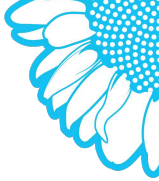


IE0323

Sistemas Digitales I

Laboratorio #6

Sumadores



Objetivos

- Comprender el funcionamiento de un sumador completo (full adder) y su integración en estructuras más complejas.
- Diseñar e implementar sumadores en cascada y analizar su funcionamiento.
- Explorar nuevas topologías de sumadores, como los sumadores de acarreo anticipado, y comprender sus diferencias con el diseño en cascada.
- Desarrollar habilidades prácticas en diseño y programación en Intel Quartus Prime Lite.

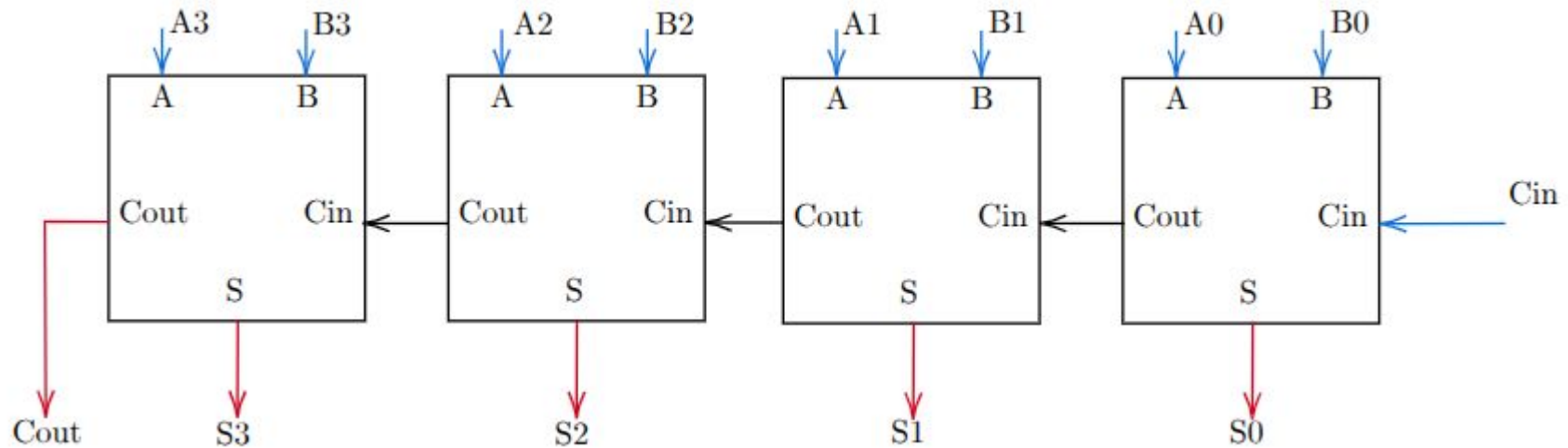
Sumador Completo

$$\begin{array}{r} A \\ + B \\ \hline C_{out} S \end{array}$$

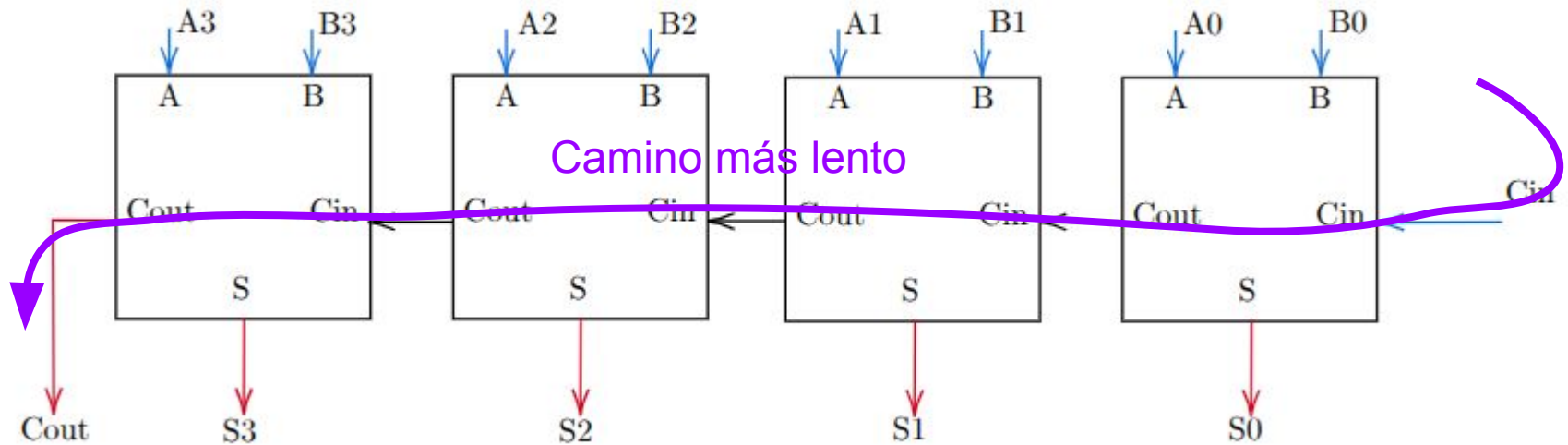
fulladder.v

```
1 //Declaracion del modulo fulladder
2 module fulladder (A, B, Cin, Cout, S);
3     //Declaracion de entradas y salidas
4     input A, B, Cin;
5     output reg Cout, S;
6
7     //Cada vez que cambie alguna entrada
8     always @(*) begin
9         //Realice la suma A+B+Cin
