

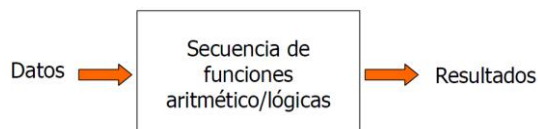
Clase 1

Temas de clase

- Temas básicos
- Programas
- Arquitectura Von Neumann
- Repertorio de instrucciones
- Ciclo de instrucción
- Simulador
- Subrutinas
- Pasaje de argumentos

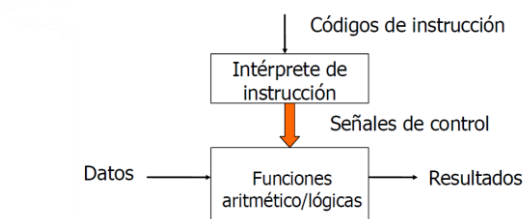
Programas

- **Antes:**



Programación en hardware: cuando cambiamos las tareas, debemos cambiar el hardware

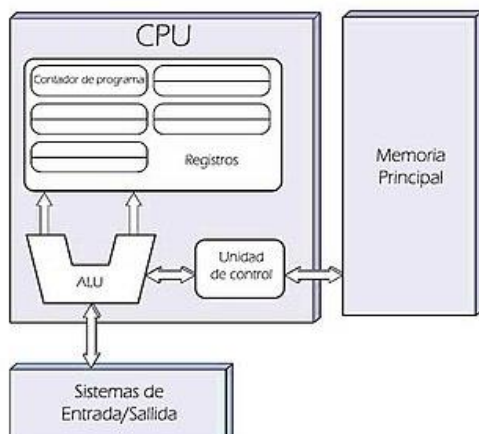
- **Ahora:**



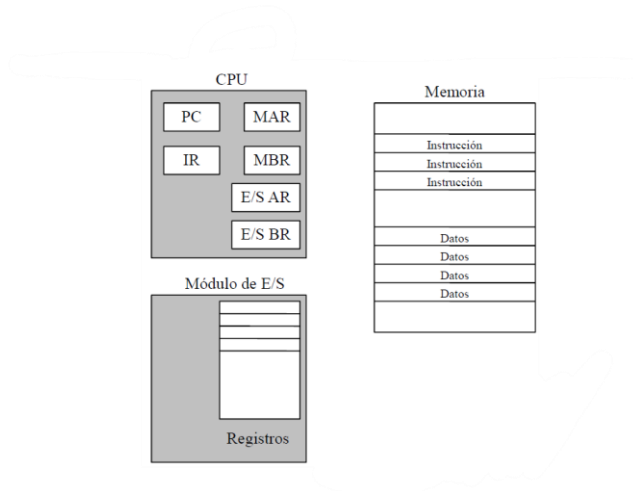
Programación en software: en cada paso se efectúa alguna operación sobre los datos

Arquitectura Von Neumann

- La unidad central de procesamiento (CPU) está constituida por la unidad de control (UC) y la unidad aritmético-lógica (ALU).
- Datos e instrucciones deben introducirse en el sistema y los resultados se proporcionarán mediante componentes de entrada/salida (E/S).
- Se necesita almacenar temporalmente datos e instrucciones:
 - Memoria Principal



Componentes de una computadora:



PC = Contador de programa

IR = Registro de instrucción

MAR = Registro de dirección de memoria

MBR = Registro de buffer de memoria

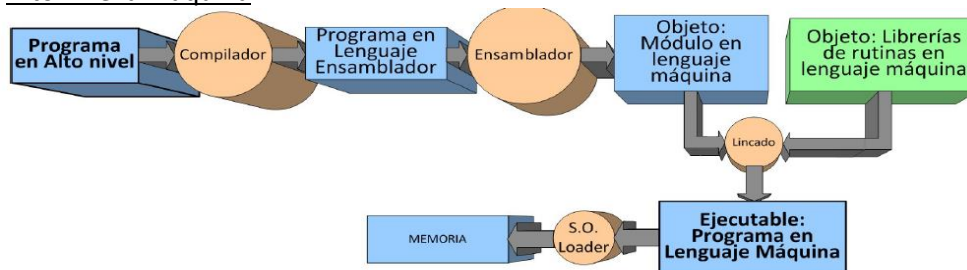
E/S AR = Registro de dirección de E / S

E/S BR = Registro buffer de E / S

Repertorio de instrucciones

- Es el conjunto completo de instrucciones que se realizan en una CPU.
 - Código máquina
 - Binario
- Representado simbólicamente por un conjunto de códigos de ensamblaje
 - De operaciones:
 - ADD (sumar), SUB (restar), LOAD (cargar datos en un registro)
 - De operandos:
 - ADD BX, PEPE; sumar contenidos de reg BX y dirección PEPE, el resultado se guarda en reg BX

Alto nivel a maquina



Elementos de una instrucción

En una arquitectura Von Neumann, las instrucciones de máquina que componen un programa y los datos que estas manipulan están almacenadas en la misma memoria. Cada instrucción consta de varios elementos esenciales:

1. **Código de Operación (COD OP):** Es una porción de la instrucción que especifica la operación que se debe realizar, como suma, resta o multiplicación.
2. **Referencia a operandos fuentes:** Son las direcciones de memoria donde se encuentran los datos que se utilizarán en la operación. Por ejemplo, en una operación de suma, los sumandos son los operandos fuentes.
3. **Referencia al operando resultado:** Es la dirección de memoria donde se almacenará el resultado de la operación. Después de realizar una suma, por ejemplo, la máquina debe saber dónde guardar el valor numérico que representa el resultado.
4. **Referencia a la siguiente instrucción:** Indica qué instrucción se debe ejecutar después de completar la operación actual. Esto permite que la máquina siga ejecutando el programa secuencialmente.

En resumen, cada instrucción de máquina en una arquitectura Von Neumann contiene información sobre la operación a realizar, las direcciones de memoria de los operandos fuentes, la dirección de memoria donde se guardará el resultado y la referencia a la siguiente instrucción a ejecutar.

Los operandos se almacenan en Memoria Principal(o virtual o cache), registro de la CPU, dispositivo E/S, por pila, acumulador, memoria-memoria o registro-registro

Tipos de instrucciones

- **Procesamiento de datos:** instrucciones aritmético-lógicas
- **Almacenamiento de datos:** instrucciones de memoria
- **Transferencia de datos:** instrucciones de E/S
- **Control:** instrucciones de testeo y flujo del programa

Decisiones en el diseño del conjunto de instrucciones

El diseño de un conjunto de instrucciones involucra cinco elementos fundamentales que están interrelacionados y afectan las características del conjunto de instrucciones:

1. **Tipos de operandos:** Se refiere a las diferentes formas de representar información en binario, como números, direcciones, datos lógicos y codificaciones (por ejemplo, ASCII).
2. **Repertorio de operaciones:** Incluye la cantidad y tipos de operaciones que la máquina puede realizar, así como la complejidad de dichas operaciones y el hardware necesario para implementarlas.
3. **Formatos de instrucción:** Define la longitud de las instrucciones y cómo se dividen en campos, como el código de operación y las direcciones de los operandos. La longitud y cantidad de campos afecta la flexibilidad y complejidad de la máquina.
4. **Registros:** Son elementos de hardware que almacenan datos o direcciones y pueden ser específicos para ciertas operaciones. La cantidad y tipo de registros influyen en la arquitectura de la CPU y en la forma de programarla.
5. **Modos de direccionamiento:** Determinan cómo se especifica la ubicación de los operandos en las instrucciones. Los diferentes modos de direccionamiento permiten mayor flexibilidad y eficiencia en la programación.

Estos cinco elementos interactúan entre sí y su diseño afecta las ventajas y desventajas de la arquitectura de la computadora en cuestión. La elección de cada uno de estos elementos depende de factores como la tecnología disponible, la intención del diseñador y la complejidad deseada.

