



ARQUITECTURA DE COMPUTADORAS

Trabajo Práctico N° 5

Implementación de las Operaciones Básicas: Multiplicación ¹

Primer Cuatrimestre de 2018

Ejercicios

1. Realizar las siguientes multiplicaciones ($X * Y$) en *dos-complemento* para los siguientes pares X, Y . Explicar en cada caso los pasos realizados y dar la justificación correspondiente a las correcciones realizadas según el caso. Trabajar con operandos de $n = 5$ bits.

- a) 3×5
- b) -3×5
- c) 3×-5
- d) -3×-5

Nota: Si $X < 0$, trabajar con *extensión virtual de signo*, es decir modificar lo que entre por izquierda en los movimientos a derecha, ingresando el signo (X_{n-1}) luego de la primer suma. Si $Y < 0$ realizar la corrección del último paso.

2. Dado el proceso de multiplicación en *dos-complemento*:
 - a) Esquematizar el hardware de multimplicación.
 - b) Asumiendo que la ALU de suma demora un tiempo T en realizar la suma. Calcule el tiempo que demoran las multiplicaciones del ejercicio 1 incisos b y c.
3. Se desea utilizar recodificación básica de a c bits para reducir el número de iteraciones de un multiplicador. Considerando que X es el multiplicando e Y es el multiplicador:
 - a) Determinar el número de múltiplos de X que serán necesarios para poder procesar c bits en cada iteración.
 - b) Determinar el número de iteraciones resultantes para operandos de n bits.
 - c) ¿ Cuántas iteraciones y cuántos múltiplos se requieren para $n = 12$ bits con $c = 2$ y $c = 3$?
 - d) Para cada uno de los siguientes multiplicadores (Y s), hacer explícitos los productos necesarios para los siguientes operandos de $n = 12$ bits, con $c = 2$ y $c = 3$ y comprobar el resultado calculando la suma matemática de dichos productos.
 - 1) $Y = 172$
 - 2) $Y = 1287$
 - e) ¿Qué desventaja puede presentar esta alternativa de recodificación en cuanto al costo de obtención de los productos parciales?
4. Utilizando la técnica de *recodificación de Booth*. Hacer explícitos los productos parciales, asumiendo en este caso operandos de seis bits.

¹Fecha sugerida de finalización de este trabajo práctico: martes 24 de abril de 2018.

a) 5×-19

b) -7×13

En base a los resultados, dejar expresado el valor de MQ (Y recodificado) que se utilizaría para realizar la multiplicación.

5. Se desea operar utilizando *recodificación de Booth* para *radix 4* ($\text{radix-}2^c$) es decir que recodificando de a $c = 2$ bits:

a) Construir la tabla correspondiente a la recodificación de Booth radix-4 **mirando al futuro**, en la cual se deben tener en cuenta los dos bits a recodificar (b_{i+1}, b_i) y el último bit de la parte aun no codificada b_{i+2} . Especificar los múltiplos del multiplicando requeridos previos a la realización de la multiplicación.

b) Construir la tabla correspondiente a la recodificación de Booth radix-4 **mirando al pasado**, en la cual se deben tener en cuenta los dos bits a recodificar (b_{i+1}, b_i) y el último bit de la parte codificada b_{i-1} , el cual se inicializa en 0 antes del primer paso. Especificar los múltiplos del multiplicando requeridos previos a la realización de la multiplicación.

c) Hacer explícitos los productos parciales, asumiendo operandos de ocho bits. Utilizar las dos técnicas de recodificación de Booth (mirando al pasado y mirando al futuro).

1) 45×-39

2) -12×65

6. Sean los enteros en *dos-complemento* $X = 10010010$ e $Y = 11011101$:

a) Calcular el producto $X \times Y$ trabajando con el algoritmo de multiplicación básico para enteros signados, haciendo las correcciones que correspondan.

b) Determinar los distintos múltiplos de X requeridos previo a la realización de la multiplicación al hacer uso de la recodificación de Booth sobre Y , agrupando de a *dos bits* (en otras palabras, *radix 4*), siguiendo el esquema *look-ahead*. Tener presente que bajo este esquema se deben analizar los dos bits a recodificar $(b_i$ y $b_{i+1})$ junto a el último bit de la parte aún no codificada (b_{i+2}) .

c) Rehacer el inciso anterior, en esta oportunidad siguiendo el esquema *look-behind*. Tener presente que bajo este esquema se deben analizar los dos bits a recodificar $(b_i$ y $b_{i+1})$ junto con el último bit de la parte ya codificada (b_{i-1}) . ¿Cambian los múltiplos de X que se deben calcular previo a la multiplicación? ¿Cuáles resultan más fáciles de obtener?

7. Se desea operar utilizando la recodificación de Booth radix-8, es decir recodificando de a tres bits.

a) Construir la tabla correspondiente a la *recodificación de Booth* radix-8 **mirando al pasado**, donde se debe tener en cuenta el último bit de la parte ya codificada y los tres bits a codificar en el paso actual, es decir (b_{i+2}, b_{i+1}, b_i) que son los que se codifican y b_{i-1} que es el último bit ya codificado ¿Qué puede afirmar con respecto a los múltiplos requeridos previos a la multiplicación en este caso en comparación con los requeridos en la recodificación estándar?

b) Construir la tabla correspondiente a la *recodificación de Booth* radix-8 **mirando al futuro**, donde se debe tener en cuenta el último bit de la parte aun no codificada

y los tres bits a codificar en el paso actual, es decir (b_{i+2}, b_{i+1}, b_i) que son los que se codifican y b_{i+3} que es el último bit de la parte aun no codificada ¿Qué puede afirmar con respecto a los múltiplos requeridos previos a la multiplicación en este caso en comparación con los requeridos en la recodificación **mirando al pasado**?

8. Esquematizar un *Wallace Tree* con CSAs para resolver el producto de dos operandos de 16 bits. ¿Cuántos CSAs se requieren? ¿Respeto la cota presentada en la teoría? ¿Cuántos niveles de CSAs hacen falta?
9. Esquematizar un *Wallace Tree* con CSAs para resolver el producto de dos operandos de *ocho bits*, indicando como quedan los operandos de entrada al árbol al suponer $X = 89$ e $Y = 55$. Determinar los operandos intermedios y finales a la salida de los CSA y efectuar también la suma paralela a manera de verificación.
10. Esquematizar un *Wallace Tree* con CSAs para resolver el producto de dos operandos $X = 67$ e $Y = -12$ de *ocho bits*, recodificando de a *dos bits por vez* (esto es, tomando como entrada a los distintos CSAs al producto $Y_{i+1}Y_i \times X$), indicando claramente la entrada a ser recibida por los distintos CSAs del primer nivel. Bajo esta nueva configuración, ¿cuántos niveles de CSAs hacen falta? ¿Qué ganancia se obtiene al recodificar de a varios bits? ¿Qué costo adicional aparece?
11. Bosquejar un *árbol binario de multiplicación* que permita calcular el producto de dos operandos $X = 16$ e $Y = 24$ de *ocho bits* al utilizar *dígito signado*, indicando a su vez como quedan los operandos de entrada al árbol y a los nodos interiores. A manera de verificación, resolver la suma paralela que convierte a binario el resultado.

Referencias

- [Bae80] BAER, J. L. *Computer Systems Architecture*. Computer Science Press, 1980.
- [HP96] HENNESSY, J., AND PATTERSON, D. *Computer Architecture*, second ed. Morgan Kaufmann, 1996.