SUMADORES

Diseño de Sistemas con FPGA

Patricia Borensztejn

Sumadores

Full Adder

$$S_i = a_i \oplus b_i \oplus c_i$$
$$C_{i+1} = a_i b_i + a_i c_i + b_i c_i$$

Half Adder

$$S_i = a_i \oplus b_i$$
$$C_{i+1} = a_i * b_i$$

Ripple-Carry Adder: el “normal” con propagación de acarreo

- Ripple-carry adder: suma dos números de n-bits con n full adders. El delay del ripple-carry adder depende de la longitud n de los operandos.
- Fácil de construir. Ocupa poca área.

Ripple-Carry Adder: el “normal” con propagación de acarreo

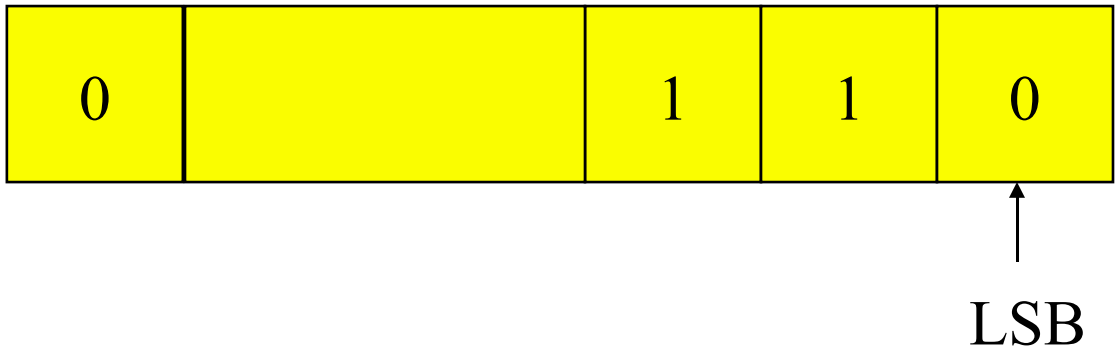
```
module nBitAdder(f, cOut, a, b, cIn);
parameter n = 7;

output reg [n:0] f;
output reg cOut;
input [n:0] a;
input [n:0] b;
input cIn;

always @(a, b, cIn)
{cOut, f} = a + b + cIn;
endmodule
```

Sumador Serial

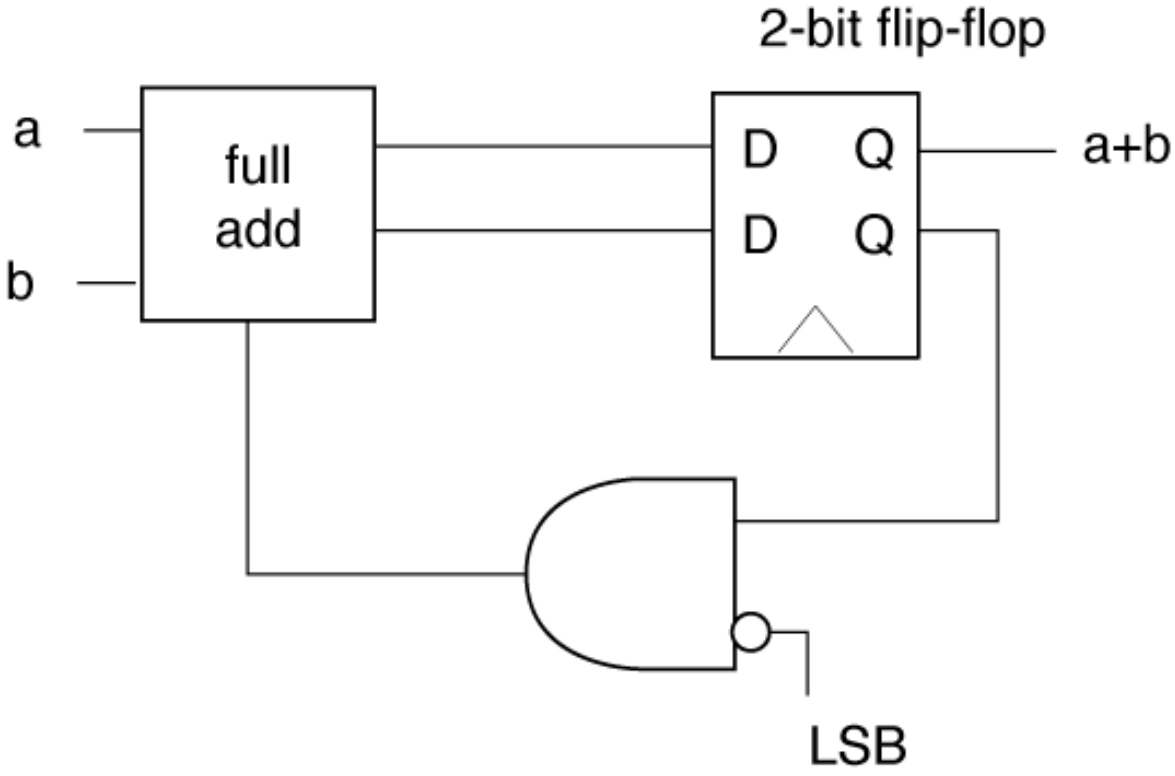
- Se usa en procesamiento de señales. Es importante que el tiempo de ciclo sea corto. No importa tanto la latencia.
- Formato de los datos (LSB primero):



