

# ARQUITECTURA DE COMPUTADORAS

## Unidad 1

### Organización de una computadora:

Arquitectura de von Neumann: Consta de 4 unidades interconectadas entre sí que se envían información para hacer operaciones. La maquina no diferencia si esta información es un dato o una instrucción.

- Instrucciones: Son las ordenes que se le dan a la unidad de control para que realice la operación.
- Datos: Cantidades con las que se hacen operaciones.

Unidad lógica aritmética (ULA): Realiza operaciones con la información guardada en los registros según lo indique la unidad de control.

- Registros: Guarda los datos con los que se realizan las operaciones y los resultados de las mismas.
- Operador booleano: Es el circuito capaz de realizar varias operaciones con los datos.

Unidad de memoria (UM): Almacena la información de forma ordenada. También busca y entrega la información a la unidad que sea indicada por la unidad de control.

- Almacén: Donde se guarda la información.
- Sistema de escritura: Guarda la información proveniente de la ULA o la UE.
- Sistema de lectura: Envía datos a la UC, la ULA o la US.
- Registro de dirección: Guarda las direcciones de la información, lo que permite individualizar la posición de una porción de información específica.
- Decodificador de direcciones: Interpreta o decodifica las instrucciones en el registro de direcciones.

Unidades de entrada y salida (UE/S): Conecta periféricos a la computadora. Reordena y adecua las velocidades de la maquina y del periférico para que puedan transferir información entre ellas.

Unidad de control (UC): Lee las instrucciones y las ejecuta enviándole señales a las otras unidades para que actúen sobre los datos como indica la instrucción.

- Registro de instrucciones: Almacena la instrucción, la cual está formada por:
  - Mando: Referencia a la acción que se debe realizar.
  - Dirección: Referencia a los datos con los que se ejecutara la instrucción.
- Decodificador de instrucciones: Interpreta las instrucciones en el registro.
- Reloj: Permite el funcionamiento sincrónico de los componentes.

### Formas de funcionamiento de una computadora digital

- Programación en hardware: Circuitos con hardware especializado que se interconectan para resolver problemas específicos.

- Programación en software: Serie de circuitos por el cual pasa la información a ser transformada. Es más flexible y barata que la programación en hardware, pero mas lenta.

Interconexión de las unidades de una computadora: En una computadora, las unidades descriptas anteriormente, se interconectan a través de la barra de ómnibus, la cual consta de un conductor que se conecta a todas las unidades.

Funcionamiento de una computadora: La UC habilita los circuitos que realizan la búsqueda y transferencia de la instrucción de la memoria a el registro de instrucciones. Luego se ejecuta la instrucción. Esto se conoce como ciclo de búsqueda y ejecución y se puede subdividir de la siguiente manera:

- CDI (Cálculo de la dirección de la instrucción): LA dirección no es conocida por el contador de programa, por lo tanto, debe ser calculada.
- BI (Búsqueda de la instrucción): Lee la instrucción en la dirección de memoria calculada y la transporta al registro de instrucciones.
- DI (Decodificación de la instrucción): Se interpreta el código de la instrucción para conseguir los comandos necesarios para llevarla a cabo.
- CDO (Cálculo de la dirección del operando): Calcula la dirección de memoria del operando al que hace referencia la instrucción o si debe ser obtenido en la unidad de entrada.
- BO (Búsqueda del operando): Se transporta el operando de la dirección indicada al registro indicado.
- EJEC (Ejecución): La UC envía las señales de control necesarias para llevar a cabo la instrucción.
- Cálculo de la dirección del resultado (CDR): Se calcula la dirección donde se va a almacenar el resultado obtenido.
- AR (Almacenamiento del resultado): Escribe el resultado en la memoria o lo envía a la unidad de salida.

Interrupciones: Capacidad de otras unidades para interrumpir el proceso que se está llevando a cabo en la Unidad central de procesamiento (Conjunto de la unidad de control y unidad lógica aritmética)

- Por programa: El mismo programa interrumpe su ejecución por overflow, división por cero, ejecución de un mando ilegal o referencias a espacios de memoria externa.
- Por tiempo: Generadas por el reloj de la máquina para realizar operaciones de control periódicas.
- De entrada/Salida: Generadas por la UE/S para indicar su funcionamiento o condiciones de error.
- Fallas de hardware: Fallas en controles de paridad o falla en la alimentación.

#### Interrupciones y ciclo de maquina

Cuando una instrucción requiere un dato de un periférico, debe interrumpirse el proceso de ejecución para que la UC pueda enviar señales de control a la unidad de E/S y esta pueda transferir la información necesaria para la ejecución de la instrucción.

Interrupciones múltiples: Surgen cuando en la ejecución de una interrupción, aparece otra interrupción.

- Interrupciones secuenciales: se ejecutan por orden de aparición, hasta no terminar la ejecución de la primera interrupción, no se ejecuta la próxima.
- Interrupciones anidadas: se ejecutan apenas aparecen y al terminar una, se continua con la otra.

- Prioridades: las interrupciones se ejecutan inmediatamente si su prioridad es mayor al de la interrupción que se esta ejecutando, si no, se ejecutan después.

**Funcionamiento simplificado de una computadora:** La forma en la que la unidad de control envía señales de control, se basa en el hecho de que las unidades están desconectadas por defecto. Cuando la unidad de control “envía una señal de control” lo que hace es conectar las rutas por las que va a pasar la información.

- Transferencia en serie: Transfiere un bit a la vez
- Transferencia en paralelo: Transfiere varios bits de una sola vez.

**Lenguaje de maquina:**

**Ejemplo de lenguaje de maquina**

- Ejecución de un programa

**Sistema operativo:** Intermediario para la comunicación entre el usuario y la máquina.

**Sistema de computación:** Sistema lógico que se emplea para llevar a cabo operaciones aritméticas y conversión de código binario a un código legible por el usuario.

