

Niveles de organización de computadoras

Índice General

1.1	Introducción	53
1.1.1	Hardware y software. Principio de organización de computadoras	54
1.2	Enfoque de máquina multinivel	54
1.3	Obtención de programas	55
1.4	Clasificación de computadoras	57
1.5	Generaciones de computadoras	58

1.1 Introducción

202. La *informática* es el campo del conocimiento que se ocupa del estudio, diseño y uso de sistemas (esquemas) de tratamiento automático de información. Para ésto, el informático se vale de medios físicas (*computadora y calculadora*¹) que consiguen hacer funcionar según un programa; el resultado es el procesamiento de la información.

203. Informalmente se define programa como la expresión de un algoritmo en un lenguaje de programación, y *algoritmo* como una forma de abordar un problema

¹Se entiende usualmente como calculadora, una máquina capaz de aplicar algoritmos aritméticos, mientras que la computadora permite aplicar algoritmos de corte lógico y matemático, mediante lenguajes de programación habituales.

con el fin de dar solución a éste. Siendo esta definición superficial, nos sugiera que la diferencia entre programa y algoritmo es la forma de expresión.

programa

Un *programa* es una secuencia de instrucciones conforme a un lenguaje comprensible por la computadora, al ejecutarse sobre datos de entrada genera otros nuevos, de salida. Los datos de salida serán o no la solución al problema, pues la definición no indica que el programa tenga que ser correcto. La definición de algoritmo es más exigente, lo cual no quiere decir que la forma idónea de expresar un algoritmo no sea empleando un lenguaje de programación.

proceso

204. La forma más ortodoxa de definir *proceso* es “un programa en ejecución”. Tiene entidad física real y encuentra parangón en las reacciones químicas, procesos industriales, etc. Visto así, una computadora resulta ser una máquina que procesa datos conforme a un programa.

1.1.1 Hardware y software. Principio de organización de computadoras

hardware

205. Con el fin de sentar unas definiciones precisas nos vemos obligados a definir *hardware* como: objetos de un proceso informático que pueden describirse, propiamente, mediante parámetros y expresiones físicas. También se denomina soporte físico.

software

Su complemento es el *software*: objetos de un proceso informático que pueden describirse, propiamente, mediante expresiones lógicas. También se denomina soporte lógico o logicial.

firmware

206. Entre ambos términos existen expresiones intermedias como *firmware*: software incorporado, al soporte físico. Este sustantivo hace alusión al software que la empresa vendedora incorpora al hardware, de mano.

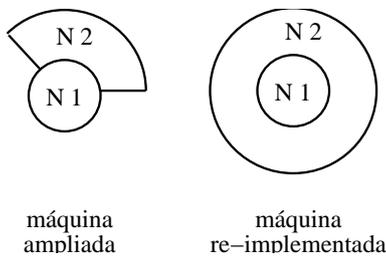
207. No siempre hay una forma clara de calificar los elementos del procesamiento y desde un punto de vista abstracto funcional *hardware y software son equivalentes*. Esto funciona como un principio de organización de computadoras, y viene a decir tanto que puede construirse físicamente (p.ej.: con circuitos electrónicos) cualquier programa, como que cualquier computadora puede implementarse partiendo de algún modelo de cómputo, y un programa adecuado². En la práctica, los obstáculos relativos a velocidad y complejidad de diseño pueden ser insalvables.

1.2 Enfoque de máquina multinivel

208. La programación es una tarea complicada y costosa. La realización de aplicaciones partiendo del nivel físico se vé facilitada si se emplea un enfoque de máquina multinivel. Según éste, escribimos programas que simulen *máquinas virtuales* partiendo de algún nivel de máquina; se acumulan tantas etapas como sea necesario

²Recuérdese que una máquina de Turing universal es una plataforma en la que se puede expresar cualquier otra máquina de Turing.

hasta que el problema que se quiere resolver se exprese fácilmente en alguna de ellas.



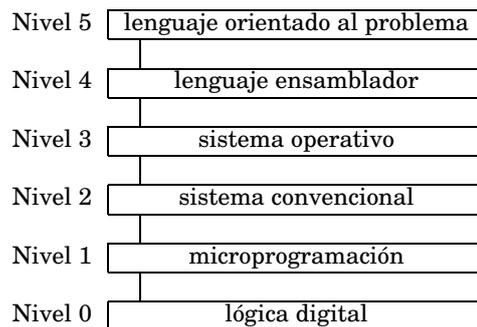
Así, una de las tareas de un programador consiste en construir máquinas virtuales apoyadas en el nivel del hardware, hasta que sea posible escribir programas en el nivel de abstracción adecuado.

209. Estos niveles se pueden construir (a) añadiendo más funciones a las aportadas por el nivel de máquina inferior (máquina ampliada), o bien (b) ocultando total o parcialmente las funciones del nivel

inferior y reimplementando el nivel de máquina.

210. Cuando esta reimplementación tiene como fin suplantar (simular) un nivel de máquina virtual correspondiente a otro nivel inferior de otra máquina, se suele denominar *emulación*. Es habitual cuando se desea ejecutar programas sobre una máquina, que en principio están escritos para ejecutarse sobre el nivel físico de otra máquina diferente.

211. Este esquema conceptual, nos permite descomponer el estudio y diseño de las computadoras de una forma sencilla, pues se puede estudiar la computadora como un edificio compuesto de varios pisos, en cada uno de ellos se emplea un **lenguaje** y unas **reglas** posiblemente diferentes, aquellas más apropiados al nivel (máquina virtual) de estudio.



212. En este esquema merece una mención especial el nivel 2, denominado nivel de sistema convencional o nivel de máquina convencional (NMC), lo que expresa que es éste el nivel el que se entiende normalmente que la computadora, despojada de cualquier software.

Figura 1.1: Esquema de máquina multinivel.

1.3 Obtención de programas

213. Para ejecutar cada programa en su nivel de máquina virtual, hay dos aproximaciones:

- *traducción*: Se toma el programa escrito en el lenguaje actual y se substituye por un programa equivalente escrito en el lenguaje que soporta el nivel de máquina virtual actual; hecho ésto, se ejecuta en este nivel inferior.

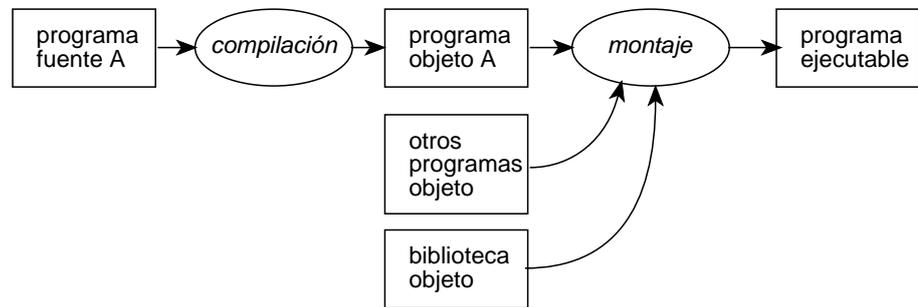


Figura 1.2: Procedimiento de compilación.

interpretación

- *interpretación*: Se toma cada sentencia del programa escrito en el lenguaje actual y se ejecutan las sentencias equivalentes del nivel de máquina inferior al actual.

El lenguaje del nivel de máquina actual, y el del nivel inferior no deben ser demasiado diferentes. Caso contrario, la experiencia demuestra que es preferible introducir niveles intermedios, por ejemplo, lenguaje NMC³→Prolog.

214. Traducción e interpretación se diferencian en que mientras en el primero hay una fase previa de traducción –tras la cual desaparece el nivel de máquina virtual actual– en el segundo existe un proceso que emula el nivel de máquina virtual actual. Para este proceso emulador (el intérprete) el programa a ejecutar es parte de los datos⁴.

programa fuente
programa objeto

215. En la traducción de programas (también denominada compilación), cierto proceso se encarga de traducir sentencias del programa original (*programa fuente*) en sentencias programa objetivo (*programa objeto*), ejecutándose posteriormente este último. No tiene por qué haber correspondencia una a una entre instrucciones de ambos lenguajes.

216. Esta estructura es suficientemente flexible para que en la realidad haya modificaciones importantes que aparecen en la figura 1.2. Se emplea un lenguaje intermedio muy próximo al lenguaje del nivel de máquina virtual inferior, de modo que se pueda incluir en este momento, bibliotecas con módulos de programas y nuevas funciones, a la vez que trozos independientes del programa escritos previamente.

montador

217. El programa *montador* (*linker*) se encarga de resolver las ligaduras de nombres y direcciones simbólicas que puedan existir entre los diferentes fragmentos de programa provenientes de otros compiladores y de bibliotecas con rutinas ya cons-

³Warren Abstract Machine

⁴Aparecen dos conjuntos de datos de entrada: los datos del problema, propiamente hablando, y el programa que se ejecuta sobre los datos, que es un dato del intérprete.

	l. palabra	Velocidad	Memoria	N. usr.	Ejemplo
Supercomputador	64 - 128 bits	x000 MIPS	x0 GB	-	CRAY
Computadora grande (Mainframe)	32 - 128 bits	<100 MIPS	x00 MB	x00	IBM/3090
Minicomputadora (Mini)	32 - 64 bits	x0 MIPS	<200 MB	< 200	VAX/8350
Estación de trabajo (Workstation)	32 - 64 bits	x0 MIPS	<200 MB	1	Apollo
Computadora perso- nal (PC)	16 - 32 bits	x MIPS	x0 MB	1	Compaq
Computadora de jue- gos (Videoconsola)	16 - 64! bits	0.1-x MIPS	x0 Mbits	1	Nintendo64

Tabla 1.1: Clasificación de computadoras (1998 d.C.).

truidas, en definitiva, enlaza los diferentes fragmentos para producir el programa ejecutable en el nivel de máquina adecuado (típicamente éste es el NMC).

1.4 Clasificación de computadoras

218. Entre las computadoras se distingue entre las de propósito específico y las de propósito general. Las primeras se construyen bien atadas a una aplicación muy concreta, como por ejemplo: el guiado de un cohete, el control un robot, una caja registradora, etc; o bien para un rango concreto de aplicaciones, como puede ser: computadoras de cálculo científico, computadoras vectoriales, masivamente paralelas, computadoras de transacciones etc.

219. En el extremo opuesto se encuentran las computadoras de propósito general, que se construyen pensando en un rango de aplicación amplio: gestión de almacenes, procesamiento de textos, bases de datos, etc. Como es lógico, no proporcionan los mismos rendimientos que las computadoras de propósito específico en cada tarea, pero su adquisición, mantenimiento y operación es más económica y son más versátiles.

220. También es posible clasificar las computadoras en función de un conjunto de aspectos como: velocidad, número de usuarios habitual, precio, etc. De esto resulta un conjunto de familias, o segmentos de mercado, de naturaleza más bien borrosa, pero generalmente admitida. De ella se muestran algunos detalles en la tabla 1.1.

221. Términos como “supercomputador” se refieren a computadoras de uso científico o ingenieril, para centros de cálculo. El término “computadora grande” se refiere a centros de cálculo de bancos y corporaciones. Es cierto que hoy en día las diferencias entre computadora personal y estación de trabajo se difuminan, entre sí, y con las minicomputadoras si es el caso que se emplean sistemas operativos multiusuario. Las estaciones de trabajo son computadoras dedicadas, típicamente a un usuario, para tareas de diseño y programación que consumen muchos recursos,

sobre todo de índole gráfico.

Al mismo tiempo, las computadoras de videojuegos multiplican periódicamente sus prestaciones, acercándose poco a poco a las estaciones de trabajo.

222. La velocidad de las computadoras se suele medir empleando ciertos tests de velocidad, que incluyen: programas reales, benchmarks⁵ específicos, núcleos de programas y benchmarks sintéticos (mezcla factores de velocidad). El término MIPS es una medida posible de velocidad de cómputo, no la mejor, que mide el número de millones de operaciones que se completan en un segundo, sin distinguir el tipo de operación. MFLOPS mide el número de millones de operaciones en punto flotante, por segundo.

1.5 Generaciones de computadoras

223. Los orígenes de la informática se remontan al primer uso de algoritmos para resolver de forma sistemática problemas como la suma, el producto o el cálculo del máximo común divisor (algoritmo de Euclides s.IV a.C.). La palabra cálculo (cómputo) proviene del latín *calculus-i* que significa piedra, un elemento común para realizar operaciones aritméticas de donde proviene el ábaco, empleado aun hoy en día.

Previo a la era electrónica de computadoras, como las que se conocen hoy en día, se encuentra la era mecánica y la era electromecánica.

224. Con anterioridad al siglo XX se construyen dispositivos como la regla de cálculo de Napier, la máquina aritmética de Pascal, capaz de realizar las cuatro reglas aritméticas, y los diseños de máquinas de Babbage, con su máquina diferencial (cálculo numérico) y su máquina analítica (de propósito general).

225. Desde principios del siglo XX se sucede el empleo de relés eléctricos para el diseño de circuitos complicados, entre los que se incluyen los autómatas de Torres Quevedo hasta las computadoras de K. Zuse. Por su tamaño y complejidad no logran imponerse a las calculadoras mecánicas de la época, utilizadas en su mayoría para el cálculo balístico.

226. Con el advenimiento de los estudios de Shannon y el empuje de la II G.M. se construyen las primeras computadoras electrónicas (Colossus, E.N.I.A.C., E.D.V.A.C. ...). Para poder clasificar los avances producidos se habla de *generaciones de computadoras* donde ciertos hitos tecnológicos marcan avances cualitativos en el desarrollo de la informática.

generaciones de compu-
tadoras

1^a (1946-1955). Computadores basados en válvulas de vacío que se programaban en lenguaje máquina o en lenguaje ensamblador.

2^a (1953-1964). Computadores de transistores. Evolucionan los modos de direccionamiento y surgen los lenguajes de alto nivel.

⁵Programas destinados a medir la velocidad de computadoras

3^a (1964-1974). Computadores basados en circuitos integrados y con la posibilidad de trabajar en tiempo compartido.

4^a (1974 -) Computadores que integran la CPU en un solo circuito integrado (microprocesadores). Comienzan a proliferar las redes de computadores.

227. Las tendencias actuales de la informática se decantan en dos vías: Por una parte en el software, aplicaciones distribuidas, inteligencia artificial, recuperación de información; por otra parte en el hardware proceso paralelo, redes de alta velocidad.

Nivel de lógica digital y sistemas digitales

Índice General

2.1 Sistema digital	62
2.1.1 Información analógica e información digital.	62
2.1.2 Sistemas digitales	62
2.2 Álgebra de Conmutación	63
2.2.1 Álgebra de Boole	63
2.2.2 Álgebra de Conmutación.	64
2.2.3 Forma normal y teorema de Shannon	65
2.3 Circuitos combinacionales básicos	66
2.3.1 Codificadores y decodificadores	66
2.3.2 Multiplexores y demultiplexores	67
2.4 Circuitos secuenciales básicos	67
2.4.1 Biestables y registros	68
2.4.2 Bus	69
2.4.3 Contador	70
2.5 Controladores	70

228. El nivel de lógica digital es el nivel de máquina más cercano a la física, que estudia la informática. Por debajo de él se encuentra la disciplina de la electrónica de circuitos. El contexto de la lógica digital son las expresiones booleanas y el álgebra de conmutación.

2.1 Sistema digital

2.1.1 Información analógica e información digital.

229. La información se encuentra en el entorno humano en las magnitudes físicas de los objetos que nos rodean, tanto en un valor instantáneo como en la variación o cambio de estas magnitudes. Estas magnitudes incluyen desde la forma espacial, campos eléctricos, temperatura, luminancia, velocidad, etc.

230. Las magnitudes físicas suelen utilizarse bajo la forma de números reales, se dice entonces que estamos frente a *información analógica*. La variación del dial de un amperímetro de aguja, un barómetro de mercurio, un termómetro de alcohol proporcionan magnitudes reales analógicas.

información discreta

Cuando cierta medición física puede tomar un valores de un conjunto finito decimos que se presenta *información discreta*. El número de vueltas de una rueda, la cantidad de personas en un ascensor, el número de moléculas en un volumen son magnitudes discretas.

discretización

231. Las magnitudes discretas se obtienen directamente por la propia naturaleza de la medición, o mediante un proceso de *discretización*¹ de la señal analógica. Cuando se da cierta medida de longitud en centímetros, se discretiza por el número de centímetros más cercano. Esto se denomina discretización en magnitud.

Ciertas magnitudes físicas, además presentan una evolución temporal, si esta función del tiempo se mide a intervalos, se pasa de una función continua a una sucesión discreta. Esto se denomina discretización en tiempo o muestreo temporal.

232. Por último, estas medidas de magnitudes son susceptibles de ser representadas simbólicamente, y codificadas en cierto lenguaje con un alfabeto arbitrario.

2.1.2 Sistemas digitales

233. Un *sistema digital* es un dispositivo físico destinado a la generación, procesamiento, transmisión y almacenamiento de información, representada por magnitudes físicas discretas, tanto en número como en dimensión.

sistema digital binario

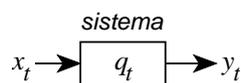
234. Un *sistema digital binario* es un sistema digital donde las medidas involucradas se representan con un alfabeto bivaluado.

sistemas multivaluados

Hay también *sistemas multivaluados*, en ellos el alfabeto de representación es multi-valuado.

sistema combinacional

235. Un *sistema combinacional* es aquél en el que la respuesta del sistema (salida) es función del estado (configuración interna), y éste es función solo de la entrada del sistema.



$$(2.1) \quad y_t = (g \circ f)(x_t) \quad \{ y_t = g(q_t) \quad q_t = f(x_t)$$

sistema secuencial

Figura 2.1: Sistema: entrada, estado y salida. Un *sistema secuencial* es aquél en el que la respuesta del

¹En inglés *quantization*.

sistema es función del estado del sistema y este es función de la entrada en instantes anteriores y el estado anterior.

$$(2.2) \quad y_t = g(q_{t-1}, x_t) \quad \{ y_t = g(q_{t-1}, x_t) \quad q_t = f(q_{t-1}, x_t)$$

donde q_t , x_t y y_t son el estado, la entrada y la salida, respectivamente, en el momento t .

236. Mientras que la expresión (2.1) es sencilla y directa, en la expresión (2.2) aparece un término de realimentación $q_t = f(q_{t-1}, x_t)$ que obliga a tener en cuenta el estado anterior del sistema. Se dice que estos sistemas son *sistemas con memoria*. A pesar de ello y como se verá, se puede construir empleando sistemas combinatoriales y realimentaciones de estado.

2.2 Álgebra de Conmutación

2.2.1 Álgebra de Boole

237. Previo al estudio del Álgebra de Conmutación se suele incluir el álgebra de Boole. Un álgebra de Boole es una tripleta $B = \langle B, +, \cdot \rangle$ donde B es un conjunto y $+$ y \cdot son dos operaciones internas a B y que cumple las siguientes condiciones:

1. B tiene al menos dos elementos, diferentes por supuesto.
2. Existe un único elemento neutro de $+$, llamémosle 0 .
Existe un único elemento neutro de \cdot , llamémosle 1 .
3. Tanto $+$ como \cdot son conmutativas.
4. Tanto $+$ como \cdot son asociativas..
5. Tanto $+$ como \cdot son recíprocamente distributivas.
6. $\forall x \in B$ existe un único elemento opuesto con respecto a $+$, que coincide con el elemento opuesto con respecto a \cdot , llamémosle \bar{x} .
Al elemento \bar{x} se le denomina “*complemento de x*”.

238. Esta estructura algebraica es similar a la lógica de proposiciones, aunque su planteamiento se hace desde el punto de vista del álgebra. Las propiedades que exhibe quedaron reflejadas en la sección ?? y aquí solo se mostrarán los teoremas fundamentales.

1. El álgebra de Boole respeta el Metateorema de la Dualidad, de $0 \leftrightarrow 1$, $+$ \leftrightarrow \cdot y $x \leftrightarrow \bar{x}$, de donde se sigue el teorema de "De Morgan".
2. $\forall a \in B : \quad a + a = a \quad a \cdot a = a$ (idempotencia).
3. $\forall a \in B : \quad a + 1 = 1 \quad a \cdot 0 = 0$ (absorción por e.n.).

4. $\forall a \in B : \overline{\overline{a}} = a$ (complemento sucesivo).

5. $\overline{1} = 0 \quad \overline{0} = 1.$

6. $\forall a, b \in B : \overline{a + b} = \overline{a} \cdot \overline{b} \quad \overline{a \cdot b} = \overline{a} + \overline{b}$ (De Morgan).

7. $\forall a, b \in B : a + (\overline{a} \cdot b) = a + b \quad a \cdot (\overline{a} + b) = a \cdot b$ (absorción).

