

PRÁCTICA 3. Trabajo final

Implementación de un multiplicador secuencial de números naturales.

Objetivos que se deben alcanzar al realizar la práctica

Después de realizar esta práctica, el estudiante será capaz de:

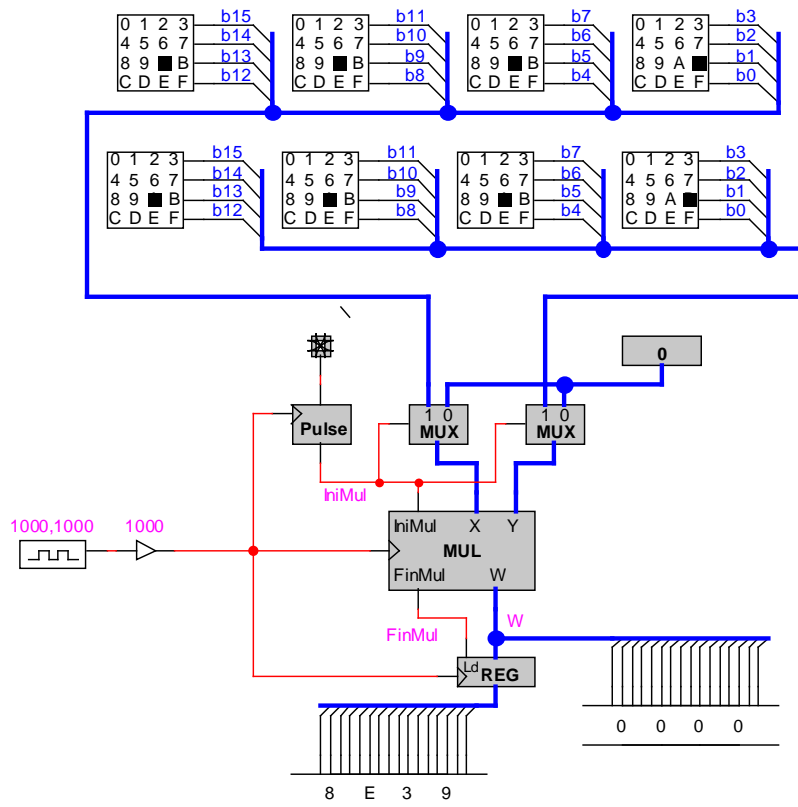
- 1) Multiplicar dos números naturales codificados en binario siguiendo los pasos del algoritmo de multiplicación secuencial basado en desplazamientos y sumas.
- 2) Dibujar el esquema a bloques de la unidad de proceso del multiplicador secuencial y el esquema interno de todos los bloques que la componen.
- 3) Dibujar el grafo de estados de la unidad de control del multiplicador y el esquema de su implementación con el número mínimo de biestables y dos memorias ROM.
- 4) Obtener el tiempo de ciclo mínimo del multiplicador secuencial y el tiempo de cálculo de una multiplicación.

Directorio de la práctica

Mi PC\R:\Logic\Ver_4.1\lc\Prac3

3.1 Circuito para multiplicar dos números naturales

El circuito ProbeMUL que se encuentra en la carpeta Prac3 y que hemos construido para vosotros, es una implementación completa del multiplicador que hemos visto aquí junto con los dispositivos para entrar datos y visualizar el resultado. Abried el circuito con LogicKorks y veréis el circuito que se muestra en la siguiente figura.



Jugad con el circuito para comprobar que funciona correctamente. Para ello, con el simulador corriendo, haced clic en los teclados hexadecimales para dar valores a los vectores de bits de entrada del multiplicando y del multiplicador. Después haced clic en el dispositivo pulsador que hemos construido. Este pulsador alimenta al circuito secuencial Pulse, que genera un pulso de un ciclo de duración sincronizado con el reloj. Lo genera al ciclo siguiente de pulsar. Solamente durante el ciclo que IniMUL vale 1 los números que salen de los teclados hexadecimales están presentes en la entrada del multiplicador, el resto de ciclos hay ceros. Pasados unos ciclos aparecerá el resultado de la multiplicación en el bus de salida del multiplicador. Sólo podemos asegurar que el resultado final de la multiplicación se muestra en la salida durante el ciclo en que FinMul vale 1. Para capturar el resultado y poderlo ver en los displays hexadecimales de salida hemos puesto un registro con su señal de carga conectada con la salida FinMul.

