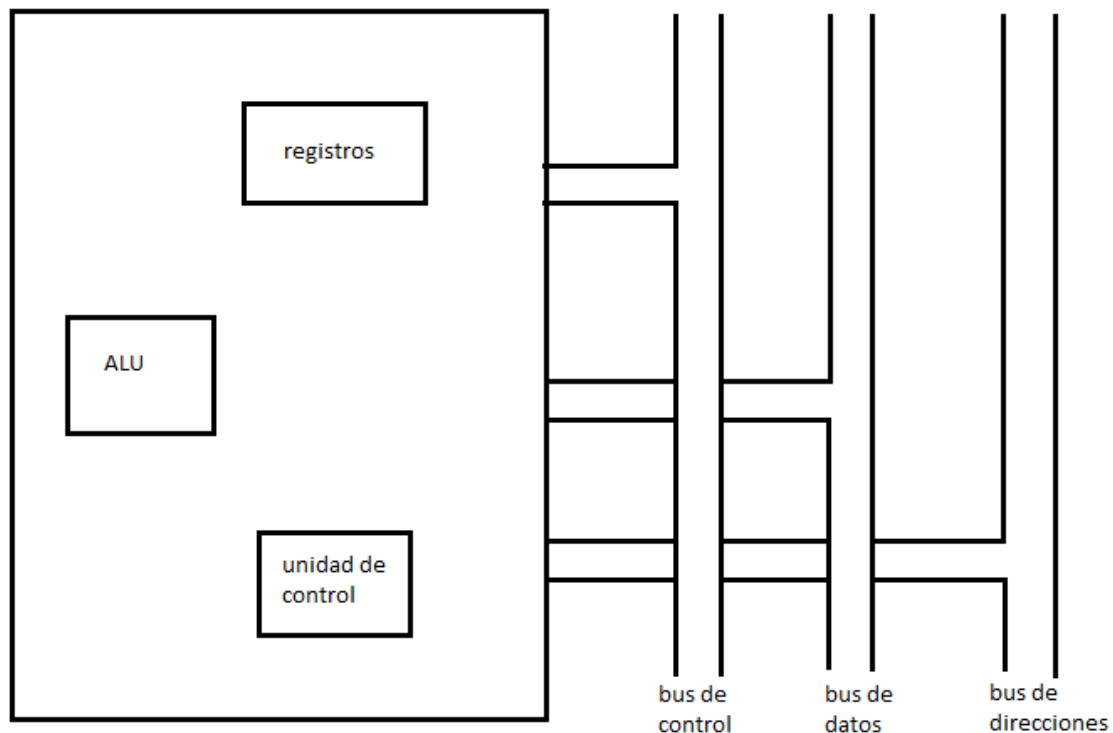


## Unidad 4 Unidad Central de Procesamiento

Organización de la CPU. Descripción de microprocesadores actuales. Modelo de ejecución de instrucciones. Ciclo de instrucción, fases. Comunicación CPU – memoria, dato y dirección. Interconexión de subsistemas, buses, ejemplos reales. Concepto de instrucción. Conjunto de instrucciones: operaciones, formato y modos de direccionamiento. Organización de registros. Lenguaje de máquina y assembly.

### CPU

Estructura de la CPU y buses



Es el encargado de ejecutar los programas, desde el sistema operativo hasta las aplicaciones de usuario; sólo ejecuta instrucciones programadas en lenguaje de bajo nivel, realizando operaciones aritméticas y lógicas simples, tales como sumar, restar, multiplicar, dividir, las lógicas binarias y accesos a memoria.