239. El álgebra del Boole mínima es $\mathcal{B} = \langle \{0, 1\}, +, \cdot \rangle$. $+$ y \cdot se construyen de forma unívoca para respetar las leyes anteriores.

2.2.2 Álgebra de Conmutación.

variable de conmutación

240. Una *variable de conmutación* es un símbolo que puede representar de modo indeterminado un valor del conjunto $C = \{0, 1\}$. Típicamente se escogen las letras x, y, z, \dots

241. Una función de conmutación f de n variables es

$$\begin{array}{ccc} C^n & @ << f < < C \\ f : \mathbf{x} & \mapsto & f(\mathbf{x}) = y \end{array}$$

dato $\mathbf{x} \in C^n$ y $y \in C$. Dado n existen $VR_2^{VR_2^n} = 2^{2^n}$ funciones de conmutación diferentes.

álgebra de conmutación

242. La tripleta $\mathcal{F} = \langle F, +, \cdot \rangle$, siendo F el conjunto de todas las funciones de conmutación de n variables es un álgebra de Boole, y se denomina *álgebra de conmutación*. Dos funciones son iguales si cumplen la regla de equivalencia de Leibnitz descrita en la sección ??.

243. La operación $+$ (**O** o **OR**) se define: $\forall f, g \in F, \forall \mathbf{x} \in C^n, f + g \in F$:

$$(f + g)(\mathbf{x}) = \begin{array}{ll} 0 & f(\mathbf{x}) = g(\mathbf{x}) = 0 \\ 1 & \text{de otro modo} \end{array}$$

244. La operación \cdot (**Y** o **AND**) se define: $\forall f, g \in F, \forall \mathbf{x} \in C^n, f \cdot g \in F$:

$$(f \cdot g)(\mathbf{x}) = \begin{array}{ll} 1 & f(\mathbf{x}) = g(\mathbf{x}) = 1 \\ 0 & \text{de otro modo} \end{array}$$

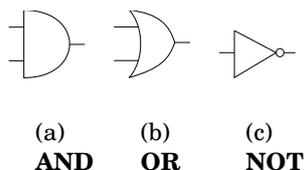
245. El complemento (**NO** o **NOT**) se define: $\forall f \in F \forall \mathbf{x} \in C^n, \overline{f} \in F$:

$$\overline{f}(\mathbf{x}) = \begin{array}{ll} 1 & f(\mathbf{x}) = 0 \\ 0 & f(\mathbf{x}) = 1 \end{array}$$

246. Finalmente se define la función $0 = 0^n$, como aquella que asigna a cada entrada posible el valor 0, y la función $1 = 1^n$, como aquella que asigna a cada entrada posible el valor 1.

Existen otras funciones lógicas útiles de dos variables, como la función “o exclusiva” (**XOR**), “no o” (**NOR**) o “no y” (**NAND**). Estas y cualquier otra se puede transcribir en función de las tres primeras funciones.

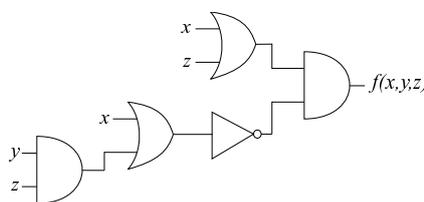
247. Para transcribir funciones de conmutación se puede emplear la sintaxis del álgebra de Boole o, dado que es una función finita, mediante la enumeración en forma de tabla de la salida en función de la entrada. Esta representación es especialmente útil en el momento de diseñar la función.



También se representan en forma de diagrama, donde se simboliza cada función primitiva, denominada también *puerta lógica*, como un componente cuyas entradas se representan como conexiones a la izquierda del símbolo y la salida en la parte derecha.

puerta lógica

248. Estos diagramas se inspiran en representaciones de circuitos electrónicos. En la práctica, la *electrónica digital* obtiene componentes físicos que se comportan de forma análoga a las funciones de conmutación. Cada puerta lógica primitiva se puede describir en términos de condensadores y transistores, y se pueden interpretar los potenciales en puntos del circuito como variables lógicas.



electrónica digital

2.2.3 Forma normal y teorema de Shannon

249. *Funciones minterm*: dado un número entero I , con $0 \leq I \leq 2^n - 1$, y su expresión en base 2 con n bits (i_1, \dots, i_n) , se define la función minterm

Funciones minterm

$$m_I(x_1, \dots, x_n) = \begin{cases} 1 & x_1 = i_1, \dots, x_n = i_n \\ 0 & \text{de otro modo} \end{cases}$$

Funciones maxterm: dado un número entero I , con $0 \leq I \leq 2^n - 1$, y su expresión en base 2 con n bits (i_1, \dots, i_n) , se define la función minterm

Funciones maxterm

$$M_I(x_1, \dots, x_n) = \begin{cases} 0 & x_1 = i_1, \dots, x_n = i_n \\ 1 & \text{de otro modo} \end{cases}$$

250. Se dice que una función de conmutación está en *forma normal disyuntiva* si está representada como una función **OR** de minterms (también se dice que es una suma de minterms).

De la misma forma, está en *forma normal conjuntiva* si está representada como una función **AND** de maxterms (también se dice que es un producto de maxterms).

251. El *teorema de Shannon* en su forma disyuntiva dice que toda función de conmutación de n variables, $f(x_1, \dots, x_n)$, puede expresarse como una suma única de minterms.

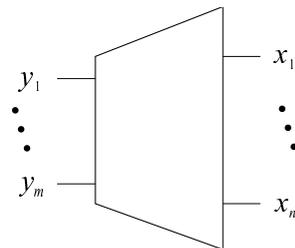
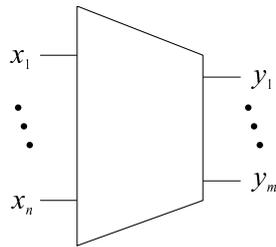
Si además, ordenamos los minterms numéricamente, la representación anterior es canónica. Es decir, podemos comparar funciones de conmutación reduciéndolas a su forma canónica y comparando.

252. Escribir una función de conmutación, partiendo de su representación tabular es una tarea sencilla. Para cada posible entrada tenemos un minterm asociado, luego en la suma de minterms se incluirán aquellos que lleven asociado un 1 en la salida (el equivalente en forma normal conjuntiva es ligeramente más complicado). La expresión resultante puede ser larga, pero correcta; existen métodos para minimizar expresiones de este tipo, como las “tablas de Karnaugh” o el método de Quine-McCluskey, entre otras.

2.3 Circuitos combinacionales básicos

253. Existe variedad de herramientas de diseño de circuitos lógicos secuenciales y combinacionales, la mayoría de ellos se apoyan en bibliotecas de componentes. Entre los componentes combinacionales básicos se encuentran el codificador, descodificador, multiplexor y demultiplexor. El alcance de estos componentes sobrepasa el campo del diseño digital y existen símiles en estructuras de programación.

2.3.1 Codificadores y descodificadores



254. Dado un alfabeto Σ con n símbolos $\{x_1, \dots, x_n\}$, un codificador es un dispositivo combinacional con n entradas (una para cada símbolo) y m salidas, tal que en un instante cualquiera, solo una de las entradas puede tomar el valor 1 (las restantes tomarán el valor 0); el codificador genera como salida una combinación de m bits que es única para cada entrada al valor 1, este patrón de bits es la codificación de ese símbolo de entrada según cierto código arbitrario.

255. Por extensión, se llama también también codificador a cualquier dispositivo que efectúa una traducción de símbolos según cierto código, a otro código destino.

Existen otros dispositivos, como los codificadores de prioridad, que presentan variantes con respecto al descrito arriba, pero que no se estudian aquí.

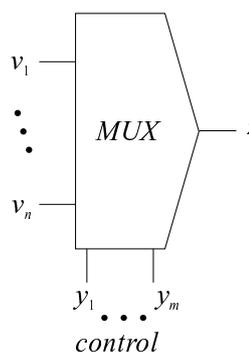
256. El descodificador funciona de modo complementario al codificador, de modo que dado un alfabeto Σ con n símbolos $\{x_1, \dots, x_n\}$, el descodificador con m entradas (una para cada bit del código) y n salidas (una para cada símbolo del alfabeto), en un instante cualquiera, proporciona un 1 por una de sus salidas (las restantes tomarán el valor 0); este 1 identifica al símbolo correspondiente a la codificación recibida en la entrada.

257. También por extensión, se denominan descodificadores a los dispositivos que efectúan cierta traducción de símbolos en cierto código, a otro código que suele ser

la representación natural de la información. En programación, la estructura de control análoga al descodificador es la selección múltiple.

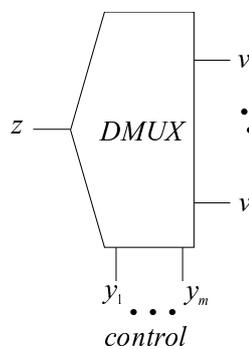
2.3.2 Multiplexores y demultiplexores

258. Sea un alfabeto Σ con n símbolos $\{x_1, \dots, x_n\}$ codificado en m bits $\{y_1, \dots, y_m\}$ y un conjunto de variables de conmutación $\{v_1, \dots, v_n\}$ con una correspondencia uno a uno con los símbolos de Σ . El multiplexor recibe un símbolo a la entrada y cada una de las variables de conmutación y propaga a su única salida aquella variable de entrada que concuerda con el símbolo recibido. Como es habitual, este símbolo se recibe en forma de código binario, para ahorrar conexiones, por lo cual, cada multiplexor conlleva internamente un descodificador.



En otras palabras, conecta una de las variables de entrada, con la salida, en función de un código de control. Esto permite muestrear cada variable de entrada mediante un código. Por extensión, se denomina multiplexor a cualquier dispositivo que permite compartir una línea de comunicación por varios dispositivos de entrada concentrando en una las conexiones de entrada. Típicamente, esta compartición puede ser por multiplexación en tiempo o en frecuencia.

259. Sea un alfabeto Σ con n símbolos $\{x_1, \dots, x_n\}$ codificado en m bits $\{y_1, \dots, y_m\}$, y un conjunto de variables de conmutación $\{v_1, \dots, v_n\}$ con una correspondencia uno a uno con los símbolos de Σ . El demultiplexor actuará como un componente complementario al multiplexor, tiene una variable de conmutación como entrada y, en función del símbolo de control recibido, propaga ésta entrada a aquella variable de conmutación de salida que corresponda. Este símbolo se recibe en forma de código binario, para ahorrar conexiones.



En programación, se puede parangonar la figura del demultiplexor a la selección múltiple, en cuanto a que dirige el flujo del programa hacia aquel fragmento de código que resulta de decodificar cierto valor entero.

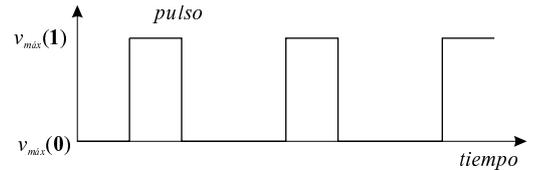
2.4 Circuitos secuenciales básicos

260. Los circuitos secuenciales se distinguen de los circuitos combinatoriales en que alguna variable de conmutación que es el resultado de cierta función, interviene también como entrada de esta función. Este tipo de “conexión” se conoce como *realimentación* (retroalimentación) de entrada o *“feedback”*.

realimentación

Físicamente, esto se traduce en que frente a cierta entrada, el sistema evoluciona en los instantes siguientes, presentando un comportamiento que es función del tiempo. Si la salida se estabiliza, el sistema combinacional es estable frente a cierta entrada, si no oscilará, y el sistema es inestable frente a cierta entrada. Esta propiedad se puede aprovechar para “almacenar” cierta configuración, en el primer caso, o para implementar algún tipo de contador/oscilador, en el segundo.

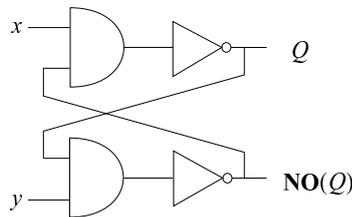
261. La evolución temporal de estos circuitos se denomina asíncrona, cuando el sistema evoluciona libremente a la velocidad que físicamente sea posible, o de modo síncrono, concordando con los pulsos que proporciona un reloj (“clock”) que fija el ritmo de evolución del sistema. Este reloj marca la tasa a que se producen los cambios (eventos) en el sistema.



2.4.1 Biestables y registros

262. El dispositivo secuencial más elemental utilizable para almacenar información es el biestable, del cual existen implantaciones muy diversas en función de su utilización. También se conoce como *flip-flop*, báscula y batidor.

flip-flop



Las configuraciones habituales son: cerrojo (latch), Set-Reset (S-R), J-K, Maestro-Esclavo (M-S), oscilador (toggle -T-) y retardo (delay -D-), (la mayoría controladas por reloj).

263. El caso más legible es el biestable tipo D: dispone de una entrada D , una salida Q y una entrada de reloj. Cuando la variable reloj envía un pulso, el biestable evoluciona, si la entrada es 1, la salida se convierte en 1 y si la

entrada es 0 la salida se convierte en 0. Esta salida se mantiene en ausencia del pulso de reloj, aunque cambien las entradas.

264. Un caso útil es el biestable tipo T: dispone de una entrada T , una salida Q y una entrada de reloj. Cuando la variable reloj envía un pulso, el biestable evoluciona, si la entrada es 1, se invierte el valor de Q que tuviera en el instante anterior (de 1 a 0 o de 0 a 1), y si la entrada es 0 la salida se mantiene inalterada. Esta salida se mantiene en ausencia del pulso de reloj, aunque cambien las entradas.

265. La unidad elemental de almacenamiento en una computadora (desde el punto de vista arquitectónico) es el registro. Un registro está formado por n biestables conectados de cierta forma con el fin de obtener ciertas funcionalidades. En un registro de se puede almacenar de una vez un patrón de bits, una vez almacenado puede ser una fuente de datos para una unidad de comparación de números, suma, multiplicación, un periférico, etc.

266. Algunas operaciones sobre información binaria se realizan empleando ciertos tipos de registro, como es el caso del registro de desplazamiento. Mediante el

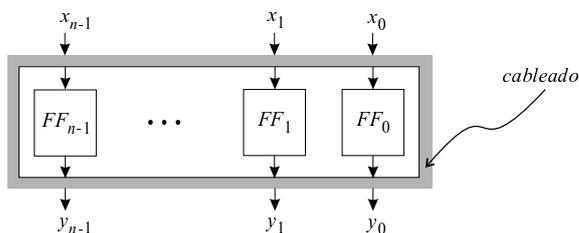


Figura 2.2: Registro.

registro de desplazamiento, una vez almacenados los bits en cada uno de los biestables, se puede desplazar (en el caso de la figura, horizontalmente) el contenido de los biestables a derecha o a izquierda. En el caso de tener representado un dato en binario natural, esta operación corresponde a una división por 2 (a derecha) o una multiplicación por 2 (a izquierda).

registro de desplazamiento

2.4.2 Bus

267. Entre los elementos básicos que definen la estructura de un computador se encuentra el bus. Se puede definir un bus como un camino que permite comunicar selectivamente un cierto número de componentes o dispositivos de acuerdo a ciertas normas de conexión.

268. La operación básica del bus se denomina ciclo del bus y se define como el conjunto de pasos necesarios para realizar una transferencia elemental entre dos dispositivos conectados al bus.

ciclo del bus

269. Un tipo especial de buses son los que comunican un computador con sus periféricos. Estos buses necesitan una interfaz de dispositivo que es el conjunto de registros y circuitos necesarios para manejar ese dispositivo (descodificador de dirección de dispositivo, registros de datos, registros de estado, circuitos de control, etc. La interfaz comunica los conductores del bus con el dispositivo.

interfaz de dispositivo

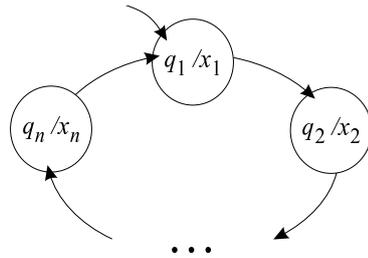
Entre las tareas encomendadas al bus se encuentran: (i) reconocer la dirección del dispositivo, (ii) almacenar el estado del dispositivo mediante el registro de estado, (iii) proporcionar un espacio de almacenamiento mediante un búffer de datos y (iv) proporcionar las señales de temporización y control adecuadas para realizar las transferencias de datos o estados.

De esto se desprende que un bus no es solo un conjunto de conductores que conecta varios elementos, sino también una serie de circuitos adicionales para realizar una serie de funciones residuales necesarias.

270. Entre las características que definen un bus se hallan: el tipo de lógica, el grado de paralelismo, la sincronización, el protocolo, la temporización, la direccionalidad de los datos, la estrategia de control, la velocidad y tasa de transferencia, la longitud máxima, la capacidad de conexión y el soporte.

271. Entre los niveles de especificación de un bus se hallan: el nivel mecánico, el nivel eléctrico (u óptico en su caso), el nivel lógico, el nivel de temporización básica, el nivel de transferencia elemental y el nivel de transferencia de bloque.

2.4.3 Contador



272. Un *contador* es un sistema secuencial con una entrada de reloj (en general, una entrada de pulsos) y unas salidas $\{z_1, \dots, z_m\}$, tal que las salidas presentan una serie de configuraciones preestablecidas y en determinado orden; cada vez que llega un pulso de reloj se pasa de una configuración a la siguiente, al cabo de una serie completa de pulsos se vuelve al estado inicial. La longitud de la serie se denomina *módulo* del contador.

Los contadores se utilizan habitualmente para contar el número de pulsos de una serie de longitud finita, de modo que para cada uno de los términos de la serie proporcionan cierta salida arbitraria. Este uso puede ser simplemente contar, o generar una secuencia de señales de control, de forma continua y predeterminada. En un reloj digital se combinan diversos contadores de módulo diferente .

2.5 Controladores

273. En términos de electrónica digital, determinadas tareas de automatización se llevan a cabo mediante el diseño de sistemas secuenciales que involucran una configuración de estados de mayor complejidad que los contadores. Los dispositivos encomendados a esta tarea se denominan genéricamente unidades de control, controladores o autómatas.

En el interior de cada procesador (y microprocesador), se encuentra una unidad de control con la tarea de gobernar el comportamiento de registros, contadores, buses, decodificadores, etc. En la lógica de control de un periférico, como un disco duro, o un periférico de comunicación serie, se encuentra un controlador que decide el comportamiento secuencial del dispositivo. Estos dispositivos pueden encargarse de tareas adyacentes a la unidad de proceso de una computadora, desde el manejo de la memoria, hasta el manejo de protocolos de comunicación de alto nivel en redes.

274. La construcción de circuitos de control se puede realizar mediante las técnicas conocidas de diseño de autómatas y circuitos digitales, o mediante el empleo de un controlador programable en algún lenguaje de programación (típicamente de bajo nivel. En el entorno de automatización industrial, se conoce a este tipo de controladores como *autómatas programables* y realizan tareas repetitivas de control de automatismos de máquinas y robots. Estos últimos se programan en lenguajes de uso específico, como por ejemplo los “lenguajes de contactos”.

En pequeñas aplicaciones, la tarea de automatización se encomienda a un tipo

especial de microprocesador denominado *microcontrolador*, de naturaleza apropiada para el control de periféricos. Tales son los dispositivos que controlan robots, lectores de videocasete, hornos microondas teléfonos, etc. microcontrolador

Nivel de máquina convencional. Unidades funcionales.

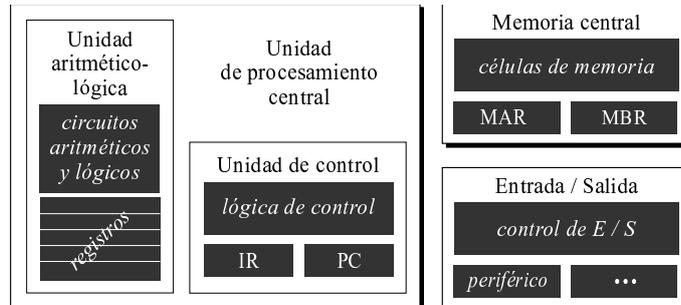
Índice General

3.1 El modelo de Von Neumann	73
3.1.1 Unidades funcionales	74
3.2 La unidad de procesamiento central	76
3.2.1 Temporización y Gestión de instrucciones	77
3.2.2 Interrupciones	78
3.3 Lenguaje máquina y ensamblador	80
3.3.1 Modos de direccionamiento	80
3.3.2 Instrucciones del NMC	82
3.3.3 Lenguaje ensamblador	82
3.3.4 Nemónicos de la norma IEEE 694	83

3.1 El modelo de Von Neumann

275. Las computadoras no son más que sofisticados dispositivos secuenciales. Si en la máquina de Turing, los bloques fundamentales son un control que describe el comportamiento de la máquina, y una cinta con datos de partida, trabajo y resultados, en la máquina de Turing Universal, el comportamiento de la máquina y los datos se almacenan en la cinta.

