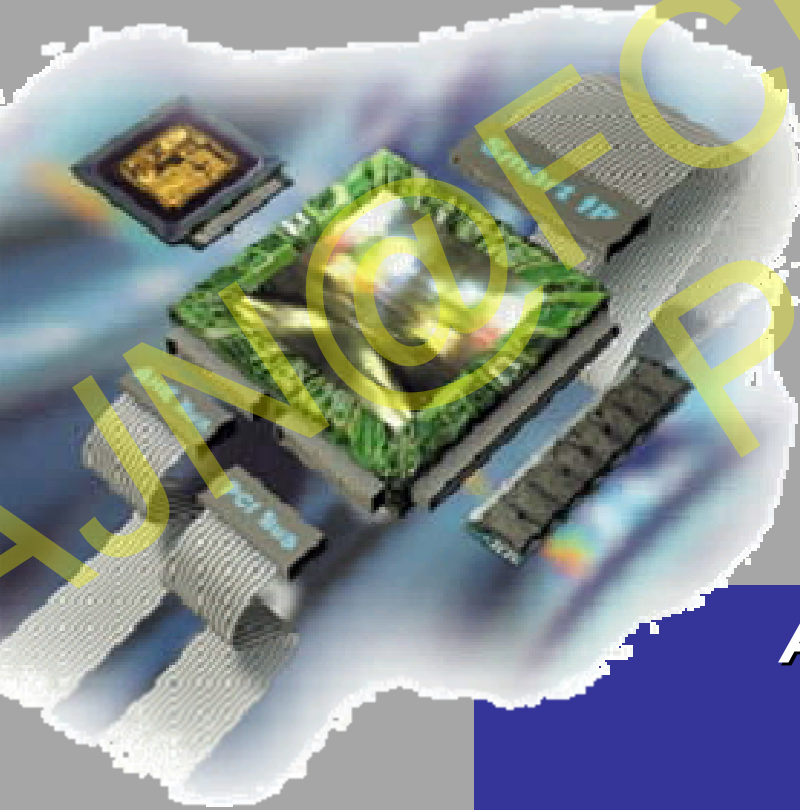


*Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Electrónica
Área de Sistemas Digitales*

Requerimientos en la Organización de Computadoras



***Aurelio Jacinto Nolasco
OArqComp P12***

ajacinto@ece.buap.mx

Repaso

- Sistemas Numéricos.
- Elementos de Hardware.
- Representación de la Información.
- Manejo de la Información.
- Software de diseño y/o simulación

Sistemas Numéricos

- Conjunto de reglas y símbolos que representan cantidades

$$V(A) = \sum_{i=-m}^n \alpha_i \beta^i$$

□ $V(A)$: valor del conjunto de símbolos

□ α_i : i -ésimo símbolo del conjunto

□ β : base numérica

□ Ej: $165.2 = \sum_{i=-1}^2 a_i 10^i = 2 * 10^{-1} + 5 * 10^0 + 6 * 10^1 + 1 * 10^2$

Sistemas Numéricos

- **Bases Numéricas:**

- Binaria: 0, 1.

- Ej: 001b, 1101)₂

- Octal: 0, 1, 2, 3, 4, 5, 6, 7.

- Ej: 1)₈, 15)₈

- Decimal: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

- Ej: 1)₁₀, 13

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

- Ej: 1h, 0xD, FF)_h

Sistemas Numéricos

- Números tienen valor simbólico y posicional:

$$\begin{aligned} 113)_{10} &= 1*10^2 + 1*10^1 + 3*10^0 \\ &= 1*100 + 1*10 + 3*1 \end{aligned}$$

$$\begin{aligned} 111\ 0001)_2 &= 1*2^6 + 1*2^5 + 1*2^4 + 0*2^3 + 0*2^2 + 0*2^1 + 1*2^0 \\ &= 1*64 + 1*32 + 1*16 + 0*8 + 0*4 + 0*2 + 1*1 \end{aligned}$$

$$\begin{aligned} 161)_8 &= 1*8^2 + 6*8^1 + 1*8^0 \\ &= 1*64 + 6*8 + 1*1 \end{aligned}$$

$$\begin{aligned} 71)_h &= 7*16^1 + 1*16^0 \\ &= 7*16 + 1*1 \end{aligned}$$

Elementos de Hardware

- 1 alambre o conductor, representa un nodo de conexión
- Bus: conjunto de alambres para llevar información en modo paralelo