RISC (Computadora de conjunto reducido de instrucciones) contrapuesto a **CISC** (Computadora de conjunto complejo de instrucciones)

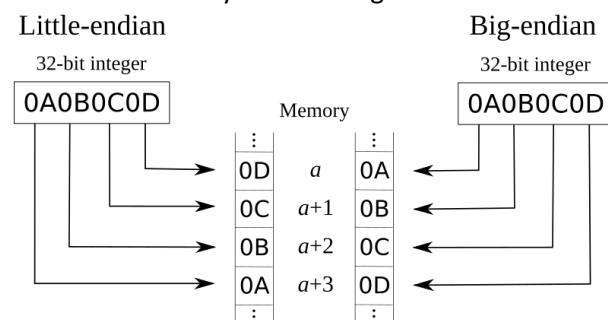
Tipos de operandos

- Direcciones
- Números
 - Punto fijo o punto flotante
- Caracteres
 - ASCII, EBCDIC...etc.
- Datos lógicos
 - Bits (1 o 0)
 - Ej: flags o indicadores

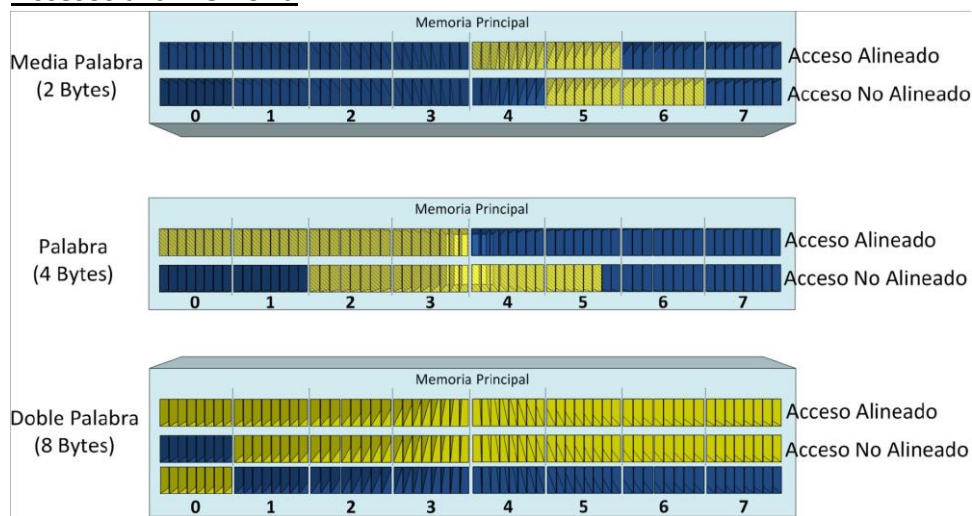
Orden de los bits

Big endian: el byte más significativo en la dirección con valor numérico más bajo

Little endian: el byte menos significativo en la dirección con valor numérico más bajo



Accesos a la memoria



(Si se permiten, los accesos no alineados son más lentos)

Tipos de operaciones en un procesador

1. **Transferencia de datos:**
 - Implica especificar ubicación de operandos fuente y destino, tamaño de datos, y modo de direccionamiento.
 - Diferentes movimientos requieren diferentes instrucciones (Reg-Reg, Reg-Mem, Mem-Reg) o una instrucción con diferentes direcciones (MOV destino, fuente).
2. **Operaciones aritméticas:**
 - Incluyen Add, Subtract, Multiply, Divide para números enteros y, posiblemente, números en punto flotante.
 - Pueden incluir operaciones como Increment, Decrement, Negate, Absolute y Shift left/right.
3. **Operaciones lógicas y de conversión:**
 - Manipulan bits individualmente con operaciones Booleanas (AND, OR, XOR, NOT) y otras como Rotate left/right.
 - Cambian formatos de datos, como conversiones de binario a decimal o de EBCDIC a ASCII.
4. **Entrada/Salida:**
 - Acciones específicas con pocas instrucciones (IN, OUT).
 - Se pueden realizar mediante instrucciones de movimiento de datos (MOVE) o a través de un controlador aparte (DMA).
5. **Control del sistema y Control de flujo:**
 - Modifican el valor en el registro PC.
 - Incluyen Salto Incondicional (JMP), Salto Condicional (JZ), y Salto con retorno o llamada a subrutina (CALL, RET).

Métodos de direccionamiento

Los modos de direccionamiento son las diferentes formas en que un programador puede especificar la ubicación de los operandos fuente, los operandos resultado y la siguiente instrucción en la memoria. Estos modos permiten localizar y acceder a los elementos en la memoria de manera eficiente y flexible. En resumen, los modos de direccionamiento son métodos para referenciar direcciones de memoria en instrucciones y operaciones.

- **Inmediato**
- **Directo de memoria o Absoluto**
- **Directo de Registro**
- **Indirecto de memoria (en desuso)**
- **Indirecto con registro**
- **Indirecto con Desplazamiento**
 - basado, indexado o relativo al PC
 - Pila (o relativo al SP)

Modos de direccionamiento y memoria:

1. **Memoria de instrucciones y memoria de datos:**
 - La memoria de instrucciones almacena instrucciones completas.
 - La memoria de datos contiene operandos fuente y resultados.
 - Ambas memorias tienen direcciones.
2. **Modo de direccionamiento inmediato:**
 - El valor del operando está contenido en la instrucción.
 - Típicamente usado para entregar valores constantes a la máquina.

3. Modos de direccionamiento directo:

- Directo de memoria (absoluto): La instrucción contiene la dirección de memoria donde se encuentra el operando.
- Directo de registro: La instrucción hace referencia a un registro en el banco de registros dentro de la CPU.

4. Modos de direccionamiento indirecto:

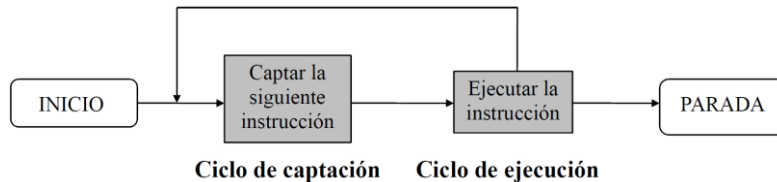
- Indirecto de memoria: En desuso, pero existe.
- Indirecto con registro: Usa registros internos.
- Indirecto con desplazamiento: Determina la ubicación del operando mediante la suma del valor en un registro y un desplazamiento.
 - Variaciones: Basado, Indexado, Relativo al contador de programa, Pila, Relativo al SP.

La separación entre memoria de instrucciones y de datos permite acceder a estas partes en diferentes tiempos, llamado ciclo de instrucción.

Ciclo de instrucción básico

El ciclo de instrucción se divide en dos grandes pasos: primero, captar la instrucción, donde la CPU busca el patrón de bits que representa la instrucción en la memoria y lo guarda en la CPU; y segundo, ejecutar la instrucción, realizando la operación requerida y utilizando los operandos indicados, dejando el resultado en el lugar especificado.

El ciclo de instrucción se compone de varios estados, que describen el algoritmo realizado por la unidad de control para ejecutar una instrucción. La ejecución puede implicar el uso de la ALU para realizar operaciones específicas, aunque algunas instrucciones no requieren su uso. El diagrama de estados del ciclo de instrucción muestra la secuencia de pasos necesarios para completar una única instrucción.



1. Ciclo de captación:

- La dirección de la instrucción se encuentra en el registro Contador de Programa (PC).
- La Unidad de Control (UC) captura la instrucción desde la Memoria, y la instrucción se almacena en el registro de instrucción (IR).
- El registro PC se incrementa, a menos que se indique lo contrario.
- La UC interpreta la instrucción captada y realiza la acción requerida.

