

Schwindt, Ignacio Andrés: Correcciones Práctica 9

Pautas de corrección			
Concepto	Descripción	Máxima nota conceptual	Puntaje máximo
C1	Diseño y simulación del multiplicador y el testbench con capturas de pantalla incluidas.	MB	8 puntos
C2	Determinación de los factores A y B, la frecuencia de reloj y los tiempos de cálculo de $A*B$ y $B*A$.	MB	8 puntos
C3	Utilización del package “Utils” propuesto en la práctica 8.	MB	3 puntos
C4	Calidad de la presentación del informe y los archivos “.vhd” adjuntos.	MB	6 puntos
T	Calificación total obtenida.	MB	25 puntos

Código VHDL:

Las señales “Aux” y “Aux2” son innecesarias: podrían haberse asignado los valores de los factores de cada multiplicación directamente a las señales “A” y “B”.

Tampoco debería requerirse el tiempo de “precarga” de 1 fs para los valores de los factores. ¿Por qué se consideró necesario?

No debería usarse un valor literal para el intervalo de espera entre cada multiplicación (“wait for 20 ns;”). De hecho, podría haberse prescindido por completo de esta demora. Aun así, si se deseaba incluirla, habría sido preferible establecer dicho intervalo a partir de la constante “clk_periodo” establecida para el período de reloj, o sino esperar a la detección del próximo flanco de reloj.

Hay varias asignaciones concurrentes de señales dentro de la arquitectura del multiplicador: “AUX”, “ShifterCant”, “ShifterSum”, “Result” y “Done”. Ésta no era la alternativa más recomendable porque se aleja de la descripción estructural originalmente propuesta para el diseño del multiplicador en la consigna. Convenía generar estas señales desarrollando nuevos componentes independientes e instanciándolos en el multiplicador con las conexiones necesarias.

Es errónea la concatenación de cuatro bits adicionales en cero a la izquierda de los factores A y B para incrementar forzosamente sus longitudes de cuatro a ocho bits: el registro de desplazamiento de N bits correspondiente a la práctica 6 era perfectamente capaz de realizar por sus propios medios esta operación.

Los diseños del multiplicador y la compuerta lógica NOR debían desarrollarse en dos archivos “.vhd” separados para que resultaran independientes entre sí.

La señal “DoneAux” es innecesaria: podía utilizarse en su lugar directamente la señal “Done” sin inconveniente alguno.

No correspondía asignar directamente la salida “LatchOutput” del acumulador como valor de la señal de salida “Result” del multiplicador. El problema aquí es que, como se podrá observar en el Waveform del testbench de AxB , se reflejan en “Result” todos los resultados

correspondientes a las sumas parciales realizadas durante el proceso de cálculo del producto de la multiplicación. Para evitar esto y lograr que “Result” adopte solamente el resultado final, debía utilizarse un registro adicional auxiliar (registro de desplazamiento o latch de 8 bits) con el objetivo de filtrar la salida del multiplicador, actualizándola únicamente una vez que haya finalizado la multiplicación actual.

Podría haberse calculado en el testbench la frecuencia del reloj a partir del número de legajo tal como se solicitaba en el ejercicio 3.

Hay una oscilación inicial en la señal “Result” hasta que comienza a realizarse la multiplicación. Habría sido preferible estabilizarla desde el inicio mismo de la simulación para evitar observar esta inestabilidad en el Waveform.

Informe:

No se ofrece descripción alguna del diseño desarrollado para el multiplicador según se solicita en el ejercicio 1. ¿A qué trabajo práctico corresponde cada uno de los componentes utilizados en el multiplicador provenientes de prácticas previas? ¿Por qué se optó por recurrir a ellos en cada caso? ¿Cómo funciona cada uno por sí mismo en el multiplicador y cómo interactúan entre sí para llevar a cabo la multiplicación?

No se ofrece ninguna descripción del testbench desarrollado para probar el funcionamiento del multiplicador: solamente se incluyó una captura de pantalla del Waveform correspondiente a su ejecución en el simulador.

La descripción que se incluye con respecto al mecanismo para calcular manualmente el resultado de una multiplicación es correcto. Sin embargo, el multiplicador aquí diseñado no opera exactamente de esta misma manera. La principal diferencia radica en que la suma parcial solo se realiza si el bit menos significativo del valor actual del factor A es igual a uno. Por lo tanto, nunca se sumará el valor cero tal como se haría en el cálculo manual. Ergo, no necesariamente si el valor de A es menor al de B la multiplicación será más veloz: B es irrelevante en este escenario. Del mismo modo, no siempre un menor valor de A implica una disminución en el tiempo de cálculo. Aunque podría extrapolarse a cualquier caso general, la diferencia de tiempo entre $A \times B$ y $B \times A$ particularmente calculada aquí no depende tanto de la cantidad de desplazamientos que deban realizarse de A hacia la derecha (posición del bit más significativo en ‘1’ de A) como del número de sumas parciales a efectuarse (cantidad total de bits en ‘1’ de A). De hecho, en el caso aquí simulado, $A \times B$ requiere cuatro desplazamientos y dos sumas, mientras que $B \times A$ solo exige dos desplazamientos y una suma. Por estas razones $B \times A$ se completa más rápidamente que $A \times B$.

En el tercer gráfico se señala como instante de inicio para determinar el tiempo de cálculo de $A \times B$ el momento en el que finaliza el pulso de STB en lugar del instante en el que comienza dicho pulso.

¿Por qué se estableció el tiempo de cálculo para una multiplicación a partir de la activación de STB en lugar de la desactivación de “Done”?



Calificación obtenida		
Concepto	Nota conceptual	Puntaje
C1	B	6 puntos
C2	B+	7 puntos
C3	MB	3 puntos
C4	B	4 puntos
T	B+	20 puntos