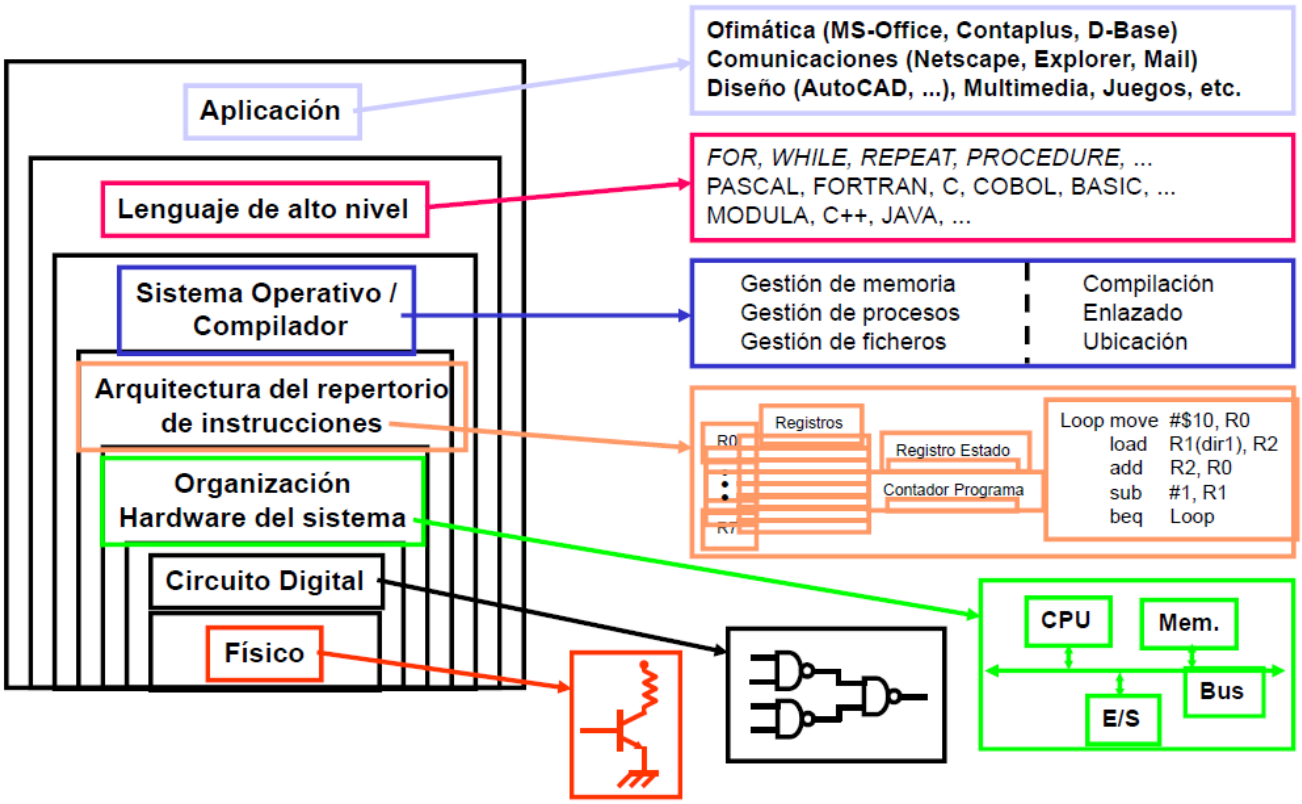
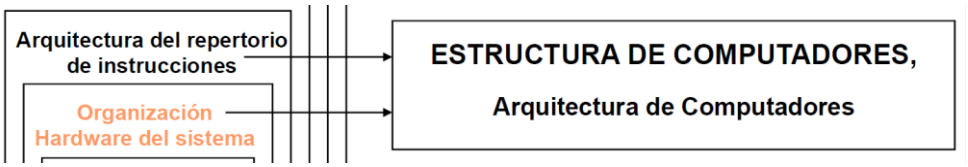
❖ Tema 1. TECNOLOGÍA, RENDIMIENTO, CONSUMO Y COSTE

➢ 1.1 Introducción

Posicionamiento de la Arquitectura de Computadores ⟹





La Arquitectura de Computadores es el siguiente paso de nivel respecto de lo estudiado en Estructura de Computadores. Los conceptos son similares, pero se profundiza mucho más en sistemas actuales y por ello más complejos.