2. Ciclo de ejecución:

- Acciones posibles: a. Procesador - memoria: Transferencia de datos entre CPU y Memoria. b. Procesador - E/S: Transferencias de datos entre CPU y módulo de E/S. c. Procesamiento de datos: Operaciones aritméticas o lógicas con los datos. d. Control: Alteración de la secuencia de ejecución, como instrucciones de salto.
- Pueden ocurrir combinaciones de las acciones anteriores.

Subrutinas

- Innovación en lenguajes de programación
- Programa auto-contenido
- Puede invocarse desde cualquier punto de un programa
 - Mediante instrucción CALL
- Brinda economía (código usado varias veces) y modularidad (subdivisión en unidades pequeñas).
- Requiere pasaje de argumentos (parámetros)
 - Por valor (copia de una variable)
 - Por referencia (dirección de la variable)

Pasaje de argumentos a subrutinas

•Vía registros

- El número de registros es la principal limitación
- Es importante documentar que registros se usan

•Vía memoria

- Se usa un área definida de memoria (RAM).
- Difícil de estandarizar

•Vía pila (stack)

- Es el método más ampliamente usado.
- El verdadero “pasaje de parámetros”.
- Independiente de memoria y registros.
- Hay que comprender bien cómo funciona porque la pila (stack) es usada por el usuario y por el sistema.

En x86, SP apunta al último lugar usado

Funcionamiento de una pila

- El operando está (de forma implícita) en la cabeza de la pila
- Se requiere un registro Puntero de Pila (SP)
 - Contiene la dirección de la cabeza de la pila
- Operaciones sobre la pila
 - PUSH; operación de Apilar
 - POP; operación de Desapilar
 - Son inversas entre sí

Posibles pasos en un procedimiento

1. Salvar el estado de BP (viejo BP)
2. Salvar estado de SP (BP=SP)
3. Reservar espacio para datos locales (opcional)
4. Salvar valores de otros registros (opcional)
5. Acceder a parámetros
6. Escribir sentencias a ejecutar
7. Retornar parámetro (opcional)
8. Regresar correctamente del procedimiento

Salida del procedimiento

- Los registros salvados en la pila deben ser descargados en orden inverso.
- Si se reservó espacio para variables locales, se debe reponer SP con el valor de BP que no cambió durante el procedimiento.
- Reponer BP.
- Volver al programa que llamó al procedimiento con RET.

Clase 2

Temas de clase

- Interrupciones

Interrupciones: Las interrupciones son mecanismos que permiten alterar el procesamiento normal de una CPU, interrumpiendo el ciclo de instrucción secuencial para ejecutar otras instrucciones. Pueden originarse interna o externamente a la CPU y se utilizan por diversas razones, como errores en la ejecución, temporizadores internos, operaciones de E/S o fallos de hardware.

Al solicitar una interrupción, la CPU transfiere el control a un Gestor de Interrupción (GI), previo a salvar el estado del procesador. El GI responde a la causa de la interrupción, resguardando los recursos de la CPU durante el proceso y retornando a la ejecución normal del programa interrumpido.

Existen GIs específicos para cada tipo de interrupción y se clasifican en:

- **No enmascarables:** requieren respuesta rápida y eficiente debido a eventos de alta prioridad o peligrosos.
- **Enmascarables:** pueden demorarse o no ser atendidas ya que sus eventos no representan peligro inmediato y pueden ser inhibidos con instrucciones especiales.

Las interrupciones se originan de dos fuentes: hardware y software.

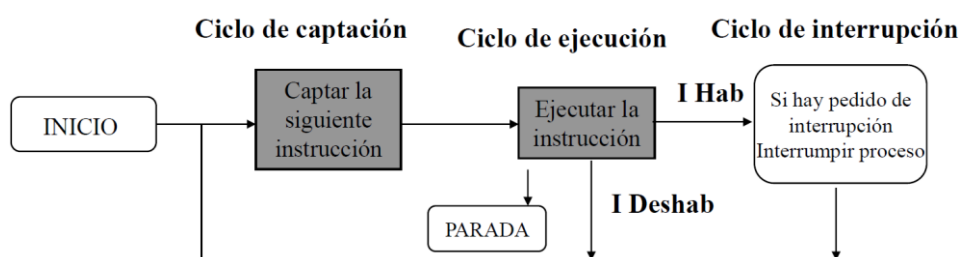
• **Interrupciones por hardware** son generadas por dispositivos de E/S y son eventos externos no planeados o asincrónicos, conocidos como interrupt request. No están relacionadas con el proceso en ejecución en ese momento.

• **Traps o excepciones** son interrupciones por hardware creadas por el procesador moderno en respuesta a eventos internos, como condiciones excepcionales, fallas de programa, fallas de hardware o accesos no alineados a zonas de memoria protegidas.

• **Pedidos de interrupción** son eventos no planificados que no están relacionados con el programa en ejecución en la CPU. Por ejemplo, periféricos conectados a dispositivos de entrada/salida generan solicitudes de interrupción para controlar la máquina y realizar tareas necesarias en relación a ese dispositivo de E/S. También pueden ser causadas por fallas de hardware como fallas de memoria o accesos no alineados a zonas de memoria protegidas.

• **Interrupciones por software** son instrucciones explícitas que afectan al procesador de la misma manera que las interrupciones por hardware. Permiten depurar gestores de interrupción, invocar funciones del S.O., cargar subrutinas del S.O. en algún lugar y utilizarlas sin conocer la dirección de la rutina en tiempo de ejecución. Si no se tuvieran, sería necesario escribir todas las funciones necesarias o reemplazar en el código las direcciones de todas las funciones invocadas del BIOS y S.O. al cargar un programa.

Ciclo de Instrucción



El Ciclo de Instrucción (CI) puede modificarse en tres pasos:

- 1) Ciclo de captación
- 2) Ciclo de ejecución
- 3) Ciclo de gestión de interrupciones (si las hay).

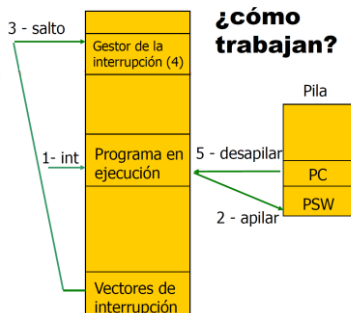
Si las interrupciones están:

- **Deshabilitadas:** el CI es un proceso tradicional de dos pasos (1 y 2).
- **Habilitadas:** se añade el tercer ciclo para verificar si hay solicitudes de interrupción.

Esto se hace mediante una bandera interna (flag) en la CPU que indica si hay un pedido de interrupción o no. Si no hay solicitudes, la máquina continúa ejecutando instrucciones y verifica nuevamente al final de cada ciclo si hay interrupciones pendientes.

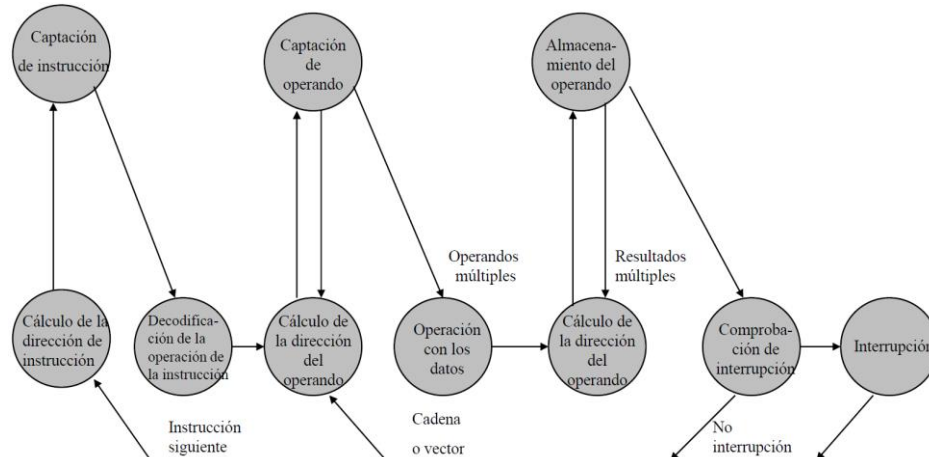
Ciclo de ininterrupción

Durante el ciclo de interrupción, se verifica si hay una solicitud de interrupción pendiente mediante una señal (flag). Si no hay señal, se procede con la siguiente instrucción. Si hay una solicitud de interrupción, se suspende la ejecución del programa y se guarda el contexto, que incluye la dirección de la próxima instrucción a ejecutar y un reflejo mínimo del estado del procesador (como el valor de PC y las flags). Luego, se empieza a ejecutar el Gestor de Interrupción, inhibiendo otras interrupciones.