Del mismo modo, las computadoras primitivas se asemejan a las máquinas de Turing habituales, y la arquitectura de Von Neumann se asemeja a la máquina de Turing Universal, siendo este último más flexible que el primero. *El programa y los datos se almacenan en la memoria de la computadora.*



276. Los elementos principales del modelo de Von Neumann son el procesador y la memoria.

- La memoria es un lugar donde se almacenan datos, siendo unos interpretados como instrucciones de un programa, y otros como los datos de éste.
- El procesador (unidad de proceso central -CPU), es una unidad que bajo el control del programa almacenado, realiza el procesamiento de los datos.

277. Junto a estos dos elementos, se encuentran los dispositivos de entrada y salida (E/S) que permiten la comunicación de la computadora con el exterior, para enviar y recibir datos. Estos últimos pueden ser datos o programas vistos, de forma preliminar, como datos. Las unidades de E/S permiten que las computadoras gobiernen dispositivos como los discos, redes de comunicaciones, unidades de visualización, actuadores mecánicos, etc.

3.1.1 Unidades funcionales

278. La mayoría de las computadoras actuales siguen los mismos patrones de la arquitectura de Von Neumann, aunque se haya progresado en cada uno de los niveles de la arquitectura de la máquina. Los bloques o *unidades funcionales* de que consta una computadora son: Unidad de procesamiento central, memoria y entrada/salida.

279. Cada una de estas secciones está especializada en realizar una función y se interconectan de diversas formas mediante uno o más buses de comunicaciones. En este momento se supondrá que todos los elementos están unidos por un único bus, que se utiliza para comunicar datos entre las diferentes unidades funcionales y que porta las señales de control y direccionamiento necesarias para el funcionamiento correcto de la computadora.

Memoria central (o principal) 280. Este dispositivo almacena información de entrada, resultados intermedios de algún cálculo y datos de salida. Adicionalmente, la memoria se utiliza para almacenar tablas de referencias a datos e instrucciones de computadora que componen los programas. Este almacenamiento se obtiene mediante mecanismos electromecánicos, magnéticos o electrónicos.

281. La memoria se organiza en forma de tabla, donde cada elemento de tabla tiene la misma capacidad (1, 2 o 4 bytes), y se identifica mediante un número entero. Este número se denomina *dirección de memoria* o *posición de memoria*. Las celdas de la memoria ocupan posiciones correlativas desde 0 hasta algún número, admitiéndose que en algunos lugares existan “huecos” donde no hay celdas de memoria.

dirección de memoria

282. Tres características importantes de la memoria central son: el tiempo e acceso (tiempo requerido para localizar, recoger y hacer utilizable una celda de memoria), el tamaño de la memoria (el número de elementos, celdas, y por extensión posiciones de memoria, de que consta) y la organización de la memoria (número de bytes que componen cada elemento de memoria).

La velocidad de la memoria central, hoy en día, ronda los nano segundos (ns) ($10^{-9}s$), por término medio, y se suele medir en bytes, aunque se organice en celdas de 2, 4 u 8 bytes.

283. Las dos operaciones básicas de la memoria son la lectura y la escritura. En la primera, se pone la dirección del dato en el registro de dirección de memoria (MAR), y se indica la operación de lectura, cuando sea posible, aparece el dato requerido en el registro de intercambio con la memoria (MBR)¹. En la escritura, se pone la dirección del dato en MAR y el dato a escribir en MBR, indicando posteriormente la operación de lectura. En este modelo simplificado, cada operación de lectura/escritura, excluye cualquier otra simultánea.

Unidad central de proceso (o procesador central) 284. Su cometido es controlar la actividad de la computadora y operar como una entidad agente que procesa los datos conforme a un programa. Al diseñar un procesador se especifica el conjunto de instrucciones que será capaz de realizar; entre ellas se encuentran operaciones aritméticas, lógicas, de transferencia de datos con la memoria y el procesador y primitivas para controlar la marcha del programa. Este repertorio de instrucciones define en gran medida la versatilidad y potencia de la computadora y se suele denominar *juego de instrucciones*.

juego de instrucciones

285. El procesador, trabaja a la manera de un intérprete, y sus tres tareas principales son:

1. Extraer la instrucción correspondiente del programa que se está ejecutando, de la memoria central, y decodificar su significado.
2. Ejecutar la instrucción leída anteriormente.

¹MAR: Memory Address Register. MBR: Memory Buffer Register

3. Localizar la siguiente instrucción y volver al paso 1.

ejecución secuencial

Este proceso continúa indefinidamente, y es la base de la *ejecución secuencial* de los programas. Entre el juego de instrucciones, existen algunas dedicadas a modificar excepcionalmente alguno de los pasos anteriores.

286. La unidad central, aparece compuesta de dos bloques:

1. **Unidad aritmético-lógica (ALU):** se encarga de realizar el procesamiento de datos individuales representados como patrones de bits. Estas son sumas, restas, comparaciones, operaciones lógicas, etc.²

registros

Esta suele estar asistida por un conjunto de *registros* de uso general que contiene los datos sobre los que operará la ALU. El tamaño de estos registros (el número de bytes de que constan) define el tamaño de palabra de la CPU.

2. **Unidad de control:** que se encarga de mantener el ciclo de la ejecución secuencial de la CPU, efectuando cada una de las etapas descritas más arriba.

Del mismo modo que la ALU, se recurre a varios registros para almacenar cierta información relativa al estado de la máquina, como el contador de programa (PC), el registro de instrucción (IR) o la palabra de estado (SW)³.

Estos registros mencionados forman parte de la memoria de la máquina, pero no se incluyen dentro de la memoria central y se direccionan de modo diferente.

Entrada/Salida 287. Las computadoras digitales admiten como entrada, números, las letras del alfabeto y otros símbolos legibles para los humanos, en un lenguaje de *representación externa* o bien otras representaciones proporcionadas por sensores externos. Los dispositivos de entrada hacen acopio de la representación y la traducen en su sistema de *representación interna*, apta para el manejo por la computadora.

representación externa

representación interna

Del mismo modo, los dispositivos de salida traducen la representación interna de la información a otra forma apta para ser almacenada, o emitida en algún dispositivo, como una red de comunicaciones o una impresora.

3.2 La unidad de procesamiento central

288. Como se comenta más arriba, cada CPU se distingue por el diseño de su ALU, el tamaño de los datos que maneja, la versatilidad y potencia de su juego de instrucciones, y también por la velocidad a que es capaz de ejecutar un programa.

La arquitectura de los procesadores es un tema complejo, en el que abunda los casos de estudio. Los diseñadores, a partir de los requisitos del sistema, inventan soluciones particulares para cada problema según la tecnología del momento y los compromisos adoptados (compatibilidad, etc).

²ALU: Arithmetic Logic Unit.

³PC: Program Counter. IR: Instruction Register. SW: Status Word.

3.2.1 Temporización y Gestión de instrucciones

Un pequeño detalle del bucle de ejecución de instrucciones muestra la interconexión de los componentes de la CPU

- **289.** Al iniciar la ejecución de un programa, se ubica en el PC la dirección de memoria donde comienza el mismo. Tiene que haber un proceso capaz de hacer la carga en memoria de programas y la carga en el PC del valor adecuado. Posteriormente, se carga esta dirección en el MAR y se indica la lectura, tras lo cual se rescata la instrucción desde el MBR hacia el IR.
- **290.** La unidad de control (CU) decodifica la instrucción del IR y descompone su ejecución en una serie de pasos que suelen denominarse *micro-operaciones*, y que pueden implicar el concurso de otros componentes. Típicamente, en una instrucción aritmética, el CU envía las señales de control hacia la memoria para traer los operandos a los registros de uso general, si es que se encuentran en la memoria central. A continuación se envían las señales a los circuitos lógicos de la ALU para realizar la operación.

micro-operaciones

Entre los registros de propósito general, suele haber una diferencia entre aquellos encomendados a contener direcciones de memoria y los empleados en operaciones de proceso de datos. Una instrucción puede referirse a estos registros e incluso al registro PC. En este caso se está cambiando el apuntador a la siguiente instrucción: esta es la forma en que se realizan las *bifurcaciones* de la secuencia principal del programa, para implementar bucles y selecciones.

bifurcaciones

- Completada la operación, se procede, de nuevo, con la dirección a que apunta el PC.

291. La unidad de control es capaz de conocer la secuencia de micro-operaciones de una instrucción del NMC merced a dos posibilidades:

- a) Posee una memoria estática, en forma de microprograma, con la secuencia de micro-operaciones. Esto es así en los procesadores con *microprogramación*.
- b) Contiene un autómata finito construido empleando técnicas de diseño lógico de circuitos secuenciales, y no posee explícitamente una tabla almacenada con las secuencias de micro-operaciones. Esto es así en los procesadores con *lógica cableada*.

microprogramación

lógica cableada

A igualdad de juego de instrucciones, las CPU microprogramadas pueden diseñarse de forma más simple y modular. Las segundas son más complicadas y difíciles de diseñar, pero gobiernan de forma más óptima los recursos y a mayor velocidad.

292. Como ya se ha dicho, de entre las instrucciones de un programa la mayoría trata con los datos del problema pero hay una minoría que reflexiona sobre el programa mismo, las instrucciones del control de flujo de programa. Entre estas

conviene distinguir en dos grupos: (i) saltos (incondicionales) y bifurcaciones (saltos condicionales); (ii) Invocaciones y retornos de procedimientos (condicionales o no). Estos últimos permiten retornar al punto de invocación sin tener que especificar el punto donde se hizo la llamada.

293. En el caso de los saltos, la secuencia de micro-operaciones es más sencilla, dado que consiste en cargar en el PC determinado valor, con la dirección de la instrucción siguiente. Esta carga se puede realizar condicionalmente al resultado de alguna operación de comparación, suma, resta, o comprobación de bits (estos datos suelen reflejarse en la palabra de estado del procesador SW).

294. Por contra en el caso de los procedimientos (subrutinas), es necesario realizar cierto trabajo extra. Las subrutinas se emplean para reutilizar fragmentos de programa en diferentes partes del código. Al escribir un procedimiento, el programador debe cumplimentar los siguientes pasos:

1. Definir la interfaz del procedimiento: donde se van a encontrar los datos de partida, donde se almacenará el resultado y qué recursos de memoria central, registros de propósito general y específico se pueden modificar al ejecutar la subrutina.
2. Codificar el procedimiento con los requerimientos.
3. Añadir al final del procedimiento una sentencia especial que indica el final del procedimiento.

295. Cuando se invoca al procedimiento el procesador almacena ordenadamente la dirección donde se debe recomenzar la secuencia en curso (PC). Tras hacer un salto al procedimiento y completarlo, la sentencia de fin de procedimiento rescata el valor del PC, almacenado previamente y continúa con la ejecución.

296. Como quiera que dentro de un procedimiento se puede comenzar otro procedimiento, la estructura de datos que permite este almacenamiento debe soportar un almacenamiento ordenado de las direcciones de retorno, permitiendo extraer, al final del procedimiento actual, el último valor almacenado. Esta estructura se conoce como *pila* (o memoria LIFO)⁴.

297. La pila de retorno de procedimiento se encuentra en la memoria central, y como es lógico, cada programa en ejecución tiene la suya propia. La unidad de control sabe como manejar esta pila y dispone de uno de los registros de direcciones para ubicarla en la memoria, el puntero de pila (SP)⁵.

3.2.2 Interrupciones

298. La ejecución de la secuencia de instrucciones de un programa puede verse interrumpida en algún momento, como consecuencia de la existencia de una interrupción (excepción, fallo, aborto, malogro o *trap*). Incluso el proceso en ejecución

⁴LIFO: Last In First Out (primero en entrar, primero en salir).

⁵SP: Stack Pointer.

puede “lanzar” interrupciones. Las interrupciones son norma general en las computadoras y plasman la forma en que esta atiende a los *sucesos asíncronos* con el programa. sucesos asíncronos

El procesador resuelve la interrupción ejecutando otro programa o procedimiento prioritario o *procedimiento de servicio o gestión de la interrupción*.

299. El origen de una interrupción puede ser:

- Externo al programa
 - Hardware: Fallo de alimentación, anomalías en el hardware, petición de E/S de un periférico (petición de usuario, teclado, disco, etc.
 - Software: Desbordamiento o excepción matemática, gestión de tiempo compartido por el Sistema Operativo.
- Interno al programa.
 - Utilización de una instrucción del NMC inexistente, violación de la protección de memoria, petición del programador.

300. Al programar un sistema operativo para un procesador, la primera tarea del diseñador consiste en escribir procedimientos de servicio para las interrupciones, entre las que se encuentra la puesta en marcha de la computadora. Cada procesador implementa un método de localización de estos procedimientos de servicio, el más usado es el método de tabla o vector de interrupciones.

El vector de interrupciones es una zona de la memoria principal que contiene las direcciones de los procedimientos de servicio de las interrupciones, y la unidad de control de la CPU conoce dónde desde el momento en que se le proporciona tensión eléctrica. La rutina de botado del sistema es el primer procedimiento de servicio, y se encarga de cargar en memoria el resto y organizar el vector de interrupciones.

301. La ejecución y servicio de una interrupción tiene, usualmente, las siguientes fases:

1. Petición de servicio (IR: Interrupt Request). Esta petición de servicio se clasifica en un esquema de prioridades.
2. Reconocimiento de la interrupción (IA: Interrupt Acknowledge). Se decodifica el identificador de interrupción, se salva el estado del procesador en ese instante (usualmente en una pila) y pasa control al procedimiento de servicio.
3. Ejecución del procedimiento de servicio.
4. Final de interrupción. Se restaura el contenido de los registros y se restablece el estado del procesador.
5. Paso del control de la CPU al proceso interrumpido.

302. El servicio de las interrupciones suele realizarse en un modo especial de funcionamiento de la CPU: *modo supervisor*, privilegiado o excepcional, que se contrapone al *modo usuario* en que se ejecutan los procesos normales. En modo supervisor, la prioridad de las subsiguientes interrupciones se modifica y es posible ejecutar ciertas instrucciones del NMC no permitidas en modo usuario. Este modo se restablece en el paso 4, y es necesario para implementar cómodamente un sistema operativo moderno.

modo su

modo u

3.3 Lenguaje máquina y ensamblador

303. Un programa en lenguaje máquina es una secuencia instrucciones, cada una de las cuales se representa mediante un patrón de bits. Al igual que en un lenguaje de alto nivel, las instrucciones hacen referencia a datos de la memoria central y en algunos casos a datos que aparecen en el texto del programa acompañando a ciertas instrucciones. Los diferentes modos en los que las instrucciones obtienen los datos se denominan *modos de direccionamiento*.

304. Codificar un programa en el NMC es una tarea complicada, dado que las instrucciones se refieren a componentes reales del computador, y no es posible abstraer tipos de datos ni representar simbólicamente operaciones ni objetos. Además, la programación directa de patrones de bits es una tarea muy farragosa y propensa a errores. Para paliar esto se emplean lenguajes de ensamble o ensambladores, en los que los patrones de bits se substituyen por nombres simbólicos (*nemónicos*) y es posible utilizar una representación externa asequible para estos programas.

nemónicos

3.3.1 Modos de direccionamiento

305. Los modos de direccionamiento de un procesador son las diferentes formas de transformación del campo de operando de la instrucción en la dirección del operando. Las computadoras emplean técnicas de direccionamiento con la finalidad de dar versatilidad de programación al usuario y reducir el número de bits del campo de operando. La potencia de una máquina se ve afectada por la variedad de modos de direccionamiento que admite.

Un procesador que ofreciera todos los modos de direccionamiento de que dispone, en todas las instrucciones de que dispone se dice que ofrece un juego de instrucciones completamente *ortogonal*, y su empleo resulta muy sencillo y potente. En la realidad se busca una situación de compromiso de la ortogonalidad con la complejidad del diseño del procesador.

ortogonal

306. La dirección de un operando en una instrucción, se manipula para obtener la *dirección efectiva* del operando en la memoria central. Esta puede depender, de la posición de la instrucción en el programa y de el estado y el valor de los registros en el momento de ejecutar la instrucción.

dirección efectiva

Los modos de direccionamiento más usuales son

- **307. Direccionamiento implícito** o inherente. El operando se especifica al definir la instrucción. Puede ser de (a) registro, o de (b) operando en pila.
- **308. Direccionamiento inmediato** o literal. El campo de operando contiene él mismo, sin transformación alguna, la información sobre la que se opera.
- **309. Direccionamiento directo por registro.** El operando reside en uno de los registros del procesador que es seleccionado por el campo de registro de k bits en la instrucción.

Sus ventajas principales son: (a) acceso muy rápido al dato y (b) compacidad, el código del registro es muy pequeño.

- **310. Direccionamiento directo** o absoluto. El campo de dirección no necesita transformación alguna para obtener la dirección efectiva. El principal problema es que una reubicación del programa puede modificar la posición efectiva de los datos, de modo que un programa que haga uso de ella, no es reubicable, a menos que los datos referenciados no cambien nunca de lugar.
- **311. Direccionamiento indirecto.** El campo de operando de la instrucción indica la ubicación de la dirección efectiva del operando. Esto es, la dirección de la instrucción indica donde se halla el puntero que indica el dato.
- **312. Direccionamiento relativo.** El campo de operando de la instrucción indica una cantidad positiva o negativa, que sumada al valor de cierto registro (implícito) proporciona la dirección efectiva. Si el rango de direcciones efectivas es reducido (*propiedad de localidad de referencia*), el direccionamiento relativo permite hacer referencias a posiciones de memoria con poco gasto de memoria, al indicar tan solo el *offset*.

Uno de los más empleados es el direccionamiento relativo al PC, empleado para realizar saltos, y bifurcaciones a partes locales del programa.

- **313. Direccionamiento por base y desplazamiento.** Basado en la propiedad de localidad de referencia, es posible direccionar el conjunto de datos de un proceso haciendo un direccionamiento relativo a un cierto registro base, con poco gasto de memoria.
- **314. Direccionamiento indexado.** La dirección efectiva se calcula sumando un registro de la CPU al campo de operando, siendo este registro uno específico para este uso, llamado *registro índice*.
- **315. Direccionamiento indirecto automodificado.** La dirección efectiva se encuentra en un registro, que es auto incrementado o decrementado, antes o después de hacer uso de la dirección del dato. o decrementado, antes o después de hacer uso de la dirección del dato.

registro índice

3.3.2 Instrucciones del NMC

316. De la cantidad y calidad del juego de instrucciones de un procesador, depende en gran medida la potencia de este. Actualmente coexisten dos grandes tendencias: CISC (juego de instrucciones complejo) y RISC (juego de instrucciones reducido)⁶.

CISC implica muchas instrucciones, y un diseño de procesador complejo, potente y de tipo microprogramado. RISC implica un conjunto parco e imprescindible, un diseño muy optimizado y lógica cableada. Parece probado que los intérpretes y compiladores de lenguajes de alto nivel de hoy en día no hacen uso más que de un conjunto de instrucciones y modos de direccionamiento bastante reducido, al contrario que los programadores de NMC que hacen uso de toda la potencia de un juego de instrucciones. Dado que las computadoras de hoy en día se programan, mayormente, en lenguajes de alto nivel parece lógica la opción RISC.

317. Las instrucciones del NMC de un procesador se pueden clasificar en cuatro grupos:

- a) transferencia de datos entre unidades funcionales.
- b) procesamiento aritmético-lógico.
- c) control de la secuencia de ejecución.
- d) otros (control del procesador, ...)

3.3.3 Lenguaje ensamblador

318. El lenguaje ensamblador ayuda al programador a mejorar su interacción con la computadora en un código comprensible, y a resolver temas tediosos como la asignación de memoria de los datos y organización del texto en procedimientos y ficheros de trabajo independientes.

319. Cualquier ensamblador moderno ofrece (a) *nemónicos* para cada una de las instrucciones de un programa,

[ETIQUETA] NEMÓNICO OPERANDO, ... [;COMENTARIO]

ETIQUETA permite hacer referencia simbólica a la dirección de la instrucción, y tanto NEMÓNICO como OPERANDO permiten hacer referencia a instrucciones, registros, posiciones de memoria, posiciones de programa, mediante identificadores similares a los de cualquier lenguaje de alto nivel. COMENTARIO permite añadir legibilidad al programa.

320. También existen pseudoinstrucciones o directivas del ensamblador, que sirven para (a) asignar valores constantes a nombres simbólicos, (b) reservar posiciones de memoria en el programa y darles valores iniciales, (c) indicar como desea el programador ubicar el programa e incluso (e) escribir macroinstrucciones que permiten escribir expresiones regulares para realizar substituciones de literales (con o sin argumentos) en el código fuente del programa, en tiempo de ensamblado.

