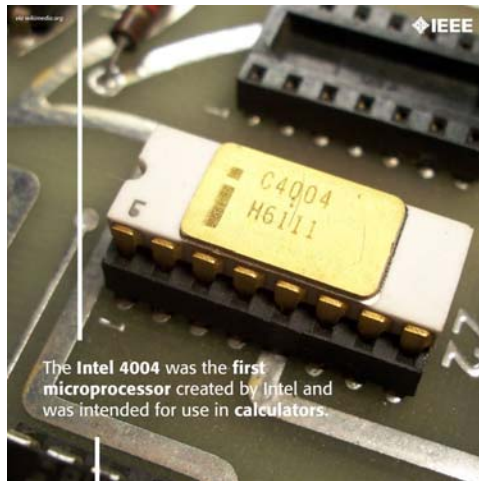


# Arquitectura de computadoras y rendimiento de sistemas



Arquitectura de Computadoras I

Prof. Marcelo Tosini  
Prof. Elías Todorovich  
2017



Arqui1-UNICEN

## Introducción

### ¿Qué se entiende por arquitectura de un computador?

- ❑ Los atributos del sistema que ve el programador
  - ❑ La estructura conceptual
  - ❑ El comportamiento funcional
  - ❑ ISA (*Instruction Set Architecture*) es la interfaz entre el hardware y el software de bajo nivel de un sistema.
- ❑ El diseño lógico y su realización física

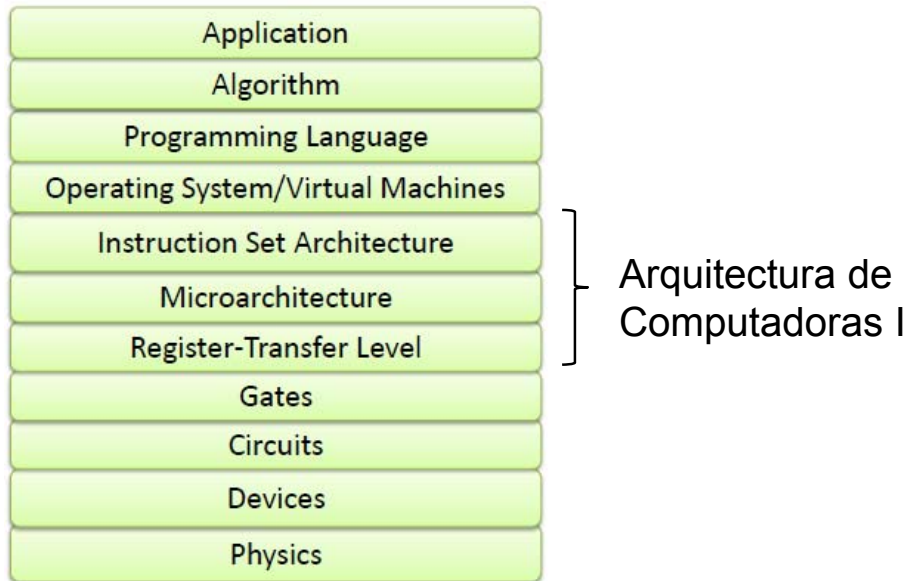
### ¿Qué se estudia en este curso de arquitectura?

- ❑ El diseño de la CPU, del sistema de memoria, y del sistema de I/O.

### ¿Por qué estudiar la organización de las computadoras?

- ❑ Para conocer las distintas arquitecturas y saber elegir mejor
- ❑ Para programar pensando en la máquina que corre ese programa y optimizar los programas escritos en alto nivel

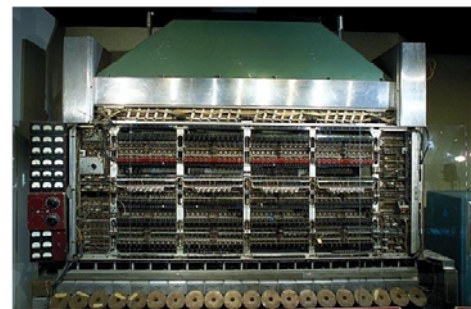
## Arquitectura de Computadoras: Abstracción



3

## Arquitectura de Computadoras. Von-Neumann (clásica)

- Las primeras computadoras tenían programas fijos. En esos casos se “diseñaba a medida”, en vez de programar el procesador.
- En la época de von-Neumann aparece el concepto de computadora con **programa almacenado**. Para ello hace falta:
  - Disponer de la arquitectura del juego de instrucciones
  - Detallar la computación como una serie de instrucciones: el programa.



