Tecnológico de Costa Rica
Escuela de Ingeniería Electrónica
Curso: Diseño Lógico

Proyecto final – FSM y máquina micro programada

Algoritmo por resolver

La multiplicación de números binarios se realiza por medio de una **secuencia** de sumas y corrimientos:

	2	3						**1**	**0**	**1**	**1**	**1**		Multiplicando
x	1	9					x	**1**	**0**	**0**	**1**	**1**		Multiplicador
								1	**0**	**1**	**1**	**1**		
							1	**0**	**1**	**1**	**1**			
						1	0	0	0	1	0	1		
						0	**0**	**0**	**0**	**0**				
						1	0	0	0	1	0	1		
					0	**0**	**0**	**0**	**0**					
					0	1	0	0	0	1	0	1		
				1	**0**	**1**	**1**	**1**						
4	3	7		**1**	**1**	**0**	**1**	**1**	**0**	**1**	**0**	**1**		producto

El algoritmo consiste en sumar sucesivamente copias con corrimiento del multiplicando. Estas sumas dependen de examinar sucesivamente los bits del multiplicador, comenzando con el menos significativo. Si el bit bajo observación del multiplicador es 1, el multiplicando son usados en la suma, de lo contrario se usa un vector de ceros. Estos vectores se les aplica un corrimiento de una posición con respecto al operando anterior. El resultado de una multiplicación puede llegar a tener el doble de bits que los operandos.

Otra forma de observar el algoritmo es por medio de un vector "resultado" al que se le aplica un corrimiento a la derecha en cada iteración (el bit en Rojo es el que se usa para tomar decisiones). Se realizan ***n*** iteraciones, donde ***n*** es el número de bits de los operandos (en este ejemplo n=5).

Vector de resultado										
Parte alta						Parte baja				
0	0	0	0	0	0	0	0	0	0	0
0	1	0	1	1	1	0	0	0	0	0
0	0	1	0	1	1	1	0	0	0	0
1	0	0	0	1	0	1	0	0	0	0
0	1	0	0	0	1	0	1	0	0	0
0	0	1	0	0	0	1	0	1	0	0
0	0	0	1	0	0	0	1	0	1	0
0	1	1	0	1	1	0	1	0	1	0
0	**0**	**1**	**1**	**0**	**1**	**1**	**0**	**1**	**0**	**1**

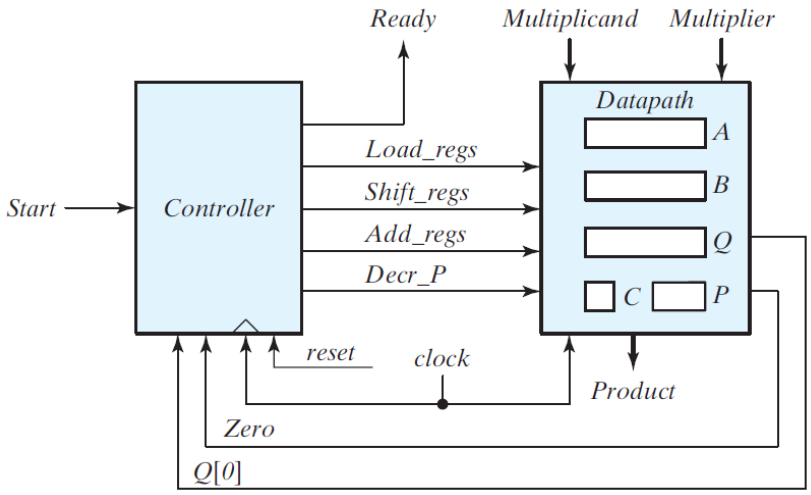
Multiplicador					iteración	Multiplicando: 10111
1	0	0	1	**1**	0	LSB = 1, sumar Multiplicando a parte alta
1	0	0	1	**1**		Correr a la derecha ambos
0	1	0	0	**1**	1	LSB = 1, sumar Multiplicando a parte alta
0	1	0	0	**1**		Correr a la derecha ambos
0	0	1	0	**0**	2	LSB = 0, solo Correr a la derecha ambos
0	0	0	1	**0**	3	LSB = 0, solo Correr a la derecha ambos
0	0	0	0	**1**	4	LSB = 1, sumar Multiplicando a parte alta
0	0	0	0	**1**		Correr a la derecha ambos
0	0	0	0	0		RESULTADO

