1 Evolución del Procesamiento de Datos

Organización y Arquitectura de una Computadora

- **Computadora** → dispositivo electrónico diseñado para aceptar datos de entrada, realizar operaciones sobre ellos y elaborar resultados (que se puedan obtener como salidas).
- **Programa** → representación de un algoritmo en un lenguaje de programación.
- Componentes de una computadora
  - **Memoria principal** → almacena
    - datos de entrada.
    - todas las instrucciones del programa.
    - resultados.
  - **Registros de la CPU** → almacena
    - la instrucción en ejecución.
    - los datos con los que va a operar.
  - **Unidad de control** → interpreta y ejecuta instrucciones.
  - **Unidades de cálculo** → realiza operaciones aritméticas y lógicas.
  - **Buses de interconexión** → transfieren
    - datos.
    - instrucciones.
    - resultados.
  - **Dispositivos de entrada** → permiten el ingreso de
    - los programas a memoria principal.
    - los datos de entrada.
  - **Dispositivos de salida** → permiten el egreso de los resultados (desde la memoria principal).
- El **set de instrucciones** permite representar los algoritmos que solucionan los problemas.

Estratificación del Software

APLICACIONES
SOFTWARE PARA PRODUCIR APLICACIONES
SOFTWARE DE GESTIÓN DE RECURSOS
ARQUITECTURA DEL SET DE INSTRUCCIONES
LENGUAJE DE SEÑALES QUE PERMITEN LA EJECUCIÓN DE LAS INSTRUCCIONES
HARDWARE

- Jerarquía más alta → programa de uso particular de los usuarios
- Lenguaje de señales que genera la unidad de control para el gobierno de los distintos dispositivos físicos.
- Jerarquía más baja

- Todos los programas se compilan a **código de máquina** (lenguaje que interpreta el CPU y pertenece al nivel de arquitectura del set de instrucciones)
- La **arquitectura del set de instrucciones** determina
  - el formato de las instrucciones.
  - los tipos de dato que puede operar.
  - las distintas formas de obtener datos de memoria
  - la forma en que se atienden eventos externos.
- Cada instrucción implica "algo que hacer", un verbo que se denomina **código de operación**.
- La forma en que se implementan los códigos de operación se denomina **nivel de microarquitectura**.
- La **microarquitectura** determina la forma en que se ejecuta la instrucción.

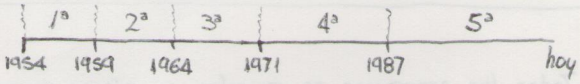
Evolución del Procesamiento de Datos

- Es probable que la primera máquina digital que utilizó el hombre haya sido ÁBACO.
- Blaise Pascal construye la primera calculadora digital, basada en ruedas dentadas.
- Von Leibnitz inventó el sistema de numeración binario
- Charles Babbage, empezó a construir la primera computadora digital, incorporando una rutina de operaciones en tarjetas perforadas. No pudo concluirla por falta de tecnología apropiada.
- George Boole desarrolló un álgebra que no usa números, sino que establece la relación entre conceptos lógicos.
- Herman Hollerith inventó una técnica para procesar gran cantidad de datos por medio de tarjetas perforadas, para luego clasificar y analizar los datos perforados en ellas.
- Howard Aiken puso en operación una computadora secuencial automática (de IBM) que constaba de partes electromecánicas y estaba controlada por una cinta de papel perforada.
- John Von Neumann creó una máquina de programa almacenado a la que se denominó computadora. Esta máquina de propósito general se fundamenta en 3 principios
  - trabajo en binario.
  - programa almacenado en memoria.
  - posibilidad de provocar una ruptura de secuencia de instrucciones en un programa.

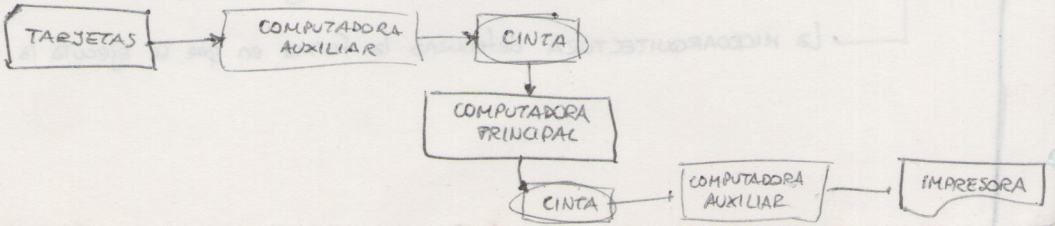
Clasificación de las Computadoras

- **Analógicas**
  - se utilizaron para estudiar un modelo semejante/análogo a una ecuación.
  - hoy, en día, se utilizan para procesos en los que sea importante representar la variación en el tiempo de magnitudes continuas.
  - Desventajas
    - la falta de exactitud en la salida.
    - el modelo construido, que representaba a una aplicación y no servía para otra.
  - Ventajas
    - su capacidad de procesar datos no discretos.
    - genera disminuciones de los costos de desarrollo.
- **Digitales** — procesan dígitos binarios (0s y 1s), que representan DATOS.
  - Ventajas
    - efectúan cálculos precisos.
    - el modelo se arma sobre un programa intercambiable.
- **Híbridas**
  - son una combinación de los sistemas analógicos y digitales.
  - mientras la parte analógica toma los datos continuos, la parte digital efectúa los cálculos.

Generaciones de Computadoras Digitales



- **1ra**
  - las computadoras estaban constituidas por válvulas de vacío, que
    - disipaban gran cantidad de calor.
    - ocupaban una superficie muy amplia.
  - las tareas se ejecutaban en forma secuencial → el programa (almacenado en tarjetas/cintas perforadas) se cargaba en memoria principal → se ejecutaba (instrucción por instrucción) → se imprimían los resultados.
- **2da**
  - las computadoras estaban constituidas por transistores y circuitos impresos ⇒ reducción de tamaño.
  - PROCESAMIENTO POR LOTES → se copiaba la información del soporte tarjeta a soporte cinta magnética y de ésta a impresora mediante un procesador auxiliar



- 3ra
  - Con la tecnología de los circuitos integrados, se incrementó la velocidad y se redujo la energía.
  - Aparece la MULTIPROGRAMACIÓN → método que consiste en que varios programas residan en forma simultánea en la memoria en estado de ejecución.
    - no es lo mismo que MULTIPROCESAMIENTO → cuando se procesan varios programas simultáneamente en distintas CPUs relacionadas entre sí.
  - Se desarrollan los primeros sistemas interactivos, permitiendo que el usuario intervenga en la actividad del procesamiento.
  - Se desarrolla el concepto de MÁQUINA VIRTUAL, que simplifica la labor del programador.
- 4ta
  - Una CPU completa incluida en un solo chip (MICROPROCESADOR).
  - Acceso a grandes bancos de datos.
  - Automatización del prácticamente el total de las tareas de oficina.
  - Equipos
    - más pequeños.
    - menos costosos.
    - más rápidos.
- 5ta
  - Todavía en proceso, se resalta la necesidad de una interfaz hombre/máquina de elevada inteligencia.
  - Los procesadores ejecutan varias etapas de las instrucciones en paralelo.
  - Características de estas máquinas
    - integración de microcircuitos en muy alta escala.
    - nuevos métodos de procesamiento de datos.
    - nuevas arquitecturas.
    - nuevos avances en el diseño de memorias con mayor capacidad / velocidad
  - El objetivo del programa de 5ta generación supone que las computadoras tengan capacidades cognitivas superiores a la humana.

PROCESAMIENTO DE DATOS Y SISTEMAS DE INFORMACIÓN

- DATOS → conjuntos de símbolos que representan un objeto concreto o abstracto.
- INFORMACIÓN → consecuencia de procesar los datos para que tengan un significado.
- SISTEMA DE INFORMACIÓN
  - se refiere al conjunto coordinado de elementos-datos-procesos cuya interacción permite la obtención de la información.
  - en él, deben coordinarse y cooperar entre sí los procesos y datos del sistema y del usuario, para la elaboración de la información.
- PROCESAR DATOS significa realizar una operación sobre ellos. INPUT → PROCESO → OUTPUT
- Los DATOS DE ENTRADA son estructuras binarias organizadas (CÓDIGO).
- Los DATOS DE SALIDA también son información codificada en binario.
- DECODIFICACIÓN → proceso en el que se convierte código binario a letras-números-puntos-comas-etc...

SISTEMAS SINCRÓNICOS DE PROPÓSITO GENERAL

- SISTEMAS DE PROPÓSITO ESPECÍFICO → la tarea que debe realizar la computadora es fija (no cambia).
- SISTEMAS DE PROPÓSITO GENERAL → la tarea asignada a la computadora (APLICACIÓN) se puede cambiar y está definida en software.
  - requieren la elaboración de un programa.
  - la tarea de la computadora cambia en tanto cambie el programa.
- Cada instrucción del programa es transferida de la memoria a la CPU.
- La UNIDAD DE CONTROL organiza el tiempo para ejecutar el conjunto de operaciones elementales. → MICROOPERACIONES

ARQUITECTURA DE COMPUTADORAS: LOS PRIMEROS CONCEPTOS

- Una computadora incluye módulos de HARDWARE (lo que constituye la estructura física – lo tangible), SOFTWARE (conjunto de programas – la parte lógica – lo intangible). Otro concepto conocido es el de FIRMWARE, usado para identificar los dispositivos físicos programados.
- En las computadoras digitales, los datos e instrucciones están constituidas por grupos de **BITS**.
- Un BIT (Binary digiT) es la unidad mínima de información.
- BYTE → combinación de 8 bits.
- NIBBLE → medio byte → 4 bits.

  } un byte está constituido por 2 nibbles (el de orden superior y el de orden inferior).
- PALABRA DE CPU
  - la unidad de trabajo o procesamiento del CPU.
  - su longitud tiene relación con la longitud de los registros asociados a la CPU.

ARQUITECTURA DE UNA CPU (UNIDAD CENTRAL DE PROCESO)

- La unidad de prebúsqueda obtiene instrucciones de la memoria principal. Estas instrucciones se disponen en una cola. Cada instrucción es interpretada por la unidad de precodificación, que la traduce a microcódigos.
- MICROCÓDIGO → lenguaje de menor nivel que el código de máquina.

  la ejecución se realiza con los soportes de la ALU (Unidad Aritmético-Lógica) y de los registros de trabajo de la CPU.

  La ALU se encarga de operar los datos que recibe la memoria y obtener el resultado.
- REGISTROS DE CPU
  - forman una pequeña memoria local.
  - guardan información temporal para el procesamiento en curso.
- La CPU está asociada a la MEMORIA PRINCIPAL para obtener las instrucciones-datos y almacenar los resultados.

LÓGICA DIGITAL Y COMPONENTES ELECTRÓNICOS

- CHIP → circuito electrónico diseñado sobre una estructura de silicio.
- CIRCUITO INTEGRADO → soporte físico que incluye en su interior conjuntos de compuertas lógicas.
- PALABRA DE PROCESADOR → cantidad de información que puede procesar en un paso, expresada en bits.
- CAPACIDAD DE DIRECCIÓN → se puede determinar por la cantidad de líneas que transfieren los bits de una dirección física.
  - Por ejemplo, con 32 bits se pueden obtener 2^{32} combinaciones distintas que representan 2^{32} direcciones posibles.
  - determina la cantidad de memoria principal a la que se puede acceder.
- La arquitectura del set de instrucciones se puede clasificar en:
  - CISC → permite al programador diseñar programas con pocas instrucciones de alta complejidad.
  - RISC
    - "la mayor parte de los programas utiliza 80% de instrucciones simples y 20% complejas".
    - tienen un set de instrucciones constituido por pocas operaciones simples y rápidas.
  - EPIC → "para ganar velocidad", se ejecutan múltiples instrucciones en paralelo.

EL SISTEMA OPERATIVO, LA DINÁMICA DEL SISTEMA

- SISTEMA OPERATIVO → constituido por "reglas" que coordinan de manera coherente hardware y software.

2 SISTEMAS NUMÉRICOS

SISTEMAS DE NOTACIÓN POSICIONAL

- El concepto es que cada dígito tiene un peso distinto según el lugar que ocupa.
  - → la base, elevada a la posición que ocupa dentro del número.

  la suma de cada dígito multiplicado por su peso permitirá obtener el valor final del número.

- Dada una base B, se tienen B dígitos posibles (de 0 a B-1).

- SISTEMA DECIMAL
  - base 10 → 10 símbolos (0, 1, 2, 3, 4, 5, 6, 7, 8, 9).
  - Ejemplo: $1023_{10} \rightarrow 3 \cdot 10^0 + 2 \cdot 10^1 + \cancel{0 \cdot 10^2} + 1 \cdot 10^3$

    $3 \cdot 1 + 2 \cdot 10 + 0 + 1 \cdot 1000$

    $3 + 20 + 1000 \rightarrow \boxed{1023_{10}}$

- SISTEMA BINARIO
  - base 2 → 2 símbolos (0, 1).
  - Ejemplo: $11011_2 \rightarrow 1 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^3 + 1 \cdot 2^4$

    "uno, uno, cero, uno, uno, en base dos"

    $1 \cdot 1 + 1 \cdot 2 + \cancel{0 \cdot 4} + 1 \cdot 8 + 1 \cdot 16$

    $1 + 2 + \cancel{0} + 8 + 16 \rightarrow \boxed{27_{10}}$

- SISTEMA OCTAL
  - base 8 → 8 símbolos (0, 1, 2, 3, 4, 5, 6, 7).
  - Ejemplo: $275_8 \rightarrow 5 \cdot 8^0 + 7 \cdot 8^1 + 2 \cdot 8^2$

    "dos, siete, cinco, en base 8"

    $5 \cdot 1 + 7 \cdot 8 + 2 \cdot 64$

    $5 + 56 + 128 \rightarrow \boxed{189_{10}}$

- SISTEMA HEXADECIMAL
  - base 16 → 16 símbolos (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F)
  - Ejemplo: $1A2_{16} \rightarrow 2 \cdot 16^0 + 10 \cdot 16^1 + 1 \cdot 16^2$

    "uno, a, dos, en base 16"

    $2 \cdot 1 + 10 \cdot 16 + 1 \cdot 256$

    $2 + 160 + 256 \rightarrow \boxed{418_{10}}$

- Los sistemas octal y hexadecimal permiten compactar bits, de modo de hacer más sencilla la tarea de reconocerlos a simple vista (en octal se agrupan de a 3 bits y en hexadecimal, 4 bits).