1 alambre

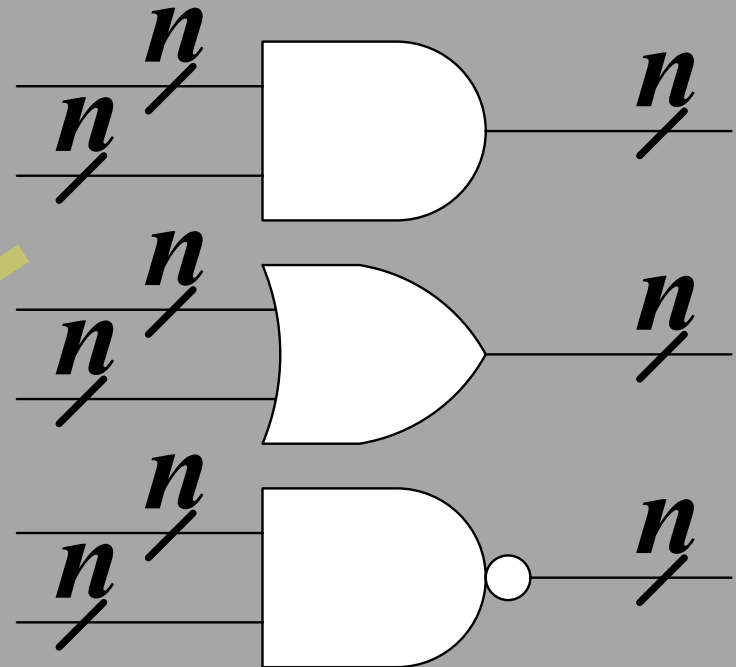


n alambres = 1 Bus



Elementos de Hardware

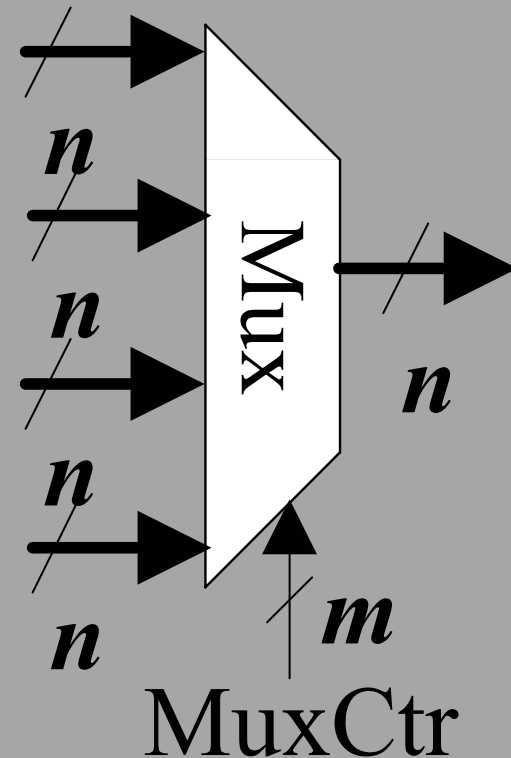
- **AND, OR, NAND, NOT:** compuertas lógicas básicas
 - compuertas con m entradas, de n bits c/u (*Buses*).
 - salida, de n bits.
 - **sin** líneas de control.
 - **Función:** realizar la operación lógica respectiva, bit a bit, con los datos de entrada.



