Curso

Matemáticas Discretas I

UNIVERSIDAD DE ANTIOQUIA

Clase 13 – Sistemas Numéricos

Agenda

- Contextualización
- Representación de la información
- Sistema numéricos posicionales
- Conversión entre sistemas numéricos
- Suma binaria

Agenda

- **Contextualización**
- Representación de la información
- Sistema numéricos posicionales
- Conversión entre sistemas numéricos
- Suma binaria

Contextualización

How exactly does binary code work?

How exactly does binary code work?

José Américo N L F Freitas

(link)

Agenda

- Contextualización
- **Representación de la información**
- Sistema numéricos posicionales
- Conversión entre sistemas numéricos
- Suma binaria

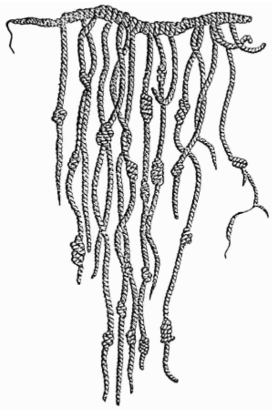
Representación de la información

UNIVERSIDAD DE ANTIOQUIA

Representación de los datos

- Es una representación física de la información.
- Los datos pueden ser almacenados, transmitidos y/o procesados.

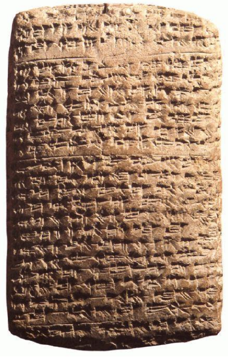
Palo tallado (link)



Quipu (link)

Ficha de arcilla (link)

Jeroglífico (link)



Escritura cuneiforme (link)

Abaco (link)

