Resumen libro Patricia Quiroga

Capítulo 1 – Evolución del procesamiento de datos.

Organización y arquitectura de la una computadora.

Una computadora es un dispositivo electrónico, diseñado para aceptar datos de entrada y realizar operaciones sobre ellos, para elaborar resultados que se puedan obtener como salidas.
Un programa es la representación de un algoritmo en un lenguaje de programación.

Componentes de una computadora:

- ✓ Memoria ppal.
- ✓ CPU.
- ✓ Buses.
- ✓ Dispositivos.
- ✓ Etc.

El **set de instrucciones** de una computadora permite representar los algoritmos que solucionan problemas.

Las **interrupciones** son eventos externos producidos por dispositivos de e/s.

El **concepto de arquitectura de computadoras** incluye aspectos relacionados con el formato del conjunto de instrucciones que el procesador puede ejecutar, representación interna de datos, estudios de módulos de hardware.

Estratificación del software.

La jerarquía más alta corresponde a los programas para los usuarios (aplicaciones).

La más baja corresponde a las señales que genera la unidad de control para gobernar distintos dispositivos.

Las **instrucciones** de un programa se ejecutan una tras otra, siguiendo un orden secuencial, excepto que haya aluna de salto condicional.

La maquina de **Von Neumann** se fundamente en 3 principios:

1. Trabaja con binarios.
2. Programa almacenado en memoria.
3. Posibilidad de provocar una ruptura de secuencia de instrucciones en un programa.
4.

Términos importantes del capítulo:

La **ejecución de una instrucción de ruptura** permite en determinado lugar del programa se salte a una instrucción que no es la siguiente.

El **código de maquina** es el lenguaje que interpreta la CPU y pertenece al novel de arquitectura del set de instrucciones.

La **arquitectura del set de instrucciones** determina el formato de las instrucciones, tipos de datos que puede operar, y las distintas formas de obtener los datos de memoria, a esto último se lo conoce como **modo de direccionamiento.**

Multiprogramación método que consiste en que varios programas residan en forma simultánea en memoria en estado de ejecución.

La **decodificación** convierte código binario a código de usuario.

Cada instrucción del programa es transferida desde la memoria a la CPU, y el módulo de control se encarga de organizar en el tiempo el conjunto de operaciones que permiten su ejecución. A cada una de estas operaciones se las denomina **microoperaciones**(son elementales).

Byte unidad minina de información.

Palabra de la CPU unidad de trabajo de CPU expresada en bits.

Denominamos **Señal de reloj o clock** a una señal que oscila a intervalos regulares entre 0 y 1.

El tiempo en que transcurre en completarse un ciclo de reloj es un **periodo de reloj.**

La frecuencia se expresa en Hz. La relación es T = 1/F, 1Ghz 1 000 000 000 por segundo. T = 1 nanosegundo. Recordar que **1 nanosegundo = 10^{-9} segundos.**

Capítulo 2 – Sistemas Numéricos.

- ✓ De **binario** a **decimal**: Barro de izquierda a derecha.

  $1\ 1\ 0\ 1\ 1 = 1.2^4 + 1.2^3 + 0.2^2 + 1.2^1 + 1.2^0 = 16+8+0+2+1 = 27$

- ✓ De **octal** a **decimal**: Barro de izquierda a derecha.

  $1\ 0\ 1\ 1 = 1.8^3 + 0.8^2 + 1.8^1 + 1.8^0 = 512 + 0 + 8 + 1 = 521$

- ✓ De **hexa** a **decimal**: Barro de izquierda a derecha.

  $0\ 1\ 1 = 0.16^2 + 1.16^1 + 1.16^0 = 0 + 16 + 1 = 17$

- ✓ De **binario** a **octal**: Barro de derecha a izquierda. Tomo de a 3.

  1 1 1 0 1 1 = 111 011 = 7 3

- ✓ De **binario** a **hexa**: Barro de derecha a izquierda. Tomo de a 4.

  1 1 1 0 1 1 = 0011 1011 = 3 B

- ✓ De **octal** a **binario**: Se toma cada digito y se lo convierte a binario

  73 = 111 011

- ✓ De **hexa** a **binario**: Se toma cada digito y se lo convierte a binario

  3B = 0011 1011

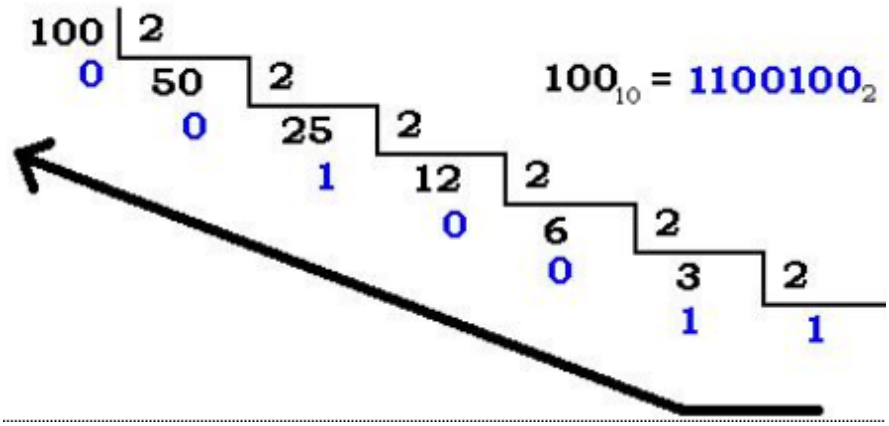
Con n bits se pueden representar los valores decimales comprendidos entre 0 y $2^n - 1$. Si hablamos de datos ordinales.

Conversión con parte fraccionaria.

- ✓ Se toma como caso ejemplo de binario a decimal (recordar que en octal y hexadecimal lo único que se cambia es la base).
  De la coma hacia la **derecha las potencias son negativas**.
  $0 , 1\ 0\ 1 = 0.2^{0} + 1.2^{-1} + 0.2^{-2} + 1.2^{-3} = 0.625$

- ✓ De **decimal** a binario.

  Análogamente para otras bases. Se **divide** sucesivamente.



  O también se puede hacer el método de los pesos….

  64 32 16 8 4 2 1 = 1 1 0 0 1 0 0
  $2^6\ 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0$

- ✓ De **fraccionario decimal** a **otras bases**.

  Multiplico a la parte fraccionaria por la base a la que lo quiero pasar hasta encontrar que se vuelve periódico, o cero o es cada vez mas pequeña. Me quedo con la parte entera de cada multiplicación. Pasaremos el 0,625 a binario.

  0,625 . 2 = **1**.250

  0,250 . 2 = **0**.5

  0,5 . 2 = **1**

  El número en binario es 0,101

El **bit de menos peso** es **LSB,** el **de mas peso** es **MSB.**

- ✓ Conversión de **binario fraccionado** a **octal/hexa.**

  Agrupo de a 3 bits en el caso de octal respecto a la coma. En hexa agrupo de a 4 bits respecto a la coma.

  1111, 1 = 001 111, 100 = 17,4 octal.

Operaciones.

Suma.

0 + 0 = 0

0 + 1 = 1

1 + 1 = 0 , **acarreo** 1.

Resta.

0 - 0 = 0

0 - 1 = 1 **produce acarreo.**

1 - 0 = 1

1 - 1 = 0

Ejemplos en distintas bases pagina 37 hasta 42.

Complemento.

Complemento restringido o Complemento a 1.

Se invierten todos los dígitos del número.

Complemento autentico o Complemento a 2.

Se barre de derecha a izquierda, hasta encontrar el primer uno (se conservan los dígitos que barrí hasta llegar al uno, luego se inviertan los siguientes a ese uno).

Capítulo 3 – Representación de datos en la computadora.

CPU = UC + ALU.

Formato de una entidad binaria: es la estructura y cantidad de bits de un determinado tipo de datos.

En ASCII los números se representan con octetos divididos en 2 partes: zona y digito.
A los números así representados se los denomina **zoneados o unpacked.**

Códigos de representación decimal (BCD).

Permiten la representación de números decimales de 0 a 9 en bloques de 4 bits.

Hay 3 clases de BCD tratadas en este capítulo, BCD puro, BCD exceso 3, BCD 2421.

- ✓ BCD Puro(8421):

  Se toma cada número y se lo convierte a binario en grupos de a 4 bits.

  256 = 0010 0101 0110

- ✓ BCD Desempaquetado y Empaquetado:

Eliminar la zona y agrupar dígitos en forma reducida se conoce como **empaquetado**, cada digito queda representado empleando 4 bits de digito.

Un número **desempaquetado** ocupa un byte.

Representación en BCD empaquetado y desempaquetado a continuación…

- **EL DECIMAL DESEMPAQUETADO :** Cada dígito decimal se representa en dos cuartetos, donde el primer cuarteto es todo lleno de 1 y el segundo es la cifra. El signo de este nº se escribe en el último cuarteto en el lugar de los 1. El sino + es 1100, - es 1101.

Porque es positivo

Ej.1992	1111/0001	1111/1001	1111/1001	1100/0010
	1	9	9	+ 2

Porque es negativo

Ej.-1992	1111/0001	1111/1001	1111/1001	1101/0010
	1	9	9	- 2

- **EL DECIMAL EMPAQUETADO:** Se elimina el cuarteto de la izquierda salvo en la última cifra, en este caso cada cuarteto lleva una cifra en BCD salvo el último que es el signo. Ej +1992

Para completar el byte

	0000/0001	1001/1001	0010/1100	
	0 1	9 9	2	+ (signo)

✓ <u>BCD exceso 3</u>:

A cada digito decimal le sumo 3.

18 = 1+3|8+3 =4 11 = 0100 1011

✓ <u>BCD AIKEN o 2421</u>:

En el **código Aiken** la distribución de pesos es: 2 - 4 - 2 – 1

Ósea que el número 3 se presentara: 0011
El número 6 : 1100

La **mantisa** representa todos los bits del número sin coma o punto decimal.

El **exponente** representa la posición de la coma o punto decimal en la mantisa.

Se almacena en la memoria solo la **mantisa** y el **exponente.**

Overflow se produce cuando el resultado se encuentra fuerda de los limites superiores del rango.

Cuando el resultado cae en el huevo definido por los limites inferiores del rango a este error se lo conoce como **underflow.**

- ✓ Punto flotante Simple precisión o Exceso 127:

32 bits en total. 1 de signo, 8 exponente, 23 mantisa.

-118,625 = **No tengo en cuenta el menos** 1110110,101…..normalizo 1,1110110101, ¿Cuántos lugares corrí la coma? 6, entonces….127(**fijo**) + 6 = 133 = 10000101

1|10000101|11011010100000000000000 = Signo, exponte, mantisa.

- ✓ Punto flotante doble precisión o Exceso 1023:

64 bits en total. 1 de signo, 11 exponente, 52 mantisa.

Bit de paridad es un bit redundante agregado a una cadena de bits en la que se pretende detectar un posible error.

Capítulo 4 – Aritmética de la computadora.

En este capítulo habla de cómo opera la ALU, tener que cuenta que solo realiza sumas. A los números negativos les realiza el complemento a dos.

El **acumulador** es un registro de cálculo referenciado por la mayor parte de las instrucciones aritméticas.

El **registro de estado** almacena banderas o flags

Registro de estado compuesto por: V,C,S,Z.

V indica overflow.
C indica acarreo.
S indica signo.
Z indica si el resuelto es distinto de cero.

El **overflow** solo se produce con operandos del mismo signo.

Para realizar operaciones en **punto flotante**, es importante el alineamiento del punto. Una rutina lleva a cabo la comparación del mismo, si resta los exponentes y el resultado es 0, están alineados.

En los convenios de exceso las características de ambos operandos son **binaria sin signo.**

Hay 2 tipos de desplazamiento, a **derecha** se pierden los bits menos significativos, el resultado es redondeado y a **izquierda** se pierden los bits más significativos, resultado erróneo.

Capítulo 5 – Algebra de Boole.

Cumple con las siguientes propiedades:

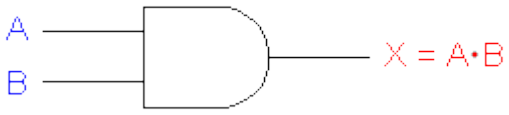
- ✓ Conmutativa.
- ✓ Distributiva.
- ✓ Leyes de identidad: a + 0 = a , a . 1 = a
- ✓ Complentacion: a + -a = 1 , a . –a = 0
- ✓ a + 1 = 1 , a . 0 = 0
- ✓ a + (a . b) = a
- ✓ a . (a + b) = a
- ✓ --a = a

Una **compuerta lógica** es el bloque elemental que permite la implementación del circuito digital o la representación de una red de conmutadores.