Elementos de Hardware

- **MULTIPLEXORES**

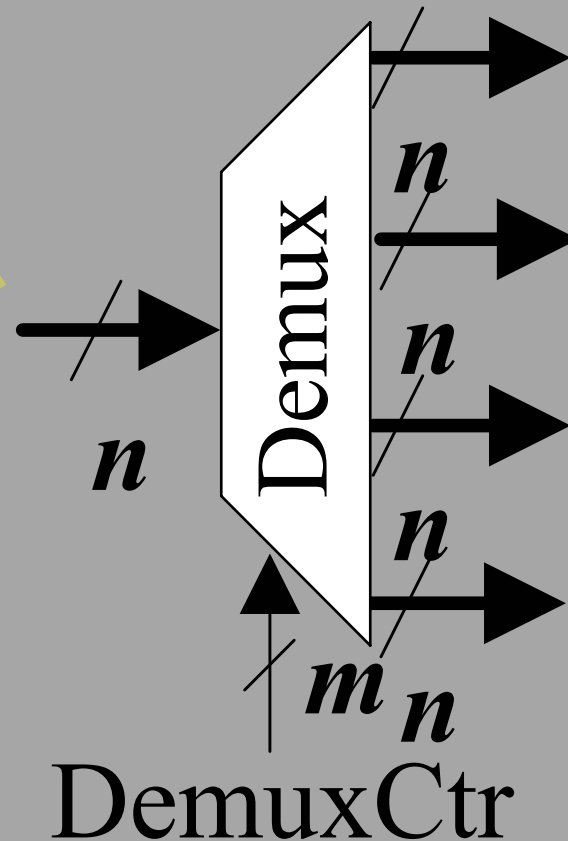
- 2^m puertos de entrada, de n bits c/u.
- salida, de n bits.
- m líneas de control.
- **Función:** seleccionar, usando las líneas de control, una de las posibles entradas y hacer que los datos aparezcan en la salida (**buses**).



Elementos de Hardware

• DEMULTIPLEXORES

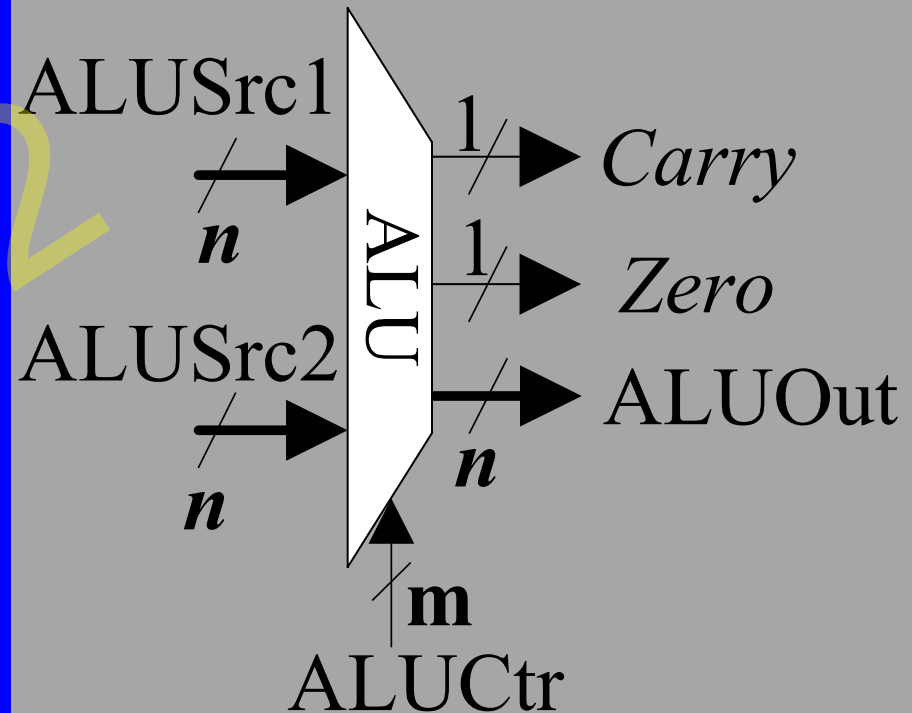
- entradas, de n bits.
- 2^m compuertas de salida, de n bits c/u.
- m líneas de control.
- **Función:** hace que las entradas aparezcan en las salidas seleccionada por las líneas de control.



