



Tecnológico Nacional de México



Instituto Tecnológico de Saltillo

Unidad II

“Estructura y funcionamiento de la Unidad Central de Procesamiento.”

Arquitectura de computadoras

María Fernanda Pérez Santana

No. Control 19052234

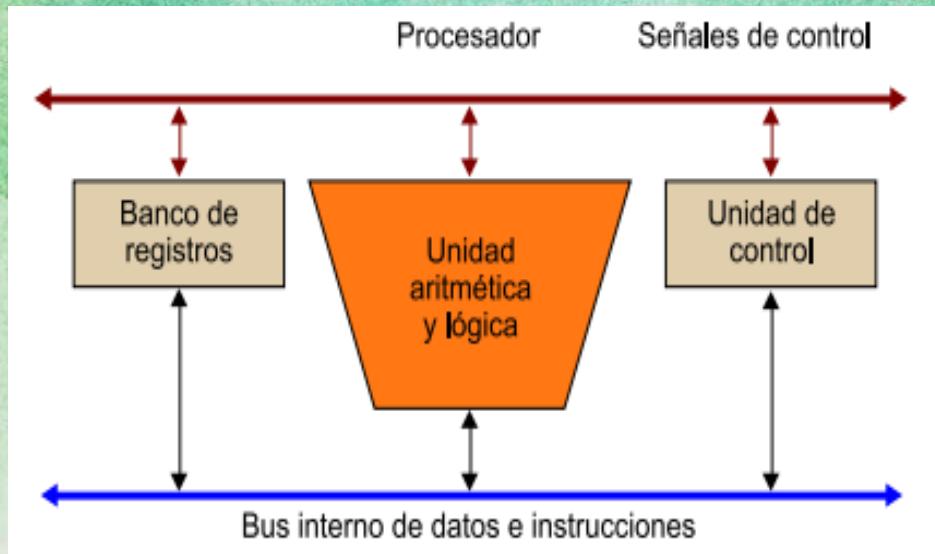
2.1 Organización del procesador.

La función principal de un procesador es ejecutar instrucciones y la organización que tiene viene condicionada por las tareas que debe realizar y por cómo debe hacerlo.

Para ejecutar las instrucciones, todo procesador dispone de tres componentes principales:

- 1) Un conjunto de registros: espacio de almacenamiento temporal de datos e instrucciones dentro del procesador.
- 2) Unidad aritmética y lógica o ALU : circuito que hace un conjunto de operaciones aritméticas y lógicas con los datos almacenados dentro del procesador.
- 3) Unidad de control: circuito que controla el funcionamiento de todos los componentes del procesador. Controla el movimiento de datos e instrucciones dentro y fuera del procesador y también las operaciones de la ALU.

La organización básica de los elementos que componen un procesador y el flujo de información entre los diferentes elementos se ve en el esquema siguiente:



Aparte de los tres componentes principales, es necesario disponer de un sistema que permita interconectar estos componentes. Este sistema de interconexión es específico para cada procesador. Distinguimos dos tipos de líneas de interconexión: líneas de control, que permiten gobernar el procesador, y líneas de datos, que permiten transferir los datos y las instrucciones entre los diferentes componentes del procesador. Este sistema de interconexión tiene que disponer de una interfaz con el bus del sistema.

Dentro de los microprocesadores se pueden encontrar otras unidades funcionales del computador. Por ejemplo:

- Unidades de ejecución SIMD: unidades especializadas en la ejecución de instrucciones SIMD (single instruction, multiple data), instrucciones que trabajan con estructuras de datos vectoriales, como por ejemplo instrucciones multimedia
- Memoria caché
- Unidad de gestión de memoria (MMU): gestiona el espacio de direcciones virtuales, traduciendo las direcciones de memoria virtual a direcciones de memoria física en tiempo de ejecución. Esta permite proteger el espacio de direcciones de un programa del espacio de direcciones de otros programas y también permite separar el espacio de memoria del sistema operativo del espacio de memoria de los programas de usuario.
- Unidad de punto flotante (FPU): unidad especializada en hacer operaciones en punto flotante; puede funcionar de manera autónoma, ya que dispone de un conjunto de registros propio.

2.2 Estructura de registros

En la estructura interna de la CPU se indican los caminos de transferencia de datos y de control lógico, que incluyen un elemento con el rótulo bus interno de la CPU.