- Con n bits se pueden representar valores entre 0 y $2^n - 1$.

MÉTODOS DE CONVERSIÓN DE NÚMEROS ENTEROS Y FRACCIONARIOS.

- DE BINARIO a DECIMAL → Ejemplo:

  $2^4\ 2^3\ 2^2\ 2^1\ 2^0\ \ 2^{-1}\ 2^{-2}\ 2^{-3}$

  $10111,101_2$

  "Se obtendrá el valor decimal multiplicando cada bit por la potencia de 2 correspondiente a su posición (en la parte fraccionaria, el exponente es negativo) y sumando los valores finales."

  $1 \cdot 2^{-3} = 0{,}125$

  $\cancel{0 \cdot 2^{-2}} = 0$

  $1 \cdot 2^{-1} = 0{,}5$

  $+$

  $1 \cdot 2^0 = 1$

  $1 \cdot 2^1 = 2$

  $1 \cdot 2^2 = 4$

  $\cancel{0 \cdot 2^3} = 0$

  $1 \cdot 2^4 = 16$

  $\boxed{23{,}625_{10}}$

De OCTAL a DECIMAL → Ejemplo:

"Se obtendrá el valor decimal multiplicando cada dígito octal por la potencia de 8 que corresponda a su posición y sumando los valores finales."

$$\overset{8^1}{1}\ \overset{8^0}{7}\ ,\ \overset{8^{-1}}{3}_8$$

$$3 \cdot 8^{-1} = 0{,}375$$
$$+\ 7 \cdot 8^{0} = 7$$
$$1 \cdot 8^{1} = 8$$
$$\boxed{15{,}375_{10}}$$

De HEXADECIMAL a DECIMAL → Ejemplo:

"Se obtendrá su valor decimal multiplicando cada dígito hexadecimal por la potencia de 16 que corresponda a su posición y sumando los valores finales"

$$\overset{16^1}{1}\ \overset{16^0}{A_{10}}\ ,\ \overset{16^{-1}}{0}\ \overset{16^{-2}}{F_{15}}{}_{16}$$

$$15 \cdot 16^{-2} = 0{,}05859375$$
$$0 \cdot 16^{-1} = 0$$
$$+\ 10 \cdot 16^{0} = 10$$
$$1 \cdot 16^{1} = 16$$
$$\boxed{26{,}05859375_{10}}$$

De DECIMAL a BINARIO → Ejemplo: $23{,}625_{10}$

PARTE ENTERA — PARTE FRACCIONARIA

divisiones sucesivas

23 | 2 → 1
11 | 2 → 1
5 | 2 → 1
2 | 2 → 0
1 → 1

10111

multiplicaciones sucesivas...

$0{,}625_{10} \cdot 2 = 1{,}25$
$0{,}25 \cdot 2 = 0{,}5$
$0{,}5 \cdot 2 = 1{,}0$

101

$$\boxed{10111{,}101_2}$$

De DECIMAL a OCTAL → Ejemplo: $15{,}375_{10}$

PARTE ENTERA — PARTE FRACCIONARIA

divisiones sucesivas

15 | 8 → 7
1 | 8 → 0
1

17

multiplicaciones sucesivas

$0{,}375 \cdot 8 = 3{,}0$

3

$$\boxed{17{,}3_8}$$

De DECIMAL a HEXADECIMAL → Ejemplo: $26{,}05859375_{10}$

PARTE ENTERA — PARTE FRACCIONARIA

divisiones sucesivas

26 | 16 → A=10
1 | 16 → 0
1

1A

multiplicaciones sucesivas

$0{,}05859375 \cdot 16 = 0{,}9375$
$0{,}9375 \cdot 16 = 15 \rightarrow F$

0F

$$\boxed{1A{,}0F_{16}}$$

Conversión directa de BINARIO a OCTAL-HEXADECIMAL → Ejemplo: $010110{,}100100_2$

"Se divide el número binario en grupos de 3-4 bits, siempre partiendo desde la coma y hacia izquierda (parte entera) y derecha (parte fraccionaria).

Se sustituye cada grupo de 3-4 bits por su correspondiente dígito octal-hexadecimal."

2 6 4 4

$$\boxed{10110{,}100_2 = 26{,}44_8}$$

0001 0110 , 1001 $_2$

1 6 9

$$\boxed{10110{,}100_2 = 16{,}9_{16}}$$

* Para convertir de OCTAL o HEXADECIMAL a BINARIO, se hace el camino inverso.

COMPLEMENTO DE UN NÚMERO

- Permite a muchas computadoras digitales, entre otras cosas, hacer restas mediante sumas.
- COMPLEMENTO A LA BASE (AUTÉNTICO) → se complementan todos los dígitos → se suma 1.
- COMPLEMENTO A LA BASE MENOS 1 (RESTRINGIDO) → se complementan todos los dígitos.
- Ejemplos:
  - BINARIO → Número: 11001_2
    - Complemento Restringido:
      $$\begin{array}{r} 11111 \\ -\ 11001 \\ \hline \boxed{110_2} \end{array}$$
    - Complemento Auténtico:
      $$\begin{array}{r} 11111 \\ -\ 11001 \\ \hline 110_2 \\ +\ 1 \\ \hline \boxed{111_2} \end{array}$$
  - OCTAL → Número: 36_8
    - Complemento Restringido:
      $$\begin{array}{r} 77 \\ -\ 36 \\ \hline \boxed{41_8} \end{array}$$
    - Complemento Auténtico:
      $$\begin{array}{r} 41_8 \\ +\ 1 \\ \hline \boxed{42_8} \end{array}$$
  - DECIMAL → Número: 20_{10}
    - Complemento Restringido:
      $$\begin{array}{r} 99 \\ -\ 20 \\ \hline \boxed{79_{10}} \end{array}$$
    - Complemento Auténtico:
      $$\begin{array}{r} 79_{10} \\ +\ 1 \\ \hline \boxed{80_{10}} \end{array}$$
  - HEXADECIMAL → Número: $B9_{16}$
    - Complemento Restringido:
      $$\begin{array}{r} FF_{16} \\ -\ B9 \\ \hline \boxed{46_{16}} \end{array}$$
    - Complemento Auténtico:
      $$\begin{array}{r} 46_{16} \\ +\ 1 \\ \hline \boxed{47_{16}} \end{array}$$

3 Representación de Datos en la Computadora

Flujo de Datos dentro de una Computadora

- Tanto las instrucciones de un programa como los datos que ingresan en la memoria desde el exterior lo hacen en un código alfanumérico de representación de caracteres (como el ASCII). Las instrucciones son transformadas por un programa de traducción a código de máquina, que es el que entiende el procesador, para luego ejecutarse. Los encargados de la transformación de datos suelen ser rutinas previstas por bibliotecas estándar del lenguaje e incluidas en el programa. Los datos se almacenarán en la memoria de diferentes formas, según los requerimientos determinados por la declaración de los datos del programa.
- Un CÓDIGO ALFANUMÉRICO establece la relación necesaria para que una computadora digital (que sólo procesa dígitos binarios) interprete el lenguaje que utiliza el usuario.
- Un CÓDIGO es una convención que determina una única combinación binaria para cada símbolo que ha de representarse.
- Al digitarse un carácter sobre el teclado, se emitirán una serie de señales binarias que quedará alojada internamente como una combinación de bits, de acuerdo con alguna representación de caracteres alfanuméricos preestablecida.
- FORMATO DE UNA ENTIDAD BINARIA → estructura y cantidad de bits de un determinado tipo de dato para su tratamiento en la computadora. Quien interpreta un grupo de bits bajo cierto formato es un programa. De esta forma, un byte puede ser un carácter alfanumérico, un operando o una instrucción, según quien lo interprete.

Códigos de Representación de Caracteres Alfanuméricos

- STRING → unidad formada por caracteres representados según un código alfanumérico.
- ASCII
  - de 7 bits
    - permite representar $2^7 = 128$ símbolos distintos.
    - los 3 bits más significativos se denominan BITS DE ZONA.
    - los últimos 4 bits son BITS DE DÍGITO
- ASCII ampliado
  - de 8 bits
    - debido a la necesidad de agregar caracteres, el código se amplió para
    - permite representar $2^8 = 256$ símbolos distintos.
    - tiene 4 bits de zona (uno más que el ASCII), y 4 bits de dígito.

VER TABLAS ASCII (página 53) ASCII ampliado (página 55).

Códigos de Representación Decimal (BCD)

- Son convenciones que permiten representar números decimales (0-9) en bloques binarios de 4 bits.
- BCD NATURAL o PURO o BCD 8421
  - Se lo suele llamar BCD 8421 porque los pesos en cada bloque coinciden con el valor de los 4 primeros pesos del sistema binario puro.
  - Ejemplos: 9 en decimal ⟷ 1001 en binario ⟷ 1001 en BCD puro.
    10 en decimal ⟷ 1010 en binario ⟷ 0001 0000 en BCD puro.
- BCD EXCESO TRES
  - Se obtiene a partir del BCD puro, sumando un 3 binario a cada cifra decimal.
  - Se logra un código simétrico, que facilita la operación de hallar el complemento de un número sólo con invertir sus dígitos.
- BCD AIKEN o BCD 2421
  - Adjudica a cada dígito binario el peso que resulta de la combinación 2421.
  - Es usado, también, en gran medida por su simetría, que permite hallar con facilidad el complemento de un número.
  - Ejemplo: el 7 decimal, en BCD 2421 es 1101, surge de la suma de la multiplicación de sus pesos, o sea: $2 + 4 + 1 = 7$.

CÓDIGOS DE REPRESENTACIÓN NUMÉRICA (NO DECIMAL)

COMA O PUNTO FIJO SIN SIGNO (ENTEROS POSITIVOS)

- es el formato más simple para representar números enteros positivos, expresados como una secuencia.
- rango de variabilidad (para un dígito X de n bits de magnitud): $\boxed{0 \leq X \leq 2^n - 1}$ Para $n=3$, $0 \leq X \leq 7$.

COMA O PUNTO FIJO CON SIGNO (ENTEROS)

- en cuanto al formato es similar al anterior, ya que reserva el bit del extremo izquierdo para el signo < 0 → positivo. / 1 → negativo.
- rango de variabilidad (para un número X de n bits de magnitud): $\boxed{-(2^{n-1}-1) \leq X \leq +(2^{n-1}-1)}$ Para $n=3$, $-3 \leq X \leq +3$.

COMA O PUNTO FIJO CON SIGNO CON NEGATIVOS COMPLEMENTADOS A 2 (ENTEROS)

- es igual al formato anterior, reservando el bit del extremo izquierdo para el signo < 0 → positivo. / 1 → negativo.
- la diferencia con el formato anterior es que los bits restantes se representan: para los < positivos → magnitud binaria real. / negativos → complemento a la base.
- rango de variabilidad (número X, de n bits de magnitud): $\boxed{-(2^{n-1}) \leq X \leq +(2^{n-1}-1)}$ Para $n=3$, $-4 \leq X \leq +3$
- la representación de los números negativos surge de aplicar el complemento a la base del número.

COMA O PUNTO FIJO CON SIGNO COMPLEMENTADOS A 1 (ENTEROS)

- como en el formato anterior, se reserva el bit del extremo izquierdo para el signo < 0 → positivo. / 1 → negativo.
- los bits restantes se representan: para los < positivos → magnitud binaria real. / negativos → complemento restringido de la magnitud.
- rango de variabilidad (número X, de n bits de magnitud): $\boxed{-(2^{n-1}-1) \leq X \leq +(2^{n-1}-1)}$

REALES EN COMA O PUNTO FLOTANTE (NÚMEROS MUY GRANDES Y NÚMEROS REALES)

- La forma de representar números fraccionarios tiene cierta semejanza con la NOTACIÓN CIENTÍFICA.
- Los números se representan considerando dos partes < 1ra → MANTISA → indica los bits que representan al número. / 2da → EXPONENTE → indica la posición del punto dentro del número.
- Como los registros que operan estos datos tienen una longitud fija, se asume un formato de < m bits / p bits > para < la mantisa / el exponente > y entonces el rango de representación de números reales es siempre finito.
- Habrá determinados números reales son "representables" en el formato y otros que no.
- Cuando el resultado de una operación supera los límites mayores definidos por el rango, se dice que hay error de OVERFLOW de resultado y, en un registro asociado a la ALU (registro de estado), se actualiza un bit.
- Cuando el resultado cae en el hueco definido por los límites inferiores del rango (el cero queda excluido), se dice que hay error de UNDERFLOW.
- Existen varias convenciones de representación de datos en punto flotante con exponente en exceso (para eliminar el signo del exponente se utilizan convenios que agregan al exponente un exceso igual a 2^{p-1}). Una de ellas es "punto flotante exceso 127", que admite dos variantes < formato de precisión simple → 4 octetos. / formato de precisión doble → 8 octetos.

REPRESENTACIONES REDUNDANTES

- Como el cambio de un bit en el almacenamiento o manipulación de datos origina resultados erróneos, se agregan "bits redundantes" para detectar e incluso corregir estas alteraciones.
- PARIDAD VERTICAL SIMPLE O A NIVEL CARÁCTER → al final de cada carácter se incluye un bit, de manera que la suma de 1s del carácter completo sea par o impar.
- PARIDAD HORIZONTAL A NIVEL DE BLOQUE → por cada bloque de caracteres se crea un byte con bits de paridad.
- PARIDAD ENTRELAZADA → combina el código de paridad vertical con el horizontal para construir un código de paridad entrelazada, que además de detectar errores permite corregirlos.
- CÓDIGO DE HAMMING → permite detectar y corregir los errores con sólo agregar p bits de paridad, de forma que se cumpla la relación siguiente: $2^p \geq i + p + 1$ donde < i → cantidad de dígitos binarios a transmitir. / p → cantidad de bits de paridad.

