

Tarea #3

Multiplicador Iterativo de 32 bits

JUAN SÁNCHEZ, B16068

LUIS DIEGO SOTO, B26613

ERICK DUARTE, B22305

IE-0521: Estructuras de Computadores II

Escuela de Ingeniería Eléctrica

Universidad de Costa Rica

I. INTRODUCCIÓN

Este trabajo tiene como fin desarrollar habilidades de diseño y análisis de circuitos combinatoriales y secuenciales mediante el lenguaje de descripción de hardware Verilog. Lo cual se encuentra estrechamente relacionado con la materia del curso **estructuras de computadoras** desde la perspectiva de diseño de hardware.

Respecto a la distribución de tiempos el laboratorio fue completado exitosamente, sin embargo, los tiempos invertidos en cada etapa del proyecto variaron respecto a lo planeado inicialmente.

Finalmente, se implementó el multiplicador de 2 números de 32 bits (sin signo) a través de los 2 módulos principales propuestos:

1. **Datapath:** Se encarga de llevar los datos a través de los registros, desplazadores, sumadores y muxes.
2. **Controlador:** Es una máquina de estados encargada de manejar el funcionamiento del datapath.

El diseño implementado completó satisfactoriamente las pruebas que se le realizaron y , por lo tanto, se concluyó que el multiplicador descrito en Verilog cumplió las funciones para las cuales fue concebido.

II. MANEJO DEL PROYECTO

La distribución de roles en el proyecto se efectuó de la siguiente manera:

- **Luis Diego Soto:** Fue el arquitecto del equipo, aunque también desempeñó labores de diseñador y verificador, enfocadas al módulo del controlador del datapath. Conforme el proyecto avanzó se integró al proceso de verificación del multiplicador.
- **Erick Duarte:** Fue el Diseñador principal y tuvo a cargo el módulo del datapath en cuanto arquitectura y diseño. También formó parte del proceso de verificación del datapath y conforme al proyecto avanzó se integró al proceso de verificación del multiplicador.

- **Juan Sánchez:** Fue el verificador del equipo y tuvo a cargo la integración de los 2 módulos para armar el multiplicador de 32 bits, también fue el encargado de elaborar el testbench del controlador, el datapath y multiplicador.

Por otra parte, como se mencionó anteriormente los plazos de tiempo tuvieron variaciones desde el inicio hasta el final del proyecto. Las figura 1 y 2 muestran los diagramas de Gantt referentes al inicio y final del proyecto.