Nos encontramos por debajo del nivel software (el Sistema Operativo), pero no alcanzamos a entrar en un detalle de diseño digital.

- **Visión actual de la Arquitectura de Computadores**

El buen funcionamiento de un computador, su eficacia y su eficiencia vienen determinados por un cúmulo de factores tremendamente amplio.

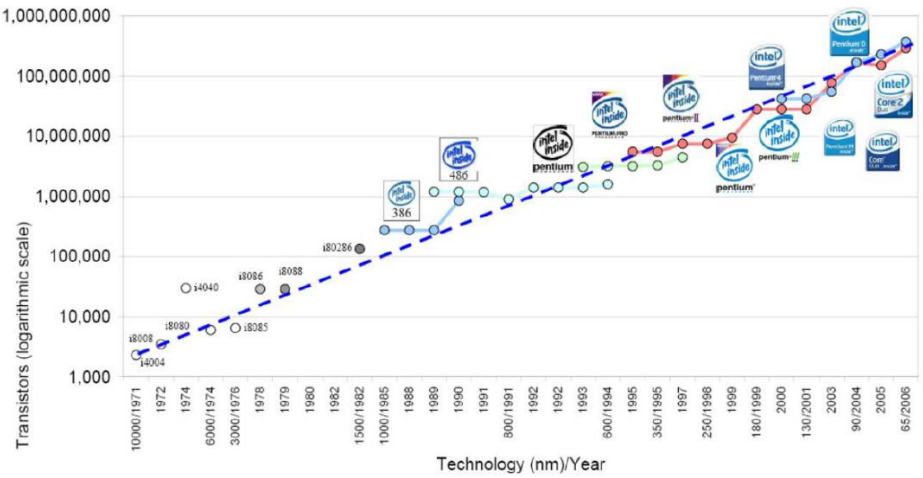
La parte que aquí nos atañe de este entramado es el componente HARDWARE y la gestión de las instrucciones de bajo nivel (*Instruction Set Architecture* = ISA)

- ¿Para qué tarea será utilizado el computador? ¿Tendrá sistema operativo?
- ¿Cuáles son las restricciones? Coste, consumo de energía, tamaño, ancho de banda.
- ¿Qué tipo de ISA soporta la CPU?
- ¿Qué tecnologías de diseño disponemos para la fabricación? Microelectrónica.

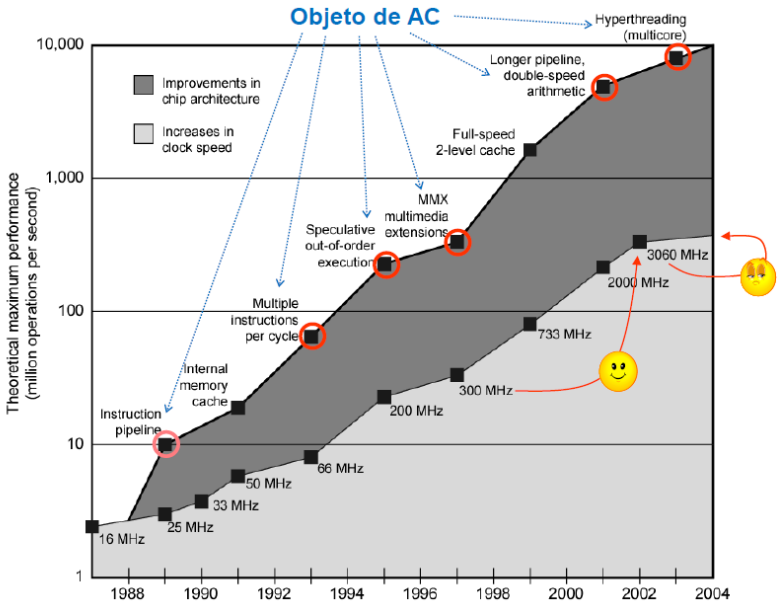
Arquitectura de Computadores = ISA + Microelectrónica + Hardware

➢ 1.2 Evolución tecnológica

Ley de Moore: el número de transistores que componen un microprocesador se duplica aproximadamente cada 2 años.



