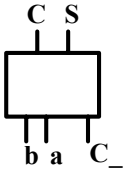
A2 Apéndice a los capítulos 3 y 4

Propagación rápida de acarreo en los sumadores

El sumador es un bloque muy común en los sistemas digitales; además, una forma simplificada del sumador, el incrementador (A + 1), se utiliza para construir contadores de gran tamaño.

Un sumador se configura en forma modular mediante celdas sumadoras:



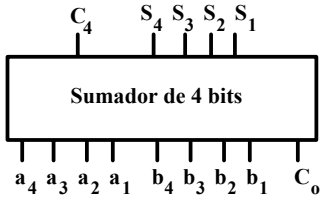
$$S = (a \oplus b) \oplus C\_$$

$$C = a \cdot b + (a + b) \cdot C\_$$

Al conectar en cascada estas celdas los retrasos con los que se calculan los sucesivos acarreos son acumulativos; este tipo de propagación del arrastre recibe el nombre de «acarreo sucesivo»: *ripple carry* (acarreo propagado por onda).

En el capítulo 2 se introdujo un indicador, el índice de propagación **i_p**, referido al número de puertas sucesivas que afectan a una función: el índice de propagación de la función $C_i = a_i.b_i + (a_i+b_i).C_{i-1}$ (construido con puertas inversoras) es 2; un sumador de 64 bits tendrá un índice de propagación 128, referido al bit más significativo $S_{65} = C_{64}$. Los tiempos de propagación asociados a un índice tan elevado limitan mucho la velocidad de trabajo de los sumadores.

Una forma de disminuir los tiempos de propagación de un sumador grande consiste en utilizar celdas sumadoras más amplias (de «granularidad» más gruesa), por ejemplo, celdas sumadoras de 4 dígitos.



Las funciones booleanas que conforman este sumador de 4 bits son las siguientes:

$$S_i = (a_i \oplus b_i) \oplus C_{i-1} \qquad \text{para } i = 1 \,..\, 4$$

$$C_1 = a_1.b_1 + (a_1 + b_1).C_0$$

$$C_2 = a_2.b_2 + (a_2 + b_2).(a_1.b_1 + (a_1 + b_1).C_0)$$

$$C_3 = a_3.b_3 + (a_3 + b_3).(a_2.b_2 + (a_2 + b_2).(a_1.b_1 + (a_1 + b_1).C_0))$$

$$S_5 = C_4 =$$

$$= a_4.b_4 + (a_4 + b_4).(a_3.b_3 + (a_3 + b_3).(a_2.b_2 + (a_2 + b_2).(a_1.b_1 + (a_1 + b_1).C_0)))$$

Tomás Pollán Santamaría. Tecnología Electrónica. E.U.I.T.I.Z. Universidad de Zaragoza. tpollan@posta.unizar.es

En las funciones anteriores (celda sumadora de 4 dígitos) se ha aplicado la recursividad del acarreo para calcular los sucesivos arrastres directamente, a partir de las entradas de la celda; todos los arrastres (construidos con puertas inversoras) presentan índice de propagación 2, incluyendo el arrastre global de la celda C_4, que irá conectado a la celda siguiente para construir sumadores más grandes.

Para un sumador de 64 dígitos se necesitan 16 celdas sucesivas de 4 bits, de forma que el dígito más significativo S_{65} tendrá un índice de propagación 32 (4 veces menor que en el sumador de celdas de 1 bit).

Aún es posible conseguir una propagación más «rápida» del arrastre (*lookahead carry*: acarreo anticipado), aprovechando intensivamente la recursividad de la función booleana C_i en la forma que sigue.