Este elemento es necesario para transferir datos entre los diversos registros y la ALU, ya que ésta en realidad sólo opera con datos de la memoria interna de la CPU.

Los registros de la CPU son:

2.2.1 Registros visibles al usuario

Son aquellos que pueden ser referenciado por medio del lenguaje maquina que ejecuta la CPU, los registro que normalmente disponibles son:

Registros de propósito general, son aquellos que pueden guardar tanto datos como direcciones

Registro de datos, que pueden ser asignados por el programador a diversas funciones. En algunos casos son de propósito general y pueden ser empleados por cualquier instrucción de máquina que lleve a cabo operaciones sobre los datos.

Registros de direcciones, contienen direcciones en la memoria principal de datos y este tipo de registro puede ser de propósito general o estar a un modo específico de direccionamiento.

Códigos de condición, también conocidos como indicadores o flags. Los códigos de condición, son bits activados por el procesador como resultado de determinadas operaciones.

2.2.2 Registros de control y de estado

Se utilizan para controlar las operaciones del procesador, la mayor parte de estos registros no son visibles al usuario y algunos pueden ser accesibles a las instrucciones de maquina ejecutadas en un modo de control.

Registro de direcciones de memoria (MAR), el cual contiene la dirección en donde se efectuará la próxima lectura o escritura de datos. El numero de direcciones depende del tamaño de la MAR.

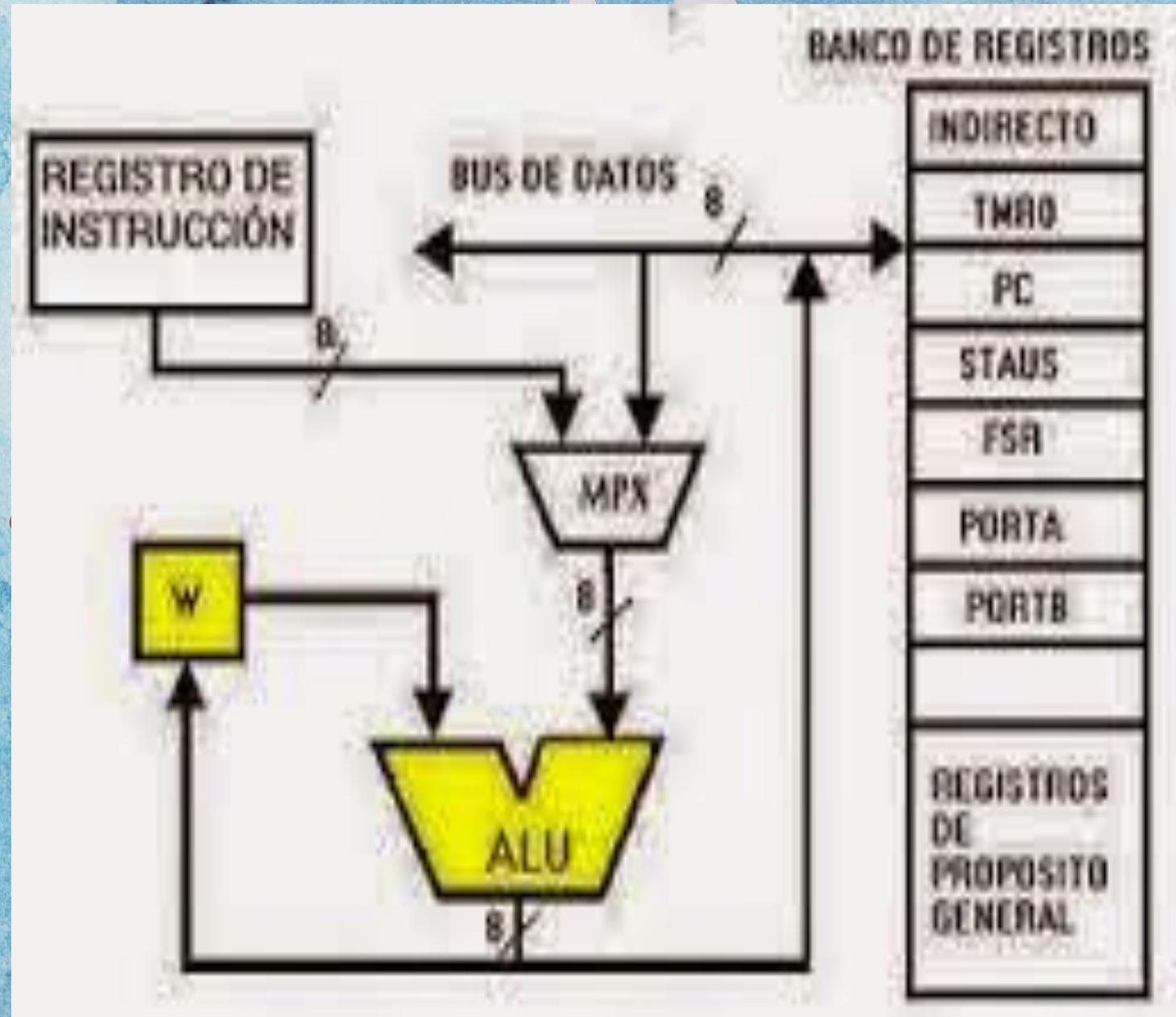
Registro de datos de memoria (MBR), contiene los datos que van a ser escritos en la memoria o los que fueron leídos en ella.