10        //Y asigne el resultado a {Cout, S}
11        //Cout contiene el bit mas significativo del resultado
12        //S contiene el bit menos significativo del resultado
13        {Cout, S} = A + B + Cin;
14    //Termina always
15    end
16 //Termina modulo
17 endmodule
```

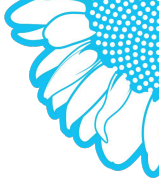
Ripple Carry Adder (4 bits)



Ripple Carry Adder (4 bits) – Camino crítico



$$C_{i+1} = (A_i \cdot B_i) + C_i \cdot (A_i \oplus B_i)$$



Generación y Propagación de acarreo

- **Generación**

$$G_i = A_i \cdot B_i$$

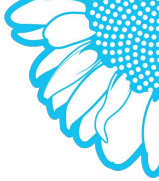
Si A y B son uno (ambos), **se generará** un acarreo, sin importar lo que venga de etapas anteriores

- **Propagación**

$$P_i = A_i \oplus B_i$$

Si solo una de las entradas (A o B), pero no ambas son uno, el valor del acarreo de entrada **se propagará** al acarreo de salida

$$C_{i+1} = G_i + C_i \cdot P_i$$



Ecuaciones de acarreo

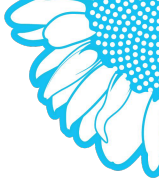
*Sigue dependiendo de
acarreo anteriores*