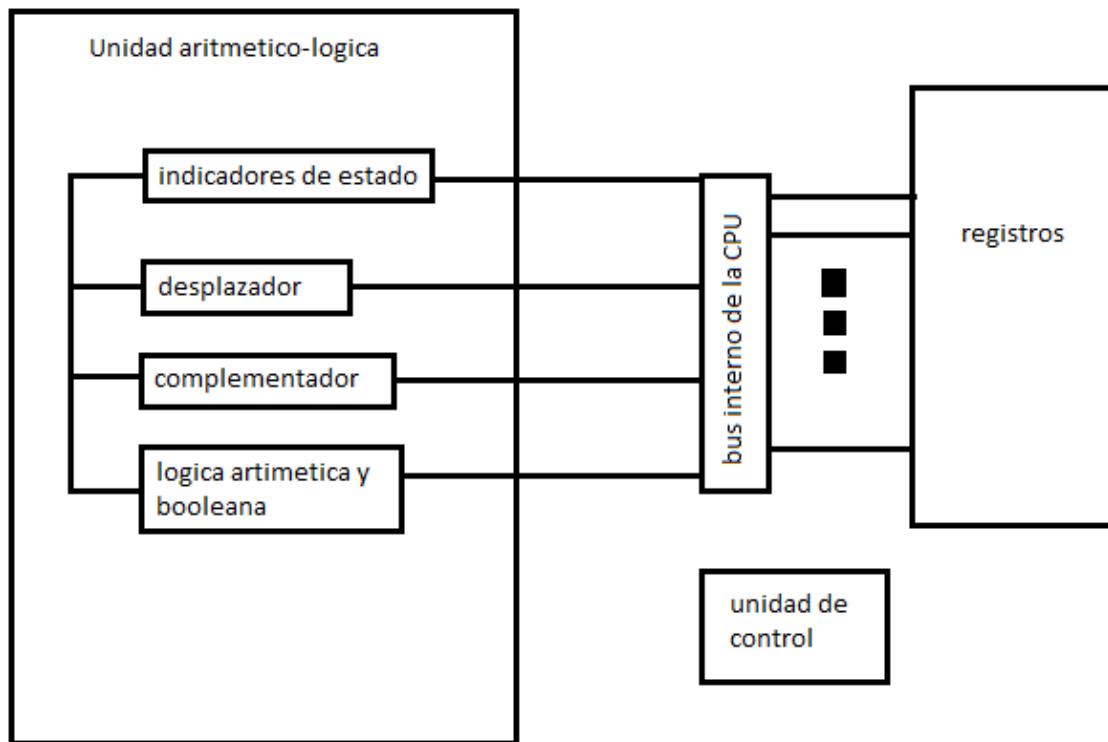
Esta unidad central de procesamiento (CPU) está constituida, esencialmente, por registros, una unidad de control, una unidad aritmético-lógica (ALU) y una unidad de cálculo en coma flotante (conocida antiguamente como «coprocesador matemático»).

Las cosas que debe hacer una CPU son:

- Captar instrucciones
- Interpretar instrucciones
- Captar datos
- Procesar datos

- Escribir datos

#### estructura interna de la CPU



#### Organización de los registros:

Un computador emplea una jerarquía de memoria, siendo en los niveles más altos de la jerarquía, la memoria más rápida, pequeña y cara (por bit). Dentro de la CPU hay un conjunto de registros que funciona como un nivel de memoria por encima de la memoria principal y de la cache en la jerarquía.

- Registros visibles para el usuario: permite al programador minimizar las referencias a memoria principal cuando optimiza el uso de registros.
  - De uso general
  - Datos
  - Direcciones
  - Códigos de condición
- Registros de control y estado: son utilizados por la unidad de control para controlar el funcionamiento de la CPU, y por programas privilegiados el sistema operativo para controlar la ejecución de programas.

Son esenciales 4 registros para la ejecución de una instrucción:

- Contador de programa (program counter, PC) contiene la dirección de la instrucción a captar
- Registro de instrucción (Instruction register, IR) contiene la instrucción captada más recientemente.
- Registro de dirección de memoria (memory address register, MAR) contiene la dirección de una posición de memoria.

- Registro intermedio de memoria (memory buffer register, MBR) contiene la palabra de datos a escribir en memoria o la palabra leída mas recientemente.

Esos cuatro registros se usan para transferir datos entre la CPU y la memoria. La ALU puede tener acceso directo al MBR. También puede haber registros intermedios adicionales en torno a la ALU e intercambian datos con MBR y los registros visibles para el usuario.

Además, se incluye un registro PSW(program status word) que contiene normalmente códigos de condición además de información de estado. Entre los campos comunes que contiene se encuentran

- Signo
- Cero
- Acarreo
- Igual
- Desbordamiento
- Interrupciones habilitadas/inhabilitadas
- Supervisor

También puede existir otro registro que sea un puntero a memoria donde se almacena información de estado adicional, ej.: bloques de control de procesos.

## Buses

Un bus es un camino de comunicación entre dos o mas dispositivos. Una característica clave de un bus es que se trata de un medio de transmisión compartido. Al bus se conectan varios dispositivos conectados al bus puedan acceder a ella.

