Lógica digital

Organización de computadoras

Universidad Nacional de Quilmes

20 de agosto de 2013

Fechas Importantes

- Primer parcial Sábado 5 de octubre, 9:30 hs
- Segundo parcial Martes 19 de noviembre, 19 hs
- Recuperatorio de ambos parciales Sábado 10 de noviembre, 9:30 hs
- Integrador Viernes 13 de diciembre

Repaso

1. Evolución de la computación
2. Instrucciones y Programa
3. Ejecución de Instrucciones (ciclo)
4. Sistema de numeración Binario
    1. Interpretar
    2. Representar
    3. Sumar
    4. Restar
5. Sistema de numeración hexadecimal
    1. Interpretar
    2. Representar
6. Conversión de cadenas binarias a hexadecimales

Lógica Proposicional

Componentes de la lógica proposicional

Lógica Proposicional

Componentes de la lógica proposicional

⇓

Variables proposicionales: Enunciados que pueden ser verdaderos o falsos

Operadores: Conjunción, disyunción, negación, etc.

Lógica Proposicional

Componentes de la lógica proposicional

⇓

Variables proposicionales: Enunciados que pueden ser verdaderos o falsos

Operadores: Conjunción, disyunción, negación, etc.

Lógica Proposicional cómo expresión de situaciones

Lógica Proposicional

Lógica Proposicional cómo expresión de situaciones

Lógica Proposicional

⇓

Permite expresar situaciones formalmente

Lógica Proposicional cómo expresión de situaciones

Lógica Proposicional

⇓

Permite expresar situaciones formalmente

A El tanque está lleno
B La llave de paso está cerrada

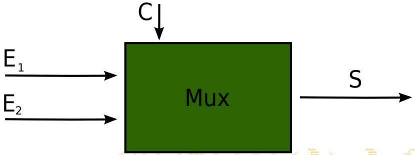
$$A \rightarrow B$$

La operación de la computadora está basada en el almacenamiento y procesamiento de datos binarios

La operación de la computadora está basada en el almacenamiento y procesamiento de datos binarios

Se utilizan circuitos

- para operar sobre datos binarios
- bajo el control de señales de control
- causan el efecto que se necesita en las instrucciones de los programas

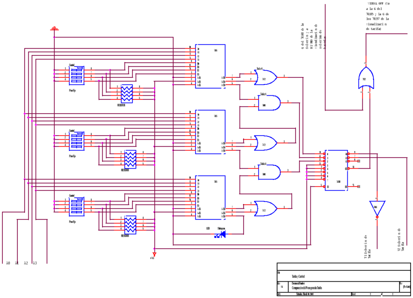


La operación de la computadora está basada en el almacenamiento y procesamiento de datos binarios

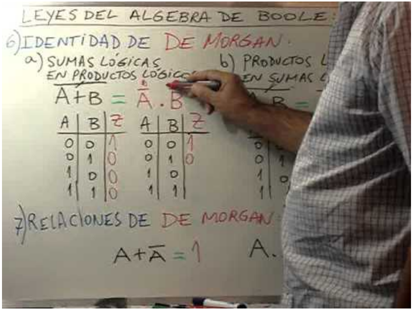
Se utilizan circuitos

Los circuitos se construyen mediante lógica digital

Lógica digital



Álgebra de boole



El **Álgebra de Boole** (basada en la lógica proposicional) permite diseñar y analizar el comportamiento de los circuitos digitales.

Niveles de abstracción

Lógica proposicional para resolver problemas

Niveles de abstracción

Lógica proposicional para resolver problemas

⇓

Lógica digital para automatizar soluciones de problemas

Lógica Digital

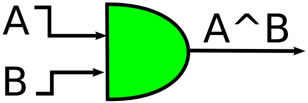
Compuertas lógicas

Compuertas lógicas

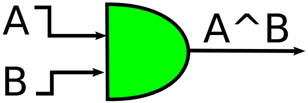
Compuerta lógica

es un dispositivo que implementa una función lógica simple. Traduce un conjunto de entradas (una o más) en **una sola salida**

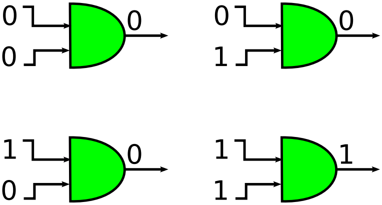
Compuertas lógicas: AND



Compuertas lógicas: AND

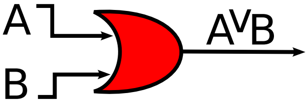


Casos:

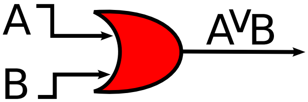


A	B	$A^{\wedge}B$
0	0	0
0	1	0
1	0	0
1	1	1

Compuertas lógicas: OR

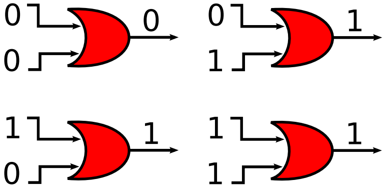


Compuertas lógicas: OR

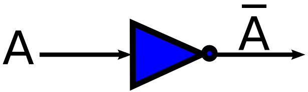


A	B	$A^{\vee}B$
0	0	0
0	1	1
1	0	1
1	1	1

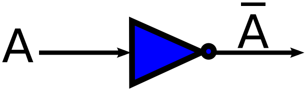
Casos:



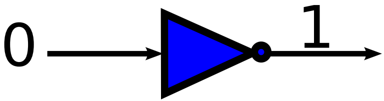
Compuertas lógicas: NOT

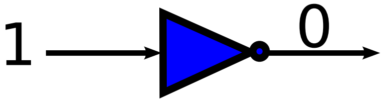


Compuertas lógicas: NOT



Casos:

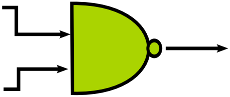




A	$\overline{A}$
0	1
1	0

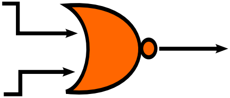
Compuertas lógicas adicionales

1. Compuerta NAND



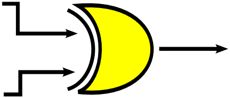
$$a \uparrow b = \overline{a \wedge b}$$

2. Compuerta NOR



$$a \downarrow b = \overline{a \vee b}$$

3. Compuerta XOR



$$a \oplus b = (\overline{a} \wedge b) \vee (a \wedge \overline{b})$$

¿Qué son los circuitos?