Elementos de Hardware

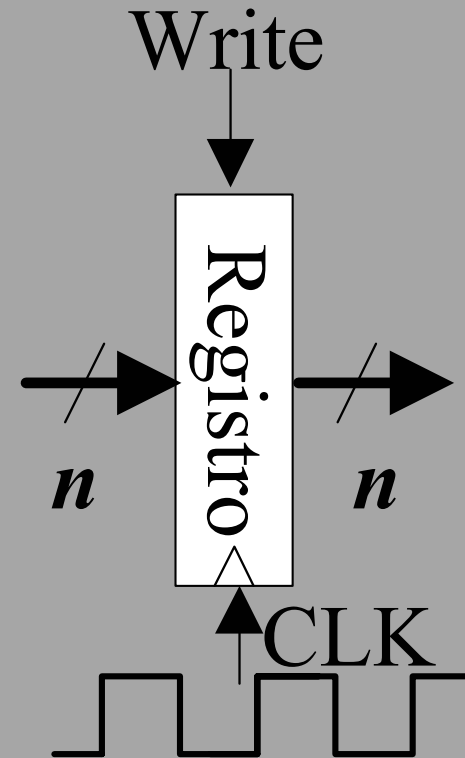
- **ALUs**

- **2** buses de entrada, de n bits (ALUSrcX) (operandos)
- **1** bus de salida (ALUOut), de n bits
- **1** bus de control (ALUCtr) de m bits
- **Función:** realizar la operación especificada en ALUCtr sobre los operandos. Ejecuta 2^m operaciones distintas.



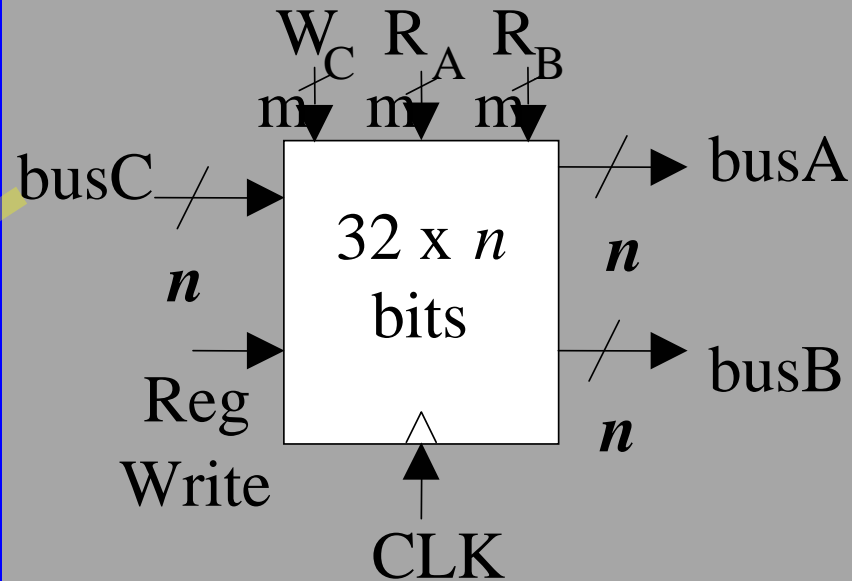
Elementos de Hardware

- **Relojes** (Clock, CLK)
 - Proveen sincronismo.
 - Oscila periódicamente.
- **Registros:** elementos de almacenamiento de n bits
 - 1 bus de entrada, de n bits
 - 1 bus de salida, de n bits
 - 1 línea de control para habilitar la escritura.
 - 1 línea de entrada de reloj (síncronos)



