Carry lookahead adder

¿ Como funciona ?

La principal caracteristica del carry lookahead adder **(CLA)** es que permite obtener el carry de cualquier suma desde el momento incial, sin necesidad de esperar a que termine la suma anterior.

A continuación se puede ver la tabla de verdad que relaciona las entradas y los carries.

a	b	Cin	Cout
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

a y b son los números a sumar, Cin es el carry de entrada y Cout el carry de salida.

Analizando la tabla sabremos que el carry out será uno cuando: **(A and B) or (A xor B) and Cin** o expresado de otra forma:

Cout = (A * B) + (A⊕B) * Cin

Podemos diferenciar partes de esta expresión, donde la sección (A*B) se conocerá como *"carry generator **(G)** "* y la sección (A⊕B) como *"carry propagator **(P)** "*. G no tiene dependencia de Cin, solo de los valores de entrada a y b, mientras que P depende del carry de entrada. G nos permite conocer si el bit de la suma debe generar carry, P indica si el bit de la suma debe propagar carries generados por bits anteriores.

Cout = G + P*Cin

Calcularemos G y P para cada bit de la suma, podemos generalizar la expresión a:

Cn = Gn + Pn * Cn-1

De esta manera, si tenemos las entradas a,b y C0

C1 = G1 + P1 * C0

C2 = G2 + P2 * C1, si expandimos esto obtenemos:

C2 = G2 + P2 * (G1 + P1 * C0), lo que es equivalente a :

C2 = G2 + P2 * G1 + P2 * P1 * C0

Podemos dar un ejemplo más con C3

C3 = G3 + P3 * C2

C3 = G3 + P3 * (G2 + P2 * G1 + P2 * P1 * C0)

C3 = G3 + P3 * G2 + P3 * P2 * G1 + P3 * P2 * P1 * C0

Como podemos notar podemos obtener el carry en cualquier posición conociendo el Carry inicial, sin necesidad de esperar que se realicen las operaciones anteriores.

¿ Porque es más rápido que ripple adder ?

Al generar los carries de manera anticipada e independiente, la suma de bits se puede realizar de manera paralela, lo que permite procesar multiples bits al mismo tiempo a diferencia del ripple adder en el cual los bits se deben sumar uno tras otro, reduciendo su velocidad.

¿ Cuales son sus desventajas ?

Una de las desventajas más claras del CLA es que es mucho mas complejo y menos intuitivo que el ripple adder, debido a esto requiere utilizar más puertas lógicas para calculos como el de G y P, lo que dificulta el diseño del circuito y aumenta la cantidad de transistores necesarios para su implementación.