4 ARITMÉTICA DE LA COMPUTADORA

ARITMÉTICA BINARIA

- Una unidad aritmética es un circuito asociado con uno o más registros (que contienen los operandos de la operación) cuyo diseño varía según la computadora tratada, ya que depende de las operaciones que realice.
- Las operaciones simples (suma-desplazamiento-complemento) están implementadas en hardware, mientras que las más complejas (radicación-multiplicación) se realizan por medio de la ejecución de un software.
- REPRESENTACIÓN DE DATOS EN PUNTO FIJO (BINARIOS ENTEROS)
  - El punto o la coma no está presente, sino que su presencia se supone manteniendo la posición a extrema derecha fija.
  - Los números positivos se representan en la forma magnitud con peso y los negativos pueden representarse en "magnitud con signo", en "complemento a 1" o en "complemento a 2".
- OPERACIONES ARITMÉTICAS CON ENTEROS SIGNADOS
  - Los negativos se representan en complemento a 2.
  - En algunos casos, al realizar una suma binaria entre los pares de bits de igual peso puede que estas sumas parciales afecten al par de bits de la siguiente columna con un acarreo (CARRY).
  - En toda suma-resta
    - los operandos negativos están complementados a la base, así como los resultados obtenidos.
    - el signo forma parte de la operación, sumándose como cualquier otro bit.
  - Suponiendo un esquema (de una unidad aritmética) de registros para suma de 8 bits, éste está formado por:
    - un registro de 8 bits acumulador, que almacena el primer operando antes de la operación y el resultado después de ella.
    - un registro de 8 bits que almacena el segundo operando.
    - un registro de (por ahora, 4 bits) estado con 4 bits-bandera
      - V → OVERFLOW.
      - C → CARRY.
      - S → SIGNO.
      - Z → CERO
- OPERACIONES ARITMÉTICAS EN PUNTO FLOTANTE
  - Para sumar dos operandos en representación de punto flotante, se debe tener en cuenta el alineamiento del punto (o sea, que ambos exponentes sean iguales).
  - Cuando un operando en punto flotante está desalineado respecto del otro, los exponentes son distintos. Entonces, se debe desplazar una de las mantisas para alinearla con la otra.

ARITMÉTICA DECIMAL

- SUMA CON OPERANDOS BCD:

$$\begin{array}{lcl} & C_n\ C_{n-1}\ C_{i-1}\ C_0 & \text{acarreo} \\ A & a_n \dots a_i \dots a_0 & \text{dígitos respectivos de los decimales A, B.} \\ + B & b_n \dots b_i \dots b_0 & \\ \hline S & C_n\ S_n \dots S_i \dots S_0 & \end{array}$$

números decimales

Si $S_i > 9_{10}$ se suma 6_{10} (codificado en BCD) para obtener el resultado correcto

Ejemplo:

$$\begin{array}{r} 111 \\ 456 \\ +\ 789 \\ \hline 1245 \end{array}$$

$$\begin{array}{rrrr} & 0100 & 0101 & 0110 \\ + & 0111 & 1000 & 1001 \\ \hline & 1011 & 1101 & 1111 \\ + & 0110 & 0110 & 0110 \\ \hline & 10001 & 10011 & 10101 \\ + & 1 & 1 & \\ \hline & 0001 & 0010 & 0100 \quad 0101 \\ & 1 & 2 & 4 \quad 5 \end{array}$$

se suma 6

- RESTA CON OPERANDOS BCD:
  - Cuando el minuendo es mayor que el sustraendo → "si sumamos el minuendo al complemento del sustraendo, obtenemos el mismo resultado incrementado en 10^n, siendo n la cantidad de dígitos de M y de S."
  - Cuando el minuendo es menor al sustraendo → "el resultado obtenido es el complemento de la diferencia real".
  - El código BCD 8421 no es autocomplementario, ya que el complemento a 9 de un número no se obtiene simplemente invirtiendo sus bits, sino que se debe incluir una corrección.

Para el cálculo del complemento restringido de un dígito BCD 8421 (Natural)

al dígito BCD se le suman 6 unidades (0110)

al resultado se le invierten los bits

Ver más ejemplos (páginas 80-84)

5 ÁLGEBRA DE BOOLE

ÁLGEBRA DE LOS CIRCUITOS DIGITALES

- El Álgebra proposicional estudia la operación entre PROPOSICIONES (sentencias declarativas que puede ser verdadera o falsa) utilizando OPERADORES LÓGICOS (como la conjunción, la disyunción inclusiva o la disyunción exclusiva).
- Dadas 2 proposiciones p y q se puede confeccionar una tabla de verdad que represente las $2^2=4$ combinaciones.
- Una PROPOSICIÓN COMPUESTA está constituida por 2 o más proposiciones simples.
- El operador NEGACIÓN se aplica a una única proposición.

ÁLGEBRA DE BOOLE

- Constituye el fundamento teórico para el diseño de circuitos digitales.
  - OPERADORES BOOLEANOS
    - PRODUCTO LÓGICO → · "punto" → conjunción.
    - SUMA LÓGICA → + "más" → disyunción inclusiva.
    - COMPLEMENTO → negación.
  - TABLAS DE VERDAD → listas ordenadas de las 2^n combinaciones distintas de 0s y 1s, que se pueden obtener de la combinación del valor de n variables binarias.
  - PROPIEDADES
    - LEYES CONMUTATIVAS (suma y producto) → $a+b = b+a$; $a \cdot b = b \cdot a$
    - LEYES DISTRIBUTIVAS (suma y producto) → $a \cdot (b+c) = (a \cdot b) + (a \cdot c)$; $a + (b \cdot c) = (a+b) \cdot (a+c)$
    - LEYES DE IDENTIDAD → elementos neutros
      - 0 para la suma. → $a+0 = a$
      - 1 para el producto. → $a \cdot 1 = a$
    - LEYES DE COMPLEMENTACIÓN → $a + \bar{a} = 1$; $a \cdot \bar{a} = 0$
  - TEOREMAS
    - PRINCIPIO DE DUALIDAD.
    - LEY DE IDEMPOTENCIA.
    - LEY DE ABSORCIÓN.
    - ASOCIATIVIDAD en la suma y el producto.
    - LEY DE INVOLUCIÓN.
    - LEYES DE DE MORGAN.

FUNCIÓN BOOLEANA

- es una expresión algebraica constituida por variables binarias, los operadores binarios suma lógica y producto lógico, el operador complemento, el paréntesis y el signo igual.
- para n variables, se pueden definir 2^{2^n} funciones distintas.

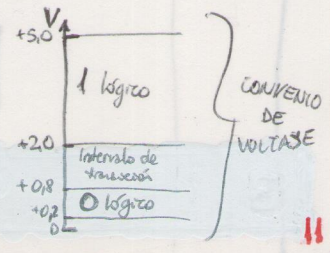
COMPUERTAS LÓGICAS

- Son representaciones de una red de conmutadores, donde cada conmutador se controla con una señal binaria. Cada una de estas señales constituye una "entrada". La estructura de la red genera una señal binaria de "salida".
- CONMUTADOR → dispositivo físico que permite controlar el flujo de un elemento (corriente eléctrica, agua, etcétera).
- CONMUTADOR IDEAL
  - modelo teórico que representa el comportamiento de un modelo real.
  - se utilizan los dígitos 0, 1 para representar el estado de una entrada o de una salida.
- CIRCUITO DE CONMUTADORES → grupo de varios conmutadores relacionados entre sí.
- SEÑAL DE ENTRADA → señal binaria usada para cambiar el estado de un conmutador.
- SEÑAL DE SALIDA → señal binaria ya regulada por un flujo de voltaje del conmutador.
- LÓGICA POSITIVA

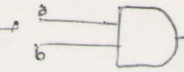
Valor lógico	Valor de la señal
1	High +5V
0	Low +0,2V

↔ LÓGICA NEGATIVA

Valor lógico	Valor de la señal
1	Low −4V
0	High −0,3V

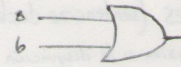


- **COMPUERTA AND** → la salida tomará el valor de 1 cuando todas sus entradas sean también un 1.



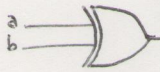
$F(a,b) = a \cdot b$ → su salida responde a la tabla de verdad del producto lógico.

- **COMPUERTA OR** → la salida de la compuerta toma el valor de 0 solamente cuando todas sus entradas son 0.



$F(a,b) = a+b$ → su salida responde a la tabla de verdad de la suma lógica.

- **COMPUERTA XOR** → la salida sólo toma un 0 cuando
  - todas sus entradas son 0, o cuando
  - una cantidad par de entradas son 1.

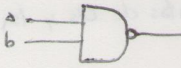


$F(a,b) = a \oplus b$

- **COMPUERTA NOT** → tiene una única entrada y una sola salida, siendo este último el complemento del valor de la entrada.

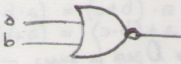
$F(a) = \overline{a}$

- **COMPUERTA NAND** → la salida solamente toma el valor 0 cuando todas las entradas son 1.



$F(a,b) = \overline{a \cdot b}$

- **COMPUERTA NOR** → la salida solamente toma el valor 1 cuando todas las entradas son 0.



$F(a,b) = \overline{a+b}$

La negación de la suma de dos variables puede expresarse como el producto de las variables negadas

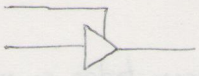
La negación del producto de dos variables puede expresarse como la suma de las variables negadas.

- **COMPUERTA YES** → no produce ninguna función lógica particular, dado que el valor binario de la salida es el mismo que el de la entrada.
  - se utiliza para amplificar la señal.

$F(a) = a$

Circuito Lógico

- es la representación de la estructura de una red de compuertas.
- su comportamiento se expresa algebraica por funciones booleanas.
- **TRANSISTOR** → conmutador electrónico que controla el flujo de tensión, con la particularidad de que puede cambiar de estado rápidamente.
  - elemento básico en la construcción de una compuerta.
- **COMPUERTA TRIESTADO** → aparece un tercer estado posible en la salida, denominado de **alta impedancia** y significa que la salida se desconecta. Dicho estado puede controlarse con una señal adicional denominada **entrada de habilitación**.
- **BUFFER TRIESTADO** → tiene una línea de entrada adicional d (desconectado) que opera como entrada de habilitación.



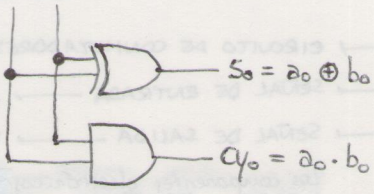
Circuito Sumador-Binario en Paralelo

- **CIRCUITO SEMISUMADOR** → su función es sumar 2 bits sin tener en cuenta el acarreo anterior
  - tiene dos salidas: suma, carry.

Ejemplo:

tabla de verdad para la suma de 2 bits

a_0	b_0	S_0	Cy_0
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

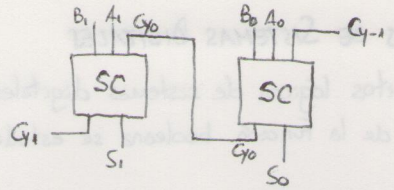


circuito lógico del semisumador

- **CIRCUITO SUMADOR COMPLETO**
  - tiene como finalidad sumar 2 bits y el acarreo anterior.
  - tiene dos salidas
    - la suma.
    - el nuevo acarreo.

Ejemplo: sumador completo de 2 números de 2 bits cada uno.

$$\begin{array}{rccc} & (C_n) & (C_{n0}) & (C_{n-1}) \\ + & & B_1 & B_0 \\ & & A_1 & A_0 \\ \hline & C_n & S_1 & S_0 \end{array}$$



FORMAS NORMALES O CANÓNICAS DE UNA FUNCIÓN

- Una función se puede representar por una serie de términos canónicos.
- TÉRMINO CANÓNICO → todo producto/suma en los que aparecen todas las variables que componen una función, en su forma directa o inversa.
- MINITÉRMINO → el producto de las variables en juego o sus negaciones individuales que hacen que el producto valga 1.
- MAXITÉRMINO → la suma de variables en juego o sus negaciones individuales que hacen que la suma valga 0.
- Si se expresa una función como la suma lógica de aquellos minitérminos que en su tabla de verdad tengan valor 1, se obtiene la expresión de su **FORMA NATURAL DISYUNTIVA**.
- Si se expresa una función como el producto lógico de aquellos maxitérminos que en su tabla de verdad tengan valor 0, se obtiene la expresión de su **FORMA NORMAL CONJUNTIVA**.

Ver ejemplos (páginas 105-107)

CIRCUITOS EQUIVALENTES

- Dos circuitos son equivalentes cuando son representados con distintas formas algebraicas pero responden a la misma tabla de verdad.

MINIMIZACIÓN DE CIRCUITOS

- Se realiza para que el diseño de un circuito sea lo más simple posible, que cumpla la misma función con la menor cantidad de compuertas posibles, la menor cantidad de entradas a ellas y en dos niveles.
- Para minimizar un circuito se debe minimizar la función.
- Uno de los métodos más usados para la representación de funciones y su posterior reducción es usando un mapa de Veitch-Karnaugh", que consiste en representar la matriz que contemple todas las combinaciones posibles.

Ver ejemplos (páginas 108-110).

6 Lógica Digital

Circuitos Lógicos de Sistemas Digitales

- En los circuitos lógicos de sistemas digitales, los valores 0 o 1 se representan con dos niveles de tensión.
- Las variables de la función booleana se estudian como las entradas en el circuito para luego estudiar su comportamiento.

Circuitos Combinacionales

- permiten que en las salidas se obtengan valores binarios transformados por la operación de las compuertas vinculadas en él y cuyo valor depende únicamente de los valores establecidos en las entradas.
- para su análisis, se consideran todos los posibles valores que pueden darse en las entradas. Hay $2^{\text{cantidad de entradas}}$ combinaciones posibles.
- NIVELES DE DESCRIPCIÓN → de menor a mayor nivel de abstracción, se encuentran →
  - 1° → la descripción del circuito electrónico.
  - 2° → los diagramas de lógica.
  - 3° → la representación algebraica.
  - 4° → el nivel algorítmico de diseño de componentes.
- CIRCUITO GENERADOR DE PARIDAD → permite la detección de errores en la transmisión de información binaria entre un emisor y un receptor.
- CIRCUITO COMPARADOR DE MAGNITUD → permite comparar 2 bits y así decidir si son iguales, si uno es mayor que el otro, si uno es menor que el otro.
- CIRCUITOS CONVERTIDORES DE CÓDIGO
  - su función digital es obtener en las salidas combinaciones binarias pertenecientes a alguna convención de representación de caracteres.
  - todo circuito derivado de una expresión minimizada con un mapa tiene dos niveles de compuertas AND y OR (el nivel lógico de la entrada debe atravesar dos compuertas hasta alcanzar la salida).
- CIRCUITOS CODIFICADORES → las señales binarias que entran se transforman en salidas codificadas.
- CIRCUITOS DECODIFICADORES → la información que ingresa al circuito son combinaciones binarias pertenecientes a alguna convención de representación de caracteres y las convierte a señales binarias para representar en algún dispositivo de salida de información.

