PARCIAL I ARQUITECTURA DE COMPUTADORAS

1. OBJETIVOS

* Comprender cómo la arquitectura de computadoras afecta al diseño y rendimiento de programas.
* Aplicar la teoría de arquitectura de computadoras para hacer programas más eficientes.

1. IMPORTANCIA

* Permite realizar programas, más eficientes, aprovechando al máximo las fortalezas del sistema, y manteniendo a raya los puntos débiles.
* Permite la selección adecuada de herramientas y/o sistemas dependiendo de la tarea a realizar.

**Ej.**

La diferencia está que en la decodificación por hardware traslada todo el procesamiento gráfico del CPU al GPU, dejando al primero libre para ejecutar otras tareas que no sean de índole gráfica.   
El GPU es como un CPU (procesador) pero especializado para el procesamiento de gráficos.   
O sea que cuando seleccionas la decodificación por hardware estás distribuyendo la carga de tareas de manera inteligente ya que cualquier algoritmo en hardware es más rápido que uno en software.

* **Ley de Moore:** La tecnología de semiconductores multiplicará su efectividad cada 18 meses.

1. ESQUEMA BASICO DE COMPUTADORA

DISPOSITIVOS DE SALIDA

Dispositivos de salida

Microprocesador

Dispositivos de entrada

MEMORIA

**Microprocesador:** Es el cerebro de la computadora. Tiene como función coordinar y controlar todas las operaciones que realiza el sistema; lee cada instrucción de los programas, las analiza y las ejecuta.

Puede:

* + Ejecutar una serie de instrucciones
  + Ejecutar un programa de computadora.

**Dispositivos de entrada:** Permiten introducir instrucciones y datos a la computadora. Convierten las señales externas en códigos binarios que son interpretados por el procesador.

**Dispositivos de salida:** Permiten mostrar la información procesada por la computadora.

**Memoria:** Se utiliza para almacenar datos e instrucciones. Es el espacio de almacenamiento en la computadora donde los datos van a ser procesados y se almacenan las instrucciones necesarias para el procesamiento.

**Bus:** Es un grupo de señales que tienen significado como grupo o realizan una función común.

Se encarga de comunicar los diferentes dispositivos entre sí con el microprocesador.

En un sistema de computadora se puede identificar tres buses:

* Address Bus: selecciona hacia dónde va la información y a qué posición de la memoria o dispositivo externo.
* Control Bus: indican que se hará con la información (lectura o escritura) y si es dato o programa.
* Data Bus: es bidireccional y es por donde viajan los datos o instrucciones.

1. PROCESAMIENTO

El computador trabaja con un lenguaje particular para procesar la información, conocido como código binario, en el que los números, letras e imágenes que recibe en forma de señales eléctricas, se transforman en series de ceros y uno. Para realizar este proceso, el computador está formado por dos partes: una tangible o hardware (la maquinaria y los circuitos) y una parte no tangible denominada software, que son los programas que se instalan en el computador, incluido su sistema operativo. Este permite su encendido y es la plataforma sobre la que trabaja el resto de los programas o aplicaciones instaladas.

El nexo entre ambas partes es el microprocesador o CPU (Unidad Central de Proceso), que controla todas las funciones del computador.

El proceso es el siguiente:

– La información es recibida a través del teclado, un CD, un disquette o una unidad de comunicación que lo conecta a Internet o a otros computadores, con los que se encuentre vinculado mediante una red.

– Luego intervienen las memorias: la RAM, memoria interna o de acceso directo, que almacena los programas y las informaciones variables y tiene capacidad limitada, y la ROM, memoria externa o de solo lectura, que contiene las instrucciones permanentes (el sistema operativo y los programas) y es de acceso más lento.

– A continuación, la CPU recibe la información captada por la RAM y selecciona, entre los softwares almacenados en la ROM, el programa apropiado dependiendo del tipo de información de la que se trate (texto, imágenes, fotografía, audio, estadísticas, etc.).

Procesa la información mediante las instrucciones del programa escogido y muestra un resultado en la pantalla, que puede ser la solución de un cálculo o una modificación en un texto o imagen.

1. ARQUITECTURAS DE ACCESO

Definen la forma en que el microprocesador maneja la memoria. Pueden ser:

**Von Newman:**

* Mismo bus para la memoria de dato y programas.
* Misma memoria para almacenar datos y programas.
* No es muy costosa su implementación.
* Más lento el acceso, es decir, la lectura y la escritura de información.
* Expandidle.

**Harvard:**

* Buses separados para cada uno de los tipos de memoria.
* Memorias separadas para datos y para programa.
* Más costosa de implementar.
* Velocidad de acceso bastante alta, es decir, la lectura y la escritura de información.
* No es expandidle.