Si queréis observar señales de dentro del circuito usando los cronogramas (ahora las señales a visualizar pueden ser buses y su valor se indica en el cronograma en hexadecimal) encontraréis que el reloj va muy deprisa. Podéis hacer que vaya más despacio modificando la velocidad de simulación con el botón de la barra de herramientas de simulación:



También podéis generar vosotros “a mano” la señal de reloj, quitando el reloj y poniendo un binary switch. Si generáis vosotros la señal de reloj, tened cuidado de no producir el flanco de subida del reloj antes de que la simulación del ciclo actual haya terminado. Mientras LogicWorks simula el funcionamiento del circuito, el contador de pasos de simulación que se encuentra en la barra de herramientas de simulación se va incrementando. Cuando se para este contador ya podéis provocar el siguiente flanco ascendente de reloj.



□ Informe final

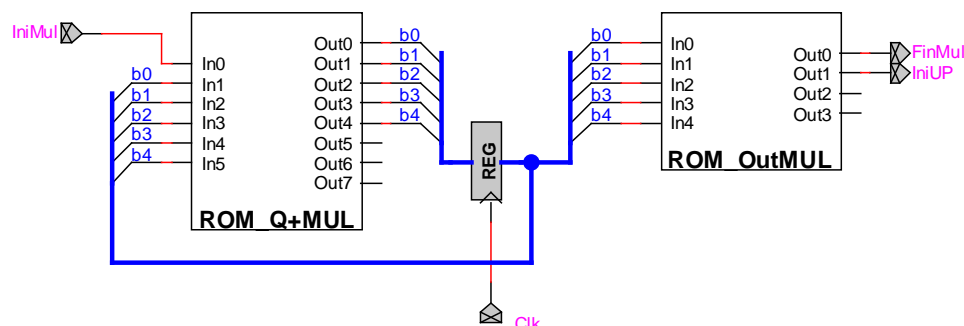
Pregunta 1:

Indicad para cada multiplicación de los dos apartados siguientes cuál es el resultado que se visualiza en los displays hexadecimales de la salida en el ciclo en que FinMul vale 1. Indicad también si el resultado de la multiplicación es incorrecto (resultado no representable en 16 bits), y en caso de no ser correcto, cuál sería el resultado correcto de la multiplicación. Para ello debéis realizar los cálculos vosotros con lápiz y papel y comprobar el resultado que genera el circuito. Las letras 0x antes del vector de dígitos indican que los dígitos son hexadecimales.

- a) X=0x0003, Y=0x0005
- b) X=0x6752, Y=0x0004


3.1.1 Implementación de la Unidad de Control del circuito MUL

Implementad ahora en LogicWorks el grafo que habéis diseñado sobre papel en la pregunta 4 del informe previo usando el número mínimo de biestables (5 biestables, ya que el grafo tiene 18 estados) y dos memorias ROM, una para obtener el estado siguiente, Q^+ , en función de la entrada, IniMul, y del estado actual, Q, y otra para obtener las salidas, IniUP y FinMul, a partir del estado actual, Q. El contenido de cada una de las dos ROM debe ser el que habéis indicado en la pregunta 5 del informe previo. El esquema lógico resultante que tenéis que crear tiene que tener el aspecto que se muestra en la siguiente figura.



Encontraréis el registro de estado, REG, en la LibPrac3. Tened cuidado de colocar la caja/símbolo REG con el bus de entrada y de salida correctamente (El pin Clk y los caracteres REG están más cerca del bus de salida que del de entrada. Esto puede servir para diferenciar los buses). Si al hacer Reset del circuito aparecen x en los bits de entrada a la ROM_OUT, posiblemente habéis puesto el registro al revés. Ponemos un registro de 16 bits, porque es el tamaño que tenemos en la librería, pero sólo usamos los 5 bits de menor peso para codificar el estado de la unidad de control. También encontraréis el dispositivo de la unidad de control (UCMUL) donde debéis encapsular el circuito, ya que el circuito interno del dispositivo UCMUL de la librería sólo dispone de los conectores.