**Codificación**

- Códigos por peso: Cada posición binaria tiene un valor numérico, el cual se convierte a decimal con un polinomio. Este sistema facilita hallar el complemento.
- Códigos alfanuméricos: Se le asigna a cada combinación de bits binarios un símbolo alfanumérico.
- Código BCD: Utiliza los números binarios de 4 bits del 0 al 9 y descarta los números del 10 al 15. Los números de 2 dígitos binarios se representan con 8 bits, los de tres dígitos con 12 bits, etc.
  - Operación entre números codificados en BCD: Cuando se operan números BCD se pueden presentar tres situaciones:
    - Resultado esta entre 0 y 9: El resultado es correcto.
    - Resultado esta entre 10 y 15: Se debe sumar un 6 al resultado.
    - Resultado mayor a 16: Se le suma 6 al resultado.

**Números en coma fija y en coma flotante:** Coma fija es la notación tradicional, mientras que, Coma flotante es notación científica. El numero en coma flotante tiene un bit para el signo del exponente, un bit para el signo de la mantisa (Cifras significativas del número) además de bits que representen los valores de el exponente y la mantisa.

- Operaciones en coma flotante:
  - Suma y resta: Primero se debe igualar los exponentes.
  - Multiplicación y división: No es necesario igualar los exponentes, ya que el exponente del resultado es la suma (o, en la división, resta) de los exponentes de los operandos. Se debe normalizar (Poner el primer dígito significativo como la unidad frente a la coma) los resultados.

Complementos decimal y binaria: El complemento de un número es el módulo del sistema menos el valor del número. Donde el modulo es el valor máximo del sistema. Como no se utilizan sistemas con un número máximo, se complementa cada cifra. En números binarios existe el complemento a 1 (el módulo menos el número) y el complemento a 2 (complemento a 1 más uno).

Suma algebraica: Suma de elementos con signo.

- Suma algebraica con complementos a 10:
  - Cada vez que se complementa se cambia de signo.
  - Se descarta el rebose.
  - Si no hay rebose se debe volver a complementar el resultado.
- Suma algebraica con complementos a 9:
  - El rebose se le suma a la cifra menos significativa.
  - Si no hay rebose se debe volver a complementar.

Suma algebraica entre números binarios:

- Suma algebraica entre números binarios con complemento a 2:
  - Cada vez que se complementa se cambia de signo.
  - Se descarta el rebose.
  - Si no hay rebose se debe volver a complementar el resultado.
- Suma algebraica entre números binarios con complemento a 1
  - El rebose se le suma a la cifra menos significativa.
  - Si no hay rebose se debe volver a complementar.

Multiplicación

- Método normal
- Métodos de las componentes derechas e izquierdas: Se suman los arrastres por separado y luego se suman ambos resultados.
- Multiplicación por sumas repetidas: Se suma el multiplicando tantas veces como indica el multiplicador

Multiplicación binaria:  $1 \times 1 = 1$ ,  $0 \times 0 = 0$ ,  $1 \times 0 = 0$ ,  $0 \times 1 = 0$ .

- Método normal
- Por sumas repetidas: no existe

División

- Método normal
- Por sumas repetidas:
- División decimal
- Método común
- División por restas sucesivas, método restaurativo
- Método no restaurativo
- División binaria

Notación en el complemento a dos

- Producto de números en notación complemento a 2

#### Implementacion de las operaciones logicas y aritmeticas

- Complementadores
  - Complementadores binarios
  - Complementador decimal
- Comparadores
- Sumador binario
- Mecanización de la suma entre números binarios
  - Suma en serie
  - Suma en paralelo
    - Suma en paralelo con anticipo de arrastre
- Multiplicacion de números binarios
  - Multiplicacion binaria por sumas sucesivas
  - Multiplicacion acelerada
  - Multiplicador celular
- Mecanismos para la división entre números binarios
  - División binaria por restas sucesivas
  - Division acelerada entre números binarios
  - Division celular
- Operaciones entre números codificados en binario
  - Suma entre decimales codificados
  - Sumador algebraico
- Multiplicador decimal
- Mecanización de operaciones con números en coma flotante

## Unidad 2

**Instrucciones:** Son las unidades de información que determinan la secuencia de eventos y la dirección de los operandos. Con esta información, la unidad de control envía señales a las otras unidades para que actúen sobre los datos, lo cual permite llegar a un resultado.

### Elementos de una instrucción de maquina

- Código de operación: Especifica la operación que debe ser llevada a cabo.
- Referencia al operando fuente: Indica la dirección de los operandos que se van a utilizar.
- Referencia al operando resultado: Indica la dirección de almacenamiento del resultado
- Referencia a la próxima instrucción: Indica la dirección de la próxima instrucción a ser ejecutada. Si no hay una referencia específica, la próxima instrucción está en la casilla de memoria siguiente a la que se esta ejecutando.

### Localización de Los operandos fuente y resultado:

- Memoria principal o virtual: El elemento de la instrucción debe proveer la dirección en la que se encuentra la información a la que hace referencia, ya sea un operando o una instrucción.
- Registro de la UCP: La información de las instrucciones puede estar en uno de los registros de la UCP, en este caso, se debe especificar cuál de los registros contiene el dato.
- Dispositivo de E/S: La instrucción debe indicar cual modulo y cual dispositivo se utilizarán en la ejecución.

**Representación de instrucciones:** Una instrucción es una secuencia de bits. La unidad de control sabe cuantos bits de esta secuencia pertenecen a el código operativo y cuantos pertenecen a las referencias a operandos. Cada computadora puede tener distintas cantidades para cada elemento de la instrucción. La forma de la instrucción en una computadora determinada, se llama formato.

### Tipos de instrucciones

- Procesamiento de datos:
  - Aritméticas: Procesa datos numéricos.
  - Lógicas: Trata los bits de una palabra individualmente
- Almacenamiento de datos: Instrucciones con referencia a memoria. Mueven datos de registros a memorias y viceversa.
- Movimiento de datos: Instrucciones de E/S. Permiten transferir información a la memoria y muestran los resultados al usuario
- Control:
  - Prueba: Comprueban el contenido de una palabra o determinan el estado de una ejecución.
  - Bifurcación: Transfiere el control de el programa a otro punto del programa.

