**//////////////////////////////////////////////////Clases 1-6//////////////////////////////////////////////////**

Teorema fundamental de la numeración:

Se utiliza para calcular un número mediante la suma de cada dígito multiplicado por la potencia de la base correspondiente (por ejemplo la base en binario es 2, en decimal 10 y en hexadecimal 16) a la posición que ocupa el número.

Error absoluto en punto flotante y fijo: En punto fijo, el error absoluto es siempre el mismo; en cambio en punto flotante no es constante, puede variar.

Casos especiales IEEE754:

￼

Flip flops: hay 4 tipos de flip flops (sr, j-k, d, t) son biestables ya que son circuitos que solo poseen dos estados posibles de funcionamiento. Se pueden clasificar en dos tipos; asincrónicos (cuando en la entrada se establece una combinación, las salidas van a cambiar), sincrónicos (hay una entrada “especial” denominada clock que determina cuando cambian las salidas acorde a las entradas.

Computadora ¿qué es?

• Máquina

• Digital

• Sincrónica

• Cálculo numérico

• Cálculo lógico

• Controlada por programa

• Comunicación con el mundo exterior

Arquitectura: son aquellos atributos visibles al programador

-Conjunto de instrucciones, número de bits usados para representación de datos, mecanismos de E/S, técnicas de direccionamiento.

Organización: es cómo son implementados

-Señales de control, interfaces, tecnología de memoria.

Estructura: es el modo en el cual los componentes se relacionan entre sí.

Función: es la operación de los componentes individuales como parte de la estructura.

Funciones de todas las computadoras:

-Procesamiento de datos

-Almacenamiento de datos

-Movimiento de datos

-Control

Composición de la computadora: CPU, memoria principal, sistema de interconexión, entrada/salida.

Composición de la CPU: registros, Alu, interconexión interna de CPU, unidad de control.

Mar= registro de dirección de memoria

Mbr= registro de dato de memoria

Pc= registro contador de programa (contiene la posición de memoria que se debe captar)

IR= es un registro de la unidad de control de la CPU, donde se almacena la instrucción que se está ejecutando; la IR también cumple la función de decodificar la instrucción

Unidad de control: controla todos los eventos, genera las señales, controla los eventos a nivel de instrucción. Va a ser capaz de decodificar instrucciones. Va a ser la encargada de secuenciar los eventos.

Interacción de la CPU con los registros: la CPU interactúa con la memoria a través de registros que están “ocultos” para el programador. Estos son; la mar y el mbr, ambos dos pueden estar conectados a los buses.

Composición de la unidad de control: descodificadores y registros, lógica de secuenciamiento, memoria de control

Modelo Von Neumann: consta de 5 componentes principales;

-unidad de entrada (provee las instrucciones y los datos)

-unidad de memoria (donde se almacenan los datos e instrucciones)

-unidad aritmético-lógica (procesa los datos)

-unidad de control (dirige la operación)

-unidad de salida (se envían los resultados)

Interconexión de un sistema de cómputo: un sistema de cómputo está constituido por 3 subsistemas;

-CPU

-memoria

-E/S

Bus: es un camino que conecta dos o más dispositivos. Los buses tienen un número de canales que son la cantidad de bits que pueden transportar. Hay 3 tipos de buses (bus de control, bus de datos y bus de direcciones)

Bus de datos:

-transporta datos (no hay diferencia entre dato e instrucción en este nivel)

Bus de direcciones:

-identifica el origen o destino de los datos

-el ancho del bus determina la máxima capacidad de memoria del sistema

Bus de control:

-controla y ordena toda la interconexión que hay entre las distintos componentes de la computadora. A su vez también controla el envío y la recepción de información.

¿Qué es un programa?

-es una secuencia de pasos

-se hace una operación aritmético/lógica por cada paso

-se necesitan diferentes señales de control para cada operación.

Ciclo de instrucción básico:

-tiene dos pasos; el ciclo de búsqueda (busca la siguiente instrucción) y el ciclo de ejecución (ejecuta la instrucción). Esto entra en un loop hasta que se interrumpe el ciclo.

-le ejecución del problema se interrumpe sólo si la máquina se apaga, hay un error o una instrucción que interrumpa la computadora.

Diagrama de estados:

1-determina la dirección de la siguiente instrucción a ejecutarse

2-lee la instrucción de su posición de memoria a la CPU

3-analiza la instrucción para determinar el tipo de operación a realizar y los operándoos que se usarán

4-determina si la operación implica la referencia a un operando en la memoria de e/s, entonces se determina la dirección

5-busca el operando en la memoria de e/s

6-ejecuta la instrucción

7-calculo dirección resultado

8-almacenamiento resultado

**//////////////////////////////////////////////////Clase 7//////////////////////////////////////////////////**

Alto nivel- se trabaja con variables

Bajo nivel- se trabaja con registros y memoria

Las instrucciones se representan en mnemónicos, los mnemónicos son códigos de operación representados por medio de abreviaturas. Se utiliza esta representación para facilitar el entendimiento de las instrucciones, ya que las representaciones binarias son mucho más complejas de tratar.

