

Multiprocesadores: Práctica 2

Eric Rufart y David Soldevila

Grupo 11

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.

lect																						
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	
dec o																						
esc																						
lect																						

escr																					
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0
dec o																					
esc																					
lect																					

En el cuadro 1 podemos observar los retardos de los componentes en una operación de lectura. La lectura requiere de 3 ns que se corresponden al retardo ret_dat_leer y siendo los últimos 3 ns del ciclo el momento más tardío en el que se ejecuta correctamente. Dado que la lectura es síncrona con el reloj los valores deben ser decodificados antes del flanco ascendente anterior a esta y como el retardo de decodificación son 2ns, ret_deco_dat, debe haberse estabilizado durante los dos últimos nanosegundos del ciclo anterior.

Respecto a la escritura, observable en el cuadro 2, la lectura y decodificación siguen las mismas normas, pero se le añade un retraso de 2ns a la lectura por culpa de la escritura.

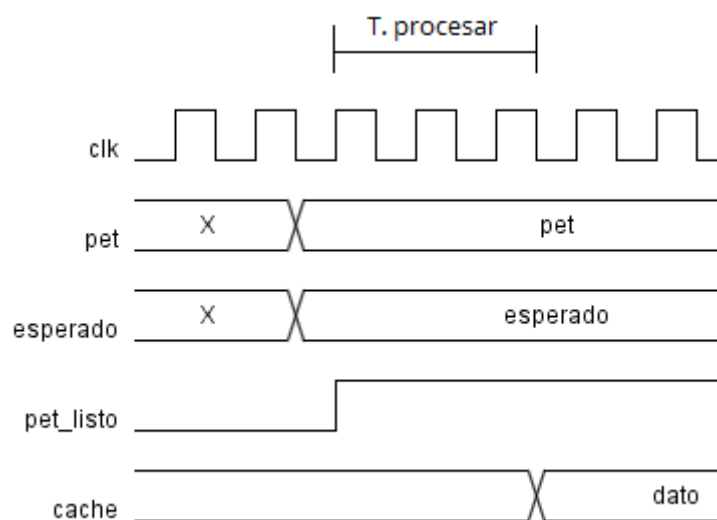
Trabajo 9: Para el controlador de caché de la Figura 16, construya una tabla de transiciones entre estados (Apéndice 2.3), donde también se especifique la lógica de salida en la segunda subfila de cada estado (activación, desactivación de las señales). En el caso de la lógica de salida indique sólo la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

Estados	Procesador			Señales de estado				Memoria		
	Petición		!Petición	lectura		escritura		resp. memoria		
	inicio	!inicio		si	no	si	no	si	no	
DESO	INI	COMPET	DESO							
	nada	nada ET_acc, EST_acc	desocupada pc_listo							
INI										ESCINI
ESCINI										HECHOE
										actualiza campos,ET_esc, EST_esc, DAT_esc, EST_DE, muxE
DES		COMPET	DES							
		nada ET_acc, EST_acc	desocupada pc_listo							
COMPET				LEC	PML	PMEA	PMEF			
				leer etiqu. y comparar, leer estado						
LEC										HECHOL
										lec campo dato DAT_ACC
HECHOL										DES
										inidicacion de dato rc_val
PML										ESPL
										pet. lec. memoria pm_val
ESPL								ESB	ESPL	
								nada	nada	
ESB										LEC
										actualiza campos ET_esc, EST_esc, DAT_esc, EST_DE
PMEA										ESPEA
										pet. esc. memoria pm_esc
ESPEA								ESCP	ESPEA	
								nada	nada	
ESPC										HECHOE
										actualiza campos DAT_esc
PMEF										ESPEF
										pet. esc. memoria pm_esc
ESPEF								HECHOE	ESPEF	
								nada	nada	
HECHOE										DES