  Ver ejemplos en páginas 118-121.
- CIRCUITO DECODIFICADOR N.2^N
  - permiten señalar en una de sus salidas qué combinación binaria se dio en la entrada.
  - solo una salida obtendrá el valor 1, mientras que las demás permanecen en 0.
  - un ejemplo de aplicación del circuito es el decodificador que se puede utilizar para el acceso aleatorio de una dirección en memoria.

  Ver ejemplos en páginas 121-123
- CIRCUITO DEMULTIPLEXOR
  - su función digital es "encaminar" la información que viene en una sola línea de entrada, en 1 de 2^n
  - una aplicación → distribuir una entrada de datos en serie a una salida en paralelo.
- CIRCUITO MULTIPLEXOR
  - su función digital es "encaminar" las señales binarias de 1 de 2^n líneas posibles de entrada en una única línea de salida.
  - una aplicación → convierte una entrada paralela a una cadena en serie.

  ambos se utilizan para transmitir datos a través de una sola línea en forma de serie → el multiplexor envía. → el demultiplexor recibe.
- CIRCUITOS "PROGRAMABLES" PARA MÚLTIPLES FUNCIONES
  - su proceso de programación es físico, y por lo tanto casi siempre es inalterable.
  - los dispositivos de lógica programable (PLD) son circuitos integrados con cientos de compuertas AND-OR que se relacionan por medio de enlaces electrónicos y conexiones fijas.
  - MEMORIAS DE SÓLO LECTURA (ROMs)
    - forman parte de los PLD y se caracterizan por tener → conexiones fijas en el arreglo de compuertas AND. → conexiones programables en el arreglo de compuertas OR.
    - cuando se las usan para implementar una función lógica, las entradas al decodificador AND constituyen las variables de la expresión booleana. Cuando se usan como programa fijo, constituyen la dirección de memoria para acceder.
    - cada línea de salida del decodificador se considera una palabra de memoria y cada enlace con un enlace se considera una celda de almacenamiento de un bit.
    - la cantidad de compuertas OR determina la cantidad de bits por palabra.
  - DISPOSITIVOS TIPO PAL
    - su función es similar a las ROMs, sólo que la red AND es programable y la OR es fija.
    - son más flexibles y rentables que las ROMs.
  - DISPOSITIVOS TIPO PLA o F-PLA
    - su función es similar a las anteriores.
    - son aún más flexibles, dado que ambas redes (AND y OR) son programables.

CIRCUITOS SECUENCIALES

- un circuito es secuencial cuando sus salidas dependen del valor de las entradas.
  del estado anterior del circuito representado.
- las funciones de salida deben contemplarse como entradas → el circuito tiene uno o más lazos de realimentación.
- se dividen en:
  - ASINCRÓNICOS → los cambios sólo se producen cuando están presentes las entradas sin necesidad de una señal de reloj.
  - SINCRÓNICOS → los cambios se producen cuando se establecen las entradas y, además, se genera una transición de señal de reloj.
- BIESTABLES o FLIP-FLOPS
  - celda binaria capaz de almacenar un bit.
  - tiene sólo 2 estados posibles de funcionamiento, permaneciendo en cualquiera de ellos si los niveles lógicos de las entradas no fuerzan su cambio.
  - al tener lazos de realimentación, pueden memorizar el efecto de combinaciones anteriores a las entradas.
  - BIESTABLE R-S ASINCRÓNICO
  - BIESTABLE R-S SINCRÓNICO (TEMPORIZADO) → la presencia de un pulso en una entrada especial determina el instante a partir del cual las entradas modifican su estado. Estos pulsos son generados por un sistema de reloj.
  - BIESTABLE J-K SINCRÓNICO → similar comportamiento al biestable R-S sólo que con algunas diferencias.
  - BIESTABLE T SINCRÓNICO → derivado del J-K.
  - BIESTABLE D SINCRÓNICO → derivado del J-K.

  Ver ejemplos en las páginas 131-136
- REGISTROS
  - funciones digitales que suelen utilizarse para retener información binaria en forma temporal.
  - pueden estar formados por 1 o más biestables.
  - la CPU tiene una pequeña memoria de acceso rápido formada por varios registros que almacenan información intermedia
  - REGISTROS PARALELO-PARALELO → transfieren la información que viene por las líneas de entrada en paralelo a salidas en paralelo.
  - REGISTROS CONTADORES
    - CONTADOR PROGRESIVO CON BIESTABLES T
      - cuenta eventos en forma ascendente.
      - Si se tiene un registro de n biestables, contará de 0 a 2^n-1, después de lo cual volverá a su estado inicial.
    - CONTADOR REGRESIVO CON BIESTABLES T
      - cuenta eventos en forma descendente.
      - contará de 2^n-1 a 0, después de lo cual volverá a su estado inicial 2^n-1.
  - REGISTROS CON FACILIDAD DE DESPLAZAMIENTO
    - la multiplicación-división de dos números se puede realizar sobre la base de sumas-restas sucesivas y desplazamientos.
    - en el sistema binario los desplazamientos se realizan corriendo los bits de un registro hacia la derecha o hacia la izquierda.

    instrucciones utilizadas:
    - SHL → desplazamiento lógico a la izquierda.
    - SHR → desplazamiento lógico a la derecha
    - SAL → desplazamiento aritmético a la izquierda, para valores signados
    - SAR → desplazamiento aritmético a la derecha, para valores signados.

    los desplazamientos lógicos no permiten conservar el signo del operando, razón por la cual para valores signados se deberán realizar desplazamientos aritméticos.
    - ROL → rotación a la izquierda.
    - ROR → rotación a la derecha.
    - RCL → rotación con acarreo a la izquierda.
    - RCR → rotación con acarreo a la derecha

    Ver gráficos de las instrucciones de las páginas 140-141

7 Diseño de una Computadora Digital

Módulo de Cálculo en una Computadora Digital

- Computadora Digital
  - su diseño es la organización de módulos de hardware relacionados por rutas de control y datos.
  - su función es permitir el flujo de señales binarias para transformar datos de entrada en información útil al usuario.
- Módulo
  - constituido por una configuración determinada de compuertas.
  - cada uno de ellos realiza operaciones sobre datos codificados en sistema binario, que se almacenan en registros asociados al módulo mientras dura la operación.
- Una operación aplicada a un registro se denomina MICROOPERACIÓN y se activa en un instante de tiempo sincronizado por los pulsos del reloj.

Relación entre el Diseño del Hardware y la Ejecución de Instrucciones

- La programación en lenguaje de máquina implica el conocimiento del tipo de instrucciones en CÓDIGO DE MÁQUINA a las que obedece la computadora.
- Se ha desarrollado un lenguaje simbólico de bajo nivel conocido como Assembler (cada computadora tiene su propio Assembler).
- Tanto el lenguaje simbólico de alto nivel como el lenguaje simbólico de bajo nivel permiten obtener las mismas instrucciones en LENGUAJE DE MÁQUINA (o sea, las que el procesador entiende y puede ejecutar).
- El programador utiliza un lenguaje simbólico que debe traducirse a código de máquina antes de su ejecución.
- INSTRUCCIONES
  - constituyen el lenguaje de máquina
  - el CÓDIGO DE UNA INSTRUCCIÓN es la combinación de bits que la unidad de control de la CPU interpreta para generar las microoperaciones que permitan su ejecución.
  - la forma de agrupar esos bits determina la estructura de la instrucción (FORMATO DE LA INSTRUCCIÓN)

CódigoDeOperación	DatoEnMemoria

  - Cuando una instrucción está alojada en la unidad de control, está en ESTADO DE EJECUCIÓN

Presentación del Modelo de Estudio

- La función de la CPU se puede separar en 2 partes:
  - tratamiento de instrucciones (la unidad de control se encarga de eso).
  - operación de datos (la unidad aritmético-lógica se encarga de eso).
- El medio de comunicación entre la memoria y la CPU es un bus que transmite órdenes y datos
- Proceso de interpretación y ejecución de instrucciones (por parte de la Unidad de Control)

  Cuando la UC ejecuta cada instrucción del programa, debe alternar sus etapas de búsqueda y ejecución desde la primera instrucción hasta la última. Esta secuencia del ciclo se denomina CICLO DE INSTRUCCIÓN

  1. Búsqueda de la instrucción en memoria } FASE DE BÚSQUEDA
  2. Interpretación del código de instrucción.
  3. Búsqueda del dato afectado por la instrucción.
  4. Generación de órdenes al módulo que opera sobre ese dato.

  } FASE DE EJECUCIÓN

- FASE DE BÚSQUEDA
  - La CU retiene la dirección de la instrucción en un registro especial denominado PUNTERO DE INSTRUCCIÓN (IP), cuya longitud depende de la cantidad de bits que se necesiten para direccionar cualquier instrucción en la memoria asignada al programa.
  - Las instrucciones se almacenan en palabras sucesivas y el IP debe poder incrementarse (tantas unidades como palabras ocupen las instrucciones) para señalar siempre la próxima instrucción para buscar, luego de una ejecución.
  - Una vez que la UC envía a memoria el contenido del IP, da la orden de lectura para que la palabra implicada se almacene en el registro de palabra de memoria (MDR).
  - Finalmente, se transfiere la instrucción leída a un registro interno de la UC, donde la instrucción permanece almacenada mientras dure su ejecución.
    - registro de instrucción (IR).
- Para saber en qué etapa se encuentra una instrucción, existe un Flag F que:
  - si F=1 → fase de búsqueda.
  - si F=0 → fase de ejecución.
- El que genera la secuencia de microoperaciones de las fases de búsqueda y de ejecución es un circuito denominado CONTROL DE INSTRUCCIÓN

- **FASE DE EJECUCIÓN** → Cada código de instrucción tiene asignada una secuencia de microoperaciones definida en el control:
  - HLT → inhibe la generación de microoperaciones.
    - determina fin de programa.
    - es imprescindible para que no se sigan buscando posiciones de memoria que no almacenan instrucciones.
  - INP → permite la entrada de un dato desde un periférico.
  - LDA → transfiere el dato almacenado en la palabra (que precede a la instrucción) al acumulador de la ALU.
  - SHR → se ordena al acumulador a desplazar su información un bit a la derecha.
  - STA → almacena el resultado final en memoria.

  Ver "flujo de datos entre los registros de la computadora básica" en páginas 161-163.
- **Otras instrucciones**
  - JMP HHH → instrucción de salto incondicional, que debe ser usada cuando se necesite romper con el orden secuencial del programa.
    - HHH es la dirección de la palabra de la instrucción a la que se quiere "saltar".
  - ANA HHH → asocia el acumulador y el dato obtenido mediante el operador AND, de manera que el contenido final del acumulador sea el resultado de la operación "producto lógico".
  - XOA HHH → asocia el acumulador y el dato obtenido mediante el operador lógico XOR, de manera que el contenido final del acumulador sea el resultado de la operación "suma exclusiva".
  - CLA → pone en 0 al acumulador.
  - INC → incrementa una unidad al dato cargado en el acumulador.
  - CMA → invierte los bits del acumulador (calculando el complemento restringido).
  - SNA HHH → salta siempre que el dato almacenado en el acumulador sea negativo (Flag S = 1) a la dirección HHH.
    - Si S=0, no se produce ninguna microoperación de ejecución.
  - SZA HHH → salta a la dirección HHH siempre que el dato almacenado en el acumulador sea 0 (Z=1).
  - SCA HHH → salta a la dirección HHH siempre que el último acarreo (flag C = 1) sea 1.
- **UNIDAD DE CONTROL Y SINCRONIZACIÓN DEL TIEMPO**
  - Toda microoperación está controlada por el tiempo.
  - CICLO DE RELOJ → intervalo entre dos pulsos consecutivos de reloj (0 y 1).
  - Cada ciclo de reloj se identifica como 1 Hz.
  - Una computadora controlada por un sistema reloj se denomina SINCRÓNICA.
  - El tiempo de una secuencia repetitiva constituye el CICLO DE MÁQUINA.
  - TIEMPO DE ACCESO A MEMORIA → tiempo que tarda la UC en buscar la información en la memoria y dejarla disponible para su uso.
  - Si la memoria de lectura es
    - no destructiva → CICLO DE MEMORIA = TIEMPO DE ACCESO A LA MEMORIA
    - destructiva → CICLO DE MEMORIA = TIEMPO DE ACCESO A LA MEMORIA + TIEMPO DE RESTAURACIÓN

$$\text{FRECUENCIA} = \frac{\text{Cant. Ciclos de Reloj}}{\text{Segundos}}$$

- **EL MÓDULO DE CÁLCULO**
  - el módulo de tratamiento de datos es la U-AL, donde los datos son tratados según órdenes de la UC, que interpreta la instrucción durante su estado de ejecución.
  - cada bandera se actualiza después de una operación en la ALU.
  - el estado de las banderas permite que el programador controle la condición final de una operación para establecer o no una ruptura de la secuencia normal del programa.
- El desplazamiento de bits en los registros de la ALU se realiza para multiplicar, dividir o "testear" un bit.
- Al comparar dos operandos, se efectúa una sustracción sin necesidad de almacenar el resultado, ya que las instrucciones permiten preguntar por la bandera indicadora de cero o signo.
- Cada vez que la ALU realiza una operación elemental, mantiene cierta información que indica el estado final de esa operación. Esta información queda retenida en el REGISTRO DE ESTADO, donde cada bit se denomina FLAG.

S	Z	V	H	O	P	Y	C

- S → indicador de signo
  - S=1 → resultado es negativo.
  - S=0 → resultado es positivo.
- Z → indicador de cero
  - Z=1 → resultado es cero.
  - Z=0 → resultado no es cero.
- V → indica overflow
  - V=1 → se produjo desborde
  - V=0 → no se produjo desborde.
- H → acarreo de los últimos 4 bits
  - H=1 → hubo.
  - H=0 → no hubo.
- P → paridad
  - P=1 → cantidad de 1s impar.
  - P=0 → cantidad de 1s es par.
- Y → interrupción
  - Y=1 → hay un requerimiento de atención de la CPU.
  - Y=0 → lo contrario.
- C → acarreo
  - C=1 → hubo acarreo
  - C=0 → no hubo acarreo

8 MICROPROCESADORES

MICROPROCESADORES Y MICROCONTROLADORES

- Un MICROPROCESADOR es un chip que, junto con la memoria principal, los buses del sistema, los módulos de E/S y los buses de E/S constituyen la estructura de una computadora completa integrada en una ARQUITECTURA ABIERTA (dado que es configurable según la necesidad del negocio, tanto hardware como software).
- Un MICROCONTROLADOR es una computadora completa con un programa de propósito no configurable (ARQUITECTURA CERRADA).