Para construir un dispositivo de tipo ROM tenéis que seguir los pasos que os indica el *PROM/RAM/PLA*

Wizard  que se encuentra en la barra de herramientas del LogicWorks. Después de hacer clic en el icono tenéis que seleccionar *PROM* (ya que la ROM que usamos es realmente una *Programmable ROM*, PROM), en la siguiente ventana poner el número de líneas (número de bits) de dirección de la ROM (*Address Lines*) y de datos (*Bits per Word*), seleccionar *Enter hex data manually* y por último entrar los datos del contenido de la ROM en hexadecimal, separando cada palabra con un espacio o con un return. Podéis hacer clic en *Format Help* para saber otras formas de separar cada palabra del contenido de la ROM. Como las ROM son bastante grandes para entrar los datos manualmente es fácil cometer errores. Hay un problema, que una vez creada una ROM en LogicWorks no se puede ver su contenido ni modificarlo, a no ser que se construya un circuito para ello. Si se quiere cambiar el contenido hay que crear otra ROM. Así, si se produce un error al entrar los datos es muy difícil detectarlo y corregirlo. Podéis escribir el contenido de la ROM en un fichero tipo texto (con el bloc de notas, por ejemplo) y recortar y pegar su contenido en la ventana de entrada de datos manual. Sólo hace falta que peguéis el contenido de las 36 primeras palabras de la ROM del estado siguiente y las 18 de la ROM de las salidas, ya que las posiciones que no se especifican al crear una PROM en LogicWorks se cargan con valores x (no importa). Por último decir que para poder salvar cada ROM tenéis que tener abierta una librería de dispositivos en el escritorio, ya que en el disco donde está LibPrac3 no podéis escribir.

Una vez creadas vuestras dos ROM, copiar el dispositivo UCMUL de la LibPrac3 en una hoja de circuitos en blanco, abridlo (veréis que solo contiene los conectores de entrada salida) y dibujad en su interior el circuito de la unidad de control con las dos ROMs y el registro de estado. Cerradlo y salvad el dispositivo en vuestra copia de la LibPrac3, como ya habéis hecho en prácticas anteriores (Schematics -> Save Part to Library...).

❑ Informe final

Pregunta 2:

Sustituid la Unidad de Control del multiplicador del circuito ProbeMUL por la que vosotros habéis implementado. Comprobad que funciona correctamente con los mismos datos que para la pregunta 1 del informe final. Comprobad que el multiplicador funciona correctamente (no olvidéis hacer clic en el botón de reset del simulador, para inicializar los registros de la unidad de proceso y de la de control). Cuando estéis seguros del correcto funcionamiento del circuito, avisad al profesor de laboratorio y pedidle que revise vuestro trabajo y firme en el informe final.

3.1.2 Tiempo de una multiplicación

❑ Informe final

Pregunta 3:

Para contar los ciclos que tarda la multiplicación podéis añadir la señal del reloj al cronograma de la ventana de tiempos de LogicWork.

- Si en el ciclo c la señal IniMul vale 1 y en el ciclo $c + m$ la señal FinMul vale 1 ¿Cuánto vale m (numero de ciclos del cálculo de una multiplicación)? ¿Cuál es el tiempo de cálculo para multiplicar dos números de 16 bits con el multiplicador que hemos diseñado usando el reloj con el tiempo de ciclo mínimo? ¿Cada cuántos ciclos puede iniciarse una nueva multiplicación (número mínimo de ciclos)?
- Si en el ciclo c la señal IniMul vale 1 y el resultado correcto de la multiplicación se ve en el bus W en el ciclo $c + k$ ¿Cuántos vale k para la multiplicación del ejemplo a) de la pregunta 1 del informe final? ¿Por qué puede verse bastantes ciclos antes de que la señal de validación del resultado, FinMul, valga 1?

Informe final Práctica-3

Apellidos y nombre: Grupo:

Apellidos y nombre: Grupo:

(por orden alfabético)

Pregunta 1: (1 punto)

a) (0.5 puntos)

b) (0.5 puntos)

Pregunta 2: (6 puntos)

Comentario del profesor:

Firma del profesor:

Pregunta 3: (3 puntos)

a) (1.5 puntos)

b) (1.5 puntos)