El **conmutador** es un dispositivo físico que permite controlar el flujo de un elemento.

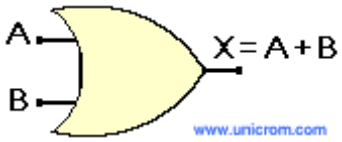
Compuerta AND.

Toma 1 cuando tolas las entradas son 1.



Compuerta OR.

Toma 0 cuando todas las entradas son 0.



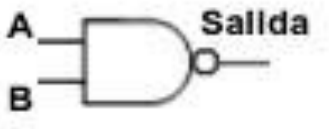
Compuerta XOR.

Toma 0 cuando todas las entradas son 0 o cuando las dos entradas son 1.

Compuerta NOT.

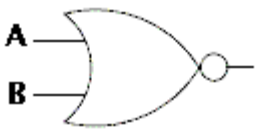
Compuerta NAND.

Toma 0 cuando todas las entradas son 1.



Compuerta NOR.

Toma 1 cuando todas las entradas son 0.



Compuerta YES.

No altera el valor lógico de la salida ni su tensión, tiene como finalidad aumentar la corriente de salida, osea es una amplificador de señal.

Transistor.

Es un conmutador electrónico que controla el flujo de un nivel de tensión, tiene la capacidad de conmutar rápidamente.

Las compuertas se sustituyen por una red de transistores y permiten calcular una función de una o más señales binarias.

El transistor reproduce la función complemento del Algebra de Boole.

En la compuerta NAND los transistores están montados en serie.

En la compuerta NOR los transistores están montados en paralelo.

Compuerta Triestado.

Es el tercer estado posible de una salida de una compuerta, se denomina **alta impedancia**, significa que la salida se desconecta o admite un estado flotante.

Este estado se puede controlar con una señal llamada **entrada de habilitación**. La lógica triestado es útil para transferir datos entre varios registros a través de un bus.

Buffer Triestado.

Tiene una línea adicional d (desconectado), opera como entrada de habilitación, con un 1 habilita el estado de alta impedancia.

Semisumador completo.

Suma 2 bits sin tener en cuenta el acarreo.

Sumador completo.

Suma 2 bits y tiene en cuenta el acarreo anterior.

Funciones normales o canónicas de una función.

Los Maxi términos agrupan bits bajos 0's y se multiplican mediante mediante funciones AND,se niegan aquellos bits que son 1’s.

Los mini términos agrupan bits altos1's y se suman mediante funciones OR, se niegan aquellos bits que son 0’s.

Ej:

A	B	So	Co
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

miniterminos o FND (So)= (-a . b) + (a . –b)

maxiterminos o FNC (So)= (a+b) . (-a + -b)

Capítulo 6 – Lógica Digital.

Hay circuitos combinacionales y secuenciales.

El **circuito combinacional** permite que en las salidas se obtengan valores binarios transformados por la operación de compuertas vinculadas en él y cuyo valor depende de los valores en las entradas.

El **circuito secuencial** los valores binarios de salida no solo dependen de los valores de entrada sino también de por los menos uno de sus salidas.

Un **circuito generador de paridad** permite la detección de error en la transmisión de información binaria entre un emisor y receptor.

Los **decodificadores** se encuentran en:

- Chips de memoria, posibilitan el acceso random.
- Bancos de memoria, posibilitan la decodificación que identifica el banco.
- Permiten la selección de dispositivos de E/S de microoperaciones.
- Facilitan la decodificación de instrucciones de la CPU para activar señales de control.

En una memoria **ROM**, las direcciones se indican por medio del bus de direcciones y los datos se leen desde el bus de datos.

Hay 2 tipos de circuitos secuenciales:

Asincronicos: Los cambios solo se producen cuando están presentes las entradas sin necesidad de una señal de reloj.

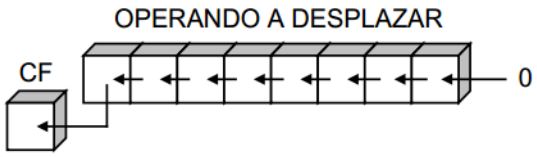
Sincronicos: Los cambios se producen cuando se establecen las entradas y además se genera y una transmisión de señal de reloj.

Biestables:

Celda binaria capaz de almacenar un bit, es un arreglo de compuertas.
Para que un circuito sea un elmento de memoria tienq ue poder cumplir con dos condiciones, **retener un bit y poder ser puesto a 0 o 1.**

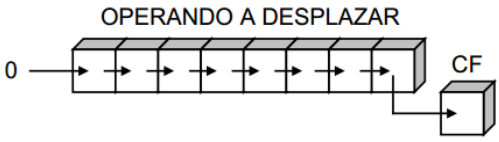
SHL(desplazamiento a izquierda)

Agrega ceros a la derecha, el bit más significativo lo guarda en CF.



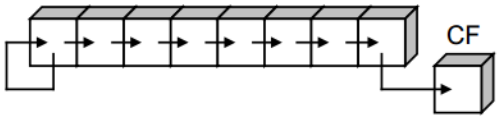
SHR(desplazamiento a derecha)

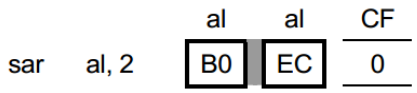
Agrega ceros a la izquierda, el bit menos significativo lo guarda en CF.

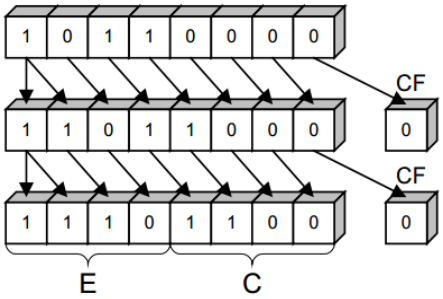


SAR(desplazamiento aritmético a derecha).
Análogamente para SAL(pero no conserva el bit)

Sirve para dividir un operando entre una potencia de 2. Conserva el bit mas significativo.







Capítulo 7 – Diseño de una computadora Digital.

Un modulo esta constituido por compuertas.

Una operación aplicada a un registro se llama **microoperacion**, se activa en un instante de tiempo sincronizado por los pulsos del reloj.

Quien entiende las instrucciones y genera microoperaciones para su ejecución es la **unidad de control**.

Instrucciones.

Cuando la PC realiza una tarea, se ejecutan una serie de pasos representados por el set de instrucciones.

El **código de una instrucción** es la combinación de bits que la unidad de control de la CPU interpreta para generar microoperacion que permitan su ejecución.

La forma de agrupar estos bits en entidades diferencias se define como **formato de la instrucción.**

Una misma **unidad de control** puede comprender distintos formatos de instrucción.

La cantidad de bits del **cop** determina la cantidad de acciones distintas que se pueden definir. Utiliza 2^n

COP=4bits,DATA=12bits

Data o dato hace referencia a un dato en memoria o sea determina la dirección de la posición de memoria donde se aloja el dato.

Cuando una instrucción está en la unidad de control se dice que están en **estado de ejecución**, su código binario indica donde está el dato y el modulo hardware que la lleva a cabo.

Se implementan en hardware las funciones que se utilizan con mayor frecuencia.

Ejecutar un programa por vez significa que el procesamiento es **monoprogamacion**.

La función de la CPU se puede separar en 2 partes:

1. Tratamiento de instrucciones: De esto se encarga la unidad de control, sincronizada por pulsos de reloj.

2. Operación de datos: Es llevada a cabo por la ALU.

La **UC,CU** toma las instrucciones del programa almacenado en memoria para interpretarlas y ejecutarlas. Esto se puede dividir en:

1. Búsqueda de la instrucción en memoria.
2. Interpretación del código de instrucción.
3. Búsqueda del dato afectado.
4. Generación de ordenes al módulo que opera sobre ese dato.

Fase Fetch: Búsqueda de la instrucción en memoria:

Cuando la UC ejecuta instrucciones de un programa debe alternar sus etapas fetch y execute desde la primera a la última.

La secuencia de ciclo se denomina **ciclo de instrucción.**

La CU retine la dirección de la instrucción en un registro llamado **"puntero de instrucción IP" o "contador del programa PC"**. La longitud de este depende de la cantidad de bits que se necesiten para dirección cualquier instrucción en la memoria asignada al programa,

Las instrucciones se almacenan en palabras sucesivas, por esta razón el IP debe incrementarse en una unidad para señalar siempre la próxima instrucción para buscar y luego de una ejecución.

Una vez que la CU envía a la memoria el contenido de IP, da la orden de lectura para que la palabra, se almacenen en el registro de palabra de memoria MDR(Memory Data Register).

En el **registro de instrucción IR**, se almacena la instrucción leída, la cual queda almacena hasta que termine su ejecución.

La zona DATA, se relaciona con la **MAR** cuando se debe tomar un dato de memoria y con el **IP** cuando se debe romper la secuencia normal del programa.

Para saber si se encuentra en etapa de **Fetch o Excuete**, existe un flag, llamado F, que si tiene el valor 1 es fase de búsqueda, de lo contrario (0) está en fase de ejecución. Este flag es vital ya que le permite a la CU alternar de una fase a la otra.

El que genera las microoperaciones de las fases fetch y execute es el control de instrucción. Estas microoperaciones no pueden enviarse simultáneamente.

La fase **Fetch** depende de la variable tiempo y la de control de estado, tiene que estar en 1.

Fase Execute:

Cada código de instrucción tiene asignada una secuencia de microoperaciones definida en el control.

La **instrucción HLT**, inhibe la generación de microoperaciones. Determina fin de programa, es imprescindible para que no se sigan buscando posiciones de memoria que no almacenan instrucciones.

La **instrucción INP**, permite la entrada de un dato desde un periférico.

Hay otras mas…pero solían aparecer en los finales de antes.

Unidad de control y sincronización del tiempo.

Cada microoperacion debe ser controlada por el tiempo (variable fundamental).

En la mayoria de las PC's, las señales de tiempo son generadas por un sistema de reloj. Se denomina **ciclo de reloj** al tiempo que transcurre entre dos pulsos adyacentes.

10^{-9} seg = 1 seg.	Frecuencia = cantidad de ciclos de reloj / seg.

¿Cuántos nanosegundos tarda un ciclo de reloj para una computadora que trabaja con una frecuencia de 25MHz?

25 MHz = 25000000 / seg

1 ciclo = 1 seg / 25000000 = 1 seg / 25 x 10^6 = 0.04 seg x 10^{-6} = 40 x 10^{-6} x 10^{-3} seg = 40 x x 10^{-9} = 40nanosegundos **1**

1 = Multiplique y dividi x 1000. ¿Por qué? Es lo que me falta para llegar 10^{-9}

Una computadora controlada por un sistema de reloj se denomina **sincrónica**.

El tiempo de una secuencia repetitiva constituye el **ciclo de la computadora o ciclo de maquina.**

El **ciclo de memoria** es igual al tiempo de acceso a la memoria si se trata de una memoria de lectura no destructiva, caso contrario es igual al tiempo de acceso + tiempo restauración.

El **tiempo de acceso a memoria** es el tiempo que tarda la CU en buscar la información en la memoria y dejarla disponible en el MDR.

El módulo de cálculo**(ALU)**.

Es el modulo de tratamiento de datos, los datos se tratan según ordenes de la CU, que interpreta la instrucción durante su estado de ejecución.

Cada bandera se actualiza después de una operación en la ALU.

El **overflow** se produce cuando el resultado excede la capacidad del acumulador.

El desplazamiento de los bits en la ALU se realiza por 3 motivos:

1. Multiplicar. (no existe en la alu)
2. Dividir. (no existe en la alu)
3. Testear un bit.

Capítulo 8 – Microprocesadores.

La **arquitectura abierta** es configurable según la necesidad del negocio, desde el punto de vista de hardware como de software.

Un microcontrolador es una PC con un programa de propósito especifico, esto recibe el nombre de **arquitectura cerrada**.

Longitud de palabra:

Un procesador **procesa** bits que obtiene de una memoria, los opera y almacena el resultado de nuevo en memoria. Una palabra es un grupo de 16bits.

Una **palabra de memoria** es la cantidad de bits a los que se puede acceder por vez.

