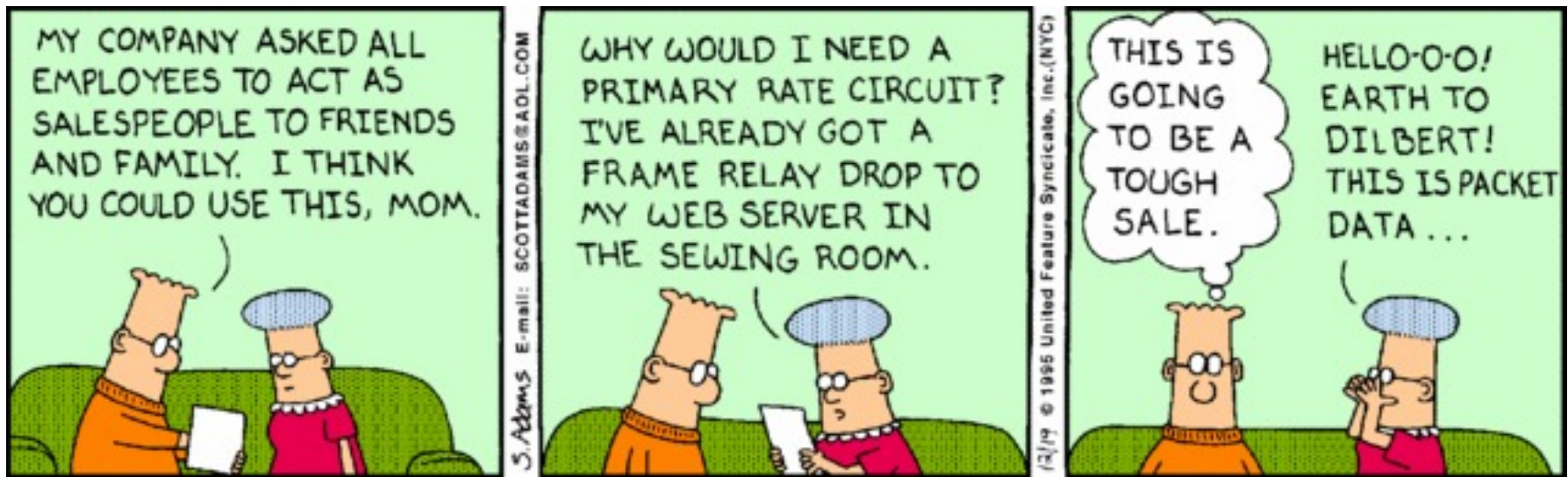


# Sisteme Încorporate

## Cursul 4

## Consumul de energie în Embedded(2)

Facultatea de Automatică și Calculatoare  
Universitatea Politehnica București



<http://dilbert.com/strips/comic/1995-12-19/>

## Chestiuni generale

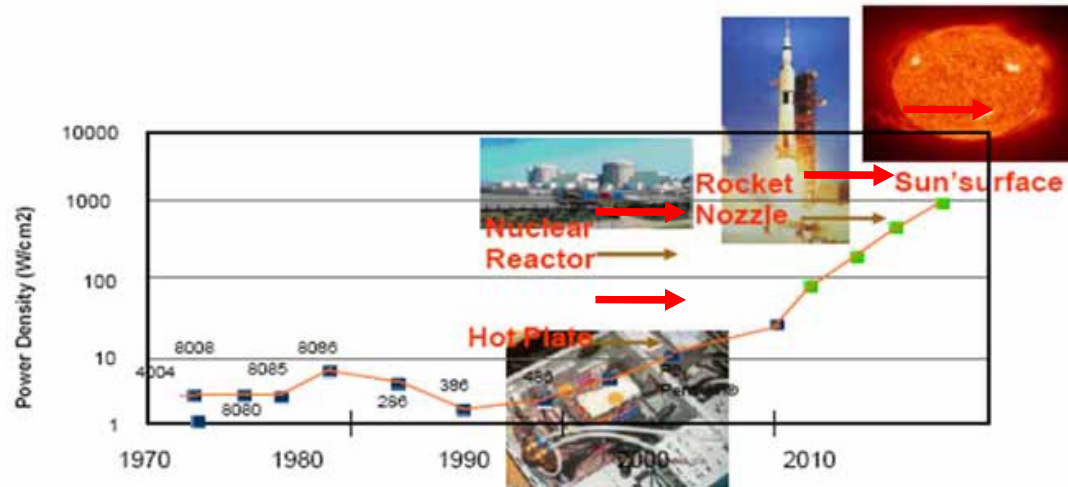
Putere vs. Energie

