

# Tema 15: Diseño para bajo consumo de energía

**Arquitectura de Computadoras**

**Ing. Nicolás Majorel Padilla ([npadilla@herrera.unt.edu.ar](mailto:npadilla@herrera.unt.edu.ar))**

<http://microprocesadores.unt.edu.ar/arqcom/>

# Temas que veremos

---

- ▶ Repaso consumo de energía y *Power Wall*.
- ▶ Técnicas de reducción del consumo de energía.
  - ▶ Mejora del proceso de fabricación.
  - ▶ Uso de paralelismo.
  - ▶ Escalado dinámico de tensión y frecuencia.
  - ▶ Técnicas de *Gating*.
  - ▶ Procesadores heterogéneos.
- ▶ Cambio de paradigma.

# Lectura recomendada

---

- ▶ Computer Organization and Design, RISC-V Edition (2da ed, 2021)
  - ▶ Sección 1.7: *The Power Wall*
  - ▶ Sección 1.11: *Fallacies and Pitfalls*

# ¿Dónde estamos parados?

---

- ▶ “Las tres P” como criterios de diseño en Arquitectura de Computadoras.
  - ▶ Vimos un poco de Costo y su influencia.
  - ▶ ¡Vimos mucho sobre Performance!
    - ▶ Cómo medirla, cómo mejorarla.
    - ▶  $t = CI * CPI * T$
    - ▶ Cómo es afectada por los distintos componentes de una computadora.
  - ▶ Vimos una introducción al consumo de energía.
    - ▶ Ahora profundizaremos esos conceptos, y veremos cómo produjeron un cambio de paradigma.

# Repaso: Consumo de Energía

---

- ▶ En tecnología CMOS, cada transición lógica (de 0 a 1 o viceversa) disipa energía de conmutación.
  - ▶ La energía disipada depende de la capacitancia de carga y del cuadrado de la tensión de alimentación.
- ▶  $P = N * A * C * V^2 * f$ 
  - ▶ Ecuación de **consumo de energía dinámico**.
- ▶ No es sencillo de reducir el consumo de energía:
  - ▶ Disminuir la frecuencia, afecta la performance.
  - ▶ Disminuir la tensión de alimentación, obliga a bajar la frecuencia.
  - ▶ Disminuir la cantidad de transistores, afecta la performance.
  - ▶ Disminuir la capacitancia, es muy costoso y tiene limitaciones.

# Repaso: *The Power Wall*

---

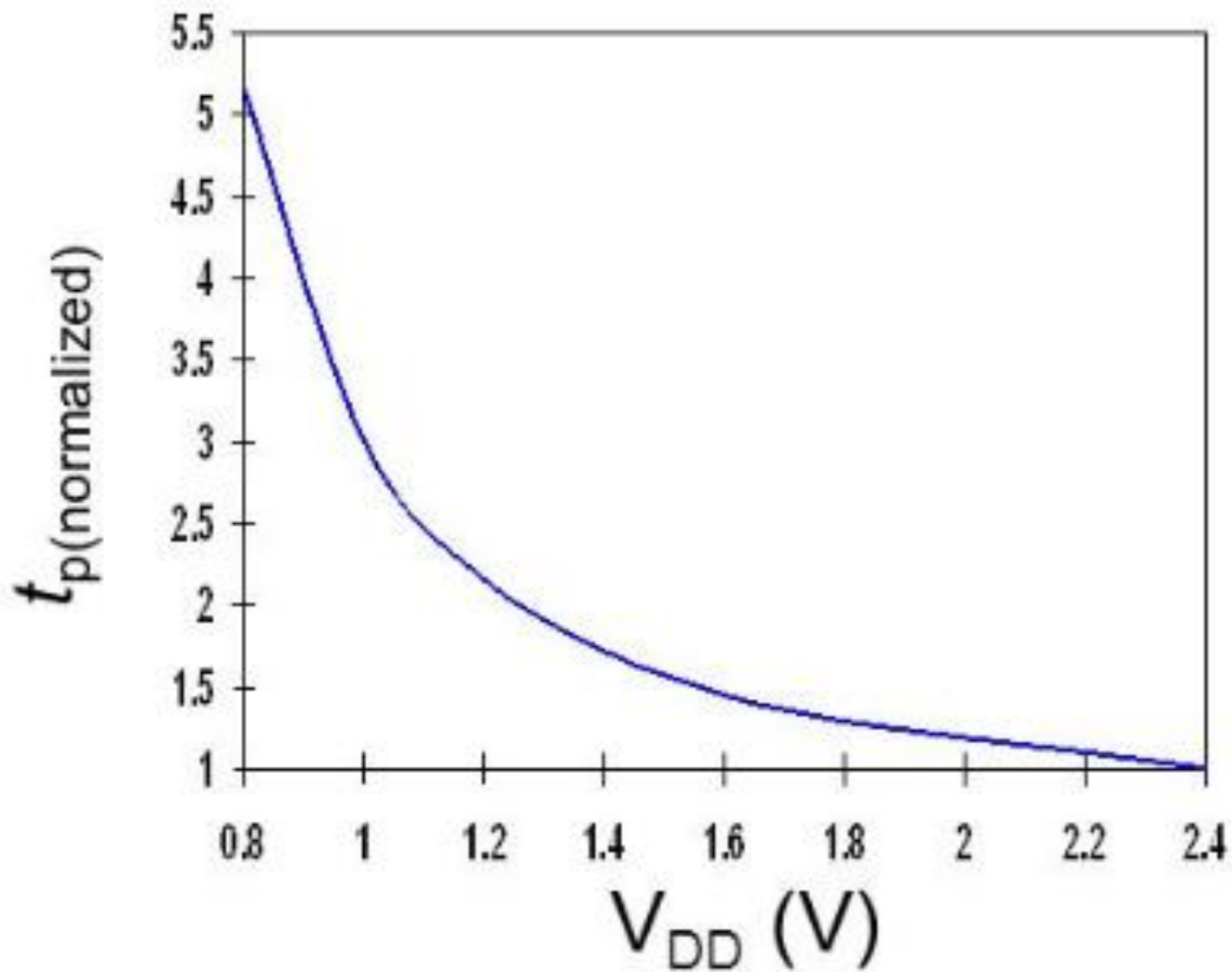
- ▶ Hasta el año 2004, el diseño era motivado por costo y performance.
  - ▶ Impulsado principalmente por la Ley de Moore.
    - ▶ Los transistores son baratos. Agregarlos es “fácil”.
  - ▶ Y por el *Dennard Scaling*
    - ▶ Los transistores eran cada vez más rápidos, y disipaban menos energía.
- ▶ A partir de entonces, se empezó a considerar el consumo de energía como criterio principal.
  - ▶ En parte por el fin del *Dennard Scaling*.
  - ▶ *¿Cómo seguir mejorando la performance, pero sin aumentar el consumo de energía?*
  - ▶ Motivó un cambio de paradigma.

# Repaso: Nuevos requerimientos

---

- ▶ Desde hace 15 años surgieron multiplicidad de dispositivos móviles personales.
  - ▶ Operados a batería. Conectados a Internet.
- ▶ También apareció el concepto de **Cloud Computing**.
  - ▶ Mega servidores, grandes instalaciones. Software as a Service (SaaS).
- ▶ Y más recientemente: **IoT**.
  - ▶ Sistemas embebidos. Conectados a Internet.
- ▶ La evolución tecnológica requiere:
  - ▶ Más portabilidad → Más miniaturización → Baterías más pequeñas y duraderas → Menor consumo de energía
  - ▶ Todos los nuevos sistemas tienen una limitación de consumo de energía.

