Universidad del País Vasco
Facultad de Informática

Departamento de Arquitectura y Tecnología de Computadores

Arquitectura de Computadores I

Aritméticos 7 (solución)
División con restauración

Haz la división entre los números C = 27 y D = 6 (C/D) con restauración. Los operandos están representados en base 2, con el mínimo número posible de bits. ¿Cuál es el resto obtenido?

Solución

Nos piden utilizar el mínimo número posible de bits. Tenemos que recordar que estamos trabajando con números naturales, pero a veces tenemos que hacer restas y para ello tenemos que negar los bits y añadir un uno, es decir, tenemos que calcular el complemento a 2. Por lo tanto, aunque el valor `6` se puede representar en 3 bits, para hacer la resta, puesto que hay que sumar, necesitamos el valor `-6`, para el que son necesarios 4 bits. Por otro lado, para representar el `27` son necesarios al menos 5 bits.

El valor `6` representado en 4 bits es `0110,` y su complemento a 2 es el `1010`. Por su parte, el valor `27` representado en cinco bits será `11011`.

En las divisiones hay que dividir un número de $2n$ bits, con otro de n bits, siendo el resultado y el resto de n bits. Por eso, representaríamos el valor `27` en 8 bits, pero para evitar desbordamientos intermedios, las sumas parciales las haremos utilizando $n+1$ bits. Así pues, en un principio, representaremos el valor `27` con 9 bits: `000011011`.

En este método, antes de realizar las operaciones hay que desplazar una vez hacia la izquierda. En la primera pasada, a los bits de mayor peso se les suma `-5` y el bit de mayor peso del resultado (en este caso con valor 1) indica, por un lado, el valor del bit de mayor peso de la división (`0`, valor contrario a `1`), y, por otro lado, si es necesario hacer restauración (sumar `+6`, puesto que el valor de la operación anterior ha sido negativo). Así pues, en los siguientes pasos, cuando el bit de mayor peso sea `0` el bit correspondiente de la división será `1` y no habrá que hacer restauración, y cuando el bit de mayor peso sea `1` el bit correspondiente de la división será `0` y habrá que sumar `+6` llevando a cabo así la restauración en ese paso.

Como en el último paso el resto ha sido negativo, es necesario hacer una restauración también en este paso. Por eso mismo, al final se vuelve a sumar `+6`, para poder obtener el resto real (valor 3, secuencia de bits `0011`).

h(0):	00001	1011	
2h(0):	00011	011	
-6:	11010		

	11101	011	$z_3=0$
+6:	00110		
	-----		
h(1):	00011	011	
2h(1):	00110	11	
-6:	11010		
	-----		
h(2):	00000	11	$z_2=1$
2h(2):	00001	1	
-6:	11010		

	11011	1	$z_1=0$
+6:	00110		
	-----		
h(3):	00001	1	
2h(3):	00011		
-6:	11010		
	-----		
h(4):	11101		$z_0=0$
+6:	00110		

Resto:	00011		División: 0100