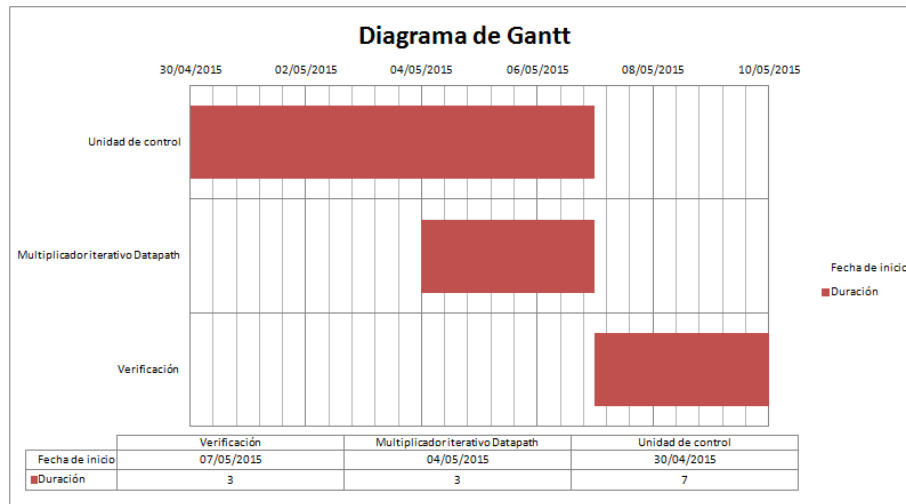


Figura 1: Diagrama de Gantt inicial

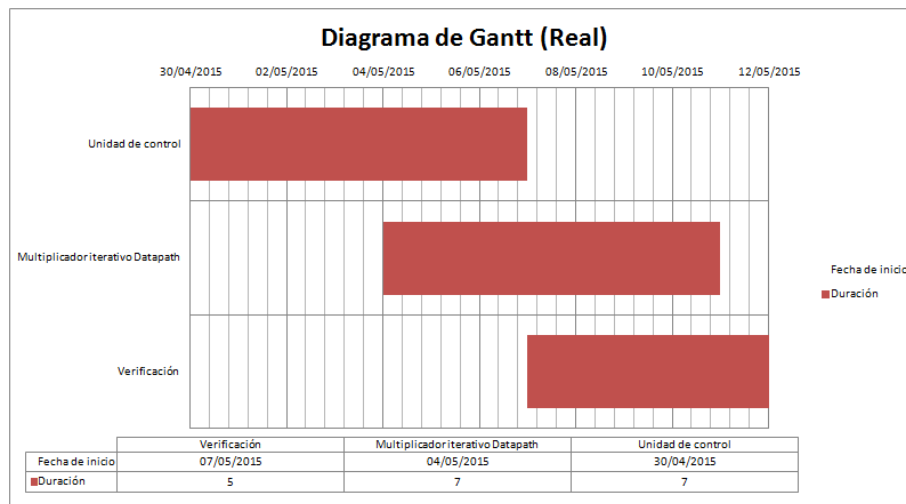


Figura 2: Diagrama de Gantt Real

Al inicio del proyecto se pensó en invertir 7 días en la máquina de estados, 3 días en el multiplicador y el datapath y 3 días en verificación del módulo completo. No obstante, conforme se inicio el proceso La unidad de control si estuvo diseñada y verificada en 7 días, pero con el datapath se tardaron 7 días en completar el módulo y tenerlo funcionando optimamente. Al tener

el controlador a un 100% de funcionalidad y el datapath en desarrollo se iniciaron el proceso de integración de ambos módulos así como la construcción del testbench para el multiplicador completo. La verificación del módulo completo tomó 5 días.

Finalmente, del 12 al 14 de Mayo se realizó la documentación para el reporte entre los 3 miembros del equipo.

III. DISEÑO

La implementación del multiplicador iterativo de 32 bits consideró los módulos de la siguiente manera:

I. Datapath

Para la parte del Datapath se tienen básicamente los módulos necesarios para realizar una multiplicación con las entradas adecuadas en el tiempo adecuado, lo cual es proporcionado y sincronizado por el controlador. En este módulo se utilizaron varios otros módulos como registros, multiplexores y registros desplazantes de 32 y 64 bits, además de un sumador de 64 bits.

El algoritmo seguido para la implementación de este módulo consiste en que se tienen 2 entradas de 32 bits cada una, para uno de los datos (B) se van desplazando los bits hacia la derecha, y cada vez que el bit menos significativo de esta palabra sea 1 se manda una señal que diga si se debe sumar o no el valor del dato A al subproducto actual tal y como se ve en la figura 3. Además en cada iteración el dato A se va desplazando hacia la izquierda para aumentar su magnitud y así lograr el producto correcto.

Nótese que a pesar de que el dato A que entra en el datapath es de 32 bits, la salida del primer multiplexor de A es de 64 bits, esto se debe a que al ir desplazándose hacia la izquierda, si es de 32 bits se pierden algunos de los bits por lo que no daría el resultado correcto, por lo cual se le asignaron otros 32 bits en la parte alta con un valor de 0 y así además se completan todos los bits para los sumandos a la hora de obtener el subproducto.