Podemos expresar el arrastre C_i en términos de generación y propagación del mismo

$$C_i = a_i.b_i + (a_i+b_i).C_{i-1} = g_i + p_i \cdot C_{i-1}$$

siendo $g_i = a_i \cdot b_i$ generación de acarreo

$p_i = a_i + b_i$ propagación de acarreo

y aplicando la recursividad propia de esta fórmula ($C_i \rightarrow C_{i-1}$)

$$C_i = g_i + p_i \cdot g_{i-1} + p_i \cdot p_{i-1} \cdot g_{i-2} + \ldots + p_i \cdot p_{i-1} \cdot p_{i-2} \cdot \ldots \cdot p_{m+1} \cdot C_m$$

se obtiene una función que permite calcular, «en un solo paso», un acarreo avanzado (el i-ésimo) a partir de otro inferior (el m-ésimo).

Dado que las puertas lógicas a utilizar son inversoras, conviene efectuar las siguientes transformaciones para configurar la función anterior con puertas inversoras:

$$g'_i = \overline{g_i} = \overline{a_i.b_i} = a_i * b_i$$

$$p'_i = \overline{p_i} = \overline{a_i + b_i} = a_i \,\Delta\, b_i;$$

en la expresión de C_i se puede sustituir $g_i = p_i \cdot g_i$, ya que siempre que $g_i = 1$, $p_i = 1$

$$C_i = g_i + p_i \cdot C_{i-1} = g_i \cdot p_i + p_i \cdot C_{i-1} = p_i \cdot (g_i + C_{i-1}) = \overline{\overline{p_i} + \overline{g_i}.\overline{C_{i-1}}} =$$

$$= \overline{p'_i + g'_i.\overline{C_{i-1}}}$$

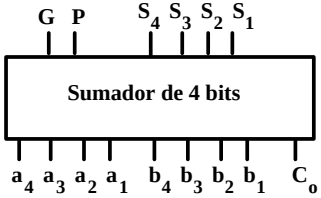
y aprovechando la recursividad

$$C_i = \overline{p'_i + g'_i.p'_{i-1} + g'_i.g'_{i-1}.p'_{i-2} + \ldots + g'_i.g'_{i-1}.g'_{i-2} \cdot \ldots \cdot g'_{m+1}.\overline{C_m}}\,.$$

Las operaciones iniciales p'_i , g'_i y $\overline{C_m}$ tienen índice de propagación 1 (una sola puerta lógica) y, también, C_i requiere una sola puerta a partir de ellas; de forma que el cálculo de C_i supone el paso a través de dos series de puertas booleanas sucesivas: índice de propagación **$i_p = 2$**.

Conforme a lo anterior y como ejemplo demostrativo de la propagación rápida de acarreo (*lookahead carry*), se desarrolla el diseño de un sumador de 64 dígitos:

Primer paso: configuración de un módulo de 4 bits



Las funciones booleanas que conforman este sumador de 4 bits son las siguientes:

$S_1 = (a_1 \oplus b_1) \oplus C_0$

$S_2 = (a_2 \oplus b_2) \oplus C_1;\ C_1 = \overline{p'_1 + g'_1.\overline{C_0}}$

$S_3 = (a_3 \oplus b_3) \oplus C_2;\ C_2 = \overline{p'_2 + g'_2.p'_1 + g'_2.g'_1.\overline{C_0}}$

$S_4 = (a_4 \oplus b_4) \oplus C_3;\ C_3 = \overline{p'_3 + g'_3.p'_2 + g'_3.g'_2.p'_1 + g'_3.g'_2.g'_1.\overline{C_0}}$

El módulo debe contener, además, dos funciones «globales» de generación y propagación del acarreo

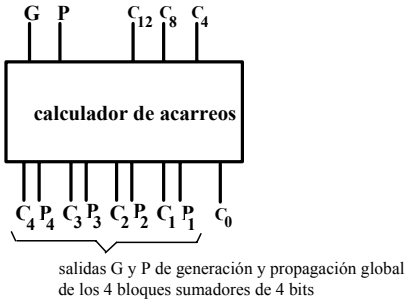
$$G = g_4 + p_4.g_3 + p_4.p_3.g_2 + p_4.p_3.p_2.g_1$$
$$= \overline{g'_4.(p'_4 + g'_3).(p'_4 + p'_3 + g'_2).(p'_4 + p'_3 + p'_2 + g'_1)}$$