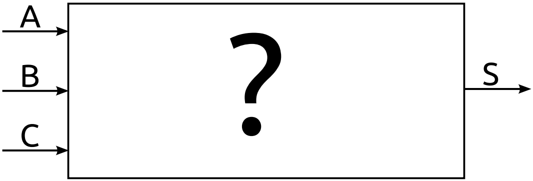
Compuertas y circuitos lógicos

Circuito lógico

- Composición de compuertas
- Traduce un conjunto de entradas en un conjunto de salidas de acuerdo a una o mas funciones lógicas
- Cada salida es estrictamente una función de las entradas
- Las salidas se actualizan de inmediato luego de que cambien las entradas

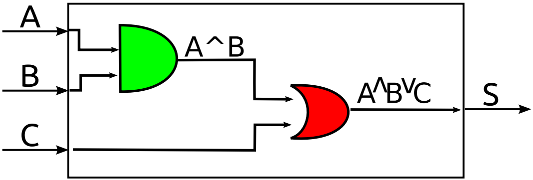
Circuitos lógicos

Ejemplo: ¿Cómo es el circuito de $A^{\wedge}B^{\vee}C$?



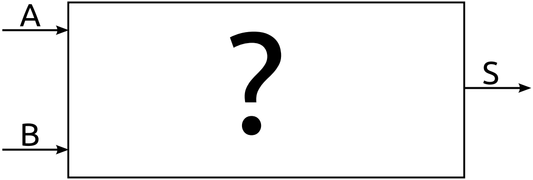
Circuitos lógicos

Ejemplo: ¿Cómo es el circuito de $A^{\wedge} B^{\vee} C$?



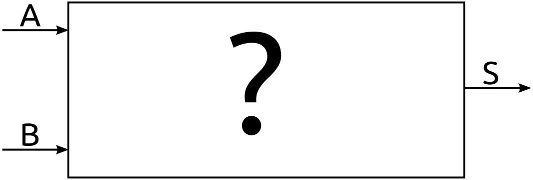
Circuitos lógicos

Ejercicio: ¿Cómo es el circuito de $(A^\wedge \overline{B})$?



Circuitos lógicos

Ejercicio: ¿Cómo es el circuito de $(A^{\wedge}\overline{B})^{\vee}(\overline{A}^{\wedge}B)$?



Circuitos lógicos

¿Cómo se construye un circuito?

Circuitos lógicos

¿Cómo se construye un circuito?

⇓

A partir de su fórmula de verdad

Circuitos lógicos

Los circuitos se construyen a partir de...

a Una tabla de verdad
b Un enunciado en lenguaje natural
c Una fórmula

Circuitos lógicos

Los circuitos se construyen a partir de...

a Una tabla de verdad ⇝ fórmula

b Un enunciado en lenguaje natural

c Una fórmula

Circuitos lógicos

Los circuitos se construyen a partir de...

a Una tabla de verdad ⇢ fórmula

b Un enunciado en lenguaje natural ⇢ tabla ⇢ fórmula

c Una fórmula

Circuitos lógicos

Los circuitos se construyen a partir de...

a Una tabla de verdad ⇢ fórmula
b Un enunciado en lenguaje natural ⇢ tabla ⇢ fórmula
c Una fórmula

Hagamos un circuito

Ejemplo de construcción de un circuito

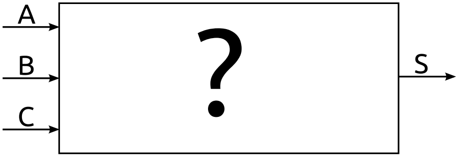
Realizar un circuito de 3 entradas que calcule la función mayoría:

- si dos o mas entradas valen 1: se obtiene un 1
- caso contrario: se obtiene un 0

Ejemplo de construcción de un circuito

Realizar un circuito de 3 entradas que calcule la función mayoría:

- si dos o mas entradas valen 1: se obtiene un 1
- caso contrario: se obtiene un 0



Ejemplo de construcción de un circuito

Ejemplo de construcción de un circuito

Ejemplo de construcción de un circuito

Lenguaje natural ⇝ Tabla de verdad ⇝ Fórmula booleana (SOP)

Ejemplo de construcción de un circuito

Lenguaje natural ⇝ Tabla de verdad ⇝ Fórmula booleana (SOP)

Realizar un circuito de 3 entradas que calcule la función mayoría:

- si dos o mas entradas valen 1: se obtiene un 1
- caso contrario: se obtiene un 0

E_1	E_2	E_3	S
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

Ejemplo de construcción de un circuito

Lenguaje natural ⇢ Tabla de verdad ⇢ Fórmula booleana (SOP)

Ejemplo de construcción de un circuito

Tabla de verdad ⇝ Fórmula booleana (SOP)

1. Construir la tabla de verdad
2. Plantear la fórmula que describe cada caso donde la salida vale 1
3. Unir los casos con disyunción

E_1	E_2	E_3	S
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

Ejemplo de construcción de un circuito

Tabla de verdad ⇝ Fórmula booleana (SOP)