Tehnici de bază

- Paralelism
- VLIW (paralelism și overhead redus)
- Dynamic Voltage Scaling
- Dynamic Power Management

# Consumul de energie și putere

De ce avem nevoie de eficiență energetică:



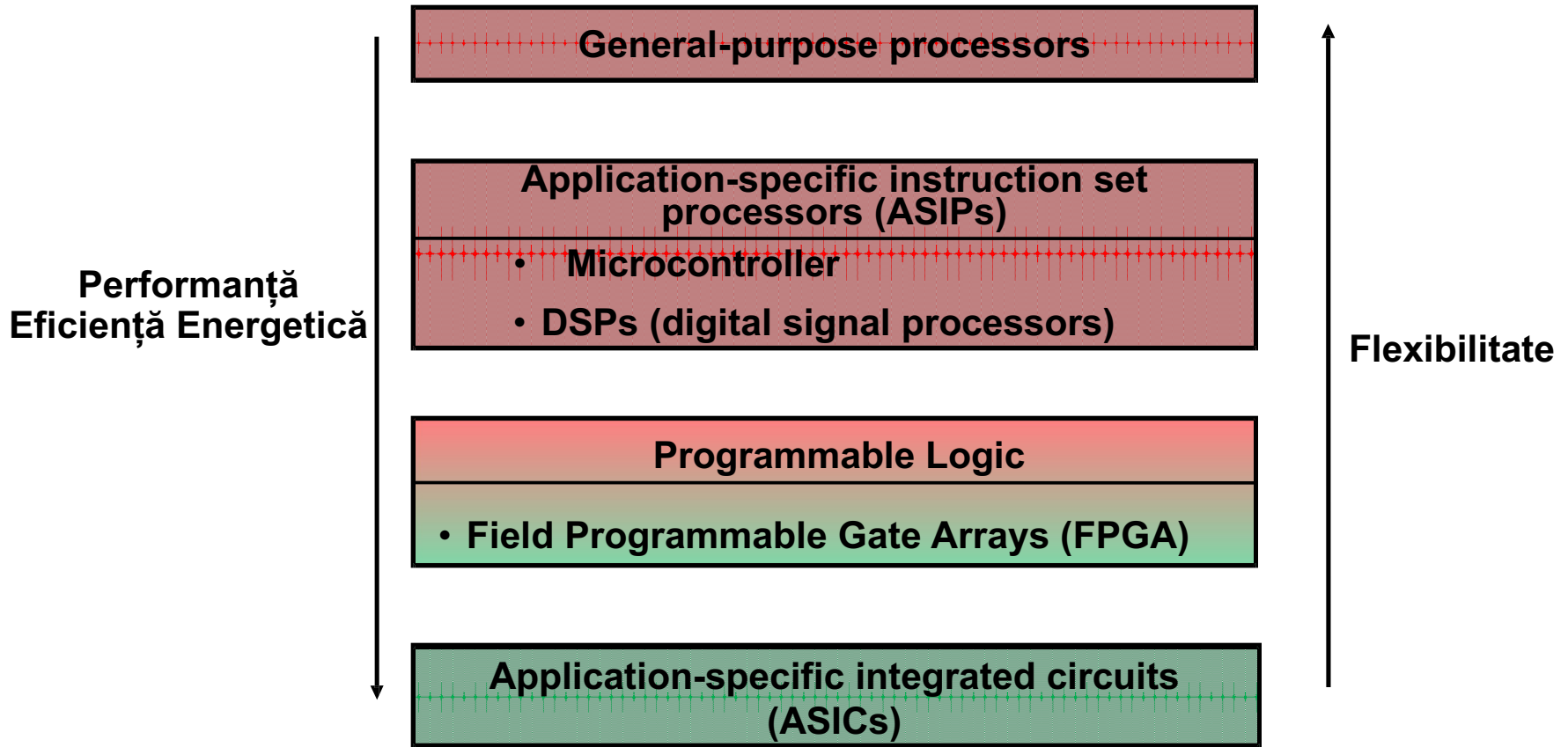
***„Power is considered as the most important constraint in embedded systems”***