Si dos dispositivos transmiten durante el mismo periodo de tiempo, sus señales pueden solaparse y distorsionarse, de modo que solo un dispositivo puede transmitir con éxito en un momento dado.

Un bus esta constituido por varios caminos de comunicación o líneas. Una línea es capaz de transmitir señales binarias representadas por 1 y por 0. En un intervalo de tiempo se puede transmitir una secuencia de dígitos binarios a través de una única línea. Se pueden utilizar varias líneas del bus para transmitir dígitos binarios simultáneamente.

El bus que conecta los componentes principales del computador se denomina bus de sistema.

## Estructura del bus

Las líneas del bus de sistema se pueden clasificar en tres grupos funcionales

Líneas de datos, de direcciones y de control. Además, pueden existir líneas de alimentación para suministrar energía a los módulos conectados al bus.

Las líneas de datos proporcionan un camino para transmitir datos entre los módulos del sistema. El conjunto constituido por estas líneas se denomina **bus de datos**. Generalmente consta de 8, 16, 32 líneas distintas (anchura de bus). Puesto que cada línea solo transmite un

bit, el número de líneas determina cuantos bits se pueden transferir al mismo tiempo. Por ejemplo, si el bus de datos tiene una anchura de 8bits, y las instrucciones son de 16bits entonces el procesador debe acceder al modulo de memoria dos veces por cada ciclo de instrucciones.

Las líneas de direcciones se utilizan para designar la fuente o el destino del dato situado en el bus de datos.

Las líneas de control se utilizan para controlar el acceso y el uso de las líneas de datos y direcciones. Puesto que esas líneas son compartidas por todos los componentes debe existir una forma de controlar su uso. Las señales de control, transmiten tanto ordenes como información de temporización entre los módulos del sistema.

### **Jerarquía de buses**

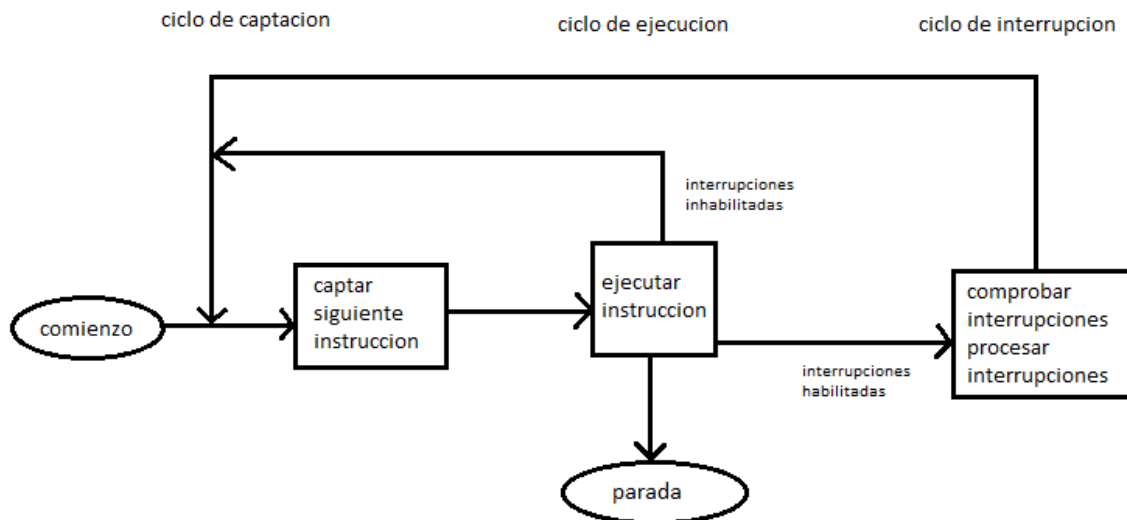
A mayor cantidad de dispositivos, mayor retardo de propagación.

El bus puede convertirse en un cuello de botella a medida que las peticiones de transferencia acumuladas se aproximan a la capacidad el bus. Este problema se puede resolver en alguna medida incrementando la velocidad a la que el bus puede transferir los datos y utilizando buses mas anchos.

Hay un bus local que conecta el procesador a una memoria cache y al que puede conectarse también uno o más dispositivos locales. El controlador de memoria cache conecta la cache no solo al bus local, sino también al bus de sistema, donde se conectan todos los módulos de memoria principal. El uso de una cache alivia la exigencia de soportar los accesos frecuentes del procesador a memoria principal. Las transferencias de E/S con la memoria principal a través del bus de sistema no interfieren la actividad del procesador.