⁶CISC: Complex Instruction Set Code. RISC: Reduced Instruction Set Code.

321. Cada procesador es especial en cuanto a su juego de instrucciones, y modos de direccionamiento, y no podía de ser diferente en cuanto al estilo del ensamblador. Aun así existe una norma IEEE 694 que permite diseñar ensambladores con nombres uniformes que permitan a los programadores aprender con facilidad a programar en cada uno de los procesadores.

3.3.4 Nemónicos de la norma IEEE 694

Resumen de nemotécnicos definidos en el estándar IEEE 694

Nemónico	Explicación	Sufijos
INSTRUCCIONES DE TRANSFERENCIA DE DATOS		
LD	Cargar desde la memoria	
ST	Memorizar	
MOVE	Copiar de fuente a destino	
FILL	Memorizar en varias posiciones	
PUSH	Apilar el operando	
POP	Desapilar el operando	
XCH	Intercambiar contenidos	
IN	Lectura desde puerto a registro o memoria	
OUT	Escritura a puerto desde registro o memoria	
INSTRUCCIONES ARITMÉTICAS		
ADD-, SUB-	Suma o resta	-U, -S, -C, -R
MUL-, DIV-	Multiplicación o división	-U, -S
TEST	Actualizar SW según el operando	
TESTSET	Lee y actualiza una posición de memoria	
NEG	Cambia el signo del operando	
ADJ	Ajusta a BCD	
CVT	Convierte el tipo o formato del operando	
INSTRUCCIONES LÓGICAS Y DE DESPLAZAMIENTO		
AND, OR	Producto o suma lógica	
XOR	OR exclusivo	
NOT	Complemento lógico	-C, -V
SET, CLR	Establecer a 1 o 0 los operandos	-C, -V
SHL, SHR	Desplazamiento de bit a izquierda o derecha	
SHLA, SHRA	Desplazamiento aritmético a izquierda o derecha	
ROL, ROR	Rotación a izquierda o derecha sin acarreo	
ROLC, RORC	Rotación a izquierda o derecha con acarreo	
INSTRUCCIONES LÓGICAS Y DE DESPLAZAMIENTO		
BR	Salto incondicional	
-B-	Salto condicional	(*)
CALL-	Invocación a procedimiento	(*)
RET-	Retorno de procedimiento	(*)
RETI	Retorno de servicio de interrupción	
RETSKIP	Retorno de procedimiento incrementado	
SKIP-	Salto hacia delante	(*)
INSTRUCCIONES DE CONTROL Y MISCELÁNEA		
BRK	Iniciar una interrupción	
CC-	El operador destino toma el valor de la condición	
EI, DI	Habilita o inhabilita interrupciones	
ENTER	Suministro de protocolos de entrada (ENTER)	
EXIT	y salida (EXIT) de procedimientos	

Resumen de nemotécnicos definidos en el estándar IEEE 694

HALT	Parar la CPU
INS	Inserción de campo de bits
NOP	Avanzar a la instrucción siguiente
TR	Reemplazar operador-dirección por contenido
WAIT	Parar hasta interrupción

(*) Prefijos: D-, I-

Sufijos: -Z, -NZ, -E, -NE, -GT, -GE, -LT, -LE, -H, -NH, -L, -NL, -C, -P, -N, -V, -PE, -NC, -NP, -N, -NV, -PO, -T, -F.

Sufijos de algunas instrucciones del estándar IEEE 694

Sufijos	Significado
Sufijos de instrucciones aritméticas	
-U	<i>Unsigned</i> sin signo
-S	<i>Signed</i> con signo
-C	Con acarreo
-R	<i>Reverse</i> con resta inversa
Sufijos de instrucciones lógicas y de desplazamiento	
-C	Con acarreo
-V	Con <i>overflow</i>
Sufijos de instrucciones lógicas y de desplazamiento	
D-, I-	Decrementar o incrementar operando antes del salto
-Z, -NZ	Si cero o no cero
-E, -NE	Si igual o no igual
-GT, -GE	Si mayor o mayor-o-igual
-LT, -LE	Si menor o menor-o-igual
-H, -NH	Si mayor o no mayor
-L, -NL	Si menor o no menor
-C, -NC	Si hay acarreo o no
-P, -NP	Si es positivo o no positivo
-N, -NN	Si negativo o no negativo
-V, -NV	Si desbordamiento o no
-PE, -PO	Si paridad par o impar
-T, -F	Si <i>cierto</i> o <i>falso</i>

Organización del sistema de memoria

Índice General

4.1 Características de las memorias	86
4.1.1 Clasificación de las memorias por su utilización.	87
4.2 Jerarquía de las memorias	87
4.3 Organización de la memoria	88
4.3.1 Asociaciones de memorias	89
4.3.2 Ampliación del espacio de memoria direccionable	89
4.4 Memorias especiales	90
4.4.1 Colas	90
4.4.2 Pilas	90
4.4.3 Memorias asociativas	90
4.5 Memorias caché	91

322. En la memoria central se almacena el texto y la memoria de trabajo de los procesos que se ejecutan en cada computadora. La memoria es un objeto de estudio tanto desde el punto de vista de la construcción de la máquina como para el sistema operativo, como recurso que se puede gestionar. En esta parte no se aborda más que como componente de la arquitectura de la máquina.

323. Una de los parámetros de diseño de una arquitectura, es el rango de direcciones posibles de memoria (memoria direccionable). Esta capacidad puede ser muy superior a la cantidad real de memoria instalada en una computadora (memoria física).

4.1 Características de las memorias

324. La memoria de una computadora se puede caracterizar conforme a ciertos parámetros. Según los diferentes tipos de memoria presentan propiedades diferentes, son aptas para cometidos diferentes. Las características bajo estudio son las siguientes:

- **Capacidad:** Cantidad de información total que puede almacenar, habitualmente. Se mide en Kilo, Mega o Giga bytes.
- **Longitud de palabra:** Es el tamaño del átomo de memoria direccionable en la computadora. Se denomina también *granularidad* de la memoria.
- **Tiempo de acceso:** Tiempo medio necesario para realizar una operación de lectura, que puede ser diferente del tiempo medio de acceso para una operación de escritura. Su inverso se denomina *frecuencia de acceso* y se mide en bits/s.
- **Tiempo de ciclo:** Tiempo mínimo que debe transcurrir entre dos lecturas. Su inverso se denomina *tasa de transferencia, ancho de banda o caudal* y se mide en bits/s.
- **Modo de acceso:** Método empleado para acceder a una escritura o lectura en una celda de memoria. Puede ser:
 - *secuencial*: si es menester acceder a la información por orden de almacenamiento en el dispositivo. Penaliza los acceso a celdas de memoria con una posición elevada.
 - *arbitrario* o *aleatorio*: si se puede acceder a la celda directamente por su posición. No se penaliza la situación de la celda en la memoria.
 - *directo* o *semi-aleatorio*: Si para acceder a una celda hay que emplear ambos métodos, en dos etapas diferenciadas.
- **Volatilidad:** Una memoria es volátil si se pierden los contenidos de ésta al interrumpir el suministro de energía. Lo contrario es **persistencia**.
- **Destrucción:** Una memoria es de lectura destructiva (DRO) si el acceso a los datos destruye la información de la celda. Si no es así se dice que es de lectura no destructiva (NDRO).¹
- **Orden:** Los bits de una palabra con información se pueden numerar de izquierda a derecha (*big endian*) o de izquierda a derecha (*little endian*).
- **Densidad de almacenamiento:** Cantidad de bits por unidad de volumen o superficie.

¹DRO: Destructive readout.

- **Fiabilidad:** Tiempo medio entre dos averías consecutivas (MTBF).² La fiabilidad es mayor cuanto mayor es MTBF, y se puede aumentar mediante dispositivos de detección y corrección de errores.
- **Acoplamiento:** La relación de proximidad lógica entre los datos almacenados en la memoria y la unidad de proceso. En un sistema monoprocesador el acoplamiento es fuerte. En un sistema multiprocesador la memoria puede estar distribuida entre los nodos (*sistemas débilmente acoplados*) o puede estar *centralizada* en una única unidad funcional (*sistemas fuertemente acoplados*).

sistemas débilmente acoplados
sistemas fuertemente acoplados

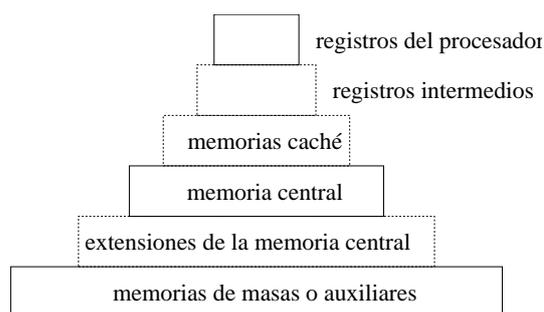
4.1.1 Clasificación de las memorias por su utilización.

325. Las memorias se pueden clasificar según se diseñen para ser actualizadas o para ser leídas únicamente, en dos grupos memoria de lectura y escritura (RAM) y memoria de lectura (ROM)³:

- **RAM:** Pueden ser de ferritas o electrónicas. En este último grupo, pueden ser bien estáticas (SRAM: Static RAM) o (DRAM: Dinamic RAM). En la DRAM es pertinente realizar un refresco periódico de la información mediante circuitos adicionales.
- **ROM:** Estas memorias, durante su ciclo de uso normal, solo se pueden leer. Existen varias posibilidades: ROM de fábrica (o por máscara), cuyo contenido viene fijado en su fabricación; ROM programables una sola vez (PROM); PROM borrables mediante rayos ultravioleta (Erasable-PROM); PROM borrables eléctricamente (Electrically EPROM). Estas últimas también se denominan NVRAM (no volátil RAM).

4.2 Jerarquía de las memorias

326. La memoria de una computadora es un recurso sujeto a restricciones de índole físico, de disponibilidad de espacio y económico. En consecuencia, no es posible disponer de toda la memoria central que se quisiera, y si fuera así no todo lo rápida que se deseara, y si fuera así no todo lo persistente que se deseara, etc. En consecuencia el diseñador de computadoras hace uso de las memorias comercialmente disponibles construyendo una jerarquía de memorias que se acomodan de forma piramidal: en la base de la pirámide hay memorias persistentes, masivas, pero lentas; en la cúspide hay memorias rápidas, pero escasas y volátiles. Entre ambos extremos hay un conjunto de memorias con características en gradación de volatilidad, precio, volumen y rapidez.



²MTBF: Mean time between failures.
³RAM: Random Access Memory. ROM: Read Only Memory. En realidad persiste un equívoco al respecto, pues bajo el acrónimo RAM no se hace referencia a lectura-escritura sino al modo de acceso. A pesar de ello se sigue utilizando la nomenclatura aunque conviene tener este aspecto en cuenta.

327. Los registros del procesador interaccionan continuamente en la CPU, y tienen un tiempo de acceso muy pequeño, aunque con una capacidad mínima, generalmente igual al ancho de palabra del procesador. El número de estos en la CPU suele ser del orden de decenas, hasta un máximo de algunos centenares. No forman parte de la memoria direccionable de la máquina.

Los registros intermedios constituyen un paso intermedio entre el procesador y la memoria. Actúan como eslabones con la memoria central y tampoco forman parte de la memoria direccionable de la máquina.

328. Las memorias caché o ante-memorias son memorias de poca capacidad, en comparación con la memoria central, aunque con un tiempo de acceso sensiblemente menor. Las computadoras modernas suelen tener dos niveles de caché, L1 en el mismo chip del microprocesador, y L2 en la tarjeta de soporte, siendo L1 más rápida y con menor capacidad que L2.

Ambas son memorias no direccionables y su modo de organización es asociativa. En los microprocesadores actuales, la memoria caché, incluso es capaz de diferenciar entre instrucciones y datos, de modo que la CPU puede recuperar simultáneamente datos e instrucciones.

329. Esta es la memoria física de la computadora y es direccionable por la CPU. Es este el nivel que habitualmente se considera al hablar de la memoria central de la máquina. Tienen un tiempo de acceso relativamente rápido y gran capacidad. La capacidad habitual de la memoria central ronda las centenas de Mbytes en estaciones de trabajo.

330. Las extensiones de memoria central son memorias de la misma naturaleza que la memoria central y que amplían su capacidad de forma modular. Tienen una capacidad mayor y tiempo de acceso similar. No son direccionables directamente por la CPU, mas sí con algún dispositivo de apoyo.

331. Las memorias masivas o auxiliares son memorias que residen en periféricos, como cintas, cartuchos, discos, disquetes, etc. En ellas se almacenan datos organizados típicamente en archivos. Suelen tener gran capacidad, pero un tiempo de acceso, variable en función del dispositivo, pero mucho mayor que la memoria central.

Estas memorias pueden funcionar como apoyo de la memoria central, para formar un sistema de memoria virtual, gestionado por el sistema operativo.

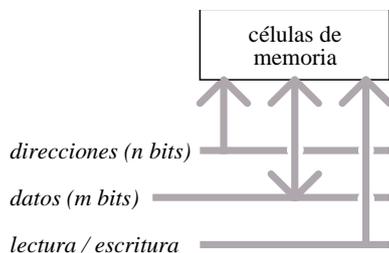
4.3 Organización de la memoria

332. Externamente, la memoria se puede ver como un bloque que obedece a dos instrucciones: lectura y escritura; en la operación de lectura se accede a la celda, cuya dirección indica el bus de direcciones y se deposita en el bus de datos, en la operación de escritura se deposita en la celda, cuya dirección indica el bus de direcciones, la información presente en el bus de datos.

333. Las direcciones de memoria se suceden como un vector, desde la posición 0 en adelante. La memoria decodifica esta dirección para activar una de las celdas,

según el bus de direcciones; esta celda es la que se lee o se escribe. Esta forma de de estructurar la memoria se llama *organización lineal* o de una dimensión.

organización lineal



Al organizar la memoria de esta forma, la complejidad del decodificador crece exponencialmente con el número de entradas, de modo que se construye internamente mediante la cooperación de dos decodificadores en la organización denominada de *dos dimensiones*.

4.3.1 Asociaciones de memorias

334. Los módulos de memoria habituales proporcionan tamaños de celdas del orden de 1, 2, 4 y 8 bits. Para construir memorias cuya unidad de organización sea superior, es necesario agrupar varios módulos, cada uno de los cuales da cuenta de ciertos bits de una celda.

335. Por otra parte puede ocurrir que cada módulo, por separado, no ocupe el espacio de direccionamiento requerido, con lo cual, es necesario agrupar varios módulos, de modo que previa decodificación del rango de direcciones que ocupe el módulo, superpongan sus capacidades en el espacio de direcciones.

336. Las *memorias entrelazadas* se construyen asociando módulos de memoria de modo que las celdas con posiciones consecutivas se dispersen en módulos diferentes. Con esto se persigue que el tiempo de espera que se produce en las operaciones de lectura y escritura no se acumule sobre un solo módulo, y los tiempos acceso a posiciones consecutivas de memoria se dispersen entre los módulos.

memorias entrelazadas

4.3.2 Ampliación del espacio de memoria direccionable

337. En algunas arquitecturas es menester manejar una memoria central mayor que la capacidad de direccionamiento de la CPU. Para paliar este problema se recurre a la técnica de *conmutación de bancos* y a la técnica de *extensión de direcciones*.

338. En la técnica de conmutación de bancos se activa o desactiva uno de los diferentes bancos memoria mediante un control por software: típicamente un puerto de E/S. La memoria se presenta en regiones, y se puede conmutar una región de memoria ocupada por un banco, por otro banco diferente. Presenta el problema principal, de que el código del programa se encuentre en otro banco diferente a los datos, dentro de la misma región de memoria. Para resolver este problema se requiere cierto soporte hardware asociado.

339. En la técnica de extensión de direcciones, se emplean varios *registros base* en los que se aloja los bits más significativos de la posición de una región de memoria. Puede ocurrir que el procesador provea ciertos registros base, con instrucciones para emplearlos, o puede que haya que emplear puertos de entrada y salida como registros base.

registros base

4.4 Memorias especiales

4.4.1 Colas

FIFO **340.** Una *cola* o *FIFO*⁴ es un dispositivo de memoria desprovisto de direcciones y con dos operaciones: leer y escribir. Al escribir se deposita un dato en la cola, y al leer se lee el dato más antiguo de la cola.

341. Las colas pueden realizarse por hardware (*colas cableadas*) y por software (*colas programadas*).

búfer **342.** Las colas hardware se suelen usar como *búfer* o memorias intermedias para compensar la diferencia de velocidad entre el procesador y los periféricos: la CPU deposita datos en la cola a la velocidad habitual, y el periférico lee estos datos a su propio ritmo.

343. Cuando la diferencia de velocidad es muy grande, o cuando se construye esta cola desde el interior de un programa, se emplean colas software. En este caso se emplea una zona de memoria central y variables que almacenan el comienzo y el fin de la cola.

A menudo se realiza esto mediante una cola o búfer circular para aprovechar mejor la memoria disponible.

4.4.2 Pilas

LIFO **344.** Una *pila* o *LIFO*⁵ es un dispositivo de memoria desprovisto de direcciones y con dos operaciones: leer y escribir. Al escribir se deposita un dato en la cola, y al leer se lee el dato más reciente de la cola.

stack También se denomina *stack*, y se emplean ampliamente en programación, dándose el caso de que algunas computadoras y lenguajes de programación basan su arquitectura en una o varias pilas LIFO.

345. Las pilas pueden realizarse por hardware y por software mediante apun­tadores. Las pilas software se emplean en la implementación de computadoras como estructura en la que se almacenan argumentos y datos en las llamadas a procedimientos, disponiendo los procesadores de instrucciones para manipular directa o indirectamente los contenidos de esta pila.

4.4.3 Memorias asociativas

CAM **346.** Una memoria asociativa o memoria direccionable por contenido o *CAM*⁶ es un dispositivo de memoria que identifica la información almacenada en cada una de sus celdas mediante el valor que proporciona una función aplicable a esta información. Esta función puede consistir en extraer ciertos bits del patrón de bits almacenados en la celda.

⁴FIFO:First Input-First Output.

⁵LIFO>Last Input-First Output.

⁶CAM: Content Addressable Memory.

Al leer un dato de una memoria CAM, se proporciona este valor (típicamente ciertos bits) que actúa como máscara que permite filtrar los datos de la memoria hasta que uno de ellos concuerda con esta máscara. A continuación se da salida a este valor.

4.5 Memorias caché

347. Una memoria caché⁷ es un dispositivo de memoria que se inserta ante un sistema donde se almacena y recupera información. Sirve para agilizar, e incluso adelantarse a la lectura de datos de este sistema.

348. Se puede insertar una memoria caché entre la memoria central y el procesador, y en este caso se denomina *antememoria*. La antememoria posiblemente almacenará los datos que se están empleando en ese momento, y la velocidad de acceso se multiplica en un factor entre 5 y 10.

antememoria

Esta caché puede venir implementada en la tarjeta de soporte de la memoria central (*on board*) o incluso en el interior del procesador (*on chip*).

349. La utilidad de la memoria caché se basa en el fenómeno de *localidad de referencia* que consiste en que las direcciones de memoria solicitadas por la CPU en un intervalo de tiempo suelen estar en un entorno cercano de direcciones de memoria y suelen repetirse en un intervalo de tiempo dado. Como es lógico esta localidad no solo se refiere a direcciones con datos, sino también a la localidad de referencia del texto de un proceso.

350. La memoria caché se puede construir de diversos modos, y un principio habitual suele ser el empleo de memorias asociativas. Además surgen cuestiones de diseño importantes como:

- *Función de mapa*: o correspondencia entre los bloques de la memoria principal y los de la memoria caché.
- *Algoritmo de remplazo*: Procedimiento que permite vaciar de la memoria caché aquellos datos que no se van a utilizar y substituirlos por otros nuevos, para manejar los diferentes tipos de faltas de bloque.
- *Políticas de extracción y actualización de memoria principal*: que se implementan en el algoritmo de remplazo y que permite saber cuándo es más probable que se vaya a utilizar o no una región de memoria.

⁷caché: del francés *acher*, ocultar.

