



Instituto Politécnico Nacional

Escuela Superior de Cómputo



Diseño de Sistemas Digitales

Práctica 4(Repaso): Sumador y Restador

Integrantes: Bravo Esquivel Gustavo

Colín Ramiro Joel

Pasten Juarez Joshua Michael

Profesor: Mújica Ascencio Cesar

Grupo: 4CV3

Introducción

En toda computadora, cualquier aparato electrónico dentro de la infinidad de circuitos que hay, el más importante es el microprocesador y dentro del microprocesador lo más importante es la **ALU** (Unidad Aritmética Lógica) y esta ALU está compuesta por lo que llamamos "sumadores".

Entonces, un sumador es este circuito digital que se encarga de realizar la adición de números. No obstante, se pueden utilizar en otras partes de este procesador no necesariamente en la ALU, también se utilizan para calcular direcciones, índices de tablas, operadores de incremento, decremento, etc. En esta práctica trabajaremos con sumadores y restadores de 2 bits.

Un sumador completo lo que hace es que suma dos números binarios de n bits junto con las cantidades de acarreo. Se puede observar como un componente en cascada de sumadores que suman 8, 16, 32, ... números binarios de bits. El circuito produce una salida de 2 bits, al igual que el sumador semi, denominadas como **Cout** (Acarreo de Salida) y **S** (suma). Un ejemplo de su implementación más común y con la que estaremos trabajando es la siguiente:

$$S = A \oplus B \oplus C$$

$$\text{Cout} = (A \cdot B) + \text{Cin} (A \oplus B)$$

En cuanto a los restadores es exactamente lo mismo solo que en este caso con signo de resta.

Desarrollo

Diseño

Sumador de 2 bits

$$\begin{array}{r} + A_1 A_0 \\ B_1 B_0 \\ \hline \Sigma_2 \Sigma_1 \Sigma_0 \end{array}$$

$$\Sigma = S = \text{Suma}$$

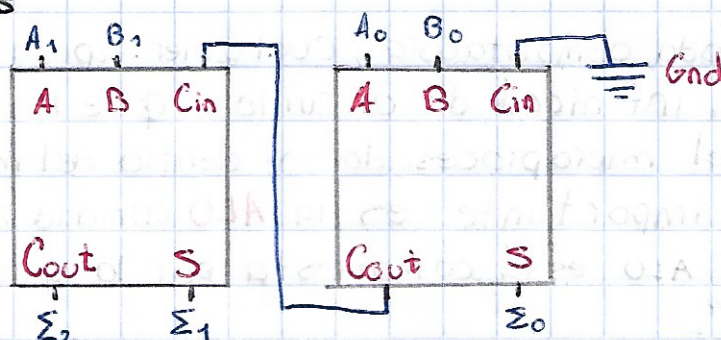


Tabla de verdad

Al ser de 2 bits, la tabla de verdad del sumador quedaria de la siguiente manera:

A	B	Cin	Cout	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Realizando mapas de Karnaugh:

$$S = A \oplus B \oplus C_{in}$$

A \ B \ C _{in}	00	01	11	10
0	0	1	1	1
1	1	1	1	0

$$S = \bar{A}\bar{B}C_{in} + \bar{A}B\bar{C}_{in} + A\bar{B}C_{in} + AB\bar{C}_{in}$$

$$S = A \oplus B \oplus C_{in}$$

$$C_{out} = A \cdot B + A \cdot C_{in} + B \cdot C_{in}$$

A \ B \ C _{in}	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$$C_{out} = AC_{in} + AB + BC_{in}$$

Restador de 2 bits

El restador es la misma concepto que el sumador únicamente que con signo de resta en la operación y con una pequeña restricción: No puede ser número negativo mientras se resta.

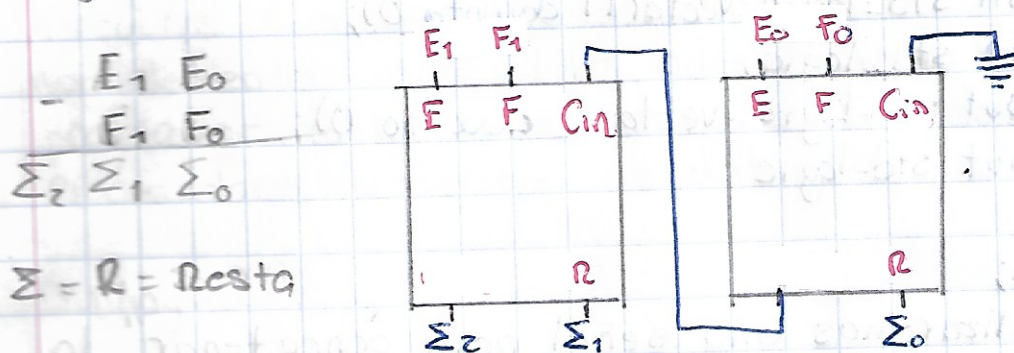


Tabla de verdad

E	F	Cin	Cout	R
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Mapas de Karnaugh

$S =$

E\F Cin	00	01	11	10
0		1		1
1	1		1	

Al igual que el sumador

$$S = E \oplus F \oplus Cin$$

Cout:

