

Bloque I. METODOLOGÍAS Y HERRAMIENTAS DE DISEÑO

Bloque II. PRESTACIONES DE LOS SISTEMAS DIGITALES Y  
TÉCNICAS DE OPTIMIZACIÓN

Bloque III. CIRCUITOS DE PROCESADO DE DATOS

Bloque IV. ASPECTOS PRÁCTICOS DE DISEÑO

Tema 8: Optimización del consumo de potencia

Tema 9: Interfaces Digitales

Tema 10. Caso de aplicación: diseño de procesadores  
basados en RISC-V

1

Tema 8: Consumo de potencia

Bloque IV. ASPECTOS PRÁCTICOS DE DISEÑO

---

## Tema 8: Optimización del consumo de potencia



2



## Tema 8: Optimización del consumo de potencia

- Medida de la potencia disipada
- Nivel de tecnología
- Nivel de circuito
- Nivel lógico
- Nivel RT
- Nivel arquitectural
- Nivel de software




## ¿Por qué necesitamos reducir el consumo de potencia?


### Prestaciones

-  La potencia disipada por un circuito se convierte en calor.
-  El calor excesivo puede reducir las prestaciones del circuito.

### Empaquetado del chip

-  La disipación de calor en los chips encarece el empaquetado.

### Baterías

-  Los sistemas portables que dependen de baterías autónomas constituyen la principal razón de las técnicas de bajo consumo.



## Reducir el consumo de potencia: Baterías

- ❑ Baterías en aplicaciones móviles:
  - Duración (tiempo entre recargas)
  - Coste
  - Peso / Tamaño
- ❑ Posibilidad de usar fuentes de **energía alternativas**
- ❑ Posibilidad de introducir la electrónica en aplicaciones que tradicionalmente no la usaban
- ❑ Posibilidad de “**hacer móviles**” aplicaciones que tradicionalmente no lo eran



Ingeniería de  
- Diseño de S

Departamento de Electrónica y Electromagnetismo



5

## Reducir el consumo de potencia: Temperatura

- ❑ La energía eléctrica consumida por un circuito eléctrico es disipada fundamentalmente en forma de calor, lo que hace que su temperatura se eleve.



$$\Delta E_Q = \int_{t_0}^{t_1} V(t) \cdot I(t) \cdot dt$$



Ingeniería de Computadores  
- Diseño de Sistemas Digitales -

6

Departamento de Electrónica y Electromagnetismo



6

## Reducir el consumo de potencia: Temperatura

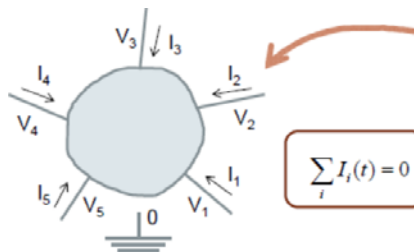
❑ El aumento de temperatura representa varios problemas:

- **Límite destructivo.** Límite a la complejidad máxima.
- Acortamiento de la **vida útil** del dispositivo.
- Acortamiento de la fiabilidad del dispositivo.
- Aumento del **coste** de las cápsulas (cerámica/metálica)
- Necesidad de **mecanismos de disipación** (disipadores, ventiladores, circulación de agua): coste, peso, tamaño.
- Pérdida/limitación de **velocidad** (empeoramiento de la respuesta temporal).



## Potencia eléctrica: Fundamentos

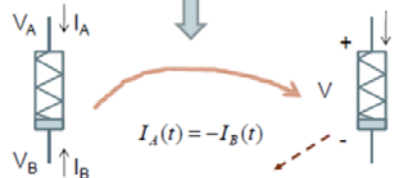
❑ Potencia instantánea entrante en un dispositivo eléctrico o electrónico:



$$\sum_i I_i(t) = 0$$

$$P(t) = \sum_i V_i(t) \cdot I_i(t)$$

→ Invariante con la "referencia de tensiones". Podemos elegir cualquiera de los terminales del dispositivo como terminal de referencia de tensiones



$$P(t) = V(t) \cdot I(t)$$

❑ La energía/potencia neta entrante en un dispositivo se invierte en:

- ✓ Almacenarla (elementos reactivos)
- ✓ Calor (elementos resistivos)
- ✓ Energía mecánica (motores)
- ✓ Energía radioeléctrica (antenas)
- ✓ Energía química (baterías)
- ✓ etc.

❑ En general se invertirá parcialmente en **varias formas de energía.**

## Potencia Instantánea y Potencia Promedio (dispositivo)

### • Potencia Instantánea:

$$P(t) = \sum_i V_i(t) \cdot I_i(t)$$

- Depende del circuito en que se encuentre el dispositivo, de las señales o estímulos externos, etc.
- **Diferente en cada instante.**

### • Potencia Promedio:

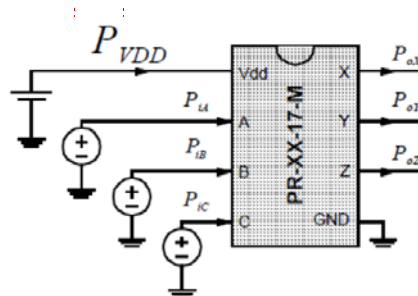
$$\bar{P} = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} P(t) \cdot dt$$

- También depende del circuito en que se encuentre el dispositivo, de las señales o estímulos externos, etc.
- En general, dependerá del intervalo de promediado  $[t_1, t_2]$ .
- Si el intervalo de promediado es "estadísticamente significativo", la medida será "significativa".



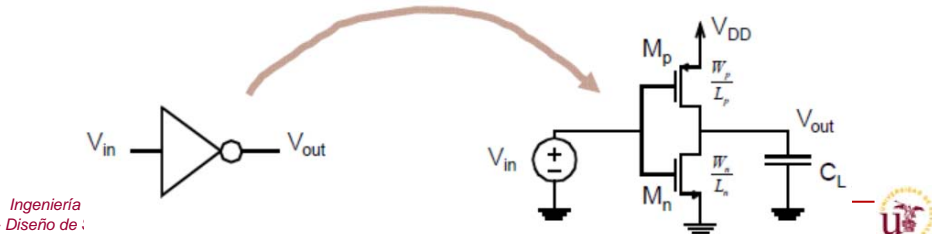
## Potencia Promedio a nivel de Sistemas

- ❑ En general, tanto en lo referente a las baterías como a la temperatura, lo significativo es la **potencia promedio**, calculada en un intervalo significativo.
- ❑ En promedios temporales significativos, los sistemas en los que estamos interesados (circuitos digitales) no almacenan energía. Tampoco producen energía mecánica, radioeléctrica, o química, en cantidades significativas.
- ❑ La energía eléctrica que entra en un sistema viene de las **fuentes de alimentación**, y en menor medida, de las fuentes que generan las **señales de entrada**.
- ❑ La energía suministrada al sistema se invertirá esencialmente en **calor** (se disipará). Una fracción, que a veces es significativa, es **transmitida por las señales de salida**.