El progreso de la tecnología de integración nos ha proporcionado microchips cada vez más rápidos durante muchos más años de los que a priori se esperaba. Sin embargo la velocidad de procesado no es el principal cuello de botella en los computadores actuales.



La velocidad de reloj de los procesadores ha sufrido un tremendo frenazo en la última década. Entre 2004 y 2018 apenas hemos pasado de 3 GHz a 4'5 GHz.

Aunque no se ha abandonado el camino de la mejora de integración de transistores que aumente la velocidad de reloj se empezaron a seguir otros senderos para mejorar el rendimiento:

- Pipelining
- Ejecución fuera de orden
- Paralelismo (Multicores)

Algunas de estas líneas de investigación afectan a la propia fabricación de los chips y otras van directamente a modificar los ISA.

- **<u>Paralelismo</u>**

Buena parte del temario de esta asignatura se centra en la explotación del paralelismo, es decir, realizar varias tareas a la vez por componentes diferentes sin que se modifique la estructura puramente lógica y secuencial de la ejecución (proveniente de niveles más altos de abstracción).

- <u>Paralelismo a nivel de Instrucciones (ILP):</u> se ejecutan varias instrucciones a la vez, potencialmente desordenadas. Es un proceso transparente al programador.

- <u>Paralelismo a nivel de Datos (DLP):</u> arquitecturas vectoriales, instrucciones multimedia y GPU's. Se utilizan las mismas instrucciones para multitud de datos.

- <u>Paralelismo a nivel de Hilo (TLP):</u> ejecución de hilos independientes de cálculo que se ejecutan de forma concurrente (en uno o varios procesadores).

- **Limitaciones de la Ley de Moore**

¿Por qué la ley de Moore tiene tope?

Es evidente que llegará un momento en que no nos quepan más transistores por mm^2 por mucho que mejoremos la tecnología de miniaturización, sin embargo hemos llegado a tropezar con otros problemas diferentes.

- 1er Problema: ***"Power Wall"***

  Haciendo uso de conocimientos básicos de física, sabemos que los circuitos electrónicos se calientan cuando por ellos circula corriente. Si hacemos circuitos más pequeños y a la vez más rápidos la densidad de potencia que disipan (y por tanto calor que hay que refrigerar) aumenta dramáticamente.

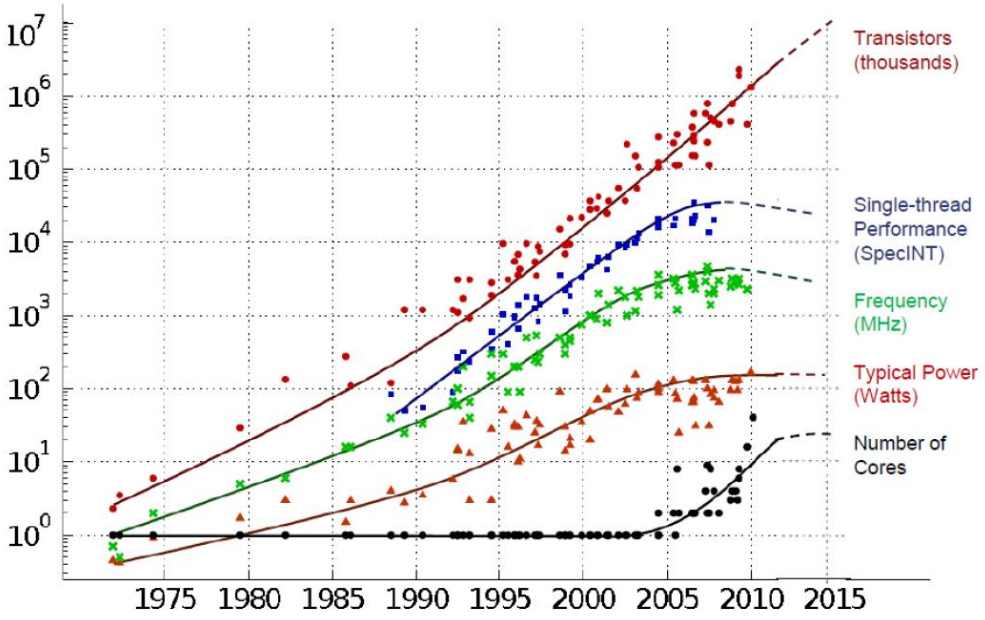
  En otras palabras, que los *"microventiladores"* no hacen suficiente trabajo.