Capacidad de direccionamiento:

Tiene relación con el acceso a líneas que transfieren direcciones. Con n bits del bus de direcciones se obtiene un mapa de direcciones de 2^n.

Numero de registros internos:

Es la cantidad de registros con la que cuenta el micro.

Al conjunto de registros que pueden actualizarse por las aplicaciones se denomina **registros para el programador de aplicaciones o registros visibles.**

A los registros que se pueden acceder por medio de los programas del sistema operativo, se los denomina **registros para el programador de sistemas o registros invisibles.**

Se hace presente una técnica llamada **pipeline**, consiste en dividir el procesamiento de cada instrucción en etapas y que estas operan en paralelo.

El X86 contaba con dos unidades funcionales, una se llamaba **unidad de ejecución** que se encargaba de decodificar y ejecutar instrucciones.

Registros de cálculo de 16 bits:

15 8	7 0	Tipo
AH	AL	Acumulador
BH	BL	Registro Base
CH	CL	Registro Contador
DH	DL	Registro para datos

Registros punteros:

Son utilizados para desplazarse dentro de un bloque o zona de memoria.

31 16	15 0	Tipo	Modo Direccionamiento
EIP	IP	Puntero de instrucciones.	-
ESP	SP	Puntero de pila.	A la pila
EBP	BP	Base de pila.	A la pila
ESI	SI	Indice Fuente	Indexado
EDI	DI	Indice Destino	Indexado

Registros de estado: (0-32)

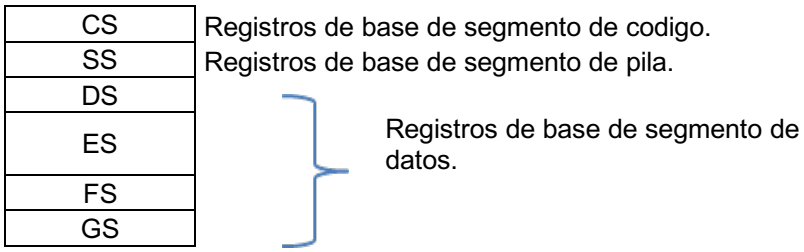
Se alojan todas las banderas aritméticas, banderas de modo de trabajo del microprocesador, banderas asociadas a interrupciones, etc.

Registros de segmento:

Concebidos para brindar soporte para aquellos SO que administran la memoria como una agrupación de segmentos. Un **segmento** es un bloque lógico de tamaño ajustado al objeto que contiene. Se denomina **objeto** al código de un programa, los datos que utiliza y una estructura de dato denominada **pila.**

Los registros de segmento almacenan la referencia binaria a la base de un segmento en memoria, donde empieza la zona de memoria para ese objeto.

Hay 6 de ellos:



El segmento es lógico, por ende aparte de la base necesita un desplazamiento para recorrerlo. En los **IA-16** los RS hacían referencia al comienzo de un área de memoria de hasta 64KB y el IP recorría el segmento de código.

Durante la fase de búsqueda o fetch se utilizaba el registro CS como base y el IP, generando una dirección segmentada, con este formato CS:IP.

Relación entre los registros y el modo de direccionamiento a datos:

La base de un segmento se encuentra en los registros de segmento, DS,ES o SS y se multiplica por 16.

El valor de una base para direccionamiento se almacena en BX o BP, el valor índice SI o DI.

Existe un registro puntero de 64bit denominado XIP y uno de banderas XPCR.

Los registros de gran longitud reducen la complejidad del hardware, disminuye la expansión del código fuente que utiliza tres registros base rotativo. Su uso minimiza la cantidad de accesos a la memoria cache.

Ciclo de instrucciones:

[xxxx] contenido de xxxxx = Modo direccionamiento directo. **Accede a MP.**

[BX] contenido de BX= Modo direccionamiento directo por registro. **No accede a MP.**

……. AX, 1 = Modo direccionamiento inmediato a registro. **No accede a MP.**

Secuencia de llenado de la cola:

BIU se encarga de controlar la transferencia entre el entorno y el microprocesador, también entrega las instrucciones a la unidad de decodificación a medida que se necesiten.

La **unidad de decodificación**, asociada a la BIU por un externo y a la EU, es la que interpreta el código de operación, reconoce el verbo de la instrucción, cuantos bytes mide en total para solicitarlos si es necesario y como se obtiene el dato según el modo de direccionamiento especificado en el código de operación.

En un modelo de arquitectura no segmentado, cada instrucción debería estar ejecutada por completo para pasar a la búsqueda siguiente.

La tendencia “ideal” es la de ejecutar una instrucción por ciclo de reloj.

La ejecución en paralelo de las instrucciones es casi optimo hasta que aparece una instrucción de salto.

La **EU** debe contar con registros especiales e invisibles al programador de aplicaciones, para resguardar el entorno de ejecución al momento en que se produzca el salto.

Capacidad de interrupción:

Las **interrupciones** y las **excepciones** son acontecimientos causados tanto por los dispositivos E/S como por el programa que se ejecuta en el microprocesador, su efecto produce una suspensión de la actividad actual del micro, para pasar a ejecutar un servicio que interprete el manejo de esa interrupción.

Los dispositivos externos utilizan interrupciones para informa su estado o solicitar la ejecución de actividades que le son necesarias. Los programas a su vez solicitan información de los dispositivos de E/S.

Cada interrupción está asociada a un número que la identifica, esto permite convocar el servicio que la atiende, puede ser provisto por el SO o por un servicio de BIOS.

Las **interrupciones programadas** se denominan **interrupciones internas/o de software**, causan la suspensión momentánea del programa que las convoca para bifurcar el servicio solicitado, este se ejecuta y retorna el programa interrumpido.

Cuando el micro recibe una señal de interrupción desde afuera, deja la ejecución del programa actual y bifurca al servicio residente en la memoria principal, están se clasifican como **internas o externas**.

	No enmascarablas (NMI)	
	La CPU es alertada por la señal de control NMI	Siempre son atendidas
Externas o hardware	**Enmascarables**	
Son convocadas de forma asincronica	La CPU es avisada por una señal distinta a la MMI	Hay una bandera que autoriza estas interrupciones. Flag F=1, la CPU suspende de manera nmomentanea la ejecucion de programa activo y ejecuta el servicio de int.
Internas o externas	-	La forma de convocarla es INT #, siendo 3 el numero de la interrupcion.
Son convocadas por el programa		
Excepciones	**Faltas o errores**	Son las que pueden detectar y corregir antes de que se produzca la ejecucion de una instrucción determinada.
Son provocadas como consecuencia que se producen y detectan durante la ejecucion del prog.	**Trampas**	Se detectan una vez ejecutada la instrucción que las provoca.
	Abortos	Se detectan sin localizar la instrucción que las procova, abortado la ejecucion del programa.

Toda información de la CPU se almacena en la **pila**. Este procedimiento permite **resguardar el entorno de** CPU para reanudar la ejecución a partir de que se produjo la interrupción.

La **restauración del contexto de** CPU es cuando se rescata la información que tenían los registros internos desde la pila.

Cuando se detecta una interrupción podemos indicar 3 pasos:

1. **Resguardo del contexto en la Pila.**
2. **Ejecución del servicio asociado a la interrupción.**
3. **Restauración del contexto**.

Los servicios se hallan en memoria principal, también para conocer donde se aloja se mantiene en memoria una tabla de vectores de interrupción de n entradas.

Hay tantas entradas como servicios definidos, cada vector contiene la posición del servicio.

Al conjunto de servicios se lo denomina **manejadores de interrupciones.**

Cada vector usa 4 bytes, para encontrar la dirección de memoria que corresponda, Interrupción Driver, deberá seguir los pasos siguientes:

- Multiplicar el número de interrupción por 4.
- Tomar los 4 bytes que se encuentran en esa localidad, que están invertidos, y asumirlos como dos entidades separadas de 2 bytes.
- Convertirlos a segmento: desplazamiento.
- Invertir los bytes de cada palabra, ya que en memoria el almacenamiento se encuentra en orden inverso.

Concepto de pila.

La pila es una estructura de dato en memoria de acceso LIFO.

El registro de segmento SS o segmento de pila, se accede con criterio LIFO. La que se encarga del acceso a la pila es la CPU, ejecutando instrucciones PUSH y POP.

La CPU utiliza la pila para:

- Almacenar la dirección de retorno IP.
- Almacenar el estado del procesador cuando se produce una interrupción. Los registros que apila son el CS y el IP y estado de Flags.
- Pasar parámetros entre procedimientos.

El acceso a la pila se realiza mediante los registros punteros SP y BP. El SP es el registro que contiene la dirección del próximo elemento de la pila vacío

La carga o extracción de datos de la pila es un procedimiento software.

Estas operaciones se llevan a cabo incrementando o decrementando el registro SP.

Call y **Ret** son instrucciones que sirven para invocar y dar retorno a un procedimiento o subrutina.

INT y **IRET** cumplen la misma función cunado se invoca una subrutina de interrupción.

PUSH pone palabra en la pila y decrementa el **SP.**

POP saca palabra de la pila e incrementa el **SP.**

El comando **e** escribe, el **a** edición y compilación, **t** permite visualizar la traza de ejecución, esto referido a *debug*.

La pila no se ve porque no es un registro, solo se ve el decremento del puntero de la pila SP.

El registro SP se va decrementado antes de que se ingresen los datos en la pila, luego de extraerlos se incrementa.

Tecnología.

CISC:

Se pueden ejecutar instrucciones simples o complejas. Una instrucción compleja utiliza varias microinstrucciones, por ende la unidad de control para un set de instrucciones CISC utiliza una ROM CISC admite múltiples modos de obtención del dato.

Cuando un set de instrucciones admite tanta variedad de modos de direccionamiento, aumenta el número de instrucciones del set.

RISC:

Todas las instrucciones tienen el mismo tamaño, facilita el pipeline.

Cada unidad de ejecución está "cableada", como así también lo están los microprocesadores RISC.

Hay 2 instrucciones para acceder a memoriam **Load** y **Store.**

Las instrucciones que realizan operaciones aritméticas son de referencia a registros.

EPIC:

Permite agrupar instrucciones para ejecutarlas de manera paralela en forma explícita. EPIC designa un tipo de arquitectura diferente al de las pc's RISC y CISC.

Su paralelismo es mediante código de maquina secuencial, implica un paralelismo solo a nivel de ejecución.

EPIC organiza la ejecución de instrucciones de bifurcación o salto condicionado.

Las instrucciones de las distintas ramas de un salto condicionado son marcadas por registros de atributo. **Predicación** es un método para manejar saltos condicionales, que en EPIC se denominan *"ramificaciones condicionales".*

Otra característica es el concepto de carga especulativa. La especulación trata de aprovechar el microprocesador cuando se encuentra en periodos de latencia, en ese momento “especula” sobre las instrucciones y los datos que va a necesitar más adelante y los carga. El compilador también está involucrado, que planifica cargas de datos a nivel compilación.

.

Capítulo 9 – Memorias.

Clasificación según el modo de acceso a la unidad de información:

1. De **acceso aleatorio** cuando un componente de selección habilita una palabra e inhabilita las demás.

   El **tiempo de acceso** es independiente del medio físico.

2. De **acceso secuencial** cuando para acceder a una unidad de información se establece una posición de referencia.

   El **tiempo de acceso** depende de la distancia entre la posición inicial y la unidad de información.

3. De **acceso asociativo** cuando la búsqueda de la unidad de información implica la comparación de un grupo de bits de la unidad de información con el contenido de una posición de memoria.

Clasificación según las operaciones que aceptan por cada acceso:

Una memoria es de **lectura/escritura** cuando admite ambas operaciones, se las denomina **vivas**, si es de solo lectura se las denomina **muertas.**

Clasificación según la duración de la información:

Las memorias son **volátiles** cuando pierden su información con el corte de suministro de corriente y **no volátiles – permanentes** en el caso contrario.

Dimensión de la memoria:

Se denomina capacidad de memoria a la cantidad de información que se puede almacenar en ella.

Memorias RAM estáticas y dinámicas.

Las **SRAM** son memorias vivas, volátiles y estáticas. Cada celda es un elemento biestable diseñado con compuertas.

Las **DRAM** son memorias vivas, volátiles y dinámicas. Estas degradan su información con el transcurso del tiempo

Cada celda almacena un 1 que se representa con la carga de un condensador, antes de que la información se pierda hay que restablecer la carga, se denomina **ciclo de refresco** (refresh cycle), esto debe estar cargo del **controlador de memoria**. Son memorias mas lentas que las SRAM, pero tienen mayor capacidad.

