Unidad 2: Aritmética de las computadoras

Definición de bit, nibble, byte, palabra, palabra doble, relación con lenguajes de alto nivel. Representaciones numéricas: números enteros con y sin signo. Aritmética con enteros. Fundamentos de la representación en punto flotante, normalización, error de la representación. Representación estándar del IEEE. Aritmética en punto flotante. Representaciones alfanuméricas, ASCII, EBCDIC.

Rango: diferencia entre el número mayor y el menor

Resolución: diferencia entre dos números consecutivos

Teorema fundamental de la numeracion

Este teorema establece la forma general de construir números en un sistema de numeración posicional.

Representación en signo-magnitud

El bit mas significativo de la palabra se toma como bit de signo. Si dicho bit es 0 el numero es positivo. Si el bit es 1, el numero es negativo.

Esta representación tiene varias limitaciones:

Tanto en la suma como en la resta debe tenerse en cuenta el signo y la magnitud relativa de cada numero. Otra limitación es que hay dos representaciones para el 0, -0 y +0.

Complemento a 1

El bit más significativo representa el signo de N (mismo convenio que signo y magnitud). Si el número es positivo se representa en binario natural y si es negativo con el complemento a 1 de su magnitud.

El Ca1 de un número en base 2 se obtiene invirtiendo todos los bits.

$+32_{10}$= 00100000 -32_{10}=11011111

$+7_{10}$= 00000111 -7_{10}= 11111000

$+41_{10}$= 00101001 -41_{10}=11010110

- El intervalo es simétrico
- Los n bits representan al número
- Los positivos empiezan con cero (0)
- Los negativos empiezan con uno (1)
- Hay dos ceros
- Números distintos 2^n

Complemento a dos

Esta representación usa el bit mas significativo como bit de signo. La diferencia esta en la forma de tratar el resto de los bits.

Consideremos un entero de n bits, A representado en complemento a dos. Si A es positivo, el bit de signo a_{n-1} es 0. Los restantes bits representan la magnitud del numero de la misma forma que en BSS.

El numero 0 se identifica como positivo y tiene por tanto, un bit 0 de signo y una magnitud compuesta de todos ceros.

Ahora, para un numero negativo A, el bit de signo a_{n-1} es 1. Los n-1 bits restantes pueden tomar cualquiera de las 2^{n-1} combinaciones. Por tanto, el rango de los enteros negativos que pueden representarse es desde -1 hasta -2^{n-1}.

El Ca2 de un número (en base 2) se obtiene invirtiendo todos los bits (Ca1) y luego sumándole 1.

Otra forma: "mirando" desde la derecha se escribe el número (base 2) igual hasta el primer "1" uno inclusive y luego se invierten los demás dígitos

- Los positivos empiezan con cero (0)
- Los negativos empiezan con uno (1)
- El rango es asimétrico y va desde $-(2^{n-1})$ a $+(2^{n-1}-1)$
- Hay un solo cero

Técnica del Exceso

La representación de un número A es la que corresponde a la SUMA del mismo y un valor constante E (o exceso).

Dado un valor, el número representado se obtiene RESTANDO el valor del exceso

El signo del número A resulta de una resta En binario, NO sigue la regla del bit mas significativo

decimal	BSS	BCS	CA1	CA2	Exceso 2^{n-1}
+7	0111	0111	0111	0111	1111
+6	0110	0110	0110	0110	1110
+5	0101	0101	0101	0101	1101
+4	0100	0100	0100	0100	1100
+3	0011	0011	0011	0011	1011
+2	0010	0010	0010	0010	1010
+1	0001	0001	0001	0001	1001
+0	0000	0000	0000	0000	1000
-0	----	1000	1111	----	0111
-1	----	1001	1110	1111	0110
-2	----	1010	1101	1110	0101
-3	----	1011	1100	1101	0100
-4	----	1100	1011	1100	0011
-5	----	1101	1010	1011	0010
-6	----	1110	1001	1010	0001
-7	----	1111	1000	1001	0000
-8	----	----	----	1000	----

Punto flotante

Se representa los números con una palabra binaria de dos campos: mantisa (M) y exponente (E).

M y E están representados en alguno de los sistemas en punto fijo que ya conocíamos como BSS, BCS, Ca2, Ca1, Exceso.