Registro de direcciones de entrada y salida (I/O AR), especifica al dispositivo ya sea de entrada o salida

Registro de datos de entrada y salida (I/O BR), es una área temporal en donde se lleva a cabo el intercambio de datos entre el procesador y el dispositivo de entrada y salida que esta especificado en IOAR.

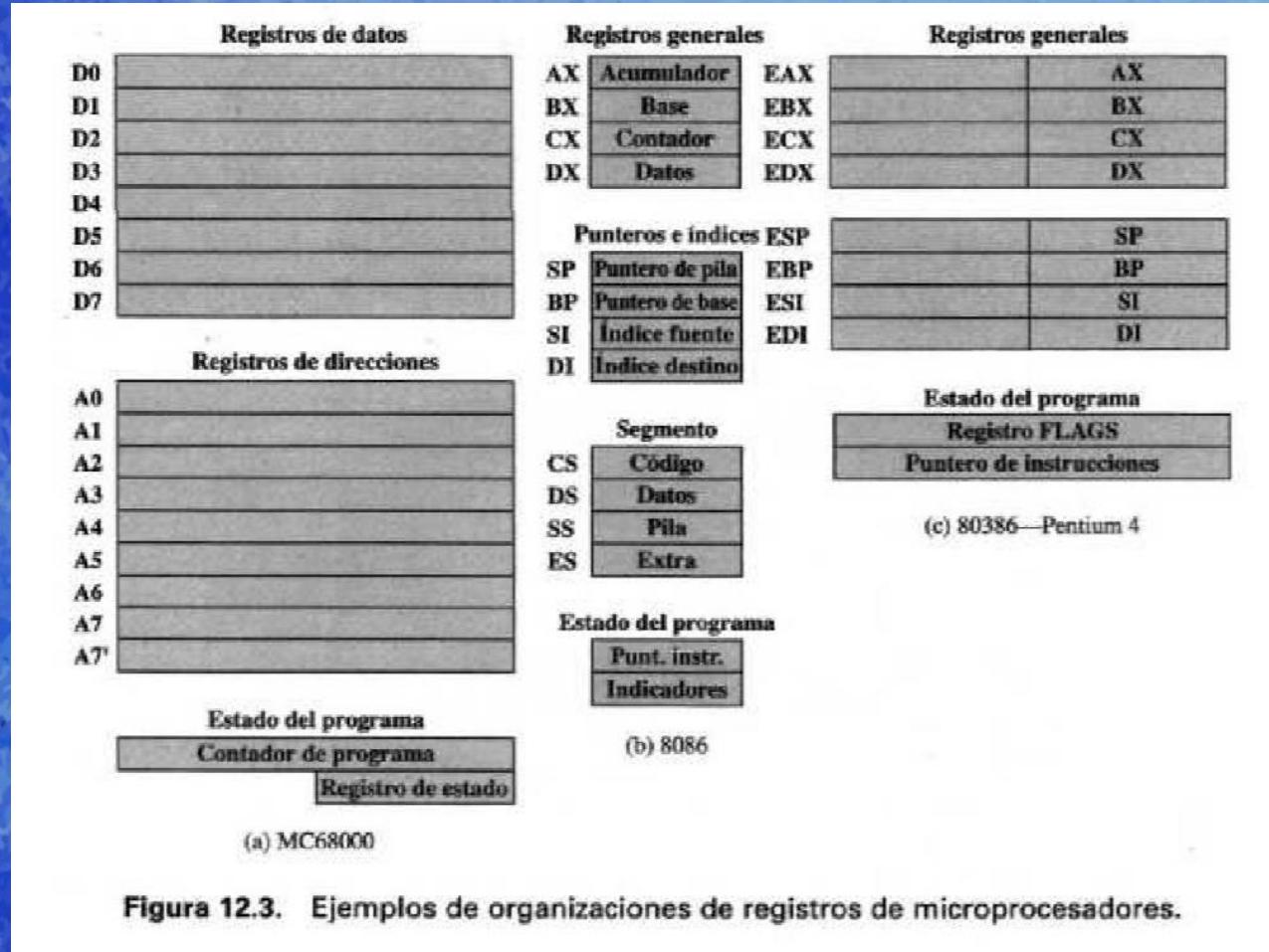
Registro de instrucciones (IR), contiene la dirección de la siguiente instrucción que se va a ejecutar.