LONGITUD DE PALABRA

- Un microprocesador "procesa" bits que obtiene de una memoria, los opera y luego almacena el resultado en memoria. La cantidad de esos bits que el microprocesador procesa se denomina (como grupo) LONGITUD DE PALABRA.
- PALABRA DE MEMORIA → cantidad de bits a los que se puede acceder por vez.

CAPACIDAD DE DIRECCIONAMIENTO

- Tiene relación con el acceso a líneas que transfieren direcciones. (una dirección permite individualizar la zona de memoria principal que se va a leer o escribir).
- Con n bits del bus de direcciones, se obtiene un mapa de direcciones de 2^n.

NÚMERO DE INSTRUCCIONES

- La cantidad de instrucciones diferentes que un microprocesador puede "entender" en la mayoría de los casos implica su mejor capacidad para "hacer cosas distintas".

NÚMERO DE REGISTROS INTERNOS

- Refiérese a la cantidad de registros (con los que cuenta el procesador), cuya función es sustentar las necesidades de almacenamiento temporal durante la ejecución.
- REGISTROS VISIBLES → conjunto de registros que pueden actualizarse por las aplicaciones.
- REGISTROS INVISIBLES → registros utilizados para la administración del sistema.
- PIPELINES → técnica de ejecución en cadena que consiste en dividir el procesamiento en etapas y que éstas operen en paralelo.
- El x86 contaba con 2 unidades funcionales
  - UNIDAD DE EJECUCIÓN → decodificaba y ejecutaba las instrucciones.
  - UNIDAD DE INTERFAZ → mediante su bus (BIU) se encargaba de obtener y almacenar los datos en memoria principal y también de obtener las instrucciones del programa.
- REGISTROS DE CÁLCULO DE 16 BITS

15	8 7	0	
AH	AL		Ax → registro acumulador.
BH	BL		Bx → registro base (para direccionamiento).
CH	CL		Cx → registro contador
DH	DL		Dx → registro para datos.

Todos los registros se pueden utilizar en operaciones de cálculo aritmético o lógico

Es factible emplearlos con instrucciones que los afecten en su totalidad (Ax) o en la parte alta (AH) o en la parte baja (AL).

- REGISTROS PUNTEROS

	31	16 15	0	
EIP				IP → registro puntero de instrucción.
ESP				SP → registro puntero de pila.
EBP				BP → registro base de pila.
ESI				SI → registro índice fuente.
ESI				DI → registro índice destino.

- REGISTRO DE ESTADO

	31	16 15	0	
EFLAGS				F → se alojan todas las banderas

- aritméticas.
- de modo de trabajo del microprocesador.
- asociadas a interrupciones.
- acreítera.

REGISTROS DE SEGMENTOS

- registros que brindan soporte a aquellos sistemas operativos que administran la memoria como una agrupación de segmentos (utiliza un modelo segmentado).
- SEGMENTO → bloque lógico de tamaño ajustado al objeto que contiene.
- OBJETO → código de un programa, los datos que utiliza y una estructura de dato llamada PILA.
- almacenan la referencia binaria a la base de un segmento en memoria.

15 … 0	
CS	registro de base de segmento de código.
SS	registro de base de segmento de pila.
DS	registros de base de segmentos de datos.
ES	
FS	
GS	

RELACIÓN ENTRE LOS REGISTROS Y EL MODO DE DIRECCIONAMIENTO A DATOS →

DIRECCIÓN DE SEGMENTO + DIRECCIÓN BASE + ÍNDICE + DESPLAZAMIENTO

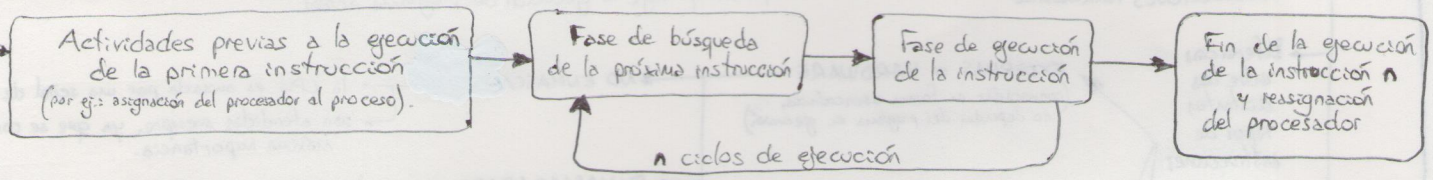
la base de un segmento se encuentra en los registros de segmento DS, ES o SS y se multiplica por 16, antes de sumarse al resto. El valor de una base para direccionamiento base "desnudo" del segmento se almacena en cualquiera de los registros BX o BP. El valor índice que recorre una estructura de dato se encuentra en uno de los registros índice SI o DI, y el desplazamiento puede ser una constante de 8, 16 o 0 bits.

La base y el índice pueden modificarse durante el processamiento, pero el desplazamiento es una constante.

VELOCIDAD DEL MICROPROCESADOR

→ Cada microprocesador tiene su propio reloj interno, cuya frecuencia indica con qué velocidad puede procesar bits.

CICLO DE INSTRUCCIONES



→ Programa de ejemplo: → se halla almacenado en un segmento de código, cuya base es 1531.

① 1531:0100 8B1E0002 MOV BX,[0200].
② 1531:0104 0307 ADD AX,[BX].
③ 1531:0106 D1E0 SHL AX,1.
④ 1531:0108 83C302 ADD BX,+02.

① Es una instrucción de transferencia y está ubicada a partir del desplazamiento 0100 respecto de 15310 (dirección física).
La instrucción ocupa 4 bytes: 0100, 0101 (corresponden al código de operación 8B1E), 0102 y 0103 (corresponden al campo DATA).
Es una instrucción que requiere la lectura de un dato de memoria alojado en el desplazamiento 0200 y su carga posterior en el registro general BX. El modo de direccionamiento sería directo.

② Es una instrucción aritmética que comienza en el byte 0104 y ocupa 2 bytes: el 0104 y el 0105, que corresponden al código de operación 0307.
Su ejecución suma al registro general AX el valor de BX (los corchetes indican "contenido de") y no accede a memoria para buscar dato. El modo de direccionamiento sería directo de registro.

③ Es una instrucción de desplazamiento que comienza en el byte 0106 y ocupa 2 bytes: el 0106 y el 0107, correspondientes al código de operación D1E0, que permite el desplazamiento de 1 bit a la izquierda (SHIFT LEFT) del registro AX y, por lo tanto, no accede a la búsqueda de un dato en memoria principal: el modo de direccionamiento sería inmediato a registro.

④ Es una instrucción aritmética que comienza en el byte 0108 y ocupa 3 bytes: el 0108, el 0109 y el 010A, los dos primeros correspondientes al código de operación 83C3, que efectúa la suma del valor 02 al registro BX.
No accede a memoria para obtener el dato y pertenece al modo de direccionamiento inmediato a registro.

Dicho programa no tiene sentido lógico, ya que es una rutina de un programa mayor.
Ocupa 11 (once) bytes en el segmento de código en la memoria principal.

- SECUENCIA DE LLENADO DE COLA
  - La BIU se encarga de controlar la transferencia entre el entorno y el microprocesador, y a su vez "entrega" las instrucciones a la unidad de decodificación a medida que se necesiten.
  - La unidad de decodificación es la que interpreta el código de operación y "reconoce" la instrucción, además de obtener el dato según el modo de direccionamiento especificado en el código de operación.
  - De esa manera, genera la demanda de cola y, como es independiente de la Unidad de Ejecución, puede darse el caso de que mientras se decodifica una instrucción todavía se esté ejecutando la anterior, generando un paralelismo.
  - En una situación ideal, la tendencia es lograr la ejecución de una instrucción por ciclo de reloj.
  - El objetivo es extraer una instrucción de memoria, decodificarla y redirigirla a unidades funcionales de la Unidad de Ejecución. Mientras estas unidades "ejecutan" la instrucción, deberá extraerse y decodificarse otra instrucción para redirigirla a otras unidades de la Unidad de Ejecución no comprometidas en la ejecución de la anterior. Así, cuantas más unidades formen parte de la UE, mayor será la posibilidad de "ejecución en paralelo". Este procedimiento es casi óptimo, hasta que aparece una instrucción de salto.
  - La UE debe contar con registros especiales e invisibles al programador de aplicaciones para resguardar el entorno de ejecución al momento que se produzca el salto.

CAPACIDAD DE INTERRUPCIÓN

- Las interrupciones y las excepciones son acontecimientos causados tanto por los dispositivos de E/S como por el programa que se ejecuta en el microprocesador, y su efecto produce una suspensión de la actividad actual del microprocesador, para pasar a ejecutar un servicio que "interprete el manejo de esa interrupción".
- Los dispositivos externos utilizan interrupciones para informar su estado o solicitar la ejecución de actividades que le son necesarias.
- Cada interrupción está asociada con un número que la identifica.
- INSTRUCCIONES INTERNAS o INSTRUCCIONES SOFTWARE → instrucciones programadas que causan la suspensión momentánea del programa que las convoca para bifurcar el servicio solicitado. Este se ejecuta y se retoma el programa interrumpido.
- INSTRUCCIONES EXTERNAS o INSTRUCCIONES HARDWARE → servicios residentes en memoria principal que son bifurcados luego de que el microprocesador reciba una señal de interrupción y deje la ejecución del programa actual.
- Diferencias entre los distintos tipos de interrupciones
  - EXTERNAS O HARDWARE (convocadas en forma asincrónica, no dependen del programa en ejecución)
    - NO ENMASCARABLES
      - la CPU es avisada por una señal de control (NMI)
      - son atendidas siempre, ya que se consideran de máxima importancia.
    - ENMASCARABLES
      - la CPU es avisada por otra señal (INTR).
      - No siempre debe ser atendida, ya que se consulta una bandera de estado de habilitación de interrupciones.
        - Si la bandera lo autoriza, la CPU suspende en forma momentánea la ejecución del programa activo y ejecuta el servicio de interrupción.
        - Si la bandera está desactivada, la CPU no la tiene en cuenta y continúa con la ejecución del programa activo.
  - INTERNAS O SOFTWARE (convocadas por el programa) → se pueden producir por la ejecución de una instrucción específica dentro de un programa, para solicitar una interrupción que cumpla alguna función determinada.
  - EXCEPCIONES (provocadas como consecuencia de anomalías que se producen y detectan durante la ejecución del programa y a causa de ella)
    - FALTAS O ERRORES → se pueden detectar y corregir antes de que se produzca la ejecución de una instrucción determinada.
    - TRAMPAS → se detectan una vez ejecutada la instrucción que las provoca.
    - ABORTOS → se detectan sin localizar la instrucción que las provoca, abortando la ejecución del programa.
- Toda la información de la CPU asociada con la ejecución se almacena en una pila (ubicada en memoria principal). Este procedimiento permite resguardar el entorno de CPU para luego reanudar la ejecución a partir del momento en el que se produjo la interrupción.
- RESTAURACIÓN DE CONTEXTO → procedimiento que consta de restaurar el entorno del CPU, "rescatando" la información que tenían los registros internos de la pila.
- Al detectarse una interrupción y atenderse, se pueden indicar tres pasos:
  - PROGRAMA EN EJECUCIÓN
  - PRESENTACIÓN DE LA INTERRUPCIÓN
    - Resguardo de contexto de CPU en la pila.
    - Ejecución del servicio de atención de interrupción.
    - Restauración del contexto de CPU.
  - PROGRAMA NUEVAMENTE EN EJECUCIÓN.

- CONCEPTO DE PILA
  - La PILA es una estructura de dato en memoria de acceso LIFO.
  - El segmento de pila (SS) se accede, también, con criterio LIFO.
  - La CPU se encarga del acceso a la pila y la utiliza para:
    - almacenar la dirección de retorno (IP) y el segmento de código (CS) cuando se llama a una subrutina.
    - almacenar el estado del procesador cuando se produce una interrupción. Apila el CS, el IP y los Flags.
    - pasar parámetros entre dos procedimientos.
  - El acceso a pila se realiza a través de los registros punteros SP (que contiene la dirección del próximo elemento de la pila; por eso, cualquier acceso a la pila utilizará el par "base de segmento : desplazamiento") y BP.
  - La carga/escritura y la extracción/lectura de datos es un procedimiento software.
  - Las instrucciones que a causa de su ejecución afectan a la pila son: CALL, RET (que sirven para invocar y dar retorno a un procedimiento o subrutina), INT y IRET (cumplen la misma función que CALL y RET, pero sólo cuando se invoca una subrutina de interrupción).
  - Las instrucciones que permiten el acceso a la pila son PUSH (que empuja o produce el almacenamiento de la palabra en la pila y luego decrementa el SP) y POP (que extrae o saca una palabra de la pila y "desafecta" las posiciones de memoria incrementando el SP.

ALIMENTACIÓN

- La alimentación de los distintos componentes de la computadora proviene de una fuente.
- FUENTE → dispositivo que transforma la corriente de la red eléctrica (alterna) para que sea aceptable para los circuitos electrónicos (corriente continua).

TECNOLOGÍA

- CISC
  - Se pueden ejecutar instrucciones simples o complejas (éstas últimas utilizan varias microinstrucciones).
  - Aumenta el número de instrucciones del set cuando éste admite tanta variedad de modos de direccionamiento.
- RISC
  - Cada instrucción tiene la mayor cantidad de microoperaciones solapadas posibles, de modo que la mayoría de ellas se ejecute en un ciclo de reloj.
  - Las instrucciones tienen el mismo tamaño, lo que facilita el funcionamiento del pipeline.
  - Las instrucciones que realizan operaciones aritméticas son de referencia a registro y, en general, tienen especificados 2 para los operandos y 1 para el resultado.
  - El formato típico de una instrucción aritmética RISC es:

Código de Operación	Registro 1	Registro 2	Registro 3

- EPIC
  - Permite agrupar instrucciones para ejecutarlas de forma paralela en forma específica.
  - PREDICACIÓN → método para manejar saltos condicionales ("ramificaciones condicionales"). La idea es que el compilador planifique ambos caminos posibles de la ramificación, para que se ejecuten en el procesador simultáneamente.
  - CARGA ESPECULATIVA → la especulación trata de aprovechar el microprocesador cuando está en períodos de latencia. En ese momento "especula" sobre las instrucciones y los datos que va a necesitar más adelante y los carga.