Una solución consiste en utilizar uno o mas buses de expansión. La interfaz del bus de expansión regula las transferencias de datos entre el bus de sistema y los controladores conectados al bus de expansión.

### **Ciclo de instrucciones:**



Esto siempre incluye dos subciclos: captación(llevar una instrucción de la memoria a CPU) y ejecución(interpretar el código de operación y llevarla a cabo).

### Ciclo de captación

Se da al principio del ciclo de una instrucción y hace que una instrucción sea captada de la memoria. Hay 4 registros involucrados.

- Contador de programa (program counter, PC) contiene la dirección de la instrucción a captar
- Registro de instrucción (Instruction register, IR) contiene la instrucción captada más recientemente.
- Registro de dirección de memoria (memory address register, MAR) contiene la dirección de una posición de memoria.
- Registro intermedio de memoria (memory buffer register, MBR) contiene la palabra de datos a escribir en memoria o la palabra leída mas recientemente.

La secuencia de eventos sobre los registros es la siguiente. Al comienzo la dirección de la siguiente instrucción a ejecutar esta en el contador de programa. El primer paso es llevar esa dirección al registro de dirección de memoria. El segundo paso es traer la instrucción. La dirección deseada (en MAR) se coloca en el bus de direcciones. La unidad de control emite una orden read por el bus de control. El resultado aparece en el bus de datos y se copia en el registro intermedio de memoria. Es necesario incrementar el PC en 1 para que esté preparado para la siguiente instrucción. Estas últimas dos acciones se realizan en simultáneo. El tercer paso es transferir el contenido de MBR al registro de instrucción. Esto libera MBR para su uso durante un posible ciclo indirecto.

### Ciclo indirecto

La principal línea de actividad consiste en alternar las actividades de captación y ejecución de instrucciones. Después de que una instrucción sea captada, es examinada para determinar si implica algún direccionamiento indirecto. Tras la ejecución se puede procesar una interrupción antes de la captación de la siguiente instrucción.

## **Ciclo de interrupción**

Cuando termina el ciclo de ejecución, se realiza una comprobación para determinar si ha ocurrido alguna interrupción habilitada.

## **Ciclo de ejecución**

## **Flujo de datos**

Una vez concluido el ciclo de captación, la unidad de control examina los contenidos de IR para determinar si contiene un campo de operando que use direccionamiento indirecto. De ser así, se lleva a cabo un ciclo indirecto.

## **Arquitectura RISC (repertorio reducido de instrucciones)**

Se basa en tres elementos:

- Gran numero de registros de propósito general
- Repertorio de instrucciones limitado y sencillo
- Énfasis en optimización de la segmentación de instrucciones

Otros elementos comunes son:

- Una instrucción por ciclo
- Operaciones registro a registro
- Modos de direccionamiento sencillos
- Formatos de instrucción sencillos

## **Instrucciones**

### **Elementos de una instrucción de maquina**

- Código de operación: especifica la operación a realizar, ej. suma. La operación se indica mediante un código binario denominado código de operación.
- Referencia a operandos fuente: la operación puede implicar a uno o más operandos fuentes, es decir operandos que son entradas para la instrucción
- Referencia al operando resultado: la operación puede producir un resultado.
- Referencia a la siguiente instrucción: dice a la CPU de donde captar la siguiente instrucción tras completarse la ejecución de la instrucción actual.

La siguiente instrucción a captar esta e memoria principal o bien, en memoria secundaria. En la mayoría de los casos la siguiente instrucción a captar sigue inmediatamente a la instrucción en ejecución. En dichos casos, no hay referencia explícita a la siguiente instrucción. Cuando sea

necesaria una referencia explícita. Debe suministrarse la dirección de memoria principal o de memoria virtual.

Los operandos fuente y resultado pueden estar en alguna de las siguientes áreas:

- Memoria principal o virtual como en las referencias a instrucciones siguientes, debe indicarse la dirección de memoria principal o memoria virtual.
- Registro de la CPU: salvo raras excepciones, una CPU contiene uno o más registros que pueden ser referenciados por instrucciones máquina. Si solo existe un registro, la referencia a él puede ser implícita. Si existe más de uno, cada registro tendrá asignado un número único y la instrucción debe contener el número del registro deseado.
- Dispositivos de e/s: la instrucción debe especificar el módulo o dispositivo de E/S para la operación. En el caso de e/s asignadas en memoria, se dará otra dirección de memoria principal o virtual.