## Potencia en un Inversor CMOS

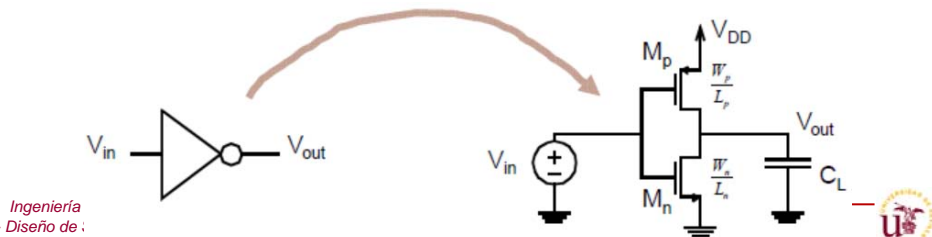
- ❑ Podemos calcular la potencia disipada en un sistema digital sumando la disipada en sus elementos.
- ❑ Los elementos básicos de los circuitos digitales son las puertas.
- ❑ La puerta más sencilla es el inversor. A partir de él, se estudian por extensión otras puertas más complejas.
- ❑ Es un modelo simple, al que pueden reducirse fácilmente la mayoría de los casos más complejos de puertas CMOS.



11

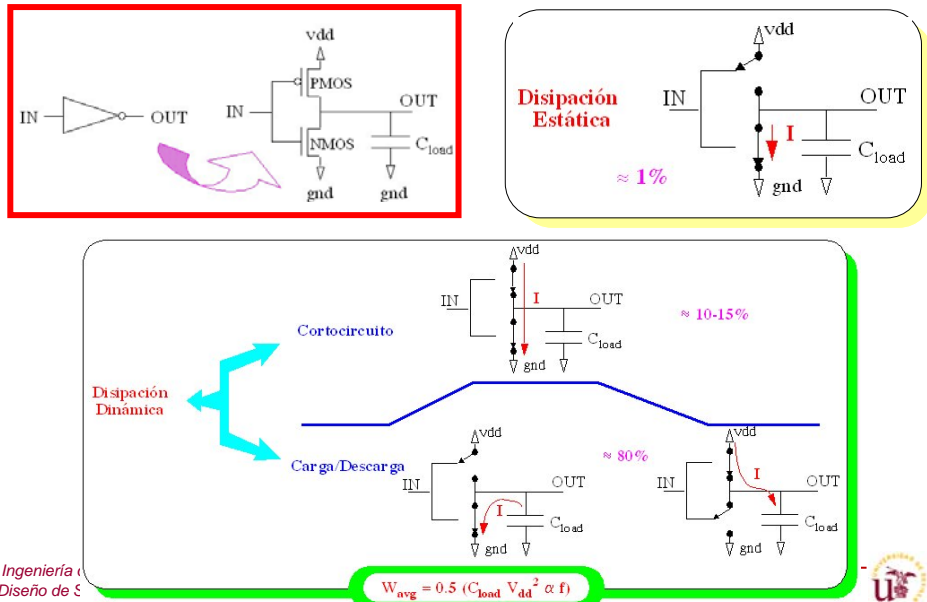
## Potencia en un Inversor CMOS

- ❑ La “impedancia” de entrada es **capacitiva**
- ❑ La capacidad de carga aglutina:
  - la capacidad **parásita del nudo de salida** del inversor
  - las capacidades de **entrada** de las puertas atacadas por el inversor
  - la capacidad del **cableado** del nudo de salida



12

## Consumo de potencia en circuitos CMOS



13

## Extensión a sistemas completos (CMOS “estáticos”)

- ❑ En un sistema digital, existe un gran número de puertas de **diversos tipos**, (funcionalidad, “fuerza”, *fan-in*, *fan-out*, etc). Cada una de ellas (y cada “instancia” de ellas) tendrá un **comportamiento diferente** en potencia estática, corriente de cortocircuito, y una capacidad de carga diferente.
- ❑ Se recurre a un **modelo de referencia**: **puerta equivalente**, que representa un comportamiento “promedio” de las puertas presentes en el sistema. Habitualmente es una NAND-2 o una NOR-2.
- ❑ La complejidad de los sistemas se mide habitualmente usando el “**número de puertas equivalentes**”,  $N_{pe}$ .
- ❑ No todas las puertas conmutarán el mismo número de veces en un determinado cálculo o intervalo de operación. Pero se puede considerar un promedio típico: **coeficiente de actividad promedio** ( $\eta$ )
- ❑ Los sistemas digitales complejos suelen ser síncronos, por lo que la **frecuencia** de reloj modulará directamente el número de conmutaciones por segundo.

14

## Extensión a sistemas completos (CMOS “estáticos”)

Diagram illustrating the components of the power consumption equation for CMOS static systems:

- n° de puertas equivalentes (Number of equivalent gates) →  $N$
- Potencia estática promedio (Average static power) →  $\overline{P}_S$
- fracción de nudos que conmutan en cada ciclo (Fraction of nodes that switch in each cycle) →  $\overline{\eta}$
- frecuencia de reloj (Clock frequency) →  $f_{ck}$
- energía de CC promedio (Average DC energy) →  $\overline{E}_{SC}$
- capacidad promedio de los nudos (Average capacitance of the nodes) →  $\overline{C}$

$$\overline{P}_{Sistema} = N \cdot \left\{ \overline{P}_S + \overline{\eta} \cdot f_{ck} \cdot (\overline{E}_{SC} + \overline{C} \cdot V_{DD}^2) \right\}$$

- Si despreciamos el consumo de cortocircuito y estático, solo se considera el consumo en la carga/descarga de condensadores.

$$\overline{P}_{Sistema} \cong N \cdot \overline{\eta} \cdot f_{ck} \cdot \overline{C} \cdot V_{DD}^2$$



## Tema 8: Optimización del consumo de potencia

- Medida de la potencia disipada
- Nivel de tecnología
- Nivel de circuito
- Nivel lógico
- Nivel RT
- Nivel arquitectural
- Nivel de software





## Técnicas de baja potencia

- ❑ La optimización del consumo de potencia (durante la fase de diseño del circuito) está muy condicionada por
  - el contexto de diseño
    - ✓ FPGA
    - ✓ ASIC *semi-custom*
    - ✓ ASIC *full-custom*
    - ✓ disponibilidad de librerías de celdas específicas
    - ✓ tecnología fija
    - ✓ tecnología elegible
  - las especificaciones
    - ✓ funcionalidad/complejidad
    - ✓ frecuencia/velocidad mínima
    - ✓ tensión de alimentación



## Técnicas de baja potencia

- ❑ Una buena optimización en potencia se debe abordar a diferentes niveles:
  - ✓ Nivel de tecnología
  - ✓ Nivel de circuito
  - ✓ Nivel lógico
  - ✓ Nivel RT
  - ✓ Nivel arquitectural
  - ✓ Nivel de software
- ❑ Con frecuencia, los mejores resultados se obtienen con una combinación de actuaciones a todos los niveles.