[in: L. Eggermont (ed): Embedded Systems Roadmap 2002,  
STW]

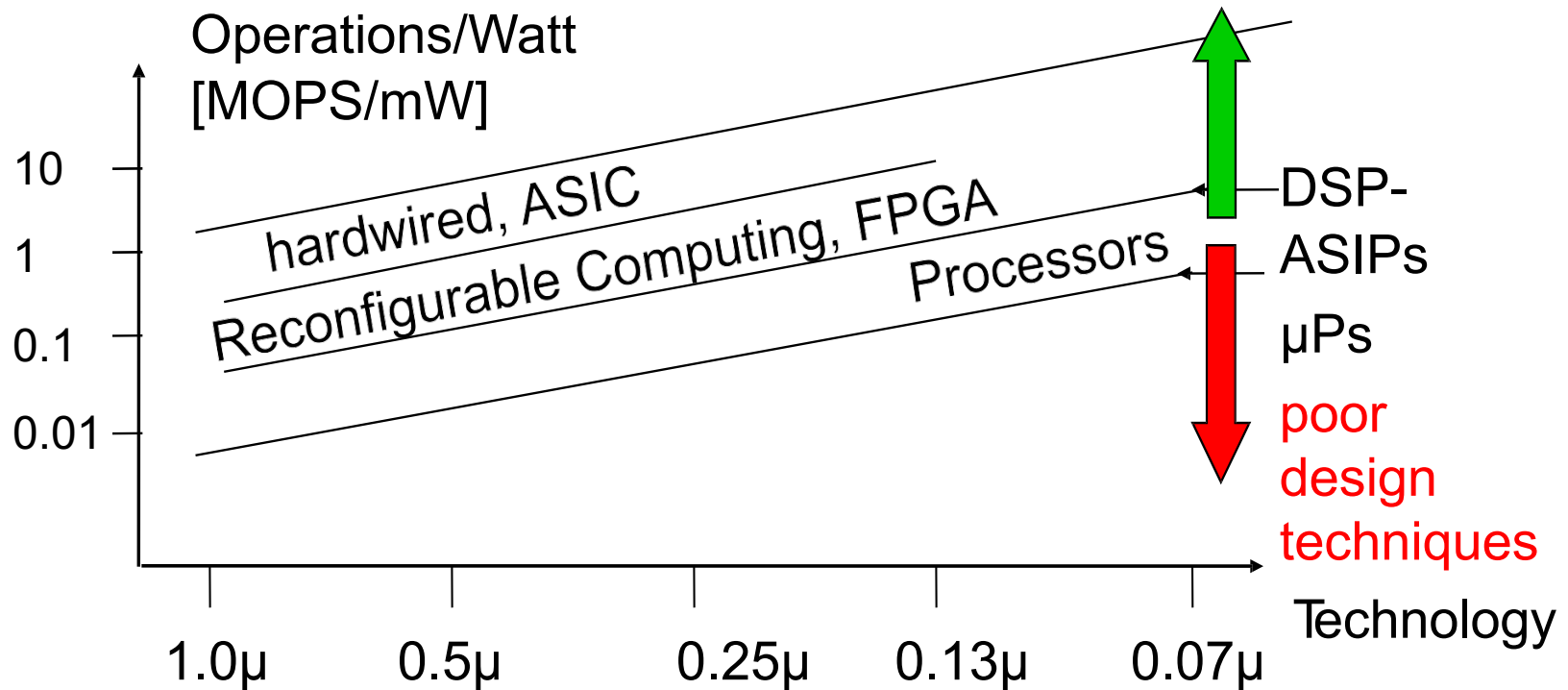
***“Power demands are increasing rapidly, yet battery capacity cannot keep up”.***

