BLOQUE MULTIPLICADOR

Laura Navarro Galván, Andrés Suárez Prada Resumen— El diseño y construcción de un procesador, es llevado a cabo en el curso de Arquitectura de computadores. En este trabajo, se evidencia el diseño utilizado para el bloque de multiplicación que contiene dicho procesador.

I. INTRODUCCIÓN

Dentro de la elaboración de un procesador, es necesario la construcción de un bloque de multiplicación. Dicho bloque debe ser capaz de multiplicar dos números (32 bits) en la menor cantidad de ciclos de reloj posibles, para lograr esto, este multiplicador utiliza el algoritmo de Booth junto con el algoritmo de Karatsuba.

II. MÉTODOS

A. Algoritmo de Booth

El algoritmo de Booth sirve para multiplicar números binarios con signo de manera rápida y sencilla en complemento a dos (Ilustración 1). Este algoritmo examina pares adyacentes de bits del multiplicador R2 de N-bits, incluyendo un bit implícito debajo del bit menos significativo, $R2_{-1} = 0$. Para cada bit $R2_i$, donde i abarca desde 0 hasta N-1, los bits menos significativos $R2_0$ e $R2_1$ son considerados y se establecen los casos de la siguiente forma:

$R2_{\theta}$	$R2_{-1}$	Operación			
0	0	No se realiza ninguna operación			
0	1	R3 = R3 + R1 (Cadena de 0's)			
1	0	R3 = R3-R1 (Cadena de 1's)			
1	1	No se realiza ninguna operación			

En cada iteración se incrementa el contador (CONT), se desplaza C-R3-R4 a la derecha y se rota circularmente a la derecha R2.

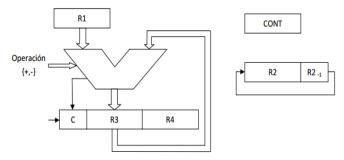


Ilustración 1. Modelo Algorítmico Booth para un Multiplicador.

B. Algoritmo de Karatsuba

El fundamento principal del algoritmo es llevar a cabo la multiplicación de dos números grandes, mediante la subdivisión de los datos de entrada, es decir, el multiplicando y el multiplicador se seccionan en datos más pequeños haciendo que la multiplicación original se descomponga en tres multiplicaciones más pequeñas.

Si rs1 y rs2, están representados como cadenas de n dígitos en alguna base B. Para cualquier entero positivo m menor que n, se puede dividir los dos números dados de la siguiente manera:

$$rs 1 = X 1B^{m} + X 0$$

$$rs 2 = Y 1B^{m} + Y 0$$

Donde X0 y Y0 son menores que B^m , entonces:

$$rs1rs2 = (X1B^m + X0)(Y1B^m + Y0) = Z2B^{2m} + Z1B^m + Z0$$

Y por agrupación de términos se puede expresar el producto

final como: Z = X 1 Y 1Z = X 0 Y 0

Z1 = (X1 + X0)(Y1 + Y0) - Z2 - Z0

Que implica una serie de sumas y desplazamientos que de hecho no interfieren en el tiempo de respuesta del circuito pues dichas operaciones implican un tiempo de reloj inferior al producto entre dos numeros grandes.

III. RESULTADOS Y ANÁLISIS

Al momento de usar el algoritmo de Booth, este tarda 35 ciclos de reloj en realizar el proceso de multiplicación, pero

cuando este algoritmo es combinado con el algoritmo de karatsuba, este tiempo de respuesta es disminuido drásticamente a 19 ciclos de reloj (Anexo A.).

El diseño de esta multiplicación (Ilustración 2), consta de 6 entradas y 2 salidas:

- 1. clk: el reloj a una frecuencia de 500 MHz.
- 2. Reset: reinicio de la multiplicación
- 3. rs1: multiplicando (32 bits)
- 4. rs2: multiplicador (32 bits)
- 5. Enable: activa la multiplicación, es decir, la multiplicación no se ejecutará hasta que no se dé la orden (activo en alto)
- funct3: Asigna la clase de multiplicación MulH[[S]U] de acuerdo al estándar de RV32M, la cual indica:

funct7	rs2	rs1	funct3	r	opcode
				d	
MUL rd,rs1,rs2	rs2	rs1	00	rd	0110011
MULH rd,rs1,rs2	rs2	rs1	01	rd	0110011
MULHSU rd,rs1,rs2	rs2	rs1	10	rd	0110011
MULHU rd,rs1,rs2	rs2	rs1	11	rd	0110011

- 7. rd: salida del multiplicador (64 bits).
- 8. Busy: señal de salida que indica cuando el bloque está disponible para una nueva multiplicación.

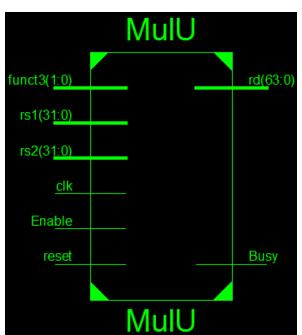


Ilustración 2. Bloque multiplicación

IV. CONCLUSIONES

Usando los dos algoritmos en conjunto no solo muestran una eficacia igual a cualquier conjunto de funciones que tengan el objeto de multiplicar números directamente, sino que también los ciclos de reloj utilizados son inferiores, esto se debe a que los dos procesos se formulan para optimizar el tiempo de respuesta del sistema de multiplicación para circuitos digitales, pero su simplificación se ve limitada por los recursos del procesador, ya que se deben realizar tres multiplicaciones en paralelo, utilizando recursos adicionales del mismo.

ANEXOS

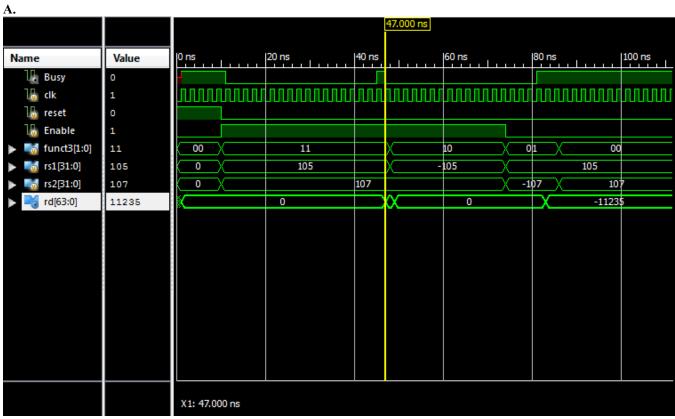


Ilustración 3. Simulación bloque multiplicación