# Repaso: Relación entre F y Vdd

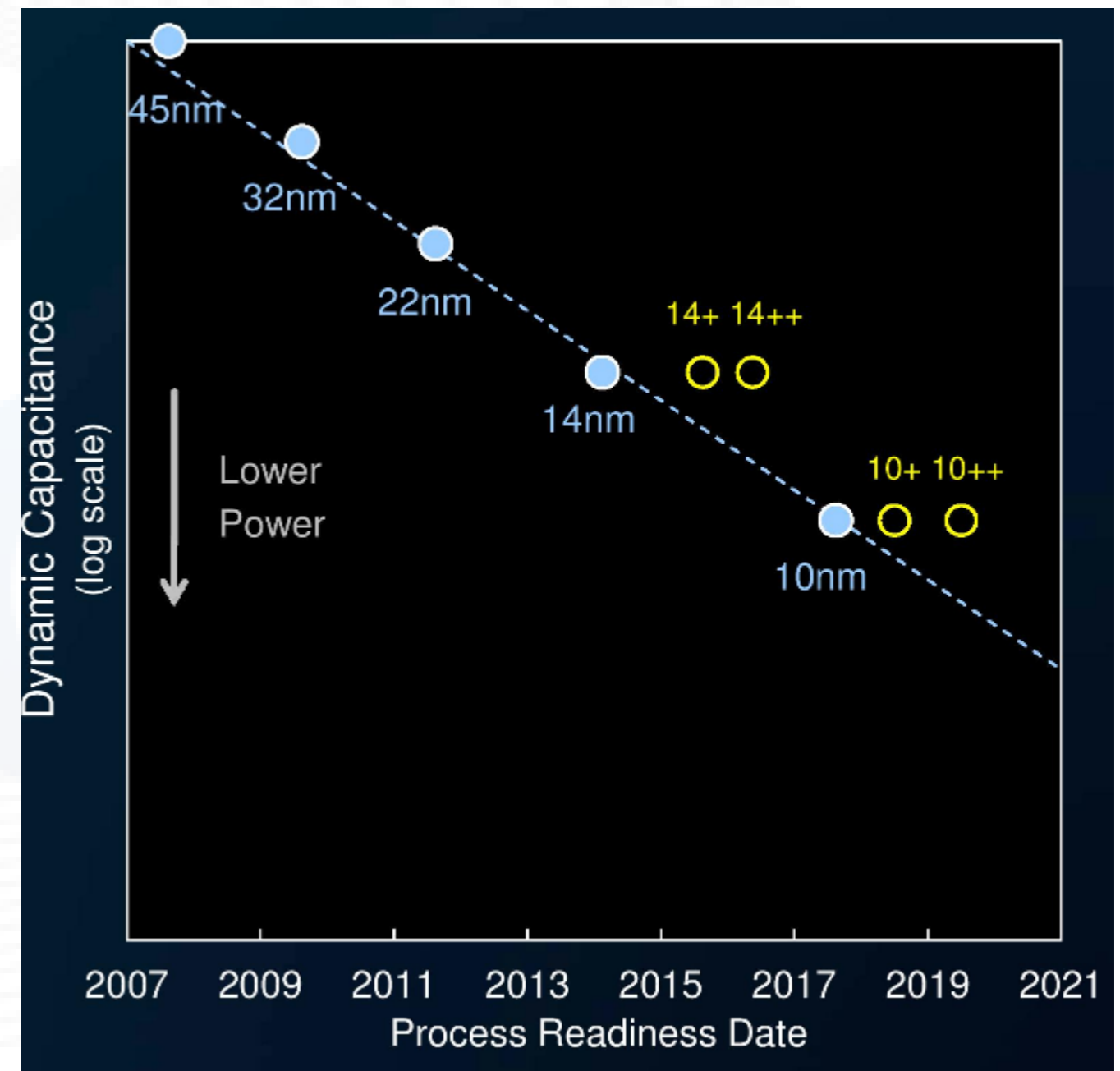


- ▶ Viene dada por la curva de retardo (*delay*) de los transistores.
- ▶ Hay una relación “casi lineal” hasta llegar a una tensión de umbral.



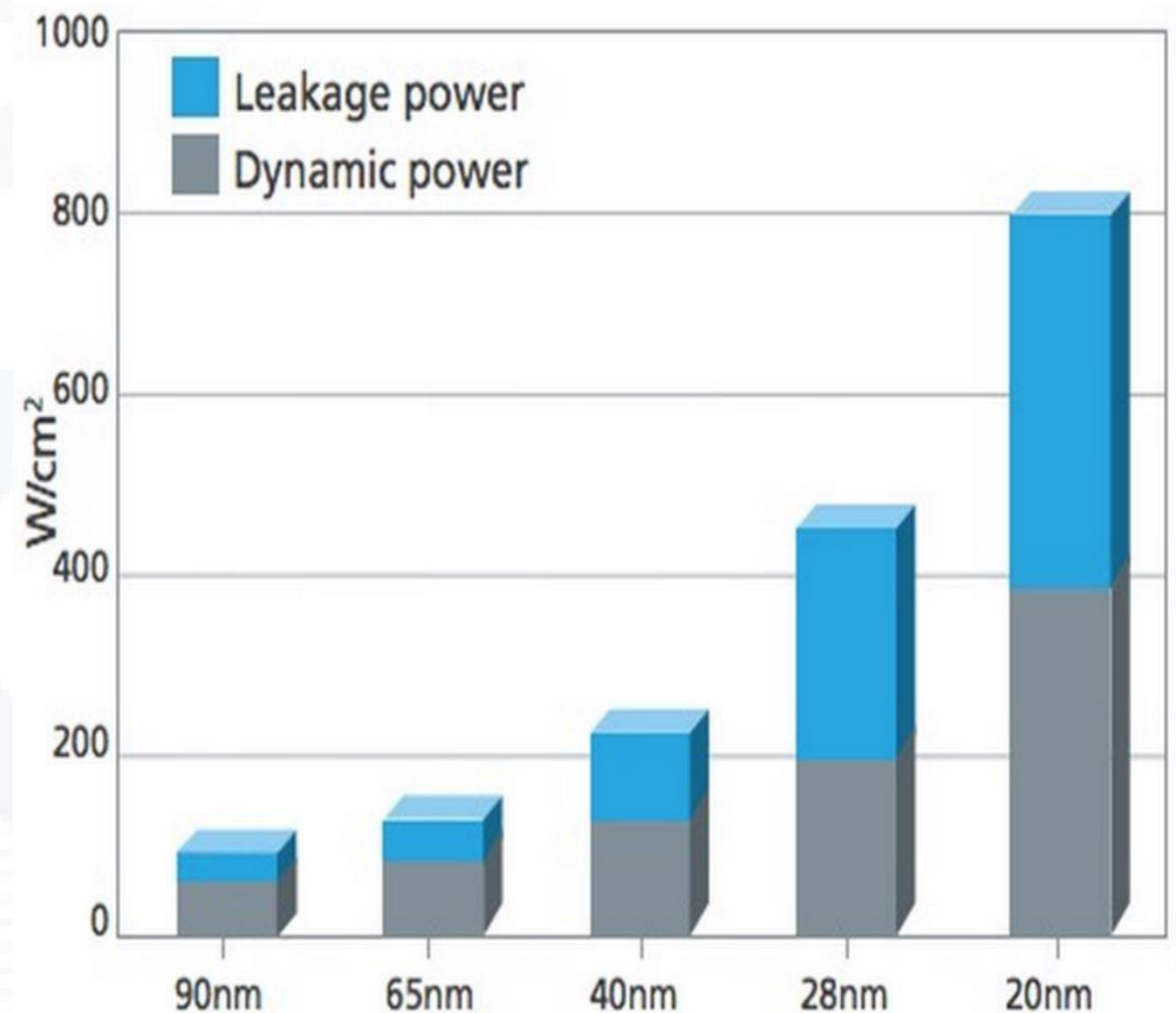
# Mejora del proceso de fabricación

- ▶ Disminuye la capacitancia dinámica.
- ▶ Sin embargo, las compuertas de los transistores tienen sólo unos átomos de ancho, y no son ideales.
- ▶ Esto genera que se produzcan corrientes de pérdida (*leakage*).
  - ▶ Corrientes que no deberían existir.