$$C_1 = G_0 + C_{in} \cdot P_0$$

$$C_2 = G_1 + C_1 \cdot P_1$$

$$C_3 = G_2 + C_2 \cdot P_2$$

$$C_{out} = G_3 + C_3 \cdot P_3$$



Ecuaciones de acarreo

Solo depende de las
entradas originales

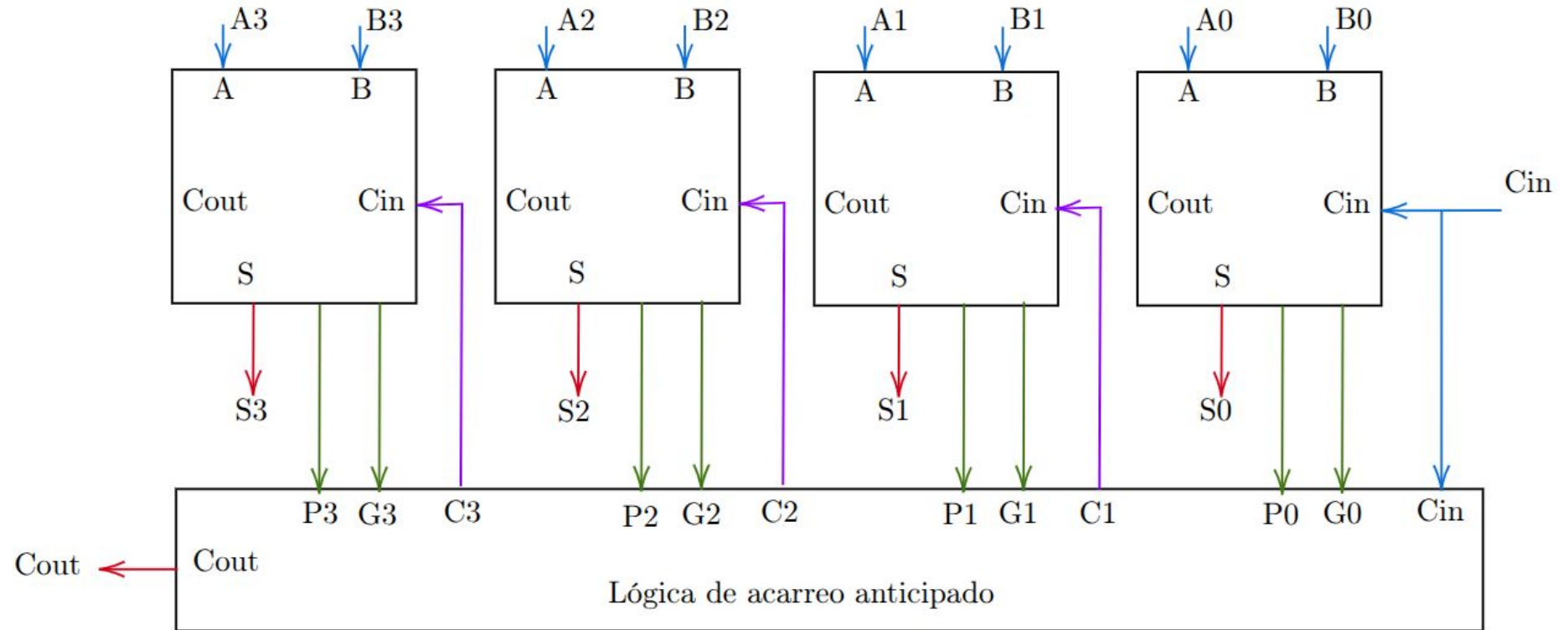
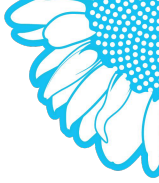
$$C_1 = G_0 + C_{in} \cdot P_0$$

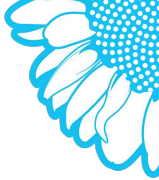
$$C_2 = G_1 + (G_0 + C_{in} \cdot P_0) \cdot P_1$$

$$C_3 = G_2 + (G_1 + (G_0 + C_{in} \cdot P_0) \cdot P_1) \cdot P_2$$

$$C_{out} = G_3 + (G_2 + (G_1 + (G_0 + C_{in} \cdot P_0) \cdot P_1) \cdot P_2) \cdot P_3$$

Carry Lookahead Adder (4 bits)





Trabajo en Clase

Implemente un **sumador de acarreo en cascada de 4 bits**.

Este módulo **deberá contener múltiples instancias del módulo fulladde**.

Utilice el mapeo descrito a continuación, respetando el orden de significancia para los sumandos:

Sumando 1 (A): {SW3, SW2, SW1, SW0}

Sumando 2 (B) : {SW9, SW8, SW7, SW6}

Acarreo de Entrada (Cin) : Key1

Despliegue el resultado de la suma utilizando los LEDs que necesite de la tarjeta de desarrollo, asigne LEDR0 como el bit menos significativo del resultado.

Nota: Tome en cuenta que **los botones Key0 y Key1 producen un nivel lógico alto cuando no están siendo presionados** y que adoptan un estado lógico bajo cuando son presionados.



A[3:0]