**Cantidad de direcciones:** Las instrucciones siempre necesitan las direcciones del primer operando, del segundo operando, del resultado y de la próxima instrucción. Sin embargo, una UCP que funcione con cuatro direcciones requiere de mucho hardware, por lo que lo mas común, es que la dirección de la

próxima instrucción sea implícita (La próxima casilla de memoria) y el resultado se almacene en un registro acumulador. Así, la instrucción solamente debe hacer referencia a dos direcciones, las de los operandos.

### **Tipos de operandos**

Direcciones: hacen referencia a la localización de otros datos en una memoria o un registro direccionable. Se pueden realizar operaciones sobre las direcciones para obtener direcciones efectivas en la memoria principal o virtual.

Números: Siempre tienen una longitud acotada. Existen tres tipos de datos numéricos:

- Coma fija o enteros.
- Coma flotante o notación científica.
- Decimales codificados.

Caracteres: Secuencias de 8 o 7 bits que representan caracteres alfabéticos.

Datos lógicos: Datos formados por conjuntos de bits que se tratan como bits independientes.

### **Tipos de operaciones**

#### **Transferencia de datos**

- La instrucción debe especificar:
  - Localización de los operandos fuente y destino
  - Longitud de los datos a transmitir
  - Modo de direccionamiento de cada operando
- Cuando uno de los datos está en la memoria:
  - Calcular la dirección de memoria.
  - Si la dirección se refiere a memoria virtual, se debe trasladar el dato a la zona de memoria real.
  - Determinar si el ítem referenciado está en la cache.
  - Si no está, buscarlo en el módulo de memoria correspondiente.

#### **Aritméticas**

- Suma
- Resta
- Multiplicación
- División
- Absolute
- Negate
- Increment
- Decrement

Lógicas: operaciones booleanas.

De Conversión: Modifican el formato de los datos, cambian la codificación.

#### De E/S:

- Programada
- Por DMA (Acceso Directo a Memoria)
- Uso de procesador de E/S

De Control del sistema: Utilizadas por los sistemas operativos.

De Transferencia del control: Se utilizan para que el programa ejecute una instrucción en una dirección de memoria específica.

- Bifurcación: Salto condicional. Contiene como operando de la instrucción, a la dirección de la próxima instrucción. Se produce un salto si el valor almacenado en el acumulador cumple la condición dada por la instrucción de bifurcación o, si se trata de una instrucción de más direcciones, se pueden agregar operandos que determinaran si la condición de salto se cumple o no.
- Salto: Pasa por alto la próxima instrucción. Como no requiere de direcciones, se pueden agregar condiciones en el campo de dirección. Un ejemplo es la instrucción que incrementa en uno una dirección de memoria y si esta dirección es igual a 0, se produce el salto.
- Llamado a subrutina: Una subrutina es un programa que puede ser llamado a ejecución a través de una instrucción. Cuando una instrucción llama a una subrutina, el contador de programas salta a la dirección de inicio de la subrutina, es decir, interrumpe el programa principal. La subrutina mantiene el control del contador de programas hasta que se ejecuta la instrucción de return, que devuelve al contador de programas a la dirección siguiente a la que llamo a la subrutina.

#### Direccionamiento

Inmediato: La dirección del operando es el valor del operando.

- Ventaja: No requiere acceder a la memoria para obtener el operando.
- Desventaja: El tamaño del numero es restringido por la capacidad del campo de dirección.

Directo: El contenido en el campo de dirección es la dirección efectiva en la memoria donde se encuentra el operando.

- Ventaja: Es simple
- Desventaja: solo se pueden direccionar memorias de poca capacidad

Indirecto: El campo de dirección hace referencia a una palabra en la memoria que contiene la dirección completa del operando.

- Ventaja: Con un campo de dirección de n bits de longitud se pueden direccionar  $2^n$  posiciones en la memoria.
- Desventaja: Requiere dos accesos a memoria, por lo que es más lento.

A registros: Direcciona a un operando que se encuentra en un registro.

- Ventajas: No se hace referencia a memoria y solo se requieren tres o cuatro bits en el campo de dirección.



- Desventaja: El espacio de direcciones es limitado.

Indirecto por registros: La dirección del operando se encuentra en un registro, tiene la misma ventaja que el direccionamiento indirecto lo cual disminuye la desventaja de tener pocos registros.

Por desplazamiento: Requiere dos campos de direccionamiento. Uno referente a un registro y el otro a un desplazamiento a agregar al contenido del registro.

- Relativo: Le suma el valor del campo de dirección a la dirección de la instrucción que se esta ejecutando. Utiliza pocos bits en el direccionamiento.
- Por registro base: Direcciona a un registro y le agrega un desplazamiento al contenido del mismo. Permite la carga de varios programas referenciando cada uno con un registro distinto.
- Indexado: El campo de dirección contiene una dirección a memoria y una dirección a un registro, el cual contiene el desplazamiento que se le aplicara a la dirección de memoria.
  - Preindexado: Primero se suman los valores del registro y el campo de dirección. Esa suma es la dirección efectiva.
  - Postindexado: Primero se busca la dirección que marca el campo de dirección, cuyo valor contenido más el valor en el registro es la dirección efectiva.

Por pila: Es una matriz lineal de posiciones de memoria, consta de un puntero (Registro que marca una dirección de memoria) que se mueve hacia arriba cuando se agrega un dato y hacia abajo cuando se extrae. Las instrucciones no requieren de un campo de direccionamiento cuando se utiliza el direccionamiento a pila, ya que es implícito.

### **Formatos de las instrucciones**

Longitud de una instrucción: Dependen de la capacidad y organización de la memoria, la estructura de los buses, la complejidad y velocidad de la UCP, etc. Las instrucciones de gran longitud hacen que la programación sea más simple, pero requieren de grandes capacidades de almacenamiento, e instrucciones de corta longitud son más rápidas y eficientes, pero más difíciles de programar.