- Cuando empieza un par de datos nuevos, el carry in se pone a cero.

Estructura del sumador serial

- La señal LSB pone a cero carry shift register.
- El tiempo de ciclo es igual al del full adder mas el delay del registro.



Carry-lookahead adder

- Descompone el carry en dos partes: acarreo propagado y acarreo generado.
  - Generado: si los dos sumandos son 1 (a_i b_i=1)
  - Propagado: si alguno de los dos es 1, propaga el carry de la suma anterior.

$$C_{i+1} = \underbrace{a_i b_i}_{G} + \underbrace{(a_i \oplus b_i)}_{P} c_i$$

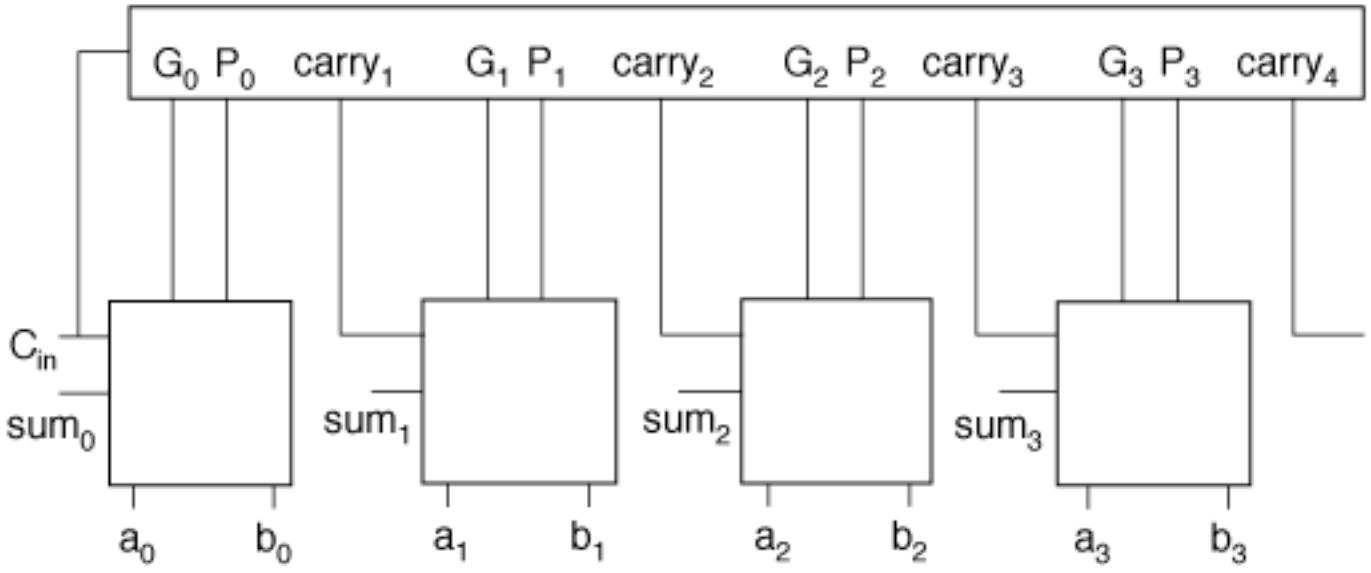
- P y G no dependen del carry anterior! :
  - $P_i = a_i$ xor b_i
  - $G_i = a_i$ b_i
- Reescribimos S y C usando P and G:
  - $s_i = c_i$ xor P_i
  - $c_{i+1} = G_i + P_i c_i$

Carry Lookahead Adder

- No hay dependencia de los valores anteriores.
- Los acarreos de cada bit pueden calcularse independientemente.

