionamiento básico del camino de datos de la cache y la parte implementada del controlador de cache.

Para especificar el autómata hay que utilizar la tabla que se muestra.

		procesador			señales de estado (AF, est.)				memoria		
		peticion		peticion	lectura		escritura		resp. memoria		
		inicio	fin		si	no	si	no	si	no	
Estados	DES0										
	INI										
	ESCINI										
	DES										
	COMPET										
	LEC										
	HECHOL										
	PML										
	ESPL										
	ESB										
	PMEA										
	ESPEA										
	ESCP										
	PMEF										
	ESPEF										
	HECHOE										

Los siguientes diseños que se solicitan son diseños incrementales que parten de este diseño. Por ello, hay que comprobar su funcionamiento de forma detallada. En los siguientes diseños incrementales se eliminan estados. Entonces, conceptualmente las operaciones que se efectúan en un estado que se elimina deben efectuarse en el estado(s) previo.

Especifique el código. (tamaño de letra 8). En la especificación del controlador en VHDL hay que utilizar los procedimientos que se suministran.

Pueden distinguirse tres tipos de procedimientos y funciones: a) relacionados con las interface, b) relacionado con el control del camino de datos y c) relacionado con la evaluación de condiciones en señales del dispositivo (señales internas o de las interfaces).

En el diseño de un autómata se establecen valores por defecto en el valor de cada señal. En estas condiciones, es suficiente en cada estado establecer el valor de las señales cuyo valor debe ser distinto.

-- logica de proximo estado

-- logica de proximo estado



-- logica de salida	-- logica de salida

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 cache 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 (ini, load, store)	dirección	trans (Pt, PtE)	variable	valor	contenedor	etiqueta	variable	valor	estado	

Note que la comprobación del camino de datos debe ser INCREMENTAL. Un acceso utiliza la parte del camino de datos comprobada en un acceso previo, si es el caso. Hay que comprobar tanto la lectura como la escritura en los campos de la cache y la gestión de conflictos. Tenga en cuenta que “ini” es un tipo de acceso específico. Esto es, no es ni un load, ni un store. Suponga que hay 8 conjuntos. El número de accesos necesarios es menor de 8.

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

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.

Explique de forma concisa la comprobación que se efectúa para cada uno de los casos.

Acceso				Cache				Explicación
ciclo	tipo (ini, load, store)	dirección	trans (Pt, PtE)	acierto	fallo	conflicto	valor	

Justifique si las siguientes sentencias que proponen comprobaciones son de utilidad para comprobar el funcionamiento de la cache.

sentencia	Justificación
Es necesario comprobar la utilización de todos los contenedores.	
Es necesario comprobar ráfagas sin ciclos de espera en la generación de accesos por parte del productor tanto en instrucciones load como en instrucciones store.	
Es necesario comprobar ráfagas con ciclos de espera en la generación de accesos por parte del productor tanto en instrucciones load como en instrucciones store.	
Es necesario comprobar el entrelazado de accesos load y store con y sin ciclos de espera en la generación.	
Es necesario comprobar la gestión de conflictos en instrucciones load.	
Es necesario comprobar el funcionamiento de instrucciones store que fallan en cache y el contenedor almacena un bloque.	
Es necesario comprobar el consumo del valor de un load por parte del procesador con ciclos de espera.	



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.

Analice en detalle cuándo se utilizan las señales en el camino de datos. Más en concreto, identifique los estados del autómata en los cuales se utilizan las señales.

señal	Etapas donde se utiliza	Necesidad		Justificación
		multiplexor	registro	

Esquema de circuito.

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.

Recuerde que la petición “inicio” debe seguir operativa, además de las operaciones load y store. Recuerde que las señales necesarias para acceder a un elemento de memorización deben ser estables en el ciclo previo. Por otro lado, recuerde que es un diseño incremental. En estas condiciones, un estado previo al que se absorbe debe de efectuar las acciones del absorbido. El resto de estados no ven modificadas las acciones que realizaban.

Para especificar el autómata hay que utilizar la tabla que se suministra.

		procesador			señales de estado (AF, est.)				memoria			
		peticion		peticion	lectura		escritura		resp. memoria			
		inicio	início		si	no	si	no	si	no		
Estados	DES0											
	INI											
	ESCINI											
	DES											
	CMPET											
	LEC											
	ESPL											
	ESB											
	ESPEA											
	ESCP											
	ESPEF											



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.

Utilice la tabla que se suministra sin modificar.

		procesador			señales de estado (AF, est.)				memoria		
		peticion		petición	lectura		escritura		resp. memoria		
		inicio	início		si	no	si	no	si	no	
Estados	DES0										
	INI										
	ESCINI										
	DES										
	CMPET										
	ESPL										
	ESB										
	ESPEA										
	ESCP										
	ESPEF										

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

En el ciclo en el cual se utiliza el multiplexor, analice en detalle los recursos que se están utilizando y el retardo de los mismos.

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

Analice en detalle los recursos que se utilizan en estos estados y los que se utilizan en el estado que sigue a los estados ESB y ESCP.

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.

Utilice la tabla que se suministra sin modificar.

.....

		procesador y señales de estado							memoria			
		inicio		peticion				$\overline{\text{peticion}}$	resp. memoria			
				lectura		escritura			si	no		
		si	no	si	no	si	no					
Estados	DES0											
	INI											
	ESCINI											
	DES											
	ESPL											
	ESB											
	ESPEA											
	ESCP											
	ESPEF											

Especifique el código. (tamaño de letra 8). En la especificación del controlador en VHDL hay que utilizar los procedimientos que se suministran.

-- logica de proximo estado	-- logica de proximo estado



-- logica de salida	-- logica de salida

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.

Analice en detalle cuándo se utilizan las señales en el camino de datos. Más en concreto, identifique los estados del autómata en los cuales se utilizan las señales.

señal	solo si hay petición pendiente $pc_listo \text{ and } pc_val$	Necesidad de multiplexor	Justificación

Esquema de circuito.

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

Utilice el diagrama que se suministra. No modifique la ordenación de las señales.