[in Ditzel et al.: Power-Aware Architecting for data-dominated applications, 2007,  
Springer]

# Alternative de implementare



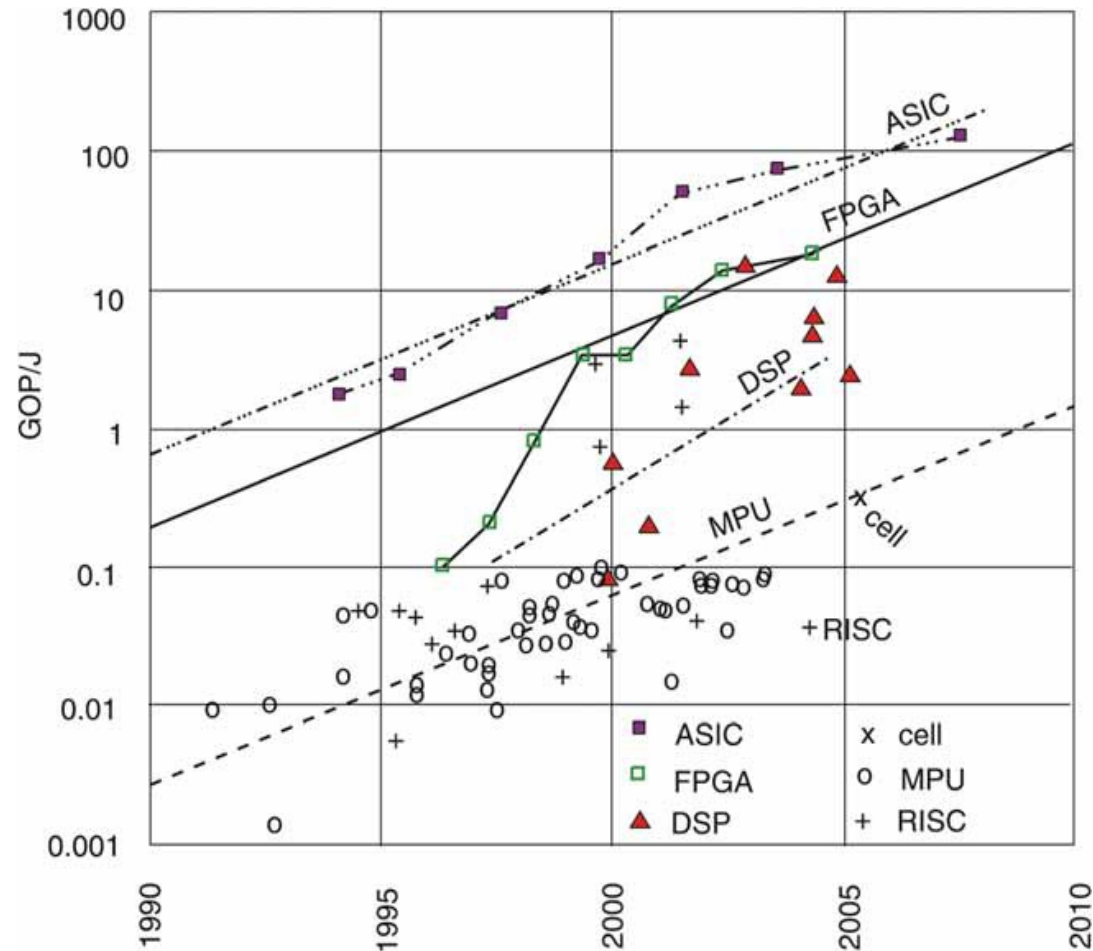
# Conflictul putere/flexibilitate



Este necesară **optimizarea HW și SW**,  
folosirea **arhitecturilor eterogene** și de **tehnici  
specializate de design**.