Pi = ai xor bi
Gi = ai bi

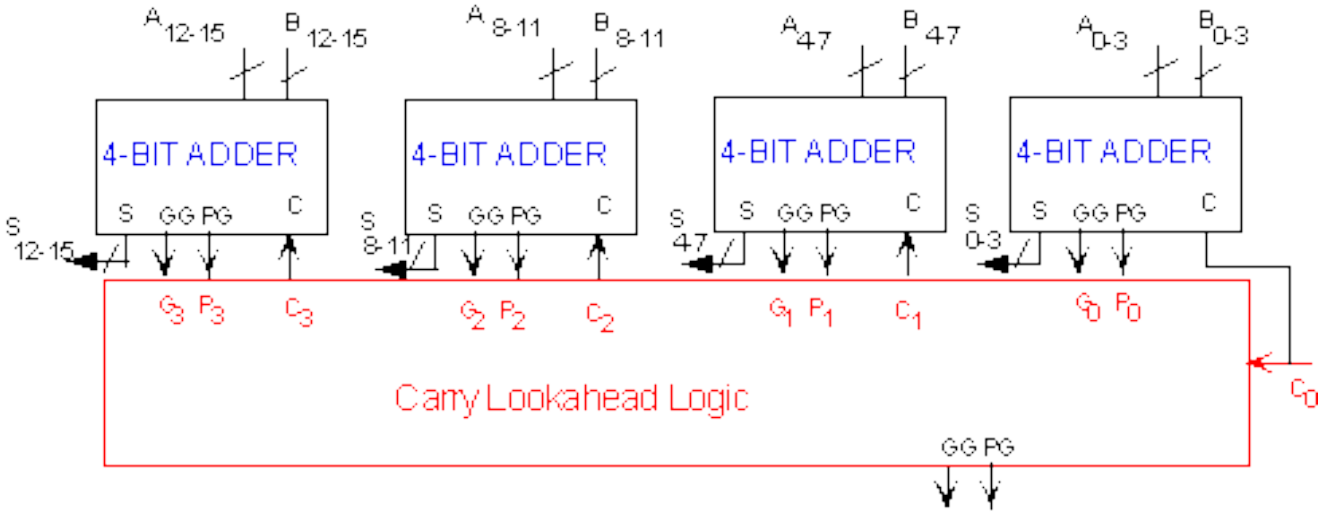
C1 = G0 + P0.C0
C2 = G1 + P1.C1 = G1 + P1.G0 + P1.P0.C0
C3 = G2 + P2.G1 + P2.P1.G0 + P2.P1.P0.C0
C4 = G3 + P3.G2 + P3.P2.G1 + P3P2.P1.G0 + P3P2.P1.P0.C0



16-bit CLA

- Desventaja del CLA: para muchos bits, la lógica se complica…. Generación de carry, requiere puertas con mas fanin (número de entradas ➔ mas lento)

- Se usan módulos de 4 bits (CLA) y se encadenan como los ripple carry adders



16 CLA Adder

- Los módulos de 4 bits son CLA porque calculan sus P y sus G, pero además calculan S propagando el acarreo interno.
- En este caso, cada módulo de 4 bits calcula su P y su G, que ahora llamaremos PG y GG:

  PG = P0.P1.P2.P3

  GG= G3 + G2P3+G1P3P2+G0P3P2P1
- La unidad CLL (carry lookahead logic) calcula los carrys según :
  - Cout=GG +PG.Cin