1. Construir la tabla de verdad
2. Plantear la fórmula que describe cada caso donde la salida vale 1
3. Unir los casos con disyunción

E_1	E_2	E_3	S
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1 $(\overline{E}_1 \wedge E_2 \wedge E_3)$
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Ejemplo de construcción de un circuito

Tabla de verdad ⇝ Fórmula booleana (SOP)

1. Construir la tabla de verdad
2. Plantear la fórmula que describe cada caso donde la salida vale 1
3. Unir los casos con disyunción

E_1	E_2	E_3	S
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1 $(E_1 \wedge \overline{E_2} \wedge E_3)$
1	1	0	1
1	1	1	1

Ejemplo de construcción de un circuito

Tabla de verdad ⇝ Fórmula booleana (SOP)

1. Construir la tabla de verdad
2. Plantear la fórmula que describe cada caso donde la salida vale 1
3. Unir los casos con disyunción

E_1	E_2	E_3	S
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 $(E_1 \wedge E_2 \wedge \overline{E}_3)$
1	1	1	1

Ejemplo de construcción de un circuito

Tabla de verdad ⇢ Fórmula booleana (SOP)

1. Construir la tabla de verdad
2. Plantear la fórmula que describe cada caso donde la salida vale 1
3. Unir los casos con disyunción

E_1	E_2	E_3	S
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 $(E_1 \wedge E_2 \wedge E_3)$

Ejemplo de construcción de un circuito

Tabla de verdad ⇝ Fórmula booleana (SOP)

1. Construir la tabla de verdad
2. Plantear la fórmula que describe cada caso donde la salida vale 1
3. Unir los casos con disyunción

E_1	E_2	E_3	S
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

$$s = (\overline{E_1} \wedge E_2 \wedge E_3) \vee (E_1 \wedge \overline{E_2} \wedge E_3) \vee (E_1 \wedge E_2 \wedge \overline{E_3}) \vee (E_1 \wedge E_2 \wedge E_3)$$

Ejemplo de construcción de un circuito

Tabla de verdad ⇝ Fórmula booleana (SOP)

1. Construir la tabla de verdad
2. Plantear la fórmula que describe cada caso donde la salida vale 1
3. Unir los casos con disyunción

E_1	E_2	E_3	S
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1 $(\overline{E}_1 \wedge E_2 \wedge E_3)$
1	0	0	0
1	0	1	1 $(E_1 \wedge \overline{E}_2 \wedge E_3)$
1	1	0	1 $(E_1 \wedge E_2 \wedge \overline{E}_3)$
1	1	1	1 $(E_1 \wedge E_2 \wedge E_3)$

$$s = (\overline{E}_1 \wedge E_2 \wedge E_3) \vee (E_1 \wedge \overline{E}_2 \wedge E_3) \vee (E_1 \wedge E_2 \wedge \overline{E}_3) \vee (E_1 \wedge E_2 \wedge E_3)$$

Suma de Productos

Obtener la Suma de Productos

Suma de Productos

Obtener la Suma de Productos

$$s = (\overline{E}_1 \wedge E_2 \wedge E_3) \vee (E_1 \wedge \overline{E}_2 \wedge E_3) \vee$$

$$\vee (E_1 \wedge E_2 \wedge \overline{E}_3) \vee (E_1 \wedge E_2 \wedge E_3)$$

SOP

(Suma de productos) Fórmula booleana compuesta por disyunciones ($\vee$) entre términos que son conjunciones ($\wedge$) de literales (a ó $\overline{a}$)

Suma de Productos

Obtener la Suma de Productos

$$s = \underbrace{(\overline{E_1} \wedge E_2 \wedge E_3)}_{\text{término}} \vee \underbrace{(E_1 \wedge \overline{E_2} \wedge E_3)}_{\text{término}} \vee$$

$$\vee \underbrace{(E_1 \wedge E_2 \wedge \overline{E_3})}_{\text{término}} \vee \underbrace{(E_1 \wedge E_2 \wedge E_3)}_{\text{término}}$$

SOP

(Suma de productos) Fórmula booleana compuesta por disyunciones ($^\vee$) entre términos que son conjunciones ($^\wedge$) de literales (a ó $\overline{a}$)

Ejemplo de construcción de un circuito

¿Es posible simplificar?

$$s = (\overline{E_1} \wedge E_2 \wedge E_3) \vee (E_1 \wedge \overline{E_2} \wedge E_3) \vee (E_1 \wedge E_2 \wedge \overline{E_3}) \vee (E_1 \wedge E_2 \wedge E_3) =$$

Ejemplo de construcción de un circuito

¿Es posible simplificar?

$$s = (\overline{E}_1^{\wedge} E_2^{\wedge} E_3)^{\vee}(E_1^{\wedge} \overline{E}_2^{\wedge} E_3)^{\vee}(E_1^{\wedge} E_2^{\wedge} \overline{E}_3)^{\vee}(E_1^{\wedge} E_2^{\wedge} E_3) =$$

Por propiedad distributiva:

$$= ((\overline{E}_1^{\wedge} E_2)^{\vee}(E_1^{\wedge} \overline{E}_2))^{\wedge} E_3{}^{\vee}(E_1^{\wedge} E_2)^{\wedge}(E_3^{\vee} \overline{E}_3)$$

Ejemplo de construcción de un circuito

¿Es posible simplificar?

$$s = (\overline{E}_1{}^{\wedge} E_2{}^{\wedge} E_3)^{\vee}(E_1{}^{\wedge}\overline{E}_2{}^{\wedge} E_3)^{\vee}(E_1{}^{\wedge} E_2{}^{\wedge}\overline{E}_3)^{\vee}(E_1{}^{\wedge} E_2{}^{\wedge} E_3) =$$