- El rango en punto flotante es mayor
- La cantidad de combinaciones binarias distintas es la misma que en otros sistemas 2^8 =256
- En punto flotante la resolución no es constante a lo largo del intervalo

|S|Exponente|Mantisa|

Existen distintos valores de mantisa y exponente para representar un mismo número. Con el objetivo de tener un único par de valores de mantisa y exponente para un número, se introduce la normalización.

Con el objetivo anterior, las mantisas fraccionarias se definen de la forma:

0,1dddddd.....ddd

• donde d es un dígito binario que vale 0 ó 1.

Todas las mantisas empiezan con 0,1

Bit implícito

Como todos los números comienzan con 0,1 no es necesario almacenar ese 1

Si no lo almaceno, puedo “adicionar” un bit más a la mantisa. El bit no almacenado se conoce como bit implícito.

Resolución: es la diferencia entre dos representaciones sucesivas, y varía a lo largo del rango, no es constante como en el caso de punto fijo

Error Absoluto: es la diferencia entre el valor representado y el valor a representar

Estándar IEEE 754

Mantisa: fraccionaria normalizada, con la coma después del primer bit que es siempre uno (1,) en M y S.

Exponente: representado en exceso $2^{n-1} - 1$

	Simple precisión	Doble precisión
Bit de signo	1	1
Bits de exponente	8	11
Bits de fracción	23	52
Total de bits	32	64
Exponente en exceso	127	1023
Rango de exponente	-126 a +127	-1022 a +1023
Rango de numeros	2^{-126} a ~2^{+128}	2^{-1022} a ~2^{+1024}

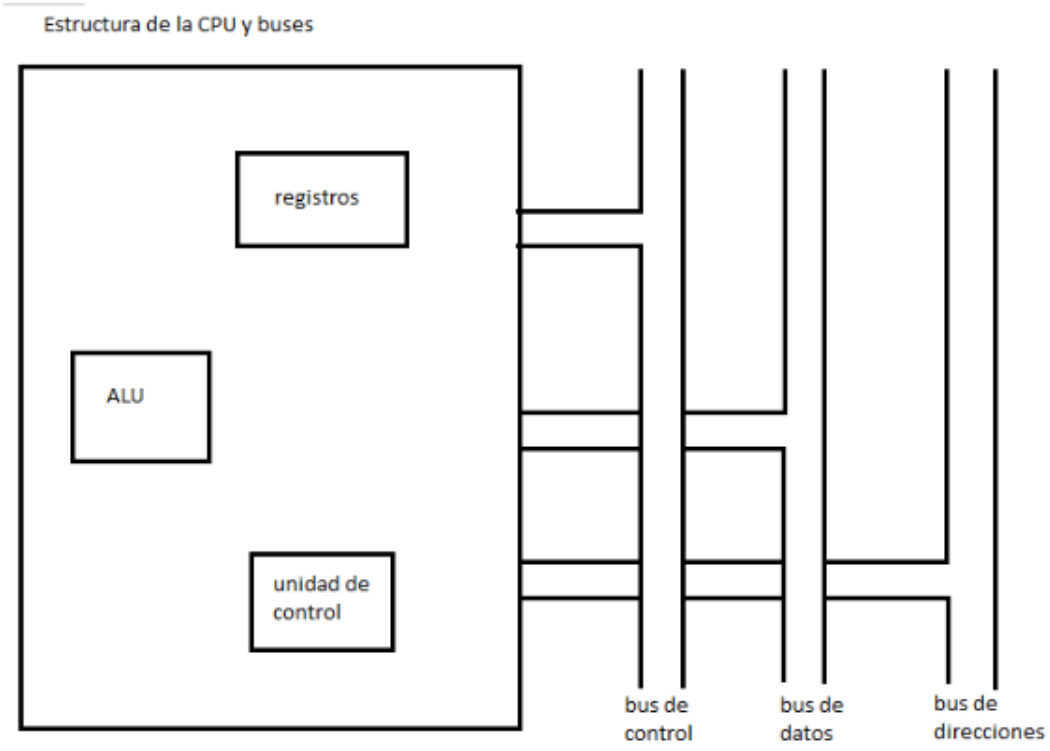
Casos especiales:

- E = 255/2047, M ≠ 0 ⇒ NaN -Not a Number-

    - Exponente máximo, mantisa distinta de 0.
- E = 255/2047, M = 0 ⇒ Infinito
    - Exponente máximo, mantisa igual a 0. El signo implica si es mas o menos infinito
- E = 0, M = 0 ⇒ Cero
    - Mantisa y exponente igual a cero
- E = 0, M ≠ 0 ⇒ Denormalizado
    - Exponente cero, mantisa distinta de 0.
    - ± 0,mantisa_s-p 2^{-126}
    - ± 0,mantisa_d-p 2^{-1022}

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



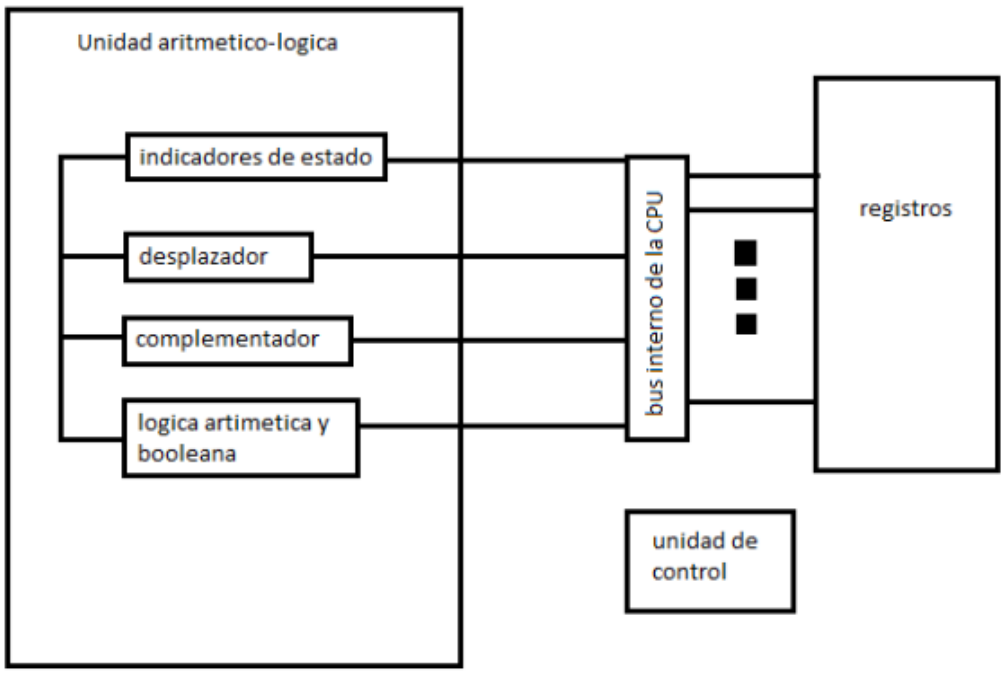
Es el encargado de ejecutar los programas, desde el sistema operativo hasta las aplicaciones de usuario; sólo ejecuta instrucciones programadas en lenguaje de bajo nivel, realizando operaciones aritméticas y lógicas simples, tales como sumar, restar, multiplicar, dividir, las lógicas binarias y accesos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

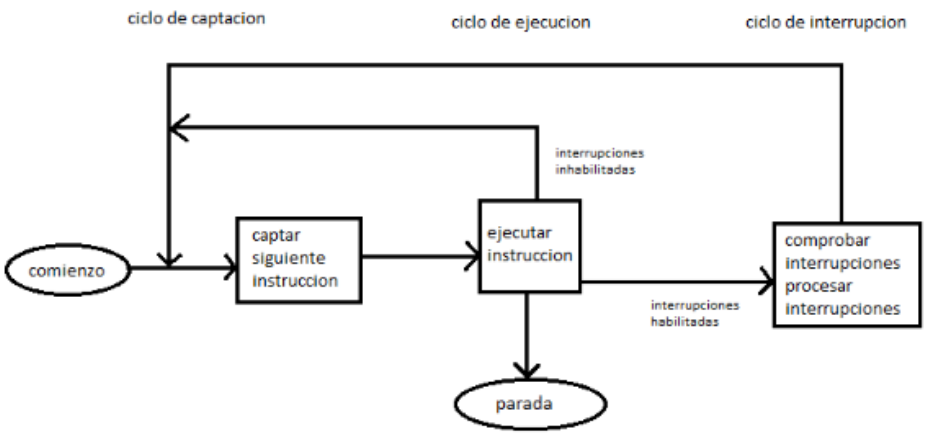
A mayor cantidad de dispositivos, mayor retardo de propagación.