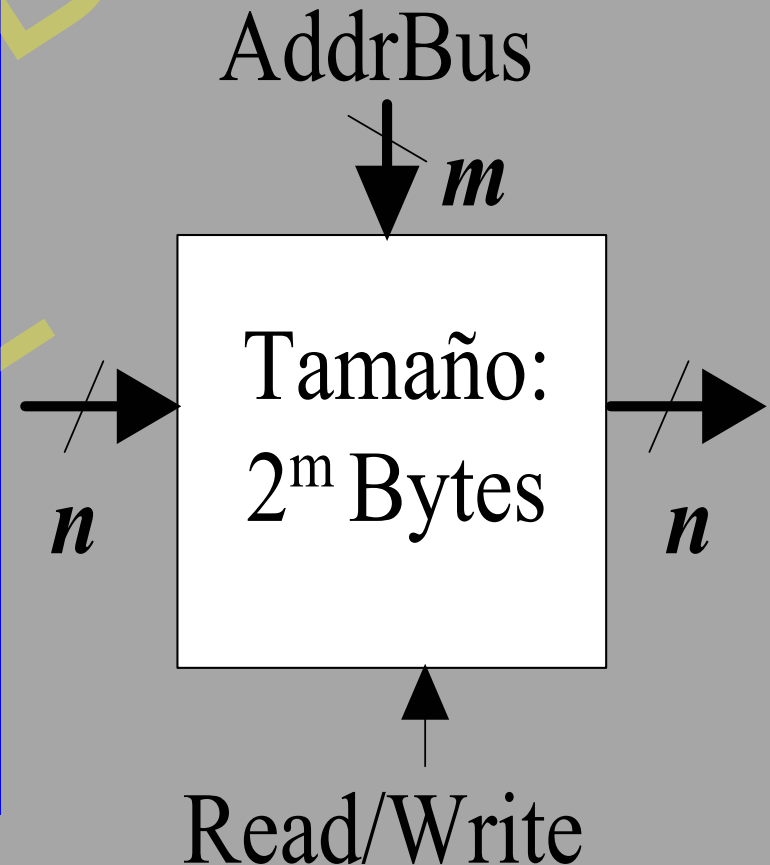
Elementos de Hardware

- **Banco de registros:** agrupación de 2^m registros, de n bits c/u.
 - 1 bus de entrada, de n bits.
 - 1 bus de control, de m bits, para direccionar registro a escribir.
 - y buses de salida, de n bits c/u.
 - y buses de control, de m bits c/u, para direccionar registros a leer.
 - 1 bus de control para habilitar las escrituras.
 - 1 línea de entrada de reloj.



Elementos de Hardware

- **Memoria:** elemento de almacenamiento de información, que es capaz de almacenar 2^m Bytes.
 - 1 bus de entrada/salida (datos), de n bits
 - 1 bus de direcciones, de m bits, para direccionar la posición de memoria a leer/escribir
 - 1 línea de control para habilitar las lecturas/escrituras



Representación de la Información

- **Computadoras (Digitales):**
 - lógica binaria para funcionar.
 - representación binaria para los datos.
- **Tanto las Instrucciones, como los datos, deben ser representados en forma unívoca.**
 - Esto evita múltiples interpretaciones.
 - Permite también generar múltiples representaciones
 - requiere un cierto conocimiento previo de la situación (protocolo)
- **Necesidad de representar múltiples tipos de datos: señales, números, objetos, etc.**

Representación de la Información

- **Bit (b):** 1 dígito binario. Unidad mínima de información.
 - Ej: 0; 1
- **Nibble:** 4 dígitos binarios
 - Ej: 0000; 0110
- **Byte (By):** 8 dígitos binarios. Unidad mínima de almacenamiento.
 - Ej: 0000 0000; 0000 0001; 0010 1110
- **Word (wd):** n Bytes, (depende del valor de $n=1, 2, 4, 8 \dots$)
 - Ej: En Intel $n = 2$, es decir, $1 \text{ Word} \leftrightarrow 2 \text{ By} \leftrightarrow 16 \text{ b}$

Representación de la Información

- **Números enteros sin signo**

- Representación binaria directa
- La cantidad de bits determina los límites
 - Ej: 8 bits: 0 – 255; $[0, 2^8-1]$
 - Ej: 16 bits: 0 – 65535; $[0, 2^{16}-1]$
 - Ej: 32 bits: 0 – 4294967295; $[0, 2^{32}-1]$

- **Números enteros con signo (opción 1)**

- Representación binaria modificada: bit Mas Significativo el Signo
- 0 representa positivos, 1 negativos
 - Ej: 8 bits: -128 .. 127; $[-2^7, 2^7-1]$
 - Ej: 16 bits: -32768 .. 32767; $[-2^{15}, 2^{15}-1]$
 - Ej: 32 bits: -2147483648 .. 2147483647; $[-2^{31}, 2^{31}-1]$