Interfaz de Entrada / Salida

Índice General

5.1 Organización de Entrada / Salida	93
5.1.1 Sistema de E/S y periféricos	93
5.1.2 Transferencia elemental y operaciones de E/S	94
5.1.3 Sincronización y transferencia de datos	94
5.1.4 Soporte de Interrupciones para operaciones de E/S	96
5.1.5 Canales de E/S	96
5.2 Buses e interfaces de E/S	97
5.2.1 Características de los buses	97
5.2.2 Niveles de especificación	98
5.2.3 Jerarquía de buses	99
5.2.4 Ejemplos de buses	100

5.1 Organización de Entrada / Salida

5.1.1 Sistema de E/S y periféricos

351. Un *periférico* es cualquier elemento que pueda realizar un intercambio de datos con la CPU o la memoria. Todo aquello que hace referencia a la comunicación de la CPU y la Memoria con los periféricos es lo que se conoce como “*ENTRADA/SALIDA*” (E/S) y todo el hardware relacionado con el tránsito de la información es lo que constituye el sistema de E/S. periférico

352. La cantidad y variedad de periféricos es grande, y el sistema de E/S debe proveer soluciones para los siguientes problemas:

- *Direccionamiento.* Suelen existir varios periféricos, y las operaciones de lectura, escritura y control se deben realizar independientemente.
- *Velocidades diferentes.* Los periféricos pueden tener un rango de velocidades muy amplio, y pueden ir desde varios bytes/s a Mbytes/s.
- *Acceso lectura / escritura.* Existen periféricos sólo de lectura, otros de escritura y otros de lectura-escritura. Siempre que se hable de estas operaciones se hace desde el punto de vista de la CPU.

5.1.2 Transferencia elemental y operaciones de E/S

353. Los modos de operación con la E/S son diferentes en función de que se trate de una transferencia elemental, o que se trate de una operación de entrada/salida. En el primer caso se intercambia una única unidad de información con el periférico, mientras que en el segundo tiene lugar la transferencia de un conjunto de datos.

354. En el caso de la transferencia elemental, es necesario establecer una comunicación física entre el periférico y el resto del sistema y a continuación llevar a cabo el control del periférico mediante la consulta y modificación de su estado.

355. Para la operación de E/S, a parte de contar el número de datos hay que sincronizar las velocidades de los periféricos con el resto del sistema, donde difiere del caso anterior, y en tercer lugar detectar posibles errores mediante códigos de paridad o polinomiales, y si es preciso repetir la transmisión. Toda esta operación se realiza mediante búferes temporales y un mecanismo implícito de conversión de códigos internos del periférico y conversión de formato serie-paralelo, si es preciso.

5.1.3 Sincronización y transferencia de datos

356. Un periférico puede comunicarse o bien directamente con la CPU o bien directamente con la memoria. Mientras que en el primer caso la transferencia elemental tendrá lugar por la ejecución de una instrucción de la CPU (entrada-salida programada), en el segundo caso se habla de acceso directo a memoria (DMA¹)

E/S programada

357. La E/S programada resulta de la ejecución de una instrucción de E/S por la CPU. Como resultado se transfiere información desde un registro interno de la CPU hacia un registro interno de un periférico o viceversa.

Dado que la CPU interacciona directamente con el dispositivo, debe resolver los siguientes problemas

¹DMA: Direct Memory Access.

- **358.** *Direccionamiento:* El dispositivo ofrece una o varias puertas para comunicarse con la CPU. El conjunto de direcciones de que dispone la CPU para interactuar conforma el mapa de E/S.

Algunos computadores no diferencian entre el mapa de direcciones de E/S y la memoria principal, tratando las puertas de E/S como si fueran direcciones de memoria. Esta alternativa simplifica el diseño de la CPU pues no es necesario un conjunto de instrucciones específicas para E/S. A pesar de ello son posiciones de memoria “volátiles” y se emplean e modo diferente.

- **359.** *Líneas de datos:* Los datos pueden transmitirse por líneas de datos bidireccionales o por dos conjuntos unidireccionales. Además esta conexión puede ser mediante un bus común o mediante multiplexor/demultiplexor.
- **360.** *Temporización de la transferencia:* El modo de transferencia puede ser (i) síncrona, donde el control corresponde a la CPU y ésta preve el tiempo de respuesta del dispositivo, o (ii) asíncrona (o con interbloqueo) donde se establece un diálogo “handshake” entre CPU y periférico mediante líneas del bus de control. Esta última alternativa permite conectar periféricos con distintos requisitos de tiempo y es más fiable pues puede encomendarse al protocolo la validación de los datos leídos/escritos.

E/S por DMA

361. En el acceso directo a memoria (DMA)² el periférico se comunica directamente con la memoria principal de la computadora sin intervención de la CPU, por lo cual se acelera el proceso. Este modo requiere una circuitería especial conocida como *controlador DMA*, y es especialmente adecuado para bloques de datos.

362. Para poder realizar operaciones E/S por DMA, el controlador debe conocer:

1. Dirección del periférico.
2. Rango de memoria donde se realiza la operación.
3. Tipo de operación (lectura o escritura).
4. Número de operaciones elementales que van a integrar esa operación de entrada/salida.

363. El procedimiento de acceso directo a memoria depende de si la memoria es de características *multipuerta*. En este caso se reserva una para el uso exclusivo de la CPU, encomendándose el resto al los periféricos. En este caso los accesos a la memoria por cada puerto son transparentes.

En caso de no disponer de multipuerta, se utiliza el procedimiento de *robo de ciclo*. En este caso el usuario prioritario del canal es la CPU, y el controlador DMA solicita el uso del canal cuando es necesario.

²DMA: Direct Memory Access.

5.1.4 Soporte de Interrupciones para operaciones de E/S

364. Como se comenta en el la sección ?? las interrupciones son bifurcaciones externas del programa en ejecución, provocadas por señales y cuyo objetivo es que el sistema responda a un hecho determinado, aunque en ese momento se encuentre ejecutando otro programa.

Originalmente, las interrupciones se presentan como un medio de interaccionar asincrónicamente con los dispositivos. Para ello la CPU se encuentra conectada a la circuitería de E/S de modo que ciertos dispositivos requieren la atención del procesador para realizar alguna operación de E/S.

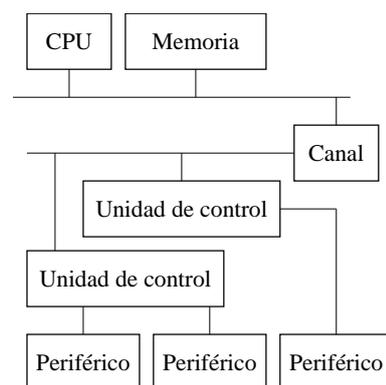
365. El sistema hardware empleado para solicitar a la CPU una interrupción se puede clasificar en dos grupos básicos:

- *Sistemas con línea de interrupción única.* El resto del sistema advierte a la CPU su petición de interrupción por una sola línea que recolecta las líneas de petición de los canales de E/S.

Dado que este método no proporciona la identificación del dispositivo (o dispositivos) solicitante, el procesador deberá o (a) recorrer los registros disponibles en busca del dispositivo que solicita la interrupción, para conocer su estado (esta técnica se denomina de encuesta, muestreo o *polling*) o bien (b) se sirve de circuitería de apoyo que realiza la detección y el control de interrupciones.

- *Interrupciones vectorizadas.* Los periféricos que interrumpen suministran al procesador la dirección de bifurcación en la que se encuentra su rutina de tratamiento, mediante el bus de datos. Esta dirección es una dirección de la memoria central, o un índice a un elemento del vector de interrupciones de la máquina.

5.1.5 Canales de E/S



366. En los sistemas grandes (minicomputadoras, mainframes, ...) las tareas de E/S se encomiendan a hardware especial que se encarga de realizar las operaciones con registros de los periféricos o DMA. Estos elementos se denominan "canales de E/S", y además de implementar posiblemente un controlador DMA, disponen de toda una serie de funciones adicionales para gestionar a alto nivel la E/S con los periféricos.

367. Los canales se estructuran internamente como un pequeño ordenador cuyas únicas instrucciones fueran las de E/S, empleando acceso directo a memoria. Poseen su propio controlador y memoria en la cual reside el programa que ejecuta el control, llamado "programa de canal" que interpreta un lenguaje propio con el que puede interactuar con la CPU.

368. En los sistemas dotados de canales se encuentran habitualmente tres tipos de canales:

- *Canales multiplexados.* Estos se usan para conectar varios dispositivos de baja velocidad por un mismo canal compartido, mediante multiplexación temporal (entresacado de byte) o bien en modo ráfaga.
- *Canales selectores.* Estos suelen ser canales de alta velocidad en los que el canal se comparte por varios periféricos de alta velocidad en modo ráfaga.
- *Canales multiplexados por bloques.* Estos canales de alta velocidad operan en modo de multiplexación temporal pero con bloques de gran tamaño, propios de dispositivos de tipo bloque como los discos. Esto permite mantener una tasa constante de E/S mediante el uso de memorias auxiliares que actúen de búfer reteniendo los datos intercambiados con los periféricos mientras se envían/reciben los bloques de otros dispositivos del canal.

5.2 Buses e interfaces de E/S

369. Un bus es un camino que permite comunicar selectivamente un cierto número de componentes o dispositivos de acuerdo a ciertas normas de conexión. Su operación básica se denomina *ciclo de bus* que es el conjunto de pasos necesarios para realizar una transferencia elemental entre dos dispositivos conectados al bus.

ciclo de bus

370. Al conectar ciertos dispositivos con la computadora se requieren buses especiales denominados *interfaz de dispositivo* con registros y circuitos necesarios para: (i) reconocer la dirección del dispositivo, (ii) almacenar el estado de éste, (iii) proporcionar un espacio de almacenamiento a los datos y (iv) las señales de temporización y control.

interfaz de dispositivo

5.2.1 Características de los buses

371. Hay un conjunto de especificaciones y parámetros que cualifican el bus; las más importantes son:

- *Tipo de lógica:* positiva o negativa.
- *Grado de paralelismo:* número de bits que se envían simultáneamente por el bus, y puede ser:
 - Serie: paralelismo 1.
 - Paralelos: paralelismo igual o mayor que el ancho del dato.
 - Multiplexados: paralelismo menor que el ancho del dato.
- *Sincronización:* Síncrono o asíncrono.
- *Protocolo:* Forma de establecer el diálogo entre los dispositivos y el interfaz de dispositivo.

- *Temporización*: Puede ser
 - de ciclo completo: donde el bus se encuentra ocupado el tiempo necesario que ocupe la transferencia de datos.
 - de ciclo partido: se multiplexa el uso del bus, por entresacado de bytes o por bloques.
- *Bidireccionalidad*: Posibilidad de mantener transferencias en ambos sentidos.
 - unidireccional (*simplex*): La comunicación solo se establece en un sentido.
 - semibidireccional (*half-duplex*): Existe comunicación en ambos sentidos pero no a la vez.
 - bidireccional (*ful-duplex*): Existe comunicación en ambos sentidos simultáneamente.
- *Estrategia de control*: Puede establecerse un uso del bus *entre pares o maestro-esclavo*.
- *Velocidad y tasa de transferencia*: La velocidad mide el ancho de banda de funcionamiento del bus. La tasa de transferencia el número de bits o bytes por segundo.
- *Longitud máxima*: La separación máxima entre los dispositivos conectados.
- *Capacidad de conexión*: Número máximo de dispositivos conectables.
- *Soporte*: La tecnología física empleada (cable plano, coaxial, par trenzado, fibra óptica, etc).

simplex

half-duplex

ful-duplex

5.2.2 Niveles de especificación

372. El diseño de un bus de conexiones es una tarea compleja que conlleva repercusiones en el diseño de los dispositivos que más tarde se conectarán a él. Se establece una serie de normas dependientes del fabricante, que pueden llegar a convertirse en norma.

373. La especificación de un bus de conexiones se estructura en los siguientes niveles:

- *Mecánico*; Define el tipo de soporte, número de conductores y tipo.
- *Eléctrico (u óptico en su caso)*: Especifica el circuito equivalente cuyo comportamiento debe conservar el dispositivo.
- *Lógico*: Establece la interpretación de las líneas del bus.

- *Temporización básica y transferencia elemental*: Protocolo de señales en una operación de lectura / escritura, y método empleado en una transferencia elemental, respectivamente. Se realiza mediante cronogramas y diagramas lógicos.
- *Transferencia de bloque*: Operación básica en buses con transferencia de bloques, significado de las tramas de datos, etc.

5.2.3 Jerarquía de buses

374. Se puede establecer una jerarquía de buses, de modo similar a como se hace en el caso de las memorias. Características como velocidad de transferencia, ancho del bus, soporte (características mecánico-eléctricas), varían dependiendo de se trate de un bus entre componentes de una misma unidad funcional, entre unidades funcionales, entre el procesador y memoria, etc. hasta llegar a las redes de computadoras:

- **375.** *Buses internos de los circuitos integrados*: Son buses muy rápidos y de longitudes muy pequeñas y vienen integrados en el soporte del circuito. Suelen ser síncronos y la velocidad está muy bien determinado pues los retardos son un factor crítico de diseño. Suelen ser propiedad del fabricante.
- **376.** *Buses del circuito impreso*: Son algo más lentos que los anteriores, dependiendo de las características del soporte físico. Su longitud es del orden de centímetros. Suelen ser síncronos, y habitualmente conectan las unidades funcionales insertadas en una misma placa de circuito integrado, como el procesador, la memoria y los puertos de E/S. El diseño de estos buses deriva habitualmente del bus de conexiones del procesador de la computadora (*partillaje* de la CPU) y permite la conexión de circuitos de diverso origen. Un ejemplo de este tipo es el *bus de sistema* y subsistema. buses de sistema
- **377.** *Buses de panel posterior (backplane)*: Conectan los diferentes circuitos impresos que constituyen la computadora y su longitud varía desde fracciones menores que un metro hasta poco más de un metro. El diseño de estos buses es una norma de conexión que permite extender las funciones de la computadora por dispositivos construidos, posiblemente, por diferentes fabricantes; por esto se diseñan independientemente de la arquitectura de los dispositivos que se conectan, en la medida de lo posible. Un ejemplo de este tipo son los *buses de expansión*. buses de expansión

378. Los controladores de periféricos suelen insertarse en buses de expansión, aunque también es posible encontrar periféricos insertos directamente en los buses de expansión. En este caso se denominan periféricos locales.

379. En algunas ocasiones este bus es una simple modificación del bus del sistema del procesador persiguiendo buses de expansión muy rápidos. Este tipo de buses se conocen habitualmente como *buses locales*. buses locales

- **380.** *Buses de entrada/salida:* Estos buses conectan los controladores de los dispositivos con los dispositivos, y se diseñan en función de la naturaleza de la conexión y del dispositivo, para satisfacer necesidades de compatibilidad, longitud, velocidad, etc. Ejemplos de estos buses conectan modems, impresoras, localizadores, escáneres, transductores y actuadores, etc.
- **381.** *Redes de computadoras:* Estos buses se emplean para interconectar computadoras. Estas redes pueden ser de área local o de área amplia, y las características pueden ser muy diversas a todos los niveles.

Normalmente son buses de tipo serie donde la información se transmite en forma de paquetes, donde se especifica el origen y el destino de la información además de los datos transmitidos, acompañados de la información necesaria para detección y corrección de errores.

5.2.4 Ejemplos de buses

382. Buses de expansión:

ISA Bus: Industry Standard Architecture. Bus de expansión diseñado por IBM para computadoras personales. Su capacidad de transferencia es de 8 bits de datos, aunque el número de hilos es 62, donde se incluyen varias conexiones de tierra, línea de dirección de 20 líneas, etc.

ISA AT Bus: Expansión del bus ISA para adecuarlo a las posibilidades de 16 bits de las computadoras basadas en el procesador 80286 de Intel. Su velocidad de funcionamiento es 8 MHz. Mantiene la compatibilidad hacia abajo con el bus ISA.

EISA: Extended Industry Standard Architecture. Diseñado por un consorcio de fabricantes de equipamiento de hardware para las computadoras compatible con el IBM-PC para ampliar las capacidades de transferencia a 32 bits y 33 MHz de velocidad. Es compatible hacia abajo con el bus ISA AT y añade características útiles que permiten la configuración de las tarjetas por software y admite el modo de control maestro del bus.

MCA: Micro-Channel Architecture. Diseñado por IBM para renovar la arquitectura de las computadoras personales basadas en la familia x86. Emplea un bus de 32 bits de datos y es autoconfigurable además de otras ventajas. Es un bus propietario con baja aceptación.

HIPPI : High Performance Peripheral Interface para conexión de supercomputadoras de cálculo intensivo.

SCI : Scalable Coherent Interface (IEEE P1596-1992): Define la conexión punto a punto entre sistemas a velocidades de hasta 1 Gbyte/s para la implementación de sistemas multiprocesador de elevadas prestaciones respetando la sencillez de los sistemas de bus común.

Existen muchos otros buses de expansión y de sistema como VME Bus para la familia de Motorola 68K, VXI (una extensión industrial del bus VME), Nu-Bus para Apple MacIntosh, M-bus y S-Bus para estaciones de trabajo Sun, Unibus, FutureBus, Multibus, etc.

383. Buses de entrada/salida:

GPIB: General Purpose Interface Bus (IEEE 488): Permite la conexión de hasta 15 dispositivos como unidades de almacenamiento masivo, instrumentos de medida y de control e impresoras.

SCSI: Small Computer Standard Interface. Es un estándar de conexión de periféricos rápidos (discos duros, CD-ROM, escáneres, ...). Admite siete dispositivos y permite velocidades de transferencia de 5 Mbyte/s a 400 Mbyte/s según versiones.

Interfaz RS-232-C: Diseñada por el CCITT³ describe la forma de conectar un terminal a un módem, aunque se usa de forma generalizada para la conexión de computadores y periféricos, en forma serie punto a punto. Las especificaciones eléctricas y mecánicas son variadas y permiten conectar periféricos a distancias de decenas de metros.

384. Buses locales:

VL Bus: Ideada por un consorcio de empresas de construcción de expansiones para IBM-PC, con el fin de dar una alternativa económica a EISA y MCA. PCI: Peripheral Component Interconnected, promovida por Intel como alternativa y mejora a VL-Bus.

³CCITT: Comité Consultivo Internacional de Telegrafía y Telefonía.

Clasificación de periféricos

Índice General

6.1	Introducción	104
6.1.1	Codificación y periféricos	104
6.2	Dispositivos de almacenamiento permanente	105
6.2.1	Discos	107
6.2.2	Discos flexibles	108
6.2.3	Discos duros	110
6.2.4	Cintas magnéticas	111
6.2.5	Discos ópticos	113
6.3	Impresoras	114
6.3.1	Características	114
6.3.2	Impresoras de molde y composición de caracteres	114
6.3.3	Impresoras de composición de páginas	116
6.3.4	Lectora/perforadora de tarjetas	117
6.4	Visualización	118
6.4.1	Modos de trabajo	118
6.4.2	Monitores	119
6.4.3	Memoria de refresco	119
6.5	Terminal interactivo	120
6.6	Otros dispositivos de E/S	120

6.1 Introducción

385. Un periférico es cualquier componente de un sistema informático en el que se delegan funciones de envío y recepción de datos mediante de un mecanismo de entrada salida.

Dentro de esta definición se incluyen dispositivos que comunican el procesador con el mundo exterior y sistemas que permiten almacenar o archivar información de modo persistente auxiliando a la memoria central en capacidad o tiempo de permanencia (memoria masiva o auxiliar).

386. Entre los dispositivos de almacenamiento persistente se puede diferenciar entre aquellos cuyo fin es servir de almacenamiento definitivo (a largo plazo), como las unidades de cinta, disquetes, cartuchos, CD-ROM, y los que se diseñan como memoria secundaria de la memoria central.

387. En el caso de los periféricos de corte general, cabe distinguir entre aquellos especializados en el *interfaz hombre-máquina*, entre los que se incluyen los terminales alfanuméricos, impresoras, sintetizadores, etc. y aquellos dedicados a tareas de interfaz con otros dispositivos, pudiendo ser éstos otras computadoras o bien sensores y actuadores físicos, de utilidad en el control de automatismos.

388. Los dispositivos de E/S se pueden clasificar en grandes bloques, como sigue:

- Dispositivos de almacenamiento permanente: Discos, cintas, etc.
- Unidades de interfaz hombre-máquina: que incluyen monitores de visualización, teclados y controles de tipo teclado, e impresoras.
- Unidades de E/S para dispositivos físicos: donde se incluyen modems, escáneres, Conversores A/D, D/A, etc.

6.1.1 Codificación y periféricos

389. Bien sea el dispositivo útil para la interacción humana con la computadora, la transmisión de la información, o el almacenamiento, se emplean diversos códigos que sirven para representar la información externa a la computadora.

En el caso de impresoras, unidades de visualización, teclados, etc. se diseñan códigos de entrada/salida, que manejan una *representación externa* de la información inteligible por el ser humano, ya sea un código ASCII, gráficos, iconos o señales audibles.

390. En el caso de otros dispositivos, se emplean códigos específicos para cada tarea, con características como: eficiencia y robustez en la transmisión y almacenamiento de información; robustez en el caso de los códigos de barras y caracteres OCR.¹; u otras cualidades en códigos más específicos. La descripción de estos códigos va desde el conjunto de símbolos representables hasta las características materiales.