- 2º Problema: ***"Retardo en interconexiones"***

  Al mejorar la tecnología de los transistores que componen los chips mejoramos linealmente la respuesta en velocidad. El problema es que las conexiones internas y externas de los chips no soportan esa mejora, su retardo de respuesta no evoluciona a la par.

  En este apartado se podría incluir un tipo de latencia de conexión con nombre y apellidos, que es la latencia con memoria. Las memorias son bastante más lentas que los chips, existe ***"Memory Wall"***. No porque las memorias no mejoren en velocidad, sino porque no lo hacen a igual ritmo que los procesadores.

- **Panorámica de la evolución tecnológica en 40 años**



➢ 1.3 Rendimiento

Recordemos algunas de las ecuaciones básicas del rendimiento computacional

$$T_{CPU} = N \cdot CPI \cdot t_{ciclo} \qquad f(Hz) = \frac{1}{t_{ciclo}(s)}$$

$$CPI = \frac{Ciclos\ totales}{N} \qquad CPI = \sum_{i=1}^{k} CPI_i \cdot f_i$$

$N \rightarrow Número\ de\ instruccionesde\ programa$
$CPI \rightarrow Ciclos\ promedio\ por\ instrucción$
$CPI_i \rightarrow Ciclos\ promedio\ de\ una\ clase\ de\ instrucción$
$f_i \rightarrow Frecuencia\ de\ aparición\ (en\ tanto\ por\ 1)\ de\ la\ clase\ de\ instrucción$

A estas ecuaciones de rendimiento podemos añadir dos conceptos más, MIPS y MFLOPS.

$$MIPS = \frac{N}{T_{CPU} \cdot 10^6} \qquad MFLOPS = \frac{N_{FP}}{T_{CPU} \cdot 10^6}$$

$MIPS \rightarrow Millions\ of\ Instruction\ Per\ Second$
$MFLOPS \rightarrow Millions\ of\ FLoating\ point\ Operations\ Per\ Second$

Estas dos últimas ecuaciones no se pueden utilizar en cualquier entorno, se deben tomar bajo ciertas condiciones de prueba para que no den resultados irreales.

La medida de MFLOPS se usa mucho en la actualidad para relatar el rendimiento de un computador. La cuestión es que para medir el rendimiento de las operaciones en punto flotante de una forma justa se deben utilizar programas de prueba llamados *Benchmarks* que no estén preparados para ningún tipo de arquitectura en concreto.
Además no es suficiente con probar un solo Benchmark, se deben hacer con varios.

- **SPEC (*Standard Performance Evaluation Corporation*)**

SPEC establece unos bancos de pruebas estándar que comparan nuestro computador con una máquina de referencia. Es la medida actualmente más aceptada.