## Técnicas de baja potencia

- ❑ Es primordial, lógicamente, focalizarse en los **términos dominantes del consumo** (subsistemas concretos, nodos concretos, etc.)
- ❑ El objetivo global se resume en minimizar la **complejidad** del sistema (N) y el **número de conmutaciones** (de una capacidad “unitaria” equivalente) **por unidad de tiempo** ( $N\eta f_{ck}$ )

$$\overline{P_{Sistema}} \cong N \cdot \overline{\eta} \cdot f_{ck} \cdot \overline{C} \cdot V_{DD}^2$$



## Nivel de tecnología

- ❑ Elegir la tecnología más avanzada posible (la de mayor resolución):
  - Menores **capacidades** en los nudos ( $<C$ )
  - Menor **tensión** de alimentación ( $<V_{DD}$ )
- ❑ Aumentar las tensiones umbrales, o usar las más altas disponibles
  - 😊 Reduce la **potencia estática** y la de **corto-circuito**
  - 😞 No afecta a la **potencia de conmutación**
  - 😞 Reduce la **velocidad**
  - 😞 Reduce los **márgenes de ruido**

$$\overline{P_{Sistema}} \cong N \cdot \overline{\eta} \cdot f_{ck} \cdot \overline{C} \cdot V_{DD}^2$$



## Nivel de tecnología

- ❑ Usar tensión de alimentación más baja que la máxima permitida por la tecnología ( $<V_{DD}$ )
  - 😊 Reduce la **potencia estática**, la de **corto-circuito**, y la de **conmutación**
  - 😞 Reduce la **velocidad**
  - 😞 Reduce los **márgenes de ruido**
- ❑ No usar **frecuencias de reloj** (velocidad de procesamiento) más altas de lo que sea estrictamente necesario para la aplicación ( $<f_{ck}$ )
- ❑ No diseñar para **velocidades** mayores que las necesarias (si ello supone aumentar “C”)

$$\overline{P_{Sistema}} \cong N \cdot \overline{\eta} \cdot f_{ck} \cdot \overline{C} \cdot V_{DD}^2$$



## Nivel de circuito

- ❑ Reducción de la **potencia de conmutación**
  - No usar celdas de más “fuerza” que la necesaria, para evitar aumentar las **capacidades parásitas**. Usar celdas de poca “fuerza” en caminos no-críticos.
  - Si la capacidad está dominada por la **capacidad de cableado**, tratar de reducirla con un emplazamiento mejor de los bloques, o partiendo el nudo introduciendo modificaciones a nivel lógico
  - Si la capacidad está dominada por las **capacidades de entrada** de los bloques (fan-out alto) tratar de dividir el nudo en varios más pequeños, introduciendo modificaciones a nivel lógico.

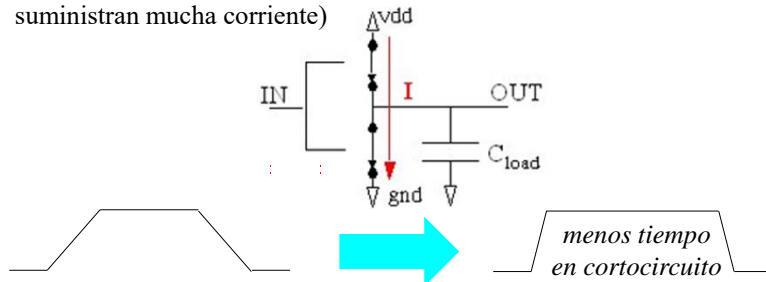
$$\overline{P_{Sistema}} \cong N \cdot \overline{\eta} \cdot f_{ck} \cdot \overline{C} \cdot V_{DD}^2$$



## Nivel de circuito

### Reducción de la potencia por corriente de cortocircuito

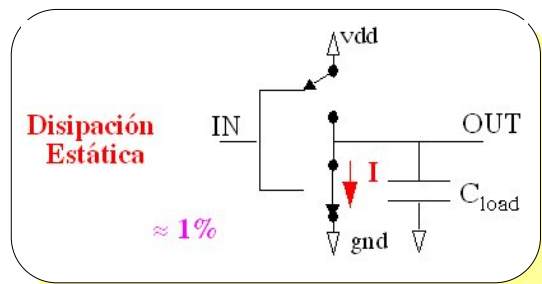
- Aumentar las **pendientes** de las señales en los nudos de los circuitos, pero no excesivamente (compromiso con potencia de conmutación). Idealmente, las pendientes deben ser similares en todos los nudos.
- Evitar **señales lentas** a la entrada de buffers o puertas “potentes” (que suministran mucha corriente)



## Nivel de circuito

### Reducción de la potencia estática

- Usar transistores/celdas con **tensiones umbrales altas** (tecnologías con múltiples tensiones umbrales)
- Usar transistores con **longitudes mayores** que las mínimas
- **Desconectar la alimentación** de bloques de circuitos inactivos (**power gating**).



## Tema 8: Optimización del consumo de potencia

- Medida de la potencia disipada
- Nivel de tecnología
- Nivel de circuito
- Nivel lógico
- Nivel RT
- Nivel arquitectural
- Nivel de software

Se intenta reducir la actividad de conmutación

$$\overline{P}_{Sistema} \cong N \cdot \overline{\eta} \cdot f_{ck} \cdot \overline{C} \cdot V_{DD}^2$$



## Tema 8: Optimización del consumo de potencia

- Medida de la potencia disipada
- Nivel de tecnología
- Nivel de circuito
- Nivel lógico
- Nivel RT
- Nivel arquitectural
- Nivel de software



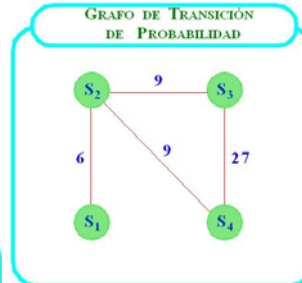
## Nivel lógico: Codificación

L. Benini, G. De Micheli: "State Assignment for Low Power Dissipation", IEEE Journal of Solid-State Circuits, vol. 30, no. 3, pp. 258-268, March 1995.



Dos estados con un alto número de transiciones tienen códigos con distancia de un bit.

Se aplica un asignamiento de código que reduce la actividad de conmutación de las variables de estado en las transiciones.



⊗ En algunos casos la parte combinacional de la FSM es más compleja.



## Nivel lógico: Codificación

P. Surti, L.F. Chao, A. Tyagi: "Low Power FSM Design Using Huffman-Style Encoding", IEEE European Design and Test Conference (EDTC'97), pp. 521-525, 1997

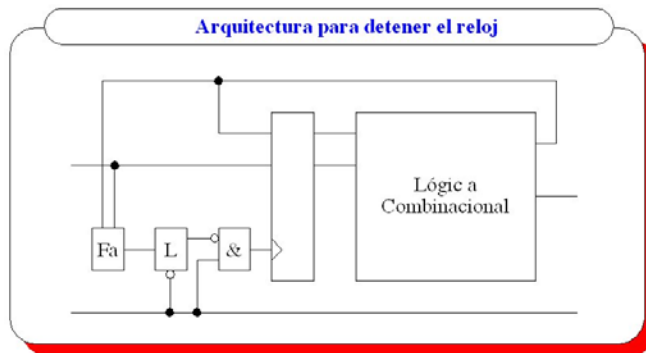
➤ Aplican codificación Huffman.