El programa en ejecución se detiene para guardar el contexto y apilar la palabra de estado (PSW) y la dirección de retorno (PC - PROGRAM COUNTER) en la pila. Luego, el control se transfiere al gestor de interrupción mediante el mecanismo de vectores de interrupción. La ejecución del gestor de interrupción sigue una secuencia de instrucciones para atender la interrupción. Al finalizar, se desapilan los datos del estado del procesador y se retoma la ejecución del programa en el punto de interrupción.

Diagrama de estados de un ciclo de instrucciones con interrupciones



El ciclo de instrucción incluye interrupciones. Si están enmascaradas, se sigue el ciclo tradicional sin comprobación. Durante una interrupción, no se pueden recibir más interrupciones. La jerarquía de interrupciones determina cuántas pueden interrumpir simultáneamente, permitiendo interrupciones múltiples.

Interrupciones Múltiples

Cola de solicitudes: Si las solicitudes tienen la misma prioridad, se atiende una y las otras se ponen en cola. Las interrupciones pueden estar habilitadas o inhabilitadas, lo que afecta cómo se manejan.

Interrupciones inhabilitadas:

- El procesador puede ignorar la señal de petición si se produce una interrupción.
- Las interrupciones pendientes se examinan cuando se vuelven a habilitar.

Interrupciones habilitadas:

- Las interrupciones se inhabilitan al recibir una solicitud, se gestionan y luego se habilitan nuevamente.
- Se manejan en orden secuencial estricto.

Máquinas con múltiples fuentes de interrupción pueden tener interrupciones habilitadas/inhabilitadas (atención secuencial) o establecer prioridades para permitir interrupciones anidadas, donde un solicitador de mayor prioridad interrumpe a uno de menor prioridad.

Reconocimiento de Interrupciones

Interrupciones multinivel: Cada dispositivo con capacidad de interrupción tiene una entrada física conectada a la CPU. Es simple pero costoso.

Línea de interrupción única: Una entrada física conectada a todos los dispositivos. Se debe preguntar a cada dispositivo si ha generado una interrupción (técnica Polling/encuesta).

Interrupciones vectorizadas: El dispositivo que desea interrumpir coloca un identificador (vector) en el bus de datos además de la señal de interrupción. Puede ser colocado directamente por el periférico o por un controlador de interrupciones.

Cómo se llega a la primera instrucción del gestor que corresponde

Existen dos métodos principales para llegar a la primera instrucción del gestor de interrupción:

1. **El periférico coloca un valor (ID) en el bus de datos:** El dispositivo que solicita la interrupción coloca su identificador en el bus de datos, lo que permite buscar en memoria la información necesaria para atender la interrupción.
2. **Controlador de Interrupciones (PIC):** Para evitar que el periférico sea el que coloca el ID en el bus de datos, se implementó un PIC que actúa como intermediario, ordenando las interrupciones en fila y notificando a la CPU cada vez que haya una interrupción pendiente.

Interrupción Vectorizada

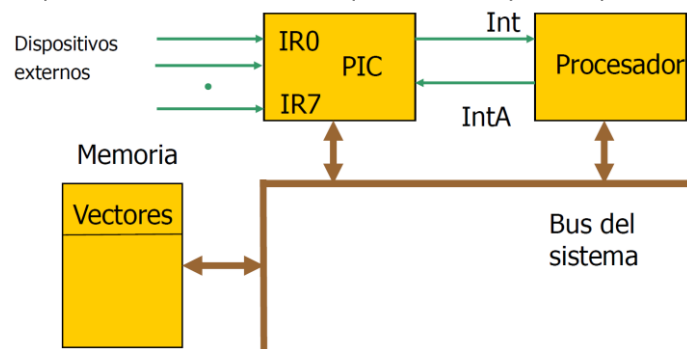
La interrupción vectorizada es una técnica que automatiza el proceso de identificar la fuente de interrupción utilizando una única entrada. Cuando la CPU acepta atender la interrupción, el solicitante envía automáticamente un número identificador (vector) a la CPU. Este número proporciona información sobre la fuente de la interrupción y el Gestor de Interrupción que se encargará de atenderla.

El vector puede ser proporcionado directamente por el periférico o, más comúnmente, por un controlador específico (como el PIC) que se encarga de gestionar las solicitudes y entregar los

identificadores o vectores correspondientes. Esto permite una mayor organización y control sobre el proceso de interrupción.

Nuevo conexionado y arquitectura

El bus del sistema se compone de buses de datos, control y direcciones. La estructura interna se modifica para incluir un Controlador de Interrupciones Programable (PIC), en lugar de la arquitectura Von Neumann que solo incluye CPU y memoria.



Controlador de Interrupciones Programable (PIC)

El PIC se conecta con el procesador a través de dos líneas, una para la solicitud de interrupción y otra para la respuesta del procesador. El PIC tiene 8 registros, uno por cada solicitador, y trabaja con el esquema de vectores de interrupción.

Atención de interrupciones

El procesador recibe una señal de interrupción y la marca en los flags. A través de la línea IntA, el procesador le informa al PIC que atenderá la solicitud. El PIC actúa como nexo entre el procesador y los dispositivos externos.

El número de vector (8 bits, valor de 0 a 255) es enviado por el PIC al procesador, que utiliza este valor para acceder a la zona de vectores de interrupción en memoria. La Unidad de Control (UC) ejecuta la primera instrucción del Gestor de Interrupción (GI), que debe terminar con una instrucción de retorno de interrupción (IRET).

Funcionamiento del PIC

El PIC es una variante de un módulo de E/S y utiliza 4 de las 8 posibles interrupciones (Int0 a Int3). Es programable, lo que significa que la CPU debe configurarlo. Contiene registros específicos para el control y la administración del intercambio de información entre el periférico y la CPU.

Registros del PIC

- EOI (20H): Registro de control o comando del controlador.
- IMR (21H): Registro de máscara, define si se atenderá o no la solicitud.
- IRR (22H): Registro de petición, guarda información sobre el solicitador de interrupción.
- ISR (23H): Registro de servicio, indica qué interrupción se está atendiendo.

Proceso de atención de interrupciones

Cuando el procesador acepta una interrupción, el PIC coloca en el bus de datos el valor almacenado en uno de los registros INT 0 a INT 7. La CPU continúa con sus tareas y espera la orden EOI (20H) para terminar la atención de interrupción y proceder a limpiar el registro de servicio.

Este sistema permite atender múltiples interrupciones de forma secuencial o trabajar con todas las interrupciones de manera anidada, según la prioridad implícita en los registros de atención de interrupciones.