¹OCR: Optical Character Recognition.

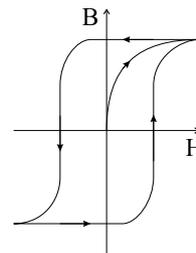
interfaz hombre-máquina

representación externa

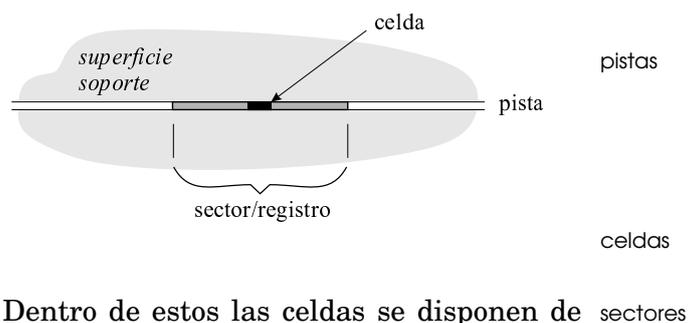
391. Los periféricos se encargan habitualmente de las tareas de codificación y decodificación de los sistemas de representación externa de la información, a la representación interna de la información en la computadora.

6.2 Dispositivos de almacenamiento permanente

392. La mayoría de los dispositivos de almacenamiento masivo emplean una superficie recubierta de un material ferrimagnético. Estos materiales, entre otros, pueden presentar un campo magnético (B) en ausencia de excitación magnética (H). Mediante algún dispositivo, se puede modificar y medir la magnitud de este campo magnético residual que permanece aún cuando se extraiga el dispositivo del sistema informático.

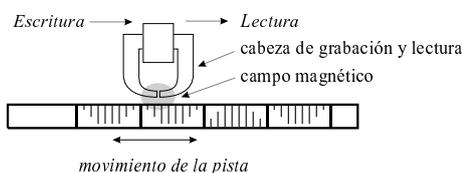


393. Esta superficie, recubierta de un material del tipo descrito, se organiza en *pistas* ("tracks") que discurren de forma paralela. Estas pistas se localizan por un dispositivo de localización y conciernen a la disposición lógica de la información en el "medio". Las pistas se componen de *celdas* que contienen datos individuales y que se pueden organizar en *sectores* o registros. Dentro de estos las celdas se disponen de modo secuencial.



394. La grabación de estos medios se produce mediante un electroimán; en él distingue una abertura denominada entrehierro, donde las líneas de flujo magnético se dispersan en el aire, y si se aproxima suficientemente un material ferrimagnético, la orientación de su magnetización residual puede variar si el campo inducido es suficientemente grande.

Este electroimán o cabezal puede servir tanto para la magnetización del medio, a través de un arrollamiento por el que se hace circular una corriente eléctrica, como para percibir un campo magnético existente.



En este caso el arrollamiento se inspecciona por un circuito electrónico que proporciona señales eléctricas a la lógica de control del dispositivo.

Con el fin de almacenar una serie de datos en el medio, se imprime un movimiento a la superficie. La banda de superficie a la que accede el cabezal, cuando éste permanece estático, y se mueve la superficie es precisamente una pista. Algunos cabezales permiten seleccionar una de varias pistas, e incluso si se acumulan varios cabezales, leer varias pistas a la vez.

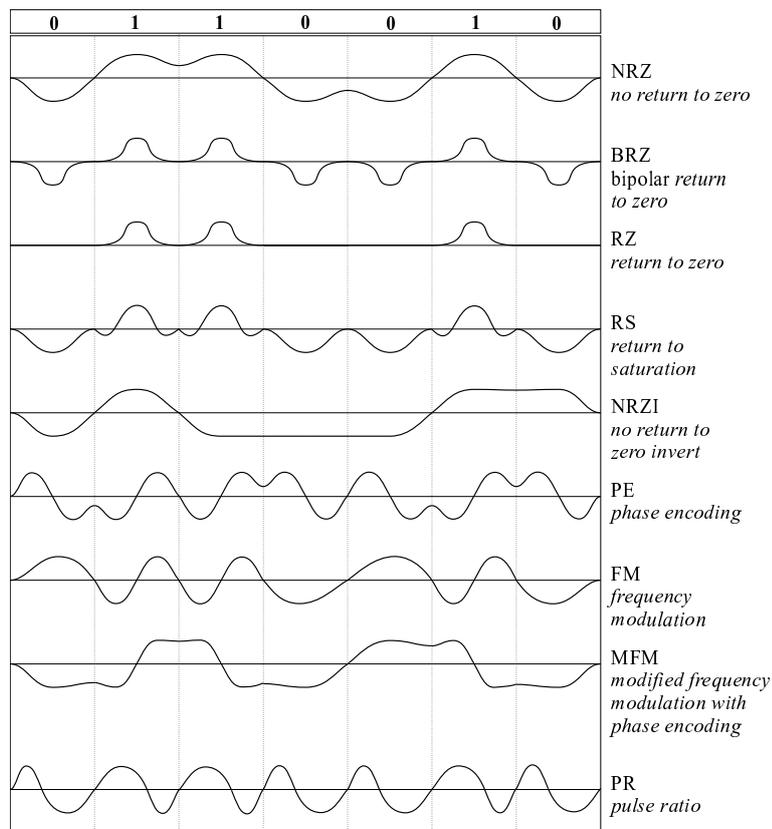


Figura 6.1: Algunos códigos de grabación magnética.

Códigos de grabación magnética

395. Dado que la superficie de grabación acoge una señal de carácter analógica, es necesario diseñar un código que especifique a nivel físico, cómo ha de estar almacenada la información. El procedimiento más sencillo es magnetizar cada celda de modo que el 0 se represente como una magnetización de un signo y el 1 como una magnetización de signo contrario (NRZ). Al leer o escribir la información, se considera la velocidad de desplazamiento de la pista y un reloj externo que sirve para medir la anchura de la celda.

396. En la figura ?? se resumen algunos códigos de grabación. Al ser el medio final de almacenamiento analógico, el aspecto se aprecia suavizado, desde el momento en que se apura el ancho de banda del dispositivo. Algunos de estos códigos aprovechan las cualidades analógicas de la señal (magnitud, frecuencia, fase) para aumentar la densidad de grabación.

Modo de acceso y tiempo de acceso

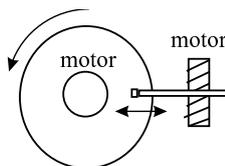
397. Para acceder a la información almacenada se puede proceder de dos modos, que en algún momento pueden superponerse en niveles diferentes:

- *Secuencial*. Cuando para acceder a un elemento de información es necesario que el cabezal lea uno a uno los elementos que median entre su posición actual hasta llegar al elemento requerido. Secuencial
- *Directo*. Cuando el cabezal puede posicionarse directamente sobre cualquier elemento al que se desee acceder (leer/escribir). Directo

398. El tiempo empleado en la operación de cualquier dispositivo resulta una acumulación de tiempos de cada dispositivo mas el ancho de banda del canal que se emplea para la comunicación. Ente ellos, se denomina *tiempo medio de acceso* al tiempo que se invierte, por término medio, en acceder a cualquier registro físico del dispositivo.

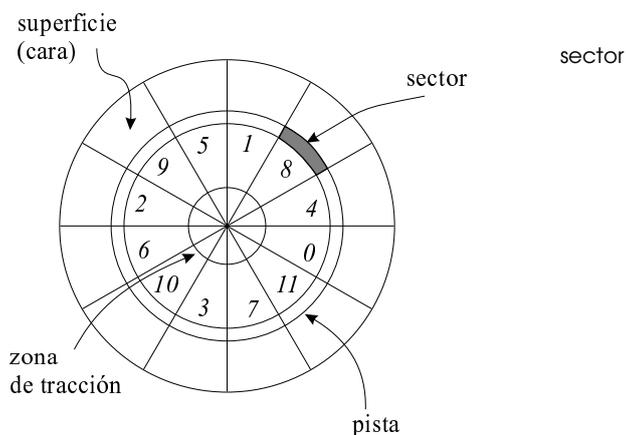
6.2.1 Discos

399. Los discos son la forma más utilizada de almacenamiento masivo. En estos dispositivos, se dispone de una o más superficies de forma circular. La cabeza de lectura es capaz de moverse sobre un radio de este círculo, desde el borde de éste hasta algún punto cercano al centro, gracias a un motor paso a paso que se gobierna electrónicamente.



400. En los discos las pistas tienen forma circular, y el la banda de la superficie, que recorre un cabezal cuando se hace girar a la superficie en torno a su eje. Habitualmente la velocidad de rotación es constante. La numeración de pistas suele partir desde 0 a $N - 1$ desde el borde exterior, en el caso de los discos flexibles y puede partir desde el punto medio del radio en los discos duros.

401. Las pistas particionan en un número habitualmente fijo de *sectores* de circunferencia que juegan el papel de registro que coleccionan un conjunto de celdas de grabación. Cada uno de estos sectores puede almacenar desde 512 bytes en adelante. El orden de numeración de los sectores puede ser consecutivo, aunque es habitual que se numeren con un factor de entrelazado (*interleave*) que optimiza los tiempos de latencia en el acceso a sectores consecutivos.



402. Dada esta disposición, la densidad de grabación en las pistas interiores suele ser mayor que en las exteriores. Para paliar estos efectos se puede recurrir a modi-

ficar la velocidad del disco según el cabezal accede a las pistas interiores o, lo más habitual, disminuir la sectorización en estas mismas pistas.

403. En el tiempo medio de acceso influye la velocidad de desplazamiento transversal del cabezal, la velocidad de rotación del disco y el factor de entrelazado (entre otros).

- El tiempo de búsqueda es el que se invierte en posicionar la cabeza en la pista correcta.
- El tiempo de latencia es el tiempo necesario para que el sector requerido aparezca bajo el cabezal, una vez se ha encontrado la pista.

Formato de un disco

404. Antes de poder disponer de modo efectivo de un disco, es necesario realizar sobre el una operación conocida como formateado. El formateado de un disco se compone de dos etapas: la primera se conoce como “formateo de bajo nivel” donde se organiza la superficie del disco y la segunda “formateo de alto nivel” o inicialización del sistema de archivos.

405. En el formateo de bajo nivel se realiza la organización lógica de la superficie del disco, y supone:

- Escritura de inicialización sobre cada sector.
- Verificación de pistas y sectores, y construcción de una tabla de sectores sanos, donde se excluyen los sectores deteriorados e inutilizables

406. En el formateo de alto nivel, y dependiendo de la finalidad del disco y el sistema operativo que lo gestiona, se realiza:

- Configuración de la unidad: nombre, formato, fecha, etc.
- Construcción de la tabla de contenidos y sectores libres.
- Grabación de información específica: botado, configuración del sistema, etc.

Esta información se suele grabar sobre pistas y sectores predefinidos de antemano, de modo que el sistema operativo sea capaz de obtener esta información para conocer el tipo de formato del disco y otros parámetros.

6.2.2 Discos flexibles

407. Un disco flexible, disquete o “*floppy disk*” es un disco recubierto por ambas caras de un material ferrimagnético y que viene provisto de una funda flexible en el caso de los discos de 8” y 5”1/4 y de plástico rígido en los discos de 3”1/2.

408. La característica principal de un disco flexible es que puede separarse físicamente del sistema. La unidad de lectura y grabación está provista de un alojamiento que permite intercambiar fácilmente el medio. A cambio los mecanismos

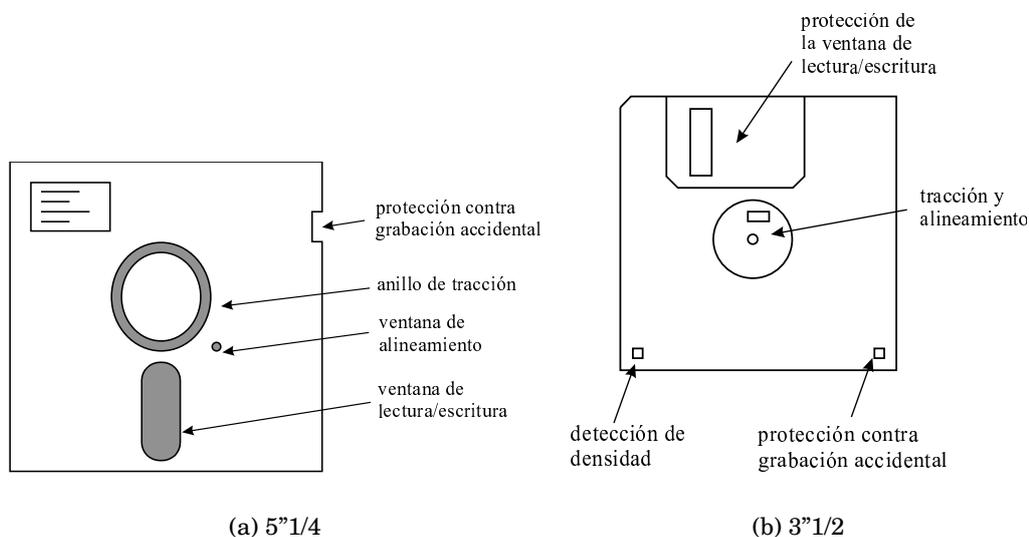


Figura 6.2: Aspecto de un disco flexible.

de alineamiento han de ser muy tolerantes y como consecuencia: (i) el cabezal de lectura/escritura está en contacto con la superficie, (ii) la velocidad de rotación y búsqueda son reducidas (300 r.p.m.), y (iii) las densidades de grabación son pequeñas. Un disquete puede almacenar hasta 2 Mbytes.²

409. Los discos flexibles disponen de una ventana de lectura/escritura en la funda por la cual es posible acceder a la superficie, y que en el caso del disco de 3 1/2 va protegido con una cubierta metálica deslizante.

La tracción se realiza mediante una superficie que presiona en el anillo de tracción del disco, o bien mediante un acoplamiento mecánico sobre una superficie metálica.

410. Existe un método de fijar el origen de numeración de los sectores, que en los discos de tracción por presión se realiza mediante una perforación en la superficie del disco y una célula fotoeléctrica, y en el caso de acoplamiento mecánico, la superficie metálica contiene el sistema de alineamiento.

411. En cualquier caso, mediante un sistema mecánico, cada tipo de disco dispone de un sistema que permite proteger los datos contra la grabación accidental.

²Con la tecnología actual existen unidades de almacenamiento que obtienen hasta 20Mbytes con discos del mismo aspecto pero de características de altas prestaciones.

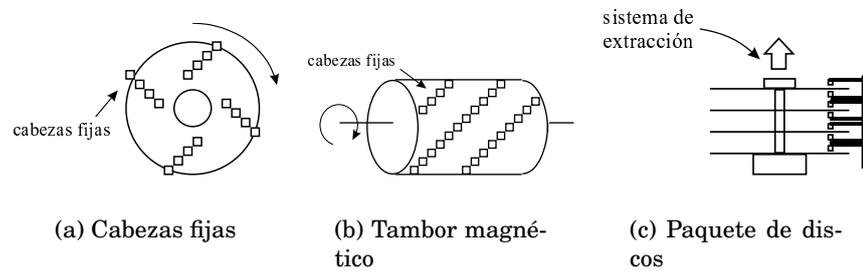
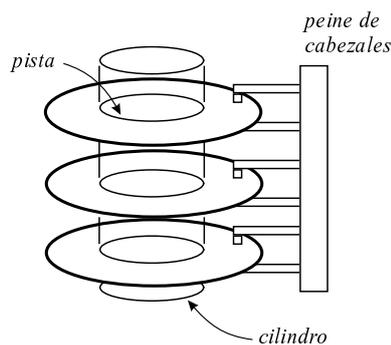


Figura 6.3: Algunos tipos de discos duros.

6.2.3 Discos duros

412. Un disco duro (*“hard disk”*) es un dispositivo formado por una o más superficies ferrimagnéticas montadas sobre un soporte rígido. Las dimensiones varían desde 5”1/4 hasta 1” siendo, por lo general de diseño propietario y diferentes entre si, por lo que no se puede hablar de formatos estándar. La capacidad de estos dispositivos se encuentra actualmente desde los centenares a las decenas de miles de Mbytes.

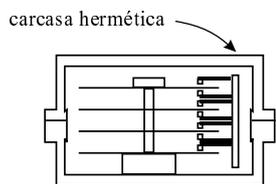
La característica principal de estas unidades es una alta velocidad de rotación (aprox. 3000 r.p.m.); los cabezales de lectura escritura operan a una pequeña distancia de la superficie del disco que puede llegar a ser del orden de decenas de micras. En consecuencia la sectorización y la densidad de pistas puede ser mucho mayor, aunque en contrapartida exige un comportamiento mecánico mucho más preciso y ausencia de vibraciones externas.



413. El conjunto se organiza en superficies y cilindros. Un cilindro es el conjunto de pistas que pueden ser leídas al mismo tiempo por las cabezas de la unidad, y está formado por la intersección de las superficies útiles del disco y un cilindro imaginario concéntrico al eje del disco. El conjunto de cabezales de lectura/escritura se dispone en una estructura denominada “peine” que sitúa las cabezas de modo simultáneo sobre un mismo cilindro.

414. Los bloques del disco se numeran correlativamente, aunque la localización física de cada uno de ellos obedece a tres índices: número de superficie, número de cilindro y número de sector en la pista. Si el tamaño de cada bloque no es muy grande (512 bytes) las transacciones de lectura/escritura se pueden realizar en grupos de sectores (*“clusters”*) cuyo tamaño puede ir desde 1 Mbyte a 4 u 8 Mbytes.

415. La tecnología de este tipo de dispositivos ha evolucionado considerablemente. En los primeros tiempos se empleaban discos y tambores magnéticos de cabezas fijas, donde el tiempo de búsqueda es nulo y se buscaba simplicidad mecánica.



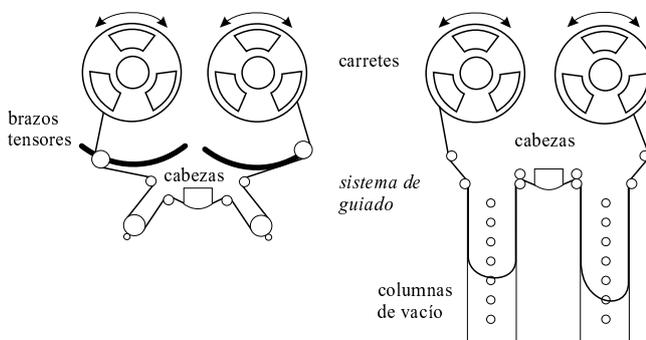
Más tarde aparecen los discos duros extraíbles de 2 superficies y posteriormente de más de un disco (“disk pack”). La capacidad de estas unidades es muy baja y el tiempo medio entre fallos muy alto, en comparación con los discos actuales, con lo cual era necesario reducir el coste de la unidad permitiendo el intercambio de los discos, y cuando fuese necesario de las cabezas.

416. En la actualidad, el tipo de disco más utilizado es Winchester (IBM), donde las especificaciones técnicas se hacen mucho más rigurosas y se encapsula el disco en una carcasa hermética conteniendo un gas inerte libre de partículas. Con ello, se eliminaba la mayor razón de fallos que consistía en la colisión del cabezal con partículas de polvo sobre la superficie del disco. El tiempo medio entre fallos se puede situar en el orden de los centenares de miles de horas.

6.2.4 Cintas magnéticas

417. Por lo general las cintas magnéticas son superficies ferrimagnéticas donde un cierto número de pistas discurren paralelamente sobre una cinta de una longitud regular (desde 10 hasta más de 100 metros).

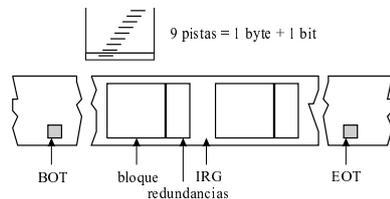
El dispositivo de lectura/escritura hace pasar la cinta en contacto con uno o varios cabezales multipista, a una velocidad y tensión constantes. El modo de acceso a cada bloque o registro es secuencial, y los tiempos de acceso son muy altos, por lo cual solo son aptas como medios de almacenamiento masivo permanente para copias de respaldo y distribución de software.



418. En las cintas de carrete abierto, el dispositivo de guía de la cinta es externo y la cinta se hace circular por dos brazos tensores o dos columnas de vacío que mediante tracción y succión, respectivamente, mantienen la tensión constante. El dispositivo de arrastre mantiene la velocidad constante, y sendos motores bajo los carretes se encargan de soltar y recoger la cinta.

Mientras tanto, en las unidades de cartucho, la cinta se encuentra confinada en una carcasa de plástico o metal y plástico. Las guías se encuentran en la misma, y los cabezales de lectura/escritura y el mecanismo de tracción son externos. De este modo la manipulación es más sencilla y las tolerancias mecánicas menores.