Representación de la información

UNIVERSIDAD DE ANTIOQUIA

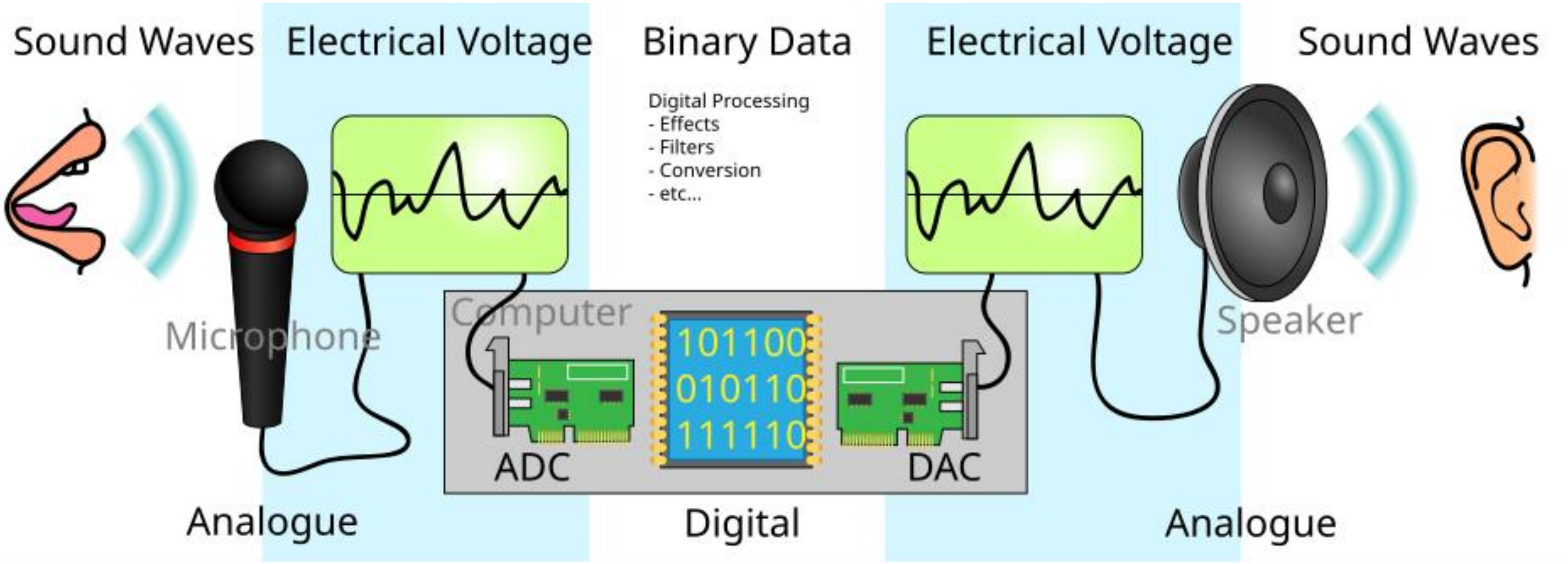
Representación de los datos

- A pesar de que la información puede ser muy diversa (imágenes, sonidos, números, códigos), es posible representarla con niveles de voltaje.
  - Siguen siendo una representación física de la información.
  - Los datos pueden ser almacenados, transmitidos y procesados
- En las computadoras actuales la forma más simple es mediante dos valores:
  - **Falso (0)**: 0V
  - **Verdadero (1)**: 5V

Representación de la información

UNIVERSIDAD DE ANTIOQUIA

Conversión Análogo/Digital (A/D) y Digital/Análogo (D/A)



Representación de la información

Señal binaria

- Es una señal digital que sólo tiene dos valores posibles:
  - Alto, Bajo
  - ON, OFF
  - Verdadero, Falso
  - Sí, No
- Los niveles lógicos en un sistema digital se suelen representar con los dígitos binarios ‘1’ y ‘0’
- Desde el punto de vista eléctrico, los valores binarios corresponden a niveles de voltaje.

Representación de la información

UNIVERSIDAD DE ANTIOQUIA

Ventaja de usar una representación binaria

- Fácil implementación en hardware electrónico (transistor operando como interruptor controlado eléctricamente)
- Buena tolerancia al ruido eléctrico. (Comunicaciones)

Agenda

UNIVERSIDAD
DE ANTIOQUIA

- Contextualización
- Representación de la información
- **Sistema numéricos posicionales**
- Conversión entre sistemas numéricos
- Suma binaria

Sistemas numéricos posicionales

Conceptos clave

- En los sistemas numéricos posicionales, un número se representa mediante un cadena de dígitos en la cual la posición de cada dígito tiene un peso asociado.
- Para un número cualquiera N, compuesto por una parte entera y una parte fraccionaria en una base b, su valor se representa como:

$$N_b = \sum_{i=-m}^{n} d_i \times b^i$$

Donde:
- $\boldsymbol{N}$ es el valor total del número
- $\boldsymbol{b}$ es la base del sistema numérico (un número entero mayor que 1).
- $\boldsymbol{d_i}$ es el dígito en la posición $\boldsymbol{i}$.
- El valor de cada dígito debe ser $\boldsymbol{0 \leq i < d}$.
- $\boldsymbol{n}$ es la posición del dígito más significativo de la parte entera.
- $\boldsymbol{m}$ es el número de dígitos de la parte fraccionaria.

- **Desglose de la formula**: La fórmula se puede entender mejor si la separamos en su parte entera y su parte fraccionaria:

$$N_b = (d_n d_{n-1} \ldots d_1 d_0 . d_{-1} d_{-2} \ldots d_{-m+1} d_{-m})$$

Parte fraccionaria: $d_{-1} \times b^{-1} + d_{-2} \times b^{-2} + \cdots + d_{-m+1} \times b^{-m+1} + d_{-m} \times b^{-m}$

Parte entera: $d_n \times b^n + d_{n-1} \times b^{n-1} + \cdots + d_1 \times b^1 + d_0 \times b^0$

Sistemas numéricos posicionales

UNIVERSIDAD DE ANTIOQUIA

Ejemplos

Exprese en notación posicional los siguientes números:

1. $432_{10} = 4 \times 10^2 + 3 \times 10^1 + 2 \times 10^0 = 4 \times 100 + 3 \times 10 + 2 \times 1$
2. $432_8 = 4 \times 8^2 + 3 \times 8^1 + 2 \times 8^0 = 4 \times 64 + 3 \times 8 + 2 \times 1$
3. $432_{16} = 4 \times 16^2 + 3 \times 16^1 + 2 \times 16^0 = 4 \times 256 + 3 \times 16 + 2 \times 1$
4. $25.4_{10} = 2 \times 10^1 + 5 \times 10^0 + 4 \times 10^{-1} = 2 \times 10 + 5 \times 1 + 2 \times 0.1$
5. $1101_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1$

Sistemas numéricos posicionales

UNIVERSIDAD DE ANTIOQUIA

Sistema numérico decimal (base 10)