9 MEMORIAS

→ Se utilizan para retener los bits que constituyen el alfabeto digital para representar programas y datos.

CLASIFICACIÓN

- SEGÚN EL MODO DE ACCESO A LA UNIDAD DE INFORMACIÓN
  - ACCESO ALEATORIO → cuando un componente de selección habilita una palabra o posición e inhabilita a las demás.
  - ACCESO SECUENCIAL → cuando para acceder a una unidad de información se establece una posición de referencia, a partir de la cual comienza un rastreo de la unidad de información que consiste en la lectura de todas las unidades que la precedan
  - ACCESO ASOCIATIVO → cuando la búsqueda de la unidad de información implica la comparación de un grupo de bits de la unidad de información con el contenido de una posición de memoria. Las palabras no se identifican por su dirección y el objetivo de su lectura es verificar si este grupo de bits coincide con el contenido de alguna de ellas.
- SEGÚN LAS OPERACIONES QUE ACEPTAN POR CADA ACCESO
  - LECTURA/ESCRITURA → cuando admite ambas operaciones. (vivas)
  - SÓLO LECTURA → cuando permite únicamente dicha operación (muertas)
- SEGÚN LA DURACIÓN DE LA INFORMACIÓN
  - VOLÁTILES → cuando pierden su información con el corte del suministro de corriente.
  - PERMANENTES → cuando su información no se pierde al cortarse

DIMENSIÓN

- CAPACIDAD → cantidad de información que se puede almacenar (en una memoria).
  - puede expresarse en bits, bytes, kilobytes, megabytes, gigabytes, etc.
- BIT → la capacidad medida en bits se utiliza para los registros, que son los soportes de información de menor ca
- BYTE → también es aplicable a registros.
  - la capacidad de memorias mayores se expresa en unidades que expresan bytes.
- KILOBYTE → 1024 Bytes.
- MEGABYTE → 1024 KB.
- GIGABYTE → 1024 MB.
- TERABYTE → 1024 GB.
- PETABYTE → 1024 TB.
- EXABYTE → 1024 PB.
- ZETTABYTE → 1024 EB.
- YOTTABYTE → 1024 ZB.

MEMORIAS RAM ESTÁTICAS Y DINÁMICAS

- En una memoria de semiconductores se dice que cada bit se aloja en una celda binaria.
- SRAM → son memorias vivas, volátiles y estáticas.
  - puede modificarse con un proceso de escritura. Si no, permanece inalterable.
- DRAM → son memorias vivas, volátiles y dinámicas (el hecho de que sean dinámicas hace que su información se vaya degradando con el tiempo).
  - son más lentas que las SDRAM pero tienen mayor capacidad. Por otro lado, el costo por bit de almacenamiento es mucho menor que en las SDRAM.
  - cada celda almacena un 1 que se representa con la carga de un condensador. Este conserva su carga con cierto nivel de deterioro durante un período preestablecido; por lo tanto, antes de que se pierda la información hay que restablecer la carga. Este proceso se denomina CICLO DE REFRESCO, y está a cargo (junto con el control de acceso) del CONTROLADOR DE MEMORIA.
- RAM con acceso directo → para permitir el acceso a la información en forma aleatoria se organiza una matriz, cuyas columnas están formadas por los bits de igual peso que todas las filas.
  - el número que identifica la palabra en un acceso aleatorio se denomina DIRECCIÓN FÍSICA.

Ver ejemplos en las páginas 212-215

- RAM con acceso asociativo → son accesibles por contenido. Dicho contenido buscado se denomina RÓTULO/DESCRIPTOR/ARGUMENTO.
  - la posibilidad de asociación requiere que todas las celdas de almacenamiento se relacionen con circuitos que permitan la comparación, razón por la cual se vuelven más caras y su uso se justifica en aplicaciones en las que sea imprescindible la búsqueda por contenido.

JERARQUÍA DE MEMORIAS

- Está dada por 3 atributos
  - velocidad de acceso.
  - costo de la celda (bit).
  - capacidad de almacenamiento.
- NIVELES
  - REGISTROS INTERNOS DEL PROCESADOR
    - denominados también "registros de propósito general"
    - son de alto nivel y los más importantes.
    - se utilizan como registros de almacenamiento de operandos o registros de cálculo.
  - MEMORIA CACHÉ
    - memoria de semiconductores.
    - es más rápida que la DRAM
    - de mayor complejidad que la DRAM, y de poca capacidad.
  - MEMORIA SDRAM
    - memoria de semiconductores lenta.
    - de menor complejidad, dotan al sistema de mayor capacidad.

    } soportes para el almacenamiento temporal de instrucciones y datos intercambiables, a los que el microprocesador accede en forma directa.
  - MEMORIA AUXILIAR
    - su capacidad de almacenamiento es mucho mayor que la de la memoria DRAM.
    - la demora en el acceso a la información se debe, en su mayor parte, a los movimientos mecánicos necesarios para el acceso al soporte.
    - el soporte de bits en las memorias auxiliares corresponde a la tecnología de medios magnéticos y ópticos.

MEMORIAS CACHÉ

- Son de tecnología SDRAM.
- Se utiliza como memoria intermedia entre el procesador y la memoria DRAM.
- Almacena temporalmente la información a la que se accede con mayor frecuencia en la memoria.
- Una caché es un subsistema constituido por una memoria de etiquetas, una memoria de datos y un controlador.
- El CONTROLADOR de la memoria caché es el "cerebro" del sistema de este tipo de memorias. Cuando el controlador "trae" una instrucción de la memoria principal, trae también las próximas instrucciones y las almacena en caché, especulando con la posibilidad de que la próxima instrucción se necesite. En ese caso, se optimiza el tiempo.
- La cantidad de líneas de la caché depende de su capacidad expresada en bytes y que dicha medida es el tamaño de la memoria de datos.
- CONEXIÓN EN SERIE
  - es la más usada por los procesadores actuales.
  - VENTAJA → cuando hay un acierto, deja el bus libre para su uso.
  - DESVENTAJA → penalización de tiempo: si el dato no está en la caché, tiene que buscarlo en la memoria principal.
- CONEXIÓN EN PARALELO
  - todas las solicitudes del procesador llegan en forma simultánea a la memoria principal y a la caché.
  - en caso de acierto (la posición buscada se encuentra en la caché), el controlador entrega la posición buscada al microprocesador y genera una señal que aborte el ciclo iniciado para la lectura en la memoria principal.
  - en caso de falla, la posición se obtiene desde la memoria principal.
- PRINCIPIOS DE FUNCIONAMIENTO
  - La comunicación entre el procesador y la memoria RAM es continua, ya que el microprocesador busca, en la memoria principal, la instrucción para ejecutarla o un dato requerido por la ejecución de una instrucción.
  - La comunicación entre el procesador y la RAM se establece mediante
    - el bus de direcciones.
    - el bus de datos.
- CACHING
  - procedimiento gestionado por el controlador que anticipa las necesidades de posiciones de memoria principal de acuerdo con cierto cálculo de probabilidad de uso y utilizando criterios que consideren los principios de vecindad espacial y temporal.
  - la capacidad de la caché varía de acuerdo a su tecnología, y su rendimiento depende tanto de la efectividad de la gestión de caching como de su tamaño.
- TRADUCCIÓN DE LA DIRECCIÓN FÍSICA → Una dirección física deberá ser mapeada por el controlador, ya que el tamaño de la memoria DRAM no coincide con el de la caché
  - La gestión de traducción puede clasificarse según 3 formas de organización del sistema.
    - TOTALMENTE ASOCIATIVA → a cada línea le corresponde una etiqueta.
    - ASOCIATIVA DE 1 VÍA → hay una etiqueta por número sector RAM o línea caché.
    - ASOCIATIVA DE N VÍAS → cada línea admite n etiquetas y n matrices de datos, implicando una caché n veces más grande y menor posibilidad de fracaso.
- ACTUALIZACIÓN DE CACHÉ → Debe actualizarse una caché cuando se detecta una falla o ausencia de la palabra buscada.
  - Se puede clasificar en por lo menos 3 categorías
    - LRU → la de uso menos reciente.
    - FIFO → la primera en entrar es la primera en salir.
    - RND → aleatorio

- ACTUALIZACIÓN DE LA MEMORIA PRINCIPAL → Una modificación en caché implica la actualización de memoria.
  - Existen políticas de actualización de la memoria principal que pueden clasificarse en:
    - ESCRITURA INMEDIATA
      - consiste en actualizar simultáneamente ambas memorias, de modo tal que no se genere incongruencia entre la información almacenada en ambos niveles.
      - reduce el rendimiento.
      - son los subsistemas más económicos.
    - ESCRITURA OBLIGADA
      - admite sólo las actualizaciones estrictamente necesarias en memoria principal.
      - la información en la caché puede ser actualizada varias veces antes de ser actualizada en memoria principal.
- NIVELES DE CACHÉ
  - la ventaja de contar con varios niveles de caché es que se reducirá el número de ciclos de bus utilizados en el acceso a RAM.
  - 1
    - es de tamaño reducido.
    - se ubica interceptando los datos que entran en el microprocesador desde la memoria o los que salen ya procesados hacia la memoria.
  - 2
    - es de tamaño mayor.
    - se ubica (de estar incluida en el sistema) entre la caché de 1er nivel y la memoria principal.
    - atiende pedidos menos frecuentes que la primaria.
    - se utiliza como ampliación de la caché de nivel 1.
    - dada su mayor capacidad de almacenamiento, y además de contener la misma información almacenada en la primaria, contendrá unos Kbytes más.

MEMORIA PRINCIPAL

Ver ejemplos en las páginas 227-229

- MEMORIAS RAM DINÁMICAS (SDRAM)
  - son de lectura/escritura y se utilizan para almacenar una gran cantidad de bytes en la MP de las PC.
  - son de bajo costo y tienen una gran capacidad de almacenamiento, en relación a las estáticas.
  - son más lentas que las estáticas.
  - REFRESH → proceso que consiste en recargar los condensadores que tienen un 1 para evitar que la información se pierda.
  - CONTROLADOR DE MEMORIA DINÁMICA → se "multiplexan" las direcciones: para acceder a una celda se considera una parte de la dirección física para acceder a la fila y la otra para acceder a la columna.
    como ambas direcciones se presentan al chip de memoria sobre el mismo bus de direcciones, se deben activar dos señales: RAS (que habilita la selección de la fila) y CAS (que habilita la selección de la columna).
  - LATENCIA CAS → indica el número de ciclos de reloj que transcurren desde que se realiza la demanda de datos hasta que éstos se ponen a disposición del bus de datos.
  - TIEMPO DE PRECARGA → tiempo adicional (al tiempo de acceso) generado al producirse dos lecturas consecutivas sobre el mismo chip.
  - MÓDULOS → pequeñas placas en las que se sueldan los chips de RAM, con pines en un extremo.
  - VELOCIDAD DEL BUS DE MEMORIA → para calcular el ancho de banda de las memorias se realiza el producto entre el ancho de bus expresado en bytes y la frecuencia de trabajo efectiva.

    Ver ejemplo "DDR SDRAM" en la página 232
  - DETECCIÓN Y CORRECCIÓN POR ECC
    - el ECC 743 es un código de bits que permite la detección y la corrección de errores.
    - los bits que se anexan permiten realizar una comprobación de los datos a la hora de la lectura de los mismos.

LA MEMORIA COMO UN ESPACIO LÓGICO

- Para el procesador, la memoria es un hardware, con forma de matriz de m.n, identificables por una dirección física.
- El microprocesador no hace diferencias entre las distintas tecnologías que pueden constituir una memoria principal, simplemente envía la dirección física del byte vía bus de direcciones.
- La manera en que una memoria se administra y se gestionan sus accesos depende fundamentalmente del sistema operativo.
- MAPEAR (una dirección) → aplicar un algoritmo para establecer la correspondencia entre direcciones físicas y lógicas.
- ALMACENAMIENTO DE BYTES EN MEMORIA → ya sea de datos o de código de programa, sigue un orden específico.

  0x4A3B2C1D
  - BIG-ENDIAN → {4A, 3B, 2C, 1D}
  - LITTLE-ENDIAN → {1D, 2C, 3B, 4A}
- GESTIÓN DE MEMORIA Y MODOS DE OPERACIÓN DE LOS PROCESADORES
  - El modo de operación del procesador (MODO PROTEGIDO) implica protección de la memoria.
  - Modelos de gestión de memoria
    1. como un espacio de direcciones físicas (MODELO PLANO)
    2. un espacio que alberga segmentos puros (MODELO SEGMENTADO)
    3. un espacio que alberga páginas (MODELO PAGINADO).
    4. un espacio que alberga segmentos-paginados (MODELO HÍBRIDO)

- Cálculo de direcciones físicas en modo real

Dirección Segmentada: 3A2B:1301 ← Desplazamiento — SS:SP o CS:IP (pila)

Base del segmento

$3A2B_{16} \cdot 10_{16} = 3A2B0_{16}$
$+ 1301_{16}$

$\boxed{3B5B1_{16}}$ → DIRECCIÓN FÍSICA

- MODELO DE MEMORIA SEGMENTADA PURA
  - Para acceder a la memoria física se utiliza una técnica que consiste en dividirla en territorios pequeños con límites determinados por el software (SEGMENTACIÓN).
  - Un segmento es un bloque lógico en memoria, de tamaño variable
  - DIRECCIÓN LÓGICA
    - entidad binaria que identifica una locación de memoria a nivel software, que no se corresponde con la dirección física y necesita ser mapeada.
    - cada dirección lógica se compone de dos entidades: una para identificar el segmento y otra para identificar un desplazamiento dentro del segmento.

- MODELO DE MEMORIA VIRTUAL
  - técnica empleada para que se hagan en forma automática (y sin intervención del procesador) la fragmentación y el intercambio de fragmentos entre el disco y la memoria.
  - la memoria virtual es una forma de administrar la memoria para asignar más de ella a cada tarea, utilizando almacenamiento en disco, que suele ser el dispositivo de acceso directo más rápido. Así, se simula que se dispone de una memoria mucho más amplia que la que el sistema tiene en realidad.