IAS Machine. Design directed by John von Neumann.  
First booted in Princeton NJ in 1952  
Smithsonian Institution Archives  
(Smithsonian Image 95-06151)  
3m x 3m x 0.80m

4

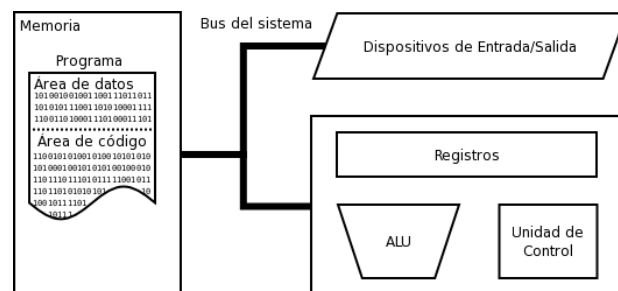
## Computadoras Actuales



5

## Arquitectura de Ordenadores. **Von-Neumann (clásica)**

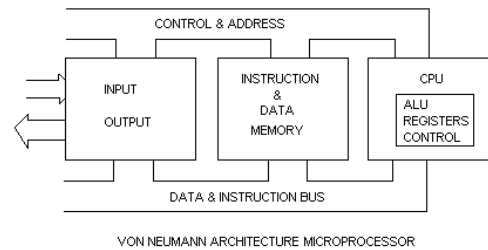
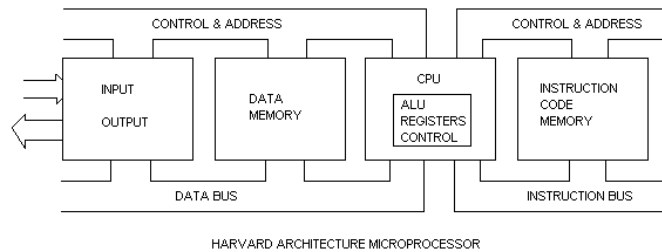
- La arquitectura de **von Neumann** (Eckert-Mauchly) es un modelo de diseño de computadores que usa:
  - Unidad de procesamiento
  - Una sola memoria para instrucciones y datos.
- Problemas de la arquitectura de **von Neumann** :
  - Cuello de botella: velocidad de transferencia de datos entre memoria y CPU



6

# Arquitectura de Ordenadores. Harvard

- Característica: Dos memorias físicamente separadas para datos e instrucciones.
- De esta manera se puede acceder a instrucciones y datos en paralelo.



## Single Processor Performance

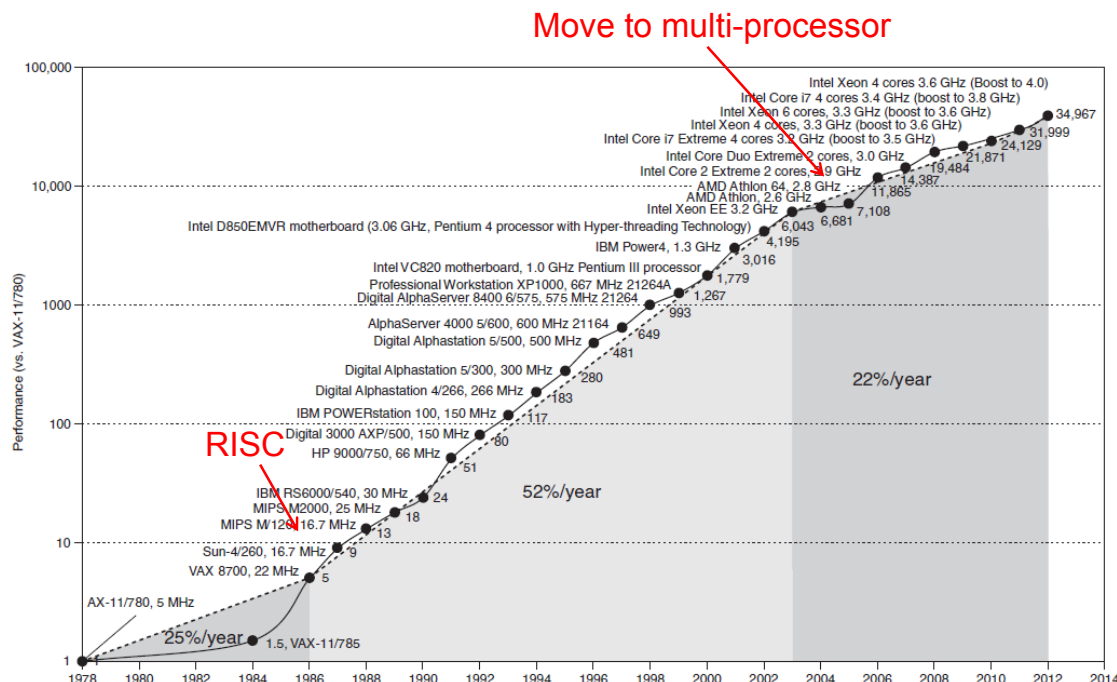


Fig. 1.17. Patterson and Hennessy, Computer Organization and Design, Fifth Edition: The Hardware/Software Interface, 2013