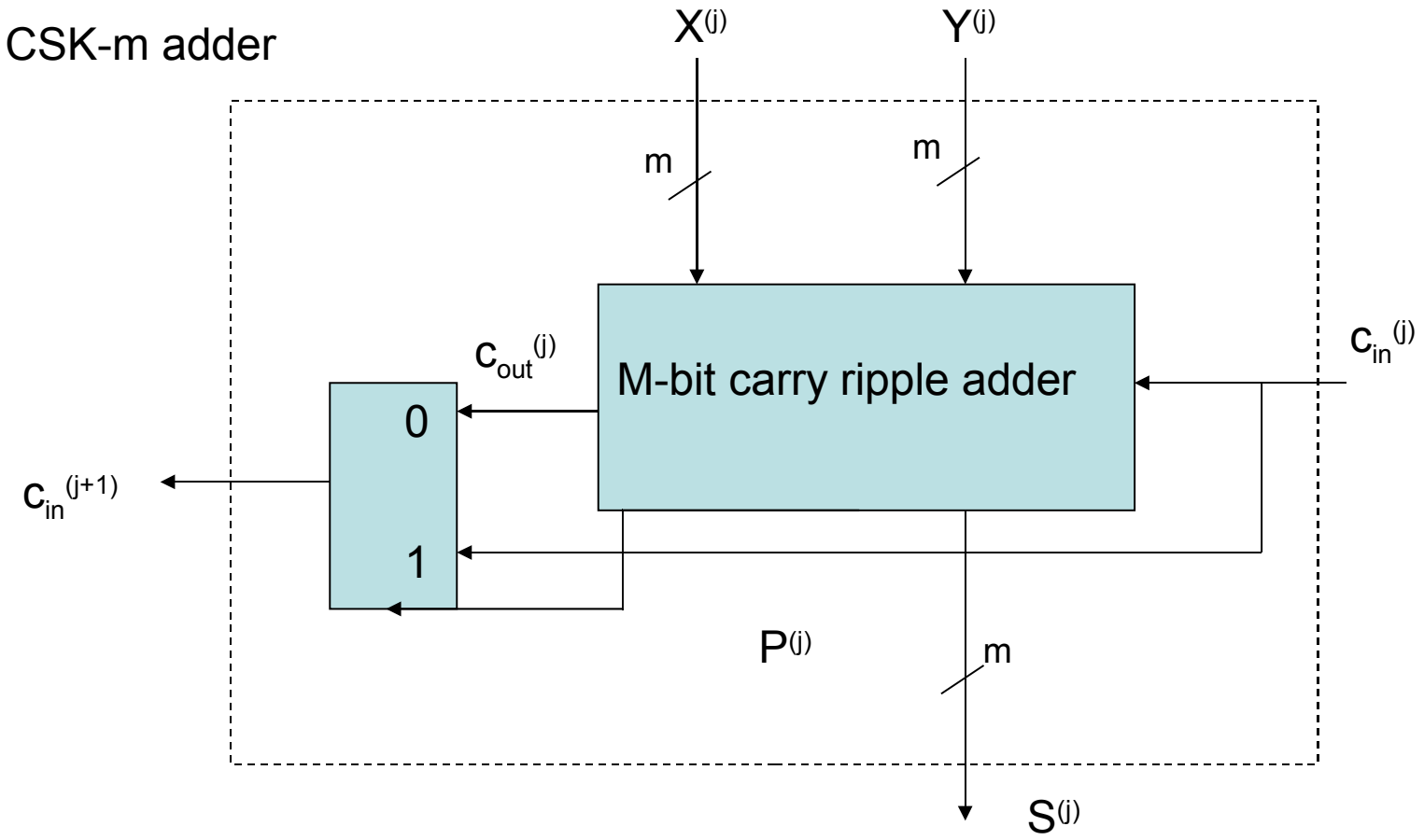
16-bit CLA Adder

- Análisis de tiempos:
  - Cada CLA calcula:
    - Tiempo 1: todos los Pi y Gi
    - Tiempo 2: todos los PG (1 nivel de puertas)
    - Tiempo 3: todos los GG (2 niveles de puertas)
    - Tiempo 3: todos sus Ci (hay que propagar el carry)
  - Cada CLA recibe su entrada del LCU
    - CLA0 en tiempo 0 (carry inicial)
    - CLA1, CLA2, CLA3 en tiempo 5 (dos niveles de puertas)
  - Cada CLA calcula su suma S
    - CLA0 en tiempo 4 (con carry inicial, propaga acarreo)
    - CLA1, CLA2, CLA3 en tiempo 8 (dos niveles de puertas)
  - Cálculo de C16 por la LCU: tiempo 5
- Comparación con un CRA:
  - Tiempo de propagación de acarreo: 16 para la S y Cout.

Carry Skip Adder

- Mira por casos donde el carry de entrada a un conjunto de bits es el mismo que el de salida.
- Típicamente organizado en etapas de m bits
- Cuando se cumple que todos los P_i del grupo de bits son 1, el grupo propaga el carry de entrada.
- El carry skip adder está formado por bloques de m bits que implementan el carry ripple adder.
- El objetivo es mejorar el tiempo de propagación de los ripple adders. Es decir, reducir el tiempo en que el carry se propaga.