- Es el que usamos a diario.
- Se llama decimal porque utiliza la base 10, es decir, diez símbolos distintos
- Dígitos:

$$0,1,2,3,4,5,6,7,8,9$$

- Representación posicional:

$$N = N_{10} = \sum_{-m}^{n} d_i \times 10^i$$

- **Ejemplo**: Represente el 5238

$$\begin{aligned} 5238 &= 5 \times 10^3 + 2 \times 10^2 + 3 \times 10^1 + 8 \times 10^0 \\ &= 5 \times 1000 + 2 \times 100 + 3 \times 10 + 8 \times 1 \\ &= 5238 \end{aligned}$$

Potencia de 10	10^3	10^2	10^1	10^0
Forma decimal	1000	100	10	1
Digito i	5	2	3	8

Sistemas numéricos posicionales

Sistema numérico binario (base 2)

- Solo utiliza dos dígitos para representar cualquier número: 0 y 1. A cada uno de estos dígitos se le conoce como bit (contracción del inglés Binary Digit).
- Dígitos:

$$0,1$$

- Representación posicional:

$$N_2 = \sum_{-m}^{n} d_i \times 2^i$$

- **Ejemplo**: Represente el 0110_2

$$\begin{aligned} 0110_2 &= 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 \\ &= 0 \times 8 + 1 \times 4 + 1 \times 2 + 0 \times 1 \\ &= 0 + 4 + 2 + 0 \\ &= 6_{10} \end{aligned}$$

Potencia de 2	2^3	2^2	2^1	2^0
Forma decimal	8	4	2	1
Digito i	0	1	1	0

Sistemas numéricos posicionales

UNIVERSIDAD DE ANTIOQUIA

Sistema numérico octal (base 8)

- Sistema de numeración que utiliza 8 dígitos.
- Dígitos:

$$0,1,2,3,4,5,6,7$$

- Representación posicional:

$$N_8 = \sum_{-m}^{n} d_i \times 8^i$$

- **Ejemplo**: Represente el 347_8

$$\begin{aligned} 347_8 &= 3 \times 8^2 + 4 \times 8^1 + 7 \times 8^0 \\ &= 3 \times 64 + 4 \times 8 + 7 \times 1 \\ &= 192 + 32 + 7 \\ &= 231_{10} \end{aligned}$$

Potencia de 8	8^2	8^1	2^0
Forma decimal	64	8	1
Digito i	3	4	7

Sistemas numéricos posicionales

UNIVERSIDAD DE ANTIOQUIA

Sistema numérico hexadecimal (base 16)

- Utiliza **16** símbolos para representar cualquier cantidad. Como solo existen 10 dígitos numéricos, este sistema se complementa con las primeras seis letras del alfabeto.
- Dígitos:

$$0,1,2,3,4,5,6,7,8,9, A, B, C, D, E, F$$

- Representación posicional:

$$N_{16} = \sum_{-m}^{n} d_i \times 16^i$$

- **Ejemplo**: Represente el $1AD_{16}$

$$\begin{aligned} 1AD_{16} &= 1 \times 16^2 + 10 \times 16^1 + 13 \times 16^0 \\ &= 1 \times 256 + 10 \times 16 + 13 \times 1 \\ &= 256 + 160 + 13 \\ &= 429_{10} \end{aligned}$$

Potencia de 16	16^2	16^1	16^0
Forma decimal	256	16	1
Digito i	1	A	D

Hexadecimal	Decimal
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	11
C	12
D	13
E	14
F	15

Sistemas numéricos posicionales

UNIVERSIDAD DE ANTIOQUIA

Tabla resumen – Principales sistemas numéricos

Característica	Decimal	Binario	Octal	Hexadecimal
Base	10	2	8	16
Dígitos Utilizados	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	0, 1	0, 1, 2, 3, 4, 5, 6, 7	0-9 y A, B, C, D, E, F
Relación con Binario	Ninguna directa. Es el sistema de referencia humano.	Es el **sistema fundamental** de la computación.	**1 dígito octal = 3 bits** ($2^3 = 8$)	**1 dígito hexadecimal = 4 bits** ($2^4 = 16$)
Uso Principal	Cálculos cotidianos, finanzas, vida diaria.	Lenguaje nativo del hardware de la computadora (circuitos, lógica).	Permisos de archivos en Unix/Linux, algunos sistemas embebidos.	Direcciones de memoria, colores web (CSS), direcciones MAC, depuración.