Por propiedad distributiva:

$$= ((\overline{E}_1{}^{\wedge} E_2)^{\vee}(E_1{}^{\wedge}\overline{E}_2))^{\wedge} E_3{}^{\vee}(E_1{}^{\wedge} E_2)^{\wedge}(E_3{}^{\vee}\overline{E}_3)$$

Por complemento en ${}^{\vee}$:

$$= ((\overline{E}_1{}^{\wedge} E_2)^{\vee}(E_1{}^{\wedge}\overline{E}_2))^{\wedge} E_3{}^{\vee}(E_1{}^{\wedge} E_2)^{\wedge} 1$$

Ejemplo de construcción de un circuito

¿Es posible simplificar?

$$s = (\overline{E}_1 \wedge E_2 \wedge E_3) \vee (E_1 \wedge \overline{E}_2 \wedge E_3) \vee (E_1 \wedge E_2 \wedge \overline{E}_3) \vee (E_1 \wedge E_2 \wedge E_3) =$$

Por propiedad distributiva:

$$= ((\overline{E}_1 \wedge E_2) \vee (E_1 \wedge \overline{E}_2)) \wedge E_3 \vee (E_1 \wedge E_2) \wedge (E_3 \vee \overline{E}_3)$$

Por complemento en $\vee$:

$$= ((\overline{E}_1 \wedge E_2) \vee (E_1 \wedge \overline{E}_2)) \wedge E_3 \vee (E_1 \wedge E_2) \wedge 1$$

Por neutro de $\wedge$:

$$= ((\overline{E}_1 \wedge E_2) \vee (E_1 \wedge \overline{E}_2)) \wedge E_3 \vee (E_1 \wedge E_2)$$

Ejemplo de construcción de un circuito

¿Es posible simplificar?

$$s = (\overline{E}_1 \wedge E_2 \wedge E_3) \vee (E_1 \wedge \overline{E}_2 \wedge E_3) \vee (E_1 \wedge E_2 \wedge \overline{E}_3) \vee (E_1 \wedge E_2 \wedge E_3) =$$

Por propiedad distributiva:

$$= ((\overline{E}_1 \wedge E_2) \vee (E_1 \wedge \overline{E}_2)) \wedge E_3 \vee (E_1 \wedge E_2) \wedge (E_3 \vee \overline{E}_3)$$

Por complemento en $\vee$:

$$= ((\overline{E}_1 \wedge E_2) \vee (E_1 \wedge \overline{E}_2)) \wedge E_3 \vee (E_1 \wedge E_2) \wedge 1$$

Por neutro de $\wedge$:

$$= ((\overline{E}_1 \wedge E_2) \vee (E_1 \wedge \overline{E}_2)) \wedge E_3 \vee (E_1 \wedge E_2)$$

Por definición de $\oplus$:

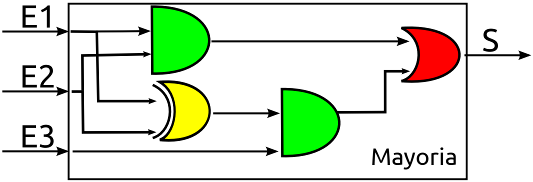
$$= (E_1 \oplus E_2) \wedge E_3 \vee (E_1 \wedge E_2)$$

Ejemplo de construcción de un circuito

$$(E_1 \oplus E_2)^\wedge E_3{}^\vee(E_1^\wedge E_2)$$

Ejemplo de construcción de un circuito

$$(E_1 \oplus E_2)^{\wedge} E_3{}^{\vee} (E_1^{\wedge} E_2)$$



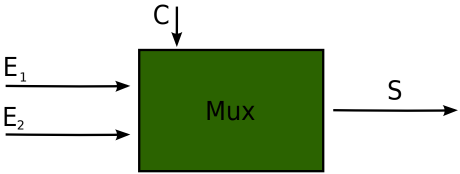
Circuitos mas usados

Multiplexor simple

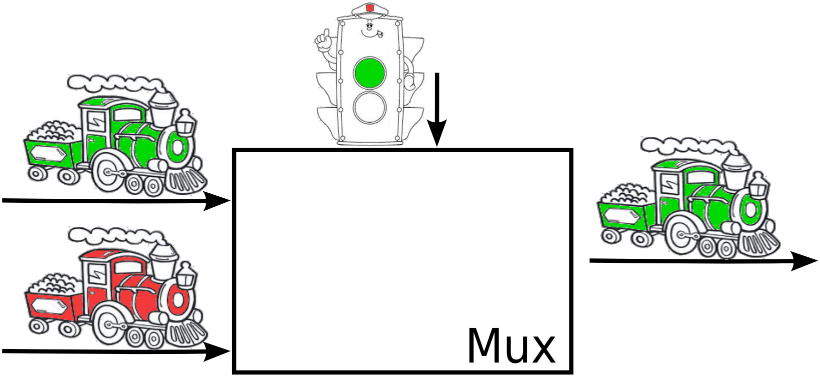
Objetivo Proyectar una de las entradas en la salida, a partir la configuración del control