Elementos de instrucción de una máquina:

Código de operación (especifica la operación a realizar; es un código binario)

Referencia del operando fuente (establece donde se encuentra el operando; la operación puede involucrar 1 o más operando fuente)

Referencia del operando resultado (establece donde se almacena el resultado)

Referencia de la siguiente instrucción (le dice a la CPU donde buscar la siguiente instrucción después de la ejecución de la anterior)

Los operando fuente y resultado pueden estar en tres lugares:

Memoria

Registro de la CPU

Dispositivo E/S

Tipos de instrucciones:

En lenguajes de alto nivel, una instrucción tan sencilla como x:= x+y; puede implicar cargar registros, sumarlos y luego almacenar el resultado en memoria; por lo tanto instrucciones más complejas pueden requerir varias instrucciones de máquina.

Una diferencia entre el lenguaje de alto nivel y el de bajo, es que en el de alto nivel las operaciones se expresan de forma “concisa”, usando variables; en cambio, en bajo nivel, la máquina expresa las operaciones de forma “básica” involucrando el movimiento de datos y uso de registros.

Cualquier lenguaje de alto nivel debe ser convertido a un lenguaje de máquina (bajo nivel) para ser ejecutado. El conjunto de instrucciones de máquina debe ser capaz de expresar cualquiera de las instrucciones de un lenguaje de alto nivel.

Las instrucciones de máquina se pueden categorizar en:

Procesamiento de datos (operaciones aritméticas y lógicas)

Almacenamiento de datos (transferencias dentro del sistema)

Instrucciones de E/S (transferencias de datos entre la computadora y los mecanismos externos)

Control

Máquina para 4 direcciones:

Direcciones explicitas para operandos, resultado y próxima instrucción

Cada campo de dirección tiene que tener bits para “acomodar” una dirección completa

Máquina para 3 direcciones:

La dirección de la próxima instrucción está almacenada en un registro llamado PC

Referencias= 72 bits

Máquina para 2 direcciones:

Reduce el tamaño de la instrucción (48 bits de referencias)

Hay que mover el operando1 a un registro temporal

Menos elección donde guardar el resultado

Máquina para 1 dirección:

Registros especiales (acumulador)

Instrucciones para cargar y descargar el acumulador

Un operando y resultado en un lugar predefinido

Instrucción más corta (24 bits de referencia)

Modos de direccionamiento:

Inmediato:

El operando se obtiene automáticamente de la memoria al mismo tiempo que la instrucción

No requiere una referencia extra a memoria de datos

Se utiliza para definir constantes y para inicializar variables

Desventaja: tamaño del operando limitado por el tamaño del campo de direccionamiento.

Directo:

El campo de dirección tiene la dirección efectiva del operando

Es simple, pero tiene un espacio limitado en direcciones por cantidad de bits del campo

Uso: acceder a variables globales cuya dirección se conoce en el momento de compilación

Por registro:

Conceptualmente igual al directo, pero se especifica un registro en lugar de una posición de memoria

La referencia a registro usa menos bits que la especificación de la dirección y no requiere acceso a memoria de datos

Desventaja: no hay muchos registros

Indirecto por memoria:

En la instrucción está la dirección de la dirección del operando. Trata de solucionar el problema del directo. Así, con una dirección de menos bits en la instrucción, se apunta a una dirección de más bits

Ventaja: espacio de direccionamiento mayor

Desventaja: múltiples accesos a memoria (más tiempo)

Indirecto por registro:

En la instrucción se especifica el registro que tiene almacenada la dirección

Ventaja: menos bits para especificar el registro que la posición de memoria. Espacio de direccionamiento grande, accede una vez a menos memoria que el indirecto. La dirección usada se llama puntero.

Por desplazamiento:

Combina capacidades del indirecto y directo. Requiere que la instrucción tenga dos campos de dirección. Estos dos campos se suman para producir la dirección efectiva. Los más comunes son:

Relativo:

El registro referenciado de manera implícita es el PC

La dirección de la instrucción actual se suma al campo de dirección para producir la dirección efectiva (el campo de la dirección se representa en CA2)

De registro base:

El registro referenciado contiene una dirección de memoria y el campo de dirección tiene un desplazamiento

Indexado:

Se direcciona la memoria con un registro más un desplazamiento (se intercambian los papeles del registro y desplazamiento en relación al de registro base)

La indexación proporciona un mecanismo eficiente para realizar operaciones iterativas

Se utiliza un índice (registro)

Stack/pila:

Es un arreglo lineal de localidades de una memoria. Es una lista o cola donde el uniformó en entrar es el primero en salir (LIFO=last in, last out). Es una zona de memoria reservada.

Asociado con la pila/stack hay un registro puntero, cuyo valor es la dirección tope de la pila

**//////////////////////////////////////////////////Clase 8//////////////////////////////////////////////////**

Registros visibles al usuario (son utilizados por el programador):

Propósito general, datos, dirección, códigos de condición, flags.

Funciones:

Cualquier registro de propósito general puede contener el operando para cualquier código de operación. (Pueden existir restricciones)

Se pueden poner utilizar para direccionamiento

Solo para datos o sólo para direcciones

Los registros de dirección pueden ser asignados para un mdd.