Agenda

UNIVERSIDAD
DE ANTIOQUIA

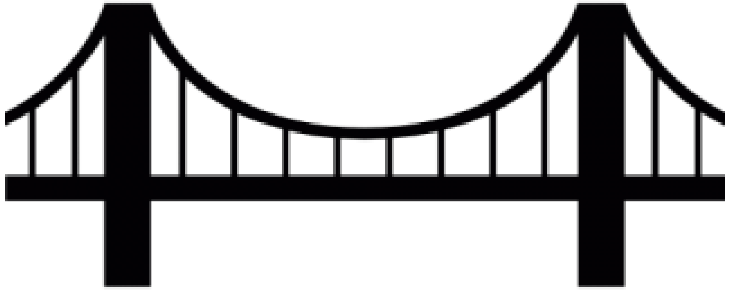
- Contextualización
- Representación de la información
- Sistema numéricos posicionales
- **Conversión entre sistemas numéricos**
- Suma binaria

Conversión entre sistemas numéricos

Puente universal

Este método funciona para convertir CUALQUIER base a CUALQUIER otra, usando el sistema decimal (base 10) como intermediario. Para ello se procede en dos partes:

1. **Cualquier Base** → **Decimal**: Método de expansión polinómica.
2. **Decimal** → **Cualquier Base**: Método de divisiones sucesivas.

$Base\ \boldsymbol{m}$  $Base\ \boldsymbol{n}$

$\boldsymbol{Decimal}$

Conversión entre sistemas numéricos

UNIVERSIDAD DE ANTIOQUIA

Expansión Polinómica (Cualquier Base → Decimal)

Regla: Se multiplica cada dígito por su base elevada a la posición que ocupa, y suma todos los resultados.

Ejemplo: Convertir $3B7_{16}$ a base 10.

$$3B7_{16} \rightarrow N_{10}$$

$$\begin{aligned} 3B7_{16} &= 3 \times 16^2 + B \times 16^1 + 7 \times 16^0 \\ &= 3 \times 16^2 + 11 \times 16^1 + 7 \times 16^0 \\ &= 3 \times 256 + 11 \times 16 + 7 \times 1 \\ &= 951 \end{aligned}$$

Potencia de 16	16^2	16^1	16^0
Forma decimal	256	16	1
Digito hexadecimal i	3	B	7
decimal i	3	11	7

Hexadecimal	Decimal
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	11
C	12
D	13
E	14
F	15

Conversión entre sistemas numéricos

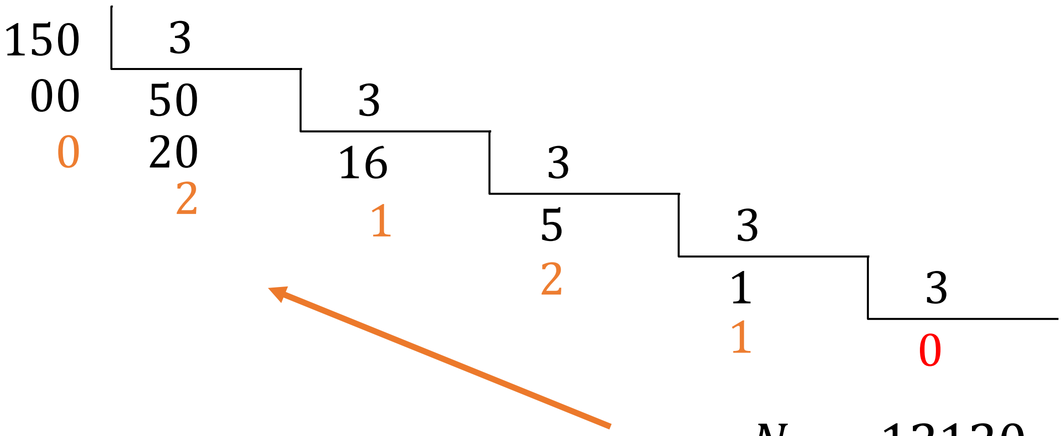
UNIVERSIDAD DE ANTIOQUIA

Método de divisiones sucesivas (Decimal → Cualquier Base)

Regla: Se divide el número decimal entre la base de destino repetidamente hasta que el cociente sea 0. El resultado se forma con los residuos, leídos desde el último hasta el primero.