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 maquina. Si solo existe un registro, la referencia a él puede ser implícita. Si existe más de uno, cada registro tendrá asignado un numero único y la instrucción debe contener el numero del registro deseado
- Dispositivos de e/s: la instrucción debe especificar el modulo 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 considerarlas números. Estas operaciones se realizar 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, incuso 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 capitulo 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 capitulo 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 maquinas 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 artimetico-logicas tales como 0, positivas, negativo, desbordamiento).

Otra aproximación seria 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 modulo 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 iníciales 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 numero está restringido a la longitud del campo de direcciones que en la mayoría de las 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 capos de direcciones, al menos uno de ellos explicito 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 pinero 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

Tipos de memorias, clasificación. Parámetros característicos, tamaño, tiempo de acceso, costo, otros. Memoria principal, formas de organización. Memoria secundaria, organización y formato de datos. Organización jerárquica de la memoria. Dispositivos de almacenamiento externo, disco, cinta, disco óptico, otros. Múltiples unidades de discos (RAID).

Jerarquía de memorias:

Mientras mas se escale en la pirámide, la memoria es mas pequeña, mas costosa por bit y mas rápida.

Para solventar la diferencia de velocidades entre una memoria y otra, se aplican distintos niveles de cache que funcionan de intermediarios, almacenando copias de porciones del programa.

Esto funciona gracias al principio de localidad de memoria.

Memoria, generalidades

Algunos conceptos relacionados con memorias internas:

- Palabra: es la unidad de organización de la memoria. El tamaño de la palabra suele coincidir con el numero de bits utilizados para representar números y con la longitud de las instrucciones.
- Unidades direccionables: en muchos sistemas la unidad direccionable es la palabra.
- Unidades de transferencia: para la memoria principal, es el numero de bits que se leen o escriben en memoria a la vez. Para la memoria externa, los datos se transfieren normalmente en unidades mas grandes que la palabra denominadas “bloques”.

Otro concepto distintivo de las memorias es el método de acceso:

- Acceso secuencial: la memoria se organiza en unidades de datos llamadas registros. El acceso debe realizarse con una secuencia lineal especifica, es decir que se lee una línea, y luego la línea que esta inmediatamente contigua.
- Acceso directo: se lleva a cabo mediante un acceso directo a una vecindad dada, seguido de una búsqueda secuencial.
- Acceso aleatorio (random): el tiempo de acceso es constante y cualquier celda de memoria puede referenciarse y ser direccionada aleatoriamente. La memoria principal y algunos sistemas de cache son de acceso aleatorio.
- Asociativa: es una memoria de acceso aleatoria que permite hacer una comparación de ciertas posiciones de bits dentro de una palabra buscando que coincidan con unos valores dados y hacer esto para todas las palabras simultáneamente. Por lo tanto, una palabra es recuperada basándose en una porción de su contenido en lugar de su dirección.

Los dos puntos fundamentales de una memoria son su capacidad y sus prestaciones. Estas ultimas se miden por los siguientes parámetros

Tiempo de acceso: en memorias de acceso random es el tiempo que tarde en realizarse una operación de escritura o de lectura. En otros tipos de memorias es el tiempo que se tarda en situar el mecanismo de lectura/escritura en la posición deseada.

Tiempo de ciclo de memoria: concepto asociado a las memorias random. Consiste en el tiempo de acceso mas algún tiempo extra que se requiere antes de realizar otra operación. Este tiempo puede ser necesario para que finalicen las transiciones en las líneas de señal o para regenerar los datos en el caso de lecturas destructivas.

Velocidad de transferencia: es la velocidad a la que se pueden transferir datos hacia o desde una unidad de memoria. Para las memorias random coincide con el inverso del tiempo de ciclo.

Tipos de memorias:

Memorias semiconductoras de acceso aleatorio (memorias RAM)

La característica principal es que es posible tanto leer datos como escribir rápidamente datos nuevos. Tanto la lectura como la escritura se ejecutan mediante señales eléctricas.

La otra características distintiva es que es volátil. Esto implica que debe estar continuamente alimentada. Si se interrumpe la alimentación se pierden los datos, de modo que solo puede usarse como almacenamiento temporal.