Clase 3

Entrada/Salida en Arquitectura Von Neumann:

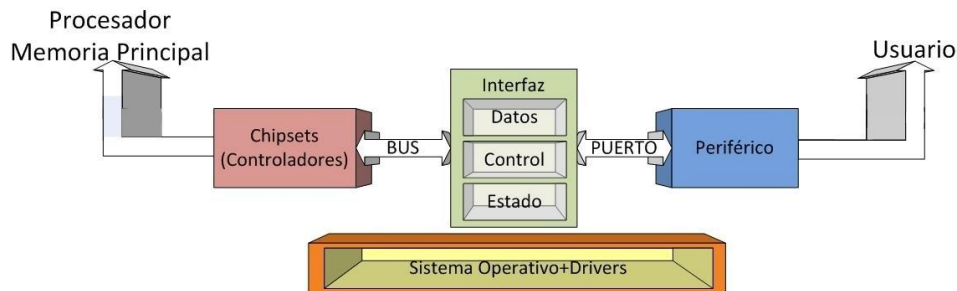
La **Entrada/Salida (E/S)** es el tercer subsistema en la arquitectura Von Neumann y representa el mecanismo que conecta la computadora con el exterior. Los dispositivos periféricos, que son generalmente más lentos que la CPU y la memoria principal, difieren en métodos de transmisión de información, velocidades, formatos de datos y tamaño de palabras.

Problemas de Entrada/Salida:

- Gran variedad de periféricos con diferentes métodos de operación.
- Transmisión de distintas cantidades de datos.
- A diferentes velocidades.
- Uso de diferentes formatos de dato y tamaño de palabra.
- Todos más lentos que la CPU y la RAM.
- Necesidad de módulos de E/S con cierta "inteligencia".

Módulos de E/S:

- Realizan la interfaz entre el procesador y la memoria (bus) y los periféricos.
- Pueden manejar uno o más periféricos.



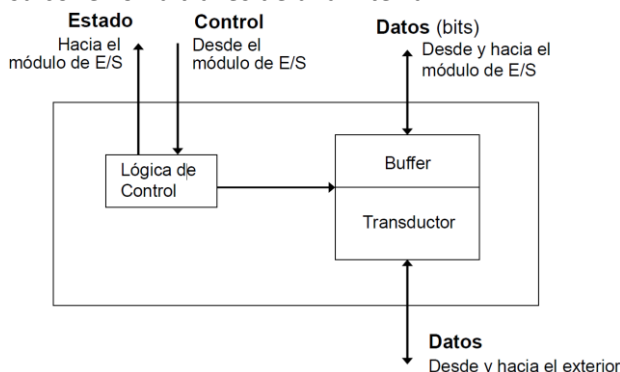
Interfaz: La interfaz es la definición estructural que permite al procesador conectarse con un periférico. Está compuesta por dos elementos:

1. Hardware: permite conectar el puerto con el periférico y se conecta al bus del sistema de la computadora para comunicarse con la memoria y el procesador.
2. Drivers: son conjuntos de programas que gestionan y administran el uso de la interfaz, normalmente asociados a la existencia de un sistema operativo (SO). Son aplicables a cualquier tipo de periférico.

Periféricos: Estos dispositivos se conectan a la interfaz a través del puerto. Ejemplos de periféricos incluyen sensores, alarmas y sistemas de adquisición de datos.

Dispositivos Externos y Puerto:

Dispositivo externo tipo e Interfaz: El dispositivo externo tipo es un esquema genérico para evaluar dispositivos externos, como impresoras, parlantes, teclados y discos rígidos, y facilitar su conexión a través de una interfaz.



Elementos de interacción en dispositivos externos:

1. Zona de datos: transferencia de información medida en bits entre el módulo y el periférico.
2. Zona de control: señales de control enviadas por el módulo de E/S para determinar el flujo de información.
3. Zona de estado: información sobre el estado del dispositivo externo que puede ser recibida por el módulo de E/S.

Dispositivo externo/periférico: Incluye una lógica de control que administra la conexión con el periférico y puede recibir órdenes a través de señales de control o enviar información de estado del dispositivo hacia el módulo de E/S. La lógica de control también maneja la conversión de datos a través de un transductor y un sistema de almacenamiento de bits.

Características de un puerto:

- Interfaz entre el periférico y el módulo de E/S.
- Incluye señales de Control, Estado y Datos.
 - Señal de Control: define la función a realizar, como INPUT/READ o OUTPUT/WRITE.
 - Señal de Estado: indica si el dispositivo está READY o NOT READY.
- Control lógico: maneja el direccionamiento.
- Transductor: realiza la conversión de datos.
- Buffer: adapta la información en 1, 8 o 16 bits.

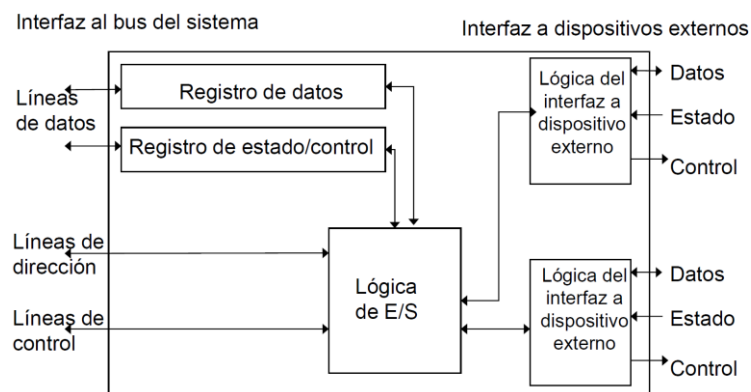
Tipos de dispositivos externos:

1. E/S básicos: monitor/pantalla, mouse, teclado.
2. Almacenamiento: disco duro, CD, DVD.
3. Impresión: impresora, escáner.
4. Comunicación con dispositivos remotos: modem, acceso/interfaz de red.
5. Multimedia: micrófono, parlantes.
6. Automatización y control: sensores, alarmas, adquisición de datos.

Resumen: Funciones y capacidades de un módulo de E/S

Funciones de un módulo de E/S:

- a. **Control y temporización:** Ajuste de velocidad en intercambio de información entre dispositivos externos.
- b. **Interpretación de órdenes:** Actúa como intermediario entre CPU y periféricos.
- c. **Comunicación:** Conecta **CPU y memoria** con dispositivos periféricos.
- d. **Control de transferencias:** Conversión de formatos y adaptación de velocidades.
- e. **Informar estado del periférico:** Comunica el estado del periférico a la CPU.
- f. **Almacenamiento temporal (buffering):** Almacena datos temporalmente entre periféricos y CPU.
- g. **Detección de errores:** Identifica errores pero no su ubicación.



Capacidades de un módulo de E/S:

- a. **Ocultar propiedades del dispositivo:** La CPU ve todos los periféricos de manera uniforme.
- b. **Controlar uno o varios dispositivos:** El módulo de E/S puede manejar múltiples dispositivos externos.
- c. **Variedad en inteligencia:** Desde procesadores de E/S en mainframes hasta controladores de E/S en microcomputadoras.

Componentes internos de un módulo de E/S:

- a. **Lógica de interfaz:** Permite la comunicación entre dispositivos externos y el módulo de E/S.
- b. **Registros:** Almacenamiento de datos, registro de estado y control del módulo de E/S.
- c. **Conexión al bus del sistema:** Los registros se conectan a las líneas del bus de datos para recibir y transmitir información.
- d. **Control de lectura y escritura:** La CPU controla la lectura y escritura en los registros a través de las líneas del bus del sistema.

Adaptación a diferentes dispositivos:

- a. **Procesadores de E/S:** Generan acciones en función de órdenes de la CPU y facilitan operaciones en sistemas con alta carga de almacenamiento.
- b. **Controladores de E/S:** Dispositivos de E/S primitivos con capacidades básicas, presentes en computadoras con diferentes capacidades de procesamiento.

En resumen, un módulo de E/S es fundamental en la comunicación entre la CPU y los dispositivos periféricos, permitiendo el control y la adaptación de velocidades y formatos. Los módulos de E/S pueden variar en inteligencia y capacidad, y cuentan con componentes internos como lógicas de interfaz y registros para gestionar la comunicación y el almacenamiento de datos.