➤ El conjunto de estados con más probabilidad es codificado de  $\lceil \log_2 ns \rceil$  bits. ( $ns$  es el número de estados).

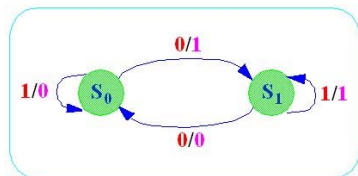
➤ La FSM adicional requiere lógica adicional para parar el reloj del subconjunto de registros cuando el estado presente tiene una longitud de código menor.



## Nivel lógico: Detener el reloj



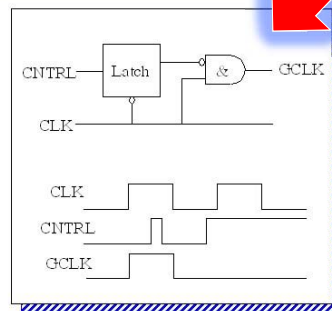
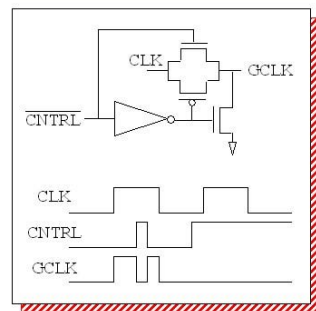
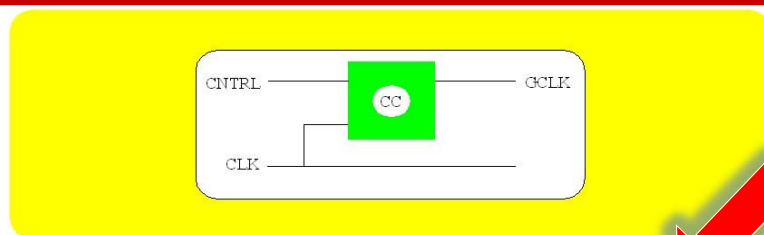
La **función de activación** Fa detecta cuando la FSM está **internamente inactiva** (no realiza transiciones de estado) y detiene el reloj.



Para el valor de entrada **1** la FSM está internamente inactiva

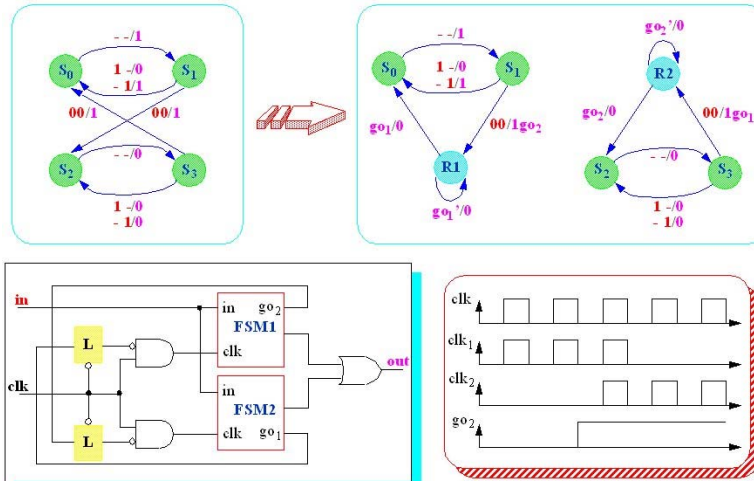


## Nivel lógico: Detener el reloj



## Nivel lógico: Particionado de FSM

L. Benini, G. De Micheli, F. Vermeulen: "Finite-State Machine Partitioning for Low Power", Int. Symp. on Circuits and Systems (ISCAS98), Monterey, California, June 1998.



## Nivel lógico: Pre-cálculo

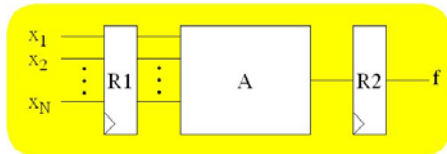
- ☒ Las salidas se pre-calculan en el ciclo anterior con objeto de reducir la actividad de conmutación.
- ☒ Si una salida puede ser pre-calculada entonces se puede “inhabilitar” el circuito lógico.
- ☹ Hay que establecer un compromiso entre la complejidad del circuito de pre-cálculo y la reducción del consumo de potencia.





## Nivel lógico: Pre-cálculo

M. Alidina, J. Monteiro, S. Devadas, A. Ghosh, M. Papaefthymiou: "Precomputation-Based Sequential Logic Optimization for Low-Power", IEEE Trans. on Very Large Scale Integration (VLSI), vol. 2, no. 4, pp. 426-436, Dec. 1994.



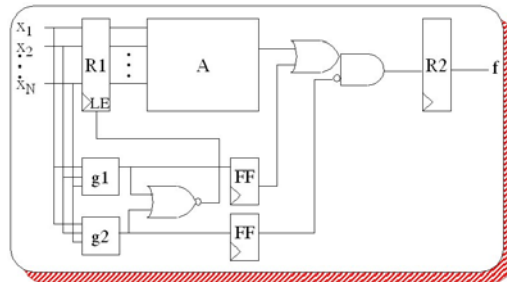
$g1$  y  $g2$  son las funciones **predictoras**:

$$g1 = 1 \Rightarrow f = 1$$

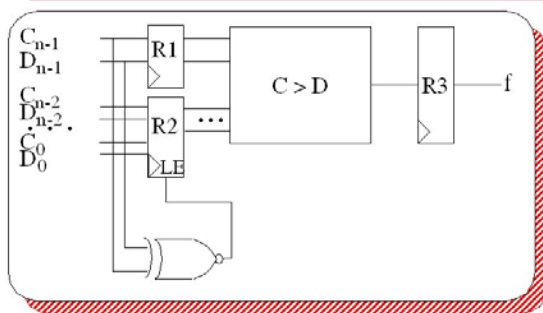
$$g2 = 1 \Rightarrow f = 0$$

$$g1 \wedge g2 = 0$$

- ⊖ El área se incrementa.
- ⊖ El retraso entre R1 y R2 se incrementa.
- ⊖ El retraso anterior a R1 se incrementa.