Palabras de estado del programa (PSW), contiene códigos de condición junto con otras informaciones de estado como el signo, acarreo, desbordamiento, entre otras.



2.2.3 Ejemplos de organización de registros de CPU reales.

examinamos dos microprocesadores de 16 bits que fueron diseñados aproximadamente al mismo tiempo el Motorola MC68000 [STR179] y el Intel 8086 [MORS78]



Las figuras 12.3 (a) y (b) representan la organización de registros de cada uno de ellos; los registros estrictamente internos, tales como el registro de dirección de memoria, no se muestran.

En la figura 12.3 (c) se ilustra un segundo aspecto instructivo acerca del diseño de la organización de los registros. Esta figura muestra la organización de los registros visibles por el usuario en el Intel 80386 [elay85], un microprocesador de 32 bits diseñado como una ampliación del 8086

Figura 12.3. Ejemplos de organizaciones de registros de microprocesadores.

El *MC68000* distribuye sus registros de 32 bits en ocho de datos y nueve de direcciones. los ocho registros de datos se usan principalmente para manipulación de datos y también se usan en direccionamiento como registros índice.

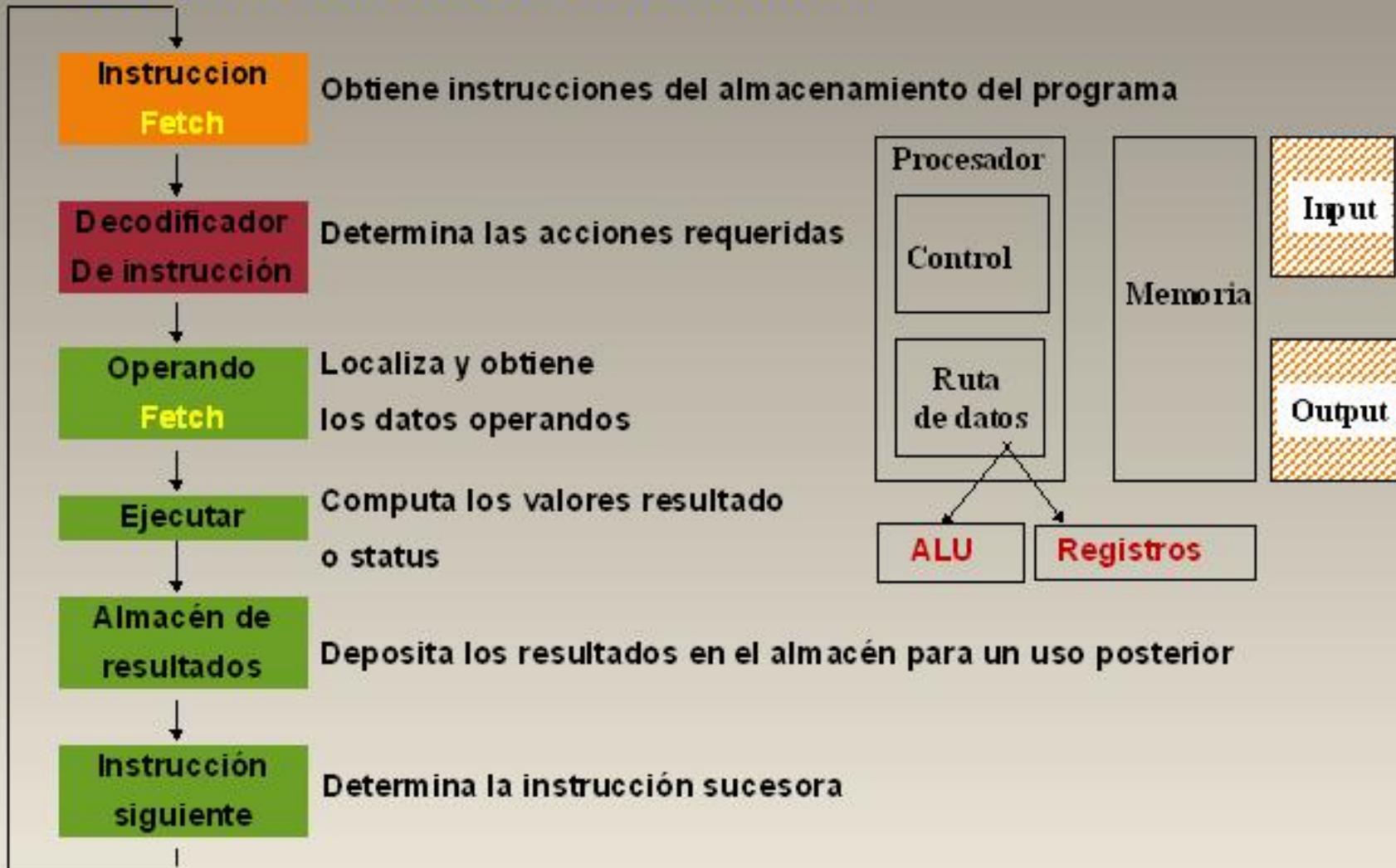
El *Intel 8086* usa un enfoque diferente para la organización de los registros. Cada uno de los registros tiene un uso especial, aunque algunos registros se pueden emplear también para un uso general. El 8086 contiene cuatro registros de datos de 16 bits que son direccionables como registros de bytes o como registros de 16 bits, y cuatro registros punteros e índices de 16 bits. Los registros de datos pueden utilizarse como de uso general en algunas instrucciones. En otras, los registros se usan implícitamente.