RAM con acceso directo.

Acceso a la información en forma random o al azar una memoria se organiza de manera matricial en filas y columnas. El número que identifica la palabra en un acceso random o al azar se denomina **dirección física**, y representa en realidad el número ordinal que le corresponde dentro de la matriz, comienzo 0 hasta p-1.

En el caso ejemplo de una RAM estática lo mas importante es:

La línea WE indica que con 1 en esta línea se da una orden de escritura, si es 0 es una orden de lectura.
La línea EN indica con un 1 que este chip se habilito para su acceso.

Biestable asociada a una matriz:

La memoria estática está constituida por biestables.

Cada uno de estos tiene 2 salidas, una para el valor normal del bit almacenado, que llamaremos Q y la otra que es el complemento no Q.

Una celda SRAM tiene 3 estados posibles:

1. **Reposo.**
2. **Lectura.**
3. **Escritura.**

RAM con acceso directo.

Las memorias asociativas son accesibles por contenido, el contenido buscado se denomina **rotulo, descriptor o segmento**.

Son caras y se justifica su uso en aplicaciones en las que sea imprescindibles búsqueda por contenido.

Jerarquía de memorias.

Se basa en 3 atributos:

- Velocidad de acceso.
- Costo.
- Capacidad de almacenamiento.

Las memorias de acceso rápido son de mayor costo y de menor capacidad de almacenamiento.

De acuerdo a la jerarquía las podemos clasificar en:

- Primer Nivel: Registros internos del procesador, denominados registros de propósito general.
- Segundo y Tercer Nivel: Soportes de almacenamiento temporal de instrucciones y datos intercambiables, a los que accede el microprocesador en forma directa.

  Estas se clasifican en 2 dos tipos:

  Memoria cache: Es una memoria de semi conductores, más rápida que la DRAM, de mayor complejidad por lo tanto de menor capacidad, su velocidad de respuesta se adapta a las exigencias del procesador.

  Memoria DRAM: Es una memoria de semi conductores lenta, menor complejidad, mayor capacidad de memoria.
- Ultimo Nivel: Memorias auxiliares.

*En una memoria de semi conductores se denomina **tiempo de acceso** al lapso que transcurre desde el momento en que el modulo de memoria recibe una solicitud de datos hasta el instante que que esos datos están disponibles para su transferencia al lugar de destino.*

Memoria Cache.

Es de tipo estático, su velocidad de respuesta se ajusta a los tiempos del procesador. La cache se usa como memoria intermedia entre el procesador y la DRAM, almacena en forma temporal la info a la que se accede con mayor frecuencia en esta última.

En la memoria de etiquetas o tags se almacenan las referencias de memoria principal asociadas a cada bloque.

El cerebro de una memoria cache es el **controlador de cache.**

Una MC está constituida por una memoria de etiquetas, una memoria de datos y un controlador.

El **controlador** se utiliza para gestionar su actividad.

Conexión en serie:

Es la más utilizada en los procesadores actuales. La **ventaja** se observa cuando hay un acierto, deja el bus del sistema libre para su uso, la **desventaja** es la penalización de tiempo, esto significa que si el dato no está en la cache, tendrá que ir a buscarlo a la MP.

Conexión en paralelo:

La solicitudes de las posiciones de memora que refiere el microprocesador llega en forma simultánea a la MP y a la MC. En caso de acierto, el controlador entrega la posición buscada al micro y genera una señal para que se aborte la búsqueda en MP. Una **ventaja** permite quitar o agregar el subsistema de cache sin necesidad de incluir modificaciones al sistema.

La **desventaja** es el alto tráfico al que se somete el bus de sistema.

Principios de funcionamiento:

La comunicación entre el procesador-RAM es de forma continua, ya que el micro busca en el MP la instrucción para ejecutarla o cuando busca un dato que requiera la ejecución de una instrucción.

La comunicación entre el procesador-RAM se estable por medio del bus de direcciones y el bus de datos.

En cualquiera de los tipos de conexión enunciadas, el controlador de cache debe capturar la dirección para verificar si se puede ofrecer al procesador su contenido. La forma en que se captura la dirección depende del tipo de organización.

Caching.

Procedimiento que gestionado por el controlador, anticipa las necesidades de posiciones de memoria principal de acuerdo con cierto cálculo de probabilidad de uso y utilizando criterios que consideran los principios de vecindad espacial y temporal.

El rendimiento depende tanto de la efectividad de la gestión de caching como de su tamaño.

Traducción de la dirección física.

El tamaño de la memoria DRAM no coincide con el de la cache, sus respectivos espacios de direccionamiento son distintos, una dirección física deberá ser traducida o "mapeada" por el controlador.

La gestión de traducción puede clasificarse según tres formas de organización del subsitemas:

- Totalmente asociativa.
- Asociativa de 1 via.
- Asociativa de n vías.

Mapeo totalmente asociativo:

A cada línea le corresponde un etiqueta.

Esquema: Etiqueta-Posición.

Es la mejor organización porque no existe una relación entre el identificador del bloque y su posición dentro de la cache.

Ante un fracaso, la palabra se mantiene en MP siguiendo 2 caminos: hacia el procesador y hacia la cache, para agregarse como bloque al que se accedió recientemente.

Una desventaja es la lógica de comparación, es compleja y cara.

Mapeo asociativo de una vía o correspondencia directa:

Es la menos utilizada en los subsistemas actuales.

El controlador interpreta la dirección física y en consecuencia ubica los bloques en cache.

La memoria ram se divide en grupos del mismo tamaño e igual estructura que la línea de la memoria de datos en cache.

Se reduce la complejidad y el costo, además de hacer el acceso mas rápido. Se puede prescindir de una memoria de carácter asociativa para la memoria de etiquetas.

Esquema: Etiqueta-Linea-Posición.

Mapeo asociativo de n vías o de n conjuntos.

Similar al mapeo directo, pero cada línea admite n etiquetas y n matrices de datos, implica una cache n veces mas grande y menor posibilidad de fracaso. Las n etiquetas y n matrices de datos de una misma línea constituyen un conjunto.

Actualización de la cache:

El momento de actualizar la cache es cuandos e detecta una falla o ausencia de la palabra buscada, esto es cuando la palabra no esta en memoria principal y el controlador debe "interceptarla" en su camino al micro mediante el bus de datos.

La asociación de una política de reemplazo afecta el rendimiento del subsistema de la misma forma que lo afecta su organización. El controlador ejecuta un algoritmo fijo, clasificado en 3 categorias:

1. LRU – la de uso menos reciente.
2. FIFO – Primera en entrar es la primera en salir.
3. RND – Aleatorio.

LRU: Sustituye la info que hace mas tiempo que fue referenciada o sea la de uso menos reciente. Esto se lleva a cabo mediante un grupo de bits que representa esta característica.

FIFO: Las posiciones pueden desplazarse en el sentido de una cola. Esto es fácil de aplicar en una cache de tipo asociativo.

RANDOM: Se realiza al azar sobre cualquier línea, la aplicación del algoritmo es rápida y sencilla, la desventaja de este tipo es que se puede sustituir una línea que se halla accedido recientemente.

Actualización de la memoria principal:

Una modificación en la cache significa la actualización de la memoria principal.

Acá nombran 2 criterios (pero falta el de escritura diferida mencionado en Angulo)

- Escritura inmediata (write through): Es simple, consiste en actualizar de manera simultanea ambas memorias, de modo tal que no se genera ninguna incongruencia entre la información almacenada en ambos niveles. El bus de datos esta continuamente ocupado.
- Escritura obligada (write back): Admite solo escrituras necesarias en memoria principal, la info en la cache puede ser actualizada varias veces antes de ser actualizada en MP. Este método genera incongruencia entre la info almacenada en ambos soportes. Esto puede ser grave, sino se actualiza la memoria principal cunado se permite que cualquier otro dispositivo tenga acceso a ella sin intervención del micro. Otra situación de este tipo es en el caso de un reemplazo.

Niveles de cache:

Cache de Nivel 1 es de tamaño reducido, se ubica funcionalmente, interceptando los datos que entran en el micro desde memoria o los que salen ya procesados a memoria.

Cache de Nivel 2 es de mayor tamaño y se ubica entra la cache de primer nivel y la MP, se utiliza como ampliación de la anterior(N1), este tipo de nivel contendrá la misma información que nivel 1 pero unos cuantos mas.

Memorias RAM dinámicas.

Son de lectura y escritura ,se utilizan para almacenar una mayor cantidad de bytes en la memoria principal de las pc's, esto es posible gracias a su bajo costo y gran capacidad de almacenamiento en relación a memorias estáticas.

La mayoría de los sistemas tiene uno o varios niveles de cache entre la CPU y la DRAM.

Hay un procedimiento denominado *"refresh"* que consiste en recargar los condensadores que tienen almacenado un 1 para evitar que la info se pierda a causa de las fugas en los condensadores.

Controlador de memoria dinámica:

En este tipo se “multiplexan” las direcciones.

Existen 2 señales las cuales habilitan la selección de una fila o columna, las cuales son: **RAS** y **CAS**, respectivamente.

La función del controlador de memo dinámica es esconder el entorno, este genera todas las señales de control y las del tiempo que necesitan para la activación de memoria.

Hay que evaluar un parámetro denominado CAS Latency, indica el número de ciclos de relej que transcurren desde que se realiza la demanda de datos hasta que estos se ponen en disposición de bus de datos.

Para acceder a la celda DRAM, el controlador tiene que activar la señal RAS. El tiempo que estuvo activa la señal CAS, es utilizado como parámetro para indicar el tiempo de acceso (CAS Latency).

El tiempo de *“precarga”* es cuando se producen dos lecturas consecutivas sobre el mismo chip el tiempo de acceso se penaliza con una cuota de tiempo adicional.

Módulos.

Para determinar el ancho de banda de las memorias se realiza el producto entre el ancho del bus expresado en bytes y la frecuencia efectuar de trabajo en MHz, denominada también velocidad física o real.

DDR SDRAM.

Envía los datos dos veces por cada ciclo de reloj, opera al doble de velocidad del bus de sistema o bus de memoria, sin necesidad de aumentar la frecuencia del reloj.

Detección y error por ECC.

El ECC 743 (Error checking and correction), es un código de bits que permite la detección y la corrección de errores, esta basado en un algoritmo complejo.

A diferencia de la paridad este puede detectar el error de un bit y corregirlo.

En el caso de la paridad o ECC, cuando se detecta un error se produce una interrupción **no mascarable.**

La memoria como un espacio lógico.

Para el micro la memoria es un hardware, con forma de matriz n x m, que contiene las instrucciones y los datos para procesar, cuyos megas o gigas son identificables por una dirección física.

El micro no hace diferencia entre las distintas tecnologías, el solo envía la dirección física del byte vía bus de direcciones.

La manera de administrar y gestionar una memoria depende del SO.

"*Mapear"* una dirección significa aplicar un algoritmo para establecer la correspondencia entre direcciones físicas y lógicas.

Almacenamiento de bytes en memoria. Big-Endian y Little-Endian.

El almacenamiento en memoria sigue un orden específico.

El byte **menos significativo** se almacena en la dirección numéricamente mas **baja** y el mas significativo en la mas alta, esta forma se denomina Big-Endian.

El byte **menos significativo** se almacena en la dirección mas **baja**, esta forma se denomina Little-Endian.

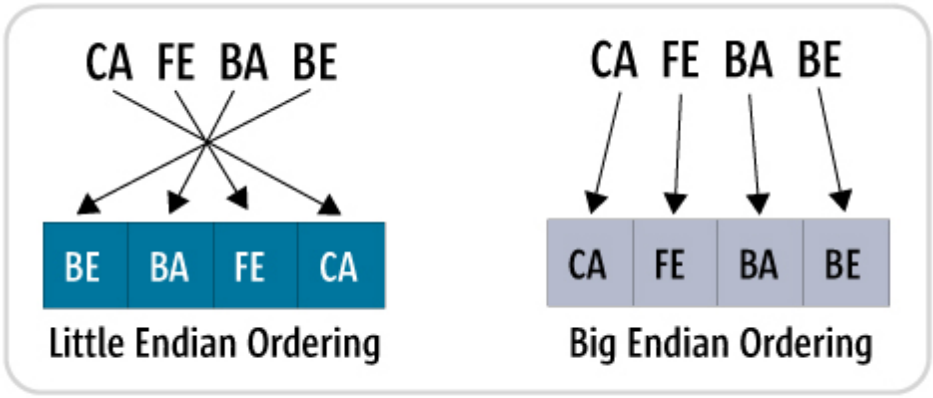


Figure 2 Byte ordering

Gestión de memoria y modo de operación de los procesadores.