### **Representación de las instrucciones**

Se representan por una secuencia de bits y está dividida en campos. La instrucción se escribe en un registro de la CPU (IR - INSTRUCTION RECORD). La CPU debe ser capaz de extraer los datos de los distintos campos de la instrucción para realizar la operación requerida.

### **Tipos de instrucciones**

Una CPU debería tener un conjunto de instrucciones que permitieran al usuario formular cualquier tarea de procesamiento de datos. Teniendo esto en cuenta los tipos de instrucciones se pueden clasificar de la siguiente manera:

**De procesamiento de datos:** instrucciones aritméticas y lógicas.

**De almacenamiento de datos:** instrucciones de memoria.

**De transferencia de datos:** instrucciones de E/S.

**De control:** instrucciones de comprobación y de bifurcación.

Las instrucciones aritméticas proporcionan capacidad computacional para procesar datos numéricos. Las instrucciones lógicas operan sobre los bits de una palabra, en lugar de considerarlos números. Estas operaciones se realizan principalmente con datos de registros de la CPU por lo tanto debe haber instrucciones de memoria para transferir los datos entre la memoria y los registros.

Una de las formas tradicionales de describir la arquitectura de un procesador es en términos del número de direcciones contenidas en cada instrucción.

La mayoría de los CPU trabajan con una, dos o tres instrucciones siendo implícita la dirección de la instrucción siguiente (obtenida a partir del contador de programa).

### **Diseño del repertorio de instrucciones**

Los aspectos fundamentales del diseño de un repertorio de instrucciones son:

- Repertorio de operaciones: cuantas y que operaciones considerar y cuan complejas deben ser.
- Tipos de datos: los distintos tipos de datos con los que se efectúan operaciones.
- Formatos de instrucciones: longitud de la instrucción en bits, numero de direcciones, tamaño de los distintos campos, etc.
- Registros: numero de registros de la CPU que pueden ser referenciados por instrucciones y su uso.
- Direccionamiento: el modo o modos de direccionamiento mediante los cuales puede especificarse la dirección de un operando.

La longitud de una instrucción es el aspecto mas básico y se ve afectada por el tamaño de la memoria, su organización, la estructura de buses, la complejidad y velocidad de la CPU, etc.

### **Formato de una instrucción**

Dentro de un set de instrucciones, puede haber diferentes formatos de instrucciones. Los elementos comunes que tienen que estar presentes son:

- Codop (código de operación)
- Referencia a operando(puede haber referencias a uno, dos o más operandos)
- Modo de direccionamiento por cada operando.

### **Tipos de operandos**

Las instrucciones maquina operan con datos. Las categorías generales más importantes de datos son:

- Direcciones
- Números
- Caracteres
- Datos lógicos

### **Números:**

Todos los lenguajes maquina incluyen tipos de datos numéricos, incluso el procesamiento no numérico se necesitan números para usar de contadores.

Cabe destacar que los números almacenados en un computador están limitados, a diferencia de los utilizados en las matemáticas ordinarias. Esto es cierto en dos sentidos: en primer lugar, hay un límite para la magnitud de los números representables; en segundo lugar, en el caso de números en coma flotante su representación está limitada. Por tanto el programador debe ser consciente de las consecuencias del redondeo, el desbordamiento o el desbordamiento a cero.

Usualmente existen tres tipos de datos numéricos:

- Enteros o en coma fija.
- En coma flotante
- En decimal.



**Caracteres:**

El texto es una forma bastante común de datos. Cada carácter es representado en código ASCII por un patrón distinto de 7 bits. El octavo bit se utiliza como paridad para detectar errores.

**Datos lógicos:**

Cuando los datos son vistos como n elementos o datos de 1 bit (que puede tener valor 0 o 1) se consideran datos lógicos.

**Tipos de operaciones**

- Transferencia de datos
- Aritméticas
- Lógicas
- De conversión
- De E/S
- De control del sistema
- De control de flujo

Transferencia de datos: es el tipo de instrucción maquina más básico. La instrucción debe especificar varias cosas: en primer lugar debe especificar las posiciones de los operandos fuente y destino. Cada posición podría ser de memoria, un registro o la cabecera de una pila. En segundo lugar debe indicarse la longitud de los datos a transferir. En tercer lugar, como en todas las instrucciones con operandos, debe especificarse el modo de direccionamiento para cada operando.