2.3 El ciclo de instrucción.

Un ciclo de instrucción (también llamado ciclo de fetch-and-execute o ciclo de fetchdecode-execute en inglés) es el período que tarda la unidad central de proceso (CPU) en ejecutar una instrucción de lenguaje máquina.

Comprende una secuencia de acciones determinada que debe llevar a cabo la CPU para ejecutar cada instrucción en un programa. Cada instrucción del juego de instrucciones de una CPU puede requerir diferente número de ciclos de instrucción para su ejecución. Un ciclo de instrucción está formado por uno o más ciclos máquina.

Ciclo de Fetch Decode Execute en MP



2.3.1.- Ciclo de Fetch-DecodeExecute

- 1.- Buscar la instrucción en la memoria principal
- 2.- Decodificar la instrucción
- 3.- Ejecutar la instrucción
- 4.- Almacenar o guardar resultados

Ciclo de búsqueda

- Los pasos 1 y 2 del ciclo de instrucción se conocen como ciclo de búsqueda (fetch). Estos pasos son idénticos en todas las instrucciones.
- El ciclo de búsqueda procesa la instrucción a partir de la palabra de instrucción, que contiene el código de operación y el operando.

Ciclo de ejecución.

Los pasos 3 y 4 del ciclo de instrucción se conocen como ciclo de ejecución. Estos pasos cambiarán con cada tipo de instrucción.

- El primer paso del ciclo de ejecución es el proceso de memoria, en que los datos se transfieren entre la CPU y el módulo de entrada/salida (I/O).
- A continuación se produce el proceso de datos, que usa operaciones matemáticas así como operaciones lógicas en referencia a los datos.
- Después tiene lugar el paso de alteraciones centrales, que son una secuencia de operaciones, por ejemplo una operación de salto. El último paso es una operación combinada de todos los otros pasos.