En los procesadores 80x86 al modo de trabajo desprotegido se lo llamo real.

El modo de operación del procesador, se llama modo protegido, implica la protección de memoria.

Ambos modos de operación no pueden estar en simultáneo, el procesador es capaz de conmutar de un modo al otro.

Las plataformas de 32 bits actuales permiten varios modelos de gestión de memoria:

- Como un espacio de direcciones físicas, conocido como espacio lineal de direcciones o modelo plano.
- Como un espacio que alberga segmentos puros, que son bloques lógicos de longitud variable o modelo segmentado.
- Como un espacio que alberga paginas, bloques lógicos de tamaño fijo o modelo paginado.
- Como un espacio que alberga segmentos-paginados, modelo hibrido.

Calculo direcciones físicas en modo real.

El rango de direccionamiento queda limitado por $2^n - 1$.

Ej: 3 A 2 B : 1 3 0 1 (Base : Segmento)

1. Agrego un 0 a 3 A 2 B = 3 A 2 B 0
2. Luego le sumo a lo anterior 1 3 0 1
3. Obtengo como resultado 3 B 5 B 1.

Esto se lo conoce como **dirección absoluta o física.**

Modelo de memoria segmentada pura.

Para acceder a la memoria física se puede utilizar una técnica denominada segmentación.

Es de tamaño variable y el mismo esta administrado por el SO. Los segmentos contienen un solo tipo de objeto.

Denominaremos dirección lógica a la entidad binaria que identifica una locacion de memoria a nivel software, no se corresponde con la dirección física por tal motivo necesita ser mapeada.

Modelo de memoria virtual.

Es una forma de administrar la memoria para asignar más de ella a cada tarea, utilizando almacenamiento en disco, que suele ser el dispositivo de acceso directo mas rápido sea se simula que se dispone de una memoria mucho mas amplia que la que el sistema tiene en realidad.

Una dirección virtual debe traducirse a una dirección física, debe mapearse. En un modelo virtual el mapeo es una actividad que se intercala durante la ejecución del programa.

Cuando no hay suficiente RAM la MV (memoria virtual) mueve datos de la RAM aun espacio llamado *archivo de paginación.*

Modelo de memoria virtual protegida o paginación por demanda.

Mapeo Directo.

Para administrar un archivo temporal, se requiere una tabla de mapeo alojada en MP, con tantas entradas como paginas virtuales tenga el archivo. Consiste en dividir la MP en bloques de longitud fija, llamados páginas físicas y el espacio de direccionamiento virtual en bloques de igual longitud, llamados paginas virtuales.

Si el programa intenta acceder a una posición perteneciente a una pagina no presente en la MP, el procesador genera un error conocido como page fault, provoca una interrupción tipo excepción. Este error se le avisa al SO y su función será buscar una pagina nueva en el disco.

Antes de esto libera una página, y si se actualizo, primero copia en el disco antes de reemplazarla. Se llama *Swap out.*

Cuando se recupera una página almacenada en el disco, el procedimiento se llama *Swap in.*

Al producirse continuas liberaciones y recuperaciones de paginas, la gestión de MV puede congestionarse y entrar en un estado conocido como *“trashing”*.

Un termino mejor para denotar intercambio excesivo de paginas es *“hiperpaginacion”*.

De todas las actividades la gestión del SO que mas tiempo demora es la de una E/S.

Mapeo asociativo.

La cantidad de paginas físicas en memoria real será siempre menor que la cantidad de paginas virtuales posibles.

Una buena idea es crear una tabla que tenga igual cantidad de entradas como paginas físicas existan y puedan implementarse en una pequeña memoria asociativa.

Cada entrada de la tabla contiene un campo que indica un número de página virtual actualmente en MP y un campo que indica el numero de pagina física correspondiente.

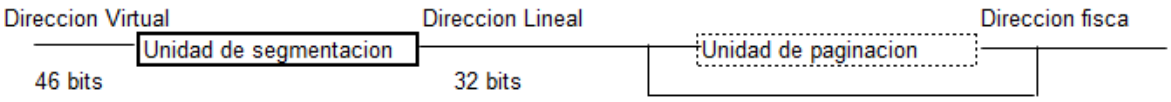
Memoria virtual segmentada o segmentación por demanda.

Se organiza el espacio de direcciones virtuales en bloques de tamaño variable, los segmentos.

Los segmentos pueden residir tanto en MP como en el disco.

La MMU esta constituida por **unidad segmentación** y **unidad de paginación.**

La unidad de segmentación se encarga de detectar si un segmento se encuentra en el disco y no en la memoria física, se lo comunica al SO para que posibilite el traslado.



Una tarea se asocia con su **tabla descriptora de segmentos**. Cada entrada de esa tabla contiene: un campo de 32bits, identificador de a base del segmento(esto tiene sentido si el segmento esta presente en MP), un campo de 20 bits que representa el tamaño marino que el segmento puede alcanzar, y un campo de 12 bits que permite determinar los atributos.

La **tabla descriptora de segmentos** se referencia con un registro base especial asociado al micro, denominado **registro de tabla local activa** (LDTR), este almacena la dirección de comienzo de la tabla asociada a la tarea que se ejecuta en la CPU.

La MMU consulta esta tabla por cada nuevo calculo de dirección física, ya se de una instrucción en fase de búsqueda o de un dato en fase de ejecución.

Una dirección virtual esta constituida por 2 campos: un campo selector y otro de desplazamiento.

En una x86 el selector es de 14 bits de orden superior del registro de segmento denominado CS y como desplazamiento el valor del EIP.

El selector se considera el índice que apunta a una de las entradas de la tabla de descriptores de segmentos.

En un IA-32: Los 14 bits de orden superior constituyen el índice en la tabla de descriptores. El TI indica si se esta accediendo a la tabla de descriptores locales o globales y RPL marca el nivel del privilegio del segmento. Los 32 bits restantes se toman del campo EDATA.

El área global cuenta con su propia **tabla de descriptora de segmentos globales** y con su propio registro base que apunta a la tabla GDTR.

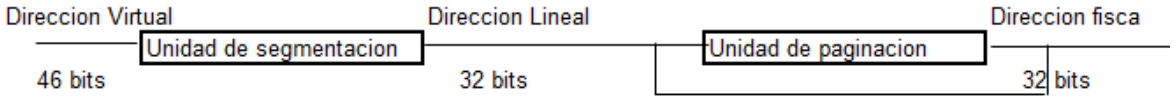
Descripción de algunos atributos.

G: bit de granularidad indica si el segmento esta dividido en paginas. Se debe considerar en este caso el campo limite como cantidad de paginas.

P: bit de presencia indica si el segmento esta en MP.

S: System indica si el segmento es de sistema.

Modelo de memoria virtual con segmentos paginados.



En este caso el segmento es una agrupación de paginas de longitud fija, la longitud del segmento varia en relación con las paginas asociadas a el.

El funcionamiento de la unidad de paginación es optativo.

Cada vez que la paginación detecta ausencia de pagina genera la excepción *"page fault"*, se genera un llamado para que el SO traiga la pagina del disco.

La unidad de paginación crea para cada tarea una tabla de 1K entradas de 32bits cada una, denominada **directorio de tablas de pagina**, reside en MP. El registro base que apunta al comienzo de esta tabla es un registro del microprocesador denominado **CR3.**

El valor de los 10bits de orden superior de la dirección lineal de 32bits constituye el desplazamiento dentro del directorio de tablas, y los 32bits del contenido de la entrada

accedida en el directorio de páginas constituyen la base de una segunda tabla de páginas, dentro de la tabla los 10 bits centrales de la dirección lineal constituyen el nuevo desplazamiento.

Algunos atributos:

D Bit dirty, si la pagina se escribió.

P Bit present, si la pagina esta presente.

A Bit access, si se accedió a la pagina, tanto para lectura o escritura.

TLB.

Es una memoria de tipo cache que almacena las ultimas direcciones físicas traducidas por la unidad de paginación, si la dirección esta en la cache se obtiene la dirección física en pocos nanosegundos, si no esta, el tiempo de traducción solo se penaliza en pocos nanosegundos.

La **densidad** de grabación en una unidad de disco rígido depende de cuan pequeña es la distancia del campo magnético (frame) que representa un bit. Mientras mas cerca del eje, mayor densidad de grabación.

Los **metadatos** son estructuras de datos que describen información.

Capítulo 10 – Instrucciones.

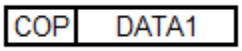
Instrucciones sin dirección. 

Representa únicamente al código de operación.

Algunos ejemplos: HLT, indica a la CPU que debe detener la ejecución de las instrucciones del programa en curso y discontinuar su actividad, no afecta a datos.

RET, indica a la CPU que es el final de un procedimiento convocado por el programa con una instrucción CALL.

LAHF, cuya funcion es cargar los flags.

Instrucciones de una sola dirección. 

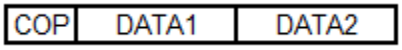
Todas las instrucciones hacen referencia implícita, en el código de operación, el data1 hace referencia al dato afectado.

Dato contenido en la instrucción:

Modo inmediato. MOV AX,0000h

Dato referido por la instrucción:

MOV AX,[0201]

Instrucciones de dos direcciones. 

Tres instrucciones podrían ejecutarse en forma simultánea, utilizando una unidad de ejecución diferente. Esta característica de ejecución de las instrucciones se denomina EPIC.

Direccionamiento directo de memoria.

La instrucción contiene la dirección real del operando.

Direccionamiento implícito.

El dato queda determinado por el mismo verbo, en el código de instrucción. Todas las instrucciones que asignen un valor por medio del COP pertenecen a esta categoría.

Direccionamiento inmediato.

Involucran el dato en la instrucción en si, pero ahora no en el campo código de operación sino en el campo DATA.

En la modalidad inmediata, el operando se incluye como parte de la instrucción. MOV AH,05

Direccionamiento indirecto.

En este modo el campo DATA contiene una dirección de una posición de memoria que contiene la referencia al dato. MOV REG,((0200)).

Direccionamiento de la CPU asociado a registros.

Los registros de la CPU se pueden utilizar para almacenar resultados parciales o direcciones que permiten establece nuevas modalidades de direccionamiento a memoria. Los registros generales pueden estar conectados entre si, por medio de buses, formando una pequeña memoria local, o a través de conexiones directas con compuertas de habilitación, en tal caso se consideran registros independientes.

Direccionamiento directo por registro.

La referencia a un registro siempre es mas rápida, puesto que no hay acceso a memoria.

Direccionamiento indexado.

Se involucran algoritmos que introducen índices.

Direccionamiento relativo a la base.

En esta modalidad se usan 2 registros el base (BX) y el BP.

Direccionamiento a una pila.

La CPU puede contar entre sus registros internos con un puntero a un pila de datos en memoria, Una pila es una estructura de acceso LIFO.

El que lleva el control de las direcciones en la pila es el SP. Su contenido direcciona la primera posición vacia de la pila y se actualiza por cada agregado y cada dato extraído.

La pila es el lugar donde se almacenan en forma temporal los registros de la CPU.

Es aconsejable utilizar el BP para acceder a los datos en la pila.

La base y el índice son datos que se pueden modificar durante el procesamiento ya que se encuentran en registros de propósito general de la CPU.

Capítulo 11 – Software del sistema.

Clasificación del software sistemas.

El software del sistema es el nexo entre las necesidades del usuario y las capacidades del hardware.

Está integrado por los siguientes componentes:

- Software de base.
- Software de comunicaciones.
- Software de administración de bases de datos.

El software de base controla y respalda en cierto modo el software de las otras categorías, todas ellas están relacionadas con el diseño del hardware.

El núcleo del software de base se denomina **sistema operativo**, sus componentes supervisan y controlan la actividad de recursos de físicos (hardware) y los recursos lógicos.

También forma parte la interfaz gráfica de usuario (GUI), el objetivo de esta es crear un entorno organizado para el usuario y los utilitarios o utilidades.

Un conjunto de sentencias constituye un programa fuente.

Un SO es una colección de programas que administran la operación de una o varias PC's. Es una plataforma software que asigna recursos y supervisa al resto de los programas que se ejecutan en la computadora.