1. INSTRUCCIONES

Una instrucción es una operación o proceso que puede ejecutar el µP. La instrucción está formada por:

* **Opcode:** se utilizará para que el procesador sepa identificar la instrucción.
* **Pay/Load(Operandos):** valores directos o localización de un valor, contiene los valores a los cuales se va a hacer la instrucción

1. ARQUITECTURAS DE EJECUCCIÓN

CISC (Complex Instruction Set Computer)

* Principios
* Proveer apoyo a diferentes lenguajes de alto nivel moviendo funciones complejas al "hardware".
* Facilitar el desarrollo de compiladores cerrando el "semantic gap".
* Las arquitecturas CISC no siguen una filosofía en particular.
* El nombre de CISC se les adjudicó para distinguirlas de las arquitecturas RISC.
* Características de CISC
* Set de instrucciones expandible.
* Tamaño de la instrucción variable.
* Velocidades de ejecución diferentes.
* Se requieren por lo general varios ciclos para ejecutar una instrucción.
* Las instrucciones son decodificadas internamente por una serie de micro instrucciones que se encuentran almacenada en la memoria identificadas por un Micro-code.

RISC (Reduced Instruction Set Computer / Computing)

* Principios

Apoyar aquellas estructuras de lenguajes de alto nivel que son utilizadas con frecuencia y que tienen un impacto considerable en el tiempo de ejecución de programas.

El apoyo a estas estructuras debe basarse en elementos arquitecturales primitivos que faciliten la Implementación de un procesador rápido, eficiente y sencillo.

* Características de CISC
* Set de instrucciones fijo.
* Instrucciones del mismo tamaño.
* Alta velocidad de ejecución.
* Misma velocidad de ejecución para todas las instrucciones.
* No se puede expandir.
* Set de instrucciones reducido con instrucciones simples que permiten realizar instrucciones complejas. Un número grande de registros.
* Operaciones se llevan a cabo con operandos mantenidos en registros.
* Los accesos a memoria se realizan exclusivamente mediante instrucciones load/store.

1. PARTES DEL MICROPROCESADOR

**ALU:** unidad aritmético lógica, se encarga de realizar las operaciones lógicas y aritméticas.

**EXECU:** es la unidad de ejecución, se encarga de coordinar los procesos necesarios para que se ejecute una instrucción.

**AGU:** generador de direcciones, es un bloque de control de buses internos y externos que supervisa el proceso de transferencia de información dentro y fuera del µP.

**Register file:** es una memoria interna donde se almacenan los datos temporalmente.

1. ASPECTOS ARQUITECTURALES DE LA COMPUTADORA

* Instrucciones.
* Modos de acceso a operandos
* Formatos de instrucciones
* Tipos de datos
* Memoria Primaria
* Registros
* Interrupciones
* Flujo de ejecución de instrucciones
* Stacks

**Aspectos que podrían ser arquitecturales:**

* Caches
* Pipelines
* I/O

1. ASPECTOS NO ARQUITECTURALES DE LA COMPUTADORA

* Memoria secundaria
* Buses
* Periféricos
* Unidad de control
* ALU
* Tecnología de fabricación
* Circuitos lógicos

1. PIPELINING

"Pipelining", en el contexto de implementación de procesadores, es la idea de procesar instrucciones por etapas como en una línea de producción.

Idealmente las instrucciones entran a la primera etapa a una razón de una por ciclo y salen de la última etapa a una razón de una por ciclo.

Idealmente las instrucciones pasan de una etapa a otra al cabo de un ciclo.

El pipeline se divide en cuatro etapas:

1. **Fetch:**
   * Leer la instrucción de la memoria de programa.
2. **Decode:**
   * Decodificar la instrucción
3. **Read:**
   * Buscar o leer en la memoria de datos la información necesaria para ejecutar la instrucción
4. **Execute:**
   * Ejecutar la instrucción
5. TIPOS DE PIPELINE

**SIMPLE:** Todas las instrucciones se ejecutan a la misma velocidad.

**PRE-EJECUTABLE:** Mejora el rendimiento del microprocesador al mejorar la ejecución de las instrucciones, adelantando el procesamiento de las instrucciones siguientes, durante el trabajo de instrucciones previas a esta. La mejora se pierde cuando se realiza un salto (La instrucción que se ejecuta luego de un salto se ejecuta como si fuera la primera instrucción del programa).

**MULTIPLE:** Parecido al anterior, pero en el paso de decodificación verifica si la instrucción es un salto, y en caso de hacerlo avanza la instrucción a ejecutar después del salto y no la que está siguiente en la memoria.

1. PIPELINE HAZARDS

La razón principal para que el "pipeline" no tenga una operación ideal se debe a ”hazards".

"Pipeline hazards" son interdependencias entre instrucciones que no permiten que las instrucciones avancen en el "pipeline" a razón de una por ciclo.