Ubicación de los bits: Para una longitud de dirección determinada, se debe diferenciar la longitud de los campos de mando y de dirección. La cantidad de bits de direccionamiento en una instrucción está determinada por:

- Cantidad de modos de direccionamiento: Si se utiliza mas de un modo de direccionamiento, se deben utilizar algunos bits en la instrucción para determinar cual de ellos se esta utilizando en dicha instrucción.
- Cantidad de operandos: Dependiendo de cuantos operandos se utilicen en la ejecución de la instrucción, se determina la cantidad de direcciones necesarias.
- Registros y memoria: Dependiendo de cuantos registros direccionables tiene la computadora, se determina la cantidad de bits que se necesitan para hacer esos direccionamientos.
- Cantidad de conjuntos de registros: Algunas computadoras tienen un conjunto de registros para uso general. Esto disminuye la cantidad de bits del campo de direccionamiento en gran medida.
- Rango de direcciones: Dependiendo de a cuantos campos de memoria se deba acceder, se determina la longitud de el campo de direcciones, este criterio afecta menos a algunos sistemas de direccionamiento que a otros.

- Granularidad de la dirección: Si se utiliza un sistema de longitud de palabras variable, se debe referenciar cual longitud de palabra se utiliza.

### **Operación de la unidad de control**

Micro operaciones: Son los pequeños eventos de accionamiento de compuertas y espera de transferencias que comprenden el sub-ciclo de búsqueda y el sub-ciclo de ejecución, que a su vez, comprenden el ciclo de instrucción.

- Sub ciclo de búsqueda: es el punto de partida de la ejecución de una instrucción.
  - Registros involucrados:
    - Registro de direcciones de memoria: Contiene la dirección de escritura o lectura. Esta conectado al bus de direcciones.
    - Registro buffer de memoria: Contiene el ultimo valor leído o escrito en la memoria. Esta conectado al bus de datos.
    - Contador de programa: Contiene la dirección de la próxima instrucción.
    - Registro de instrucciones: Contiene el valor de la última instrucción buscada.
  - Secuencia de eventos:
    - El contador de programa envía la dirección de la próxima instrucción a el bus de datos, para que llegue a el registro de direcciones de memoria.
    - La UC envía la señal de lectura, lo cual pone la información que pidió el contador de programa en el registro buffer de memoria.
    - Se incrementa en uno al contador de programa.
    - Se pasa el contenido de el buffer de memoria a el registro de instrucciones.
- Ciclo indirecto: Es el ciclo de búsqueda de los operandos necesarios para la ejecución, en este caso, se utiliza direccionamiento indirecto.
  - La dirección contenida en el registro de instrucciones es enviada a el registro de memoria.
  - Los contenidos de memoria solicitados terminan en el buffer de memoria.
  - Los contenidos de el buffer de memoria se almacenan en el registro de instrucciones.
- Sub ciclo de ejecución: Depende de la operación que se quiera realizar. En el caso de una suma:
  - La dirección del operando que se puso en el registro de instrucciones es puesto en el registro de memoria.
  - La memoria almacena el contenido solicitado en el buffer.
  - El contenido del buffer de memoria se carga en la ULA y se envía la señal de suma.
- Ciclo de interrupción: Si se produce una interrupción se debe guardar la dirección de la instrucción que se esta ejecutando para poder volver a ella.
  - El contenido de el contador de programas es almacenado en el buffer de memoria.
  - La dirección en la que se va a guardar es enviada al registro de memoria y el contador de programas se sobrescribe con la dirección de la instrucción de la interrupción.
  - El buffer de memoria transfiere su información a la memoria.
- Ciclo de instrucción: Se basa en la existencia de un registro de dos bits llamado código del ciclo de instrucción, este indica el estado de la UCP:
  - Búsqueda (00).
  - Indirecto (01).

- Ejecute (10).
- Interrupción (11).

### Control de la UCP

- Requerimientos funcionales
  - Secuenciamiento: La UCP pasa por una serie de microoperaciones determinadas por el programa que se está ejecutando.
  - Ejecución: La UC realiza acciones para cumplir cada microoperacion en la secuencia
- Señales de control
  - Entradas:
    - Reloj: Envía pulsos periódicamente. Cada vez que la UC recibe un pulso de reloj, esta envía una señal.
    - Registro de instrucciones: contiene la instrucción que se esta llevando acabo en un momento determinado. A través de esta instrucción, la UC determina las microoperaciones que debe ejecutar.
    - Banderas: Determinan el estado de la UCP. Son necesarios para las comparaciones de datos, test de paridad, overflow, etc.
    - Señales del bus de control: Provee las señales de interrupción y de reconocimiento (ACK).
  - Salidas:
    - Señales de control para la UCP: Provocan movimientos de datos entre registros o activan funciones de la ULA.
    - Señales de control para el bus de control: Destinadas a la memoria o a los módulos de E/S.

### **Unidad 3**

#### **Definiciones basicas**

Punto de memoria: Celda capaz de almacenar un bit.

Celda de memoria: Grupo de puntos de memoria que almacena una palabra. Una palabra es un conjunto de una cantidad determinada de bits.

Direccion de memoria: Rotulo que identifica cada celda de memoria.

Direccionamiento: Proceso de ubicar las celdas de memoria para efectuar una lectura o escritura.

Lectura/Escritura: Lectura es el proceso de llevar el contenido de una celda de memoria a un registro de la máquina. Escritura es el proceso inverso.

Acceso a memoria: Proceso que involucra o lectura o escritura

Tiempo de acceso: Tiempo promedio que se tarda en realizar una escritura o una lectura.

Ciclo de memoria: Proceso que comienza con la orden de acceso a memoria y termina con el fin de la transferencia.

#### **Dimensiones de memoria**

- Capacidad: Cantidad de información que se puede almacenar.
- Caudal: Cantidad de información que se puede transferir por unidad de tiempo.
- Densidad: Cantidad de información que se almacena por unidad de longitud, superficie o volumen.