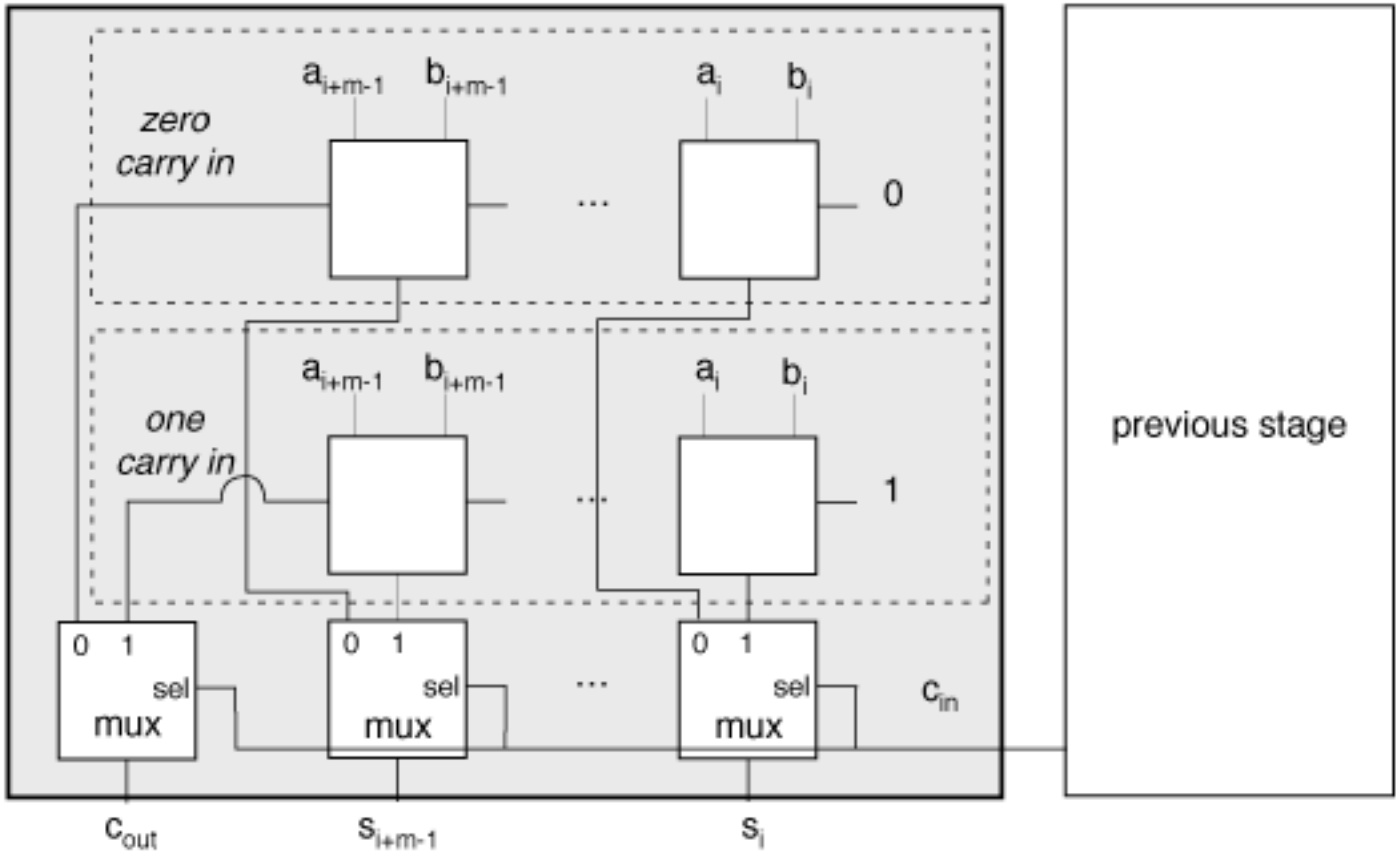
M-bit carry-skip adder



Carry-select adder

- Calcula dos resultados de la suma y el acarreo de salida en paralelo: uno para cada valor del carry de entrada (0,1)
- Luego, un multiplexor selecciona el resultado correcto.
- Los sumadores operan en paralelo. El retardo está limitado por el retardo de los multiplexores, que es menor.