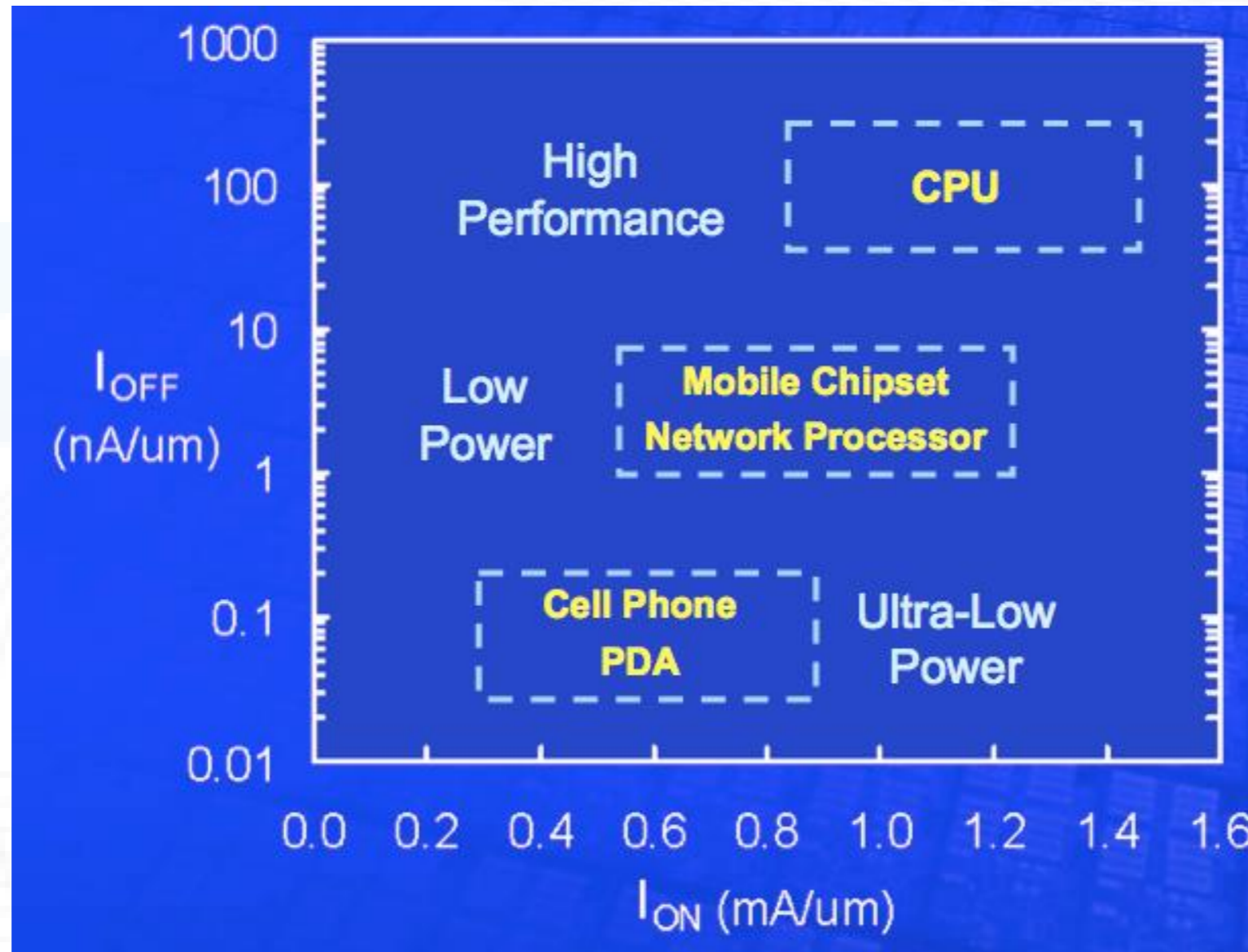
# Corrientes de pérdida

- ▶ Al disminuir el tamaño del transistor, la relación entre corriente de pérdida y corriente activa va aumentando.
  - ▶ Y aumenta el **consumo estático de energía**.
- ▶ Se puede reducir la corriente de pérdida disminuyendo la corriente activa.
  - ▶ Pero esto implicaría reducir la frecuencia.
  - ▶ El capacitor tarda más en cargarse.



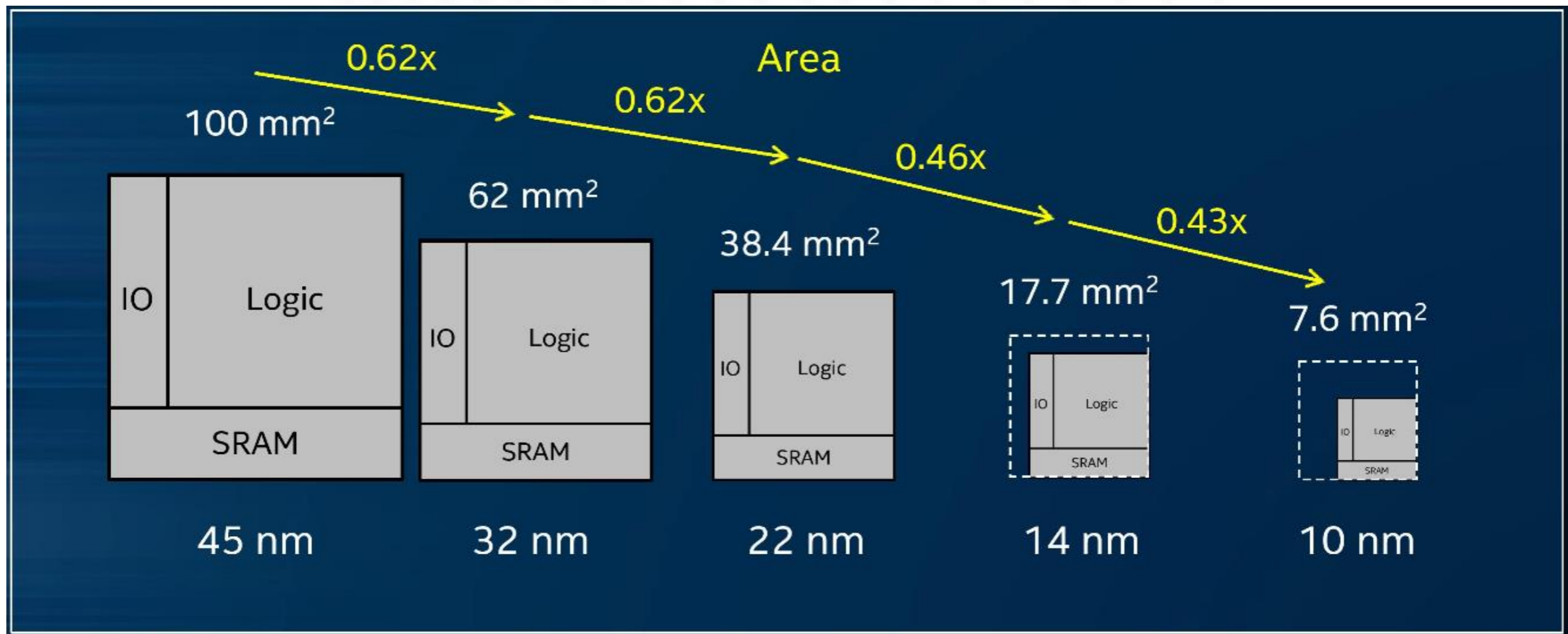
# Conclusión: adaptar proceso al producto

- ▶ No todos los dispositivos necesitan altas frecuencias.



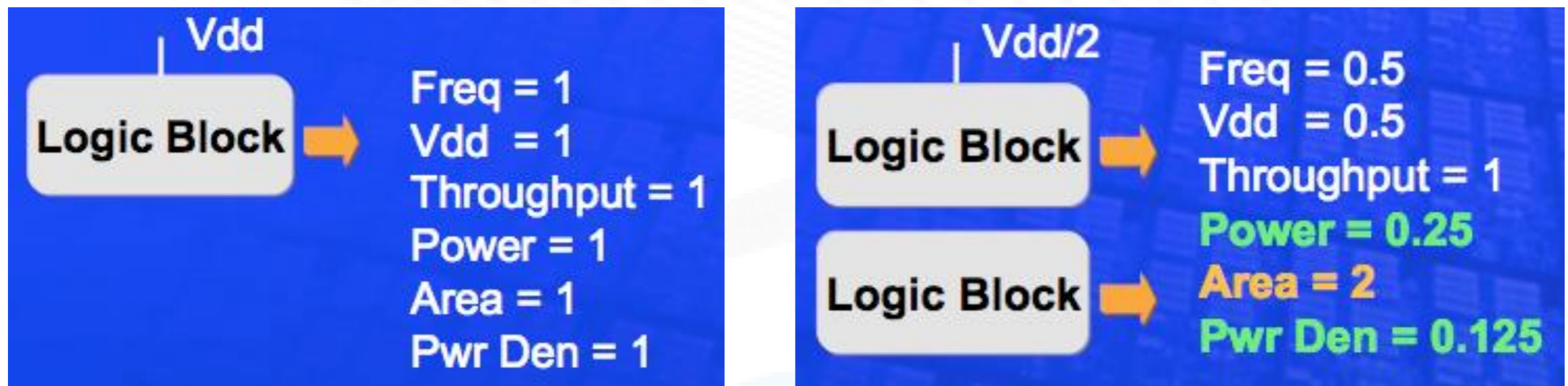
- ▶ ¿Se acuerdan de los *chipelets* que vimos en el Tema 02?