El objetivo principal del SO, al ejecutar aplicativos, es crearles un entorno organizado, abastecer sus requerimientos y solucionar problemas que surjan durante la ejecución.

Dos grandes componentes de software del SO:

Residentes: Son llamados supervisores, residen de manera permanente en MP durante todo el procesamiento.

Transitorios: Residen solo cuando se los necesita y están almacenados en memorias secundarias cuando no están en la MP.

Niveles de administración del sistema operativo.

Administración del procesador y los procesos.

Un proceso es un programa, en estado de ejecución, o que está en alguna de sus etapas de su ciclo de vida. Está constituido por el código ejecutable, los datos con los que va a operar, el estado de los registros en la CPU y su propio estado respecto de otros procesos.

El despachador es el módulo de sistema operativo que asigna el proceso al procesador, pertenece al nivel de administración más cercano al proceso.

Hay que destacar 5 estados:

- Estado nuevo: Un programa ejecutable "adquiere vida" cuando se le crea un entorno adecuado para comenzar su ciclo de vida.
- Estado listo o preparado: Un proceso tiene recursos asignados suficientes para que la CPU comience o continúe su ejecución, pero no tiene la asignación del procesador.
- Estado de ejecución de un proceso: Estado en el que se está haciendo uso de la CPU.
- Estado en espera: Proceso que no puede continuar su ejecución porque le falta algún recurso que se le proveerá luego.
- Estado parado: El proceso termino su ejecución y se requiere un tiempo en el que el sistema operativo libere los recursos que le había asignado y destruya toda la información de contexto del proceso para eliminarlo del sistema.

Administración de memoria.

El SO asigna la memoria que un proceso requiere para su ejecución. También administra la MV en los SO que cuenten con esta facilidad.

Administración de dispositivos de E/S.

Un dispositivo de E/S requiere comandos o instrucciones específicas.

Encapsular los detalles propios de cada dispositivo es tarea de los drivers.

Cuando un proceso requiere una E/S produce una *"llamada al sistema"* que convocan a la función estándar.

Un proceso en curso suspende su ejecución para solicitar un servicio del SO por medio de estas instrucciones especiales, que tienen un nivel de privilegio mayor que las restantes y se conoces como primitivas.

Administración de archivos.

Supervisa la gestión de archivos para su creación, acceso y eliminación.

Tipos de sistemas operativos.

Multitarea y tiempo compartido.

Tiempo compartido trata de administrar los recursos repartiéndolos de manera equitativa.

Los multitarea son capaces de administrar procesos concurrentes y permiten que tanto las instrucciones como los datos de varios procesos residan al mismo tiempo en MMP.

Los task o tareas acticas compiten de manera simultánea por los recursos del sistema en forma alternada.

Multitarea y tiempo compartido.

Permiten el acceso de carios usuarios desde distintas terminales administradas por el mismo SO.

Tiempo real.

Tienen como objetivo proporcionar tiempos más rápidos de respuesta. La característica más importante de estos sistemas es que sus acciones se deben ejecutar en intervalos de tiempo determinados por la dinámica de los sistemas físicos que supervisan o controlan.

Traductores de lenguaje.

Son programas cuya función es convertir los programas escritos por el usuario en lenguaje simbólico a lenguaje de máquina. Un programa escrito en lenguaje simbólico se denomina **fuente**. Un programa en lenguaje de maquina se denomina **ejecutable.**

La traducción comprende el análisis del léxico y la sintaxis de cada instrucción o sentencia.

Si del análisis surgieran errores, el traductor generara un informe donde indicara el lugar donde se produjo y cuál es el tipo de error cometido.

Para realizar el análisis sintáctico, el programa traductor controla cada sentencia del programa fuente.

Se destacan 3 tipos de traductores de lenguaje: ensambladores, intérpretes y compiladores.

Ensambladores.

Se refiere a un tipo de software traductor que se encarga de traducir un archivo fuente escrito en un lenguaje Assembler a un archivo cuyas instrucciones estén en código de máquina.

Los traductores se dividen en 2 grupos en función de la relación entre el lenguaje fuente y lenguaje de máquina.

El primer grupo traduce una instrucción de un lenguaje fuente y genera una única instrucción de máquina.

El segundo grupo lo constituyen los lenguajes de alto nivel en los que una sentencia se traduce a varias instrucciones en código de máquina.

Se puede realizar una traducción inversa, denominada *desensamble*, esto es porque hay una correspondencia 1 a 1 entre instrucciones simbólicas e instrucciones de máquina.

El ensamblador permite generar tanto el código absoluto como el que permite reubicar los módulos, lee dos veces el programa fuente.

En la primera pasada guarda todos los nombres simbólicos en una tabla de nombres simbólicos y la completa con las referencias a memoria correspondientes para cada uno.

En la segunda pasada ya se conocen los valores de todos los nombres simbólicos, con lo que soluciona el problema de las referencias adelantadas.

La seudoinstruccion ORG permite iniciar el programa generando el código absoluto a partir de la referencia indicada en ella.

Cuando un módulo necesita acceder a datos o instrucciones contenidas en otro modulo, que seguramente no serán localizados debido a que no se completó la tabla de nombres simbólicos, para realizar estoy hay 2 seudoinstrucciones: export and import.

Export contiene todos los nombres simbólicos de un módulo que serán referenciados por otro.

Import debe indicar la referencia externa dentro del módulo que la quiere utilizar.

El programa enlazador (linker) es el encargado de relacionar los distintos módulos en un solo programa. Primero produce la reubicación de módulos, a partir de determinar la

dirección de comienzo e incrementarla con la TDA o con la TNSE. La función de enlazador es unir los módulos que ya fueron traducidos por el ensamblador para que construyan una unidad, lo producido en esta etapa se almacena en un archivo ejecutable en una memoria no volátil.

La diferencia entre **macro** y **subrutina** es que cuando se utiliza una macro el ensamblador repite la secuencia de instrucciones en el programa tantas veces como sea llamada la macro. Cuando se llama a una subrutina se provoca una ruptura de secuencia en el programa principal, se ejecuta la rutina y luego se retorna al programa principal a partir de la instrucción siguiente al llamado.

Intérpretes.

Es un traductor de lenguaje que traduce una instrucción en lenguaje de alto nivel a lenguaje de máquina, y de ser correcto, la ejecuta inmediatamente. Si encuentra un error de sintaxis, lo señala e interrumpe la ejecución.

La ventaja es que el programa se va probando medida que se confecciona, o sea permite una programación **interactiva.**

La desventaja debe traducirse cada vez que se ejecuta.

Compiladores.

Es un traductor de lenguaje que traduce un programa escrito en lenguaje de alto nivel a lenguaje de nivel, pero tiene algunas diferencias significativas respecto del intérprete. Separa la traducción de la ejecución del programa y agiliza tanto una como otra.

La ejecución del programa solo se realiza cuando la compilación termino de manera satisfactoria.

La relación entre las instrucciones de alto nivel y las de maquina son 1 a n, motivo por el cual las instrucciones en lenguaje de alto nivel suelen denominarse sentencias.

El proceso de compilación de un programa puede interpretarse en tres etapas:

Análisis léxico, sintáctico, generación de código.

Hay una cuarta etapa denominada optimización, cuyo objetivo es reducir el programa o hacerlo más veloz, utilizando técnicas como detección y eliminación de instrucciones reduantes y uso de registros asociados a la CPU en vez de palabras de memoria.

Capítulo 12 – Dispositivos de entrada y salida.

Solo se tienen en cuenta los HDD(es lo más relevante).

Discos rígidos.

Los discos magnéticos pueden ser rígidos, dispuestos en unidades de cabezas fijas o móviles, en grupo de uno o más platos.

Un disco está constituido por una base, recubierta de un material magnetizable con forma de circunferencia, que gira alrededor de un eje dispuesto en la unidad, es accedido por una o más cabezas lecto/grabadoras.

Una cabeza registra los bits en círculos concéntricos denominados pistas, es una división lógica y no física de la superficie, producto de la acción de rotación del soporte y de la posición fija de la cabeza al momento de grabación o lectura.

Para cada cara grabable, habrá una cabeza lecto/grabadora. Todas ellas acceden en forma simultánea como si fueran los dientes de un peine a un cilindro, cuya dirección esa dada por un número relativo a la distancia del radio entre las cabezas y el eje. La información se graba por cilindro, esto es verticalmente.

Controladora de disco.

El conjunto de platos está incluido en una caja de aluminio soldada que provee un entorno libre de contaminación para la operación de las cabezas lecto/grabadoras.

El vínculo entre la unidad de disco rígido y el bus del sistema se realiza a través del conector de la interfaz del bus.

La controladora del disco está constituida por:

- Controladora del motor del eje y brazo actuador.
- Controladora de interfaz, para comunicarse con la CPU del sistema de transferencia adecuado.
- Un micro que “ejecuta” los comandos propios del disco.

La cobertura magnética de una unidad de disco está compuesta por áreas “dominios”.

Las velocidades de transferencia del disco miden la velocidad de transferencia entre el *buffer* del disco y el host, que es una función puramente electrónica.

La velocidad de transferencia de datos determina la relación efectiva de transferencia de datos entre el buffer de disco y el soporte de disco.

Tiempo de acceso a disco.

Tiempo de búsqueda:

Periodo que tarda el brazo en mover las cabezas lectrograbadoras entre las pistas en milisegundos (1ms = 0.001 segundo).

Se computan considerando el tiempo de posicionamiento entre pistas adyacentes, el tiempo de posicionamiento entre la pista más interna y las más externa y el tiempo de búsqueda promedio que se determina que toma posicionar las cabezas lectograbadoras de la unidad de disco para un pedido de posición aleatoria.

Tiempo de cambio de cabezas o de switch:

El brazo mueve todas las cabezas lectograbadoras sobre los platos de una forma sincrónica, solo una de las cabezas puede estar leyendo o grabando datos a la vez.

Mide el periodo medio que le lleva a la unidad de disco cambiar entre dos de las cabezas cuando esta leyendo o escribiendo datos. Se mide en milisegundos.

Latencia Rotacional:

La cabeza lectograbadora se posiciona sobre la pista adecuada, debe esperar que la unidad de disco fire el plato al sec tor correcto, esto es latencia rotacional, en milisegundos y depende de la velocidad de giro de los discos.

El disco necesita en promedio, girar solo media vuelta antes de que el próximo sector para leer o escribir este debajo de la cabeza.

Tiempo de acceso a los datos:

Es una medida de lo que se tar4da en posicionar una cabeza lecto/grabadora sobre una pista particular y encontrar el o los sectores de interés dentro de esa pista para leer o escribir.

El tiempo de acceso es una combinación del tiempo de búsqueda, el tiempo de cambio de cabezas y la latencia rotacional, en milisegundos.

El buffer cache de una unidad de disco se usa tanto en las transferencias de datos del disco al host (lectura) como del host al disco (escritura), la tecnología de cache son: DisCache o WriteCache.

La lectura de estos datos adicionales no pedidos se llama pre búsqueda o cache look ahead.

Cuando la unidad de disco transfiere los datos, estos se envían a una velocidad de transferencia denominada “velocidad de ráfaga sostenida máxima”.

Cuando la unidad de disco recibe un pedido de lectura, recupera los sectores pedidos y pre busca tantos sectores secuenciales como pueda.

Durante las operaciones de escritura de datos, el cache de escritura permite que las transferencias del sistema al buffer y del buffer al disco se produzcan en paralelo, esto elimina las latencias rotacionales.

Velocidad de transferencia.

Luego de posicionar la cabeza, la unidad de disco está lista para leer o grabar datos desde y hacia el disco, esto conlleva a una transferencia de datos entre el disco y la memoria interna.

Depende de 2 medidas: la velocidad de transferencia del disco o la rapidez con la que pasan los datos desde el disco hacia el buffer o memoria controladora y la velocidad de transferencia del host.

Esta velocidad se mide en megabits/segundo o gigabits/segundo.

Capítulo 13 – Transferencia de información.

Buses.

Es un elemento de comunicación que relaciona cierto número de componentes o dispositivos. Se puede definir como un conjunto de conductores que transfieren señales electricas en forma pasiva, asociado con un hardware que regula su actividad, denominado **controlador del bus**.

Las señales se pueden clasificar en: de dirección, control y dato.

Cada dato transferido por un bus se conoce como **transferencia elemental** y se produce en un tiempo determinado, denominado **ciclo del bus**.