419. En cada cinta, las pistas discurren paralelamente a su longitud, y se marca el comienzo y el final de la zona útil mediante una zona transparente o un recuadro espejado, de modo que sea legible por un mecanismo óptico.³



Los bloques de información suelen tener una longitud fija, en función del formateo realizado. Cada registro se acompaña de un bloque compuesto de información conforme a un código de redundancia que sirve para comprobar la integridad de la información del bloque y se separa del siguiente

bloque por un hueco en blanco de longitud fija⁴

420. Las características principales de las cintas de carrete abierto son:

- Codificación de grabación: NRZI, MNRZI o PE.
- El ancho de la cinta es usualmente $\frac{1}{2}$.
- El formateo se puede realizar previamente o en tiempo de grabación.
- La longitud habitual es: 600, 1600, 2400 o 4800 pies.
- La densidad de grabación habitual es: 300, 556, 800, 1600 o 3200 bpi (bits por pulgada).
- La capacidad máxima es = longitud total * densidad (1 pie=12 pulgadas).

421. Las características habituales de las cintas de cartucho son:

- El ancho es $\frac{1}{4}$ o $\frac{1}{8}$.
- El bobinado puede ser "sin fin".
- Los bloques son de gran longitud y el IRG muy pequeño.
- La densidad de grabación es: 6400 o 8000 bpi.
- Formatos habituales: TK50, QIC, ...

Es un medio más rápido y denso que la cinta de carrete abierto pero menos portable.

Sistema vídeo

422. Existen unidades de cintas de cartucho que emplean la tecnología de grabación de vídeo, donde el sistema de grabación es de tipo helicoidal. La ventaja de estas unidades es un ancho de banda mayor que suele acompañarse de sistemas de compresión de datos adicionales.

Son formatos habituales de cintas: Hexabyte (Sun) y DDS, son capacidades de hasta 9 Gbytes en el último caso.

³BOT: Begin of tape (comienzo de la cinta). EOT: End of tape (final de la cinta).

⁴IRG: Inter record gap (hueco entre registros).

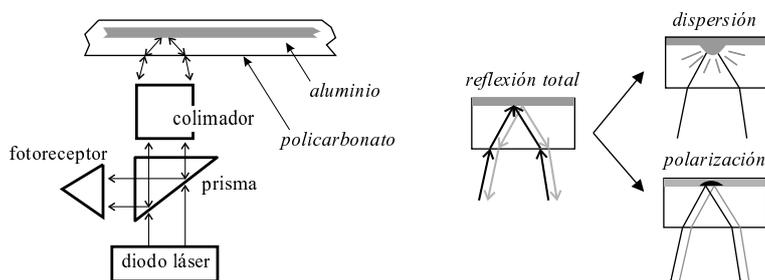


Figura 6.4: Almacenamiento en disco óptico.

6.2.5 Discos ópticos

423. Los discos ópticos son dispositivos de almacenamiento en el que la información se encuentra grabada en un soporte reflejante que puede ser leído por un procedimiento óptico, sin contacto mecánico, con unas tolerancias mecánicas relativamente poco exigentes. Un CD-ROM alcanza una capacidad de 640 Mbytes útiles.

La superficie del disco es leída a través de un haz láser y la información se codifica por medio de cambios en la capacidad del substrato para reflejar la luz que se le envía. La información se almacena en el disco en una larga espiral y la densidad de almacenamiento es muy elevada.

424. Existen dos tecnologías de grabación predominantes: óptica y magneto-óptica. En la primera la información se codifica en el disco como la presencia y ausencia de pequeñas protuberancias, éstas dispersan la luz incidente, de modo que el fotodetector es capaz de decidir la ausencia o presencia de protuberancia midiendo la intensidad de la luz devuelta por el disco.

En la segunda el substrato tiene un comportamiento ferromagnético y la información se codifica como presencia o ausencia de zonas alteradas magnéticamente, que polarizan la luz incidente, de modo que absorben la mayoría de la luz que incide en estas zonas, devolviendo solo aquella próxima a un plano de polarización. El sistema de medición es el mismo que en el caso anterior.

425. Los discos CD-ROM⁵ de tecnología óptica provienen grabados de factoría, o se pueden generar con unidades especiales, siendo solo posible realizar una grabación, en este caso se denominan CD-WORM⁶

Los discos basados en tecnología magneto-óptica son susceptibles de ser grabados más de una vez (CD-Regrabable), por unidades de un precio más moderado. Se están convirtiendo actualmente en el medio preferido de copias de respaldo.

426. A partir de estas tecnologías surgen modificaciones en la codificación y la tecnología láser que conduce a la invención del disco DVD⁷ que incluye las siguientes mejoras: menor longitud de onda del láser (puntos más pequeños), tecnología de enfoque del láser que permite varios substratos o superficies por cara, avances

⁵Compact Disc-Read Only Memory.

⁶CD-Write Once Read Many -escribir una vez, leer muchas.

⁷Digital Versatile Disk.

en la codificación de los datos y el software de procesamiento de datos, y aprovechamiento de ambas caras del disco. Como consecuencia de ello se puede alcanzar capacidades de 14,7 Gbytes.

6.3 Impresoras

427. Bajo el término impresoras se incluyen los dispositivos destinados a obtener un registro físico de la información y que en los orígenes incluía únicamente unidades de teletipo. Este término abarca también las unidades de impresión y perforación de tarjetas perforadas.

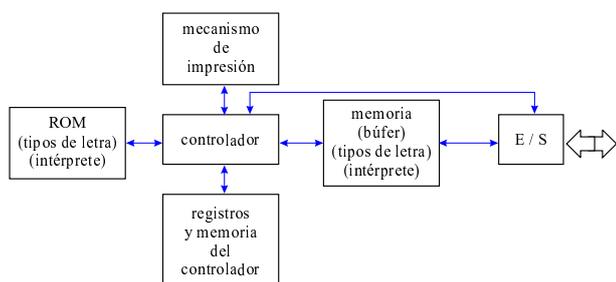
6.3.1 Características

428. Las impresoras se pueden clasificar genéricamente según el método, la unidad y la calidad de impresión:

- Según el método de impresión:
 - Por molde: de margarita, teletipo, de rueda.
 - Por composición: matriciales, de chorro de tinta, láser.
- Según la unidad de impresión:
 - De caracteres.
 - De líneas.
 - De páginas.
- Según la calidad de impresión:
 - Calidad de borrador y buena calidad (draft).
 - Calidad alta.
 - Calidad de imprenta.

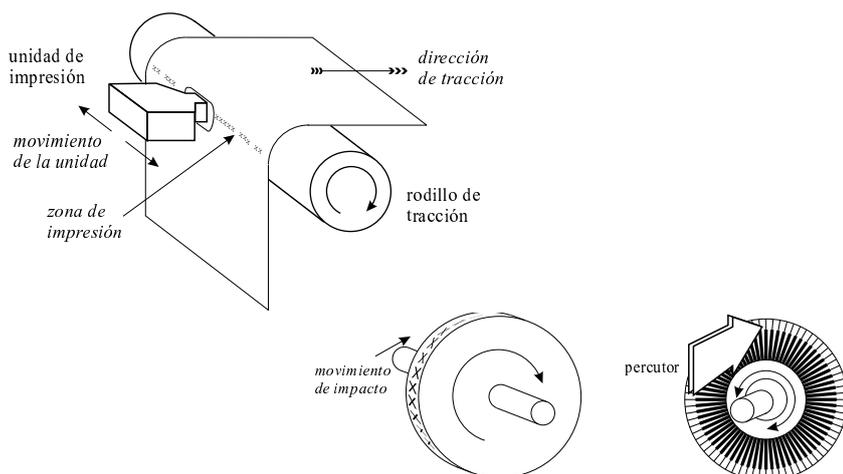
6.3.2 Impresoras de molde y composición de caracteres

429. El esquema interno de una impresora de molde y composición de caracteres, salvo ligeras variaciones consta de un mecanismo de impresión y un controlador que realiza el control directo del hardware y las operaciones de entrada salida, decodificando e interpretando las instrucciones de impresión enviadas por la computadora.



Éste controlador dispone de una memoria de solo lectura con el intérprete, el programa de control y la descripción de la forma de los caracteres (en el caso de composición de caracteres) y una memoria de propósito general donde se alojan los datos recibidos así como intérpretes y formas de caracteres suplementarias.

430. El mecanismo de impresión en esta familia es simple: una unidad de arrastre desplaza el papel mediante un motor paso a paso, de modo que cierto dispositivo de impresión puede acceder a la zona de impresión con un movimiento longitudinal, permitiendo secuencias de caracteres para formar una línea.



431. Impresoras de ruedas y margarita - Este tipo de impresoras, junto con los teletipos, funciona haciendo impacto con el molde de un carácter sobre el papel y poniendo entre medias una cinta entintada, con lo cual se transfiere al papel la tinta con la forma del molde. Empleando cinta de carbón la impresión es de muy buena calidad. Sus inconvenientes son: poca versatilidad en el juego de caracteres, imposibilidad de gráficos y baja velocidad (aprox. 80 cps -caracteres por segundo).

432. Impresoras matriciales - Las dos grandes familias de este tipo de impresoras son: impresoras matriciales o de agujas e impresoras de transferencia de tinta. Los caracteres se obtienen por composición, mediante una matriz de puntos, que habitualmente pueden solaparse.

Las impresoras de agujas emplean un cabezal de puntos en el que se pueden seleccionar cuales puntos van a realizar un impacto contra la superficie del papel. Del mismo modo que en las de margarita, se interpone una superficie entintada para transferir la tinta. Los cabezales emplean desde 9 a 24 agujas en forma lineal o diamante. Al avanzar el cabezal, va componiendo el carácter requerido y la línea de la zona de impresión. La velocidad de impresión puede ser de algunos centenares de caracteres por segundo, se pueden componer gráficos, pero la calidad (aunque puede ser aceptable) no llega a la de las impresoras láser.

En algunas ocasiones se emplea también un sistema de papel termosensible, y un cabezal de agujas a una temperatura que permite impresionar los puntos

directamente sobre el papel, sin necesidad de emplear una cinta entintada. Este tipo de impresoras son muy robustas y carecen prácticamente de mantenimiento aunque su velocidad es muy pequeña.

433. Impresoras de transferencia de tinta - Este tipo de impresoras disponen de una unidad de impresión capaz de transferir tinta al soporte, sin necesidad de impactar con él. Entre éstas se conocen: impresoras de inyectores de tinta, impresoras de burbuja de tinta e impresoras de transferencia térmica de tintas sólidas.

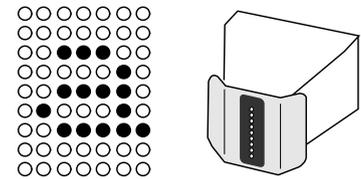


Figura 6.5: Matriz de puntos.

Las impresoras de inyección de tinta expulsan gotas de tinta a presión sobre el papel, siendo muy silenciosas y permitiendo imprimir simultáneamente con tinta de varios colores. Alcanzan más calidad que las impresoras de agujas con calidad color, y su nivel de ruido es mucho menor.

Las impresoras de burbuja emplean un sistema de proyección de tinta de tipo electrostático, con lo que consiguen una calidad ligeramente superior a las de inyección de tinta. Por último, las impresoras de transferencia térmica de cera, emplean un sistema electrostático y térmico para proyectar y fijar la tinta sobre el papel, depositando una capa de material sólido que permite incluso realizar mezcla de colores sobre el soporte.

6.3.3 Impresoras de composición de páginas

434. En las impresoras láser se compone un mapa de bits que representan los puntos entintados en la página (o una banda de ésta), según un esquema de matriz de puntos a tamaño de página. La complejidad de estas impresoras determina que en la mayoría de los casos utilicen internamente microprocesadores de propósito general similares a los usados en pequeñas estaciones de trabajo y cantidades apreciables de memoria central.

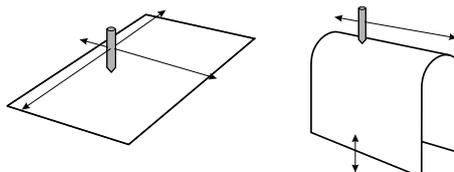
Por lo general, si se envía texto sencillo a estas impresoras, éstas se encargan de dar formato a la impresión, pero lo habitual es emplear algún lenguaje específico de composición de página, como por ejemplo PostScript o PCL 5. Estos lenguajes incluyen las primitivas de programación que dirigen la formación del documento impreso y permiten cargar tipos de letra nuevos y gráficos de gran calidad (hasta 800 dpi -puntos por pulgada).

La velocidad máxima que pueden alcanzar este tipo de impresoras varía en función de la complejidad de la tarea pero se encuentra entre 6 y 40 páginas por minuto.

435. El funcionamiento interno se basa en la construcción del patrón de bits de la imagen sobre la superficie de un cilindro de material semiconductor. Al añadir una sustancia en polvo, llamada tóner, se adhiere sobre aquellos puntos que al ser iluminados poseen una cierta carga eléctrica. Esta mancha se transfiere al papel y posteriormente se fija aplicando calor.

436. Filmadoras - Son un caso particular de este tipo de impresión son las filmadoras, que empleando un sistema similar en algunos aspectos imprimen directamente sobre los fotolitos que se emplearán para realizar la impresión de un libro, revista o cartel sobre offset. Estos dispositivos alcanzan densidades de punto de 3000 puntos por pulgada o más y su uso se restringe a talleres de composición e imprenta.

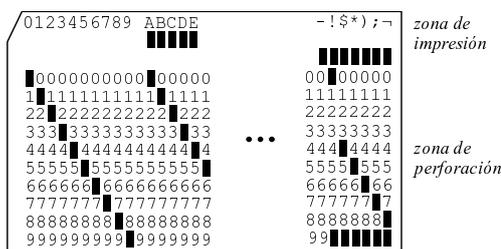
437. Trazador gráfico - También se conoce como “*plotter*” y compone imágenes y texto mediante trazos continuos al modo como lo haría un delineante. Suelen emplearse para obtener planos e imprimir (y cortar) sobre superficies de dimensiones grandes.



El método de impresión habitual es el empleo de plumas de diferentes colores y grosores, aunque actualmente se utilizan también con éxito inyectores y proyectores de tinta electrostáticos. En cuanto al mecanismo de tracción empleado varía, en función de que sea un trazador de *sobremesa*, en cuyo caso las plumas se mueven en dos direcciones sobre una superficie plana, de *rodillo*, en la que la pluma se mueve en una dirección y el papel se desplaza tangencialmente, o de tambor, similar al anterior.

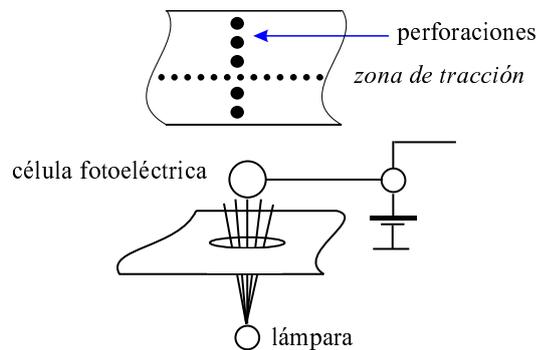
6.3.4 Lectora/perforadora de tarjetas

438. La entrada/salida en las computadoras de primera y segunda generación era la tarjeta perforada. Basada en el sistema de Jacquard del siglo XIX, consta de una cartulina con columnas de perforaciones en forma de tabla, donde cada columna codifica un dato.



dimensiones: 80 columnas - 17.85 cm
12 filas - 7.68 cm

Se suele emplear el código de Hollerit, el mismo de las máquinas tabuladoras. En la parte superior hay una zona donde se imprime el contenido de cada columna, en la parte inferior el código -la última columna se reserva para comprobar errores. Las tarjetas se perforaban desde una unidad dotada de teclado. Su uso iba desde codificación de datos hasta almacenamiento de programas.

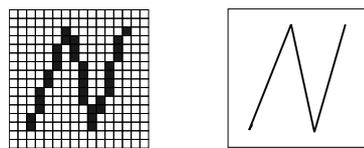


439. De menor uso, pero funcionando por el mismo principio, están las “tiras perforadas” que se utilizaban en origen para controlar teletipos a partir de datos telegráficos, y que funcionaban mediante la detección de perforación por células fotoeléctricas.

6.4 Visualización

440. El sistema de visualización más difundido es el tubo de rayos catódicos (TRC) aunque existen otros tipos de visualizadores como el cristal líquido y el plasma. Y constituye, junto al teclado, la base de los sistemas de interfaz hombre-máquina interactivos.

441. El funcionamiento del TRC se basa en que ciertas sustancias emanan luz al ser objeto de un haz de electrones. Un TRC es un tubo vacío donde en un extremo se emite un haz de electrones y en el otro se encuentra una pantalla de fósforos electroluminiscentes. Se puede controlar tanto la intensidad del haz como su deflexión. Se usan fósforos que emiten durante un periodo de tiempo corto, y periódicamente se refresca la imagen.



442. Existen dos modos de visualización: vectorial y barrido. En el primero el haz de electrones construye contornos modulando el haz a lo largo de éste. Este método es muy nítido, pero el tiempo de representación crece con la

complejidad del gráfico así como la cantidad de memoria necesaria.

En el método de barrido (*raster*) se hace recorrer el haz de electrones una trama horizontal, activando aquellos puntos de la pantalla (*pixels*) que conforman la figura (*bitmap* o *pixmap*). Este método es menos nítido, pero el tiempo de representación y la memoria requerida son constantes.

6.4.1 Modos de trabajo

443. Los modos de trabajo de un sistema de visualización son dos: modo texto, y modo gráfico. El primero permite solo mostrar caracteres de una tabla predefinida, y el segundo permite tanto gráficos como texto. Estos modos se pueden dar por separado o conjuntamente en un sistema de vídeo.

6.4.2 Monitores

444. Los monitores habituales son digitales o analógicos. En los primeros se controla digitalmente la intensidad de luz de cada color que compone cada pixel, y en los segundos la información de intensidad es analógica. El conector es DB9 en el primero y DB15 en el segundo.

Nombre del adaptador	Color	Texto (ppc)	Gráficos (colores)
MDA Monochrome Display Adapter	Mono	25x80 (9x14)	-
HGC Hercules Graphics Adapter	Mono	25x80 (9x14)	720x348 (2)
CGA Color Graphics Adapter	Color	25x80 (8x8)	320x200 (4) — 640x200 (2)
EGA Enhanced Graphics Adapter	Color	25x80 (8x14)	640x350 (16)
VGA Video Graphics Adapter	Color	25x80 (9x16)	640x480 (16) — 320x200 (256)
SVGA Super VGA	Color	(ídem)	800x600 (16) y más
PS/2 IBM 8514/A	Color	(ídem)	1024x768 (256)

Tabla 6.1: Modos de vídeo de IBM-PC

445. Existe otra clasificación atendiendo a la frecuencia de barrido: mono y multifrecuencia. En los primeros solo se contempla una frecuencia de barrido, con lo que van emparejados al modo de visualización. En los multifrecuencia soportan varias frecuencias de barrido horizontal y de cuadro, con lo que soportan varios modos.

446. Los monitores monocromos suelen trabajar con un barrido horizontal de 18.4 KHz y 70 Hz de barrido vertical. En estos rangos se pueden encontrar las frecuencias de barrido de los monitores de color mejorado y analógico.

6.4.3 Memoria de refresco

447. Toda unidad de visualización dispone de una memoria donde se almacena la información a visualizar, y que se lee a la misma velocidad que requiere el monitor. En torno a 20 ms es el tiempo requerido.

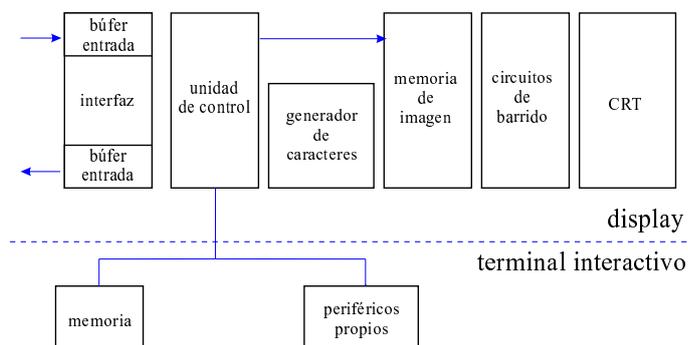
448. Desde un punto de vista lógico, se considera dividida en planos, conteniendo cada plano un bit por cada punto del visualizador, así una memoria de 4 planos permite 16 colores o tonos de gris distintos, una de 8 planos 256, etc. La resolución horizontal y vertical del modo de representación y el número de planos proporcionan la memoria total de refresco necesaria para representar un modo.