# Mejora del proceso de fabricación



- ▶ El hecho de reducir el tamaño del chip, no sólo permite reducir el costo, y el consumo de energía.
- ▶ Permite poner el doble de circuitos en el mismo espacio, y de esa manera **también reducir el consumo.**

# Reducción de consumo por paralelismo



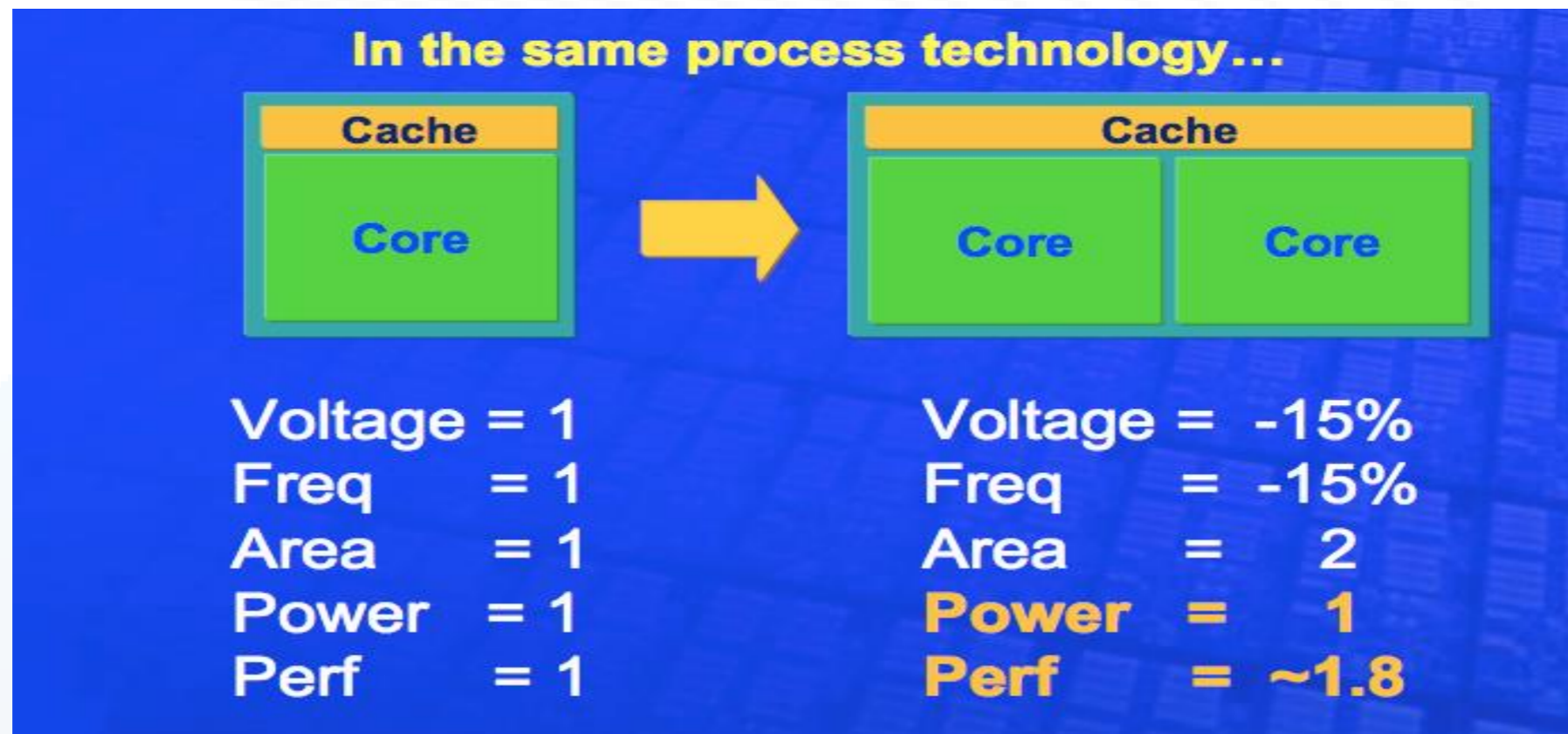
- ▶ Reemplazamos un bloque lógico que hace una tarea, por dos bloques más pequeños, que hacen media tarea cada uno.
- ▶ La frecuencia de los bloques se puede reducir a la mitad, porque tienen que realizar la mitad de la tarea en la misma unidad de tiempo.
- ▶ Mantenemos la productividad, duplicando el área.
- ▶ La reducción de frecuencia permite una reducción similar de la tensión, y por lo tanto la potencia se reduce a la cuarta parte.
  - ▶ *¿Por qué a la cuarta parte?*

# Reducción de consumo por paralelismo

---

- ▶ Esta alternativa es bastante utilizada en sistemas embebidos.
  - ▶ Por ejemplo, en sistemas de audio cada procesador puede procesar un canal.
  - ▶ También puede haber un procesador encargado de procesar audio y otro de procesar video.
    - ▶ Así fueron los primeros iPod con video (2005).

# Reducción de consumo por paralelismo



- ▶ Otra alternativa es mantener el consumo, ¡y mejorar la performance!
- ▶ Nótese que la mejora de performance no es del doble.
  - ▶ Puede aumentar, siempre y cuando los programas puedan hacer uso de los múltiples núcleos.
  - ▶ Ese “detalle” lo veremos en el Tema siguiente.