El grado de paralelismo del bus unido a la velocidad que admite para lograr la transferencia se denomina **caudal** del bus.

Si el ciclo de bus está controlado por el reloj del sistema, es una transferencia sincrónica. En cambio, cuando su operación es controlada por un dispositivo conectados a el es una transferencia asincrónica.

Buses de entrada y salida.

Los buses afectados a la entrada y la salida de información determinan una multiplicidad de estructuras que se denominan **arquitectura de buses**, esta permite definir normas de comportamiento para la transferencia de datos desde o hacia los dispositivos de entrada/salida.

En algunas pc's el bus que relaciona la cpu con la memoria seta separado del que se conecta estas unidades con los dispositivos de entrada/salida.

Señales de los buses:

- OWS: Prevenir al procesador que no inserte estados adicionales de espera.
- AEN: Indica si es la CPU o el controlador de acceso directo a memoria el que tiene control sobre las líneas de datos y direcciones en ese momento.
- ALE: Señala que la CPU coloco una dirección valida en el bus de direcciones.
- CLK: Señal de reloj que conecta directamente con un pin del procesador.
- OSC: Señal de oscilador.
- I/O CH CHK: Detección de errores (de paridad).

- I/O CH RDY: Avisar sal procesador o al DMA para indicar que un dispositivo lento necesita tiempo extra para estar preparado.
- IRQ: Se origina en un dispositivo externo para indicar al procesador que se requiere su atención inmediata. Se solicita al micro que suspenda lo que esta haciendo para atender la petición de interrupción.
- IOR: Orden de Lectura.
- IOW: Orden de escritura.
- Reset: Pin especial de procesador para reiniciarlo.
- SBHE: Si está activado, indica que se está haciendo una transferencia.

Dispositivos de entrada y salida.

Unidades periféricas en si como aquellas intermediarias, se encargan de efectivizar una transferencia entre la memoria interna y la memoria externa de los periféricos.

En toda transferencia se utilizan señales de control, dato y dirección, esta vez solo los buses de E/S. Las señales de control y tiempo se utilizan para regular la transferencia elemental, indicando como y cuando debe ocurrir.

Las funciones de un bus de E/S son:

- Comunicarse con el periférico y el sistema de CPU-memoria.
- Controlar la temporización durante la transferencia.
- Almacenar temporalmente bits para “paliar” la diferencia de velocidad entre emisor y receptor.
- Detectar si se produjeron errores durante la transferencia.

Cuando la actividad del bus de E/S es sincrónica se utilizan señales del clock que regulan la transferencia.

Un bus sincronice requiere que los dispositivos conectados a el estén sintonizados a esa frecuencia y que todas las actividades se produzcan en intervalos de tiempo fijos.

Cuando la actividad del bus de E/S es asincrónica, las señales del clk no regulan su operación. Su velocidad depende de los dispositivos conectados a ellos.

Señales comunes:

- Señal de clk.
- Señal que puede habilitar una espera.
- Señal de lectura/escritura.
- Señal de interrupción (IRQn).
- Señales de reconocimiento.
- Señal de bus cedido u ocupado.

Las señales de dirección permiten representan la dirección del emisor y el receptor.

Las señales de dato representan los bits del mensaje que se ha de transferir. El “ancho del bus” o la cantidad de líneas afectadas a la transferencia elemental dan una medida de potencial de trabajo.

Las unidades de comunicación con el bus son unidades hardware, y resuelven fundamentalmente el problema de disparidad en los tiempos de operación entre las unidades que conectan, se agrupan en:

- Interfaces paralelo.
- Interfaces serie.
- DMA’s.
- Canal o procesador de entrada-salida.

Cada intermediario puede estar asociado con más de un dispositivo.

Los dispositivos de E/S permiten la comunicación de la CPU/memoria con el medio externo.

La relación entre la CPU y los periféricos no puede ser directa sino que se necesitan nexos físicos.

Controladores.

Se utiliza en gran medida para definir cualquier unidad hardware que gobierne a otra.

Un **controlador de periférico** es un dispositivo asociado en forma directo al periférico, que puede estar físicamente integrado a el, o separado de este, y está constituido:

Un **buffer interno** que almacena la información, una **lógica de control** que interpreta comandos de periférico, genera señales para su ejecución y gobierna así la unidad.

Sus funciones son:

- Aislar el software de servicio de entrada/salida, que se ocupa de la transferencia de los detalles específicos del hardware del periférico y los convierte en invisibles.
- Compatibilizar la velocidad del periférico respecto de la del resto del sistema.

El controladora **DMA** permite una transferencia entre un dispositivo de entrada y salida y la memoria interna, sin intervención de la CPU, el controlador programable de interrupciones, atiende solicitudes de dispositivos que requieren atención de la CPU y arbitra sus demandas.

Cuando un controlador se involucra con una transferencia, la CPU recibe parámetros de un programa que se ejecuta, a través de una interfaz, este programa forma parte de la administración de entrada/salida y se denomina **manejador de dispositivo**.

El *driver* es quien establece la secuencia lógica de comandos y el controlador los reconoce y activa los mecanismos necesarios para su ejecución.

Puertos de entrada/salida.

Un puerto es un área de almacenamiento alojada en una interface, que permite la comunicación de un periférico con la memoria para enviar o recibir una secuencia de bits. El software de sistema los identifica con un nombre.

Interfaces.

Hardware que actúa de nexo entre un periférico o un adaptador y el bus. Sirve para adecuar señales y preparar la transferencia elemental basada en un protocolo. No tiene capacidad suficiente para tomar la responsabilidad de la transferencia completa. La transferencia completa está a cargo de la cpu o el canal.

Interfaz paralela.

Dispositivo hardware que permite el control de la transferencia en paralelo entre el bus de sistema y un periférico.

La interfaz cuenta con registros llamados ports, este está dividido en partes tales como: registro de datos, registros de control. La función del port es lograr la transferencia elemental. El registro de control tiene una función doble: 1. Recibe un comando que puede enviar al periférico, 2. Recibe señales de control.

Cuando la CPU inicia una transferencia, coloca un comando en este registro, se conoce como comando de *inicialización*, determina si la operación va a ser de entrada o salida y genera señales de entrada o salida que indican el sentido de la transferencia.

La CPU indica la dirección del puerto implicado, si la interfaz está constituida por 2 o más ports.

Interfaz serie.

Permite el control de la transferencia de bits en serie entre el bus y un dispositivo de E/S. Los registros de la interfaz constituyen el denominado puerto serie, la interfaz se inicializa colocando un byte en su registro de control.

La interfaz serie acepta la modalidades sincrónica y asincrónica, cuando los datos se transmiten en una línea desde un punto a otro, se produce un retardo de la señal en el medio de trasmisión, que provoca incertidumbre acerca de dónde termina un bit y donde empieza el otro, otra desventaja es que crea la dificultad de la delimitación de los caracteres, no se sabe dónde empieza y donde termina.

Canales o procesadores E/S.

Procesador dedicado o específico para controlar las transferencias de E/S sin intervención de la CPU en la ejecución del software de E/S. El IOP realiza sus actividades en paralelo con las actividades que involucre a la CPU, obtiene de la memoria y ejecuta las instrucciones de E/S y también puede realizar cálculos, saltos y otras tareas propias de la CPU pero orientadas a la gestión de E/S. La velocidad de transferencia depende del periférico que involucre.

Transferencia de entrada/salida.

Aspectos para resolver durante una transferencia son:

- Sincronizar los tiempos de transferencia entre la CPU-memoria y el periférico.
- Decodificar los bits que identifican al dispositivo.
- Convertir, un mensaje seria a paralelo o al revés (si es necesario).
- Convertir, el mensaje enviado de un formato a otro.
- Convertir, el mensaje enviado de un código a otro.
- Controlar, que el mensaje enviado se reciba de manera correcta.
- Decodificar un comando.
- Controlar las banderas de estado.

Se denominan maestros a los dispositivos que tienen el control del bus en un momento determinado, puede enviar señales de control, dirección y dato sobre el bus, conoce la dirección del emisor y el receptor.

Aquellos que no son maestros son esclavos, puede pedir un servicio de transferencia pero no lo inicializan.

La estrategia de control del bus depende de la arquitectura diseñada para el sistema y se denomina **arbitraje del bus**.

Driver.

Aquellos programas que conocen el dispositivo periférico, en cada uno de estos se hace referencia a los comandos propios para cada periférico.

Cada driver actúa como un receptor de requerimientos de otros programas, que pertenecen a otro nivel y desconocen las peculiaridades de cada uno de los distintos dispositivos externos.

El driver debe contener una serie de instrucciones que permitan la evaluación final de la o las operaciones realizadas y debe poder armar un informe de erros para el programa llamador de nivel superior.

El periférico debe entender comandos de verificación y control.

Modalidades de entrada/salida.

El SO cuenta con programas que gestionan las transferencias de entrada/salida pero esta vez en un nivel superior. Estos programas realizan funciones comunes a todos los dispositivos periféricos.

Un programa de aplicación puede hacer referencia a datos organizados en un archivo.

Transferencia controlada por programa.

La que ejecuta el programa de E/S es la CPU, esta controla el acceso a memoria para ubicar el dato, a través de sus propios registros internos y los de interfaz. La CPU debe verificar el estado de la interfaz a través del **puerto de control** en forma continua.

Si la dirección asignada al puerto pertenece a un área separada de la memoria, se indica que se utiliza la técnica de *direccionamiento aislada.*

Si la dirección asignada al puerto está en el área de espacio de direccionamiento de memoria se indica que se utiliza la técnica de *mapeo de memoria.*

Esta modalidad también se conoce como *sondeo,* significa que la CPU se ocupa de verificar el estado del periférico y no el periférico el que interrumpe a la CPU.

Transferencia iniciada por interrupción.

La CPU es quien solicita la transferencia, no verifique de manera continua el estado de un dato listo para transferirse desde el dispositivo externo, sino que la interfaz asociada genere un aviso que indique que está preparada para transferir.

La interfaz provoca una interrupción, esto se indica con un bit de estado **bit de interrupción (IRQ)**.Para cada ciclo de ejecución de una instrucción la CPU consulta por interrupciones externas, si es así guarda la info suficiente del estado del proceso que va a interrumpir, para luego continuarlo.

Cuando un SO admite multiprogramación y se requiere una transferencia de datos desde un dispositivo externo para continuar con el procesamiento del programa en ejecución, se genera una llamada al sistema, y se coloca el programa en ejecución en estado “en espera”, así la CPU sigue con otro proceso.

Transferencia con acceso directo a memoria.

Las interfaces pueden conectarse a la memoria a través del controlador de acceso directo a memoria. Los DMA están asociados a dispositivos rápidos y que transfieren información en bloques, grupos de bytes.

La CPU solo interviene indicándole al DMA como parámetros la cantidad de palabras a transferir, la pos inicial de la palabra en memoria principal y la dirección del dispositivo y queda liberada para realizar otra actividad.

El dialogo entre el DMA y la CPU se hace mediante ciertas líneas del bus común.

La CPU envía a través del bus de direcciones un address que reconocerá como propia, y así activara su operación.

El DMA realiza lo que se denomina *robo del ciclo* que significa que suspenderá momentáneamente el funcionamiento de la CPU hasta finalizar la transferencia completa.

Cuando el DMA recibe la señal *bus disponible,* envía a la memoria principal la dirección de la palabra a transferir a través del bus de direcciones y envía una señal al dispositivo que implique transferencia.

Por cada transferencia elemental el DMA decrementa su registro de *cantidad de palabras a transferir* e incrementa su registro de *dirección de palara a transferir,* si se opera en modo *robo de ciclo* inhabilita la señal *bus disponible* que posibilita el coloquio DMA-CPU.

Cuando el registro cantidad de palabras a transferir llega a cero el DMA inhabilita la señal “solicita” y genera una señal de interrupción para la CPU.

Señales del bus para la utilización del DMA.

DRQ, se utiliza para solicitar acceso directo a memoria.(similar al IRQ)

DACK, acusar recibo de la petición DRQ correspondiente.

MEMR, indica a la memoria conectada al bus que escriba los datos en el bus de datos.

MEMW, indica a la memoria que almacene los datos situados en el bus de datos.

T/C señala que el controlador DMA alcanzo el final de una transferencia.

Capítulo 14 – Procesadores avanzados.

El paralelismo tiene como objetivo aumentar la velocidad de computo incrementando la cantidad de instrucciones que se ejecutan durante un intervalo.