- RISC (Reduced Instruction Set Computer) propone una estrategia de diseño de CPU con instrucciones sencillas. La simplicidad hace que las instrucciones se ejecuten muy rápido.
- CISC (Complex Instruction Set Computer) por el contrario fue la estrategia primitiva donde las instrucciones (y el hardware) eran más complejas.
- Los procesadores RISC en general necesitan más instrucciones para ejecutar la misma tarea que un CISC.

### Características de las Arquitecturas RISC

---

- Juego de instrucciones reducido (sólo las esenciales)
- Acceso a memoria limitado a instrucciones de carga/almacenamiento
- Muchos registros de uso general
- Pocos modos de direccionamiento (inmediato, directo, indexado)
- Formato de instrucción homogéneo (misma longitud y distribución de campos)
- Todas las instrucciones se ejecutan en un ciclo de reloj
- Utiliza unidades de proceso segmentadas y memorias cache

## Arquitectura de Ordenadores. **CISC y RISC**

	<b>CISC</b>	<b>RISC</b>
Nº Instrucciones	100-300	< 128 (2 <sup>7</sup> )
Instrucciones complejas	Si	No
Modos de Direccionamiento	8-20	2-3
Nº Formatos	> 10	≤ 4
Palabras/Instrucción	1-10	1
Instruc. con memoria	Varias	2 (Load/Store)
Ciclos/Instrucción	3-10	~ 1
Nº Registros internos	2-16	≥ 32
Procesador segmentado	A Veces	Siempre
Unidad de Control	µCódigo	Secuencial

11

## Arquitectura de Ordenadores. **CISC y RISC**

- Otras estrategias de diseño de CPU:
  - Superescalar, VLIW, MISC, OISC, massive parallel processing, systolic array, reconfigurable computing, dataflow architecture.
- Si RISC es tan bueno, ¿por qué x86 dominó el mercado por tantos años?
  - Base instalada
  - Inversión en nuevos desarrollos
  - Intel adopta muchas ideas de la arquitectura RISC

12

## El rendimiento del sistema.



El Pensador de Auguste Rodin, Málaga

13

## El rendimiento del sistema. **Unidades de medida**

### **¿Qué se entiende por rendimiento de un sistema?**

Avión	Capacidad (Pasajeros)	Alcance (millas)	Velocidad (Km/h)	Productividad (Pasajeros x Km/h)
Boeing 747	470	6.675	980	460.600
DC-8-50	146	14.030	875	127.750
Concorde	132	6.435	2.175	287.100

- Conviene elegir la unidad adecuada para medir el rendimiento.
- A veces hace falta considerar más de un parámetro para hacer más correcta la decisión a tomar.

14

## Comparación de rendimientos

---

### Rapidez de una computadora

Tiempo de respuesta (tiempo transcurrido entre el comienzo y final de un evento - llamado también tiempo de ejecución o latencia)

una computadora es más rápida si ejecuta determinado programa en menor tiempo

Incremento de productividad (cantidad de trabajo realizado en determinada unidad de tiempo) o throughput (también denominado ancho de banda).

cantidad de tareas realizadas por la máquina en una determinada unidad de tiempo

15

## Rendimiento

---

Al comparar rendimientos, lo que realmente queremos hacer es comparar velocidades relativas entre dos máquinas A y B.

Decir que "A es más rápida que B" significa que el tiempo de respuesta o el tiempo de ejecución de A es menor que el de B para una tarea dada.

$$\frac{\text{Tiempo de ejecución de B}}{\text{Tiempo de ejecución de A}} = n$$

16



## Rendimiento

---

Si definimos al rendimiento como el inverso del tiempo de ejecución :

$$\text{rendimiento} = \frac{1}{\text{tiempo de ejecución}}$$

$$\frac{\text{Tiempo de ejecución de B}}{\text{Tiempo de ejecución de A}} = n = \frac{\text{Rendimiento de A}}{\text{Rendimiento de B}}$$

17

## Rendimiento de la CPU

---

El tiempo de CPU para un programa puede expresarse de dos maneras:

$$\text{Tiempo de CPU} = \text{ciclos de reloj de CPU por programa} * \text{tiempo ciclo de reloj}$$

ó

$$\text{Tiempo de CPU} = \frac{\text{ciclos de reloj de CPU por programa}}{\text{frecuencia de reloj}}$$

18

## Rendimiento de la CPU

En lugar de los ciclos de reloj se puede contar el número de instrucciones ejecutadas -recuento de instrucciones-.