Representación de la Información

- **Números enteros con signo** (opción 2)
 - Los números negativos se representan en complemento a 2
 - Ej: 0000 0001 \Rightarrow +1; 1111 1111 \Rightarrow -1
 - Conceptos de extensión lógica y aritmética.
- **Números reales**
 - Notación en punto flotante: representación IEEE 754
 - $(-1)^S F 2^E$
 - **S**: bit de signo, **F**: bits de mantisa, **E**: bits de exponente
 - Números con precisión simple 32 bits (**F**=23, **E**=8)
 - Números con precisión doble 64 bits (**F**=52, **E**=11)
 - Ej: $125 = 1111101_2 = (-1)^0 0.1111101_2 10_2^{111}_2$
 - | | | |
|---|------------|-------------------------------|
| 0 | 000 0011 1 | 0000 0000 0000 0000 0111 1101 |
|---|------------|-------------------------------|

 \Leftrightarrow 0x 0380007D
 - S

E

F

Representación de la Información

- **ASCII:**

- Permite representar caracteres numéricos y alfanuméricos
- Utiliza 7 bits

- Ej: A => ASCII 65 a => ASCII 97
 @ => ASCII 64 \ => ASCII 92
 3 => ASCII 51

- **Código BCD:**

- Representa dígitos y números usando 1 nibble por dígito

- Ej: $0_{10} \Rightarrow 0000$ $7_{10} \Rightarrow 0111$
 $15_{10} \Rightarrow 0001\ 1001$

Representación de la Información

- **b**: unidad mínima, ordenados de MSB a mSB.
 - Los datos pueden tomar más de un byte para su representación.
 - Ej: $0000\ 0001\ 0000\ 0010)_2 \leftrightarrow 258)_{10}$
- **By**: unidad mínima de almacenamiento ¿cómo los ordeno?
 - “**Little Endian**”: Byte menos significativo en parte más baja (de Memoria)
 - $0000\ 0001\ 0000\ 0010$ (Arquitectura Intel x86)
 - “**Big Endian**”: Byte menos significativo en parte más alta (de Memoria)
 - $0000\ 0010\ 0000\ 0001$ (Arquitectura SUNSparc)

Representación de la Información

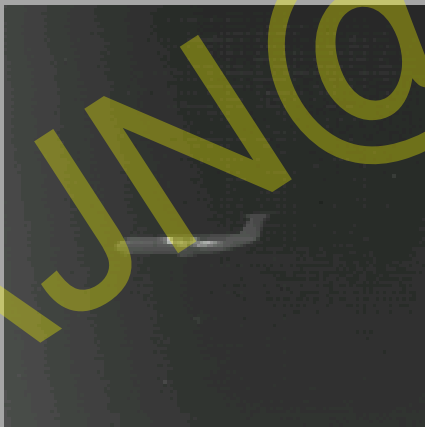
Big Endian

000000000h:	48	27	47	28
000000010h:	42	DA	42	6C
000000020h:	3D	E8	3D	89
000000030h:	39	DB	39	88
000000040h:	35	FA	35	F5

Wd 15^a

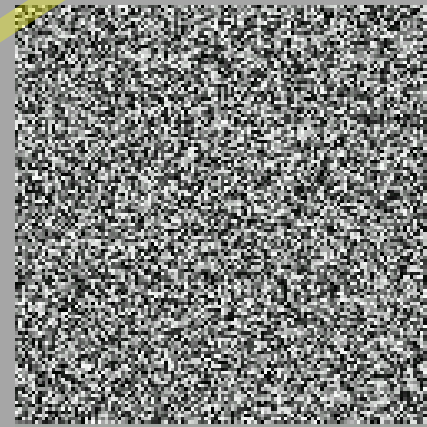
Little Endian

000000000h:	27	48	28	47
000000010h:	DA	42	6C	42
000000020h:	E8	3D	89	3D
000000030h:	DB	39	88	39
000000040h:	FA	35	F5	35