# Reducción de consumo por paralelismo

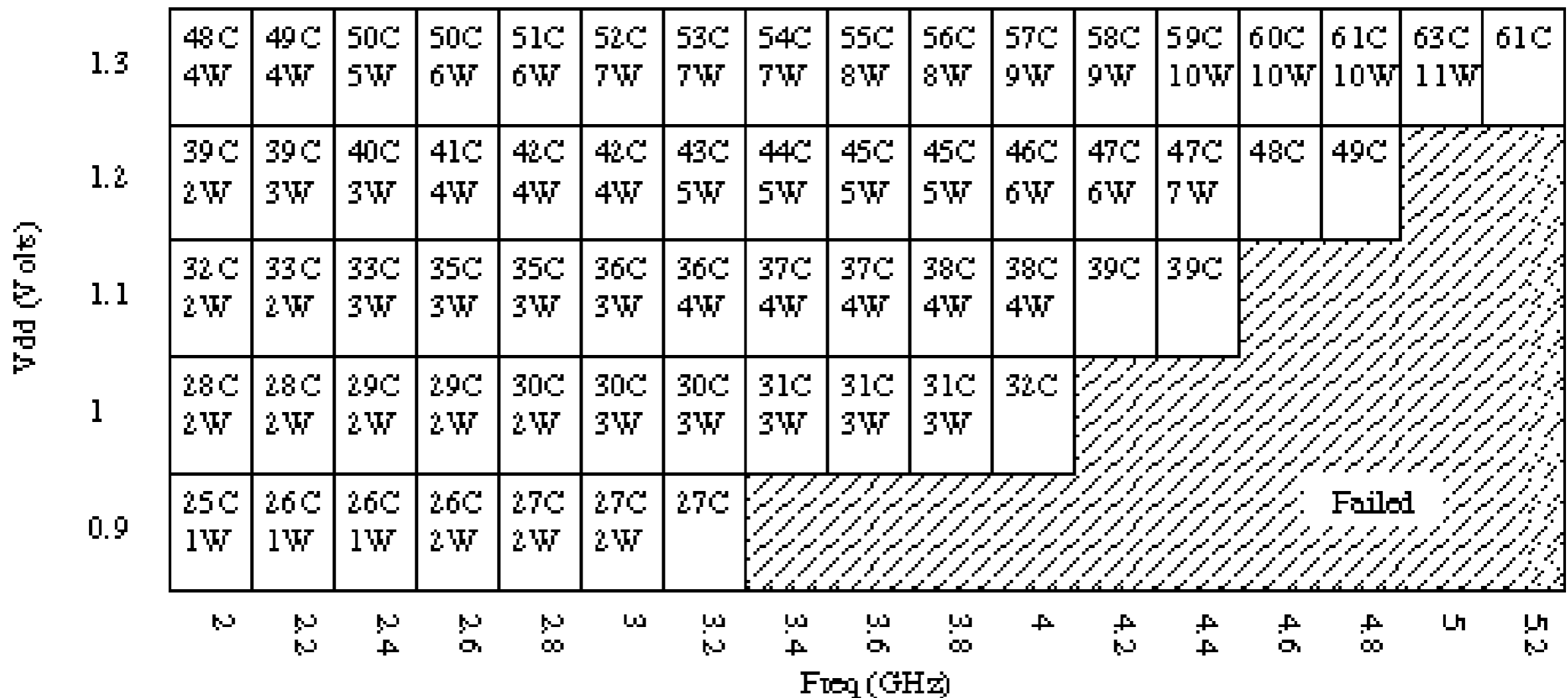
---

- ▶ Uno de los primeros procesadores conocidos en explotar esta técnica fue el procesador Cell, de la PS3 (del 2008).
  - ▶ Fabricado en conjunto entre IBM, Sony y Toshiba.
  - ▶ Tenía 8 procesadores en paralelo, más 1 extra encargado de controlar a los demás.
    - ▶ Cada procesador tenía su propio caché L1 y había un L2 compartido.
    - ▶ Cada procesador podía emitir dos instrucciones por ciclo, a 7 unidades de ejecución.
- ▶ Actualmente, este tipo de diseño es el más común.



# Variación de F y Vdd

- ▶ El procesador Cell también fue uno de los primeros en permitir variar la frecuencia de trabajo y la tensión de alimentación.
- ▶ Ciertas combinaciones no son permitidas.



# Variación de F y Vdd

---

- ▶ En el gráfico anterior se aprecia que bajando F, manteniendo Vdd, se reduce el consumo de energía.
  - ▶ Pero la tarea demora más en completarse.
- ▶ La idea es bajar F y también bajar Vdd, y aumentar los trabajadores para que la performance no se vea afectada.
  - ▶ Ejemplo: un procesador a 4,4 GHz y 1.2 V, consume 7 W.
  - ▶ Un procesador a 2,2 GHz y 0,9 V consume 1 W.
  - ▶ Si se puede cambiar al primero por 2 de los segundos, mantenemos performance y bajamos el consumo a 2 W.