[H. de Man, Keynote,  
DATE'02;  
T. Claasen, ISSCC99]

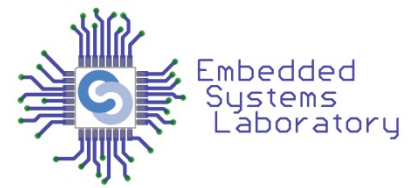
# Eficiența Energetică



© Hugo De Man,  
IMEC, Philips, 2007

# Ce poate să facă un proiectant embedded?

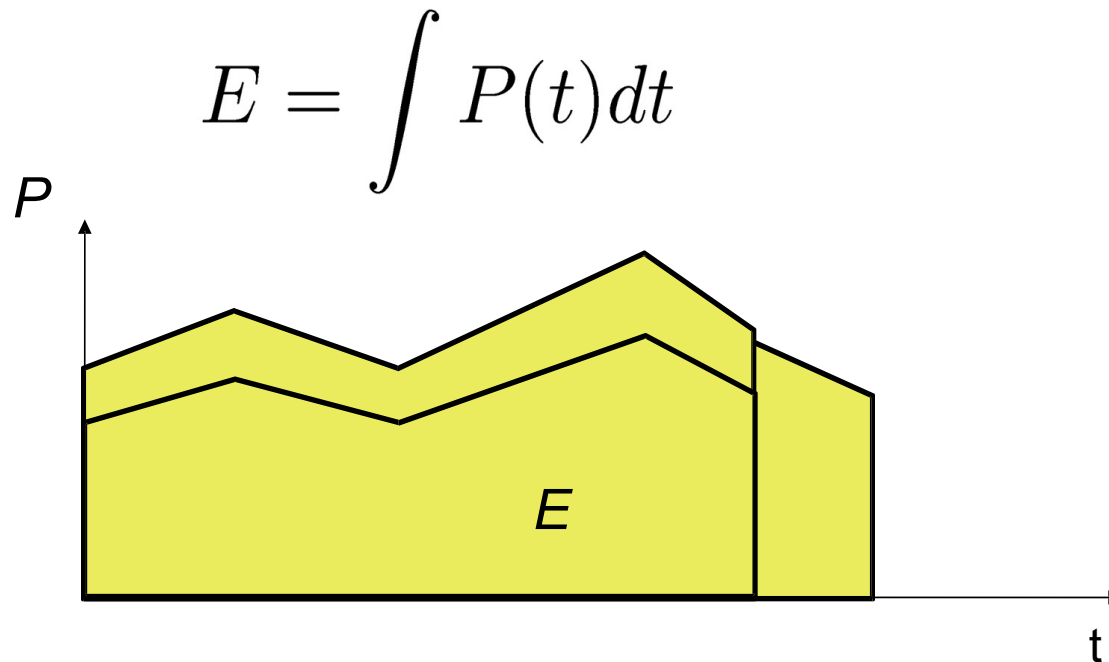
---



1. Să înțeleagă unde și de ce se disipă energia într-un circuit.
2. Să se gândească cum să reducă consumul la toate nivelele din ierarhia de design.
  - În trecut era destul de greu de implementat o astfel de abordare
  - Toate eforturile erau depuse în reducerea consumului la nivel de circuit



- ▶ Chestiuni generale
- ▶ *Putere vs. Energie*
- ▶ Tehnici de bază
  - Paralelism
  - VLIW (paralelism și overhead redus)
  - Dynamic Voltage Scaling
  - Dynamic Power Management