2.2.3 La segmentación de instrucciones

Es una técnica que permite implementar el paralelismo a nivel de instrucción en un único procesador. La segmentación intenta tener ocupadas con instrucciones todas las partes del procesador dividiendo las instrucciones en una serie de pasos secuenciales que efectuarán distintas unidades de la CPU, tratando en paralelo diferentes partes de las instrucciones. Permite una mayor tasa de transferencia efectiva por parte de la CPU que la que sería posible a una determinada frecuencia de reloj, pero puede aumentar la latencia debido al trabajo adicional que supone el propio proceso de la segmentación.

Número de pasos

El número de pasos dependientes varían según la arquitectura de la máquina. Algunos ejemplos:

- ❖ La segmentación RISC clásica comprende:
 - Lectura de instrucción
 - Decodificación de instrucción y lectura de registro
 - Ejecución
 - Acceso a memoria
 - Escritura de vuelta en el registro
- ❖ Las microcontroladoras Atmel AVR y PIC disponen cada una de segmentación de dos etapas.
- ❖ Muchos diseños incluyen segmentación de 7, 10 e incluso 20 etapas (como es el caso del Pentium 4 de Intel).

2.3.3 Conjunto de instrucciones: características y funciones.

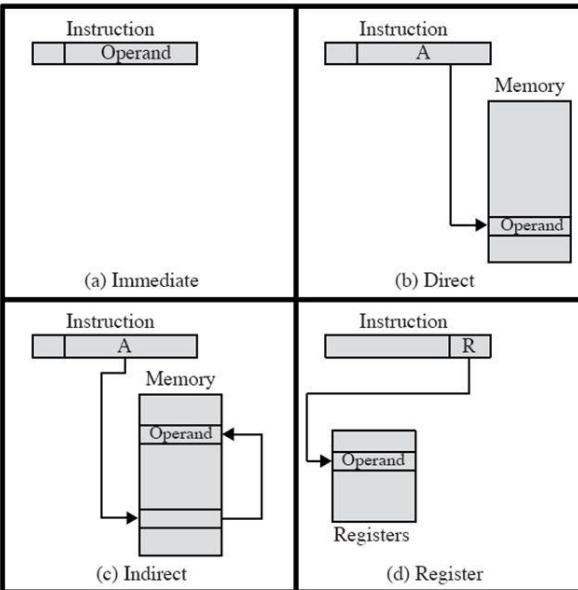
Los conjuntos de instrucciones de las máquinas deben tender a poseer una serie de propiedades, bastante ideales e imprecisas, que pueden resumirse en las siguientes:

- ❖ El conjunto de instrucciones de un computador debe ser completo en el sentido de que el número de instrucciones de ese programa no debe ser demasiado elevado.
- ❖ Los juegos de instrucciones también tienen que ser eficientes, esto significa que las funciones más necesarias deben poder realizarse usando pocas instrucciones.
- ❖ El conjunto de instrucciones de una máquina debe ser regular, es decir debe ser simétrico (por ejemplo, si existe una instrucción de desplazamiento a la izquierda, debe haber otra de desplazamiento a la derecha, etc.) y ortogonal, es decir, deben poder combinarse, en la medida de lo posible, todos las operaciones con todos los tipos de datos y modos de direccionamiento.
- ❖ Un mínimo para llegar a ese compromiso se consigue con los tipos de instrucciones siguientes: Instrucciones de transferencia de datos. Instrucciones aritméticas. Instrucciones lógicas. Instrucciones de control del flujo del programa (bifurcaciones, bucles, procedimientos, etc.). Instrucciones de entrada y salida.

2.3.4 Modos de direccionamiento y formatos

Son las diferentes maneras de especificar un operando dentro de una instrucción. Un modo de direccionamiento especifica la forma de calcular la dirección de memoria efectiva de un operando mediante el uso de la información contenida en registros dentro de una instrucción de la máquina.

Modos de Direccionamiento



Direccionamiento implícito

Depende solamente de la instrucción, es decir, la instrucción no lleva parámetros.
Particularmente en instrucciones que no accesan memoria, o bien que tienen una forma específica de accesarla.
Ejemplos: PUSHF, POPF, NOP

Modo registro

Usa solamente registros como operandos
Es el más rápido, pues minimiza los recursos necesarios (toda la información fluye dentro del EU del CPU)
Ejemplo:
MOV AX, BX

Modo inmediato

Tiene dos operandos: un registro y una constante que se usa por su valor.
El valor constante no se tiene que buscar en memoria, pues ya se obtuvo al hacer el “fetch” de la instrucción.
Ejemplo:
MOV AH, 9

Modo directo

Uno de los operandos involucra una localidad específica de memoria, el valor constante se tiene que buscar en memoria; en la localidad especificada.