En términos de la acción de la CPU las operaciones de transferencia son las más sencillas. Cuando tanto el origen como el destino son registros, la CPU simplemente hace que los datos se transfieran de un registro a otro. Si uno o ambos operandos están en memoria, la CPU debe realizar alguna o todas de las siguientes tareas:

- Calcular la dirección de memoria basándose en el modo de direccionamiento utilizado.
- Si la dirección hace referencia a memoria virtual, traducir de dirección virtual a real
- Determinar si el elemento direccionado está en la cache
- Si no, cursar la orden al modulo de memoria.

**Aritméticas:**

Proporcionan las operaciones aritméticas básicas como la suma, resta, multiplicación y división. Estas siempre se tienen para números enteros con signo. A menudo se proporcionan también para números en coma flotante y para decimales empaquetados.

Entre otras operaciones también están el absolute, negate, increment y decrement.

La ejecución de una instrucción aritmética puede implicar operaciones de transferencia de datos para ubicar los operandos como entradas a la ALU y para almacenar la salida de la ALU.

### **Lógicas:**

Estas operaciones están basadas en operaciones booleanas.

Además, la mayoría de las maquinas ofrecen diversas funciones de desplazamiento y rotación. La operación de desplazamiento aritmético trata el dato como entero con signo y no desplaza el bit de signo. La rotación preserva todos los bits con los que se está operando. Un posible uso de la rotación es ir volcando sucesivamente cada bit en la posición más a la izquierda, donde pueda ser identificado comprobando el bit de signo del dato (tratándolo como numero).

### **Conversión:**

las instrucciones de conversión son aquellas que cambia el formato u operan sobre el formato de los datos. Un ejemplo es la conversión de decimal a binario.

### **Entrada/salida:**

Las instrucciones de entrada/salida

**Ver capítulo 6**

### **Control del sistema**

Las instrucciones de control son instrucciones privilegiadas que solo se pueden ejecutar mientras el procesador esta en un estado privilegiado concreto o está ejecutando un programa de una zona privilegiada especifica de memoria. Normalmente estas instrucciones están reservadas para que las use el sistema operativo.

**Algunos ejemplos ver capítulo 11**

### **Control de flujo**

En todos los tipos de instrucciones previos, la siguiente instrucción a ejecutar es la inmediatamente posterior en memoria a la instrucción en curso. Sin embargo, una fracción significativa de las instrucciones de cualquier programa tiene como misión cambiar la secuencia de ejecución de instrucciones. Para estas instrucciones, la operación que realiza la CPU es actualizar el contador de programa para que contenga la dirección de alguna de las instrucciones que hay en memoria.

Hay varias razones por las que son necesarias estas instrucciones. Las más comunes son:

- Poder ejecutar múltiples veces una misma instrucción. Sería impensable escribir cada instrucción por separado, de modo que el uso de un bucle es fundamental.
- Tomar una decisión. En los programas es fundamental tomar decisiones si se cumple una condición específica o ejecutar otra cosa si se cumple otra condición.
- Modularización. Es muy útil realizar la tarea en trozos pequeños trabajándolos por separado.

En resumen: bifurcación, salto implícito y llamada a procedimiento.

### **Bifurcación:**

También llamadas de salto tiene como uno de sus operandos la dirección de la siguiente instrucción a ejecutar. Las más frecuentes son los saltos condicionales, es decir que se ejecuta la bifurcación si se cumple una condición.

Hay dos formas comunes de generación de la condición a comprobar. En primer lugar, la mayoría de las máquinas proporcionan un código de condición de uno o varios bits que se actualiza cuando se ejecutan algunas operaciones. Este código puede imaginarse como un registro visible para el usuario (los flags, normalmente usados en operaciones aritmético-lógicas tales como 0, positivas, negativo, desbordamiento).

Otra aproximación sería especificar una instrucción con tres direcciones. Dos para comparar y otra para especificar la bifurcación dentro de la misma instrucción.

### **Instrucciones de salto implícito**

Esta instrucción incluye una dirección de manera implícita. Normalmente, el salto implícito implica que se va a saltar una instrucción. Por lo tanto la dirección implícita es igual a la dirección siguiente más la longitud de una instrucción. Dado que este tipo de instrucciones no requiere una dirección destino, deja espacio libre para otras cosas, por ejemplo incrementar y saltar si es cero.