Si se conocen los ciclos de reloj y el recuento de instrucciones se puede calcular el número medio de ciclos de reloj por instrucción (CPI):

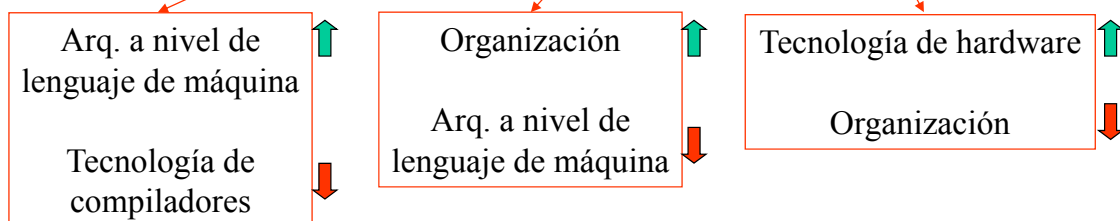
$$\text{CPI} = \frac{\text{ciclos de reloj de CPU por programa}}{\text{recuento de instrucciones}}$$

19

## Tiempo de CPU final

$$\text{Tiempo de CPU} = \text{RI} * \text{CPI} * \text{ciclo de reloj}$$

$$\frac{\text{segundos}}{\text{programa}} = \frac{\text{instrucciones}}{\text{programa}} * \frac{\text{ciclos de reloj}}{\text{instrucciones}} * \frac{\text{segundos}}{\text{ciclo de reloj}}$$



20

## Tiempo de CPU: ejemplo

El CPI indica el número medio de ciclos de reloj que necesita cada instrucción para ejecutarse

$$T_{\text{CPU}} = T_{\text{CICLO}} \times \sum_{i=1}^n \text{CPI}_i \times C_i$$

Para un cierto programa

TIPO	LD/ST	ARITH <sub>INT</sub>	ARITH <sub>FP</sub>	CONTROL
CPI <sub>PROMEDIO</sub>	6	2	8	4
Nº INS <sub>CLASE</sub>	15	50	20	15

$$T_{\text{CPU}} = T_{\text{CICLO}} \times \{ 6 \times 15 + 2 \times 50 + 8 \times 20 + 4 \times 15 \} = 410 \times T_{\text{CICLO}}$$

21

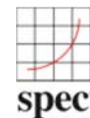
## Otras maneras de medir el rendimiento

**MIPS y MFLOPS.**- Dependen del programa elegido (valor de pico), no sirven para comparar sistemas con diferente juego de instrucciones.

**MIPS: Mega Instructions Per Second**

**MFLOPS: Mega Floating point Operations Per Second**

**System Performance Evaluation Cooperative (SPEC, [www.spec.org](http://www.spec.org))**



Es un estándar (benchmark) para la medida del rendimiento.

Ejemplo Versión 2000: Sistema base: SparcStationUltra 5\_10 (300Mhz,256MB RAM)

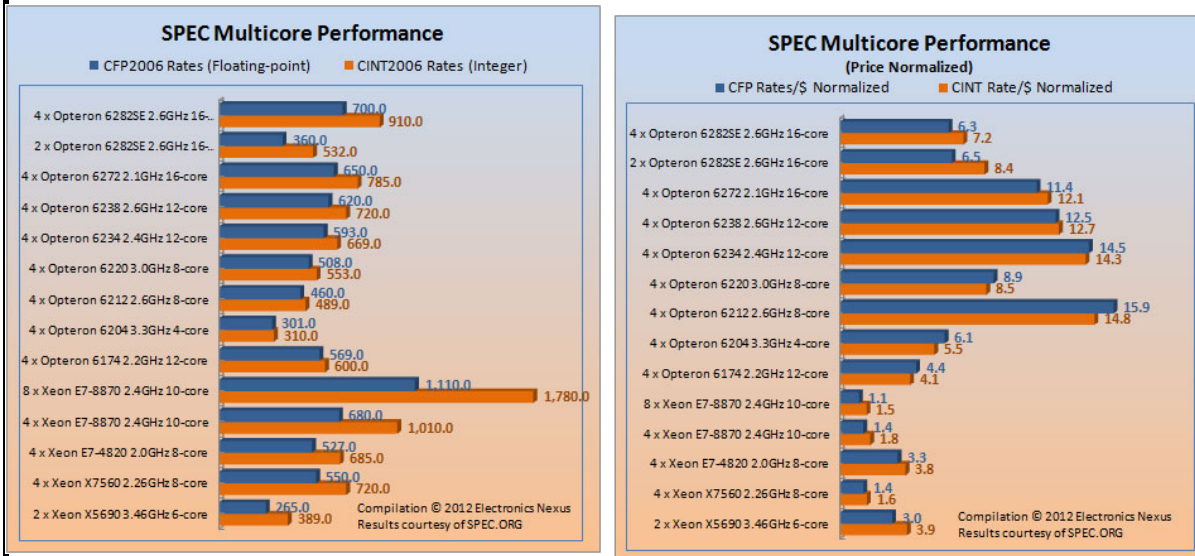
- SPECint2000: 12 programas en C y C++.
- SPECfp2000: 14 programas en FORTRAN (77 y 90) y C.