- MODELO DE MEMORIA VIRTUAL PAGINADA o PAGINACIÓN POR DEMANDA
  - MAPEO DIRECTO
    - consiste en dividir
      - la memoria principal en bloques de longitud fija (PÁGINAS FÍSICAS)
      - el espacio de direccionamiento virtual en bloques de igual longitud (PÁGINAS VIRTUALES).
    - mientras el mapeo sea exitoso, no se producen errores. Pero, si el programa intenta acceder a una posición perteneciente a una página no presente en la memoria principal, el procesador genera un error (PAGE FAULT), que provoca una interrupción de tipo excepción.
    - SWAP OUT → procedimiento en el que el programa copia la página, si se actualizó, en el disco antes de reemplazarla.
    - SWAP IN → procedimiento que consiste en recuperar una página almacenada en disco.
    - HIPERPAGINACIÓN → intercambio excesivo de páginas.
  - MAPEO ASOCIATIVO
    - la cantidad de páginas físicas en memoria real será siempre menor que la cantidad de páginas virtuales posibles.
    - se crea una tabla que tenga igual cantidad de entradas como páginas físicas existan, y que pueda implementarse en una pequeña memoria asociativa.
    - cada entrada de la tabla contiene un campo que indica un número de página virtual presente en memoria principal y un campo que indica el número de página física correspondiente.

- MEMORIA VIRTUAL SEGMENTADA o SEGMENTACIÓN POR DEMANDA
  - se organiza el espacio de direcciones virtuales en bloques de tamaño variable (segmentos).
  - los segmentos residen tanto en la memoria principal como en el disco.
  - En un Intel de 32 bits, la unidad de cálculo de dirección física se llama Unidad de Gestión de Memoria o MMU, que está constituida por la unidad de segmentación y la unidad de paginación. Como la memoria principal es más pequeña que la virtual, la unidad de segmentación se encarga de detectar si un segmento se encuentra en el disco y no en la memoria física.
  - la TABLA DESCRIPTORA DE SEGMENTOS se referencia con un registro base especial asociado al microprocesador, denominado REGISTRO DE TABLA LOCAL ACTIVA.
  - la MMU consulta dicha tabla por cada nuevo cálculo de dirección física, ya sea de una instrucción en fase de búsqueda o de un dato en fase de ejecución.
  - Una dirección virtual está constituida por un campo selector y un campo desplazamiento. En una x86, los primeros 16 bits de orden superior se consideran como selector y los 32 bits siguientes, de desplazamiento.

- descripción de algunos atributos
  - G bit de granularidad indica si el segmento está dividido en páginas.
  - P bit Present indica si el segmento está en memoria principal.
  - S System indica si el segmento es de sistema.

- la unidad de paginación crea para cada tarea una tabla (DIRECTORIO DE TABLAS DE PÁGINA).
- la tabla de páginas, para formar la dirección física final, utiliza los atributos más importantes asociados con ellos:
  - D bit dirty, que indica si la página se escribió.
  - P bit Present, que indica si la página está presente.
  - A Bit Access, que indica si se accedió a la página (tanto para lectura como para escritura).

- TLB → es una memoria de tipo caché que almacena las últimas direcciones físicas traducidas por la unidad de paginación.

ADMINISTRACIÓN DE MEMORIAS EXTERNAS

- Todas las memorias que se alojan en una unidad de E/S se denominan MEMORIAS EXTERNAS.
- Son imprescindibles para retener información cuando el sistema está desconectado de la fuente de alimentación.
- Los parámetros para tener en cuenta a la hora de elegir un soporte (elemento físico que retiene los bits de información) son:
  - el costo por bit.
  - el tiempo de acceso.
  - el modo de acceso.
  - la capacidad de almacenamiento.
  - el tipo de tarea que utiliza la información.

- **ARCHIVO**
  - estructura de las memorias externas en donde se aloja todo tipo de información.
  - un archivo se diferencia de los demás por el nombre simbólico que se le asigna en el momento de su creación.
  - los archivos se pueden estructurar de diferentes maneras
    - una cadena de bytes.
    - una agrupación de registros lógicos.
    - una base de datos (una agrupación de archivos).
  - acceso a archivos
    - un proceso que puede leer el archivo en orden → SECUENCIAL.
    - se accede al archivo en forma directa → ALEATORIO
  - atributos
    - protección → determina quién y cómo se puede acceder al archivo.
    - sólo lectura → puede leerse, pero no modificarse.
    - oculto → existe, pero no aparece en la lista del directorio.
    - sistema → archivo de uso exclusivo del sistema operativo.

- **DISCO MAGNÉTICO** (desde el punto de vista lógico)
  - están constituidos por **PLATOS** rígidos, dispuestos en unidades de cabezas fijas o móviles, agrupados. Estos platos giran alrededor de un eje, dispuesto en la unidad, al que se accede mediante cabezas lectograbadoras.
  - la forma en que las cabezas recorren la superficie del plato describen círculos concéntricos (PISTAS). Una pista es una división lógica (no física) del soporte.
  - por cada cara grabable habrá una cabeza lectograbadora, que accede a un cilindro en particular. Dado que la información se graba en sentido vertical, se completa un cilindro y luego se posiciona el brazo en el siguiente.
  - cada pista está dividida en sectores, cuya cantidad es fija. Las pistas internas son grabadas más velozmente que las externas, ya que las primeras tienen una circunferencia menor.
  - la unidad de acceso al soporte es una agrupación de registros lógicos, denominada bloque. En Windows, un bloque se denomina CLUSTER.
  - "dar formato" a un volumen permite delimitar el territorio sobre el soporte. El "formateo" es el nombre del comando que ejecuta el programa de inicialización de un disco y que genera la grabación de LABELS. Así, se puede localizar el lugar donde se graban los archivos.
  - PARTICIÓN → unidad lógica creada en un mismo disco físico.
  - En el área de sistema (territorio cuya información se utiliza para la administración del soporte) pueden haber:
    - un sector de arranque → almacena una rutina de carga del sistema operativo.
    - una tabla índice → señala la locación inicial de cada archivo grabado.
    - una tabla de declaración de archivos (DIRECTORIO).
  - FAT
    - tabla de asignación de espacio para los archivos del área de datos. Cada entrada de la tabla representa un cluster del área de datos.
    - su función es mostrar la cadena de clusters en la que se aloja un archivo.
    - DESFRAGMENTACIÓN → reestructuración de la información.
  - DIRECTORIO
    - tabla con información de los archivos que indica su nombre y atributos.
    - para acceder a un archivo guardado, se accede al directorio y se lo ubica por su nombre de archivo en el disco y un número de cluster de comienzo.
  - NTFS
    - aprovecha mejor el espacio en disco
    - reduce la fragmentación de archivos } a comparación de FAT.
    - por cada archivo o directorio que se crea en el disco, se genera un registro en la tabla de archivo maestra (MFT). Si el archivo es tan pequeño que los datos y sus atributos entran en un registro, queda almacenado en la MFT y no requiere espacio adicional en el área de datos.

- **BUFFER DE DISCO** → memoria de almacenamiento temporal en la que se aloja un bloque de información del disco.
- **CACHÉ DE DISCO**
  - optimiza aún más los accesos al disco, asociándose al sistema.
  - buffer especial que determina qué sectores son los solicitados con mayor frecuencia, para retenerlos y agilizar los tiempos de búsqueda.
- **DISCO VIRTUAL** → disco ficticio creado en una parte de la memoria RAM, de manera que pueda emularse su actividad con la consecuente ganancia de tiempo.

10 INSTRUCCIONES

- La ejecución de una instrucción se completa en 4 pasos
  - búsqueda de la instrucción en memoria.
  - decodificación de la instrucción.
  - ejecución de la instrucción.
  - actualización del puntero de instrucción (para proceder a la ejecución de la próxima).
- Cuando se presenta el set de instrucciones que puede ejecutar el CPU, se hace abstracción del contenido binario y se indican qué "campos" componen la instrucción y cómo se interpretan. Esto se conoce como "Formato" y se puede representar así: | COP | DATA |, donde: COP → código de operación de la instrucción. DATA → dato por operar (que cuando no es una constante, puede obtenerse de memoria o de un registro).

FORMATO DE INSTRUCCIÓN

- Se refiere a la manera en que deben interpretarse los bits que constituyen el código de máquina de la instrucción.
- INSTRUCCIONES SIN DIRECCIÓN
  - representa al código de operación únicamente.
  - ejemplo: HLT, que indica a la CPU que debe detener la ejecución de las instrucciones del programa en curso y descontinuar su actividad.
- INSTRUCCIONES DE 1 SOLA DIRECCIÓN
  - | COP (8) | DATA1 (16) | → todas las instrucciones hacen referencia implícita (en el código de operación) al registro acumulador; por lo tanto, DATA1 hace referencia al dato afectado.
  - ejemplo: MOV AX, 0000h → transfiere el valor 0000 (en hexadecimal) al registro acumulador AX.
  - ejemplo: MOV AX, [FFFF] → también es de transferencia al registro acumulador, pero ahora el dato se encuentra en memoria y el código de instrucción hace referencia a su posición en ella.
    - los corchetes indican "contenido de" y FFFF es una dirección efectiva
- INSTRUCCIONES DE 2 DIRECCIONES → algunos diseñadores agregaron el campo DATA2 en algunas instrucciones, de modo tal que se utilice para referenciar al segundo operando. | COP | DATA1 | DATA2 |
- INSTRUCCIONES DE 3 DIRECCIONES → se agrega el campo DATA3 para referenciar el lugar donde se almacena el resultado. | COP | DATA1 | DATA2 | DATA3 |
- INSTRUCCIONES DE 4 DIRECCIONES → el DATA1 y el DATA2 son usados para referenciar los datos afectados por el código de operación. El DATA3 referencia el resultado y el DATA4 referencia la próxima instrucción por efectuarse. | COP | DATA1 | DATA2 | DATA3 | DATA4 |

MODOS DE DIRECCIONAMIENTO

- Es el código de operación de la instrucción el que define el modo de direccionamiento.
- DIRECCIONAMIENTO DIRECTO DE MEMORIA
  - la referencia al dato que especifica el campo DATA queda sin alteraciones.
  - el tiempo de captación del dato depende del tiempo de un acceso a memoria.
- DIRECCIONAMIENTO IMPLÍCITO
  - para obtener el dato, no se accede a memoria.
  - todas las instrucciones que asignen un valor mediante el COP pertenecen a esta categoría.
  - el código de operación no requiere fase de búsqueda del operando y, por lo tanto, su ciclo de instrucción es más corto.
- DIRECCIONAMIENTO INMEDIATO
  - involucran el dato en la instrucción en sí, en el campo DATA.
  - permiten asignar un valor constante a un registro o una posición de memoria.
  - no se requiere el acceso a memoria para obtener el dato.
  - no se puede modificar el valor del operando (al ser una constante) durante el transcurso de la ejecución del programa.
  - el operando se incluye como parte de la instrucción
- DIRECCIONAMIENTO INDIRECTO
  - el campo DATA contiene una dirección de una posición en memoria que, a su vez, contiene la referencia al dato.
  - para acceder al dato se requieren dos accesos de memoria.
- DIRECCIONAMIENTO DE LA CPU ASOCIADO A REGISTROS
  - cuantos menos accesos a la memoria se necesiten, mayor será la velocidad de procesamiento.
  - la CPU cuenta con varios registros internos, para que se optimice el tiempo de ejecución de los programas (REGISTROS GENERALES). Los mismos pueden estar conectados entre sí o mediante conexiones directas con compuertas de habilitación (REGISTROS INDEPENDIENTES).

- DIRECCIONAMIENTO DIRECTO POR REGISTRO → la referencia a un registro siempre es más rápida, puesto que no hay acceso a memoria y, por lo tanto, se observa una ganancia de tiempo en el acceso al dato.
- DIRECCIONAMIENTO INDEXADO → se utilizan algoritmos que involucran índices.
- DIRECCIONAMIENTO RELATIVO A LA BASE → se usan 2 registros
  - el BX (base).
  - el BP (puntero a base).
- DIRECCIONAMIENTO A UNA PILA
  - una pila es una estructura de datos a los que se accede según el criterio LIFO.
  - la CPU puede contar entre sus registros internos con un puntero a una pila de datos en memoria.
    el que lleva el control de las direcciones en la pila es el puntero. Su contenido direcciona la primera posición vacía de la pila y se actualiza por cada dato agregado y cada dato extraído. Los datos extraídos no desaparecen físicamente de la pila, sino que sus direcciones se liberan.
  - la pila sirve para almacenar el contenido de los registros de la CPU temporalmente.
  - para acceder a los datos de la pila, es aconsejable utilizar el puntero de base (BP).
  - Modos de direccionamiento del set de instrucciones 80x86:
    - REFERENCIA A REGISTRO → el operando está en la dirección del registro especificado en la instrucción.
    - REFERENCIA A MEMORIA → para acceder al operando se deben sumar: la dirección de segmento (que se multiplica por 16 antes de ser sumada), la dirección base, el índice y el desplazamiento.

Ver ejemplos de "antes y después de la ejecución" en las páginas 267-268

TIPOS VÁLIDOS DE INSTRUCCIÓN

- Los tipos de transferencias válidas son:
  - constante a registro — ejemplo: MOV AH, 1234h.
  - constante a memoria — ejemplo: MOV [0200], 6789h.
  - registro a registro — ejemplo: MOV AX, BX.
  - registro a memoria — ejemplo: MOV [012A], AX.
  - memoria a registro — ejemplo: MOV AX, [0203].
- Los operandos fuente y destino deben ser del mismo tamaño.
- Las constantes y los registros de segmento no pueden moverse a registros de segmento.
- Los operandos fuente y destino no pueden ser ambos posiciones de memoria.

11 SOFTWARE DEL SISTEMA

CLASIFICACIÓN DEL SOFTWARE DEL SISTEMA

- El software del sistema es el nexo entre las necesidades del usuario y las capacidades del hardware. Está integrado por: → software de base
  - controla y respalda, en cierto modo, el software de otras categorías, y todas ellas están íntimamente relacionadas en mayor o menor grado con el diseño del software, lo que las hace aptas para una computadora y sus compatibles.
  - su núcleo se denomina SISTEMA OPERATIVO.
  - la interfaz gráfica de usuario forma parte de él y su objetivo principal es crear un entorno organizado para el usuario y los utilitarios.