Cuando se habla de procesamiento en paralelo la idea central es multiplicar la cantidad de unidades de ejecución que operan concurrentemente, ya sea ejecutando distintas instrucciones en paralelo o ejecutando distintas etapas de instrucciones simultáneamente.

Procesar varias etapas de instrucciones en forma simultaneo recibe el nombre de **pipeline o segmentación de instrucciones**.

Paralelismo a nivel instrucción.

Uno de los métodos utilizados para incrementar el rendimiento es iniciar la búsqueda de la instrucción en memoria y su decodificación antes de que la ejecución de la instrucción anterior haya terminado, esto es paralelismo a nivel instrucción, consiste en la simultaneidad de ejecución por etapas. La cantidad de etapas definidas se denomina grado de paralelismo.

Unidades:

- Gestión de memoria, traduce direcciones lógicas en absolutas.
- Interfaz del bus, accede a memoria y a dispositivos de E/S.
- Precarga de instrucciones, recibe 2 bytes del bus y los precarga en la cola de 16 bytes.
- Decodificación de instrucciones, decodifica el código de operación y lo traduce a micro código.
- Ejecución, ejecuta el micro código.

Existe la posibilidad cuando se está ejecutando pipeline, que una instrucción dependa del resultado de la anterior, este conflicto recibe el nombre de **dependencia de datos**.

Esto se puede resolver retrasando la ejecución de una de las etapas o resolverlo a nivel compilación, esto consiste en una técnica que produce un reordenamiento de las instrucciones sin afectar la lógica del programa, se la llama a esta técnica **ejecución fuera de orden**.

Durante la ejecución, las múltiples instrucciones se leen y pasan a un panificador, que determina cuales de ellas se pueden ejecutar en paralelo, la mayor complejidad de diseño de CPU se relación con este. Mientras más paralelismo haya mayor es la posibilidad de dependencias o saltos inesperados.

La predicción de bifurcación procura predecir qué camino o bifurcación tomara una instrucción de salto condicional, Intel la llama **rama de ejecución.**

La ejecución especulativa de cada rama recibe el nombre de flujos multiples.

Paralelismo a nivel instrucción-granulado fino.

Paralelismo a nivel proceso-granulado grueso.

El paralelismo a nivel hilo de ejecución(TLP)tiene como objetivo incrementar el número de programas individuales que una CPU pueda ejecutar en forma simultánea, los hilos son rutinas concurrentes que comparten variables globales y el mismo espacio de direccionamiento, su mejora en el rendimiento se apoya en la habilidad de solapar cálculos con operación de entrada/salida.

Para lograr el TLP se usa el multiprocesamiento a nivel instrucción a nivel de chip y el multihilado simultaneo.

Arquitectura superescalar: Implementa paralelismo a nivel instrucción y a nivel flujo.

Paralelismo a nivel arquitectura.

Para lograr un rendimiento alto, los micro no solo deben ejecutar instrucciones de manera rápida, también es preciso que se ejecutan más instrucciones por ciclo de reloj. La ejecución en paralelo requiere técnicas de predicción de salto o especulación de datos, esto es para aliviar el retrieve(tiempo que se tarda en recibir info solicitada a memoria) de memoria.

Cuando todo esto no alcanza el rendimiento deseado, se lleva a cabo el paralelismo a nivel procesador.

Taxonomia d eFlynn.

SISD: La CPU recibe una sola secuencia de instrucciones que operan en una única secuencia a los datos, no hay paralelismo.

MISD: Ejecutar distintos procesos sobre un único flujo de datos, el paralelismo esta dado porque cada procesador resuelve una parte del problema en forma simultánea.

SIMD: Procesamiento paralelo sincronizado, ejecuta la misma secuencia de instrucciones sobre diferentes flujos de datos.

MIMD: procesamiento paralelo asincrónico, para cada procesador se asigna una secuencia de instrucciones y datos.

Los procesadores vectoriales son aquellos en los que una sola instrucción opera en forma simultánea sobre un conjunto grande de datos.

Datawarehouse es un repositorio completo de datos de empresa, donde se almacenan datos estratégicos, tácticos y operativos con el objeto de obtener info que ayude al proceso de toma de decisiones gerenciales.

Datamining es un repositorio de datos que se almacenan con el objetivo de extraer conocimiento procesable implícito en ellos utilizando inteligencia artificial, análisis estadístico y patrones de comportamiento.

Descripción de la arquitectura Itanium.

Fue diseñada para obtener un mayor paralelismo a nivel instrucción pero con técnicas de predicción, especulación y predicación, su arquitectura se constituye en lo que se denomina una arquitectura de la bifurcación. Estas características se apoyan de aplicaciones de alto nivel como datawarehousing y datamining.

Esta provista de mecanismos que le permiten al código compilado gestionar la forma de uso del procesador a nivel hardware corroborando información del entorno en tiempo de ejecución que permite salvar fallas.

Modos de operación.

Cuando se ejecutan instrucciones IA-32 se conmuta a Itanium con la instrucción jmpe, cuando se ejecutan instrucciones Itanium se conmuta a IA-32 con la instrucción br.ia.m. Si se produce una interrupción, siempre se operan en entorno Itanium y se retorna del servicio con rfi.

Itanium arquitectura EPIC.

Epic esta tecnología permite la ejecución en paralelo de la mayoría de las instrucciones Isa-64bits que definen la arquitectura, además de contar con gran cantidad de registros internos.
El paralelismo explicito significa que las dependencias de instrucciones se evalúan a nivel compilación, mientras que a nivel ejecución las instrucciones de distintas ramas de un salto condicional son marcadas por registros para ejecutarse en forma simultánea.

Paralelismo explícito.

Está a cargo del compilador que organiza de manera eficiente el código para su procesamiento y hace explicito el pedido para que de este modo el procesador pueda enfocarse en la ejecución simultanea de instrucciones de la forma mas efectiva.

Registros internos: 128 enteros y 64 coma flotante, 64 registros de predicado de un bit y 8 registros de salto.

Tabla de registros de propósito general.

Son 128 de 64bits cada uno, y se denominan GR0-GR127, cuando se ejecutan aplicaciones sobre datos enteros se utilizan los registros GR0-GR31, están incluidos los registros de segmento. Los primeros 32 son estáticos, los restantes se llaman *stacked general registers*, las aplicaciones los pueden utilizar para almacenar el marco de pila de registros.

Registros de coma flotante.

Son 128 se identifican FR0 a FR127.

Los registros predicado.

Utilizados para el tratamiento de saltos, son 64 registros de 1 bit y se idéntican como PR0-PR063. Los predicados se utilizan en instrucciones de salto condicionado, hay 2 caminos de ejecución.

Una vez que se ejecuta la condición, el procesador guarda un 1 en un registro de predicado que corresponde a destino verdadero y 0 en los otros.

Los registros rama.

Son 8 de 64b cada uno y se identifican BR7-BR0.

Especulación.

Es una técnica que aplicada a instrucciones consiste en ejecutar una instrucción antes de tener la seguridad de que su resultado se vaya a utilizar.

Control Speculation es el término aplicado y una de las formas de llevar a cabo esto es el uso de predicados.

Especular es a nivel datos, la técnica consiste en leer anticipadamente un dato, suponiendo que una escritura pueda modificarla. Data Speculation es el término.

Los compiladores capaces de utilizar la técnica de especulación deben transformar una lógica que ha sido programada en forma *secuencial* en una cierta cantidad de hilos de ejecución en paralelo.

Un compilador para EPIC genera secuencias de instrucciones independientes, llamadas hilos o threads para que se ejecuten en paralelo.

El final del hilo se marca con una instrucción de parada implícita.

Predicación.

Ejecución de secuencia de instrucciones que *dependen* de una condición se cumpla.

Todas las instrucciones se ejecutan aun cuando dependen del valor de su predicado. Si se pueden ejecutar ambas ramas en forma simultánea, el salto puede evitarse utilizando instrucciones denominadas *código de predicado.*

Predicación de saltos.

Hay dos clases de predicciones:

- **Estáticas**: La predicción se produce a nivel compilación, el compilador estima una dirección para cada una de las instrucciones de salto, en los saltos incondicionales siempre se conoce la dirección, no es estimativa.

- **Dinámica**: Ocurre durante la ejecución, se construye una tabla de saltos y se guarda en un registro su comportamiento con el fin de decidir cuál es la dirección más probable de la próxima instrucción.

Formato de una instrucción.

Por efecto de la compilación se juntan tres instrucciones en un solo grupo que se llama *bundle*, por cada uno se asocia una plantilla, que indica todas las posibilidades de ejecución de las tres instrucciones, esto es:

Combinación de serie y paralelo. (Todas en serie o paralelo, etc).

Descripción de arquitectura AMD64.

El aumento de número de registros permite que los compiladores tengan una mayor flexibilidad para alojar más variables de memoria en los registros internos, beneficia las aplicaciones por la menor cantidad de acceso a memoria.

Modos de operación.

Modo largo: Es la expansión a 64bits del modo protegido legado de las x86, esta constituido por dos sub modos, el modo 64-bit y el modo compatible.

Modo de 64 bits: Soporta el direccionamiento virtual con direcciones de 64bits y las características de los registros extendidos. Las características de direccionamiento incluyen la utilización del puntero de instrucciones de 64bits y el direccionamiento relativo a este puntero. Utiliza el modelo de memoria denominada plano o flat.

Plano: Permite inicializar en forma parcial el modo protegido y permite utilizar todo el espacio de direccionamiento de la memoria existente en una computadora.

Las instrucciones de transferencia de control pueden utilizar el “modo relativo a programa” que permite relacionar el campo DATA de la instrucción con el valor actual del puntero de instrucción.

Modo compatible: Permite que los sistemas operativos de 64bits puedan ejecutar las aplicaciones existentes en modo 16 y 32, generadas por plataformas x86, sin recompilar y con la limitación de acceder solo al espacio de direccionamiento de 4gb, la habilitación del modo está a cargo del SO.

Desde el punto de vista de gestión del SO, se utilizan las técnicas del modo de 64bits para el manejo de interrupciones, traducción de direcciones y las estructuras de datos.

Modo herencia: No solo admite la compatibilidad de las aplicaciones de 16 y 32 bits, sino que también soporta sus SO, implica soportar los tres submodos antecesores: modo real, modo virtual-8086 y modo protegido.

- Modo real: una aplicación utiliza segmentación de memoria, sin paginación, las aplicaciones de 16bits utilizan un espacio de direccionamiento de 1 mega.
- Modo virtual-8086: se utiliza memoria segmentada con la opción de memoria paginada. El manejo de las interrupciones y las excepciones, que se tratan como si estuviese ejecutando en modo protegido, o sea el procesador cambia de modo virtual a modo protegido antes de llamar al servicio de interrupción, cuando se ejecuta la instrucción IRET, el procesador vuelve a cambiar el modo.
- Modo protegido: orientado a la gestión segura de entorno multitarea, como la estabilidad del sistema, la protección de memoria y el soporte hardware para la gestión de memoria virtual, a partir del 386 se agregó la paginación.

Tabla de registros de propósito general.

8 de 32 bits, que se utilizan en modo herencia, mientras que en modo largo son 16 registros identificados, todos de 64bits de largo.

Las instrucciones asociadas con estos registros son las de transferencia, que permiten la carga de datos en registros internos desde la memoria principal y que permiten la transferencia de datos alojados en los registros hacia la memoria principal, además de instrucciones de tipo aritmético o lógico que operan sobre datos alojados en ellos.

Tabla de registros MMXn (64 bits).

En modo legacy o compatible se utilizan los 8 registros de 64 bits para instrucciones MMXn y los 9 primeros registros de 128bits denominados XMMn de la tabla de registros XXMn.

Las instrucciones vectoriales pueden ejecutar una operación que afecte, de manera simultánea, a múltiples datos en forma independiente en cada uno de ellos.

Tabla de registros MMXn (128bits).

En modo 64 los registros aumentan al doble en capacidad y numero, se denominan XMM0 a XMM15.

Las instrucciones asociadas a ellos se llaman instrucciones multimedia de 128bits y son de tipo SIMD. Permiten la operación de cargas y almacenamiento. Estas instrucciones se utilizan en aplicaciones de cálculo científico y multimedia.

Tabla de registros X87.

Identificados como de FPR0 a FPR7 son de 80 bits. Las instrucciones asociadas se llaman instrucciones de punto flotante x87 y se utilizan en modo herencia.