Es más lento que los anteriores, pero es el más rápido para ir a memoria, pues ya “sabe” la localidad, la toma de la instrucción y no la tiene que calcular.

Ejemplo:

MOV AH, [0000]

MOV AH, Variable

Estas dos instrucciones serían equivalentes

Modo indirecto

Se usan los registros SI, DI como apuntadores

El operando indica una localidad de memoria, cuya dirección (sólo la parte desplazamiento) está en SI o DI.

Es más lento que los anteriores, pues tiene que “calcular” la localidad

Ejemplos:

MOV AL, [SI]

MOV BL, ES:[SI] ; Aquí se dice que se usa un “segment override”, donde se indica que en vez de usar el segmento de datos por defecto, se use en su lugar como referencia el segmento extra.

Modo indexado de base

Formato:

[BX o BP + SI o DI (opcionales) + constante (opcional)]

BX o BP indica una localidad base de la memoria, A partir de BX o BP, se puede tener un desplazamiento variable y uno constante, La diferencia es el segmento sobre el que trabajan por defecto:

BX por defecto en el segmento de datos

BP por defecto en el segmento de pila.

Ejemplos:

MOV AX, [BX]

MOV DX, [BX+2]

2.4 Casos de estudio de CPU reales.

I-8086

Los modos de direccionamiento del 8086 son muy irregulares. Los registros del procesador, se usan para contener los datos con que se está trabajando puesto que el acceso a los registros es mucho más rápido que los accesos a memoria. Se pueden realizar operaciones aritméticas y lógicas, comparaciones, entre otras.

Hay un campo para un registro, que especifica uno de los operandos, y otros dos campos para el otro.

Los modos del 8086 son indirectos por registro, indexados o directos por registro.

Registros Propósito General:

- ❖ Registro AX: El registro AX es el registro acumulador, es utilizado para operaciones que implican entrada/salida, y multiplicación y división
- Registro BX: El registro BX es el registro base, y es el único registro de propósito general que puede ser un índice para direccionamiento indexado
- ❖ Registro CX: El registro CX es conocido como el registro contador. Puede contener un valor para controlar el número de veces que un ciclo se repite.
- ❖ Registro DX: El registro DX es el registro de datos.

Registros Índice:

- ❖ Registro SI: El registro índice fuente de 16 bits es requerido por algunas operaciones con cadenas de caracteres.
- ❖ Registro DI: El registro índice destino también es requerido por algunas operaciones con cadenas de caracteres.
- ❖ Registros Apuntadores:
- ❖ Registro SP: Apuntador de pila de 16 bits proporciona un valor de desplazamiento que se refiere a la palabra actual que está siendo procesada en la pila.
- ❖ Registro BP: El apuntador base de 16 bits facilita la referencia de parámetros dentro de la pila.

Motorola 68000

En el Motorola 68000 el mismo direccionamiento lleva implícito el tipo de registro sobre el que trabaja (direcciones o datos). Está basado en dos bancos de 8 registros de 32 bits. Un banco es de datos (Dn) y el otro de punteros (An). Además contiene un contador de programa de 32 bits y un registro de estado de 16 bits. Los registros de datos (D0 a D7) se pueden usar como registros de 32 bits (.l), 16 bits (.w) y 8 bits (.b). Cualquiera de ellos puede usarse como acumulador, índice o puntero. Acepta los siguientes modos de direccionamiento:

- Implícito (o inherente).
- Inmediato.
- Absoluto.
- Directo a registro.
- Indirecto.
- Relativo a PC con desplazamiento.
- Relativo a PC con índice y desplazamiento.

I-8086: Los registros del procesador, se usan para contener los datos con que se está trabajando puesto que el acceso a los registros es mucho más rápido que los accesos a memoria. Se pueden realizar operaciones aritméticas y lógicas, comparaciones, entre otras. Los modos del 8086 son indirectos por registro, indexados o directos por registro.

Motorola 68000: El mismo direccionamiento lleva implícito el tipo de registro sobre el que trabaja (direcciones o datos). Está basado en dos bancos de 8 registros de 32 bits. Un banco es de datos (Dn) y el otro de punteros (An). Además contiene un contador de programa de 32 bits y un registro de estado de 16 bits.