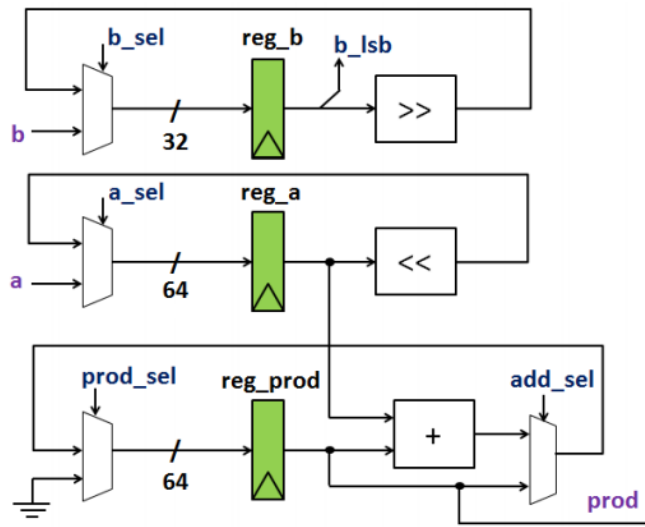


Figura 3: Estructura del Datapath

Es importante notar que en este tipo de algoritmo la lectura del resultado de la multiplicación de 2 bits, se debe esperar cierto tiempo para obtener el resultado correcto ya que si se pretende leer el resultado demasiado rápido, lo que se obtendrá son los subproductos de cada iteración del multiplicador.

El producto correcto estará listo en la salida justo después de que se termine de desplazar hacia la derecha el dato B (Segunda entrada del multiplicador), es decir, cuando todo el dato B desplazado termine con todos sus bits en 0, ya que el subproducto solo cambia cuando el bit menos significativo del dato B desplazado sea 1 y si este dato está compuesto de ceros el producto se estabilizará en el valor final de la multiplicación.

II. Controlador

En la figura 4 se observa el diagrama de flujo del controlador del multiplicador.

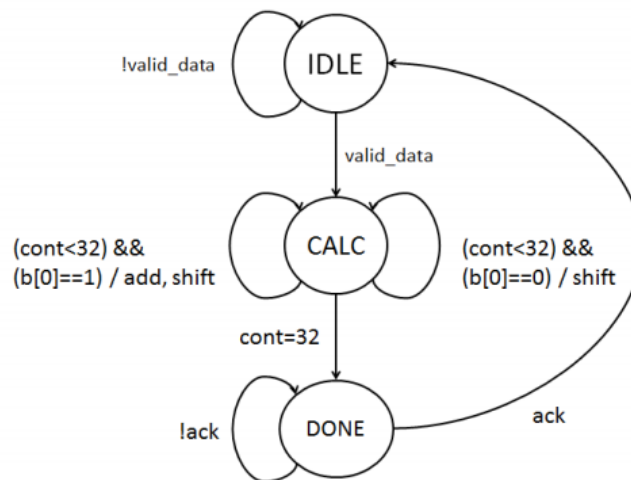


Figura 4: Controlador

La máquina cuenta con 3 estados:

1. **Estado IDLE:** es el estado por defecto del controlador. Se continuará en este estado hasta tener una entrada válida al Datapath.
2. **Estado CALC:** En este estado se lee el valor del bit menos significativo del registro B y dependiendo de su valor se suma el registro A al producto o no. También se desplazan los registros. Se estará en este estado hasta que se hayan corrido los registros 32 veces.
3. **Estado DONE:** Una vez que se ha realizado la multiplicación, el producto se mantendrá hasta que el controlador reciba una señal que indique que el dato ha sido leído.

IV. ESTRATEGIA DE PRUEBAS

La estrategia de pruebas se realizó entre ejes principales: test del controlador, test del datapath y test del multiplicador. Todas fueron realizadas exitosamente y se utilizó modelo de caja negra, es decir se estimularon las entradas y se verificó que las salidas fueran las esperadas.

El controlador se verificó de manera independiente y una vez listo se utilizó como parte del test del datapath para chequear que las respuestas a los estímulos fuera la esperada. Una vez que tanto el controlador y el datapath pasaron las pruebas se integraron ambos módulos y se sometió a verificación el multiplicador completo.

El cuadro 1 muestra los 3 grupos de pruebas macro realizadas.

Cuadro 1: Pruebas realizadas

Prueba	Resultado
Verificación del Controlador	Pasó
Verificación del Datapath	Pasó
Verificación del multiplicador	Pasó

Para las pruebas del datapath y el multiplicador se reslizaron las multiplicaciones mostradas en el cuadro 2. La pruebas del multiplicador se centraron en casos extremos y al azar.