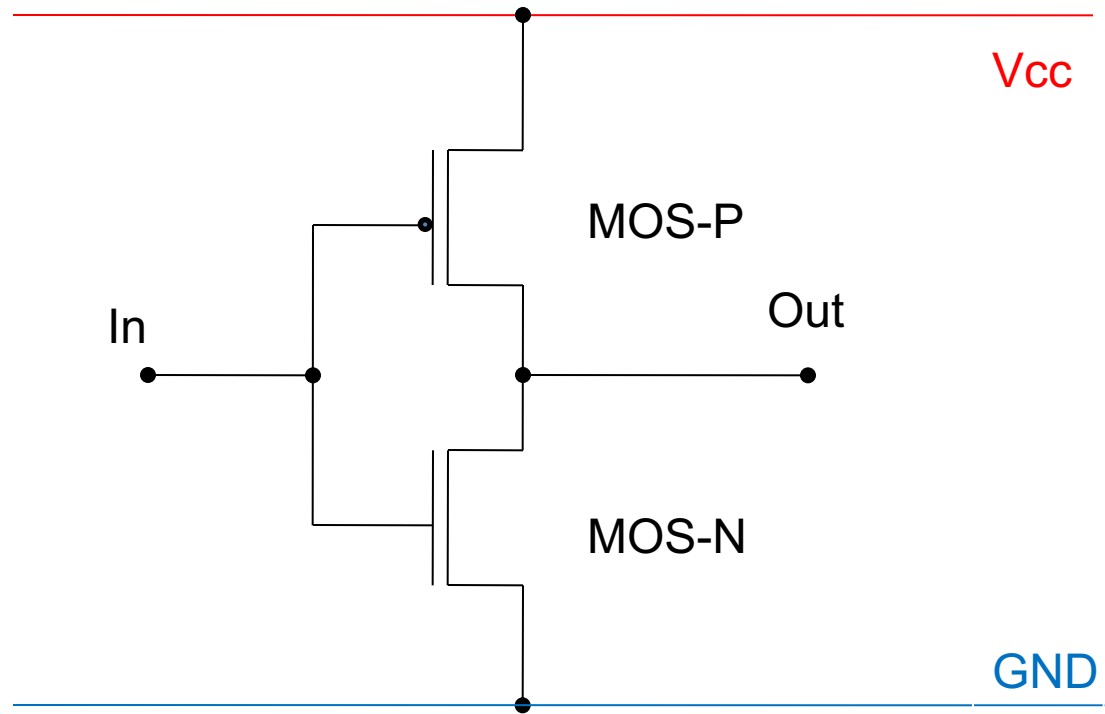


În majoritatea cazurilor, o execuție mai rapidă înseamnă și consum mai mic de energie, dar exact opusul poate să se întâmple în momentul în care puterea trebuie mărită pentru a permite execuția rapidă.

# Low Power vs. Low Energy

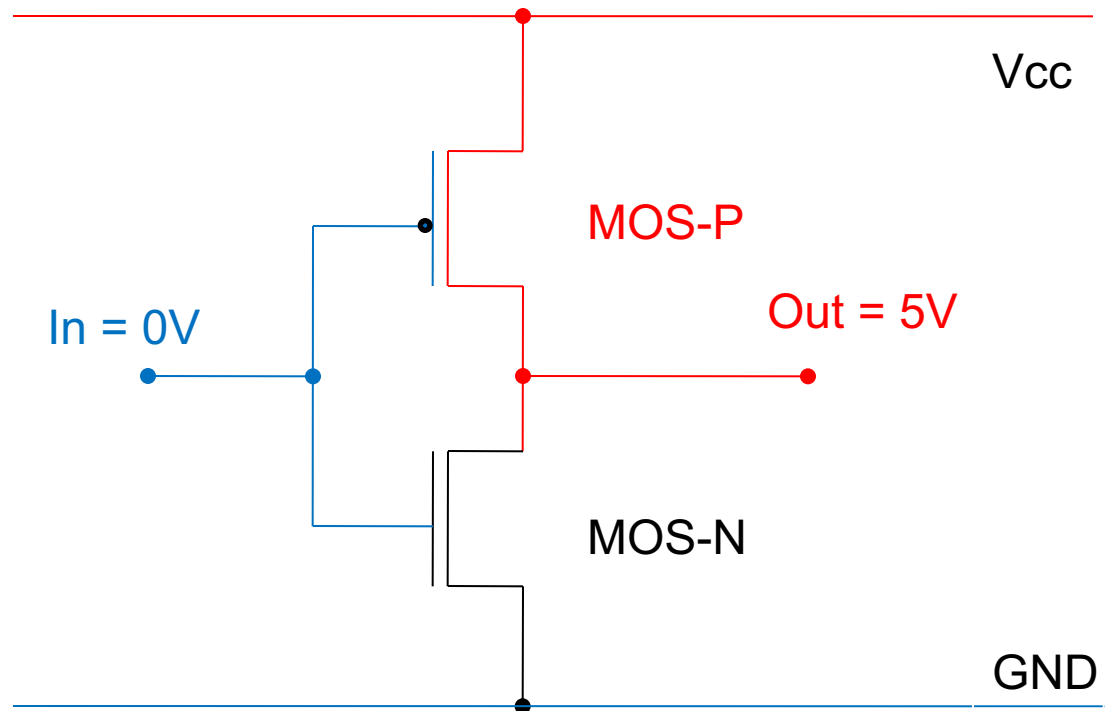
- ▶ Minimizarea **consumului de putere** este importantă pentru:
  - proiectarea sursei de alimentare
  - proiectarea stabilizatoarelor de tensiune
  - dimensiunile conexiunilor și traseelor de alimentare
  - răcire (disipare eficientă a căldurii)
    - cost ridicat (estimat că va crește de la \$1 la \$3 per Watt pentru disiparea căldurii [Skadron et al. ISCA 2003])
    - spațiu limitat
  