Si se lee como
Big Endian

Si se lee como
Little Endian



Si se lee como
Big Endian

Si se lee como
Little Endian

Manejo de la Información

- Las computadoras, trabajan con datos digitalizados
 - Representaciones binarias numéricas y alfanuméricas.
- El trabajar con datos, implica orden, el orden implica control.
- En la información existen **datos** y **control**.
 - Analogías:
 - **Tráfico vehicular**: automóviles (datos) y señales de tránsito (control)
 - **Preparación de un trago**: licores (datos) y receta (control)
 - **Programa en C**: valores de variables (datos) y código fuente (control)
 - **Computador (Hw)**: memoria (datos) y CPU (control)
 - **CPU**: camino de datos y sección de control

Manejo de la Información

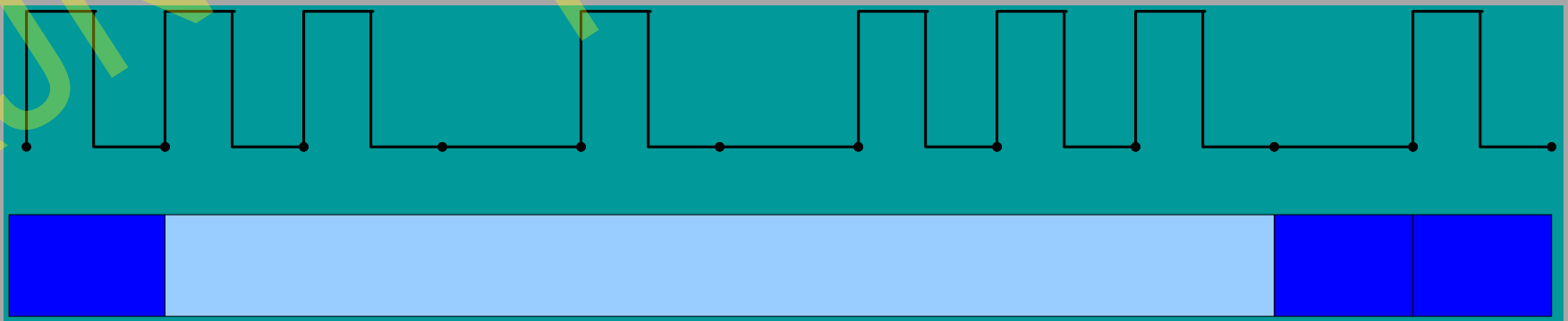
- Al existir múltiples elementos posibles, se hace necesaria la identificación de la información.
 - Esto permitirá realizar búsqueda inequívocas.
- Identificación genera direccionamiento
 - nombre, número, etc. (en general ID)

Analogías:

- **Lockers**: objetos guardados (datos) y llave numerada (dirección)
- **Programa en C**: valores de variables (datos) y nombre de la variable (dirección)
- **Computador (Hw)**: programa en memoria (datos) y ubicación dentro de la memoria (dirección)

Manejo de la Información

- Codificación: organización para la representación unívoca de la información
 - Ej. en datos: código Morse, ASCII, etc.
- Información de control debe también codificarse
 - Ej. caracteres válidos en variables, sentencias de control, protocolos de comunicación, instrucciones CPU



Nomenclatura y Terminología

- **b:** bit
- **BW:** ancho de banda
- **By:** byte
- **CU:** unidad de control
- **DP:** camino de datos
- **Fw:** firmware
- **FU:** unidad funcional
- **Hw:** hardware

Nomenclatura y Terminología

- **I/O**: entrada/salida de datos
- **mSB**: bit menos significativo
- **MSB**: bit más significativo
- **R**: lectura de datos
- **Rx**: recepción de datos
- **Sw**: software
- **SC**: sistema computacional
- **Tx**: transmisión de datos
- **W**: escritura de datos

Nomenclatura y Terminología

- **Dirección:** etiqueta de ubicación del dato o instrucción en la unidad de almacenamiento
- **Direccionar:** dar la dirección de cuál será el dato a leer o escribir de la unidad de almacenamiento
- **Procesar:** modificar con algún fin los datos

Gracias