## Nivel lógico: Pre-cálculo. Ejemplo



La lógica de precálculo es

$$g1 = C(n-1) \wedge \overline{D(n-1)}$$

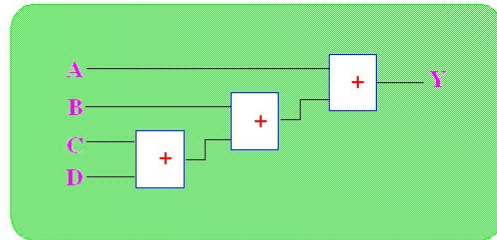
$$g2 = \overline{C(n-1)} \wedge D(n-1)$$

$$\left. \begin{array}{l} \text{Si } g1 = 1 \Rightarrow C > D \\ \text{Si } g2 = 1 \Rightarrow C < D \end{array} \right\} \rightarrow g1 \vee g2 = C(n-1) \oplus D(n-1)$$

**¡Para valores grandes de  $n$  se consigue una reducción en el consumo cercana al 50%!**

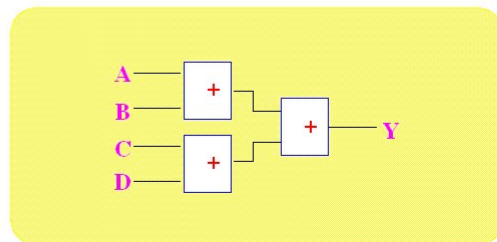
## Nivel lógico: Balancear los caminos

$$Y \leq A + B + C + D;$$



CÁSCADA VERSUS ARBOL

$$Y \leq (A + B) + (C + D);$$



## Nivel lógico: Retiming

J. Monteiro, S. Devadas, A. Ghosh: "Retiming Sequential Circuits for Low Power", IEEE Conf. on Computer-Aided Design (ICCAD'93), pp. 398-402, 1993.

- ☒ Identificar los nodos con una alta actividad de azar y una alta capacidad de carga.
- ☒ En ese nodos se colocan flip-flops.
  - ☺ Los *flip-flop* sólo capturan el último dato antes del flanco de reloj.
  - ☺ Sólo permiten propagar una transición al nudo con alta capacidad de carga.
- ☒ Restricciones al *retiming*:
  - ☒ El tiempo de ciclo del circuito no debe incrementarse.
  - ☒ Se debe especificar un límite superior en el número de registros.
- ☹ Es difícil estimar la actividad de conmutación en un circuito sin datos del *layout*.
- ☹ Cuando se mueve un *flip-flop* desaparecen los *glitches* en su salida pero se pueden generar nuevo *glitches* ya que la temporización de la parte combinacional ha cambiado.



## Tema 8: Optimización del consumo de potencia

- Medida de la potencia disipada
- Nivel de tecnología
- Nivel de circuito
- Nivel lógico
- Nivel RT
- Nivel arquitectural
- Nivel de software



## Nivel RT: Aislar operandos

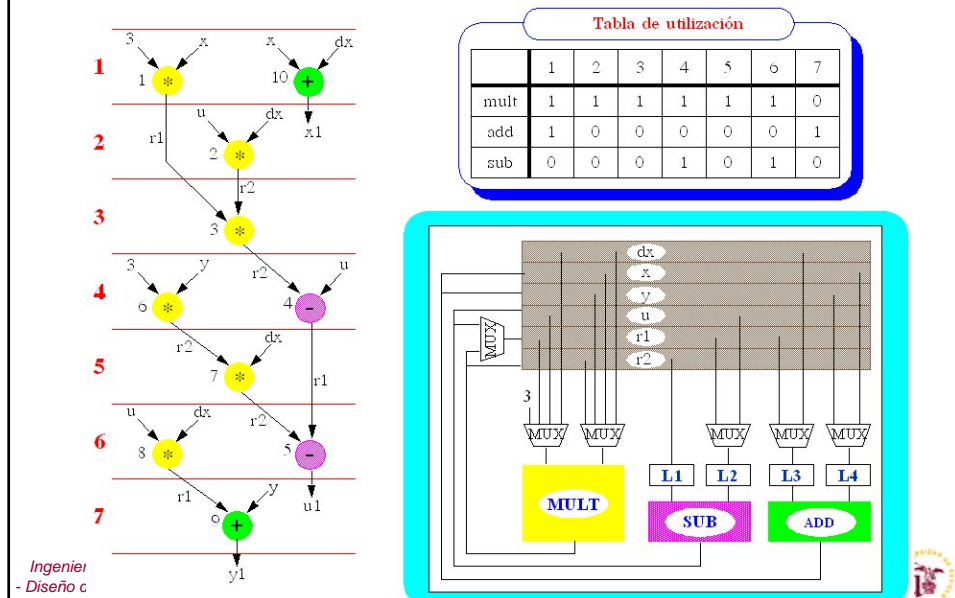
- ❑ Una unidad funcional desocupada recibe valores en sus entradas que son procesados pero el resultado no es observado ni almacenado.
- ❑ En esta técnica se emplea una **tabla de utilización** que informa de la ocupación de las unidades funcionales en cada ciclo de reloj

### Algoritmo

- ✓ Construir la tabla de utilización
- ✓ Si la utilización de cada unidad es menor que un parámetro de usuario ( $R_p$ ):
  - ✗ se insertan *latches* en cada entrada de la unidad funcional
  - ✗ la lógica de control activa o no los *latches* dependiendo de la utilización.



### Nivel RT: Aislar operandos



39

### Nivel RT: Re-especificación del control

A. Raghunathan, et. al.: "Controller Re-Specification to Minimize Switching Activity in Controller/Data Path Circuits". Int. Symp. on Low Power Electronics and Design, pp. 301-304, Aug. 1996.

- La actividad de las unidades desocupadas se puede eliminar escogiendo los valores de control de la lógica de conexión (multiplexores y *drivers* tri-estado).
- Los valores de control de la lógica de conexión suelen ser *don't care* en la FSM cuando la unidad no se utiliza.
- ⊗ Si una señal de control tiene transiciones azarosas en un ciclo desocupado entonces no tiene efecto los beneficios de la re-especificación del control.

40

## Nivel RT: Segmentación de la memoria

A.H. Farrahi, G.E. Tellez, M. Sarrafzadeh: "Memory Segmentation to Exploit Sleep Mode Operation". Proc. Design Automation Conf., pp. 36-41, June 1995.

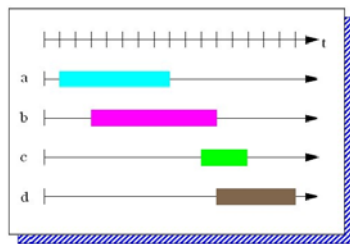
- Las variables se almacenan en registros.
- Una memoria está desocupada cuando no almacena información útil.

### Algoritmo

- ✓ Los recursos de memoria se dividen en **segmentos de memoria** (*clusters*).
- ✓ Cuando un segmento está desocupado se pone en modo dormido (*sleep mode*), o sea los relojes se paran o bien se inhabilita la señal de refresco.



## Nivel RT: Segmentación de la memoria



- ✓ Caso de todos los registros en el mismo segmento (memoria monolítica)

- ✓ El segmento sólo está desocupado en el primer ciclo.