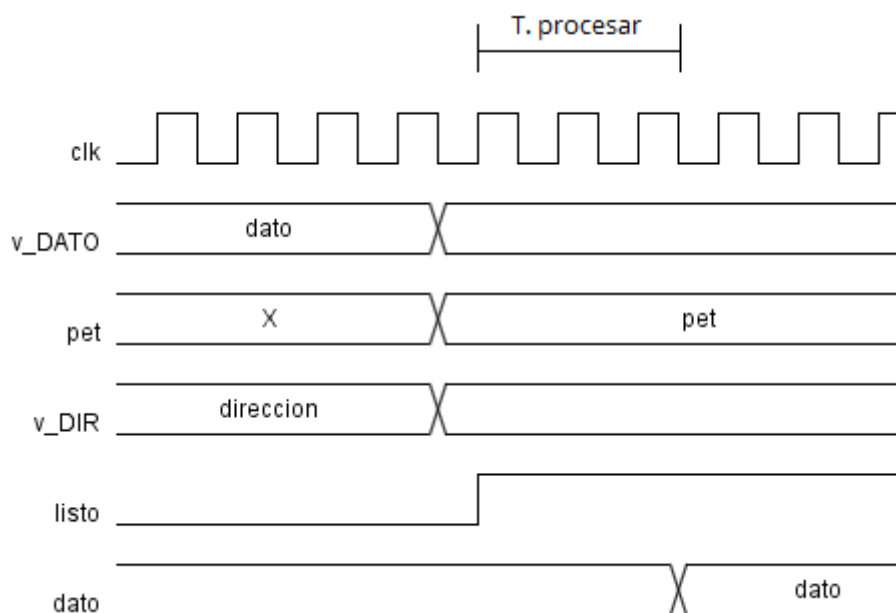
		nada
--	--	------

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.

Se genera una petición de lectura en la etapa Plectura ocn el valor adecuado en “esperando” tra el fanco descendente y acaba de poner “pet” con un retardo. Tras el siguiente flanco ascendente se pone en “listo” a 1 y la cache procesa el valor pedido, esto tardara un tiempo “t.procesar”.



“Pescritura” empieza generando la petición de escritura en función del dato y la dirección suministrada. Seguidamente activa la señal de listo y despues un tiempo de procesamiento la caché procesa el dato.



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

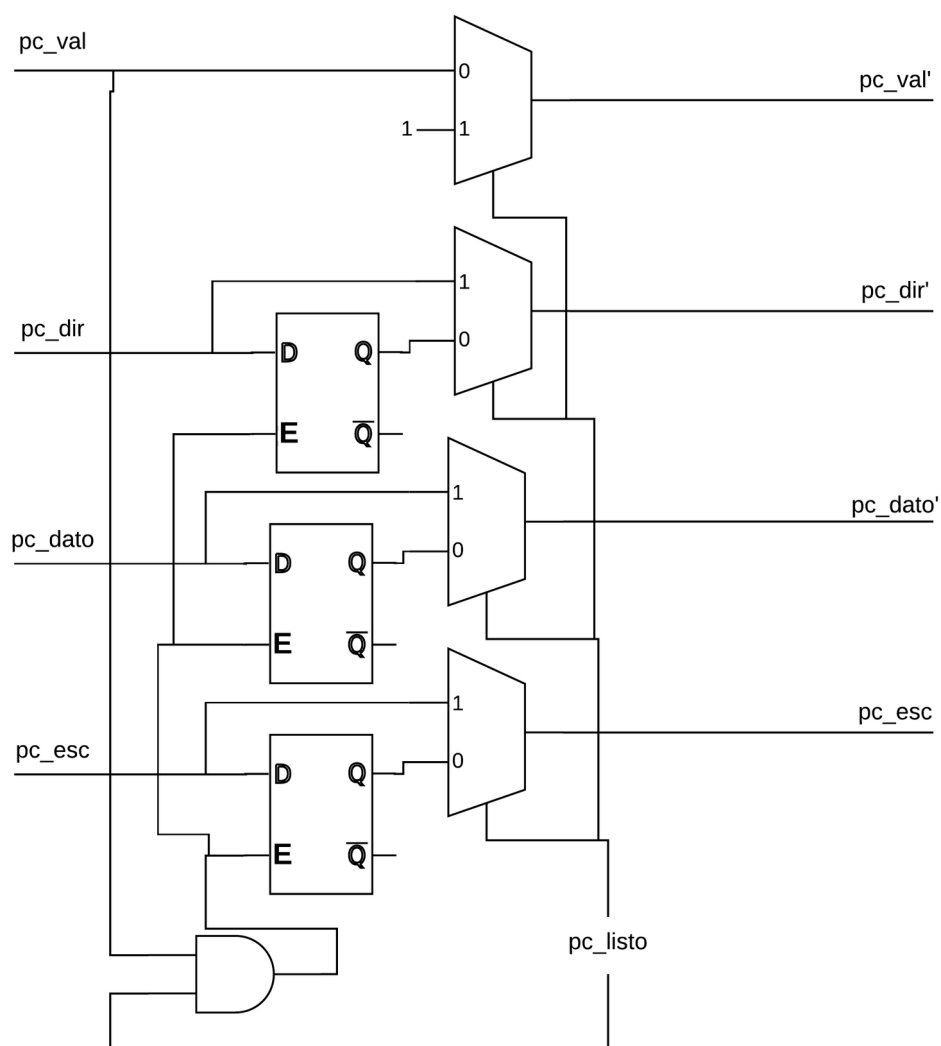
Cada fila de la tabla muestra un acceso a cache junto al estado de la cache y la memoria en el momento en el que se inicia el acceso. La última fila muestra el estado al finalizar todos los accesos.