A día de hoy existen 6 generaciones de "suites" SPEC que han ido evolucionando para adaptarse a los nuevos computadores

SPEC CPU 89, SPEC CPU 92, SPEC CPU 95, SPEC CPU 2000, SPEC CPU 2006, SPEC CPU 2017

La última es SPEC CPU 2017, consta de 4 suites de pruebas de enteros y de punto flotante

Suite	Contents
SPECspeed 2017 Integer	10 integer benchmarks
SPECspeed 2017 Floating Point	10 floating point benchmarks
SPECrate 2017 Integer	10 integer benchmarks
SPECrate 2017 Floating Point	13 floating point benchmarks

- *SPECspeed:* se mide el rendimiento del computador en la ejecución de una única tarea. Una medida diferente para cada Benchmark.

$$r_i = \frac{Tiempo\ en\ MR}{Tiempo\ en\ SAE}$$

$r_i \rightarrow Ratio\ del\ Benchmark\ i$
$MR \rightarrow Máquina\ de\ Referencia$
$SAE \rightarrow Sistema\ A\ Evaluar$

- *SPECrate:* se mide el rendimiento del computador en la ***ejecución paralela*** de varias tareas. Se evalúa para cada Benchmark, se mide el tiempo necesario para ejecutar varias instancias iguales a la vez del mismo Benchmark.

$$r_i = \frac{Tiempo\ 1\ copia\ en\ MR}{Tiempo\ N\ copias\ en\ SAE} \cdot (N\ copias)$$

Finalmente, el resultado de SPEC se da con un único número independientemente de si es SPECspeed o SPECrate obtenido de la media geométrica de sus r_i

$$SPEC = \sqrt[N]{\prod_{i=1}^{N} r_i}$$

¿Para qué sirve usar una Máquina de Referencia?

Nos aporta *neutralidad* a la hora de comparar máquinas diferentes A y B. Ejemplo

$$SPEC\ A = \sqrt{\frac{MR1}{A1} \cdot \frac{MR2}{A2}} \quad , \quad SPEC\ B = \sqrt{\frac{MR1}{B1} \cdot \frac{MR2}{B2}} \quad \Rightarrow \quad \frac{SPEC\ A}{SPEC\ B} = \frac{\sqrt{B1 \cdot B2}}{\sqrt{A1 \cdot A2}}$$

Las medidas obtenidas en la máquina de referencia no interfieren en el cociente de A con B. Podemos compararlas independientemente de qué usemos como MR.

- **Ley de Amdahl**

"La mejora obtenida en el rendimiento de un sistema debido a la alteración de uno de sus componentes está limitada por la fracción de tiempo que se utiliza dicho componente"

$$T_{Mejora} = T_{Sin\_Mejora} \cdot \left[(1 - F) + \frac{F}{x}\right]$$

$$Speedup = \frac{T_{Sin\_Mejora}}{T_{Mejora}} \qquad Speedup = \frac{1}{\left[(1 - F) + \frac{F}{x}\right]}$$

$F \rightarrow Fracción\ (en\ tanto\ por\ 1)\ en\ que\ se\ usa\ originalmente\ el\ componente$
$x \rightarrow multiplicador\ de\ mejora\ del\ componente$

Ejemplos de la Ley de Amdahl

EJ1) El 10% del tiempo de ejecución de mi programa es consumido por operaciones en PF. Se mejora la implementación de la operaciones PF reduciendo su tiempo a la mitad ($x = 2$).

$$T_{Mejora} = T_{Sin\_Mejora} \cdot \left[(1 - 0.1) + \frac{0.1}{2}\right] = 0.95 \cdot T_{Sin\_Mejora} \qquad Speedup = \frac{1}{0.95} = 1.053$$

EJ2) Para mejorar la velocidad de una aplicación, se ejecuta una parte que consumía el 90% del tiempo sobre 100 procesadores en paralelo. El 10% restante no admite la ejecución en paralelo.

$$T_{Mejora} = T_{Sin\_Mejora} \cdot \left[(1 - 0.9) + \frac{0.9}{100}\right] = 0.109 \cdot T_{Sin\_Mejora} \qquad Speedup = \frac{1}{0.109} = 9.17$$