Dos segmentos de memoria S1 y S2

$a, b \in S1; c, d \in S2$

$$I = I(S1) + I(S2) = 6 + 10 = 16$$

$$T = T_{IS} + T_{SI} = 2 + 2 = 4$$

$a, c \in S1; b, d \in S2$

$$I = I(S1) + I(S2) = 6 + 3 = 9$$

$$T = T_{IS} + T_{SI} = 3 + 3 = 6$$

$a, d \in S1; b, c \in S2$

$$I = I(S1) + I(S2) = 4 + 6 = 10$$

$$T = T_{IS} + T_{SI} = 3 + 3 = 6$$



## Tema 8: Optimización del consumo de potencia

- Medida de la potencia disipada
- Nivel de tecnología
- Nivel de circuito
- Nivel lógico
- Nivel RT
- Nivel arquitectural
- Nivel de software

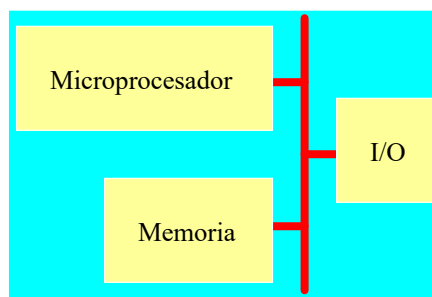


43

### Nivel arquitectural

❑ Cuando la arquitectura del sistema se basa en un procesador la carga capacitiva de las entradas/salidas es mayor (en más de tres ordenes de magnitud) que la de los nudos internos del procesador.

❑ El consumo de las memorias aumenta con su tamaño.



La reducción del consumo se consigue minimizando las transiciones de los buses conectados al procesador.



44

## Nivel arquitectural

Básicamente existen dos conjuntos de técnicas:

- Técnicas de codificación de bus.
- Técnicas de organización de la memoria.



## Nivel arquitectural. Codificación

- ✓ La potencia disipada en los buses dependen de la **distancia Hamming** entre datos sucesivos.
- ✓ La distancia Hamming mide la actividad de conmutación de los buses.
- ✓ En los algoritmos secuenciales hay una alta correlación en las direcciones.

Binary	Hamming distance	Gray	Hamming distance	One-hot	Hamming distance
00000000	-	00000000	-	000000000000000001	-
00000001	1	00000001	1	000000000000000010	2
00000010	2	00000011	1	0000000000000000100	2
00000011	1	00000010	1	00000000000000001000	2
00000100	3	00000110	1	000000000000000010000	2
00000101	1	00000111	1	0000000000000000100000	2
00000110	2	00000101	1	00000000000000001000000	2
00000111	1	00000100	1	000000000000000010000000	2
00001000	4	00001100	1	000000000000000000000	2
00001001	1	00001101	1	0000000000000000000000	2
00001010	2	00001111	1	0000000000000000000000	2
00001011	1	00001110	1	0000000000000000000000	2
00001100	3	00001010	1	0000000000000000000000	2
00001101	1	00001011	1	0000000000000000000000	2
00001110	2	00001001	1	0000000000000000000000	2
00001111	1	00001000	1	0000000000000000000000	2
00010000	5	00011000	1	0100000000000000000000	2
00010001	1	00011001	1	1000000000000000000000	2

☹ *One-hot* es la peor solución ya que requiere buses mayores e incrementa la actividad de conmutación promedio.

☹ La codificación binaria da el peor pico de actividad de conmutación.

😊 El código Gray es el mejor con una reducción en la actividad promedio del 50% y el menor pico de actividad.



## Nivel arquitectural. Codificación: *Bus-Invert*

M. R. Stan, W. P. Burleson: "Bus-Invert Coding for Low-Power I/O". IEEE Trans. on Very Large Scale Integration (VLSI) Systems, vol. 3, no. 1, pp. 49-58, March 1995.

- ❑ Se aplica a buses de datos
- ❑ Requiere una señal de control adicional denominada *invert*.

### Algoritmo

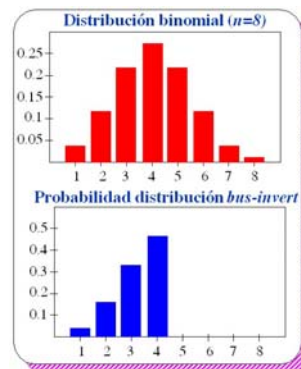
- ✍ Si la distancia Hamming entre dos valores del bus (incluida la señal *invert*) es mayor que  $n/2$  ( $n$  es la anchura del bus) entonces poner *invert=1* y transmitir el dato invertido.
- ✍ En caso contrario poner *invert=0* y transmitir el dato.
- ✍ El receptor debe invertir o no el dato dependiendo del valor de la señal *invert*.



## Nivel arquitectural. Codificación: *Bus-Invert*

- 😊 Reduce la actividad de conmutación de pico al 50%.
- 😊 Para buses de datos (sin datos correlacionados) la actividad de conmutación promedio se reduce un 20%, pero crece con la anchura de los buses.
- 😞 La codificación *bus-invert* requiere lógica adicional en ambos extremos del bus.

Binary	Hamming distance	Bus-invert	INV	Hamming distance
0010010010000001	-	0010010010000001	0	-
0010011100001100	6	0010011100001100	0	6
1001001100010010	8	1001001100010010	0	8
0100010000010000	7	0100010000010000	0	7
0000111110000100	7	0000111110000100	0	7
001100000100011	11	110011111011100	1	6
0111000010001000	6	100011110111011	1	6
010100000111110	6	101011111001001	1	7
0001010101000000	9	111010101011111	1	8
0010010101110001	5	1101101010001110	1	5
1100101110001010	13	1100101110001010	0	4
0010100001010101	12	1101011110101010	1	5





## Nivel arquitectural. Codificación: T0

L. Benini, G. De Micheli, E. Macii, D. Sciuto, C. Silvano: "Address Bus Encoding Techniques for System-Level Power Optimization". DATE-98: IEEE Design Automation and Test Europe, pp.861-866. Paris, France, Feb. 1998.

- ❑ Para codificación de direcciones en algoritmos secuenciales.
- ❑ Requiere una señal de control extra denominada **INC**.

### Algoritmo

✍ Para direcciones consecutivas la señal **INC=1** y el dato del bus no se modifica. El receptor calcula la nueva dirección incrementando la anterior.

✍ En caso contrario **INC=0** y se transmite el nuevo dato en el bus.

Binary	T0	INC
00000000	00000000	0
00000001	00000000	1
00000010	00000000	1
00000011	00000000	1
00000100	00000000	1
00110000	00110000	0
00010010	00010010	0
00010011	00010010	1
00010100	00010010	1
00010101	00010010	1



## Nivel arquitectural. Codificación: Beach Solution

L. Benini, G. De Micheli, E. Macii, M. Poncino, S. Quer: "Power Optimization of Core-Based Systems by Address Bus Encoding". IEEE Trans. on Very Large Scale Integration (VLSI) Systems, vol. 6, no. 4, pp.554-562. Dec. 1998.