ciclo	Acceso			Memoria			Cache			Camino que se	
	Tipo	Dirección	Transacción	Variable	Valor	Contenedor	Etiqueta	Variable	Valor	Estado	comprueba
4	Store	0000	PtE	0000	X	0	XXX	XXXX	X	X	STORE - MISS
13	Load	0000	Pt	0000	1	0	XXX	XXXX	X	X	LOAD - MISS
24	Load	0000	---	0000	1	0	000	0000	1	1	LOAD - HIT
28	Store	0000	PtE	0000	1	0	000	0000	1	1	STORE - HIT
38	---	---	---	0000	42	0	000	0000	42	1	---

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.

Tipo	Dirección	Descripción
Store	0000	Store con Miss contenedor 0
Load	0000	Load con Miss contenedor 0
Load	0000	Load con HIT contenedor 0
Store	0000	Store con HIT contenedor 0
Store	0001	Store con Miss contenedor 1
Load	0001	Load con Miss contenedor 1
Load	0001	Load con HIT contenedor 1
Load	0009	Load MISS contenedor 1, reemplazo
Store	0001	Store MISS contenedor 1
-	-	No se realiza acceso
Load	0009	Load con HIT contenedor 1

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 caché. 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 autómeta., en la página 18.

	0										10										20									
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
clk																														
mux_pet																														
reg_pet																														
estado																														
prx_est																														
logi_sal																														
deco_ET																														
deco_EST																														
deco_DAT																														
m_ET																														
m_EST																														
m_DAT																														
reg_cache																														
̄bus																														
reg_bus_mem																														
deco_mem																														
memoria																														
reg_mem_bus																														
reg_bus_cache																														

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 caché. Modifique la descripción VHDL del controlador de caché de forma oportuna.

Elabore el controlador de cache con Quartus., en la página 20.

Estados	Procesador			Señales de estado				Memoria	
	Petición inicio	Petición !inicio	!Petición	lectura sí	lectura no	escritura sí	escritura no	resp. memoria sí	memoria no
DESO	INI	COMPET	DESO						
INI									ESCINI
ESCINI									DES
DES		COMPET	DES						
COMPET				DES	ESPL	ESPEA	ESPEF		
LEC									DES
ESPL								ESB	ESPL
ESB									LEC
ESPEA								ESCP	ESPEA
ESCP									DES
ESPEF								DES	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.