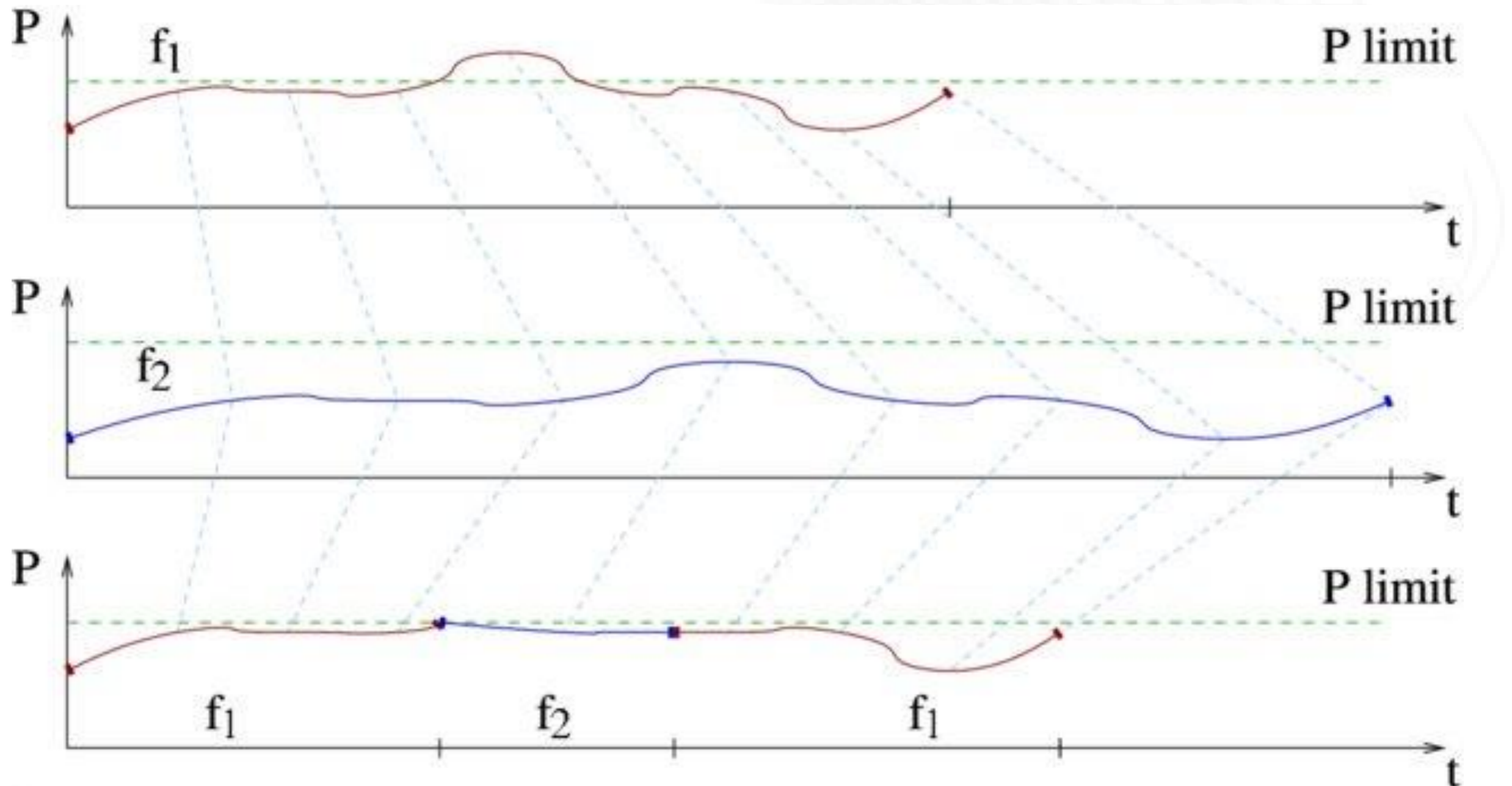
# Escalado Dinámico de Frecuencia

---

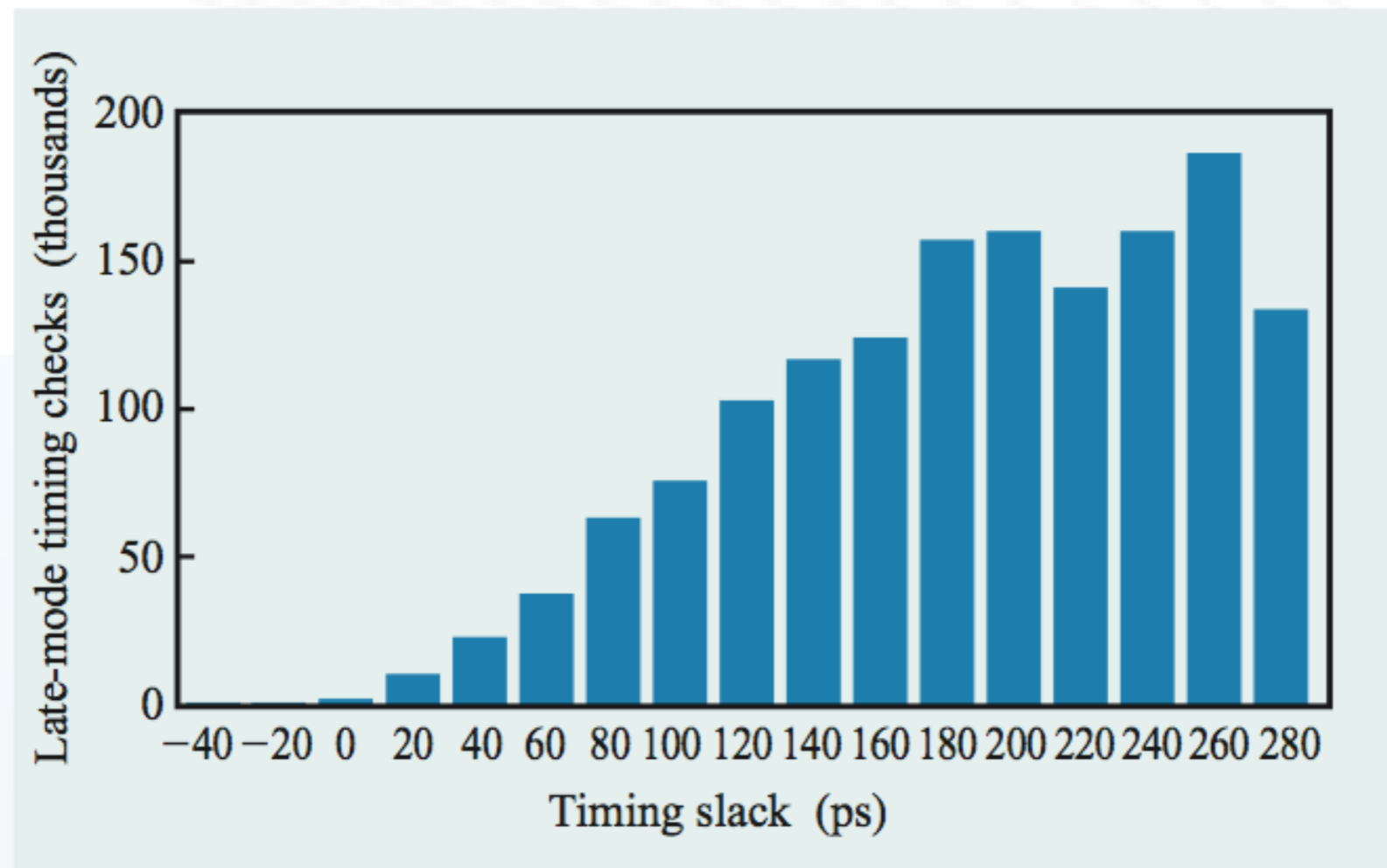
- ▶ Sería ideal que análisis como el anterior se hagan en forma automática.
- ▶ Varios nombres: ***Dynamic Frequency Scaling (DFS)***, *Dynamic Overclocking*, *Turbo Boost*.
- ▶ Cuando una aplicación requiere mayor performance, se aumenta la frecuencia.
- ▶ Cuando no requiere performance, se disminuye la frecuencia para así reducir el consumo de energía.
- ▶ Los procesadores modernos vienen con una frecuencia base y una frecuencia máxima.
  - ▶ Las variaciones son en pasos escalonados.

# Escalado Dinámico de Frecuencia

- ▶ Ejemplo de DFS, donde  $f_1 > f_2$ , y se busca no superar un límite  $P$  de consumo de energía.

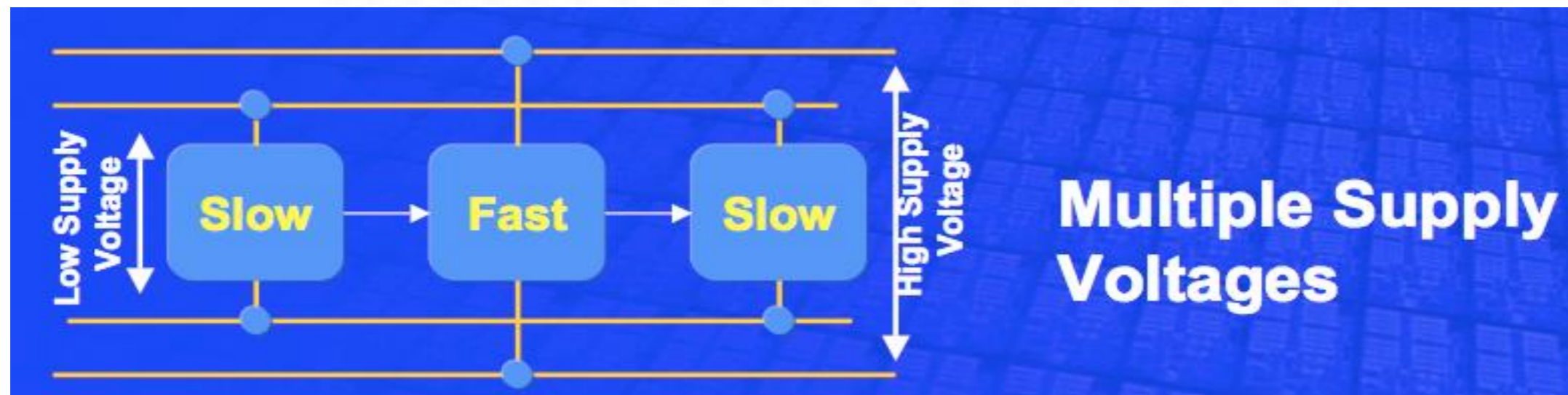


# Análisis de caminos críticos



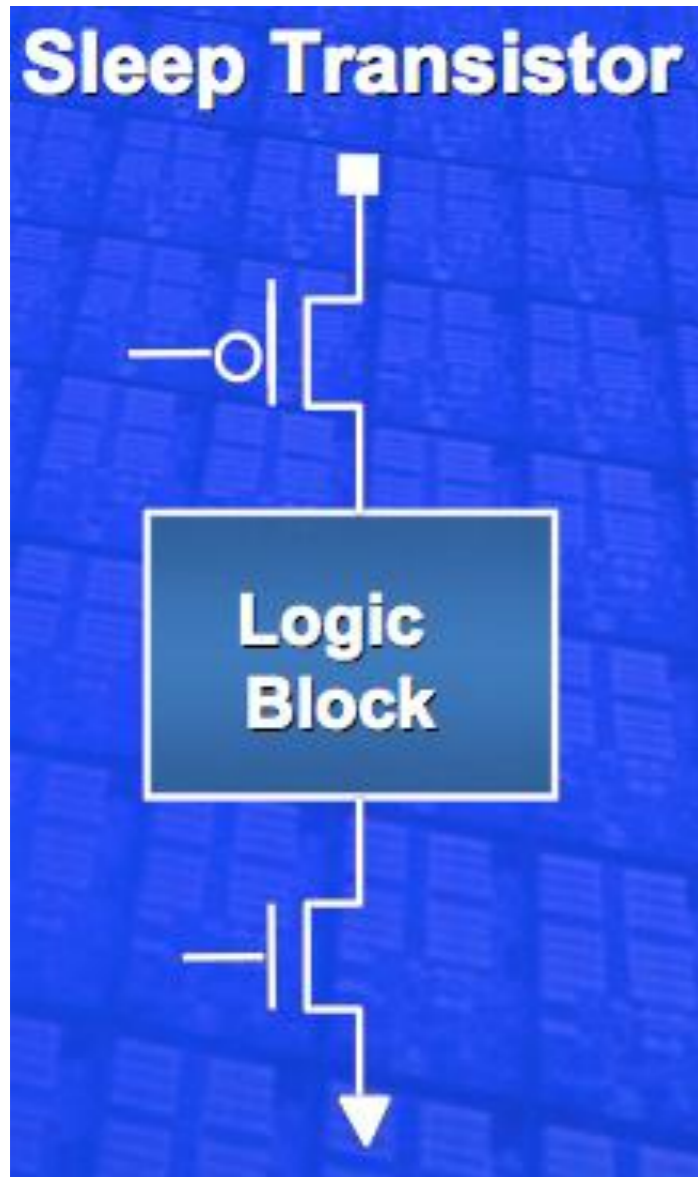
- ▶ Se muestran cuántos caminos existen en un procesador en función de su retardo con respecto al camino crítico.
- ▶ Implica que no todas las partes del procesador tienen que trabajar a la misma frecuencia.