Entradas 2 entradas, una línea de control

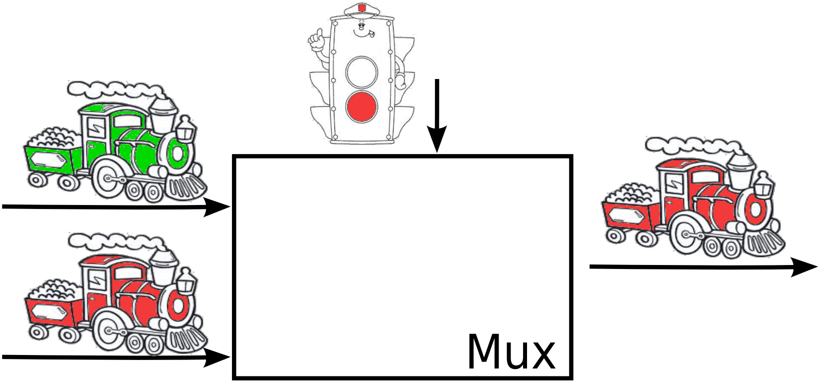
Salida 1 salida



Multiplexor simple: La idea



Multiplexor simple: La idea



Multiplexor simple

Lenguaje natural ⇝ Tabla de verdad ⇝ Fórmula booleana

Multiplexor simple

Lenguaje natural ⇝ Tabla de verdad

Tabla abreviada:

C	S
0	e_1
1	e_2

Multiplexor simple

Lenguaje natural ⇝ Tabla de verdad

Tabla abreviada:

C	S
0	e_1
1	e_2

Tabla de verdad:

C	E_1	E_2	S
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Multiplexor simple

Lenguaje natural ⇢ Tabla de verdad ⇢ Fórmula booleana

Multiplexor simple

Tabla de verdad ⇝ Fórmula booleana

C	E_1	E_2	S
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Multiplexor simple

Tabla de verdad ⇝ Fórmula booleana

C	E_1	E_2	S	
0	0	0	0	
0	0	1	0	
0	1	0	1	$(\overline{C} \wedge E_1 \wedge \overline{E}_2)$
0	1	1	1	
1	0	0	0	
1	0	1	1	
1	1	0	0	
1	1	1	1	

Multiplexor simple

Tabla de verdad ⇢ Fórmula booleana

C	E_1	E_2	S
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1 $(\overline{C} \wedge E_1 \wedge E_2)$
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Multiplexor simple

Tabla de verdad Fórmula booleana

C	E_1	E_2	S	
0	0	0	0	
0	0	1	0	
0	1	0	1	
0	1	1	1	
1	0	0	0	
1	0	1	1	$(C \wedge \overline{E}_1 \wedge E_2)$
1	1	0	0	
1	1	1	1	

Multiplexor simple

Tabla de verdad ⇝ Fórmula booleana

C	E_1	E_2	S
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1 $(C \wedge E_1 \wedge E_2)$

Multiplexor simple

Tabla de verdad ⇝ Fórmula booleana

C	E_1	E_2	S
0	0	0	0
0	0	1	0
0	1	0	1 $(\overline{C} \wedge E_1 \wedge \overline{E}_2)$
0	1	1	1 $(\overline{C} \wedge E_1 \wedge E_2)$
1	0	0	0
1	0	1	1 $(C \wedge \overline{E}_1 \wedge E_2)$
1	1	0	0
1	1	1	1 $(C \wedge E_1 \wedge E_2)$

$$s = (\overline{C} \wedge E_1 \wedge \overline{E}_2) \vee (\overline{C} \wedge E_1 \wedge E_2) \vee (C \wedge \overline{E}_1 \wedge E_2) \vee (C \wedge E_1 \wedge E_2)$$

Multiplexor simple

¡Simplificar!

$$s = (\overline{C}^{\wedge} E_1^{\wedge} \overline{E}_2)^{\vee} (\overline{C}^{\wedge} E_1^{\wedge} E_2)^{\vee} (C^{\wedge} \overline{E}_1^{\wedge} E_2)^{\vee} (C^{\wedge} E_1^{\wedge} E_2)$$

Multiplexor simple

¡Simplificar!

$$s = (\overline{C} \wedge E_1 \wedge \overline{E}_2) \vee (\overline{C} \wedge E_1 \wedge E_2) \vee (C \wedge \overline{E}_1 \wedge E_2) \vee (C \wedge E_1 \wedge E_2)$$

Por distributiva:

$$(\overline{C} \wedge E_1) \wedge (\overline{E}_2 \vee E_2) \vee (C \wedge \overline{E}_1 \wedge E_2) \vee (C \wedge E_1 \wedge E_2)$$

Multiplexor simple

¡Simplificar!

$$s = (\overline{C} \wedge E_1 \wedge \overline{E}_2) \vee (\overline{C} \wedge E_1 \wedge E_2) \vee (C \wedge \overline{E}_1 \wedge E_2) \vee (C \wedge E_1 \wedge E_2)$$

Por distributiva:

$$(\overline{C} \wedge E_1) \wedge (\overline{E}_2 \vee E_2) \vee (C \wedge \overline{E}_1 \wedge E_2) \vee (C \wedge E_1 \wedge E_2)$$

Por distributiva:

$$(\overline{C} \wedge E_1) \wedge (\overline{E}_2 \vee E_2) \vee (C \wedge E_2) \wedge (\overline{E}_1 \vee E_1)$$

Multiplexor simple

¡Simplificar!

$$s = (\overline{C}^{\wedge} E_1^{\wedge} \overline{E}_2)^{\vee} (\overline{C}^{\wedge} E_1^{\wedge} E_2)^{\vee} (C^{\wedge} \overline{E}_1^{\wedge} E_2)^{\vee} (C^{\wedge} E_1^{\wedge} E_2)$$

Por distributiva:

$$(\overline{C}^{\wedge} E_1)^{\wedge} (\overline{E}_2^{\vee} E_2)^{\vee} (C^{\wedge} \overline{E}_1^{\wedge} E_2)^{\vee} (C^{\wedge} E_1^{\wedge} E_2)$$

Por distributiva:

$$(\overline{C}^{\wedge} E_1)^{\wedge} (\overline{E}_2^{\vee} E_2)^{\vee} (C^{\wedge} E_2)^{\wedge} (\overline{E}_1^{\vee} E_1)$$