Ejemplo Versión 2006: Sistema base: 296 MHz UltraSPARC II

- SPECint2006: 12 programas en C y C++.
- SPECfp2006: 17 programas en FORTRAN, C y C++

22

## Otras maneras de medir el rendimiento



23

## Aceleración del caso común

Optimizar el rendimiento de las ocurrencias más frecuentes sobre las menos frecuentes.

Este principio tiene varios puntos a favor sobre una optimización generalizada del rendimiento de la computadora.

Un diseño que trate de minimizar el tiempo de ejecución de todas las instrucciones de una CPU determinada seguramente requerirá agregados de lógica adicional que acrecentará los costes de producción, por un lado y desmerecerá el rendimiento total del equipo en funcionamiento normal.

24

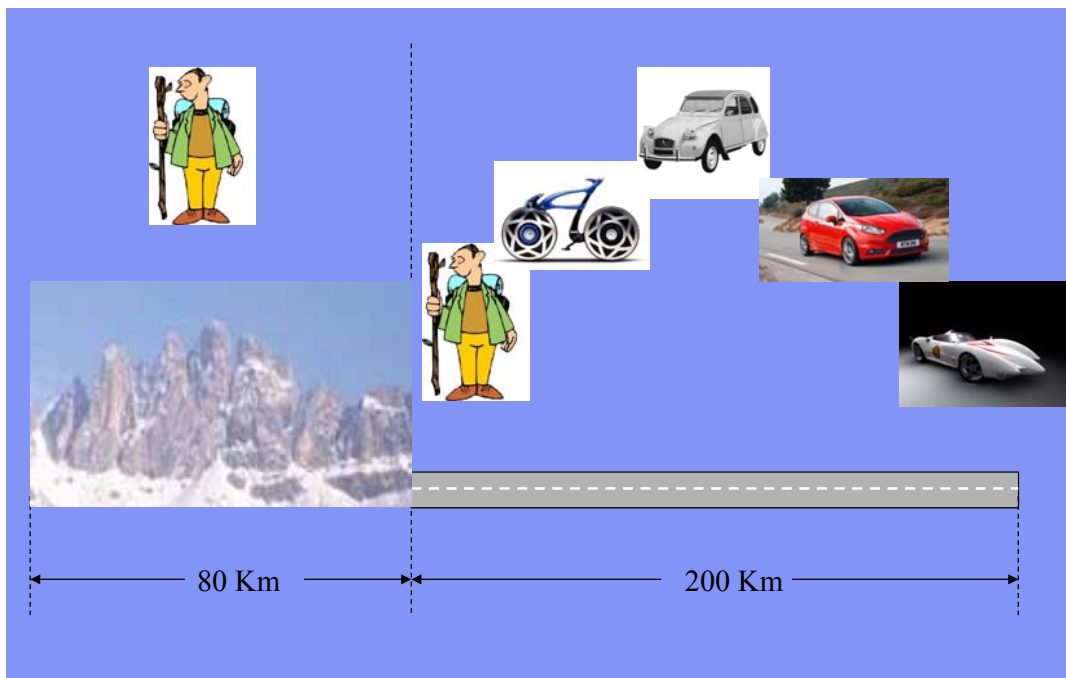
## Ejemplo

Una persona debe realizar un viaje desde un pueblo a otro atravesando zonas montañosas en las que indefectiblemente debe hacer el recorrido a pie. Se emplean 20 horas en cruzar a pie las montañas. Pero para recorrer los últimos 200 Kilómetros puede usar un vehículo de alta velocidad. Hay 5 formas de realizar la última etapa del viaje:

- 1.- Seguir a pie a una velocidad de 4 Km/h.
- 2.- En bicicleta a 10 Km/h.
- 3.- En un Citroën 2CV a 50 Km/h.
- 4.- En un Ford Fiesta a 120 Km/h.
- 5.- En un Speed Car a 600 Km/h.

