



ARQUITECTURA DE COMPUTADORAS PARA INGENIERÍA

Trabajo Práctico N° 3

Implementación de las Operaciones Básicas: Multiplicación y División

Primer Cuatrimestre de 2016

Ejercicios

1. Considerando la siguiente técnica de multiplicación para enteros signados en notación complemento a dos:

*Si el **multiplicando es negativo**, trabajar controlando la entrada serie (0 o x_{n-1} luego de la primer suma) y cuando el **multiplicador es negativo**, realizar la corrección del último paso.*

mostrar la evolución del cómputo y el resultado obtenido utilizando hardware de 8 bits, para cada uno de los siguientes casos:

- a) 7×9
 - b) -7×6
 - c) 7×-6
 - d) -7×-9
2. Efectuar los siguientes productos utilizando la técnica de *recodificación de Booth*, trabajando en complemento a dos. Resuelva la operatoria utilizando hardware de 8 bits, mostrando la evolución del cómputo y haciendo explícitos los productos parciales generados.

- a) 9×-23
- b) -7×-9

¿Resulta necesario algún paso adicional de corrección en el caso de que el multiplicando y/o multiplicador sea negativo?

3. Sean los enteros signados en notación complemento a dos $X = 10010010$ e $Y = 11011101$:
 - a) Calcular el producto $X \times Y$ trabajando con el algoritmo de multiplicación básico, haciendo las correcciones que correspondan sobre los multiplicandos y/o el resultado.
 - b) Determinar los distintos múltiplos de X que hacen falta al hacer uso de la recodificación de Booth sobre el multiplicador, en grupos **de a dos bits**, siguiendo los esquemas *mirando al pasado (look-behind)* y *mirando al futuro (look-ahead)*.
 - c) Calcular los productos parciales en ambos casos y verificar el producto acumulado.

- d) Computar el producto utilizando el esquema *look-behind*.
4. Se desea operar utilizando la *recodificación de Booth* radix 4, esto es recodificando de a **dos** bits.
- 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 aún no codificada (b_{i+2}). Especificar los múltiplos del multiplicando requeridos previos a la realización de la multiplicación.
 - 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 ya codificada (b_{i-1}). Especificar los múltiplos del multiplicando requeridos previos a la realización de la multiplicación.
 - Hacer explícitos los productos parciales, asumiendo en este caso operandos de ocho bits. Utilizar las dos técnicas (mirando al pasado y mirando al futuro) de recodificación de Booth.
 - 45×-39
 - -12×65
 - Computar ambos productos utilizando la técnica **mirando al pasado**.
5. Se desea operar utilizando la *recodificación de Booth* radix 8, esto es recodificando de a **tres** bits.
- 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 en adición a los tres bits a recodificar en cada caso (esto es, considerar no sólo b_{i+2} , b_{i+1} , b_i , sino también a b_{i-1}). ¿Qué se puede afirmar en cuanto a los múltiplos necesarios para poder realizar el producto?
 - Repetir el inciso anterior, esta vez haciendo uso de la técnica **mirando al futuro**, en la cual se deben tener en cuenta los tres bits a recodificar (b_{i+2} , b_{i+1} , b_i) y el último bit de la parte aún no codificada (b_{i+3}) ¿Qué se puede afirmar en cuánto a los múltiplos necesarios para poder realizar la operación en comparación con la técnica **mirando al pasado**?
6. Esquematizar un *Wallace Tree* con **CSAs** para resolver el producto de dos operandos de ocho bits, indicando cómo quedarían los operandos de entrada al árbol asumiendo $X = 89$ e $Y = 55$. Resolver la suma paralela indicando claramente los operandos de entrada y de salida de cada **CSA**.
7. Esquematizar un *Wallace Tree* con **CSAs** para resolver el producto de dos operandos de 16 bits.
8. Esquematizar un *árbol binario de multiplicación* para resolver el producto de dos operandos de 8 bits utilizando dígito signado, indicando como quedarían los operandos de entrada al árbol asumiendo $X = 16$ e $Y = 24$.
9. Rehacer el ejercicio 6 esta vez empleando un *árbol binario de multiplicación* y representación *dígito signado*. Indicar en el esquema el valor de cada operando en las entradas y resolver la suma paralela que convierte a binario el resultado.