Estos” hazards" afectan negativamente el rendimiento del procesador.

1. TIPOS DE HAZARDS

**Hazards estructurales** - surgen a consecuencias de conflictos de recursos limitados que son requeridos en un momento dado por más instrucciones de las que puede manejar.

**Hazards de datos** - surgen cuando una instrucción depende del resultado de otra para poder avanzar en el “pipeline”.

**Hazards de control** -surgen a consecuencia de instrucciones de brinco e instrucciones que cambian el program counter.

1. TIPOS DE CICLOS

**Ciclo de Reloj:**

Determina la frecuencia a la que puede hacer un cambio en la unidad de control (Baja, subida o ambas).

**Ciclo Máquina:**

Tiempo mínimo en que se puede ejecutar una instrucción. Cantidad de ciclos de reloj para ejecutar una instrucción (desde que inicia hasta que termina la ejecución).

**Ciclo de Bus:** Tiempo en que el bus realiza una operación.

Puede ser mayor o igual al ciclo de reloj.

1. TIEMPOS DE EJECUCIÓN

**Tiempo real de ejecución:**

Tiempo que tarda en ejecutarse una instrucción desde que entra al Pipeline.

**Tiempo aparente:**

Tiempo en que tarda en terminar de ejecutarse una instrucción con respecto a otra

1. MECANISMOS PARA REDUCIR LOS INTERLOCKS

**Superscalar:**

* Múltiples unidades funcionales (suma, multiplicación división) que pueden operar en paralelo.
* Puede producir más de una instrucción por ciclo.
* Instrucciones que requieren varios ciclos de ejecución no detienen otras instrucciones.

**Superpipeline:**

Muchas etapas de pipeline que permite reducir el periodo del reloj del sistema.

1. REGISTROS

Los registros del procesador se emplean para controlar instrucciones en ejecución, manejar direccionamiento de memoria y proporcionar capacidad aritmética. Los registros son espacios físicos dentro del microprocesador con capacidad de 4 bits hasta 64 bits dependiendo del microprocesador que se emplee.

**-Registros Uso General**

Almacenar las instrucciones. Registros de 16 bits. Se puede acceder a las partes

ax-->Acumulador--> Ah (8 bits,high)/Al(8 bits,low).Siempre se utiliza Ax como el operando principal.

bx-->Base

cx-->Contador

dx-->Datos

Ej:

Ax=FD3AH

AH=FDH

AL=3AH

MOV Al,05

Ax=FD05H

**-Registros de Índice**

Es como un puntero. Apunta a algún sitio

si} No se puede acceder en pedazos como los registros de uso general

di}

bp} Índices propios de acceso a la pila o stack

sp}

**-Registros de Segmento:** Acceder a porciones de memoria de 64 kbs. Es como si fuera un arreglo Dir\_absoluta: Segmento\*16\*Desp

CS-->Código, para acceder al programa. (No usar)

DS-->Datos. (No usar)

ES-->Extra

SS-->Stack o pila. (No usar)

**-Registros de Bandera:**

Los registros de banderas sirven parar indicar el estado actual de la maquina y el resultado del procesamiento, Cuando algunas instrucciones piden comparaciones o cálculos aritméticos cambian el estado de las banderas.

Las banderas más comunes son las siguientes:

* **OF (Over flow flag, desbordamiento).**

Indica el desbordamiento de un bit de orden alto (mas a la izquierda) después de una operación aritmética.

* **DF (Direction flag, Direccion).**

Designa la dirección hacia la izquierda o hacia la derecha para mover o comparar cadenas de caracteres.

* **IF (Interruption flag, Interrupcion).**

Indica que una interrupción externa, como la entrada desde el teclado sea procesada o ignorada.

* **TF (Trap flag, Trampa).**

Examina el efecto de una instrucción sobre los registros y la memoria. Los programas depuradores como DEBUG, activan esta bandera de manera que pueda avanzar en la ejecución de una sola interrupción a un tiempo.

* **SF (Sign flag, Signo).**

Contiene el signo resultante de una operación aritmética (0=positivo y 1= negativo).

* **ZF (Zero flag, Zero).**

Indica el resultado de una operación aritmética o de comparación (0= resultado diferente de cero y 1=resultado igual a cero).

* **AF (Auxiliary carry flag, Acarreo auxiliar).**

Contiene un acarreo externo del bit 3 en un dato de 8 bits, para aritmética especializada

* **PF (Parity flag, Paridad).**

Indica paridad par o impar de una operación en datos de ocho bits de bajo orden (mas a la derecha).

* **CF (Carry flag, Acarreo).**

Contiene el acarreo de orden mas alto (mas a la izquierda) después de una operación aritmética; también lleva el contenido del ultimo bit en una operación de corrimiento o rotación.