Rendimiento de la memoria: Se mide con el tiempo medio de acceso a la memoria, el cual se calcula como:  $\text{Tiempo de acierto} + (\text{Frecuencia de falla} * \text{Demora por falla})$  donde:

- Tiempo de acierto: Tiempo necesario para un acceso correcto.
- Frecuencia de falla: Cantidad de veces por unidad de tiempo que se produce una falla.
- Demora por falla: Tiempo necesario para buscar el dato cuando este no se encuentra en la memoria principal.

#### **Clasificación de las memorias**

Clasificación por jerarquía: Se caracteriza cada memoria dependiendo de su proximidad a la ULA. Las mas cercanas tienen mayor velocidad, pero menor capacidad, y las más alejadas, lo contrario.

#### **Clasificación por tipo de acceso**

- Secuenciales: Recorren las celdas una después de otras en forma secuencial.
- Cíclicas: Acceden a celdas de memoria en intervalos regulares de tiempo.
- Coordinadas: El acceso es independiente a la posición de la celda en la memoria.

#### **Clasificación por su funcionamiento**

- Estáticas: No necesitan ningún tipo de movimiento, ni eléctrico, ni magnético. SRAM, ROM.
- Dinámicas: requieren de un constante refresco para reponer cargas perdidas. DRAM.

#### Clasificación por la permanencia de la información

- Volátiles: La información se pierde después de cierto tiempo.
- No volátiles: La información permanece almacenada indefinidamente.

#### Clasificación por su forma de lectura

- Lectura destructiva: La lectura de los datos modifica la información. DRAM.
- Lectura no destructiva: No se altera el contenido al leerlo. SRAM.

Parámetros de las memorias: Se debe considerar el tamaño y el tiempo de acceso de las memorias.

#### **Memorias especializadas**

- Memorias en pilas: Son un tipo de memoria en la cual, todas las celdas están interconectadas una después de la otra, así, se pueden navegar en forma de lista. Existen dos tipos:
  - LIFO (Last In First Out): La última celda de memoria ocupada es la próxima celda que se va a leer.
  - FIFO (First In First Out): La primera celda que se ocupó es la que se lee primero, luego la segunda, y así sucesivamente.
- Memorias asociativas: No utiliza direcciones, sino que se compara el contenido de la memoria con el que se está buscando.

#### **Memorias de semiconductor**

##### Tipos:

- ROM (Read Only Memory): Memorias que no pueden ser alteradas de ninguna forma.
- PROM (Programmable ROM): Memorias que pueden ser escritas, pero no borradas.
- EPROM: Son una variante de las memorias PROM que permiten un blanqueado total para ser reprogramadas.
- EAPROM (Electrically Alterable ROM): Permiten la escritura y borrado de bits específicos en la memoria.
- DRAM (Dinamic Random Access Memory): Memorias que requieren de capacitores para almacenar la información, por lo tanto, se deben refrescar continuamente para no perder sus datos.
- SRAM (Static Random Access Memory): Compuestas por celdas constituidas por un multivibrador SR (set-reset).

**Memorias Cache:** La cache es un tipo de memoria diseñado para hacer mas rápido el acceso a memoria de la UCP. Son memorias de baja capacidad, pero muy veloces que contienen una copia de la memoria principal. Cuando la UCP quiere acceder a la memoria, primero se fija si el dato que quiere acceder se encuentra en la cache, si no lo está, el dato requerido se transfiere de la memoria principal a la cache.

Tipos de mapeo: Las celdas de las memorias cache deben hacer referencia a un conjunto de celdas de la memoria principal (llamado bloque) que se está almacenando. Esta referencia se llama mapeo.

- Mapeo directo: A cada línea de memoria cache le corresponden varios bloques de la memoria central y carga uno por vez.
- Mapeo totalmente asociativo: Todos los bloques de memoria pueden ocupar cualquier línea de cache.
- Asociativo por conjuntos: Tiene un conjunto de celdas de cache, las cuales se les asigna un grupo de bloques de memoria. Cualquier bloque de este grupo puede ser referenciado por cualquier celda de el mismo grupo.

Algoritmos para el reemplazo: Para los modos asociativos, cuando se lleva un nuevo bloque de memoria a la cache, es necesario reemplazar uno que este ocupado. El algoritmo usado más comúnmente es el de Least Recently Used (LRU), el cual reemplaza la casilla de cache que se uso menos recientemente.

Políticas de escritura: Cuando el programa que se ejecuta en la UCP modifica los datos que le provee la cache, estas modificaciones también deben ser hechas en la memoria principal antes de sobrescribir la celda de cache.

- Escribir mientras: Cuando se modifica la cache también se modifica el dato en la memoria principal. Así se asegura de que ambas memorias tienen la misma información, sin embargo, es poco eficiente ya que, si el dato en cache se modifica frecuentemente, se debe acceder a la memoria principal muy seguido.
- Escribir después: Este método consta de asociar un bit a cada celda cache que marca si los datos de el mismo, han sido modificados o no. Así, cuando se debe sobrescribir la celda de cache, se revisa si ha sido modificado para actualizar el bloque de memoria.

Tamaño de los bloques: Los bloques muy grandes significan que se desperdicia mucho espacio ya que no todas las palabras del bloque son referenciadas con la misma frecuencia, sin embargo, bloques muy chicos significan que las palabras adyacentes no siempre van a estar en el mismo bloque, lo cual puede ser perjudicial ya que los programas son normalmente almacenados en celdas adyacentes.

#### Cantidad de caches

- Caches en uno o dos niveles: Consta de un acache en el mismo chip que la UCP (nivel 1) y una cache externa (Nivel 2).
- Cache unificada y cache dividida: La cache dividida es un conjunto de dos caches, una para datos y otra para instrucciones, este método se hace mas eficiente en procesadores superescalares. La cache unificada es una única cache para cualquier tipo de información, la ventaja de esta es que si se cargan mas instrucciones que datos, la cache almacena mas de estas, por lo tanto, se disminuye el acceso a memoria.