Objetivo

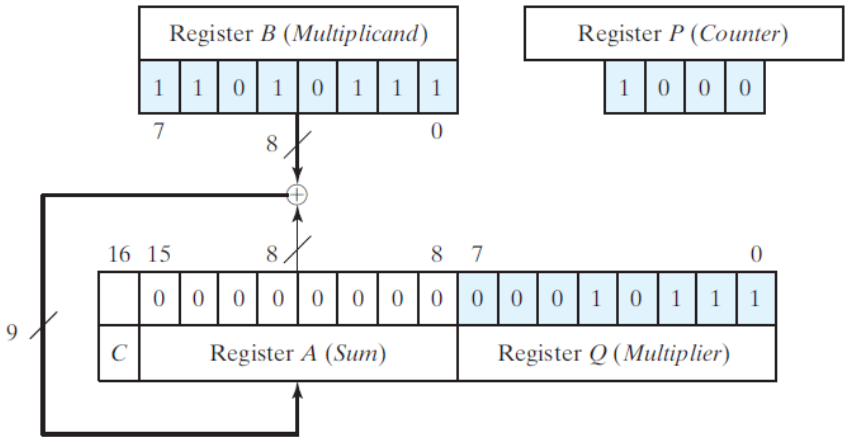
Desarrollar una unidad que permita el calculo de multiplicaciones binarias con **operandos de 8 bits.**

Procedimiento

1. El diagrama de bloques muestra la estructura general (control y ruta de datos) a utilizar para resolver el problema. De esta manera, tiene que dividir el trabajo en el desarrollo de estos dos bloques.



2. Como se muestra en el diagrama siguiente, la ruta de datos se compone de cuatro registros y un contador.
    a. El registro B debe almacenar el multiplicando.
    b. El registro Q debe almacenar el multiplicador
    c. El registro A junto con el registro C (de un bit) almacenan la salida del sumador.
    d. El conjunto de registros C, A, Q, almacenan el resultado final de la multiplicación.
    e. P es un contador
3. ¿Qué características requieren tener los registros en este sistema?



4. De los diagramas anteriores note que:
    a. Load_regs: es una señal que le indica a los registros que carguen un dato.
    b. Shift_regs: Es una señal que le indica que se debe ejecutar un corrimiento.
    c. Add_regs: Es una señal que indica que se debe almacenar el resultado de una suma.
    d. Decr_P: Es una señal que indica que se debe decrementar el contador P.
    e. Zero: Indica que el contador P llegó a un valor de cero.
    f. Q[0]: es el bit más bajo del registro Q.
    g. Start: es una señal que indica que el algoritmo se debe ejecutar.
    h. Ready: es una señal que indica que el algoritmo finalizó.