Registros de control y estado: son utilizados por la unidad de control para controlar la operación de la CPU (no son visibles al programador)

Empleados para controlar la operación de la CPU

4 registros esenciales: (pc, ir, mar, mbr)

El número de registros afecta al tamaño de la instrucción. Mayornro. de registros, mas bits para especificarlos en la instrucción.

Pocos registros= más referencias a memoria. Nro. óptimo entre 8 y 32.

Longitud de los registros

De direcciones: deben ser capaces de almacenar la dirección más grande

De datos: deben estar habilitados para almacenar la mayoría de los tipos de datos

Algunas máquinas permiten 2 registros contiguos utilizados como un solo registro para almacenar valores de doble longitud

Generalmente los registros se usan de la siguiente manera:

Ax= acumulador

Bx= puntero base

Cx= contador

Dx=datos

**//////////////////////////////////////////////////Clase 9//////////////////////////////////////////////////**

La velocidad del procesador se duplica cada 18 meses

La memoria se cuadriplica su tamaño cada 3 años. Su velocidad aumenta a razón de un 10% anual.

LaCPU es cada vez más rápida mientras la velocidad de la memoria crece más lentamente. Mientras va pasando el tiempo más grande se hace la brecha.

Jerarquía de memorias

Cúspide: registros

Segundo: caché

Tercero: memoria principal

Cuarto: discos

Base: ópticos (CD, DVD)

Cuanto más arriba está en la escala, más caro será su costo por bit, pero tendremos mayor velocidad y más cantidad de accesos.

Se emplean distintas tecnologías para acceder a los distintos tipos de memorias. Con el objetivo de aumentar la capacidad de almacenamiento y reducir el tiempo de acceso.

CARACTERÍSTICAS DE LA MEMORIA

Duración de la información:

-memorias volátiles: pueden ser leídas o escritas; la información se pierde luego de cortar la alimentación del sistema. (RAM)

-memorias no volátiles: no se necesita de alimentación eléctrica para mantener la información, se necesita solo para poder escribir y leer. (DISCOS, CINTAS)

-memorias permanentes: no pueden ser modificadas en forma normal. Memorias de lectura solamente. (ROM, EPROM)

Modos de acceso:

Por palabra: memoria principal

Por bloqué: discos, caché

Velocidad (memorias semiconductoras):

Tiempo de acceso: tiempo máximo que transcurre entre que se inicia la operación y se termina.

Tiempo de ciclo: tiempo mínimo entre dos operaciones sucesivas en una memoria

Velocidad (memorias magnéticas):

Tiempo de acceso: tiempo de posicionar el cabezal + tiempo de latencia

Velocidad de transferencia: bytes/seg

Métodos de acceso:

Aleatorio: tiempo independiente de los accesos que hice antes y es constante. Siempre va a tardar lo mismo

Secuencial: hay que hacer una secuencia específica para encontrar la información buscada. Nos obliga a hacer una serie de pasos en un orden determinado. No sabemos en qué parte específica está la información buscada.

Directo: es de tiempo variable. Los bloques de información tienen una dirección que se basa en una ubicación física. Es directo porque conocemos en qué pista del disco está la información, no hay necesidad de recorrer todo el disco.

Asociativo: no se sabe dónde está la información buscada, sino que conocemos parte de la información buscada. Se hace una comparación hasta encontrar la información faltante.

MEMORIAS DE ACCESO ALEATORIO:

Estáticas (SRAM):

-basadas en flip flops

-es más rápida (se usa como caché)

Dinámicas (Dram):

-basadas en transistores

-necesita hacer refresh cada cierto tiempo

-los capacitores se descargan

-almacenan más información en la misma superficie ya que los capacitores ocupan menos lugar que los flip flops

PROPIEDADES DE LAS CELDAS DE MEMORIA SEMICONDUCTORAS:

-dos estados estables para representar al 1 y al 0

-se puede escribir en ellas al menos 1 vez

-se pueden leer para conocer el estado

ORGANIZACIÓN DEL CHIP:

2D:

-a cada línea de memoria/palabra se la conecta con un decodificador. El decodificador selecciona toda la palabra

-las líneas verticales se conectan a cada bit de la salida

-el decodificador que está en el chip tiene 2^W salidas para W entradas

2D 1/2:

-los bits de una misma palabra están dispersos en distintos chips

-una parte va a permitir seleccionar el renglón y la otra la columna

-hay 2 decodificadores

Comparación entre 2D y 2D 1/2:

En 2D todos los bits están en el mismo chip

En 2D 1/2 los bits de una palabra están en distintos chips

2D es muy larga y estrecha, Nro. grande de palabras de pocos bits. Cada línea de selección se palabra tiene que tener un manejador y conectarse al decodificador. Ocupan mucha superficie

2D dificulta el uso eficaz de los circuitos correctores de error. En 2D 1/2, al estar los bits dispersos en distintos chips, hay menos probabilidad de error

En 2D 1/2 al usar decodificación separada de filas y columnas, reduce la complejidad de los decodificadores.