#### **Memorias externas: Memorias conectadas a través de un módulo de E/S.**

Almacenes magnéticos: Material magnético en un sustrato aislante que produce islas magnéticas que son interpretadas como datos binarios por una cabeza lectora.

- Tambor
- Núcleos
- Disco Rígido:

- Comprendido por varios discos magnéticos apilados y divididos en pistas y sectores. Las pistas de los distintos discos se agrupan formando cilindros.
- Cada sector tiene un sistema que indica el sincronismo, el número de pista, de cabeza, de sector y el CRC (Control de Redundancia Cíclica).
- Hay una cabeza por cada superficie. Estas nunca se tocan, en caso de que el disco este girando a una velocidad muy lenta las cabezas se apoyan en unas zonas denominadas “landing zones” para no dañar el campo magnético del disco.
- Para escribir se aplican pulsos de corriente para magnetizar zonas de la superficie en un determinado sentido.
- Para leer, las cabezas sienten la fuerza magnética inducida por las partes magnetizadas en el disco.
- Disco Flexible:
  - Las cabezas de lectura y escritura se apoyan sobre los discos y aplican una presión sobre las mismas.
- Cinta: Funcionan de forma parecida a los discos, pero son cintas.

### Almacenes Ópticos

- CD-ROM (Compact Disk ROM):
  - La información se graba como huecos en la superficie de un disco.
  - La lectura se realiza con un laser que detecta si en un determinado punto hay un hueco o no.
  - CLV (Constant Linear Velocity): La información puede estar almacenada en espiral, lo cual permite que se mantenga la distancia entre bits grabados en la memoria, pero requiere de la variación de la velocidad con la que gira el disco dependiendo de si se quiere leer la información más alejada o cercana al centro.
  - CAV (Constant angular Velocity): Direcciona los bloques por pistas y sectores, lo cual requiere que las pistas mas alejadas del centro tengan una mayor distancia entre los bits que los que están mas cerca del mismo, para así poder mantener una velocidad de giro constante.
  - Formato de bloques: Los bloques en las CDROM tienen los siguientes campos de bits:
    - SINC: Indica el comienzo de un bloque.
    - Encabezamiento: Dirección de bloque y byte de modo.
      - Modo 0: Campo de datos en blanco.
      - Modo 1: Código corrector de errores y 2048B de datos.
      - Modo 2: 2336B de datos sin código de corrección de errores.
    - Datos: La información guardada en la memoria.
    - Auxiliar: En el modo 1 es el código de corrección de errores de 288B y en el modo 2 son datos adicionales.
  - Ventajas:
    - Mayor capacidad de almacenamiento que un disco magnético.
    - Fácil de duplicar.
    - Es removible.
  - Desventajas:
    - Lectura exclusiva.

- Alto tiempo de acceso.
- WORM: Son programados una única vez y después funcionan como CDROM
- EOD (Erasable Optical Disk): Permite la lectura y escritura a través de campos magnéticos y láser.
  - DVD (Digital Versatile Disk): Similar al CD pero con menor espacio entre líneas y una capa extra de material grabable.



## Unidad 4

**Función de la unidad de entrada y salida:** Conecta los módulos de la maquina con dispositivos externos (Periféricos). Esta unidad es necesaria para adaptar las velocidades de los periféricos con las de la UCP.

### **Funciones de los módulos de E/S**

**Control y atiemppamiento:** Coordina el tráfico entre las fuentes internas y los dispositivos externos.

- La UCP solicita el estado del dispositivo externo.
- El modulo externo responde enviando su estado.
- Si el dispositivo esta preparado para transmitir, la UCP solicita la transferencia de datos.
- El modulo E/S obtiene los datos del periférico.
- El modulo E/S transfiere los datos a la UCP.

**Comunicación con la UCP:** El modulo de E/S se comunica con la UCP por distintas razones.

- Decodificación de comandos: El modulo de E/S acepta los comandos de la UCP.
- Datos: La UCP y el modulo de E/S se transfieren información a través del bus de datos.
- Reporte de estados: Reporta si el dispositivo esta listo para escribir o leer o si está ocupado.
- Reconocimiento de dirección: Indica de cuál de los periféricos se busca o lleva la información.

**Comunicación con los periféricos:** Debe llevar la información del periférico a la UCP. Como los estados, datos, etc.

**Almacenamiento y adecuación de datos:** Debido a las diferentes velocidades entre la UCP y los periféricos, debe almacenar información para poder transferirlas cuando se pueda generar una interrupción.

**Detección de errores:** Debe avisar a la UCP cuando los periféricos reportan un error.

**Estructura del módulo E/S:** El modulo se conecta por un conjunto de líneas de señales con buffers. Los registros de estado funcionan como registros de control. La comunicación con la UCP se hace a través de líneas de control.

**Entradas y salidas programadas:** La UCP le otorga el control total de la operación a la UE/S.

**Instrucciones de E/S:** Las instrucciones de la UCP que esten referidos a una operación de E/S deben tener un identificador que diga la dirección del periférico al que se refiere. Si la UE/S, la UCP y la memoria comparten un bus hay tres modos de direccionamiento posibles:

- Direccionamiento de E/S por mapeo de memoria: Se utiliza el mismo campo de direcciones tanto para acceder a la memoria como para acceder al módulo de E/S.
- Direccionamiento de E/S aislado: Se utilizan líneas separadas para direcciones de memoria y de dispositivos de E/S.

**Entrada/Salida por interrupción:** La UCP ejecuta la operación de E/S y por medio de una interrupción continúa ejecutando las instrucciones del programa. Una vez terminado, la UCP y el módulo se E/S se transfieren los datos necesarios.