Cuadro 2: Pruebas realizadas al multiplicador

Prueba	Resultado
$4294967295 \times 4294967295$	1.8446744×10^{19}
4294967295×0	0
10000×4500	45000000

V. EVALUACIÓN

En la imagen 5 se muestra el resultado del testbench para el controlador. El marcador A muestra el punto donde el reset vuelve a cero, el marcador B muestra el cambio de iValid_data a 1, el marcador C, muestra que un ciclo de reloj después el controlador pasa al estado 1 (CALC). Luego una vez que rCounter=32, el siguiente estado será 2 (DONE) como muestran los marcadores D y E. Finalmente una vez que la señal iAck se coloca en 1, en el siguiente ciclo de reloj el controlador vuelve al estado cero (IDLE).

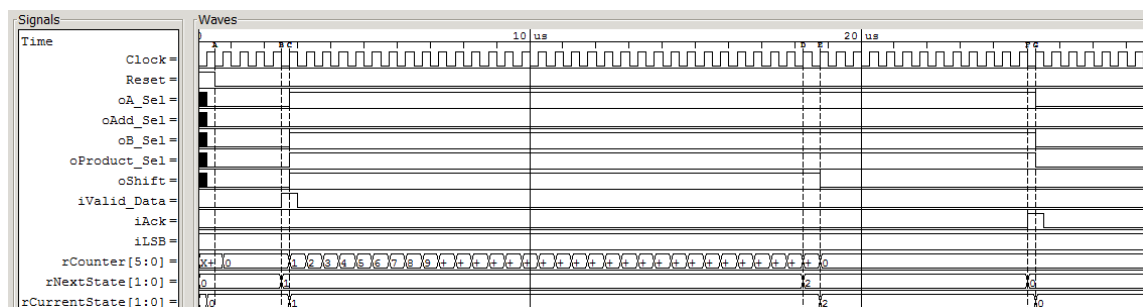


Figura 5: Testbench del controlador

Para el testbench del datapath y el multiplicador probaron 2 casos extremos; **multiplicación por cero** y **multiplicación de $4294967295 \times 4294967295$** (número sin signo más grande que se puede representar con 32 bits). Adicionalmente se agregó la multiplicación de 2 números al azar.

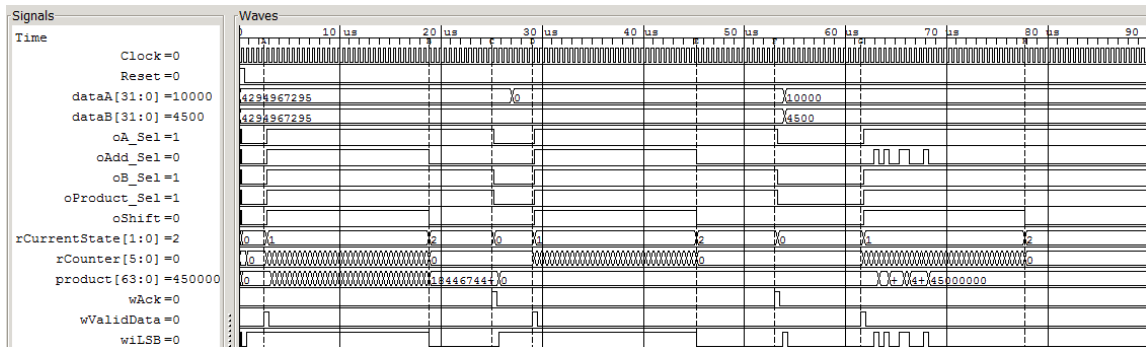


Figura 6: Testbench del datapath y multiplicador

En la figura 6 se muestra inicialmente el marcador A, el cual denota la validación de las entradas (en este caso ambas son iguales a 4294967295). El marcador B señala la entrega del resultado de la multiplicación (1.8446744×10^{19}), el marcador C muestra la vuelta al estado IDLE. Luego se carga una de las entradas a cero y se validan las entradas (marcador D), posteriormente el marcador E denota la entrega del resultado (cero en este caso) y el marcador F la vuelta al estado IDLE.

Finalmente se repite el proceso con 2 números aleatorios (10000 y 4500). El marcador G denota la validación de las entradas y el paso de IDLE a CALC; asimismo el marcador H denota la entrega del resultado (45 000 000).