Ejemplo: Convertir 150_{10} a base 3.

$$150_{10} \rightarrow N_3$$



$$150_{10} = N_3$$

$$N_3 = 12120_3$$

Conversión entre sistemas numéricos

Ejemplo

Regla: Hallar el numero 42_5 a base 9.

Paso 1: Base 5 → Base 10

$$
\begin{aligned}
42_5 &= 4 \times 5^1 + 2 \times 5^0 \\
&= 4 \times 5 + 2 \times 1 \\
&= 20 + 2 \\
&= 22_{10}
\end{aligned}
$$

Potencia de 5	5^1	16^0
Forma decimal	5	1
Digito i	4	2

Paso 2: Base 10 → Base 9

$$
\begin{array}{l|ll}
22 & 9 & \\
\cline{2-2}
4 & 2 & \multicolumn{1}{|l}{9} \\
\cline{3-3}
 & 2 & 0
\end{array}
$$

$N_9 = 24_9$

$42_5 = 24_9$

Conversión entre sistemas numéricos

UNIVERSIDAD DE ANTIOQUIA

Ejercicio

Exprese los números del 0 al 15 en base 2 y base 8.

$$\begin{array}{r|l} 10 & 8 \\ \hline 2 & 1 \;|\; 8 \\ & 1 \quad 0 \end{array}$$

$N_8 = 12_8$

$$\begin{array}{l} 10 \,|\, 2 \\ 0 \quad 5 \,|\, 2 \\ \quad 1 \quad 2 \,|\, 2 \\ \quad\quad 0 \quad 1 \,|\, 2 \\ \quad\quad\quad 1 \quad 0 \end{array}$$

$N_2 = 1010_2$

$10_{10} = 12_8 = 1010_2$

$$\begin{array}{r|l} 5 & 8 \\ \hline 5 & 0 \end{array}$$

$N_8 = 5_8$

$$\begin{array}{l} 5 \,|\, 2 \\ 1 \quad 2 \,|\, 2 \\ \quad 0 \quad 1 \,|\, 2 \\ \quad\quad 1 \quad 0 \end{array}$$

$N_2 = 101_2$

Hexadecimal	Decimal	Octal	Binario
0	0	0	000
1	1	1	001
2	2	2	010
3	3	3	011
4	4	4	100
5	5	5	101
6	6	6	110
7	7	7	111
8	8	10	1000
9	9	11	1001
A	10	12	1010
B	11	13	1011
C	12	14	1100
D	13	15	1101
E	14	16	1110
F	15	17	1111

Conversión entre sistemas numéricos

UNIVERSIDAD DE ANTIOQUIA

Atajo directo (Entre binario, octal y hexadecimal)

- Este método es mucho más rápido porque estas bases son potencias de 2.
- La clave es la **agrupación de bits**.
    - **Octal (Base 8)**: $2^3 = 8$. Se trabaja en grupos de **3 bits**.
    - **Hexadecimal (Base 16)**: $2^4 = 16$. Se trabaja en grupos de **4 bits**.

Octal	Binario
0	0000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Hexadecimal	Binario
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

Conversión entre sistemas numéricos

Binario $\longrightarrow$ Octal/Hexadecimal

Se agrupa los bits desde la derecha y sustituye cada grupo por su dígito correspondiente.

Ejemplo: Convertir 11110101101_2 a octal y hexadecimal

Conversión de binario a octal: Base 2 → Base 8

$$\underbrace{011}_{3}\underbrace{110}_{6}\underbrace{101}_{5}\underbrace{101}_{5}$$

$$11110101101_2 = 3655_8$$

Conversión de binario a hexadecimal: Base 2 → Base 16

$$\underbrace{0111}_{7}\underbrace{1010}_{A}\underbrace{1101}_{D}$$

$$11110101101_2 = 7AD_{16}$$

Conversión entre sistemas numéricos

UNIVERSIDAD DE ANTIOQUIA

Octal/Hexadecimal → Binario

Se sustituye cada dígito por su grupo de bits correspondiente

- **Ejemplo 1**: Convertir $9F2_{16}$ a binario

  Conversión de hexadecimal a binario: Base 16 → Base 2

$$\underbrace{1001}^{9}\ \underbrace{1111}^{F}\ \underbrace{0010}^{2}$$

$$9F2_{16} = 100111110010_2$$

- **Ejemplo 2**: Convertir 170_8 a binario

  Conversión de octal a binario: Base 8 → Base 2

$$\overbrace{001}^{1}\ \overbrace{111}^{7}\ \overbrace{000}^{1}$$

$$170_8 = 001111000_2$$

Agenda

- Contextualización
- Representación de la información
- Sistema numéricos posicionales
- Conversión entre sistemas numéricos
- **Suma binaria**