Estructura del Carry-select adder

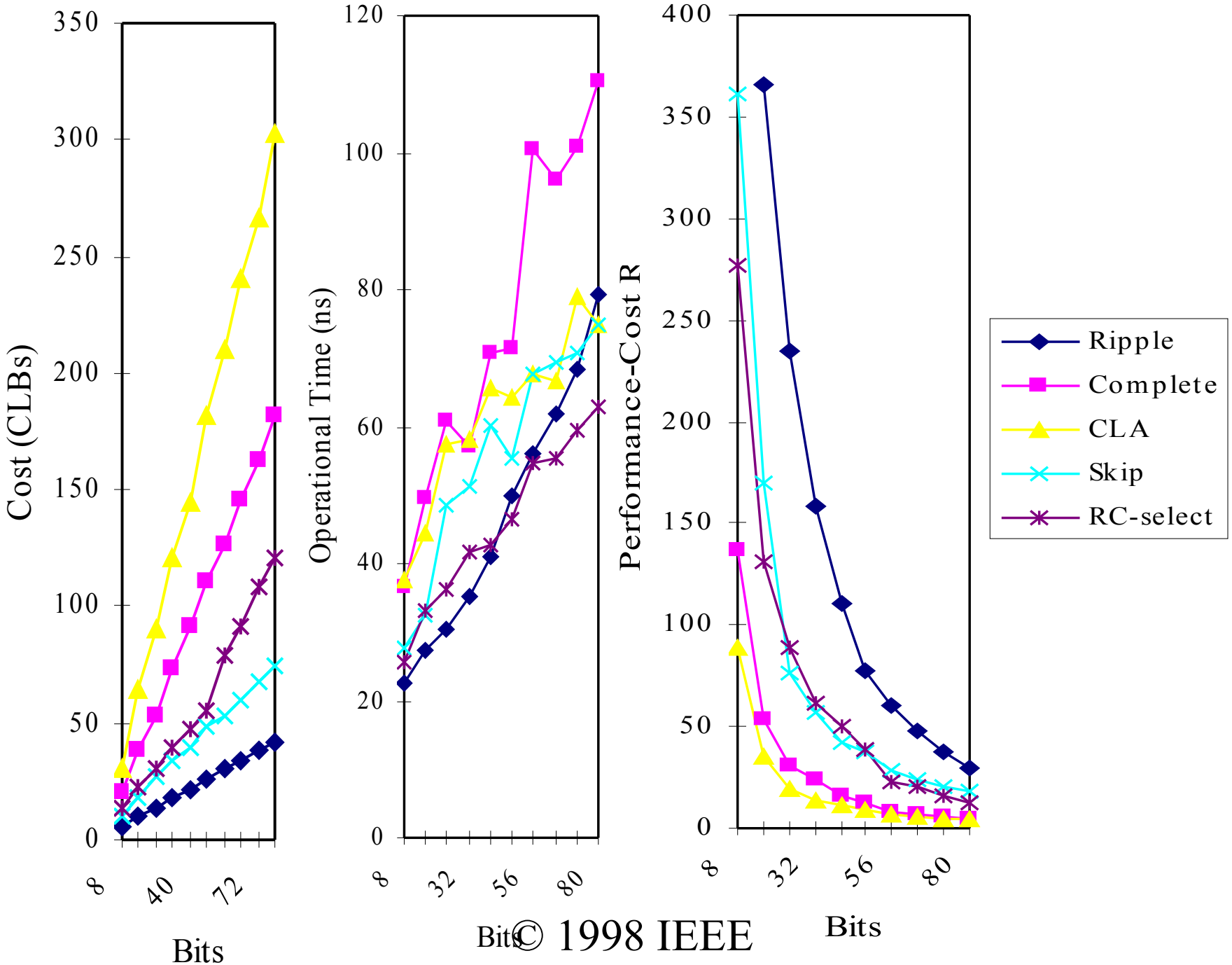


Bibliografía …

- Digital Arithmetic
- Syntesis of Aritmetic Circuits
- FPGA Based System Design

FPGA Adders

- FPGA Adders: Performance evalution and optimal design. IEEE Design & Test of Computers. Xing y W.H.Yu
  - Ripple-carry adder has highest performance/cost.
  - Optimized adders are most effective in very long bit widths (> 48 bits).


Cost (CLBs)
350
300
250
200
150
100
50
0
8
40
72
Bits
Operational Time (ns)
120
100
80
60
40
20
0
8
32
56
80
Bits
Performance-Cost R
400
350
300
250
200
150
100
50
0
8
32
56
80
Bits
Ripple
Complete
CLA
Skip
RC-select