En los modos de texto, la memoria de refresco solo requiere almacenar los símbolos que se representan, siempre que se incorpore al controlador de vídeo un generador de caracteres.

6.5 Terminal interactivo

449. Un terminal interactivo es un dispositivo de visualización y un sistema de entrada de información de usuario. Suelen incorporar un monitor, un teclado para la introducción de texto, y ocasionalmente un dispositivo localizador, como un lápiz óptico, un ratón o similar.

Los dispositivos de localización permiten controlar la posición en pantalla de un elemento denominado cursor, que permite interactuar con el interfaz de usuario de las aplicaciones, seleccionando elementos gráficos o de texto.



450. Los terminales interactivos pueden encontrarse como sistemas aislados, que se conectan a un sistema mediante un canal de comunicación serie o una red, o bien formando parte de la computadora. En este caso, el control del dispositivo lo realiza la propia unidad de proceso de la computadora, y los dispositivos como el controlador de vídeo, el teclado y el dispositivo localizador pueden estar ligados al sistema mediante los buses internos de comunicación.

6.6 Otros dispositivos de E/S

Además de los periféricos anteriores podemos encontrar muchos otros, entre los cuales están los siguientes.

451. Módem y tarjeta de red. Estos dispositivos se encaminan a la conexión de un sistema informático con otras computadoras. En el caso del módem (modulador-demodulador), la conexión es de tipo serie y se emplea típicamente para la conexión vía telefónica. Existen al respecto un conjunto de normas que estandarizan los protocolos de comunicación entre modems. La velocidad de intercambio de información se mide en baudios (bits por segundo) y puede ser desde 300 baudios a 56.400 en función de la norma y la calidad del canal telefónico.

452. Las redes son medios que permiten la interconexión de varias computadoras, requiriendo canales de comunicación específicos para cada norma de conexión. Las topologías (bus, anillo, estrella, ...) son muy variadas, así como los protocolos de conexión, tanto a nivel físico, lógico, como de enlace.

453. Escáneres, lectores ópticos y de tinta magnético. Estos periféricos se utilizan para la introducción de información en la computadora, a partir de un soporte escrito.

- Los escáneres sirven para rastrear una imagen en papel, mediante un sistema de fotodetectores, que permiten reconstruir un bitmap aproximado que representa la imagen. Pueden ser de mano, de sobremesa, de trípode, de cilindro o mediante cámara digital.

Las características más importantes de un escáner son: velocidad de adquisición, resolución (en bits por pulgada), tamaño máximo de la imagen y número de planos por punto.

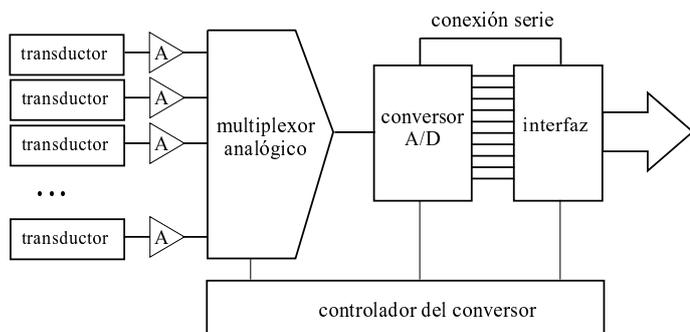
- Los lectores ópticos tratan de rastrear información, que típicamente viene codificada en un soporte, conforme a un formato específico, códigos alfanuméricos de tipo OCR⁸, códigos de barras lineales y en dos dimensiones, diseñados para ser robustos. La exploración puede realizarse mediante un haz láser en cuestión de décimas de segundo, y son apropiados para entornos industriales o poco amistosos.

⁸Optical Character Recognition.

- El lector de tinta magnética es un caso especial del tipo de lectores anteriores, con la diferencia de que la escritura se hace con un tipo de tinta que contiene un material magnetizable, de modo que se detecta con un cabezal de lectura magnética.

454. Digitalizadores Estos dispositivos se emplean para introducir coordenadas de planos y mapas. Mediante un artefacto con una pequeña lupa y un pequeño teclado permiten recorrer planos registrando coordenadas. Su construcción es similar al ratón, en cuanto a que se ubican en una superficie mediante un conjunto de ruedas con las que se ubica el dispositivo desde la computadora.

455. Conversores analógico-digital y digital analógico La categoría más general de dispositivo periférico lo conforman los conversores A/D y D/A. Con ellos se pueden construir sistemas que interactúen con sistemas físicos como plantas de producción, robots, y sensores de diverso tipo con el fin de automatizar la operación de procesos. Se pueden encontrar también en dispositivos como las tarjetas de sonido.



456. Los primeros permiten muestrear y cuantificar magnitudes analógicas que evolucionan en el tiempo, con una tasa de muestreo, y resolución determinadas. De este modo pueden recibir la señal que les proporcionan transductores de muy diverso tipo, como termopares, sensores de posición, micrófonos, etc. Habitualmente se construyen en grupos, aprovechando un convertor analógico/digital, para varios canales analógicos que se multiplexan en tiempo.

457. Los conversores digital/analógico permiten tomar la representación numérica de una señal, producida en una computadora, y generar la correspondiente señal analógica, adecuada para dispositivos como un amplificador de sonido, un regulador de una válvula, un motor paso a paso, etc. Se encuentran también en grupos, de modo que un mismo convertor alimenta de modo multiplexado a varios dispositivos externos, o lo que es más normal de modo individual.

Nivel de sistema operativo

Índice General

7.1	Conceptos de sistemas operativos	124
7.1.1	Evolución histórica.	124
7.1.2	Estructura del sistema operativo.	125
7.2	Procesos	126
7.2.1	Introducción	126
7.2.2	Comunicación entre procesos.	128
7.3	Administración de memoria	129
7.3.1	Administración básica de memoria.	130
7.3.2	Segmentación, paginación e intercambio.	130
7.3.3	Memoria virtual.	130

458. Al referirnos al sistema operativo de un sistema informático, no se habla de una parte de la máquina sino a un nivel de máquina. Desde un punto de vista habitual Sistema operativo es “Un conjunto de extensiones software de las primitivas hardware que culminan en una máquina virtual que sirve como entorno de programación de alto nivel y dirige el flujo de trabajo dentro de una red de computadoras”.

459. El nivel de sistema operativo ha de proporcionar primitivas de alto nivel para acceder al conjunto de *recursos* de una máquina entre los que se encuentran algunos del nivel de máquina convencional: Tiempo de procesamiento, canales de entrada/salida, memoria central. Otros recursos son nuevos y proceden del nuevo nivel de máquina: Sistema de archivos, procesos, canales de comunicación, ...

460. El interés del estudio del comportamiento de los sistemas operativos se centra en el conocimiento de este nivel de máquina para su uso posterior, pero también, desde el punto de vista de diseño, como ejemplo de soluciones algorítmicas a problemas concretos, y como ejemplo de sistema difícil de analizar.

7.1 Conceptos de sistemas operativos

461. La funcionalidad del S.O. es doble, como coordinador de recursos como la concurrencia entre procesos, los dispositivos de E/S, memoria, archivos, red; y como máquina virtual extendida, que permite un manejo del sistema a un nivel apropiado.

proceso

462. A la unidad de trabajo en un S.O. se le denomina *proceso*. Para resolver tareas se lanzan procesos que son despachados por el S.O. Entre ellos se encuentran procesos de usuario y procesos propios del sistema. Una definición clásica de proceso es “programa en ejecución”.

archivo
fichero

463. Al elemento de almacenamiento persistente de información se le denomina *archivo* (o *fichero*). Sobre este elemento se podrá realizar operaciones de lectura, escritura, borrado, según esté permitido.

stream

La noción de archivo ha dado lugar a otra abstracción denominada *stream* o chorro de datos, sobre el que se podrían realizar las mismas operaciones que con un archivo, pero que pudiera estar ligado a un canal de comunicación de datos de otro tipo.

llamadas al sistema

464. Las primitivas de la máquina extendida proporcionada por el S.O. se utilizan mediante acciones denominadas *llamadas al sistema*, que son realizadas por un proceso escrito en un lenguaje de programación que utiliza el interfaz de programación de aplicaciones del S.O.

7.1.1 Evolución histórica.

465. La evolución histórica de los sistemas operativos se puede estudiar desde dos puntos de vista: generacionalmente, en función de los avances tecnológicos; en función de la relación hombre-máquina.

Generaciones

1. No hay S.O., tan solo bibliotecas. Programación cableada y por tarjetas perforadas.
2. Aparecen los monitores batch. Se emplean transistores, cintas y discos magnéticos. Se separa la figura del programador y el usuario.
3. Desarrollo de sistemas polivalentes muy grandes. Ejemplos: OS/360, Multics, CTSS (introduce *time-sharing*). Como avances fundamentales: Circuitos integrados, escalabilidad en familias de computadoras, multiprogramación, particiones de memoria, spooling, tiempo compartido.

time-sharing

4. Sistemas operativos en red y distribuidos. Avances: VLSI, computadoras personales, utilización interactiva, interfaces de usuario mejoradas, redes.

Relación hombre-máquina

1. Hardware costoso, trabajo humano asequible: Se utiliza la máquina de la forma más eficiente.
2. Hardware asequible, trabajo humano costoso: Se busca optimizar el trabajo de los usuarios de las computadoras y de los desarrolladores.

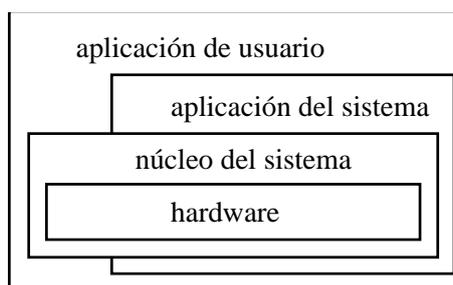
7.1.2 Estructura del sistema operativo.

466. El sistema operativo consta de un conjunto de software denominado *núcleo* (o *kernel*) que actúa proporcionando un conjunto ampliado de primitivas del hardware proporcionando nuevos recursos y protegiéndolos del uso directo por parte de cualquier programa de usuario. El núcleo del sistema se ejecuta en modo supervisor o privilegiado.

467. Adicionalmente el sistema provee un conjunto de procesos auxiliares que no se ejecutan en modo supervisor, pero proporcionan un interfaz de alto nivel a cierto tipo de recursos. Estos procesos pueden ser utilizados por otras aplicaciones del sistema y aplicaciones de usuario.

468. Finalmente, para realizar las tareas habituales, el usuario dispone de un conjunto de aplicaciones entre las que se encuentran las que le permiten la programación de nuevas aplicaciones, la administración del sistema, y cualquier otra herramienta de productividad: hoja de cálculo, procesador de textos, aplicación de contabilidad, etc...

Consideramos que los sistemas operativos, en lo referente a su construcción se clasifican en sistemas monolíticos y sistemas estructurados por capas.



Sistemas monolíticos

469. En los sistemas monolíticos todas las funcionalidades del sistema están al mismo nivel, el diseño no especifica una jerarquía. Entre las llamadas al sistema encontramos dos tipos de funciones:

1. *procedimientos de servicio*: funciones que puede usar un proceso de usuario.
2. *utilidades*: funciones empleadas por los procedimientos de servicio para resolver su tarea.

Sistemas estructurados por capas

470. En éstos, hay una jerarquía en las funcionalidades del sistema. Ésta parte del principio de encapsulamiento y ocultación de información. En este tipo de esquemas, cada nivel puede hacer uso de las funciones del nivel inferior, y no plasma una estructura de flujo de información.

Sus ventajas fundamentales son la modularidad y su construcción portable y mantenimiento sencillo.

nivel	descripción	comentario
14	shell	separa el usuario habitual del resto del S.O.
13	directorios	jerarquía de directorios del sistema de archivos.
12	procesos de usuario	máquinas virtuales por proceso.
11	E/S streams	redirección e intercambio fichero/dispositivo
10	dispositivos	entrada salida externa
9	sistema de archivos	
8	comunicaciones	entre procesos, pipes, ...
7	memoria virtual	
6	memoria local	secundaria.
5	procesos primitivos	procesos de sistema
4	interrupciones	servicios
...		

7.2 Procesos

471. Un *proceso* es un programa en ejecución, y en general, necesitará ciertos recursos (tiempo de CPU, memoria, archivos, ...) de los cuales algunos se asignan en tiempo de creación o en tiempo de ejecución.

hebra Por otra parte la unidad de expedición de un sistema operativo es el hilo o *hebra* (*thread*). Cada proceso consta de un hilo y un contexto de ejecución.

7.2.1 Introducción

concurrente **472.** La mayoría de los sistemas operativos permiten la ejecución aparentemente *concurrente* de más de un proceso en la misma máquina; estos sistemas se denominan multitarea, por contra de los sistemas monotarea. Los sistemas multitarea son útiles como sistemas de *tiempo compartido* con características de interactividad.

tiempo compartido Si además esta ejecución concurrente es real, dado que existe más de un procesador en la computadora, el sistema se denomina de *multiprocesamiento*.

multiprocesamiento **473.** Por otra parte, los sistemas operativos habituales no permiten ejecutar más de una hebra o hilo en cada contexto de ejecución. Si esto no es así, y se permite que más de una hebra comparta el mismo contexto de ejecución de un proceso, se denomina sistema operativo *multienhebrado* o multihilo (*multithreaded*).

multienhebrado **474.** El sistema operativo es el responsable de las siguientes actividades relacionadas con la administración de procesos: creación, eliminación, planificación de tiempo de ejecución, suministro de mecanismos para sincronización, comunicación, manejo de bloqueos, propiedades y seguridad.

475. Si el modelo de procesamiento es secuencial, se puede dar la impresión de estar ejecutando más de un proceso simultáneamente si se entrelaza la ejecución de las sentencias de varios programas. Ahora bien, el sistema operativo debe preservar el contexto de ejecución de cada sentencia y debe hacerlo empleando instrucciones que aseguren que desde el punto de vista subjetivo de cada proceso, el sistema se comporta como si hubiera varios procesadores.

476. Por lo que respecta al procesamiento, los sistemas operativos se clasifican groseramente en:

- Sistemas de procesamiento por lotes (*batch*): Los trabajos se sitúan por orden a la entrada del sistema. batch
 - monotarea: los trabajos se ejecutan de modo exclusivo y por orden.
 - multiprogramados: los trabajos se ejecutan en función de un algoritmo de planificación.
- Sistemas interactivos (o de tiempo compartido): Son sistemas multitarea donde se permiten procesos en los que un usuario interacciona con el sistema en línea (on-line).
- Sistemas en tiempo real: Sistemas multitarea donde se establecen restricciones más o menos rigurosas en los tiempos de respuesta de cada proceso, según los niveles de prioridad de cada tarea.

Creación y terminación de procesos

477. En cada sistema operativo existe un proceso primordial, que desencadena el resto de los procesos de la máquina, tanto de *usuario* como de *sistema*. De varias sucesivas etapas de creación de procesos surge toda una parentela de procesos, donde la relación más elemental es la de proceso padre–proceso hijo.

478. Este proceso de paternidad se realiza en unos sistemas como creación de un contexto nuevo con la carga del programa nuevo en memoria, partiendo de la nada; y en otros sistemas mediante un sistema de clonación, que puede ir acompañado de la recarga de un nuevo segmento de texto (*fork*). fork

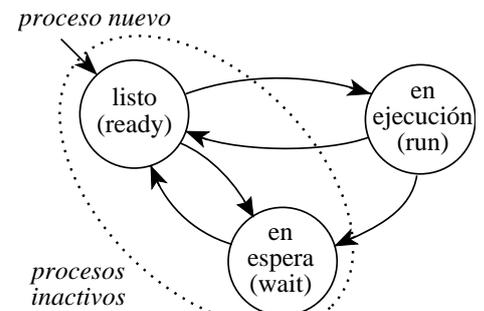
Estados de un proceso

479. En los sistemas operativos primitivos, cada proceso solo podía estar en dos estados, *ejecución* (ejecutándose en este momento) y *preparado* (o listo para ejecutarse en algún momento). Como resultado de la introducción de recursos compartidos se añaden más estados, siendo el esquema básico el de tres estados: listo–en ejecución–en espera.

Cada cambio de proceso actual requiere por parte del S.O. un cambio de contexto (*context switching*), que se encomienda a un proceso de sistema denominado despachador (*dispatcher*).

480. En los sistemas de tiempo compartido, el tiempo de CPU se reparte entre los procesos listos, de los cuales se extrae por turno uno al que se le dá la posibilidad de ejecutarse durante un tiempo o hasta que se cumpla cierta condición. El conjunto de procesos listos de un sistema se suele almacenar en una cola de espera.

Si por alguna razón (espera de alguna operación de E/S, petición de interrupción de usuario, ...) el proceso no está listo, entra a engrosar la bolsa de los procesos en espera, se la que se sale cuando se recibe alguna condición de terminación de operación de E/S, petición del sistema operativo, cualquier otro evento, etc.



Planificación de procesos

481. Con la posibilidad de alternar la ejecución entre procesos se puede optimizar el uso de los tiempos muertos, de E/S y demás, de los procesos, aumentando la productividad. De esta forma, los procesos interactivos y limitados por E/S, se mezclan bien con los procesos limitados por tiempo de cálculo.

planificación de procesos

482. La *planificación de procesos* es la toma de decisión de qué proceso se aloja en la CPU en cada momento. Ésta se puede tomar a largo y a corto plazo. La primera se puede establecer por un operador o por un algoritmo de planificación implementado en el sistema, mientras que la segunda se realiza por un proceso de sistema denominado planificador o *scheduler*.

scheduler

483. Los algoritmos de planificación de procesos a corto plazo son más efectivos cuando el sistema operativo puede interrumpir en cualquier momento al proceso actual. Cuando es así, el planificador se denomina apropiativo (*preemptive*), cuando no es así, el planificador solo puede activar el despachador cuando el proceso cambia voluntariamente de estado.

484. Entre los criterios que se usan para elegir el algoritmo de planificación se encuentra: el tiempo de utilización de la CPU, la productividad, el tiempo de retorno, el tiempo de respuesta y el tiempo de espera.

Entre otros, los algoritmos de planificación simple más utilizados son:

- Por orden de llegada (*first in first out*).
- Por primero el trabajo más breve (*shortest job first*).
- Por prioridades.
- Circular (*round robbin*). En éste se reparte el tiempo de ejecución en intervalos temporales equitativos. Requiere un despachador apropiativo.

7.2.2 Comunicación entre procesos.

485. La posibilidad de ejecutar concurrentemente varios procesos, puede ser un problema cuando dos de ellos acceden a algún recurso compartido, simultáneamen-

te. Por otro lado, existe una oportunidad de trabajo colaborativo entre procesos. Las funcionalidades básicas de procesos que se comunican involucran dos aspectos: comunicación de datos entre procesos, sincronización entre procesos.

486. Los mecanismos habituales empleados en comunicación y sincronización de procesos son los siguientes:

- Flujos de datos (*streams*). Mediante archivos compartidos, redirecciones de E/S (*pipes*),
- Memoria compartida.
- Colas de mensajes.
- Semáforos.
- Puertos de comunicaciones (*sockets*).

487. La aproximación a la resolución del problema de la comunicación se aborda desde diversas perspectivas, pudiéndose agrupar éstas como sigue;

- Resolución del problema de la sección crítica
 - Algoritmos de exclusión mútua: de Peterson, de Dekker, de Lamport (para sistemas distribuidos), etc.
 - Primitivas de bajo nivel: operación Test&Set, Intercambio...
 - Primitivas de alto nivel:
 - * Semáforos: binario, generalizado, con y sin espera activa.
 - * Monitores.
- Resolución del problema de la sincronización
 - Mediante bloqueo de recursos.
 - Sistemas de paso de mensajes: Ada *rendezvous*, etc.

7.3 Administración de memoria

488. En los sistemas operativos simples y estáticos, la memoria central se encuentra dividida en dos partes principales: Datos y programa. Tanto el S.O. como el proceso en ejecución conocen donde se sitúa esta memoria en el nivel físico, y acceden a ella directamente.

489. En los sistemas operativos modernos, la memoria es un recurso abstracto más del S.O., y éste gestiona cómo y para qué se usa, sin que se modifique la visión habitual que tiene de ella cada proceso. Presenta las siguientes características:

- Reubicabilidad: Los programas pueden re-ubicarse de un lugar de la memoria central a otro, sin alterarse su comportamiento.

- Memoria virtual: Se puede asignar más memoria de la que cuenta la computadora en su memoria central.
- Soporte de multiprogramación: Se puede gestionar más de un proceso en memoria.
- Intercambio: La memoria asignada a un proceso se puede cargar y descargar de la memoria central.
- Compartición: Los procesos pueden compartir datos y código.

7.3.1 Administración básica de memoria.

7.3.2 Segmentación, paginación e intercambio.

7.3.3 Memoria virtual.