25

## Ejemplo



26

## Ejemplo (solución)

Podemos encontrar la solución determinando cuanto durará la segunda parte del viaje y sumando ese tiempo a las 20 Hs. necesarias para cruzar las montañas. La tabla siguiente muestra los cálculos realizados para los diferentes vehículos

VEHICULO	Hs. Seg. parte	Acelerac. Seg. parte	Total horas	Acelerac. total
A pie	50,00	1,0	70,00	1,0
Bicicleta	20,00	2,5	40,00	1,8
Citroën	4,00	12,5	24,00	2,9
Fiesta	1,67	30,0	21,67	3,2
Speedcar	0,33	150,0	20,33	3,4

27

## Ley de Amdahl

*" La mejora obtenida en el rendimiento al utilizar algún modo de ejecución más rápido está limitada por la fracción de tiempo que se pueda utilizar en ese modo más rápido. "*

$$\text{Aceleración} = \frac{\text{Rendimiento de la tarea completa usando la mejora cuando sea posible}}{\text{Rendimiento de la tarea completa sin utilizar la mejora}}$$

ó

$$\text{Aceleración} = \frac{\text{Tiempo de ejecución de la tarea sin utilizar la mejora}}{\text{Tiempo de ejecución de la tarea utilizando la mejora cuando sea posible}}$$

28

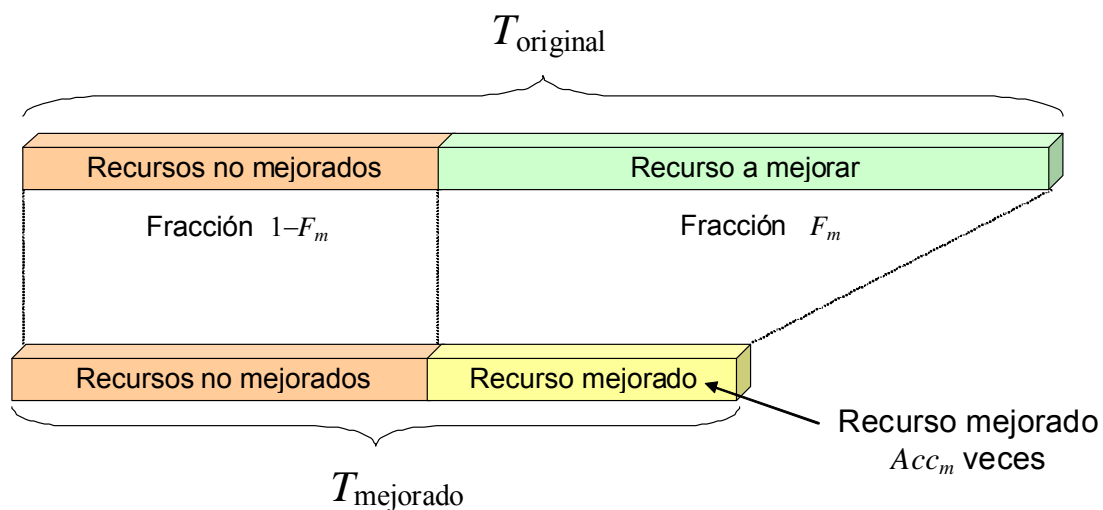
## Ley de Amdahl

La aceleración depende de dos factores:

- La fracción del tiempo de cálculo de la máquina original que puede utilizarse para aprovecharla mejor. Esta fracción (que llamaremos  $\text{fracción}_{\text{mejorada}}$ ) es siempre menor o igual que 1.
- La optimización lograda por el modo de ejecución mejorado; es decir, cuanto más rápido se ejecutaría la tarea si solamente se utilizase el modo mejorado. Este valor es el cociente entre tiempo del modo mejorado y el tiempo del modo original y es siempre mayor que 1. Llamaremos a este valor  $\text{aceleración}_{\text{mejorada}}$ .

29

## Ley de Amdahl



30

## Tiempo de ejecución

---

El tiempo de ejecución utilizando la máquina original con el modo mejorado será el tiempo empleado utilizando la parte no mejorada de la máquina más el tiempo empleado utilizando la parte mejorada.

$$\begin{aligned} \text{Tiempo de ejecución}_{\text{mejorado}} &= \text{Tiempo de ejecución}_{\text{original}} * \\ &* \left( (1 - \text{Fracción}_{\text{mejorada}}) + \frac{\text{Fracción}_{\text{mejorada}}}{\text{Aceleración}_{\text{mejorada}}} \right) \end{aligned}$$