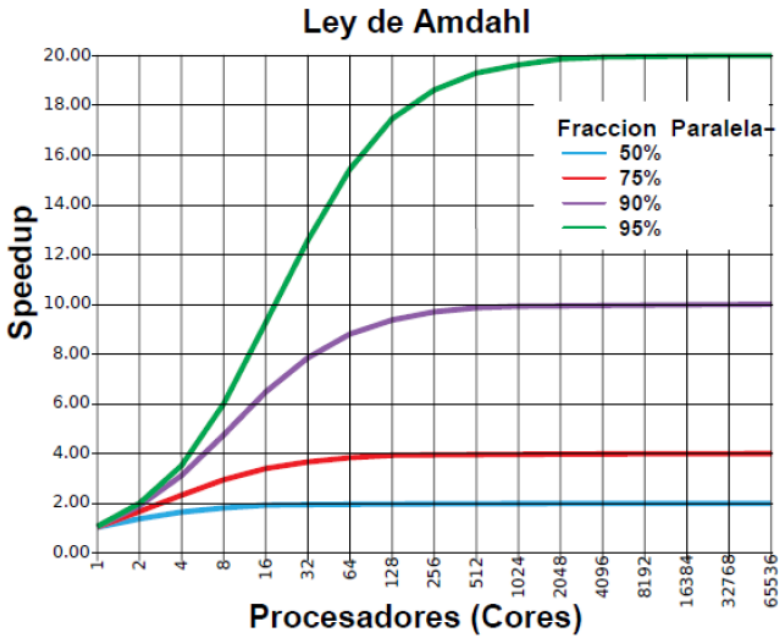
Usar 100 procesadores nos da un aumento que no llega al *(x10)* de velocidad

- **Eficiencia**

$$E = \frac{Speedup}{x}$$

La eficiencia tiene como valor máximo 1. Disminuye si se consigue poco Speedup con un gran multiplicador de mejora parcial.

Una de las mejoras más habituales en el uso de la ley de Amdahl es utilizar múltiples núcleos de procesado para una cierta fracción de programa (como en el ejemplo 2).



El estudio hecho por Thomas Puzak acerca del impacto del paralelismo de cores en aplicaciones, nos demuestra que la mejora en Speedup nunca puede ser mayor que la inversa de la fracción no paralelizable. Es algo relativamente fácil de obtener de la definición, sin embargo resulta impactante el poco efecto que tiene multiplicar el número de núcleos a partir de cierto valor.

➢ 1.4 Consumo

Uno de los principales retos del diseño de procesadores es el consumo. No tanto por la energía que debemos suministrarle sino porque toda la potencia consumida se traduce en calentamiento del circuito que hay que disipar.

$Energía \rightarrow Julios.\ Medida\ neta\ de\ consumo$

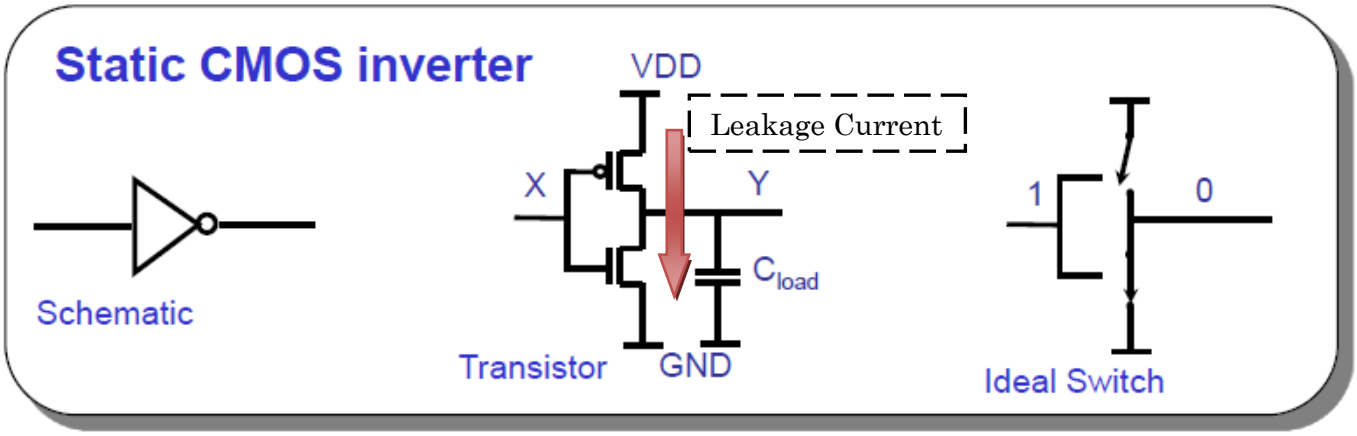
$Potencia \rightarrow \frac{Julios}{segundo} = vatios.\ Medida\ de\ consumo\ por\ unidad\ de\ tiempo$