Pasos de las operaciones de E/S:

- A. **Establecer interlocutores de la operación (direccionamiento):** Identificar dispositivos para la comunicación.
- B. **Proceso de transferencia de información:** Lectura o escritura dependiendo de si la operación es de entrada o salida (desde el punto de vista de la CPU).
- C. **Gestión de transferencia:** Mecanismos para sincronizar y controlar la transferencia correcta de datos entre la CPU, memoria y dispositivos periféricos.

Direccionamiento de E/S:

A. E/S mapeada en memoria:

- 1. Memoria principal y dispositivos de E/S comparten un mismo espacio de direcciones.
- 2. No se necesitan instrucciones específicas para E/S.
- 3. Mayor rapidez y eficiencia en la transferencia de datos.

B. E/S aislada:

- 1. Espacios de direcciones separados para memoria y dispositivos de E/S.
- 2. Se requieren líneas e instrucciones específicas para E/S (ejemplo: IN - OUT).
- 3. Lecturas o escrituras en periféricos se realizan con señales del bus de control diferentes a las de memoria.

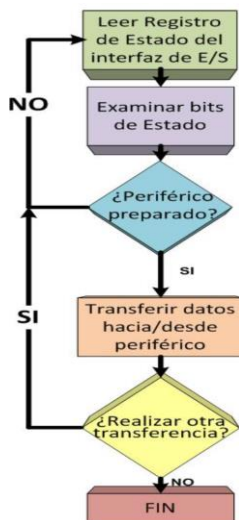
En resumen, las operaciones de E/S constan de tres pasos principales: direccionamiento, transferencia de información y gestión de transferencia. Existen dos métodos de direccionamiento: E/S mapeada en memoria y E/S aislada. Cada método tiene sus propias

características, ventajas y desventajas en términos de eficiencia, velocidad y requerimientos de instrucciones específicas.

Técnicas de gestión de E/S

E/S Programada:

En este método, la CPU tiene el control total de la operación de E/S, lo que incluye comprobar el estado del dispositivo, enviar comandos de lectura/escritura y realizar la transferencia de datos. La E/S programada sigue un enfoque síncrono, lo que significa que la CPU debe esperar a que el módulo de E/S complete la operación antes de continuar con otras tareas. Aunque esta técnica es simple de implementar, puede provocar una utilización ineficiente de los recursos de la CPU, ya que permanece inactiva durante las operaciones de E/S.



E/S con Interrupciones:

Para superar las limitaciones de la E/S programada, se introduce la E/S con interrupciones. Este enfoque permite que la CPU realice otras tareas en lugar de esperar la finalización de la operación de E/S. Cuando el módulo de E/S está listo para la transferencia de datos, envía una solicitud de interrupción a la CPU. La CPU, a su vez, guarda el contexto actual, interrumpe el proceso en curso y atiende la interrupción gestionando la operación de E/S. Este enfoque mejora la eficiencia, pero aún requiere la intervención directa de la CPU para manejar las transferencias de datos.

Para abordar los desafíos de identificar el módulo que solicita la interrupción y gestionar múltiples interrupciones, se emplean estrategias como la consulta de software (polling), la conexión en cadena (daisy chain) y el establecimiento de prioridades entre líneas de interrupción. La implementación de estas estrategias depende de la arquitectura de la computadora y el sistema de interrupción utilizado.

E/S con Acceso Directo a Memoria (DMA):

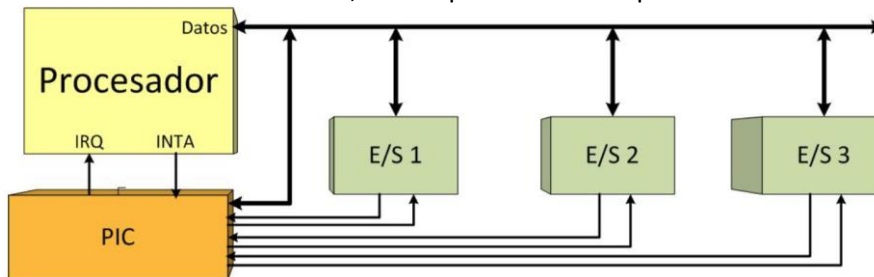
La E/S con DMA es una técnica avanzada que permite transferir grandes volúmenes de datos entre dispositivos y la memoria principal sin la intervención directa de la CPU. En lugar de involucrar a la CPU en la transferencia de datos, un controlador de DMA realiza la operación de E/S en su lugar. Esto libera a la CPU para realizar otras tareas y aumenta la velocidad de transferencia de datos.

En resumen, las técnicas de gestión de E/S son fundamentales para garantizar una comunicación eficiente entre la CPU y los dispositivos periféricos. La E/S programada es

adecuada para operaciones simples y directas, pero puede resultar ineficiente en términos de utilización de recursos. La E/S con interrupciones mejora la eficiencia al permitir que la CPU realice otras tareas simultáneamente, mientras que la E/S con DMA es óptima para transferir grandes volúmenes de datos sin sobrecargar la CPU. La elección de la técnica adecuada depende de las necesidades y restricciones específicas del sistema en cuestión.

Interrupciones múltiples

- Todas las líneas de interrupción tienen un orden de prioridad
- Las líneas con más prioridad pueden interrumpir a las líneas con menor prioridad.
- Si existe un maestro del bus, solo él puede interrumpir.



E/S con acceso directo a memoria (DMA)

Necesidad de un nuevo mecanismo de gestión de periféricos

- Periféricos rápidos y con alto volumen de transferencia de datos
- Problemas con E/S mediante interrupciones
- Evitar la intervención de la CPU en el proceso de transferencia

Características del DMA

- Dispositivo que realiza transferencias de datos entre memoria y E/S sin intervención de la CPU
- Programado previamente por el procesador
- Uso del bus del sistema y lógica de arbitraje para llevar y traer datos
- Generación de señales de control y sincronización de la transferencia

Controlador DMA (DMAC)

- Módulo de E/S específico para periféricos con alto volumen de transferencia de datos
- Capaz de manejar varios dispositivos periféricos de transferencia rápida
- Canales de DMA para transferir información de forma "simultánea"

Estructura del DMAC

- Líneas de sincronización, arbitraje, interrupción y lectura/escritura
- Conexión con módulo de E/S explícito para coordinación de acciones

Componentes internos del controlador DMA

- Registros para dirección de memoria, cantidad de palabras a transferir y sentido de transferencia
- Lógica de decodificación y control
- Incremento/decremento de registros de direcciones de memoria y número de palabras durante transferencia

Controlador DMA (DMAC) como maestro del bus

- Solicita el uso del bus mediante señales y lógica de arbitraje
- Especifica la dirección de memoria para la transferencia
- Genera señales de control del bus, incluyendo tipo de operación (lectura/escritura) y señales de sincronización

Transferencia DMA: Proceso y Etapas

1. Inicialización de la transferencia

- La CPU envía parámetros de transferencia al interfaz del periférico y al DMAC

2. Inicialización del interfaz (Bus master: CPU, Bus slave: Interfaz)

- Nº de bytes a transferir
- Tipo de transferencia (lectura/escritura)
- Otra información de control (pista, sector, etc.)