Por complemento de la $^{\vee}$:

$$(\overline{C}^{\wedge} E_1)^{\wedge} 1^{\vee} (C^{\wedge} E_2)^{\wedge} 1$$

Multiplexor simple

¡Simplificar!

$$s = (\overline{C}^{\wedge} E_1^{\wedge} \overline{E}_2)^{\vee} (\overline{C}^{\wedge} E_1^{\wedge} E_2)^{\vee} (C^{\wedge} \overline{E}_1^{\wedge} E_2)^{\vee} (C^{\wedge} E_1^{\wedge} E_2)$$

Por distributiva:

$$(\overline{C}^{\wedge} E_1)^{\wedge} (\overline{E}_2^{\vee} E_2)^{\vee} (C^{\wedge} \overline{E}_1^{\wedge} E_2)^{\vee} (C^{\wedge} E_1^{\wedge} E_2)$$

Por distributiva:

$$(\overline{C}^{\wedge} E_1)^{\wedge} (\overline{E}_2^{\vee} E_2)^{\vee} (C^{\wedge} E_2)^{\wedge} (\overline{E}_1^{\vee} E_1)$$

Por complemento de la $^{\vee}$:

$$(\overline{C}^{\wedge} E_1)^{\wedge} 1^{\vee} (C^{\wedge} E_2)^{\wedge} 1$$

Por neutro de la $^{\wedge}$:

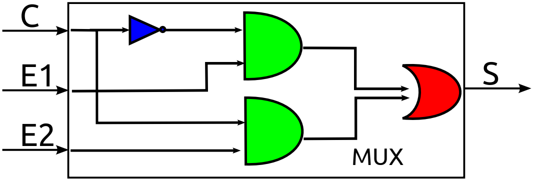
$$(\overline{C}^{\wedge} E_1)^{\vee} (C^{\wedge} E_2)$$

Multiplexor simple

$$(\overline{C}^{\wedge} E_1)^{\vee}(C^{\wedge} E_2)$$

Multiplexor simple

$$(\overline{C}^{\wedge} E_1)^{\vee}(C^{\wedge} E_2)$$

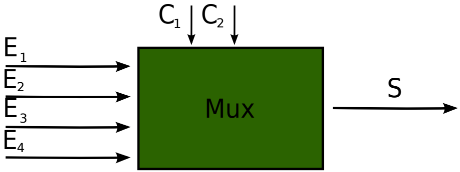


Multiplexor complejo

Objetivo Proyectar una de las entradas en la salida, a partir la configuración del control

Entradas 4 entradas

Salida 1 salida



Multiplexor complejo

Tabla abreviada:

C_1	C_2	S
0	0	e_1
0	1	e_2
1	0	e_3
1	1	e_4

Multiplexor complejo

Tabla abreviada:

C_1	C_2	S
0	0	e_1
0	1	e_2
1	0	e_3
1	1	e_4

Tabla de verdad:

C_1	C_2	E_1	E_2	E_3	E_4	S
0	0	0	0	0	0	0

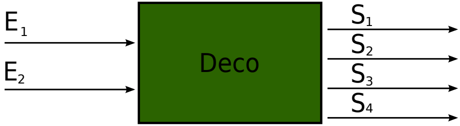
¡Completar de Tarea!

Decodificador

Objetivo Traduce un *código* de 2 bits en uno de 4 valores

Entrada 2 bits de la cadena de entrada (2 entradas)

Salida 4 líneas de salida



Decodificador

E_1	E_2	S_1	S_2	S_3	S_4

Tabla de verdad:

Decodificador

Tabla de verdad:

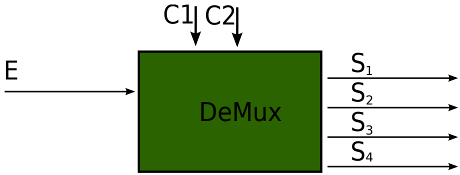
E_1	E_2	S_1	S_2	S_3	S_4
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Demultiplexor

Objetivo Permite configurar por qué salida se proyecta la entrada.

Entrada 1 línea de entrada, y dos líneas de control

Salida 4 líneas de salida



Demultiplexor

E	C_1	C_2	S_1	S_2	S_3	S_4

Tabla de verdad:

Demultiplexor

Tabla de verdad:

E	C_1	C_2	S_1	S_2	S_3	S_4
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

Demultiplexor

Tabla de verdad:

E	C_1	C_2	S_1	S_2	S_3	S_4
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

¿Cómo se contruye el circuito?

Circuitos aritméticos

Circuitos aritméticos

- La ALU se puede implementar mediante circuitos
- Cada operación aritmética podría resolverse con un circuito

Circuitos aritméticos: Half adder

Objetivo Suma 2 bits

Entradas Los bits a sumar

Salida El bit resultado y el acarreo o **carry** (C)

Circuitos aritméticos: Half adder

Tabla de verdad del Half adder

A	B	R	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Circuitos aritméticos: Half adder

Tabla de verdad del Half adder

A	B	R	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$\begin{array}{r} 0 \\ +\ 0 \\ \hline 0 \end{array}$$

Circuitos aritméticos: Half adder

Tabla de verdad del Half adder

A	B	R	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$\begin{array}{r} 1 \\ + \; 0 \\ \hline 1 \end{array}$$

Circuitos aritméticos: Half adder

Tabla de verdad del Half adder

A	B	R	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$\begin{array}{r} 0 \\ +\ 1 \\ \hline 1 \end{array}$$

Circuitos aritméticos: Half adder

Tabla de verdad del Half adder

A	B	R	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$\begin{array}{rl} & 1 \quad \text{"me llevo 1"} \\ + & 1 \\ \hline & 0 \end{array}$$

