

S5-ComputadorDigitalRendimiento

Lenin G. Falconí

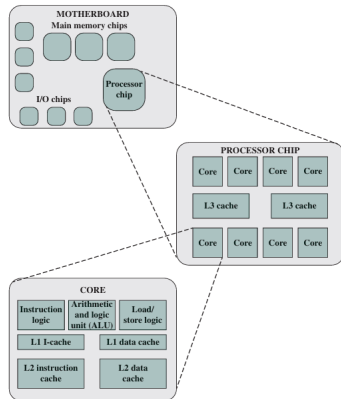
2024-05-21

1 Evolución Computador Digital

- Estructura Multi-Núcleo
- Computador IAS
- Instruction Set Architecture
- Evolución del Computador y Ley de Moore
- Tipos de Computadores
- Medidas de Desempeño
- Ley de Amdahl
- CPU Time
- Instruction Execution Rate CPI

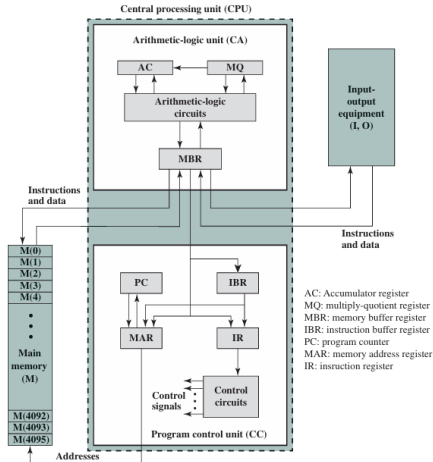
Estructura Multi-Núcleo I

- Los computadores actuales disponen de *múltiples procesadores*
- Una unidad de procesamiento tiene su unidad de control, ALU, registros y tal vez *cache*
- **Core (Núcleo)**: es una unidad de procesamiento individual en un chip. Es equivalente al CPU en un sistema de un sólo CPU
- CPU: parte del computador encargada de **captar (i.e. fetch)** y **ejecutar** instrucciones. Contiene la ALU, registros, unidad de control



Computador IAS(Institute for Advanced Study) I

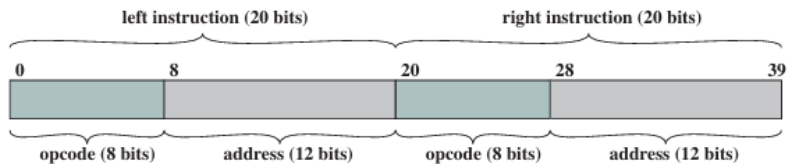
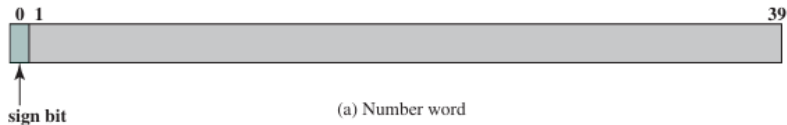
- La idea de diseño:
stored-program concept
- Idea de diseño de Von Neuman
- Consiste de una estructura conformada por:
 - Memoria Principal: guarda los datos y las instrucciones
 - ALU: operaciones binarias sobre los datos
 - Unidad de control: interpreta las instrucciones de la memoria y las ejecuta
 - E/S: interfaz de entradas y salidas
 - Bus de datos para comunicar CPU y Memoria



Memoria del Computador IAS I

- 4096 localidades denominadas *words*
- Cada *word* es de 40 bits
- Contiene tanto los datos como las instrucciones
- Los números usan 1 bit para el signo
- Cada *word* se subdivide en 2 instrucciones de 20 bits
- La instrucción se divide en 8 bits de /operation code/(opcode): especifica la operación
- La instrucción se divide en 12 bits que contiene la dirección en donde se aloja la instrucción o el dato

Memoria del Computador IAS



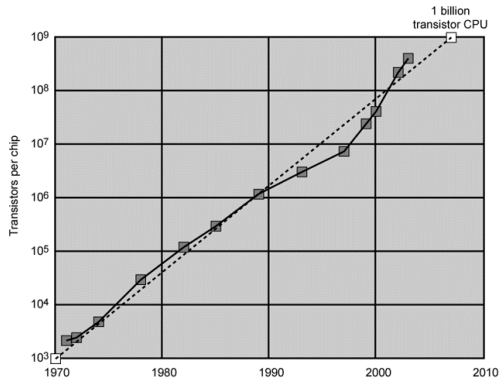
Instruction Set Architecture (ISA) I

Define:

- Formato de Instrucciones
- Opcodes
- Registros
- Memoria e instrucciones
- El efecto de la ejecución de instrucciones sobre los registros
- El algoritmo que controla la ejecución de las instrucciones
- Se propende a que el ISA sea compatible con versiones anteriores (i.e. un programa escrito en versiones anteriores debe ser ejecutable en versiones nuevas)
- El incremento de la densidad de transistores permite ISA más complejos

Evolución del Computador y Ley de Moore

- El número de transistores en un chip se duplica cada año
- En 1970 el ritmo se ralentiza a cada 18 meses
- El costo de un chip se ha mantenido sin cambios
- Una densidad mayor de componentes implica caminos eléctricos más cortos, incrementando el desempeño



Tipos de Computadores:

- Intel X86
- Arm
- Sistemas Embebidos

Ley de Amdahl

- Trata de medir las potenciales ganancias en velocidad al pasar de un programa que usa procesadores múltiples comparados con procesadores sencillos:

$$Speedup = \frac{TiempoSP}{TiempoNPP} \quad (1)$$

$$Speedup = \frac{1}{(1 - f) + \frac{f}{N}} \quad (2)$$

donde f es la fracción que puede ser paralelizada y N el número de procesadores

- El incremento de velocidad de un programa utilizando múltiples procesadores está limitado por la fracción del programa que no se puede paralelizar.
- Por ejemplo, si el 50% de un programa es secuencial, el incremento máximo teórico de velocidad, sin importar cuántos procesadores (N) se añadan, será de 2 veces.

Velocidad del Reloj I

- El computador usa un reloj para ejecutar sus operaciones (e.g. captación, decodificación, aritmética, etc.)
- Son señales sinusoidales generadas por un reloj de cuarzo que son digitalizadas en tren de pulsos.
- La velocidad de un computador está determinada por la frecuencia del reloj en Hertz (Hz)
- **Tiempo de ciclo:** El tiempo entre pulsos del reloj
- La mayoría de instrucciones en un computador requieren de varios **ciclos del reloj** para completarse
- Cuando las instrucciones se ejecutan en **pipelining**, múltiples instrucciones se ejecutan simultáneamente.

- La comparación de relojes entre dos máquinas M_1 y M_2 no es suficiente para evaluar el desempeño.

$$\tau = \frac{1}{f} \quad (3)$$

$$CT = \frac{1}{f} \quad (4)$$

- CT: tiempo de ciclo y f frecuencia del reloj
- CC: número de ciclos de reloj que la CPU usa para ejecutar una tarea
- I_c : número de ejecuciones de instrucciones

Se obtiene como el producto del número de ciclos CC y el tiempo de ciclo CT

$$CPU_T = CC \times CT \quad (5)$$

Sin embargo, no es práctico realizar la cuenta exacta de ciclos de reloj que usa un programa al realizar una tarea. Por esta razón se prefiere usar el **Promedio de ciclos de reloj por Instrucción** o **Instruction Execution Rate** CPI

Instruction Execution Rate CPI I

- Si todas las instrucciones utilizara el mismo número de ciclos, CPI sería una constante del computador.
- El número de ciclos del reloj varía dependiendo de la instrucción.
- Sea CPI_i el número de ciclos requeridos para ejecutar una instrucción tipo i , y sea I_i el número de instrucciones ejecutadas de tipo i , entonces el **promedio de ciclos de reloj por instrucción** es:

$$CPI = \frac{\sum_{i=1}^n (CPI_i \times I_i)}{I_c} \quad (6)$$

- El tiempo T que un procesador requiere para ejecutar un programa se puede expresar como:

$$T = I_c \times CPI \times \tau \quad (7)$$

- **MIPs** corresponde a la expresión de la tasa de ejecución de instrucciones

expresada en millones de instrucciones por segundo

$$MIPs_{rate} = \frac{I_c}{T \times 10^6} = \frac{f}{CPI \times 10^6} \quad (8)$$

- **MFlops** corresponde a la medida de desempeño relacionada únicamente con instrucciones de punto flotante.

$$MFLOPS_{rate} = \frac{\text{Número de operaciones de punto flotante}}{T \times 10^6} \quad (9)$$

Ejemplo I

La ejecución de un programa resulta en la ejecución de 2 millones de instrucciones en un procesador de 400 MHz. El programa consiste de 4 tipos principales de instrucciones como se indica en la Tabla. Obtenga el *CPI* promedio y la tasa de MIPS

Tipo de Instrucción	CPI	Mix de Instrucciones
Aritmética y Lógica	1	60%
Carga de dato	2	18%
Salto	4	12%
Referencia Memoria	8	10%

$$CPI = 1 \times 0.6 + 2 \times 0.18 + 4 \times 0.12 + 8 \times 0.1 = 2.24$$

$$MIPS = \frac{400 \times 10^6}{2.24 \times 10^6} \approx 178$$