A la hora de establecer una medida de consumo fiable necesitamos la potencia porque no me sirve de nada decir que consume 1000 J si no sé cuánto tiempo le lleva hacerlo.

Energía estática: se consume en cuanto el dispositivo esté encendido, incluso aunque no esté haciendo nada. No depende de la carga de trabajo.

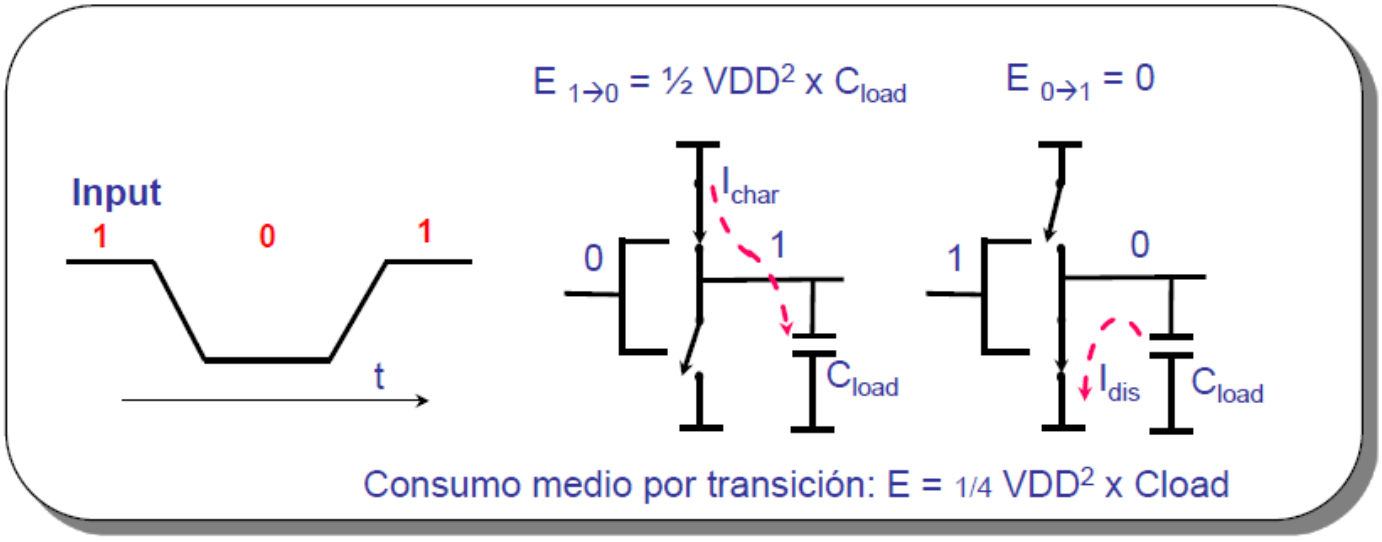
Energía dinámica: se consume por cada conmutación de los transistores. Aumenta cuanto mayor es la carga de trabajo.

Energía estática



Durante el proceso de inactividad de los transistores no debería circular corriente entre la tensión de alimentación VDD y la tierra GND. Sin embargo los CMOS no son interruptores perfectos, no están cerrados del todo. Se producen corrientes de fuga (Leakage current).