3. Inicialización controlador DMA (Bus master: CPU, Bus slave: DMAC)

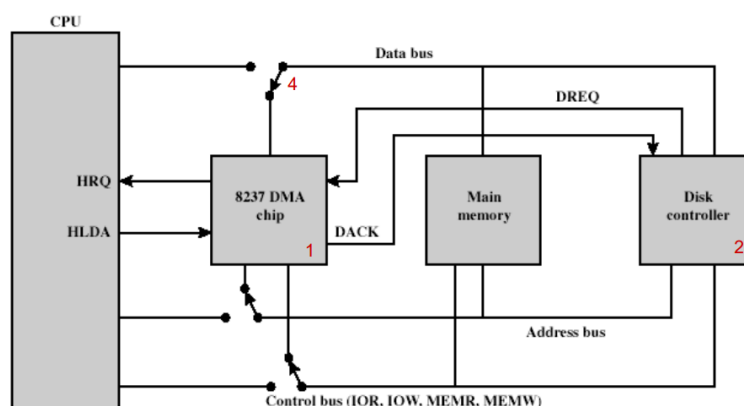
- Nº de bytes o palabras a transferir
- Tipo de transferencia (lectura/escritura)
- Dirección de memoria inicial para la transferencia
- Nº de canal (para DMAs con varios canales)
- La CPU retorna a sus tareas y no se preocupa más de la evolución de la transferencia

4. Realización de la transferencia

- Periférico indica al DMAC que está listo para realizar la transferencia
- DMAC pide el control del bus y se realiza la transferencia entre el periférico y la memoria
 - Bus master: DMAC + Periférico, Bus slave: Memoria
 - Actualización de registros del DMAC después de la transferencia de cada palabra
 - Nº de bytes o palabras a transferir
 - Dirección de memoria

5. Finalización de la transferencia

- DMAC libera el bus y devuelve el control a la CPU
- DMAC activa una señal de interrupción para indicar a la CPU la finalización de la operación de E/S solicitada



1. chip → desarrollado para ser controlador de acceso directo a memoria
2. controlador de disco → típico ej de alta velocidad, alto volumen de transferencia de datos
3. señales que muestran cómo funciona el sistema cuando el DMA está activo
4. llaves → hacen que la CPU no esté conectada al bus de control, datos y direcciones

Rendimiento de la CPU y DMA: Problemas y Soluciones

1. Problema: Degradación del rendimiento de la CPU

- La CPU no puede acceder a la memoria ni al controlador de disco durante la transferencia DMA
- Si el bus está ocupado en una transferencia DMA, la CPU no puede leer instrucciones ni datos

2. Solución: Uso de memoria caché

- La memoria caché permite a la CPU leer instrucciones sin acceder a la memoria principal ocupada por el DMA
- El DMAC aprovecha los intervalos en los que la CPU está leyendo instrucciones de la cache y no utiliza el bus de memoria

3. Solución para computadores sin memoria caché: Aprovechamiento de fases de ejecución de instrucciones

- El procesador no utiliza el bus en todas las fases de la ejecución de una instrucción
- El DMAC aprovecha las fases de ejecución de una instrucción en las que la CPU no utiliza el bus para realizar sus transferencias
- Requiere un análisis minucioso del ciclo de instrucción para no disminuir el rendimiento del sistema

Tipos de Transferencias DMA

1. Transferencia por ráfagas (Burst)

- El DMAC utiliza el bus hasta terminar la transferencia de todos los datos
- Ventajas:
 - Transferencia rápida
- Desventajas:
 - La CPU no puede usar el bus durante la transferencia, lo que puede degradar el rendimiento del sistema

2. Transferencia por robo de ciclo (Cycle-stealing)

- El DMAC transfiere una única palabra y libera el bus
- Se realizan múltiples solicitudes de control del bus hasta finalizar la transferencia del bloque completo
- Ventajas:
 - No se degrada el rendimiento del sistema
 - La CPU no se entera del uso de la memoria, no es una interrupción
- Desventajas:
 - La transferencia tarda más tiempo en llevarse a cabo
- Requiere alta sincronización entre el DMAC y la CPU

Comparación

- El DMA por robo de ciclo es el modo más rápido y común en la mayoría de las máquinas
- Es la técnica más eficiente para transferencias de E/S de múltiples palabras

La evolución tecnológica ha llevado a dispositivos de E/S cada vez más sofisticados, y el DMA es una de las optimizaciones en la gestión de E/S para transferencia de datos.

Evolución CPU vs Periféricos

1. **Control directo de la CPU:** La CPU controla directamente los periféricos por programa.
2. **Módulo de E/S:** Se agrega un módulo de E/S o controlador para trabajar con el periférico.
3. **Interrupciones:** Se incorpora el llamado de interrupción al módulo de E/S.
4. **Acceso Directo a Memoria (DMA):** El módulo de E/S provee DMA para transferencias de datos.

5. **Procesador propio:** El módulo de E/S tiene su propio procesador y conjunto de instrucciones propias.
6. **Computadora en sí mismo:** El módulo tiene memoria local y se convierte en una computadora en sí misma, como las tarjetas gráficas 3D.

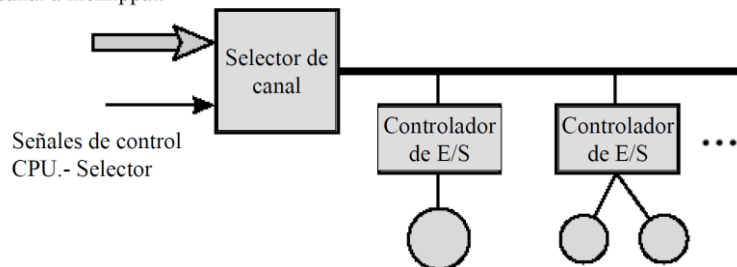
Canales de E/S

- Extensión al concepto de DMA
- Ejecutan instrucciones de E/S
- Control completo de la transferencia de datos
- La CPU no ejecuta instrucciones de E/S
- Programa almacenado en memoria principal
- La CPU inicia la transferencia de E/S
- El programa especifica dispositivos, áreas de memoria a usar, prioridades y acciones ante errores

Tipos de Canales de E/S

1. **Selector:** Controla varios dispositivos de alta velocidad, uno por vez. El canal se dedica para la transferencia de datos de ese dispositivo.

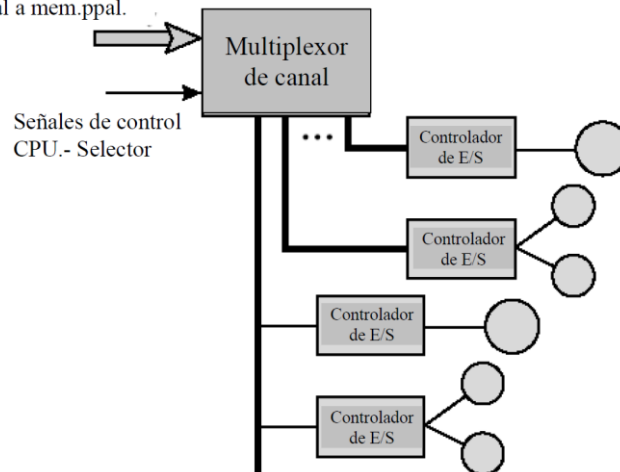
Datos y direcciones
canal a mem.ppal.



2. **Multiplexor:** Puede manejar E/S con varios dispositivos a la vez.

- **Multiplexor de bytes:** Acepta y transmite caracteres.
- **Multiplexor de bloques:** Intercala bloques de datos desde distintos dispositivos.

Datos y direcciones
canal a mem.ppal.