- Desde el punto de vista de la UE/S:
  - Recibe la señal de lectura o escritura por parte de la UCP.
  - La UE/S lee la información del periférico que indico la UCP.
  - Una vez que la información está en el registro de E/S, se transmite una señal de interrupción a la UCP.
- Desde el punto de vista de la UCP:
  - La UCP envía la señal de lectura o escritura al módulo de E/S y continúa ejecutando la próxima instrucción.
  - Cada vez que termina un ciclo de instrucción, revisa si hay interrupciones.
  - Si la hay, envía la señal de reconocimiento a la U/ES y guarda el contenido del bus de datos en la memoria y continua con la instrucción anterior.

#### Reconocimiento del módulo que realiza la interrupción:

- Múltiples líneas de interrupción: Provee varias líneas entre la UCP y la E/S. Si ocurren interrupciones simultáneas se responde a la línea con más prioridad.
- Votación por software: Cuando la UCP recibe una interrupción, revisa cuál de los módulos de E/S la emitió a través de un programa o a través de un registro de estado en los módulos E/S. El orden en el que se consultan los módulos es el orden de prioridad en caso de interrupciones simultáneas.
- Encadenado margarita: La UCP envía la señal de reconocimiento de interrupción y la unidad de E/S que la emitió envía su identificador. Todo esto funciona a través de un único bus de interrupción la que están conectados todos los módulos de E/S. El orden en el que están dispuestos los módulos determina su prioridad.
- Arbitraje del bus: Los módulos de E/S se apropian del bus de datos, así, no es necesario que la UCP reconozca el modulo de E/S, ya que solo hay un único modulo conectado al bus.

**Acceso directo a memoria:** Permite el intercambio de datos entre la UE/S y la memoria sin la intervención de la UCP (excepto por la inicialización).

**Funcionamiento:** Requiere de un módulo con acceso al bus que pueda actuar como la UCP para realizar transacciones, este es el módulo DMA.

- Cuando la UCP desea leer o escribir en un periférico envía un comando al DMA, el cual contiene:
  - El tipo de operación (L/E).
  - La dirección del periférico.
  - La locación de memoria de arranque para la lectura o escritura.
  - La cantidad de palabras que va a leer o escribir.
- La UCP delega la operación de E/S al DMA y continua con otra instrucción.
- Una vez el DMA obtuvo la información del módulo E/S, envía una solicitud de interrupción a la UCP.

El DMA debe usar el mismo bus que a UCP, por lo tanto, debe actuar solamente cuando esta no lo haga o debe suspender temporalmente la operación de la UCP.

#### Configuraciones de DMA:

- Los módulos de E/S y el DMA están todos conectados al bus del sistema y el modo de E/S es programado.
- Los módulos de E/S tienen integrado el módulo DMA
- Los módulos de E/S están conectados a través de un bus al modulo DMA, el cual , a su vez, se conecta a el bus del sistema.

### **Canales de entrada y salida y procesadores**

**Interfase externa:** Interconexión entre el modulo de E/S y el periférico. Se requiere un registro que pueda almacenar datos y entregarlos a la velocidad del periférico.

- El módulo de E/S envía una solicitud de envío de datos.
- El periférico reconoce el pedido.
- El modulo de E/S transfiere los datos.
- El periférico reconoce la aceptación de los datos.

### **Tipos de interfase**

- Conexión en serie: Se transfiere la información bit a bit.
- Conexión en paralelo: Transfiere una cierta cantidad de bits simultáneamente.

### **Configuraciones punto a punto y configuraciones multipunto**

## Unidad 6

**RISC (Reduced instruction Set Computers):** Maquinas con hardware simple diseñadas con el propósito de llegar a una mayor velocidad de ejecución de programas.

Principios fundamentales:

- Pocas y simples instrucciones de máquina.
- Gran cantidad de registros de uso general.
- Técnicas de compilación que optimizan el uso de los registros.
- Optimización del cauce de instrucciones.

**Características de la ejecución de instrucciones:** Las diferencias entre las operaciones de los lenguajes de alto nivel y las instrucciones de la maquina se llaman vacío semántico. Al disminuir este vacío, se debieron hacer mediciones sobre los programas para saber como funcionan en la práctica.

Mediciones sobre programas:

- Operandos: Las mediciones sobre los operandos permiten seleccionar la organización de memoria y los modos de direccionamiento. Lo que se debe conocer de ellos es:
  - Si son constantes o variables.
  - Si son enteros, en coma flotante.
  - Si son escalares, matrices, listas o estructuras de datos.
  - Si los operandos son globales o argumentos de procedimientos locales.
  - La cantidad, tamaño y frecuencia con que son utilizados.
- Operaciones: Se debe medir la frecuencia con la que aparecen en los programas.
- Ejecución y texto del programa:
  - Mediciones estáticas: Se refieren al texto del programa e indican el tamaño de la memoria designado a las instrucciones.
  - Mediciones dinámicas: Tienen en cuenta los códigos de maquina en vez de los códigos fuente.

Mediciones sobre las instrucciones: Las mediciones realizadas en base a las instrucciones de la máquina, se refieren a:

- La frecuencia con la que son ejecutadas.
- Tiempo gastado en llamados y retornos a procedimientos.
- La importancia de las estructuras de control.
- La cantidad de registros usados eficientemente, la necesidad de los mismos y la vida del registro (Periodo de tiempo entre la carga de un registro y el ultimo uso antes de ser cargado nuevamente).