# Escalado dinámico de tensión



- ▶ Se manejan múltiples niveles de tensión en un mismo procesador.
  - ▶ Con múltiples niveles de frecuencia.
- ▶ Porciones del procesador fuera del camino crítico reciben menos tensión.
- ▶ Idealmente, se puede cambiar de manera dinámica el nivel de tensión de un bloque (*Dynamic Voltage Scaling, DVS*).
- ▶ Como también se escala la frecuencia, reciben el nombre de **DVFS**.

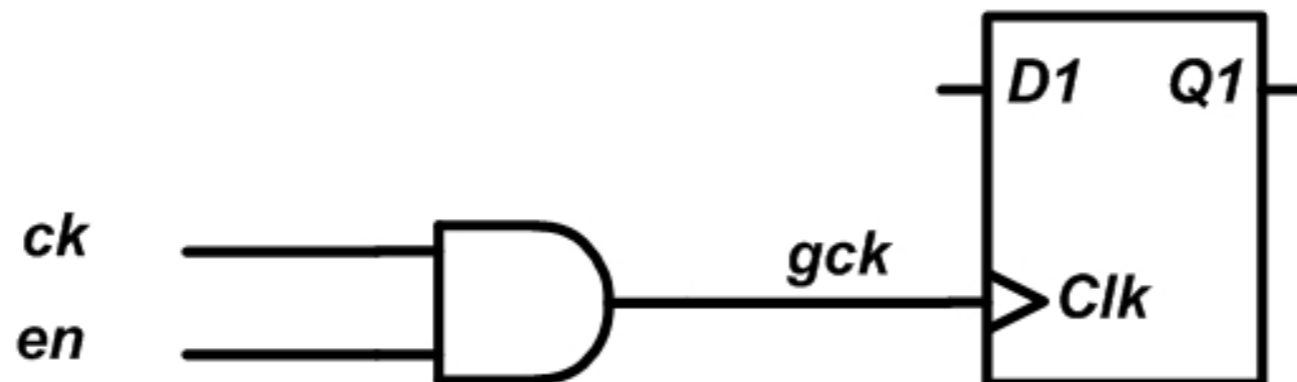
# Power Gating



- ▶ Idea: agregar transistores para “apagar” ciertos bloques de lógica que no se usan todo el tiempo.
  - ▶ P.ej: punto flotante.
  - ▶ Ojo, no se desactivan registros.
- ▶ Los transistores nuevos controlan la alimentación del circuito.
  - ▶ Baja mucho el consumo estático.
  - ▶ Estos transistores agregan retardos.
- ▶ Se suele usar también con bloques inactivos del caché.
  - ▶ Procesadores modernos pueden apagar núcleos completos.

# Clock Gating

- ▶ Estudios demostraron que gran parte del consumo de energía se producía en los registros (latches).
- ▶ Y que la mayoría de las veces, no cambiaban su valor con la llegada de un nuevo pulso de reloj.
- ▶ Se propone entonces enmascarar la señal de reloj con una compuerta AND.
  - ▶ Para habilitar/deshabilitar la llegada del clock.
- ▶ Esta técnica agrega compuertas que consumen energía, y agrega retardos.
  - ▶ Ahorra un 8% de energía en estado activo, y 35% en modo bajo consumo.





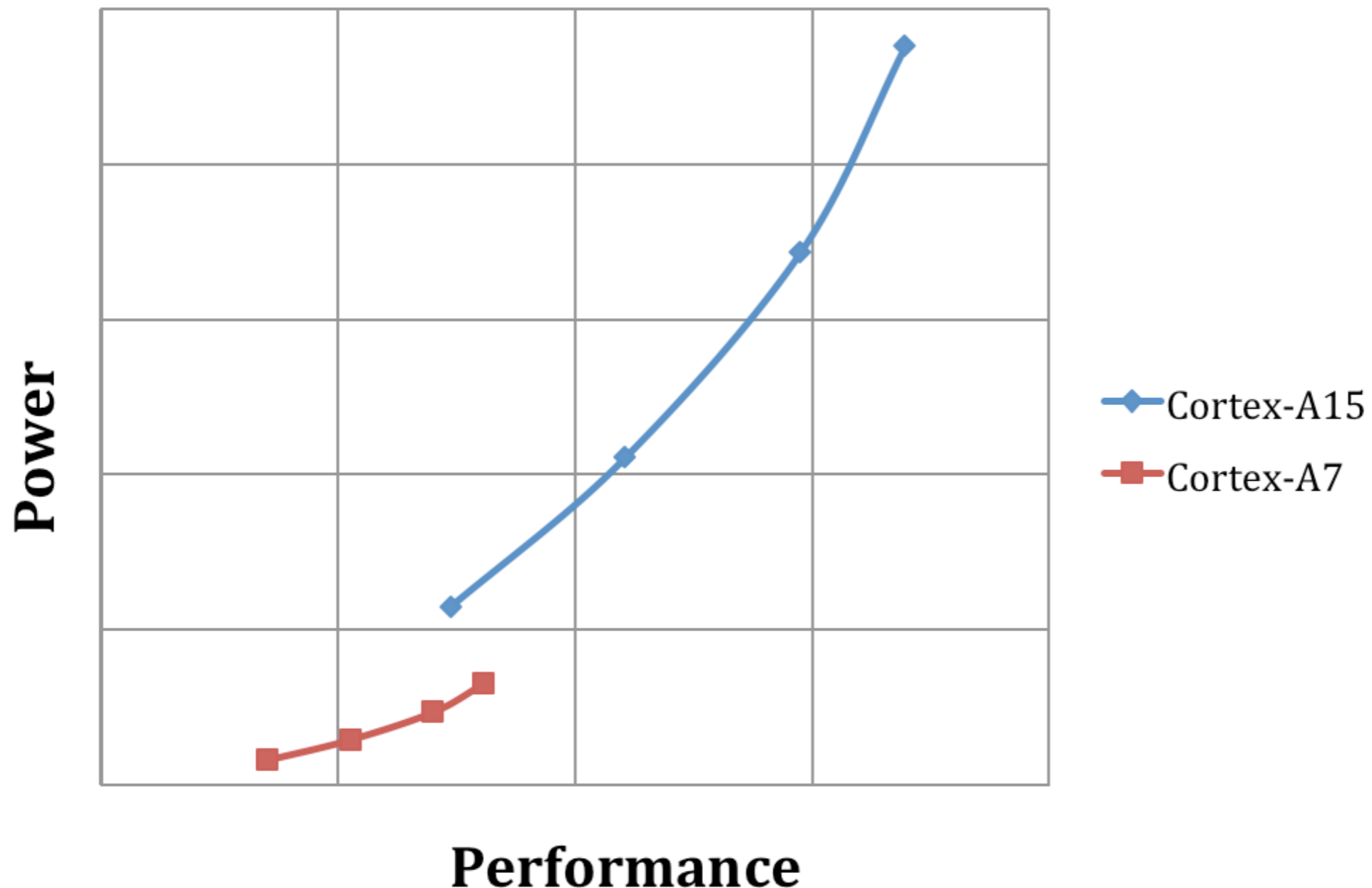
# Procesadores heterogéneos

---

- ▶ Técnica iniciada por ARM en 2011, denominada **big.LITTLE**.
  - ▶ Motivada por el uso típico de los smartphones: el 80% del tiempo en modo de muy bajo consumo, el 16% de uso medio y solamente el 4% de uso intensivo.
- ▶ Usa en un mismo chip dos núcleos: **uno grande de mayor performance y uno pequeño de menor consumo**.
  - ▶ Ambos son **implementaciones diferentes del mismo ISA**.
  - ▶ Inicialmente fueron un A15 y un A7. Luego un A76 con un A53.
- ▶ Si la aplicación no necesita performance, usa el núcleo pequeño; cuando la necesita, usa el grande.
  - ▶ Casi nunca se utilizan ambos núcleos al mismo tiempo.
- ▶ Se monitorea permanentemente la carga de trabajo, para mover la ejecución de un procesador a otro.
  - ▶ La migración tiene un costo, entre 30.000 y 50.000 ciclos. Evaluar si realmente conviene.
  - ▶ Al mover, se mueven las aplicaciones de usuario y también el SO.