E\F Cin	00	01	11	10
0		1	1	1
1			1	

$$Cout = \bar{E}Cin + \bar{E}F + FCin$$

Entidad

La entidad para esta práctica es la siguiente:

```
entity sumres2 is
```

```
port(
```

```
  A,B,E,F: in std_logic_vector(1 downto 0);
```

```
  CinSum, CinRes: in std_logic;
```

```
  S, R: out std_logic_vector(1 downto 0);
```

```
  CoutSum, CoutRes: out std_logic
```

```
);
```

```
end sumres;
```

También utilizaremos una señal para "concatenar" la Cin y la Cout de respectivos sumadores/restadores.

```
signal C,D: std_logic_vector(2 downto 0);
```

Capturas en Galaxy

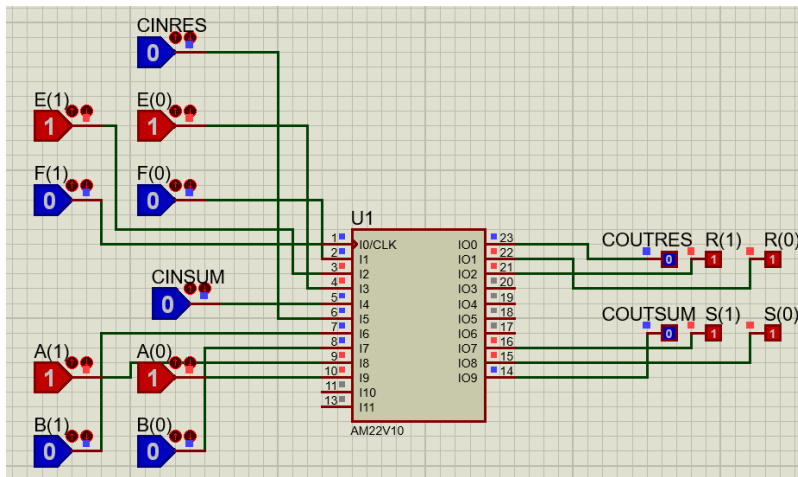
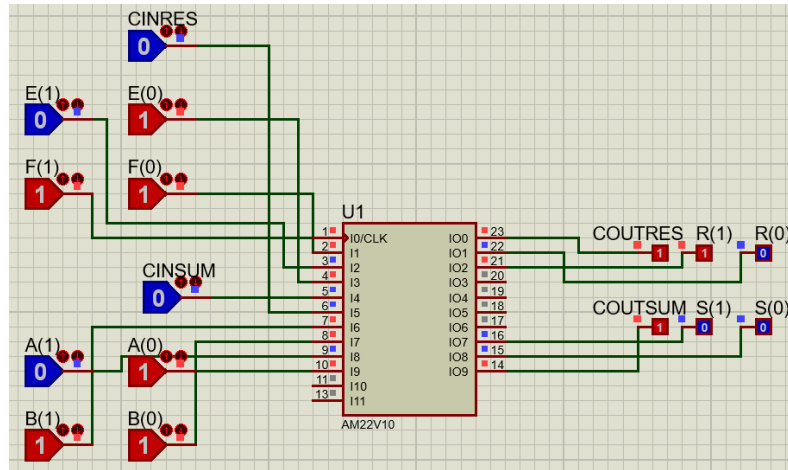
Name	Value	Stimulator	50 ns
CinSum	0	<= 0	
+ A	1	<= 01	1
+ B	3	<= 11	3
CoutSum	1		
+ S	00		00

Name	Value	Stimulator	100 ns
CinSum	0	<= 0	
+ A	3	<= 11	1 3
+ B	0	<= 00	3 0
CoutSum	0		
+ S	11		00 11

Name	Value	Stimulator	50 ns
CinRes	0	<= 0	
+ E	1	<= 01	1
+ F	3	<= 11	3
CoutRes	1		
+ R	10		10

Name	Value	Stimulator	100 ns
CinRes	0	<= 0	
+ E	3	<= 11	1 3
+ F	0	<= 00	3 0
CoutRes	0		
+ R	11		10 11

Capturas de Proteus



Conclusiones

La realización de esta práctica, aunque fue de repaso, fue fundamental para recordar como se implementaban estos circuitos, tanto en diseño como en VHDL. También es importante hacer mención que estos circuitos a pesar de resolver problemas aritméticos, son elementos muy útiles además de ser base para programar subsiguientes circuitos de memorias y procesadores numéricos de datos, etc, etc.

Bibliografía

- 1-http://profesores.fi-b.unam.mx/normaelva/sumador_restador.pdf
- 2-<http://compilandokonocimiento.com/2017/06/13/sumador-y-restador/>
- 3-<http://www2.ulpgc.es/hege/almacen/download/7055/7055018/05.pdf>
- 4-<http://www.youtube.com/watch?v=PFnM7DZ3POM&e=4645/>

			OpC = '0'		OpC = '1'	
Cin	A	B	S	CoutS	R	CoutR
0	0	0	0	0	0	0
0	0	1	1	0	1	1
0	1	0	1	0	1	1
0	1	1	0	1	0	1
1	0	0	1	0	1	0
1	0	1	0	1	0	0
1	1	0	0	1	0	0
1	1	1	1	1	1	1