10. Esquematizar el proceso de *división con restoring* $X \div Y$ para los siguientes operandos enteros positivos:

a) $X = 01100$ e $Y = 00101$

b) $X = 01110$ e $Y = 00110$

y obtener Q (cociente) y R (resto). Trabajar con seis bits, teniendo en cuenta que tanto el acumulador como la **ALU** deben ser de $n + 1$ bits, dado que se debe tener un bit adicional para representar el signo.

11. Esquematizar el proceso de *división sin restoring* $X \div Y$ para los siguientes operandos enteros positivos de $n = 7$ bits:

a) $X = 1101011$ e $Y = 0010110$

b) $X = 1001111$ e $Y = 0000110$

y obtener Q (cociente) y R (resto).

12. Sea $X = 0,11011$ e $Y = 0,11$ dos mantisas normalizadas en *punto flotante*. Realizar el cálculo del cociente $X \div Y$, empleando el método de división rápida basado en la recíproca. Hacer el cómputo de cada uno de los r_i recordando que si genéricamente,

$$Y \cdot r_1 \cdot r_2 \cdot r_3 \cdot \dots \cdot r_{k-1} = 1 - Z_k$$

el r_k resultante es $r_k = 1 + Z_k$, con lo cual cada r_k se puede calcular como el complemento a dos de $Y \cdot r_1 \cdot r_2 \cdot r_3 \cdot \dots \cdot r_{k-1}$. Obtener solo los dos primeros coeficientes r_1 y r_2 , calcular $X \cdot r_1 \cdot r_2$ y verificar el error respecto al resultado exacto.

13. Calcular en base al *algoritmo de división SRT* el cociente entre los siguientes operandos.

a) $50/9$

b) $35/7$

c) $49/6$

d) $36/5$

OBS: considere hardware de 6 bits. La determinación de cuándo el R_i está acotado por $-b/2 \leq R_i < b/2$ (lo que asegura que $|R_i| \leq b$), dado que $b \geq 1/2$ se satisface con el intervalo $[-1/4, +1/4)$ adoptado para el r_i , se puede resolver analizando **dos bits** en la posición más significativa de *A.MQ*, en lugar de tres bits. El bit de signo puede despreciarse dado que resulta redundante.

14. Calcular en base al *algoritmo SRT higher-radix 4* el cociente $X \div Y$, consultando la *lookup table* (Figura 1), con los siguientes operandos

a) $X = 64$ e $Y = 7$.

b) $X = 135$ e $Y = 4$.

OBS: Trabajar con nueve bits (signo más ocho bits para la fracción).

b	Rango de P			q	b	Rango de P			q
8	-12	-7	-2	-2	12	-18	-10	-2	-2
8	-6	-3	-1	-1	12	-10	-4	-1	-1
8	-2	1	0	0	12	-4	3	0	0
8	2	5	1	1	12	3	9	1	1
8	6	11	2	2	12	9	17	2	2
9	-14	-8	-2	-2	13	-19	-11	-2	-2
9	-7	-3	-1	-1	13	-10	-4	-1	-1
9	-3	2	0	0	13	-4	3	0	0
9	2	6	1	1	13	3	9	1	1
9	7	13	2	2	13	10	18	2	2
10	-15	-9	-2	-2	14	-20	-11	-2	-2
10	-8	-3	-1	-1	14	-11	-4	-1	-1
10	-3	2	0	0	14	-4	3	0	0
10	2	7	1	1	14	3	10	1	1
10	8	14	2	2	14	10	19	2	2
11	-16	-9	-2	-2	15	-22	-12	-2	-2
11	-9	-3	-1	-1	15	-12	-4	-1	-1
11	-3	2	0	0	15	-5	4	0	0
11	2	8	1	1	15	3	11	1	1
11	8	15	2	2	15	11	21	2	2

Figura 1: SRT Higher-Radix 4 Lookup Table