# Procesadores heterogéneos

- ▶ Consumo de energía vs performance en la primera generación big.LITTLE.
- ▶ Es como una extensión de DVFS.



# ARM DynamIQ

---

- ▶ Evolución de big.LITTLE, empezó en 2017.
- ▶ Misma idea, pero con 8 núcleos, y flexibilidad a la hora de hacer las combinaciones:
  - ▶ 1 big + 3 LITTLE; 2 + 4; 1 + 7; 0 + 8; etc.
  - ▶ En versiones más recientes, 1+3+4 o 1+5+2.
    - ▶ Y soportan hasta 14 núcleos.
- ▶ Esta flexibilidad permite una mejor adaptación del procesador a la tarea que se busca realizar.
  - ▶ Para ciertas aplicaciones, 1+3 tienen mejor performance que 0+8, **a pesar de ser la mitad de núcleos.**
  - ▶ Consiguiendo una mejor performance/watt.
  - ▶ Volveremos sobre esto en el Tema siguiente.

# Procesadores heterogéneos actuales

---

- ▶ Por ejemplo, el chip Snapdragon 8+ Gen 2 de Qualcomm (2023), usado en el Samsung Galaxy S23 Ultra, posee 8 núcleos:
  - ▶ 1 núcleo X3 a 3,36 GHz (big).
  - ▶ 2 núcleos Cortex A715 a 2,80 GHz (big).
  - ▶ 2 núcleos Cortex A710 a 2,80 GHz (big).
  - ▶ 3 núcleos Cortex A510 a 2,00 GHz (LITTLE).
  - ▶ Nótese no sólo la configuración big.LITTLE, sino también las múltiples frecuencias.
- ▶ Apple tiene en su procesador M1 (2020) 8 núcleos: 4 de alta performance (Firestorm), y 4 de bajo consumo (Icestorm).
- ▶ Por otra parte, Intel comenzó a usar este tipo de configuración en 2021, en sus procesadores de 12ma generación (Alder Lake, con sus P-core y E-core).
  - ▶ En parte, porque Windows no ofrecía buen soporte para estos procesadores heterogéneos.
- ▶ AMD no posee un diseño de este tipo, hasta ahora.

# DVFS en procesadores actuales

---

- ▶ Los procesadores actuales poseen varios núcleos, y **cada uno de ellos trabaja con su propia frecuencia y nivel de tensión.**
  - ▶ Ajustados dinámicamente mediante DVFS.
- ▶ No todos los núcleos pueden trabajar a la frecuencia máxima al mismo tiempo.
  - ▶ El i9-9900KS de Intel (2019) fue el primer procesador en alcanzar 5 GHz en todos los núcleos, todo el tiempo.
- ▶ La limitación suele estar impuesta por el consumo de energía total deseado.
  - ▶ Si se activan más núcleos, consumen más energía, entonces se les baja la frecuencia para mantenerlos dentro del rango estipulado.
    - ▶ Se arman tablas de consumo y frecuencia por núcleo.
  - ▶ Si se agregan cachés, también se suele bajar la frecuencia.

# F máxima y consumo vs Cant. Núcleos

AMD Ryzen 9 5950X Per-Core Power																	
AnandTech	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0.1	20.6	16.3	20.1	19.7	16.8	16.2	15.2	13.9	12.7	11.4	10.1	9.1	8.2	7.5	6.8	6.4
2	0.0	0.1	16.4	18.9	19.8	16.5	16.3	15.2	14.1	13.0	11.6	10.3	9.3	8.4	7.6	6.9	6.5
3	0.0	0.1	0.1	20.3	19.7	17.5	17.0	16.0	14.7	13.3	11.8	10.4	9.3	8.4	7.6	6.9	6.5
4	0.3	0.1	0.1	1.3	19.8	16.4	17.0	15.7	14.8	13.5	12.0	10.6	9.5	8.6	7.8	7.1	6.6
5	0.2	0.1	0.1	1.5	1.3	16.8	16.3	16.1	14.7	13.3	11.8	10.4	9.3	8.3	7.6	6.9	6.5
6	0.2	0.1	0.1	1.2	1.3	0.1	16.2	15.1	14.8	13.5	12.0	10.6	9.5	8.5	7.8	7.1	6.6
7	0.2	0.8	0.5	1.6	1.9	0.8	0.7	15.6	14.4	13.0	11.6	10.2	9.2	8.3	7.6	6.9	6.4
8	0.1	0.1	0.1	1.3	1.2	0.1	0.1	0.9	14.3	13.1	11.7	10.4	9.4	8.4	7.7	6.9	6.5
9	0.1	0.0	0.0	1.3	1.2	0.0	0.0	0.0	0.3	11.1	10.0	8.9	8.0	7.3	6.7	6.2	5.8
10	0.1	0.0	0.1	1.1	1.0	0.0	0.0	0.0	0.1	0.0	9.8	8.6	7.9	7.2	6.6	6.1	5.7
11	0.1	0.0	0.1	1.1	1.1	0.1	0.0	0.0	0.0	0.1	0.0	8.8	8.0	7.4	6.8	6.2	5.8
12	0.0	0.0	0.0	1.3	1.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	7.9	7.2	6.7	6.2	5.7
13	0.3	0.1	0.1	1.4	1.4	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.0	7.3	6.7	6.2	5.8
14	0.0	0.0	0.0	1.1	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	6.6	6.0	5.8
15	0.2	0.1	0.1	1.4	1.3	0.0	0.1	0.1	0.0	0.4	0.2	0.2	0.2	0.5	0.5	6.2	5.8
16	0.1	0.0	0.1	1.3	1.2	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.1	0.2	0.2	5.7
Cores (W)	2	22	34	76	94	85	100	110	116	117	114	110	107	104	102	99	98
Package (W)	16	49	59	92	110	106	120	128	135	136	136	131	128	125	123	120	119
Freq	-	5050	4800	4750	4725	4675	4650	4625	4600	4450	4375	4250	4175	4050	3975	3850	3775

# Resumen: Técnicas reducción consumo

---

- ▶ Mejorar el proceso de fabricación.
- ▶ *Chiplets*: distintos productos con distintos procesos de fabricación.
- ▶ Paralelismo: dividir la tarea en múltiples núcleos, con menor frecuencia y tensión.
  - ▶ Permite elegir entre igual performance y menos consumo, o igual consumo y mayor performance.
- ▶ *Dynamic Voltage/Frequency Scaling (DVFS)*.
- ▶ Enmascaramiento (*Gating*):
  - ▶ *Power Gating*: apagar bloques lógicos cuando no se usen.
  - ▶ *Clock Gating*: para los elementos que mantienen estado.
- ▶ Procesadores heterogéneos: (big.LITTLE).

# Resumen: Cambio de paradigma

---

- ▶ El aumento de transistores se destinaba a mejorar la performance.
  - ▶ Ahora se busca mejorar la **performance por watt**.
  - ▶ Utilizar mejor esos transistores. Si una parte del procesador no se usa, apagarla.
- ▶ Antes se buscaba un único procesador cada vez más potente.
  - ▶ Ahora se busca obtener la performance mediante varios núcleos más pequeños trabajando juntos.
  - ▶ Que puedan adaptarse a diferentes perfiles de tarea (**especialización**).
  - ▶ Similar al cambio de CISC a RISC.
- ▶ Antes los procesadores tenían una única frecuencia y una única tensión de alimentación.
  - ▶ Ahora se manejan diferentes valores fuera de los caminos críticos.
  - ▶ Distintos núcleos con distintas frecuencias y distintas prestaciones.



# Agradecimientos

---

- ▶ Las diapositivas de este tema fueron basadas en las realizadas por el Ing. Daniel Cohen.
- ▶ A su vez inspiradas en las diapositivas tomadas del curso CS 194-6 de Universidad de Berkeley, John Lazzaro, 2008.