Circuitos aritméticos: Half adder

Fórmula del Half adder

Fórmula para el resultado:

A	B	R	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Fórmula para el carry:

A	B	R	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Circuitos aritméticos: Half adder

Fórmula del Half adder

Fórmula para el resultado:

A	B	R	C
0	0	0	0
0	1	1 $\overline{A} \wedge B$	0
1	0	1 $A \wedge \overline{B}$	0
1	1	0	1

Fórmula para el carry:

A	B	R	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Circuitos aritméticos: Half adder

Fórmula del Half adder

Fórmula para el resultado:

A	B	R	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$R = (\overline{A}^{\wedge} B)^{\vee}(A^{\wedge}\overline{B})$$

Fórmula para el carry:

A	B	R	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Circuitos aritméticos: Half adder

Fórmula del Half adder

Fórmula para el resultado:

A	B	R	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$R = (\overline{A}^{\wedge} B)^{\vee}(A^{\wedge}\overline{B})$$

Fórmula para el carry:

A	B	R	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1 $A^{\wedge} B$

Circuitos aritméticos: Half adder

Fórmula del Half adder

Fórmula para el resultado:

A	B	R	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$R = (\overline{A}^{\wedge} B)^{\vee}(A^{\wedge}\overline{B})$$

Fórmula para el carry:

A	B	R	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$C = (A^{\wedge} B)$$

Circuitos aritméticos: Full Adder

Objetivo Suma 2 bits, considerando el carry anterior

Entradas Los bits a sumar, carry anterior

Salida El bit resultado y el bit de carry

Circuitos aritméticos: Full Adder

Tabla de verdad del Full adder

$CAnt$	A	B	R	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Circuitos aritméticos: Full Adder