Suma binaria

Representación de funciones booleanas

- Los métodos de cálculo de aritmética binaria son análogos a los de aritmética decimal. En aritmética binaria el número 2 (10_2 en notación binaria) desempeña un papel similar al del número 10 en aritmética decimal.
- **Ejemplo**: Sume 1001_2 y 111_2 usando notación binaria

Ya que $2_{10} = 10_2$ y $1_{10} = 1_2$, la traducción de $1_{10} + 1_{10} = 2_{10}$ en notación binaria es 10_2

$$\begin{array}{r} 1_2 \\ +\ \ 1_2 \\ \hline 10_2 \end{array}$$

De lo que se deduce que la suma de dos 1 juntos, da como resultado llevar un 1 cuando se usa la notación binaria. Sumar tres 1 juntos, también da como resultado en llevar un 1 ya que $3_{10} = 11_2$

$$\begin{array}{r} 1_2 \\ +\ \ 1_2 \\ +\ \ 1_2 \\ \hline 11_2 \end{array}$$

Así, la suma se puede realizar de la siguiente manera:

$$\begin{array}{cccccl} & 1 & 1 & 1 & & \leftarrow \text{carry row} \\ & 1 & 1 & 0 & 1_2 & \\ + & & 1 & 1 & 1_2 & \\ \hline 1 & 0 & 1 & 0 & 0_2 & \end{array}$$

Suma binaria

Circuitos para el cálculo de sumas

Considere el tema de diseñar un circuito para generar la suma de dos dígitos binarios P y Q. Tanto P como Q puede ser ya sea 0 o 1. Y se conocen los siguientes hechos:

$$
\begin{aligned}
1_2 + 1_2 &= 10_2,\\
1_2 + 0_2 = 1_2 &= 01_2,\\
0_2 + 1_2 = 1_2 &= 01_2,\\
0_2 + 0_2 = 0_2 &= 00_2.
\end{aligned}
$$

De lo que se deduce que el circuito a diseñar debe tener dos salidas —una para el digito binario de la izquierda (este se llama **carry**) y uno para el digito binario de la derecha (este se llama **sum**).

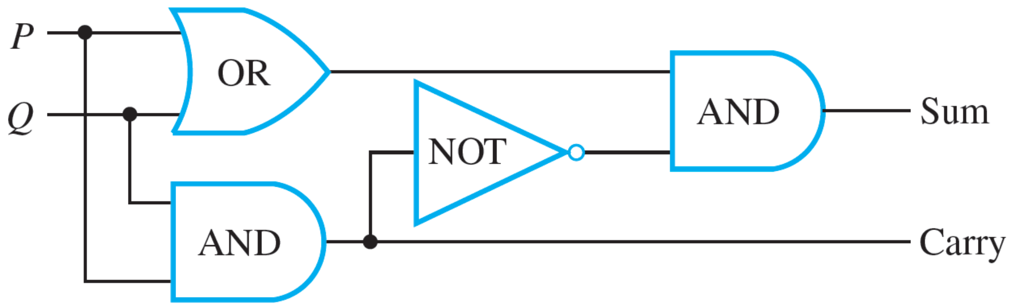
Suma binaria

Circuitos para el cálculo de sumas

- La salida de lo que se lleva es 1 si P y Q son 1; es 0 de otra manera. Así, lo que se lleva (**carry**) se puede producir usando el circuito de puerta **AND** que corresponde a la expresión booleana $P \wedge Q$.
- La salida de la suma es 1 si ya sea P o Q, pero no ambas, es 1. La suma puede, por tanto, producirse usando un circuito que corresponde a la expresión booleana para **or exclusivo** $P \oplus Q$ (que emplea la compuerta **XOR**)