### **Instrucciones de llamada a procedimiento:**

En cualquier momento un programa debe poder invocar o llamar a un procedimiento, es decir que se ordena al computador que pase a ejecutar un módulo completo separado y que retorne después al punto en que tuvo lugar la llamada.

Esto permite la reutilización de código y la modularidad facilita la tarea de programar.

El uso de procedimientos requiere de dos instrucciones básicas. Una instrucción de llamada que produce una bifurcación de la posición actual al procedimiento y una instrucción de retorno al lugar desde el que se la llamo. Ya que debe permitirse que el procedimiento se llame desde distintos puntos la CPU debe preservar la dirección de retorno en algún sitio los lugares habituales para almacenar la dirección de retorno son:

- Un registro
- Al principio del procedimiento
- En la cabecera de la pila

Además de la dirección de retorno, a menudo es necesario pasar o transferir parámetros en la llamada a un procedimiento. Estos se pueden transferir mediante registros o en la pila.

## **Direccionamiento:**

- Inmediato
- Directo
- Indirecto
- Registro
- Indirecto con registro
- Con desplazamiento
- Pila

**Direccionamiento inmediato:** es la forma más sencilla de direccionar. Es en el que el operando esta en realidad presente en la propia instrucción. Este modo puede utilizarse para definir y utilizar constantes o para fijar valores iniciales a variables. La ventaja de este modo, es que no se requiere una referencia a memoria para obtener el operando. La desventaja es que el tamaño del número está restringido a la longitud del campo de direcciones que en la mayoría de los repertorios de instrucciones es pequeño comparado con la longitud de palabra.

**Direccionamiento directo:** es una forma sencilla de direccionar, en el que el campo de direcciones contiene la dirección efectiva del operando. Solo requiere una referencia a memoria y no necesita ningún cálculo especial. La limitación obvia es que proporciona un espacio de direcciones restringida.

**Direccionamiento indirecto:** el problema de la longitud del campo de direcciones se soluciona haciendo referencia a una dirección de una palabra de memoria que contenga la dirección completa del operando. La ventaja de esta aproximación es que para una longitud de  $N$  bits se dispone ahora de un espacio de direcciones de  $2^N$ . La

desventaja es que la ejecución de la instrucción requiere dos referencias a memoria para captar el operando una para captar su dirección y otra para obtener su valor.

Una variante poco utilizada es el direccionamiento multinivel o en cascada, donde una dirección apunta a otra y así sucesivamente hasta el operando.

**Direccionamiento de registros:** es similar al directo. La diferencia es que el campo de direcciones referencia un registro en lugar de una dirección de memoria principal. Normalmente un campo de direcciones que referencia a registros consta de 3 o 4 bits de manera que pueden referenciarse un total de 8 a 16 registros de uso general.

Las ventajas son que solo es necesario un campo pequeño de direcciones en la instrucción y que no se requieren referencias a memoria, haciendo que el tiempo de acceso sea relativamente pequeño. La desventaja es que el espacio de direcciones está muy limitado.

Si se usa de forma masiva, de modo que se transfieran los operandos de memoria a registros, se opera con él una vez y se devuelve a memoria, resulta en un paso intermedio innecesario. En cambio, si el operando se usa durante varias operaciones, se estaría logrando un ahorro de real.

**Direccionamiento indirecto con registro:** este método es análogo al indirecto, de modo que se almacena en un registro la dirección del operando. Las ventajas y limitaciones de este método son básicamente las mismas que en el direccionamiento indirecto. Además, este direccionamiento emplea una referencia menos a memoria que el direccionamiento indirecto.

**Direccionamiento con desplazamiento:** este modo combina las posibilidades de direccionamiento directo e indirecto con registro. Requiere que las instrucciones tengan dos campos de direcciones, al menos uno de ellos explícito el valor contenido en uno de los campos se utiliza directamente, el otro (un campo de direcciones o una dirección implícita) se refiere a un registro cuyo contenido se suma al primero para generar la dirección efectiva.

**Direccionamiento de pila:** los elementos se añaden en la cabecera de la pila. La pila tiene asociado un puntero cuyo valor es la dirección de la cabecera o tope de la pila. El puntero de pila se mantiene en un registro, así las referencias a posiciones de la pila en memoria son direcciones de acceso indirecto con registro.

## **Lenguaje de máquina y assembly**