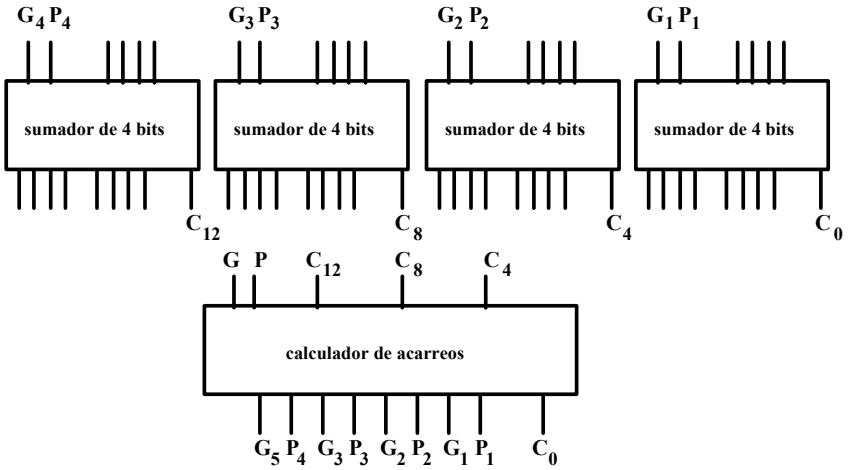
$$P = p_4.p_3.p_2.p_1 = \overline{p'_4 + p'_3 + p'_2 + p'_1)}$$

p'_i , g'_i y $\overline{C_m}$ tienen índice de propagación 1 y, con ello, las funciones que calculan los arrastres C_i y, también, las funciones G y P tienen índice de propagación 2.

Segundo paso: agrupación de 4 módulos para construir un bloque de 16 bits

Además de los cuatro sumadores de 4 bits, se requiere un nuevo módulo que permita calcular los acarreos de entrada a los tres módulos superiores C_4, C_8 y C_{12}, así como las funciones G P de generación y de propagación globales del sumador de 16 bits:





$C_4 = G_1 + P_1.C_0$ $\qquad G = G_4 + P_4.G_3 + P_4.P_3.G_2 + P_4.P_3.P_2.G_1$

$C_8 = G_2 + P_2.G_1 + P_2.P_1.C_0$ $\qquad P = P_4 . P_3 . P_2 . P_1$

$C_{12} = G_3 + P_3.G_2 + P_3.P_2.G_1 + P_3.P_2.P_1.C_0$

El índice de propagación de estas funciones es, asimismo, 2 (al aplicar una doble negación para formar puertas inversoras, resultan dos puertas sucesivas).

Tercer paso: agrupación de 4 bloques de 16 bits para configurar el sumador de 64 bits

De nuevo se requiere un «calculador de acarreos» idéntico al del apartado anterior para calcular los arrastres de entrada correspondientes a los tres bloques de 16 bits superiores: C_{16}, C_{32} , C_{48}.

Y, a partir de las salidas G P globales de este «calculador de acarreos», se calcula el dígito más significativo de la suma: $S_{65} = C_{64} = G + P.C_0$

El índice de propagación para dicho dígito S_{65} será:

	celdas de 4 bits	2
+	«calculador de acarreos» del bloque de 16 bits	2
+	«calculador de acarreos» del sumador de 64 bits	2
+	cálculo de S_{65}, a partir de este último	2

En conjunto, el índice de propagación global del dígito más significativo del sumador rápido (*lookahead carry*) de 64 bits es 8, muy inferior al correspondiente a una propagación sucesiva de acarreo (*ripple carry*) que es 128.