Energía dinámica



En la tecnología CMOS el condensador es el encargado de mostrar a la salida del inversor si hay un 1 ó un 0. Cuando el condensador está descargado (tenía un 1 a la entrada y mostraba un 0), de repente la entrada cambia a 0, entonces se carga para mostrar un 1 a la salida. La carga del condensador consume energía de la fuente de alimentación.

El proceso de descarga sin embargo no consume energía de la fuente.

$$E_{din}(un\ ciclo\ completo) = \frac{1}{4} \cdot V_{DD}^2 \cdot C_{load}$$

- **<u>Disminución del consumo</u>**

- <u>Consumo estático:</u> podemos mejorar el aislante de las conexiones en los transistores. También podemos apagar temporalmente las zonas del procesador que no estén activas (*power gating*).

- <u>Consumo dinámico:</u> evitamos conmutaciones innecesarias e intentamos incluir transistores más pequeños y rápidos. Sin embargo si son más pequeños también meteremos más.

  La energía dinámica consumida depende además de la tensión de alimentación y la Capacidad del condensador de carga. Si reducimos ambos, reducimos el consumo

$E_{din} \propto \cdot V_{DD}^2 \cdot C_{load}$

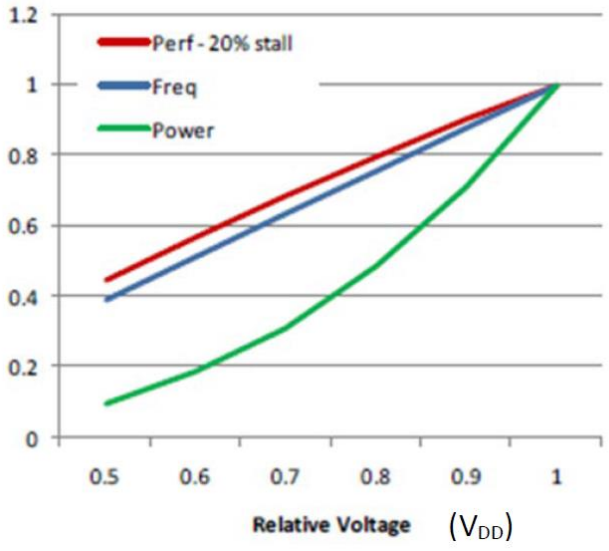
Pero la potencia dinámica depende además de la frecuencia

$P_{din} \propto \cdot V_{DD}^2 \cdot C_{load} \cdot f$

Los circuitos más rápidos tienen la misma energía promedio por ciclo, pero claro, con más frecuencia suponen m

Ésta es la razón de la aparición del *"Power Wall"* que comentamos en la ley de Moore. Existe un punto límite al aumento en la frecuencia por culpa del calentamiento.

La tensión VDD tenía valores de 5V en la lógica TTL de finales de los 90, actualmente los microchips usan tensiones por debajo de 1V.



La gráfica muestra tres curvas que parten del punto (1,1). A partir del voltaje relativo 1 disminuimos hacia atrás su valor.

Rendimiento : decrece linealmente con VDD

Frecuencia máx: decrece linealmente con VDD

Potencia : decrece cúbicamente con VDD

La potencia depende cuadráticamente de VDD, pero como la frecuencia también disminuye, en conjunto depende cúbicamente.

➢ 1.5 Costes

El coste del circuito integrado se evalúa con la siguiente expresión

$$Coste\ CI = \frac{Die\ coste + Testing\ coste + Packging\ coste}{Final\ test\ yield}$$

$$Die\ coste = \frac{coste\ del\ Wafer}{Dies\ por\ Wafer \cdot Die\ yield}$$

$Die\ coste \rightarrow Coste\ de\ cada\ dado\ funcional$
$Final\ test\ yield \rightarrow Productividad\ final$
$Coste\ del\ Wafer \rightarrow Coste\ de\ cada\ oblea\ (redonda)$
$Die\ yield \rightarrow Productividad\ de\ dados\ en\ oblea\ (\text{sin}\ defectos)$