HALF-ADDER

Circuit



Input/OutputTable

P	*Q*	Carry	Sum
1	1	1	0
1	0	0	1
0	1	0	1
0	0	0	0

Circuito para sumar *P* + *Q*, donde *P* and *Q* son digitos binarios

Suma binaria

Circuitos para el cálculo de sumas

UNIVERSIDAD DE ANTIOQUIA

- Ahora consideremos el problema de cómo construir un circuito para sumar dos números enteros binarios, cada uno con más de un dígito. Ya que la adición de dos dígitos binarios puede dar como resultado llevar a la siguiente columna a la izquierda, puede ser necesario añadir tres binarios en ciertos puntos.
- En el ejemplo siguiente, la suma en la columna de la derecha es la suma de dos dígitos binarios, y, debido a lo que se lleva, la suma en la columna de la izquierda es la suma de los tres dígitos binarios.

$$
\begin{array}{rrrl}
 & 1 & & \leftarrow \text{carry row} \\
 & 1 & 1_2 & \\
+ & 1 & 1_2 & \\
\hline
1 & 1 & 0_2 &
\end{array}
$$

Suma binaria

Circuitos para el cálculo de sumas

- Con el fin de construir un circuito que sume varios números dígitos binarios, es necesario incorporar un circuito que calcule la suma de tres dígitos binarios. Tal circuito se llama un **sumador completo**.
- Considere una suma general de tres dígitos binarios P, Q y R que da como resultado en llevar C (o el dígito en el extremo izquierdo) y una suma S (el dígito en el extremo derecho).

$$\begin{array}{rrrl} & 1 & & \leftarrow \text{carry row} \\ & 1 & 1_2 & \\ + & 1 & 1_2 & \\ \hline 1 & 1 & 0_2 & \end{array} \qquad\qquad \begin{array}{rr} & P \\ + & Q \\ + & R \\ \hline & CS \end{array}$$

Suma binaria

Circuitos para el cálculo de sumas

- El funcionamiento del sumador completo se basa en el hecho de que la suma es una operación binaria: Sólo se pueden agregar dos números a la vez. Por tanto P es el primero agregado a Q y después el resultado se suma a R. Por ejemplo, considere la siguiente suma:

$$
\begin{array}{r}
\left.\begin{array}{r} 1_2 \\ +\ \ 0_2 \end{array}\right\} 1_2 + 0_2 = 01_2 \\
+\ \ 1_2 \qquad\qquad\qquad\quad \\
\hline
10_2 \qquad\qquad\qquad\quad
\end{array}
\left.\vphantom{\begin{array}{c}1\\1\\1\end{array}}\right\} 1_2 + 1_2 = 10_2
$$

Suma binaria

UNIVERSIDAD DE ANTIOQUIA

Circuitos para el cálculo de sumas

El proceso que se muestra aquí se puede dividir en pasos que utilizan circuitos de semisumador:

- **Paso 1**: Sume P y Q utilizando un semisumador para obtener un número binario de dos Dígitos.

$$\begin{array}{r} P \\ +\,Q \\ \hline C_1S_1 \end{array}$$

- **Paso 2**: Sume R a la suma C_1S_1 de P y Q.

$$\begin{array}{r} C_1S_1 \\ +\,R \\ \hline \end{array}$$

Para esto, proceda como se muestra a continuación

- **Paso 2a**: Sume R a S_1 utilizando un semisumador para obtener el número de dos dígitos C_2S.

$$\begin{array}{r} S_1 \\ +\,R \\ \hline C_2S \end{array}$$

Entonces S es el dígito del extremo derecho de la suma total de P, Q y R.

- **Paso 2b**: Determine el dígito del extremo izquierdo, C, de la suma total de la siguiente manera: En primer lugar observe que es imposible que tanto C_1 como C_2 sean 1. Si $C_1 = 1$, entonces P y Q son 1 y así $S_1 = 1$. En consecuencia, la suma de S_1 y R da un número binario C_2S_1 donde $C_2 = 0$. Después observamos que C será un 1 en el caso de que la suma de P y Q da como resultado llevar un 1 o en el caso de que la suma de S_1 (el dígito del extremo derecho de $P + Q$) y R da como resultado llevar 1. En otras palabras, $C = 1$ si y sólo si, $C_1 = 1$ o $C_2 = 1$.

El circuito resultante suma entonces tres dígitos binarios.