- ❑ Se aplica en buses de direcciones.
- ❑ Se basa en la observación que direcciones consecutivas suelen tener una alta correlación de bloque de memoria.
- ❑ Para procesadores basados en memoria segmentada/paginada los salto intra-segmentos/página son más frecuentes que los inter-segmentos/página.

### Algoritmo

✍ Partiendo de trazas del bus se recolecta información estadística identificando posible correlación de bloque.

✍ Las líneas del bus se agrupan en *clusters* de acuerdo con la correlación (líneas del mismo *cluster* tienen alta correlación).

✍ Para cada *cluster* se genera una función de codificación adecuada.



## Nivel arquitectural. Organización de la memoria

### Ancho de banda de la memoria

Es importante minimizar el número de accesos a la memoria externa y su tamaño.

- ✍ Incrementando la densidad de código.
- ✍ Reduciendo la frecuencia de las operaciones de lectura y escritura.
- ✍ Mejorando las características de la caché *on-chip*.

### Características de la memoria caché

- Cachés grandes consumen más que las pequeñas.
- Se consigue más eficiencia mediante múltiples niveles de caché *on-chip*.



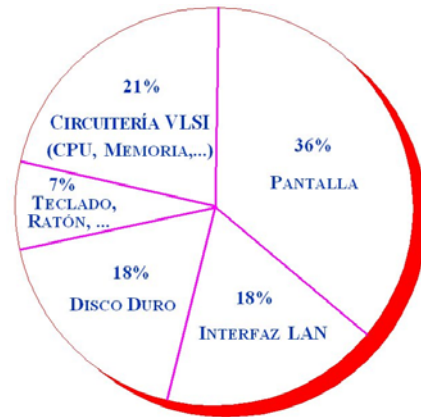
## Tema 8: Optimización del consumo de potencia

- Medida de la potencia disipada
- Nivel de tecnología
- Nivel de circuito
- Nivel lógico
- Nivel RT
- Nivel arquitectural
- Nivel de software



## Nivel de software

### Consumo de potencia en un ordenador



**¡ Reducir el consumo en la circuitería digital en un orden de magnitud 10x significa reducir el consumo global en menos del 19%!**



## Nivel de software

### Apagar el sistema

- ✓ Se basa en apagar la parte del sistema que está desocupada.
- ✓ **Técnicas de apagado no-predictivas:** se basan en apagar el sistema después que el procesador esté desocupado durante un cierto intervalo de tiempo. Este intervalo es fijo o bien lo fija el sistema operativo.
- ✓ **Técnicas de apagado predictivas:** la duración del tiempo desocupado se puede predecir basándose en la historia.

### Particionado del sistema

- ✓ **Codiseño Hardware&Software** incluye particionado Hardware&Software.
- ✓ Particionado Hardware&Software decide que componente del sistema se realizará en hardware y que parte en software.

### Selección de algoritmos

- ✓ La elección del algoritmo usado por la aplicación puede tener una influencia muy importante en el consumo de potencia.



## Control de tensiones de alimentación

### Adaptive Voltage Scaling (AVS)

- El diseño se realiza en condiciones de peor caso temporal (proceso *slow-slow*, temperatura máxima), asumiendo una tensión de alimentación igual a la máxima permitida.
- Una circuitería específica replica un camino de retraso crítico. El retraso de este circuito es monitorizado y comparado con el retraso requerido. Cuando el retraso es menor/mayor de lo necesario, se reduce/aumenta la tensión de alimentación gradualmente. El control suele estar realizado en hardware, dentro del propio sistema. Se fija un “punto de arranque” del algoritmo.
- La frecuencia de reloj permanece constante, al valor necesario o previsto. Cuando las condiciones de proceso (para la muestra concreta) o de temperatura son favorables, se ahorra potencia por la reducción de la tensión de alimentación, sin pérdida de velocidad.

### Dynamic Clock Frequency & Voltage Scaling (DCVS)

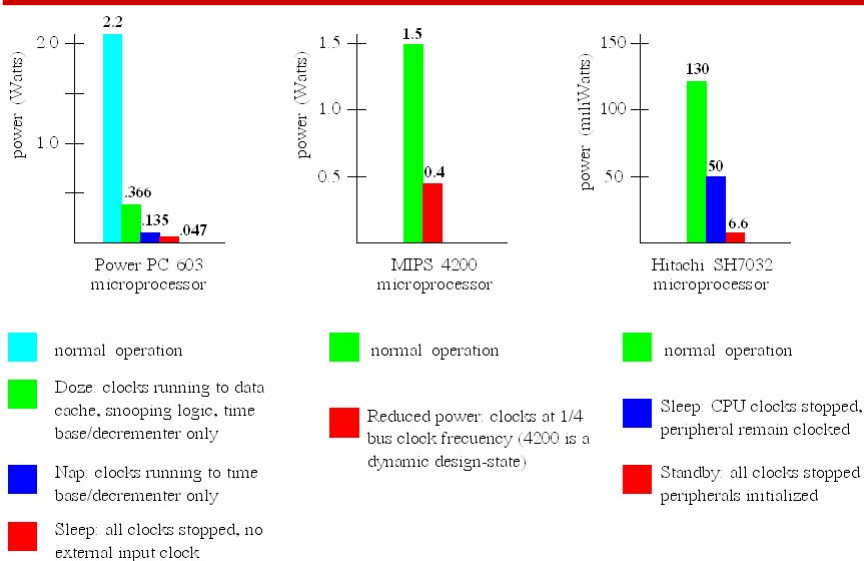
- El sistema de control (generalmente software) reduce el nivel de tensión de alimentación y la frecuencia de reloj, según una tabla precalculada, cuando la velocidad de procesamiento puede reducirse. Cuando las circunstancias permiten (o nos obligan a) reducir la velocidad, se ahorra potencia por la combinación de ambos factores.

### Static voltage scaling (SVS)

- Diseño realizado para varios modos de funcionamiento (un conjunto reducido), cada uno de ellos caracterizado por unas especificaciones temporales (frecuencia) y un valor de tensión de alimentación. El diseño debe asegurar la temporización en los peores casos de proceso y temperatura (como es habitual), en cada uno de los modos predefinidos. El sistema de control (generalmente software) selecciona el modo de operación según las circunstancias.