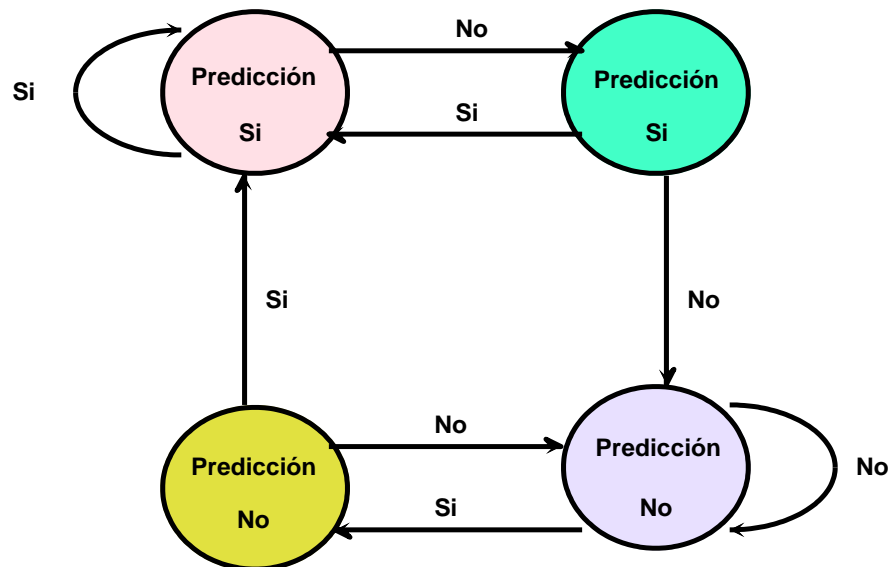
**Encauzamiento de instrucciones:** Forma de obtener paralelismo (Realización simultanea de varias instrucciones) en sistemas con un único procesador.

Secuenciamiento: Se basa en el hecho de que, en el ciclo de instrucción básico, existen momentos en los que la UCP debe esperar un resultado. El Secuenciamiento aprovecha estos momentos para que la UCP

ejecute sub ciclos de la próxima instrucción. Esto tiene problemas con las bifurcaciones ya que si en la instrucción actual se causa una interrupción, el proceso de la siguiente instrucción que se estaba llevando a cabo no sirve.

### Procesamiento de las bifurcaciones

- **Cauces múltiples:** Dos canales paralelos por los que se buscan ambas instrucciones (la próxima y la de la bifurcación) y se elige la que se va a ejecutar. Problemas:
  - Con múltiples cauces, hay mas retardos de acceso a memoria y registros.
  - Antes de resolver la bifurcación, pueden aparecer mas instrucciones, lo que se solucionaría con mas cauces.
- **Prebusqueda de blancos:** La dirección de salto se busca y se guarda antes de ejecutar la condición de la bifurcación.
- **Buffer de lazo:** Es una memoria de muy alta velocidad de acceso que guarda las ultimas instrucciones buscadas en una determinada cantidad de ciclos de instrucción. Si se produce una bifurcación, la instrucción se toma del buffer (si es que esta almacenada allí). Ventajas:
  - No se deben buscar en la memoria las instrucciones que estén en el buffer.
  - Muy útiles para loops.
- **Predicción de bifurcaciones:** Las bifurcaciones se pueden predecir con las siguientes técnicas:
  - Predicción que nunca ocurre: No busca la dirección de la instrucción de bifurcación.
  - Predicción que siempre ocurre: Siempre busca la dirección de la instrucción de bifurcación.
  - Predicción del código operativo
  - Llave ocurre/no ocurre: Asocian a la instrucción algunos bits que indican que decisión tomar cuando se ejecuta.
  - Tabla histórica de las bifurcaciones: Es un secuencial de cuatro estados:



**Si:** bifurcación positiva  
**No:** bifurcación negativa.

- Bifurcaciones demoradas

**Utilización de una gran cantidad de registros:** Mientras mas registros hay, mas rápida serán las preparaciones para las bifurcaciones ya que no habrá que guardar los datos en la memoria. Sin embargo, las instrucciones deben tener campos para direccionar a esos registros.

## **Arquitecturas RISC**

**Operaciones de ciclo único:** En sistemas RISC se trata que las instrucciones sean lo mas simples posibles, es decir, no mas complicadas que las microinstrucciones de una maquina microprogramada. Esto significa que las instrucciones mas complejas se deben utilizar instrucciones básicas para crear un programa que las lleve a cabo o, si son utilizadas frecuentemente, pueden ser incorporadas como instrucciones independientes. Además de la frecuencia de uso, se debe asegurar que no disminuya la velocidad de las instrucciones primitivas.

**Diseño Carga/almacenamiento y Registro a Registro:** Las arquitecturas RISC únicamente permiten el uso de operaciones Load y Store para accesos a memoria, ya que las instrucciones solo realizan operaciones entre registros.

**Instrucciones y de modos de direccionamiento:** Al tener pocas instrucciones se simplifica la lógica de control. Los sistemas RISC usan los sistemas de direccionamiento más simples.

**Formato fijo de instrucciones:** Los sistemas RISC utilizan un sistema fijo de instrucciones que simplifica la decodificación.

**Encauzamiento:** Las arquitecturas RISC utilizan el encauzamiento.

**Controversia RISC-CISC:** Los sistemas RISC y CISC son completamente opuestos. Por lo que son comparados frecuentemente para intentar comprobar cuál es más eficiente. Las comparaciones que se hacen pueden ser cuantitativas (Velocidades y tamaños) o cualitativas (Soporte de lenguajes de alto nivel). Problemas con las comparaciones:

- No hay maquinas RISC y CISC que sean completamente comparables en todos los ítems.
- No hay sistemas definitivos de pruebas de programas.
- Es difícil extraer los efectos de hardware de los efectos de compilador.

**Sistemas superescalares:** Consiste en la ejecución de varias instrucciones comunes de forma simultanea e independiente.

## **Superescalares y superencauzados**

- Superencauzados: Se comienza a procesar las instrucciones de forma encauzada, pero en medio tiempo de reloj.
- Superescalar: Realiza dos instrucciones al mismo tiempo

## **Limitaciones**

- Dependencia real de los datos: no se pueden realizar dos instrucciones en paralelo si los datos de una van a ser modificados por la otra.
- Dependencia de los procedimientos: No se puede realizar la próxima instrucción en paralelo si esta es una bifurcación ya que se debe saber si se cumple la condición de salto o no.

- Conflictos de los recursos: Si ambas instrucciones requieren de acceso a la memoria al mismo tiempo, una debe esperar a la otra.
- Dependencia de salida
- Antidependencia