Las tecnologías RAM se dividen en estaticas y dinamicas.

Las dinamicas están hechas con celdas que almacenan los datos como cargas en condensadores. La presencia o ausencia de carga se interpreta como el 1 o el 0 binarios. Las RAM dinamicas requieren refrescos periódicos para mantener memorizados los datos. Los valores binarios se almacenan utilizando configuraciones de puertas que forman biestables. Estas memorias son mas densas(celdas mas pequeñas=mas celdas por unidad de superficie) y mas económicas que las estaticas, aunque requieren circuitería para el refresco. Suelen usarse para memorias de mayor tamaño.

Una RAM estatica mantendrá sus datos mientras tenga alimentación y son mas rapidas que las dinamicas.

Ambos tipos de RAM son volátiles.

En contraste con las memorias RAM, existen las memorias ROM. Una memoria de este tipo contiene un patron permanente de datos que no puede alterarse, por lo tanto no pueden escribirse datos nuevos. Se utilizan en la microprogramación, subrutinas de biblioteca para funciones de uso frecuente, programas del sistema y tablas de funciones.

Organización de las memorias:

El elemento básico de una memoria semiconductora es la celda de memoria. Todas las celdas de memoria comparten ciertas propiedades:

- Presentan dos estados estables o semi estables que pueden emplearse para representar el 1 o el 0 binario.
- Puede escribirse en ellas al menos una vez para fijar su contenido.
- Pueden leerse para detectar su estado.

Básicamente una celda de memoria debe tener tres terminales para transportar señales eléctricas. La primera, la terminal de selección, selecciona la celda para la operación a realizar. La terminal de control, indica el tipo de operación (si se trata de una lectura o escritura). La tercera terminal, se utiliza como salida del estado de la celda en la lectura o proporciona la señal a fijar en la celda en el caso de una escritura.

Lógica del chip

La memoria semiconductora viene encapsulada en chips. Cada chip contiene una matriz de celdas de memoria.

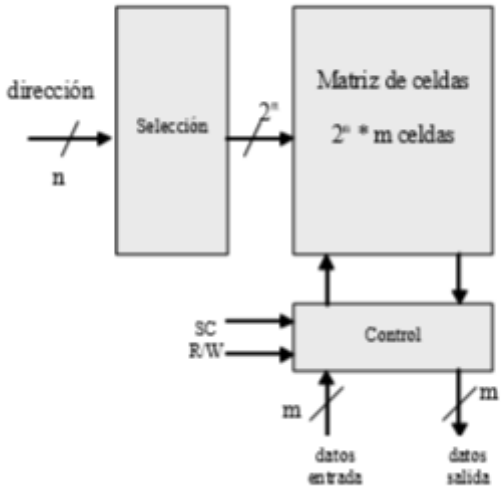
Si la memoria contiene 1 bit por palabra, se necesitaran claramente al menos un numero de chips igual al numero de bits por palabra.

Organización 2d

Es la forma mas sencilla de organización de memoria.

Las celdas forman una matriz de 2^n filas y M columnas, siendo 2^n el numero de palabras del chip y m la cantidad de bits de cada palabra. Cada fila es seleccionada por la decodificación de una configuración diferente de los N bits de dirección. Esta organización tiene el inconveniente de que el selector (decodificador) crece exponencialmente con el tamaño de la memoria.

Igual ocurre al numero de entradas de las compuertas que generan las salidas.



De esta forma, 2^n es el tamaño de bits de direcciones que el decodificador puede re direccionar. M es la cantidad de bits que tiene cada palabra, por tanto el tamaño total de la memoria en bits seria 2^n*m.

Memoria 2d/2

En lugar de una sola selección de 2n salidas, en esta organización se utilizan dos decodificadores de $2^{n/2}$, operando en coincidencia. Las líneas de direcciones se reparten entre los decodificadores. Para una configuración dada de las líneas de dirección se selecciona un único bit de la matriz. Por ello también se la denomina organización por bits.

En esta organización se necesitan múltiples matrices, tantas como bits deba tener la palabra. De este modo, para armar una palabra, se selecciona un bit de cada matriz en la misma exacta posición de cada matriz, actuando de forma paralela.