SISTEMA OPERATIVO

- es una colección de programas que administran la operación de una (o varias) computadora/s, con el fin de obtener un comportamiento eficiente. Es una plataforma software que asigna recursos y supervisa al resto de los programas que se ejecutan en la PC
- hay dos grandes componentes software en él
  - RESIDENTES → componentes que residen permanentemente en memoria principal durante todo el procesamiento.
  - TRANSITORIOS → residen sólo cuando se los necesita y están almacenados en memorias secundarias cuando no están en memoria principal.
- NIVELES DE ADMINISTRACIÓN DEL SISTEMA OPERATIVO
  - Administración del procesador y los procesos → el despachador es el módulo del SO que asigna el proceso al procesador
  - Administración de memoria → el SO asigna la memoria que un proceso requiere para su ejecución.
  - Administración de dispositivos de E/S.
  - Administración de archivos
    - supervisa la gestión de archivos para su creación, acceso y eliminación.
    - define la política que determina de qué forma serán almacenados físicamente.
- TIPOS DE SISTEMAS OPERATIVOS
  - MULTITAREA
    - son capaces de administrar procesos concurrentes.
    - permiten que tanto las instrucciones como los datos procedentes de varios programas residan al mismo tiempo en la memoria principal y eventualmente en el disco.
  - TIEMPO COMPARTIDO → tratan de administrar los recursos repartiéndolos equitativamente.
  - MULTIUSUARIO → permiten el acceso de varios usuarios desde distintas terminales administradas por el mismo sistema operativo.
  - TIEMPO REAL
    - tienen como objetivo proporcionar tiempos más rápidos de respuesta, ya que se utilizan para brindar soporte a sistemas tales como el control de transportes, el control de procesos industriales, los cajeros automáticos, etcétera.
    - sus acciones se deben ejecutar en intervalos de tiempo determinados por la dinámica de los sistemas físicos que controlan.

TRADUCTORES DE LENGUAJE

- son programas que convierten los programas escritos por el usuario en lenguaje simbólico (FUENTE) a lenguaje de máquina (EJECUTABLE).
- ENSAMBLADORES
  - software que traduce un archivo fuente escrito en un lenguaje Assembler a un archivo cuyas instrucciones están en código máquina, que es ejecutable directamente por el procesador para el que se creó.
  - los traductores se dividen en dos grupos en función de la relación entre lenguaje fuente y lenguaje máquina. El primer grupo traduce una instrucción de un lenguaje y genera una única instrucción de máquina, mientras que el segundo grupo lo componen lenguajes de alto nivel en los que una sentencia se traduce a varias instrucciones en código de máquina.
  - permiten generar tanto el código absoluto como el que permite reubicar los módulos.
- INTÉRPRETES
  - traductor de lenguaje que traduce una instrucción en lenguaje de alto nivel a lenguaje de máquina y, de ser correcta, la ejecuta de inmediato. Si hay un error sintáctico, lo señala e interrumpe la ejecución.
  - el programa se va probando a medida que se confecciona.
  - debe traducirse cada vez que se quiera ejecutar, aún cuando no haya habido modificaciones.
- COMPILADORES
  - traductor de lenguaje que traduce un programa escrito en lenguaje de alto nivel a lenguaje de máquina.
  - separa la traducción de la ejecución del programa, agilizando tanto una tarea como la otra.
  - la ejecución del programa sólo se realiza cuando la compilación terminó satisfactoriamente.
  - permite obtener el código de máquina del programa compilado.
  - permite hacer un resguardo del programa compilado en una memoria externa.

12 Dispositivos de Entrada/Salida

Discos Rígidos

- un disco está constituido por una base (que gira alrededor de un eje dispuesto en la unidad) y es accedido por una o más cabezas lectograbadoras.
- una cabeza registra los bits en círculos concéntricos denominados pistas.
- una pista es una división lógica (no física) de la superficie, producto de la acción de rotación del soporte y de la posición fija de la cabeza al momento de grabación o lectura.
- por efecto de rotación, se describen tantos cilindros concéntricos como pistas haya.
- por cada cara grabable, habrá una cabeza lectograbadora.
- CONTROLADORA DE DISCO → constituida por
  - CONTROLADORA DEL MOTOR DEL EJE Y BRAZO ACTUADOR → para asegurar que los platos giren a la velocidad correcta y que el actuador posicione las cabezas lectograbadoras en el lugar preciso sobre el plato.
  - CONTROLADORA DE INTERFAZ → para comunicarse con la CPU del sistema de transferencia adecuado.
  - UN MICROPROCESADOR → ejecuta los comandos propios del disco.
  - UNA MEMORIA ASOCIADA → almacena el bloque que se leyó o que se va a transferir.
- para ESCRIBIR información a una unidad de disco rígido, una corriente eléctrica (que fluye a través de una bobina en la cabeza lectograbadora) produce un patrón magnético en la cobertura de la superficie del medio, que se corresponde con los datos que se escriben.
- para LEER información de una unidad de disco rígido, la cabeza lectograbadora convierte los patrones magnéticos grabados en el medio a una corriente eléctrica, que se amplifica y procesa para reconstruir el patrón de datos almacenado.
- TIEMPOS DE ACCESO A DISCO
  - TIEMPO DE BÚSQUEDA → período que tarda el brazo en mover las cabezas lectograbadoras entre las pistas.
  - TIEMPO DE CAMBIO DE CABEZAS → período medio que le lleva a la unidad de disco cambiar entre dos de las cabezas cuando está leyendo o escribiendo datos.
  - LATENCIA ROTACIONAL → tiempo de espera que la cabeza debe esperar a que la unidad de disco gire el plato al sector correcto.
- TIEMPO DE ACCESO A LOS DATOS → tiempo que se tarda en posicionar una cabeza lectograbadora sobre una pista particular y encontrar el/los sector/es de interés dentro de esa pista para leer o escribir.
- luego de posicionar la cabeza lectograbadora, la unidad de disco está lista para leer/escribir datos desde/hacia el disco. Esto conlleva una transferencia de datos entre el disco y la memoria interna. Cuanto más rápida sea esa transferencia, menor será el tiempo que el proceso tenga sus datos disponibles.
- la VELOCIDAD DE TRANSFERENCIA DE DATOS es dependiente de la velocidad de transferencia del disco (rapidez con que pasan los datos desde el disco hacia el buffer o memoria controladora) y la velocidad de transferencia del host (rapidez con que la controladora pasa los datos a la memoria interna a la que accede la CPU).

Dispositivos de Almacenamiento Removibles

- DISCOS ÓPTICOS → combinan la tecnología de los medios magnéticos tradicionales con la tecnología de los discos ópticos.
- TARJETAS DE MEMORIA → consumen poca energía y son compactas, muy confiables y livianas.
- TARJETAS ROM y OTP.
- TARJETAS SRAM → los usuarios pueden escribir, modificar o borrar la información almacenada en ellas, pero estos chips deben estar alimentados por la energía constantemente.
- TARJETAS FLASH → combinan las mejores características de las SRAM como de las ROM/OTP.

13 TRANSFERENCIAS DE INFORMACIÓN

BUSES

- elementos de comunicación que relaciona cierta cantidad de componentes o de dispositivos.
- conjunto de conductores que transfieren señales eléctricas en forma pasiva, asociado con un hardware que regula su actividad (CONTROLADOR DE BUS).
  las señales que representan bits sobre el bus son digitales y pueden transferir el dato en < serie / paralelo.
- JERARQUÍA (desde los internos en un chip hacia afuera).
  - BUSES INTERNOS AL CHIP → conexiones internas que comunican registros en un chip, como por ejemplo las conexiones entre los registros internos de un microprocesador.
  - BUSES QUE CONECTAN CHIPS SOBRE UNA PLACA
    - los distintos módulos que albergan chips se comunican entre sí conectándose a una placa o tarjeta. Estos módulos se interconectan entre sí a través de buses (el cobrano los ve sobre la superficie de circuito impreso configurando pistas o caminos metálicos).
    - en general, realizan transferencias sincrónicas.
    - las señales del mensaje que constituyen el bus de datos son bidireccionales.
  - BUSES QUE CONECTAN DISTINTAS PLACAS
    - las placas se conectan entre sí mediante buses.
    - los diversos diseños de placas que provienen de distintos fabricantes crean la necesidad de compatibilizar la forma de transferir un dato.
    - son componentes de complejidad superior estandarizados e independientes, que se acoplan a los componentes del sistema hardware de la computadora.
  - BUSES DE ENTRADA/SALIDA → determinan una multiplicidad de estructuras llamadas ARQUITECTURA DE BUSES. Dicha arquitectura permite definir normas de comportamiento para la transferencia de datos desde o hacia los dispositivos de entrada/salida.

DISPOSITIVOS DE ENTRADA/SALIDA

- son todas las unidades periféricas entre sí y las "intermediarias", que se encargan de efectivizar una transferencia entre la memoria interna y la memoria externa en los periféricos.
  las funciones de un bus de E/S son:
  - comunicarse con el periférico y el sistema CPU-memoria.
  - controlar la temporización durante la transferencia.
  - almacenar temporalmente bits para "paliar" la diferencia de velocidad entre emisor y receptor.
  - detectar si se produjeron errores durante la transferencia.
- cuando la actividad del bus de E/S es sincrónica → se usan señales de clock que regulan la transferencia.
  asincrónica → las señales del clock no regulan su operación.
- CONTROLADORES
  - dispositivos asociados directamente al periférico, y está constituido por: un buffer interno (que permite el almacenamiento de la información que viaja desde/hacia el soporte) y una lógica de control (que interpreta comandos de periférico, genera señales para su ejecución y gobierna la unidad).
  - sus funciones son:
    - aislar el software de servicio de entrada/salida (que se ocupa de la transferencia de los detalles específicos del hardware del periférico y los convierte en inocuos).
    - compatibilizar la velocidad del periférico respecto de la del resto del sistema.
  - cuando un controlador se involucra con una transferencia, la CPU recibe parámetros de un programa que ejecuta; a través de una interfaz, este programa forma parte de la administración de entrada/salida y se denomina MANEJADOR DE DISPOSITIVO (DEVICE DRIVER). Los DRIVERS establecen la secuencia lógica de comandos y al controlador los reconoce y activa los mecanismos necesarios para su ejecución.
- ADAPTADORES
  - pueden residir en una placa del sistema, integrado en un chip.
  - proveen funciones para conectar y lograr la operación de un componente conectado a un bus.
  - las unidades que los constituyen son:
    - el controlador de video.
    - el procesador de video.
    - el conversor digital-analógico.
- PUERTO
  - área de almacenamiento alojada en una interface.
  - permite la comunicación de un periférico con la memoria para enviar/recibir una secuencia de bits.

- **INTERFACES**
  - hardwares que actúan de nexo entre un periférico o un adaptador y el bus.
  - sirven para adecuar las señales
    - preparar la transferencia elemental basada en un protocolo.
  - **PARALELA**
    - dispositivo hardware que permite el control de la transferencia en paralelo entre el bus de sistema y un periférico.
    - la interfaz cuenta con registros llamados PORTS, cuya función es lograr la transferencia elemental
  - **SERIE**
    - permite el control de la transferencia de bits en serie entre el bus y un dispositivo de E/S.
    - está asociado con una lógica de direccionamiento que permite establecer que esa interfaz fue seleccionada por el procesador para la transferencia.
    - los registros de la interfaz constituyen el PUERTO SERIE.
- **CANALES O PROCESADOR E/S**
  - procesadores dedicados a controlar las transferencias de E/S. sin intervención de la CPU en la ejecución del software de E/S.
- **TRANSFERENCIAS DE E/S**
  - aspectos para resolver durante una transferencia:
    - sincronizar los tiempos de transferencia entre la CPU-memoria y el periférico.
    - decodificar los bits que identifican al dispositivo.
    - convertir un mensaje serie a paralelo, o al revés.
    - convertir el mensaje enviado de un formato a otro.
    - convertir el mensaje enviado de un código a otro.
    - controlar que el mensaje enviado se reciba de manera correcta.
    - decodificar un comando.
    - controlar las banderas de estado.
  - **MAESTRO**
    - dispositivo que tiene el control del bus en un momento determinado
    - puede enviar señales de control, dirección y dato sobre el bus.
    - conoce la dirección del emisor y el receptor.
  - **ESCLAVO**
    - puede pedir un servicio de transferencia, pero no iniciarla.
  - la estrategia de control del bus depende de la arquitectura diseñada para el sistema (ARBITRAJE DEL BUS).
- **DRIVERS**
  - son programas que "conocen" el dispositivo periférico.
  - cada uno de ellos actúa como un receptor de requerimientos de otros programas, que pertenecen a otro nivel y desconocen las peculiaridades de cada uno de los distintos dispositivos externos.

MODALIDADES DE ENTRADA/SALIDA

- El sistema operativo cuenta con programas que gestionan las transferencias de entrada/salida y realizan funciones comunes a todos los dispositivos periféricos.
- Desde el punto de vista de un programa de aplicación, un proceso que ejecuta la CPU actualmente puede ir escribiendo datos en un "área de almacenamiento" hasta completar el bloque, momento en el que se llama al servicio para realizar la transferencia completa del bloque.
- Los programas de aplicación llaman a los programas de E/S del sistema operativo mediante una "llamada al sistema". La relación entre el programa de aplicación en estado de ejecución (como peticionario del pedido de transferencia) y el hardware de E/S (como su proveedor) es la siguiente: APLICACIÓN → PROGRAMA DE E/S → DRIVER → HARDWARE
- Para llevar a cabo la transferencia completa, que involucra varias transferencias elementales (varios mensajes que emergen de un emisor a un receptor), se pueden identificar varios métodos:
  - **TRANSFERENCIA CONTROLADA POR PROGRAMA**
    - la CPU ejecuta el programa de E/S y se comunica mediante una interfaz.
    - la CPU controla el acceso a memoria para ubicar el dato, a través de sus registros internos, y verifica el estado de la interfaz a través del puerto de control en forma continua.
  - **TRANSFERENCIA INICIADA POR INTERRUPCIÓN**
    - implica que la interfaz (con la que se comunica la CPU) genere un aviso que indique que está preparada para transferir. En tal caso, la interfaz provoca una interrupción.
  - **TRANSFERENCIA CON ACCESO DIRECTO A MEMORIA**
    - las interfaces pueden conectarse a la memoria a través del controlador de acceso directo a memoria (en inglés, DMA). Estos DMAs están asociados a dispositivos rápidos que transfieren la información en bloques de bytes.
    - la CPU sólo se encarga de indicarle (a los DMAs) la cantidad de palabras a transferir, la posición inicial de la palabra en memoria interna y la dirección del dispositivo como parámetros.
    - la técnica consiste en relacionar la funcionalidad DMA con el bus que conecta la CPU con la memoria principal, y que ese controlador establezca la relación memoria-interfaz sin intervención de la CPU.
  - **TRANSFERENCIA A TRAVÉS DE UN PROCESADOR IOP o MODO CANAL**