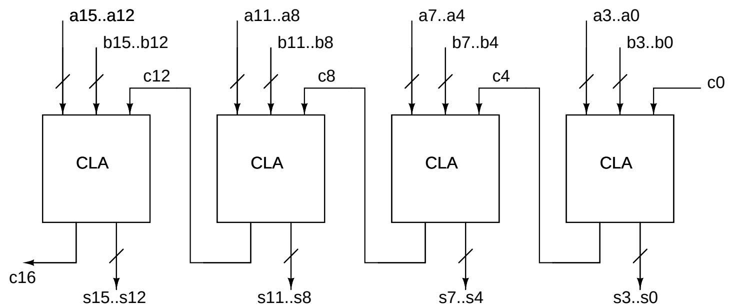
Organización del Computador
Práctico 3

B. Gonzalez Kriegel - N. Wolovick

11 de abril de 2000

1. Muestre el resultado de sumar los siguientes pares de números en complemento a 2's. Además del resultado muestre el valor de los bits de estado (z, n, c, v) *(valor a cero, valor negativo, acarreo y desbordamiento)*.
   (a) $10110 + 10111$
   (b) $01011 + 10101$
   (c) $11110 + 11101$
   (d) $11111 + 01111$
2. Efectúe la suma $1011101 + 0111011$ pensando estos números como:
   (a) complemento a 2's.
   (b) complemento a 1's.
3. Muestre el proceso de la multiplicación serial sin signo entre el multiplicando 1010 y el multiplicador 0101. Utilice el esquema propuesto por el libro.
4. Muestre el proceso de división serial sin signo entre el dividendo 1010 y el divisor 0100. Aquí también siga el esquema propuesto por el libro.
5. Se puede realizar un híbrido entre un CLA y un RCA para obtener un sumador de 16 bits como se muestra en la figura siguiente:



   (a) ¿Cuál es el mayor y menor retardo de compuerta entre cualquier entrada y cualquier salida?
   (b) ¿Cuál es el retardo de compuerta para S_{12}?

(c) ¿Cuál sería el mayor y menor retardo de compuerta para un RCA puro de 16 bits? Compare con lo obtenido en el punto 5a.

6. Multiplique 010011 (multiplicando) por 011011 (multiplicador) usando:

   (a) Algoritmo de Booth.

   (b) Algoritmo de Booth modificado.

7. El algoritmo de Booth maneja de manera directa enteros positivos y negativos (en complemento a 2's). Efectúe la multiplicación -8×-8 usando este método.

8. * El algoritmo de Booth es también conocido como *Algoritmo de Booth base-4*, pues cada grupo de 3 bits $(i+1, i, i-1)$ determina el factor que le corresponde a cada par de bits, es decir a 4^i. Explique porque no resulta conveniente extender la inspección a grupos de 4 bits (base-8) o más.

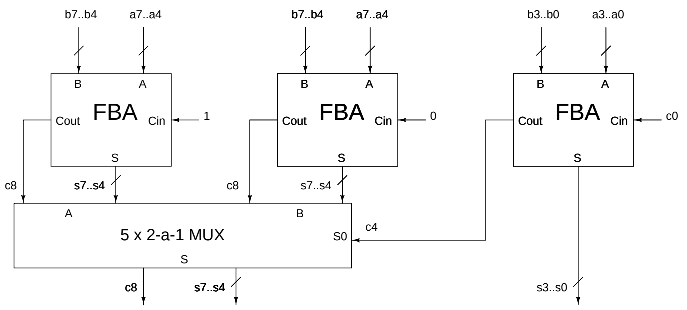
9. Para un CLA de 32 bits con entradas $a_{31} \ldots a_0, b_{31} \ldots b_0, c_0$ y salidas $s_{31} \ldots s_0, c_{32}$ compute:

   (a) Máximo retardo de compuerta.

   (b) Máximo número de entradas para una compuerta.

10. Un *carry select adder* (CSeA) se basa en el principio de efectuar 2 sumas en paralelo, una suponiendo $c_{in} = 0$ y la otra suponiendo $c_{in} = 1$. Cuando el acarreo de entrada se conoce, un multiplexor selecciona la suma apropiada.

    El siguiente es un diagrama básico para un CSeA de 8 bits dividido en 2 grupos de 4 bits.



   (a) Dibuje el diagrama de un sumador de 8 bits usando los FBA (*four bit adders*) anteriores, donde el acarreo se propaga de FBA en FBA como en los *ripple carry adders*.

   (b) Asumiendo que los FBA están implementados internamente con *carry lookahead*, compare el número de retardos de compuerta del sumador obtenido en el punto anterior y el CSeA dado en el inicio.

11. * Obtención de la raíz cuadrada en punto flotante por métodos iterativos.

    (a) Use el método de Newton para obtener un algoritmo iterativo para la raíz cuadrada. La fórmula de iteración incluye una división.

    (b) ¿Cuál es la manera más rápida que puede imaginar para dividir por 2 un número de punto flotante?

    (c) Si la división es lenta, entonces la rutina para la raíz cuadrada iterativa también lo será. Use el método de Newton sobre $f(x) = \frac{1}{x^2} - a$ para derivar un método que no utiliza división alguna.