Memoria cache

El objetivo de la memoria cache es lograr que la velocidad de la memoria sea lo mas rápida posible. La cache contiene una copia de partes de la memoria principal. Cuando el procesador inteta leer una palabra de memoria se hace una comprobación para determinar si la palabra esta en la cache. Es es asi, se entrega dicha palabra al procesador. Sino, un bloque de memoria principal consistente en un cierto numero de palabras se trasfiere a la cache y después la palabra es entregada al procesador.

Esto es posible debido al principio de localidad de memoria. Este dice que al ejecutar una instrucción, la siguiente instrucción a ejecutar del proceso es posiblemente, la instrucción contigua en memoria.

Discos magneticos

Organización y formato de los datos

Un disco magnetico es un plato circular construido con metal o plástico cubierto de un material magnetizable. Los datos se graban en el y se recuperan del disco a travez de una bobina llamada cabeza. Durante una operación de lectura o escritura la cabeza permanece quieta mientras el plato rota bajo ella. El mecanismo de escritura se basa en el campo magnetico producido por el flujo eléctrico que atraiesa la bobina.

La cabeza es un dispositivo realmente pequeño, capaz de leer o escribir una zona del plato que rota bajo ella. Esto da lugar a que los datos se organicen en un conjunto de anillos concéntricos en el plado llamados pistas. Cada pista tiene el mismo ancho que la cabeza.

Los datos se transfieren desde y hacia el disco en bloques. Normalmente el bloque es menor que la capacidad de una pista. De acuerdo con esto los datos se almacenan en regiones del tamaño de un bloque conocidas como sectores. Estos pueden ser se longitud fija o variable.

En la mayoría de los discos la cubierta magnetizable se aplica a ambas caras del plato denominándose discos de doble superficie. Algunas unidades de disco poseen arios platos apilados verticalmente y separados por cierta distancia. Estos discos disponen de multiples brazos y se le conoce como cilindro.

En estos discos las prestaciones se evalúan por los siquientes criterios:

Tiempo de búsqueda: el tiempo que tarda en desplazar el brazo del disco hasta la pista requerida.

Retardo rotacional: es el tiempo que tarda en girar el disco.

Tiempo de transferencia: el tiempo que tarda en escribir o leer un dato en el disco.

Discos RAID

1. Es un conjunto e unidades físicas de disco vistas por el sistema operativo como una única unidad lógica.
2. Los datos se distribuyen a través de las unidades físicas del conjunto de unidades.
3. La capacidad de los discos redundantes se usa para almacenar información de paridad que garantice la recuperación de los datos en caso de fallo de disco.

Los detalles de la segunda y tercera característica cambian según los distintos nieles de RAID. El nivel 0 no soporta la tercera característica.

Esta organización de discos reemplaza luna unidad de disco de gran capacidad por unidades múltiples de menor capacidad y distribuye los datos de forma que se puedan habilitar accesos simultáneos a los datos de varias unidades, mejorando por tanto, las prestaciones de E/S y permitiendo más fácilmente aumentos e la capacidad. Esta estrategia hace hincapié en la necesidad de redundancia. El uso de varios dispositivos, además de permitir que arias cabezas operen simultáneamente consiguiendo mayores velocidades de E/S y de transferencia, incrementa la probabilidad de fallos. Para comenzar esta disminución de seguridad, RAID utiliza la información de paridad almacenada que permite la recuperación de datos perdidos debido a un fallo de disco

Categoría	Nivel	Descripción	Grado de E/S solicitado Lectura/escritura	Grado de transferencia de datos Lectura/escritura	Aplicación típica
Estructura en tiras	0	No redundante	Tiras largas: excelente	Pequeñas tiras: excelente	Aplicaciones que reuqieren altas prestaciones con datos no críticos
Estructura en espejo	1	Espejo	Bueno/regular	Regular/regular	Controladores de sistemas; ficheros críticos
Acceso paralelo	2	Redundante con código haming	Pobre	Excelente	
	3	Bit de paridad intercalado	Pobre	Excelente	Aplicaciones con muchas E/S tales como imágenes cad
Acceso independiente	4	Boque de paridad intercalado	Excelente/regular	Excelente/pobre	
	5	Paridad distribuida en bloques intercalados	Excelente/regular	Excelente/pobre	Grado de petición alto, lectura intensiva, consulta de datos
	6	Paridad distribuida dual en bloques intercambiados	Excelente/regular	Excelente/pobre	Aplicaciones que requieren alta disponibilidad