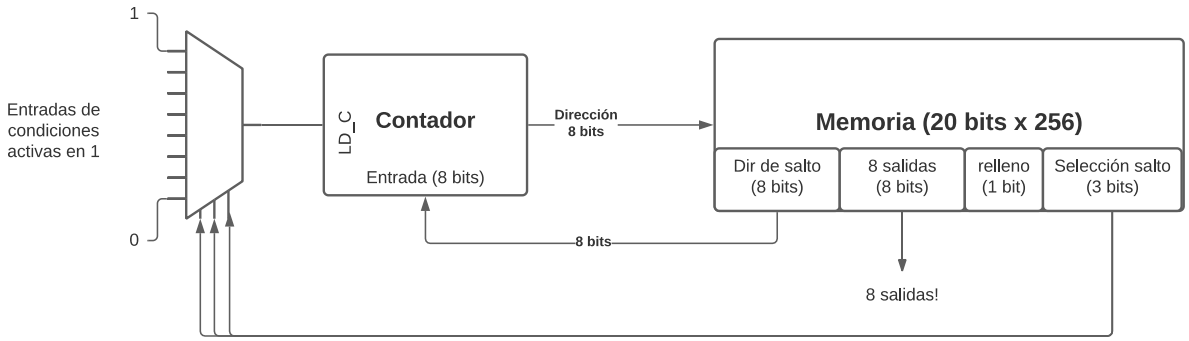
Parte 1: RUTA DE DATOS

1. Defina los requisitos de cada uno de los elementos de la *ruta de datos*.
2. Escriba el código verilog que implemente cada uno de los sub-bloques de la *ruta de datos*.
    a. Asegúrese de verificar la funcionalidad
3. Escriba el código verilog que implemente (de forma estructural) el bloque de *ruta de datos* por medio de los bloques desarrollados anteriormente.
4. Verifique la funcionalidad con un testbench.

Parte 2: UNIDAD DE CONTROL – FSM

1. Proponga un diagrama de flujo para la unidad de control.
    a. Básese en los diagramas de este enunciado y en las señales indicadas en este.
2. Genere un diagrama de estados para su máquina de estados. Haga uso de una arquitectura MOORE.
3. Escriba el código verilog que implemente su diagrama de estados.
4. Verifique la funcionalidad de su FSM.
5. Escriba en verilog una unidad superior (TOP) que una la unidad de control FSM con la ruta de datos en un solo bloque.
6. Escriba un testbench que compruebe que puede realizar multiplicaciones con operandos de 8 bits.

Parte 3: UNIDAD DE CONTROL – MICROPROGRAMADA



1. Defina que tipo de instrucciones puede ejecutar por medio de la arquitectura micro programada provista con este instructivo. Revise las clases correspondientes a este tema.
    a. La memoria ROM tiene un ancho de 20 bits y un largo de 256 palabras. Cada palabra se encuentra dividida de la siguiente manera:
        i. Bits 2:0 (bits más bajos): Corresponden a la selección del mux de salto.
            1. Si la selección es 000_2 la salida del mux es siempre 0.
            2. Si la selección es 111_2 la salida del mux es siempre 1.
        ii. Bit 3: No se usa (puede dejarse en cualquier valor, se recomienda cero)
        iii. Bits 11:4 (ocho bits): corresponden a 8 posibles salidas de la máquina.
        iv. Bits 19:12 (ocho bits): Corresponden a la dirección de salto (cuándo se requiere)
2. Modifique su diagrama de flujo / estados para que pueda ser ejecutado por la arquitectura micro programada.
3. Desarrolle el microcódigo que resuelva el algoritmo.
4. Escriba en verilog una unidad superior (TOP) que una la unidad de control micro controlada con la ruta de datos en un solo bloque.
5. Escriba un testbench que compruebe que puede realizar multiplicaciones con operandos de 8 bits. ¿Puede reutilizar el testbench de la parte anterior?

Evaluación

1. El trabajo se realizará de forma grupal (3 personas)
2. **La entrega será el 11 de noviembre.**
3. El reporte del trabajo consistirá en un reporte corto estilo IEEE de máximo 4 páginas a doble columna.

   https://www.ieee.org/conferences/publishing/templates.html

4. Además, deben de preparar una presentación con diapositivas y un video del grupo presentando el proyecto (no más de 15 minutos).
   a. La presentación debe mostrar el proceso de diseño seguido. Mostrar todos sus cálculos y diagramas
   b. Además, deben incluir una demostración del sistema tanto en máquina de estados como en máquina microprogramada.
   c. Para las conclusiones hacer énfasis en las ventajas y desventajas que ven en ambas metodologías de diseño.
5. Desglose:

   El proyecto tiene un valor de 20% de la nota.

   a. Reporte escrito y código: 14%
      i. Código sin errores de sintaxis o lógicos (sintetizable).
      ii. Proceso de diseño: 4%
      iii. Diagramas de bloques: 1%
      iv. Diagramas de flujo y estados: 3%
      v. Resultados: 1%
      vi. Conclusiones: 1%
   b. Presentación:6%
      i. Claridad de la explicación: 2%
      ii. Demostración de funcionalidad: 2%
      iii. Calidad de la presentación 2%