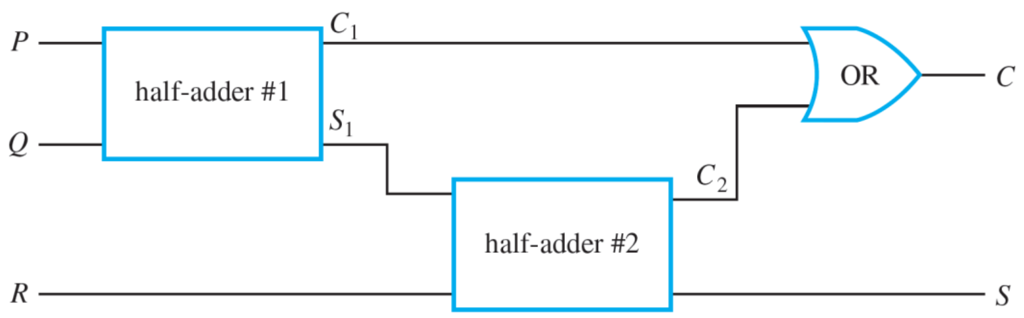
Suma binaria

Circuitos para el cálculo de sumas

El circuito mostrado a continuación permite sumar tres dígitos binarios. Este se conoce como sumador completo.

FULL-ADDER

Circuit



Input/Output Table

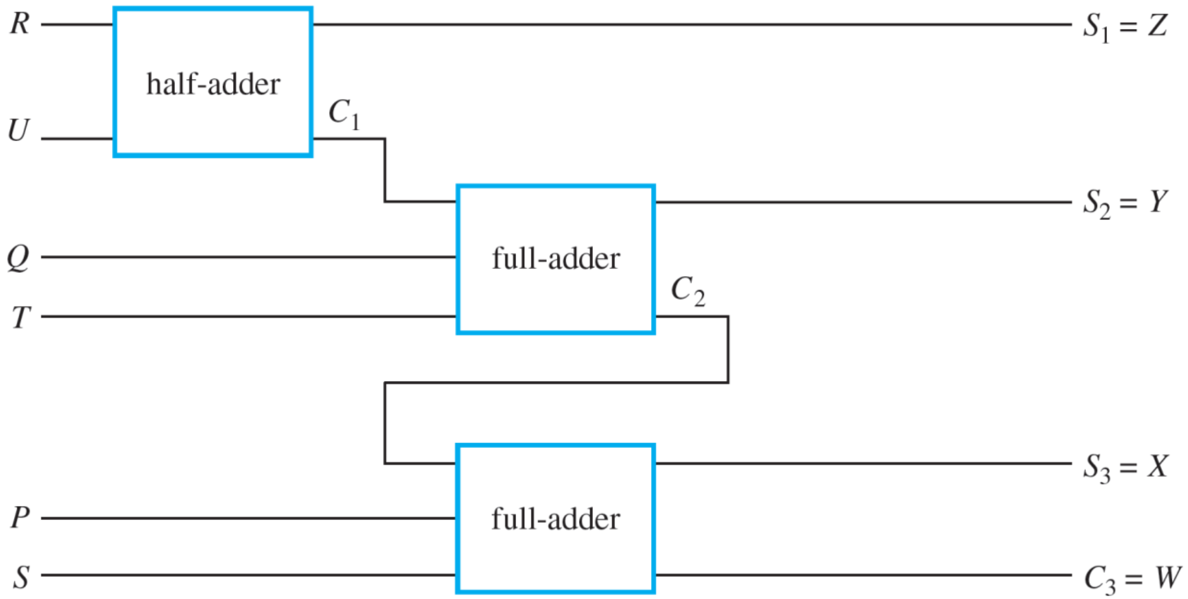
P	***Q***	***R***	***C***	***S***
1	1	1	1	1
1	1	0	1	0
1	0	1	1	0
1	0	0	0	1
0	1	1	1	0
0	1	0	0	1
0	0	1	0	1
0	0	0	0	0

Circuito para sumar $P + Q + R$, donde P, Q y R son dígitos binarios

Suma binaria

Forma Normal Disyuntiva (FND) – Suma de productos (SOP)

- Dos sumadores completos y un semisumador se pueden utilizar juntos para construir un circuito que va a sumar dos números binarios de tres dígitos PQR y STU para obtener la suma WXYZ.
- Tal circuito se llama un sumador en paralelo. Los sumadores en paralelo pueden construirse para sumar números binarios de cualquier longitud finita.



UNIVERSIDAD
DE ANTIOQUIA

Curso de Matemáticas Discretas 1

Clase 13 – Sistemas Numéricos