Estados	Procesador			Señales de estado				Memoria	
	Petición inicio	!inicio	!Petición	lectura si	no	escritura si	no	resp. memoria si	no
DESO	INI	COMPET	DESO						
		ET_acc, EST_acc	desocupad a pc_listo						
INI									ESCINI
ESCINI									DES
DES		COMPET	DES						
		ET_acc, EST_acc	desocupad a pc_listo						
COMPET				DES	ESPL	ESPEA	ESPEF		
				leer etiqueta y comparar, leer estado					
				lectura campo dato, DAT_acc	pet. lectura memoria, pm_val	pet. escritura memoria, pm_esc			
ESPL								ESB	ESPL
								muxL, actualiza campos ET_esc, EST_esc, DAT_esc, EST_DE	
ESB									DES
ESPEA								ESCP	ESPEA
								actualiza campos, DAT_esc	
ESCP									DES
ESPEF								DES	ESPEF

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

El muxL aumenta ligeramente el tiempo de ciclo dado que es un multiplexor el cual depende de las salidas del controlador de caché y de la propia salida de datos de la caché o de los datos que llegan desde la memoria principal. Sin embargo es un ligero incremento en el tiempo de ciclo que permite enviar el dato al procesador en el mismo ciclo que llegar desde la memoria principal.

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

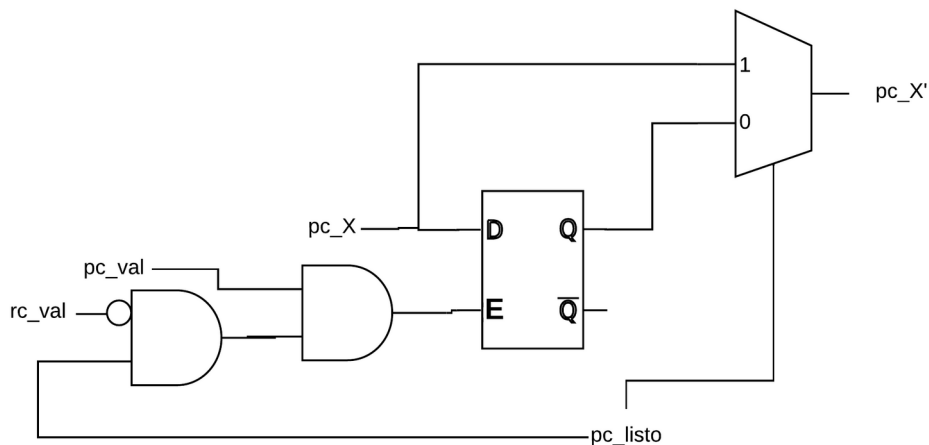
Dado que la lectura con acierto ha sido reducida a un único ciclo requerimos de los estados ESB y ESCP para actualizar los campos de la memoria caché una vez la memoria principal haya respondido a la correspondiente petición que se le haya hecho. Para eso necesitamos otro estado diferente al de esperar a la memoria principal intermedio entre dicho estado y el estado DES, los estados ESB y ESCP son los estados que lleva a cabo esa función.

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.

Estados	Procesador			Señales de estado				Memoria	
	Petición inicio	Petición !inicio	Petición !	lectura si	lectura no	escritura si	escritura no	resp. memoria si	resp. memoria no
DESO	INI	CMPET	DESO						
		ET_acc, EST_acc	desocupa da pc_listo						
INI									ESCINI
ESCINI									DES
DES		DES	DES	DES	ESPL	ESPEA	ESPEF		
		leer etiqueta y comparar, leer estado, ET_acc, EST_acc	desocupa da pc_listo	lectura campo dato DAT_acc	pet. lectura memoria, pm_va	pet. escritura memoria, pm_esc			
ESPL								ESB	ESPL
								muxL, actualiza campos ET_esc, EST_esc, DAT_esc, EST_DE	
ESB									DES
ESPEA								ESCP	ESPEA
								actualiza campos, DAT_esc	
ESCP									DES
ESPEF								DES	ESPEF

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.

Basándonos en el diseño del trabajo 25, obtenemos el siguiente diseño. En el podemos observar que solo se actualiza el registro si, además de las condiciones anteriores (cache lista y entrada válida) también se da rc_val = '0'. Esto hace que en los casos de load con hit, al finalizarse la petición en el mismo ciclo rc_val valga '1' y no se guarden datos en el registro, disminuyendo así el consumo de energía.



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.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
mux_pet																							
reg_pet																							
estado																							
prx_estado																							
logi_salida																							
deco_ET																							
deco_EST																							
deco_DAT																							
m_ET																							
m_EST																							
m_DAT																							
reg_ET																							