Tabla de verdad del Full adder

$CAnt$	A	B	R	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

C anterior=0
$$\begin{array}{r} 0 \\ + \; 0 \\ \hline 0 \end{array}$$
C=0

C anterior=0
$$\begin{array}{r} 1 \\ + \; 0 \\ \hline 1 \end{array}$$
C=0

C anterior=0
$$\begin{array}{r} 0 \\ + \; 1 \\ \hline 1 \end{array}$$
C=0

C anterior=0
$$\begin{array}{r} 1 \\ + \; 1 \\ \hline 0 \end{array}$$
C=1

Circuitos aritméticos: Full Adder

Tabla de verdad del Full adder

$CAnt$	A	B	R	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

C anterior=0
$$+\begin{array}{r} 0 \\ 0 \\ \hline 0 \end{array}$$
C=0

C anterior=0
$$+\begin{array}{r} 1 \\ 0 \\ \hline 1 \end{array}$$
C=0

C anterior=0
$$+\begin{array}{r} 0 \\ 1 \\ \hline 1 \end{array}$$
C=0

C anterior=0
$$+\begin{array}{r} 1 \\ 1 \\ \hline 0 \end{array}$$
C=1

Circuitos aritméticos: Full Adder

Tabla de verdad del Full adder

CAnt	*A*	*B*	*R*	*C*
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

C anterior=0
$$+\begin{array}{r}0\\0\\\hline 0\end{array}\ \text{C=0}$$

C anterior=0
$$+\begin{array}{r}1\\0\\\hline 1\end{array}\ \text{C=0}$$

C anterior=0
$$+\begin{array}{r}0\\1\\\hline 1\end{array}\ \text{C=0}$$

C anterior=0
$$+\begin{array}{r}1\\1\\\hline 0\end{array}\ \text{C=1}$$

Circuitos aritméticos: Full Adder

Tabla de verdad del Full adder

$CAnt$	A	B	R	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

C anterior=0
$$\begin{array}{r} 0 \\ +\ 0 \\ \hline 0 \end{array}\ C=0$$

C anterior=0
$$\begin{array}{r} 1 \\ +\ 0 \\ \hline 1 \end{array}\ C=0$$

C anterior=0
$$\begin{array}{r} 0 \\ +\ 1 \\ \hline 1 \end{array}\ C=0$$

C anterior=0
$$\begin{array}{r} 1 \\ +\ 1 \\ \hline 0 \end{array}\ C=1$$

Circuitos aritméticos: Full Adder

Tabla de verdad del Full adder

$CAnt$	A	B	R	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

C anterior=1
$$\begin{array}{r} 0 \\ +\ 0 \\ \hline 1 \end{array}$$
C=0

C anterior=1
$$\begin{array}{r} 1 \\ +\ 0 \\ \hline 0 \end{array}$$
C=1

C anterior=1
$$\begin{array}{r} 0 \\ +\ 1 \\ \hline 0 \end{array}$$
C=1

C anterior=1
$$\begin{array}{r} 1 \\ +\ 1 \\ \hline 1 \end{array}$$
C=1

Circuitos aritméticos: Full Adder

Tabla de verdad del Full adder

$CAnt$	A	B	R	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

C anterior=1
$$\begin{array}{r} 0 \\ +\ 0 \\ \hline 1 \end{array}\ \text{C=0}$$

C anterior=1
$$\begin{array}{r} 1 \\ +\ 0 \\ \hline 0 \end{array}\ \text{C=1}$$

C anterior=1
$$\begin{array}{r} 0 \\ +\ 1 \\ \hline 0 \end{array}\ \text{C=1}$$

C anterior=1
$$\begin{array}{r} 1 \\ +\ 1 \\ \hline 1 \end{array}\ \text{C=1}$$

Circuitos aritméticos: Full Adder

Tabla de verdad del Full adder

$CAnt$	A	B	R	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

C anterior=1
$$\begin{array}{r} 0 \\ +\ 0 \\ \hline 1 \end{array}$$
C=0

C anterior=1
$$\begin{array}{r} 1 \\ +\ 0 \\ \hline 0 \end{array}$$
C=1

C anterior=1
$$\begin{array}{r} 0 \\ +\ 1 \\ \hline 0 \end{array}$$
C=1

C anterior=1
$$\begin{array}{r} 1 \\ +\ 1 \\ \hline 1 \end{array}$$
C=1

Circuitos aritméticos: Full Adder

Tabla de verdad del Full adder

$CAnt$	A	B	R	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

C anterior=1

$$\begin{array}{r} 0 \\ +\ 0 \\ \hline 1 \end{array} \quad C=0$$

C anterior=1

$$\begin{array}{r} 1 \\ +\ 0 \\ \hline 0 \end{array} \quad C=1$$

C anterior=1

$$\begin{array}{r} 0 \\ +\ 1 \\ \hline 0 \end{array} \quad C=1$$

C anterior=1

$$\begin{array}{r} 1 \\ +\ 1 \\ \hline 1 \end{array} \quad C=1$$

Circuitos aritméticos: Full adder

Fórmulas del Full adder

Fórmula para el resultado:

$CAnt$	A	B	R	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Fórmula para el carry:

$CAnt$	A	B	R	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Circuitos aritméticos: Full adder

Fórmulas del Full adder

Fórmula para el resultado:

$CAnt$	A	B	R	C
0	0	0	0	0
0	0	1	1 $\overline{CAnt} \wedge \overline{A} \wedge B$	0
0	1	0	1 $\overline{CAnt} \wedge A \wedge \overline{B}$	0
0	1	1	0	1
1	0	0	1 $CAnt \wedge \overline{A} \wedge \overline{B}$	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1 $CAnt \wedge A \wedge B$	1

Fórmula para el carry:

$CAnt$	A	B	R	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Circuitos aritméticos: Full adder

Fórmulas del Full adder

Fórmula para el resultado:

$CAnt$	A	B	R	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Fórmula para el carry:

$CAnt$	A	B	R	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1 $\overline{CAnt}^\wedge A^\wedge B$
1	0	0	1	0
1	0	1	0	1 $CAnt^\wedge \overline{A}^\wedge B$
1	1	0	0	1 $CAnt^\wedge A^\wedge \overline{B}$
1	1	1	1	1 $CAnt^\wedge A^\wedge B$

Circuitos aritméticos

¿Cómo se suman cadenas de mas de un bit?

Circuitos aritméticos

¿Cómo se suman cadenas de mas de un bit?

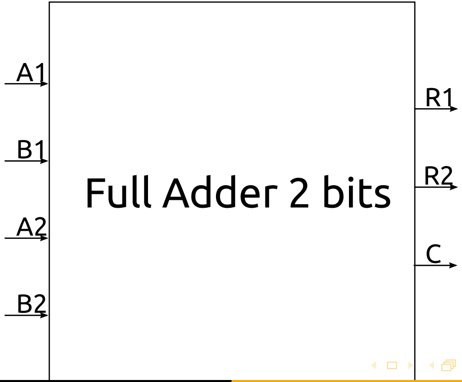
Usando múltiples Full Adders

Circuitos aritméticos

¿Cómo se suman cadenas de mas de un bit?

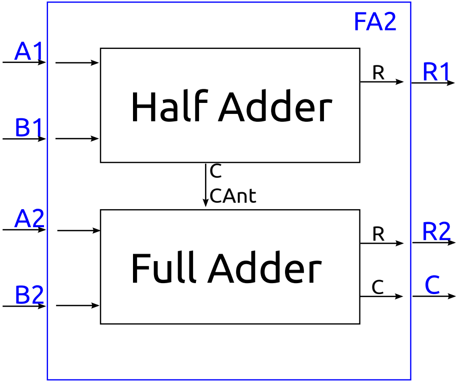
Circuitos aritméticos

¿Cómo se suman cadenas de mas de un bit?



Circuitos aritméticos

¿Cómo se suman cadenas de mas de un bit?



Restador de un bit

Completar la tabla de verdad

A	B	R	C

Restador de un bit

Completar la tabla de verdad

A	B	R	C
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Restador de un bit

Completar la tabla de verdad

A	B	R	C
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Completar las SOP

Restador de un bit

Restador de un bit: con carry anterior

Restador de un bit

Restador de un bit: con carry anterior

CAnt	*A*	*B*	*R*	*C*
0	0	0	0	0
0	0	1	1	1
0	1	0	1	0
0	1	1	0	0
1	0	0	1	1
1	0	1	0	1
1	1	0	0	0
1	1	1	1	1

C anterior=1
$$\begin{array}{r} 0 \\ -\ 0 \\ \hline 1 \end{array}\ \text{C=1}$$

C anterior=1
$$\begin{array}{r} 1 \\ -\ 0 \\ \hline 0 \end{array}\ \text{C=0}$$

C anterior=1
$$\begin{array}{r} 0 \\ -\ 1 \\ \hline 0 \end{array}\ \text{C=1}$$

C anterior=1
$$\begin{array}{r} 1 \\ -\ 1 \\ \hline 1 \end{array}\ \text{C=1}$$

Restador de un bit

Restador de un bit: con carry anterior

$CAnt$	A	B	R	C
0	0	0	0	0
0	0	1	1	1
0	1	0	1	0
0	1	1	0	0
1	0	0	1	1
1	0	1	0	1
1	1	0	0	0
1	1	1	1	1

Completar las SOP

Redondeando...

Trabajo de investigación

Trabajo de investigación

¿Qué son los circuitos secuenciales flip flop S-R y flip flop J-K?

Trabajo de investigación

¿Qué son los circuitos secuenciales flip flop S-R y flip flop J-K?

Entrega el martes 27/8

¿Preguntas?