31

## Aceleración Global

---

$$\begin{aligned} \text{Aceleración}_{\text{global}} &= \frac{\text{Tiempo de ejecución}_{\text{original}}}{\text{Tiempo de ejecución}_{\text{mejorado}}} = \\ &= \frac{1}{(1 - \text{Fracción}_{\text{mejorada}}) + \frac{\text{Fracción}_{\text{mejorada}}}{\text{Aceleración}_{\text{mejorada}}}} \end{aligned}$$

32



Que pasa si la  $A_m$  tiende a infinito?  $A_m \rightarrow \infty$

*“Si una mejora es utilizable en una fracción de una tarea no podemos aumentar la velocidad de la tarea más que el recíproco de 1 menos esa fracción”*

$$\text{Aceleración}_{\text{máxima}} = \frac{1}{1 - \text{Fracción}_{\text{mejorada}}}$$

lo que expresa que lo máximo que podemos esperar es que la fracción de la tarea mejorada se realice en un tiempo tendiente a cero.

33

## Ejemplo

En el ejemplo anterior, si la segunda parte del viaje se realiza a velocidad de la luz (1.080.000.000 Km/h):

Horas de la segunda parte:	0,000000185 (666 $\mu$ s)
Aceleración de la segunda parte:	270000000
Aceleración total:	3,49999

Y, según el corolario **3,5**

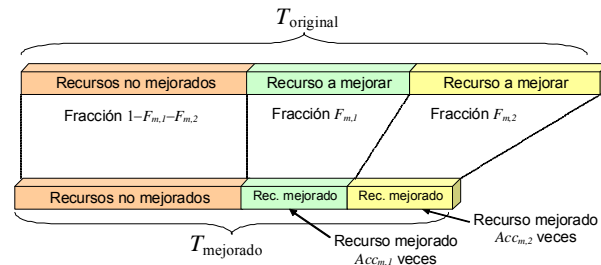
**CONCLUSIÓN: Si manejas rápido no vas a llegar antes  
Pero podés matarte**

34

## Ley de Amdahl

- Con una mejora:

$$Acc_{global} = \frac{1}{(1 - F_m) + \frac{F_m}{Acc_m}}$$



- Caso general con  $n$  mejoras:

$$Acc_{global} = \frac{1}{1 - \sum_{i=1}^n F_{m,i} + \sum_{i=1}^n \frac{F_{m,i}}{Acc_{m,i}}} = \frac{1}{\sum_{i=0}^n \frac{F_i}{Acc_i}}$$

Siendo  $F_0$  la fracción no mejorada y  $Acc_0 (=1)$  la aceleración no mejorada

35

## Un primer ejemplo...

Un circuito realiza la operación de multiplicar en 2,3 microsegundos, de los cuales 0,5  $\mu$ seg. Se necesitan para el cálculo de rebalse del resultado. Se realiza una versión optimizada de es circuito que reduce a la mitad el tiempo de análisis de rebalse.

- Hallar la fracción mejorada.
- Hallar la aceleración mejorada.
- Cuál es la aceleración global del nuevo circuito respecto del anterior?.

### Solución convencional

(puesto que se tienen todos los datos)

$$T_{original} = 2,3 \mu\text{seg}$$

$$T_{original-rebalse} = 0,5 \mu\text{seg}$$

$$T_{nuevo-rebalse} = 0,25 \mu\text{seg}$$

$$T_{mejorado} = 2,3 - 0,5 + 0,25 = 2,05 \mu\text{seg}$$

$$Acc_{global} = 2,3 / 2,05 = 1,12$$

### Solución vía ley de Amdahl

$$F_{mejorada} = 0,5 \mu\text{seg} / 2,3 \mu\text{seg}$$

$$Acc_{mejorada} = 0,5 / 0,25 = 2$$

$$Acc_{global} = \frac{1}{(1 - F_{mejorada}) + \frac{F_{mejorada}}{Acc_{mejorada}}} = \frac{1}{(1 - \frac{0,5}{2,3}) + \frac{0,5/2,3}{2}} = 1,12$$

36

otro problema de ejemplo...

Un procesador  $A$  realiza una operación en  $N$  segundos.  
Se optimiza dicha operación en un procesador  $B$  que mejora  $2/3$  de la misma en un factor de  $2,5$ .

Un tercer procesador  $C$  (basado en  $B$ ) optimiza la operación duplicando la velocidad de la parte que mejora.

Cuál es la aceleración global si la mejora de  $C$  se realiza sobre la fracción que mejoró  $B$  ( $2/3$ )?