55

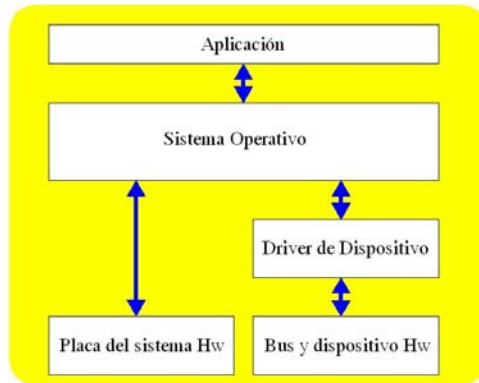
## Nivel de software: Apagado del sistema



56

## Nivel de software: OnNow

[http://msdn.microsoft.com/library/default.asp?url=/library/en-us/power/base/onnnow\\_power\\_management.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/power/base/onnnow_power_management.asp)



**Trabajando:** El sistema está en ON y operativo.

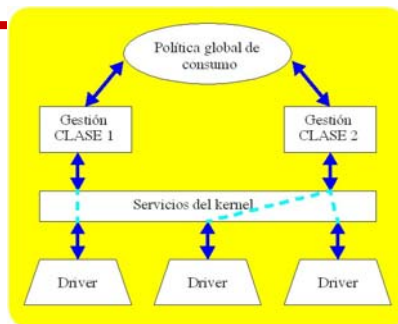
**Dormido:** El sistema parece estar OFF. El tiempo de vuelta al estado **trabajando** es inversamente proporcional al consumo de potencia.

**Soft off** El sistema parece estar OFF. El consumo es muy bajo. Se requiere rebotar el OS para volver al estado **trabajando**. Durante el rebote el contexto de trabajo se restaura de memoria no volátil.

**OFF** Sistema apagado y sin consumo (se ha desconectado la alimentación).



## Nivel de software: OnNow



➤ La política de consumo es el procedimiento que decide cuando la máquina debe pasar de un estado de consumo a otro.

➤ Las aplicaciones deben diseñarse para que sean eficientes desde el punto de vista de consumo.

```

do {
    if (TestMessageReady()) {
        done = ProcessMessage();
    } else if (BckgProcReq()) {
        DoBckgProc();
    }
} while (!done);
  
```

Aplicación para  
consumo eficiente

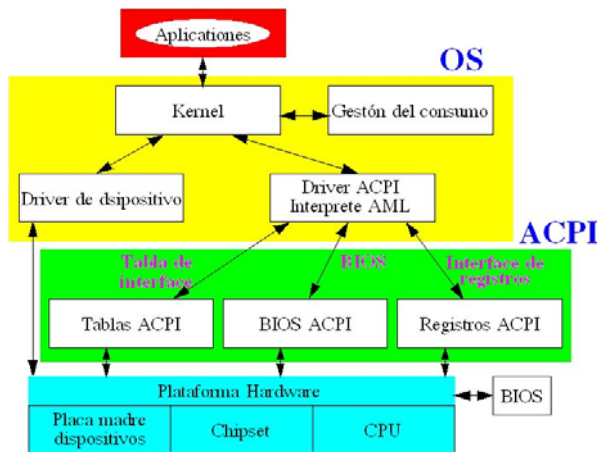
```

do {
    if (TestMessageReady()) {
        done = ProcessMessage();
    } else if (BckgProcReq()) {
        DoBckgProc();
    } else {
        WaitMessage();
    }
} while (!done);
  
```



## Nivel de software: ACPI

ACPI home page: <http://www.acpi.info/>



➤ ACPI habilita al OS para controlar el consumo de cada dispositivo conectado en el sistema.

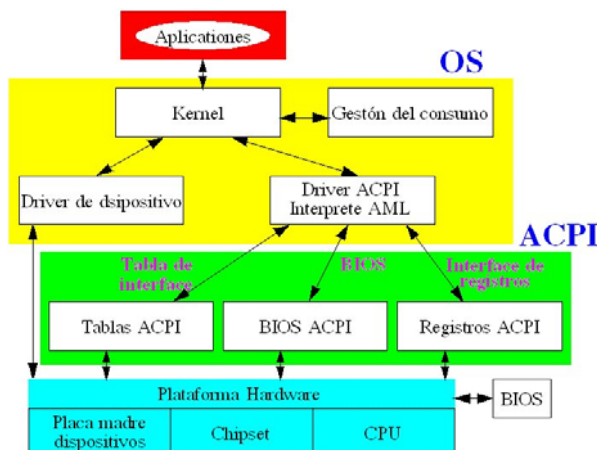
➤ ACPI integra las funciones de gestión de consumo en las rutinas de bajo nivel que interaccionan directamente con el hardware (firmware and BIOS)

➤ La especificación ACPI define las interfaces entre el OS y el hardware.



## Nivel de software: ACPI

ACPI home page: <http://www.teleport.com/~acpi/>



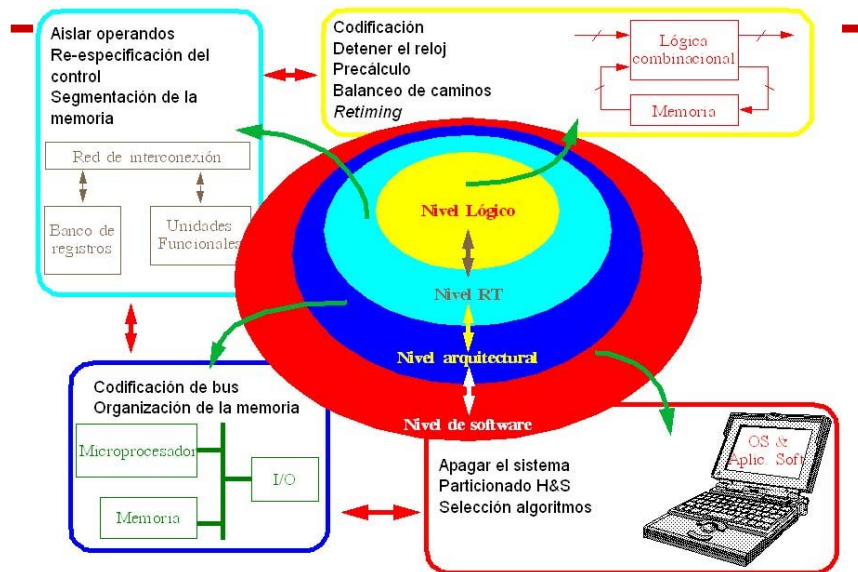
➤ ACPI habilita al OS para controlar el consumo de cada dispositivo conectado en el sistema.

➤ ACPI integra las funciones de gestión de consumo en las rutinas de bajo nivel que interaccionan directamente con el hardware (firmware and BIOS)

➤ La especificación ACPI define las interfaces entre el OS y el hardware.



## Diseño para bajo consumo de potencia



## Referencias

- Neil H. E. Weste and David Money Harris: "CMOS VLSI Design. A Circuits and Systems Perspective", Addison-Wesley, 2011
- Christian Piguet: "Low-Power CMOS Circuits. Technology, Logic Design and CAD Tools", Taylor & Francis, 2006
- L. Benini, G. De Micheli: "Dynamic Power Management. Design Techniques and CAD Tools". Kluwer Academic Pub., 1998.
- J.M. Rabaey, M- Pedram (editors): "Low Power Design Methodologies", Kluwer Academic Pub., 1996.
- A.P. Chandrakasan, R.W. Brodersen: "Low Power Digital CMOS Design", Kluwer Academic Pub., 1998.