Conceptos de Arquitectura de Computadoras Evaluación Corta de Teoría 1 – 27/04/21		
1	La Arquitectura Von Neumann establece que la CPU de una computadora posee	
	A	Unidad de Memoria, Unidades de Entrada y Salida y un sistema de interconexión
	B	Unidad Aritmético-Lógica, Unidad de Control, Registros, Unidades de Entrada y Salida
	C	Unidad Aritmético-Lógica, Unidad de Control y Registros
	D	Unidad de Memoria, Unidad Aritmético-Lógica, Unidad de Control y Registros Unidades de Entrada y Salida y un sistema de interconexión
2	Las instrucciones que puede interpretar la unidad de control de una máquina de dos direcciones deben tener en el formato de su código binario	
	A	Operación a realizar, direcciones a dos operandos fuente, dirección a un operando resultado y dirección de la próxima instrucción.
	B	Operación a realizar, direcciones a dos operandos fuente y dirección de la próxima instrucción.
	C	Operación a realizar, dirección a un operando fuente/resultado y dirección a otro operando fuente.
	D	Operación a realizar, dirección a un operando fuente/resultado y dirección de la próxima instrucción.
3	Una subrutina es	
	A	Una secuencia de instrucciones autocontenida que comienza con un rótulo y terminan con un salto incondicional
	B	Una secuencia de instrucciones autocontenida que se invoca con una instrucción de salto con retorno y terminan con una instrucción de retorno
	C	Una secuencia de instrucciones autocontenida que se invoca con una instrucción de salto condicional y terminan con un salto incondicional
	D	Una secuencia de instrucciones autocontenida que se invoca con una instrucción de salto incondicional y terminan con una instrucción de retorno
4	La secuencia de funcionamiento de la operación apilar en una estructura de Pila es	
	A	incrementar el puntero al tope de pila (SP) y luego mover un dato de un registro a una dirección de memoria
	B	mover un dato de un registro a una dirección de memoria y luego incrementar el puntero al tope de pila (SP)
	C	mover un dato de un registro a una dirección de memoria y luego decrementar el puntero al tope de pila (SP)
	D	decrementar el puntero al tope de pila (SP) y luego mover un dato de un registro a una dirección de memoria
5	El Controlador Programable de Interrupciones –PIC- es	
	A	Un dispositivo electrónico utilizado para recibir pedidos de distintas fuentes de interrupción por hardware y presentarlas a una línea de entrada a la CPU.
	B	Un dispositivo electrónico utilizado para responder a pedidos de distintas fuentes de interrupción por hardware sin intervención de la CPU.
	C	Un dispositivo electrónico utilizado para recibir pedidos de distintas fuentes de interrupción por hardware y software y presentarlas a una línea de entrada a la CPU.
	D	Un dispositivo electrónico utilizado para responder a pedidos de distintas fuentes de interrupción por software y presentarlas a una línea de entrada a la CPU.
6	Una interrupción por software es	
	A	Un pedido del Sistema Operativo de un gestor de interrupción específico
	B	Una instrucción del repertorio de instrucciones que tiene como operando un número de vector de interrupción específico
	C	Un pedido del PIC de un gestor de interrupción específico
	D	Un pedido del CDMA de un gestor de interrupción específico
7	¿Dónde se encuentra el gestor de una interrupción por hardware? en:	
	A	El PIC
	B	La CPU
	C	La Memoria Principal
	D	La Pila
8	El módulo de E/S incluye:	
	A	al menos un transductor.
	B	registros para almacenamiento temporal de datos.
	C	registro que permite definir el sentido de la transferencia de datos.
	D	registros para cargar los vectores de interrupciones.
9	La técnica de E/S mediante el uso del controlador de DMA es:	
	A	siempre la mejor opción.
	B	conveniente cuando tenemos periféricos rápidos y gran volumen de datos.
	C	la mejor opción cuando tenemos periféricos lentos y pequeño volumen de datos.
	D	una técnica que ya no se usa hoy en día.
10	Los canales de E/S:	
	A	son una extensión del concepto de DMA.
	B	son utilizados en las máquinas de escritorio actuales.
	C	no tienen la capacidad de ejecutar instrucciones de E/S.
	D	no son usados ya que el procesador controla directamente los periféricos.

1. **D** Unidad de Memoria, Unidad Aritmético-Lógica, Unidad de Control y Registros Unidades de Entrada y Salida y un sistema de interconexión
2. **C** Operación a realizar, dirección a un operando fuente/resultado y dirección a otro operando fuente.
3. **B** Una secuencia de instrucciones autocontenida que se invoca con una instrucción de salto con retorno y terminan con una instrucción de retorno
4. **D** decrementar el puntero al tope de pila (SP) y luego mover un dato de un registro a una dirección de memoria
5. **A** Un dispositivo electrónico utilizado para recibir pedidos de distintas fuentes de interrupción por hardware y presentarlas a una línea de entrada a la CPU.
6. **B** Una instrucción del repertorio de instrucciones que tiene como operando un número de vector de interrupción específico
7. **C** La Memoria Principal
8. **A** al menos un transductor, B registros para almacenamiento temporal de datos, y C registro que permite definir el sentido de la transferencia de datos.
9. **B** conveniente cuando tenemos periféricos rápidos y gran volumen de datos.
10. **A** son una extensión del concepto de DMA.

Evaluación Teórica Corta - Arquitectura de Computadoras – Tema 1

1) Los canales de E/S:

- a) son utilizados en las máquinas de escritorio actuales.
- b) no tienen la capacidad de ejecutar instrucciones de E/S.
- c) son una extensión del concepto de DMA.
- d) no son usados ya que los procesadores actualmente controlan directamente los periféricos.

2) Entre las funciones que realiza el PIC una de ellas es:

- a) pasarle al procesador la dirección del gestor.
- b) poner en el bus de datos el vector (identificador) correspondiente al periférico que generó el pedido de interrupción.
- c) atender las interrupciones por software de la misma manera que lo hace con las de hardware.
- d) pedirle el uso del bus al procesador para atender el pedido de interrupción.

3) Según lo que hemos visto, en la arquitectura CISC el método de pasaje de argumentos a subrutinas más ampliamente usado es mediante el uso de:

- a) los registros del procesador.
- b) la pila.
- c) la memoria principal.
- d) la memoria virtual.

4) Para no degradar el rendimiento del sistema en el caso de usar DMA se prefiere el método:

- a) por ráfagas.
- b) por chequeo del registro de estado de los módulos de E/S.
- c) por uso de la pila como espacio de almacenamiento.
- d) por robo de ciclo.

5) ¿Dónde se encuentra el gestor de una interrupción por hardware? en:

- a) el PIC.
- b) el procesador.
- c) la memoria principal.
- d) la pila.

6) Si se compara la técnica de E/S programada frente a la que usa interrupciones se obtiene:

- a) que ambas presentan los mismos comportamientos respecto al tiempo de uso del procesador y la velocidad de transferencia de datos.
- b) mayor performance en la primera ya que en ella participa el procesador en la transferencia de datos y en la otra no.
- c) un mal uso de los tiempos del procesador, pero transferencias más rápidas.
- d) un óptimo uso del procesador, pero transferencia de datos más lenta.**

7) La técnica de E/S mediante el uso del controlador de DMA es:

- a) la mejor opción cuando tenemos periféricos lentos y pequeño volumen de datos.
- b) siempre la mejor opción.
- c) conveniente cuando tenemos periféricos rápidos y gran volumen de datos.
- d) una técnica que ya no se usa hoy en día.

8) ¿Qué significa que una interrupción sea no enmascarable?

- a) que es de alta prioridad y no puede ser ignorada por el procesador.**
- b) que puede ser ignorada pero por un cierto tiempo previamente estipulado según su importancia.
- c) que no puede ser procesada por el procesador.
- d) que puede ser ignorada por el procesador y atenderla cuando pueda.

9) En subrutinas, el acceso a parámetros haciendo uso de la pila se realiza mediante:

- a) la sentencia POP.
- b) el uso del modo de direccionamiento directo por memoria.
- c) el uso de la técnica de robo de ciclo.
- d) el uso del modo de direccionamiento indirecto por desplazamiento.**

10) El módulo de E/S incluye:

- a) al menos un transductor.
- b) registros que indican el sentido (entrada/salida) de la transferencia de datos.
- c) registros para almacenamiento temporal de datos.**
- d) registros para cargar los vectores de interrupciones.