37

otro problema de ejemplo... (continuación)

A

$f_{nm} = 1/3$ $a_{nm} = 1$	$f_m = 2/3$
--------------------------------	-------------



B

$f_{nm} = 1/3$ $a_{nm} = 1$	$a_m = 2,5$
--------------------------------	-------------



C1

$f_{nm} = 1/3$ $a_{nm} = 1$	$a_m = 2$
--------------------------------	-----------

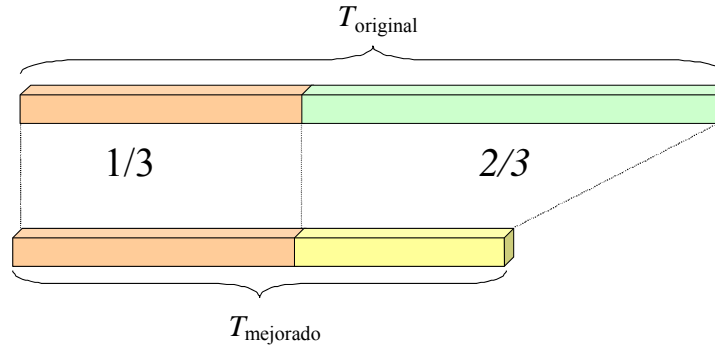
$$Acc_{B-A} = \frac{1}{1 - 2/3 + \frac{2/3}{2.5}} = 1.66$$

$$Acc_{C1-A} = \frac{1}{1 - 2/3 + \frac{2/3}{2 * 2.5}} = 2.14$$

38

otro problema de ejemplo... (continuación)

Una vez obtenido B, la proporción de las fracciones ya no es la misma



Como  $T_{original} > T_{mejorado}$  entonces las fracciones respecto de  $T_{mejorado}$  ya no son  $1/3$  y  $2/3$

**Problema...** ¿Cómo calcular las nuevas fracciones de B para C?

39

otro problema de ejemplo... (continuación)

$$N' = \frac{1}{3}N + \frac{2}{2.5}N$$

d.m.a.m por  $N'$

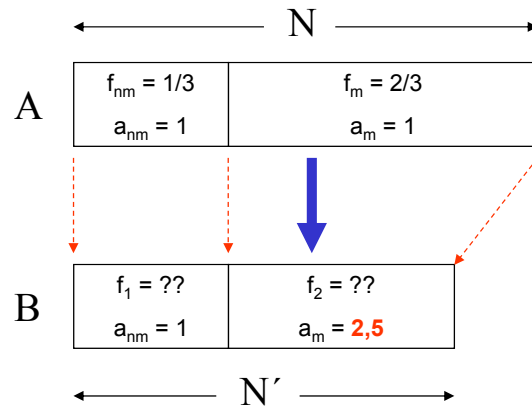
$$\frac{N'}{N'} = \frac{1}{3} \frac{N}{N'} + \frac{2}{2.5} \frac{N}{N'}$$

Ya sabemos que  $Acc_{B-A} = \frac{5}{3} = 1.66$

Entonces:  $\frac{N'}{N'} = 1 = \frac{1}{3} * \frac{5}{3} + \frac{2/3}{10/4} \frac{5}{3} = \frac{5}{9} + \frac{4}{9}$

$$Acc_{C-B} = \frac{1}{1 - 4/9 + \frac{4/9}{2}} = 1.29$$

$$Acc_{C-A} = Acc_{B-A} \cdot Acc_{C-B} = 1.29 \cdot 1.66 = 2.14$$



Por lo que  $f_1 = 5/9$  y  $f_2 = 4/9$

40

## Un tercer problema...

Se dispone de un juego de instrucciones en el que cada tipo de instrucción, el porcentaje medio de uso de cada uno y los ciclos medios necesarios para ejecutarlas se reflejan en la tabla adjunta:

Tipo instrucción	Enteros	Punto Flotante	Load	Store
Promedio de uso	45%	25%	15%	15%
CPI (ALU)	1	6	1	1
CPI (MEM)	3	3	6	6

Suponiendo que el reloj del sistema se duplica y que el tiempo de acceso a memoria no varía, calcular utilizando la ley de Amdahl el porcentaje obtenido en la mejora.

### **SOLUCIÓN:**

Si el tiempo de acceso a memoria no varía, la mejora propuesta sólo afectará a los CPI que se ejecutan en la ALU.

La mejora implica duplicar la frecuencia. Por tanto el valor de la aceleración normal es  $Acc_m=2$ . La fracción de tiempo sobre el que se aplica la mejora, sólo afecta a los ciclos que se ejecutan en la ALU. Por tanto la frecuencia mejorada

$$F_m = [0,45*1+0,25*6+0,15*1+0,15*1]/[0,45*(1+3)+0,25*(6+3)+0,15*(1+6)+0,15*(1+6)] = 0,366$$

Aplicando la ley de Amdahl.  $AG = 1,224$ . La mejora supone una aceleración del 22,4%.

41

## Las 8 grandes ideas en Arquitectura de Computadoras