- ▶ Minimizarea **consumului de energie** este importantă pentru:
  - disponibilitate redusă a energiei (sisteme mobile)
  - capacitatea limitată a bateriilor (crește prea lent)
  - costurile mari ale energiei (panouri solare, în spațiu)
  - timp de viață mărit, temperatură de operare scăzută

# Exemplu: Inversorul CMOS



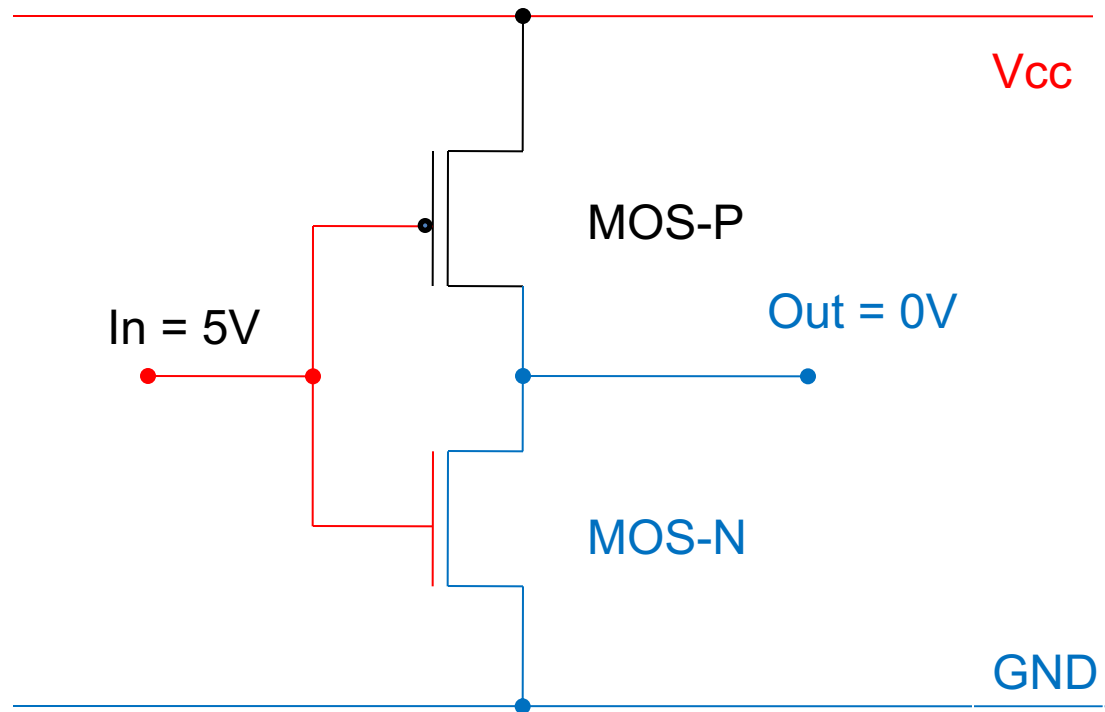
- Cea mai simplă poartă logică – doar două tranzistoare
- Funcționare complementară (MOS-N în conjuncție cu MOS-P)

# Exemplu: Inversorul CMOS



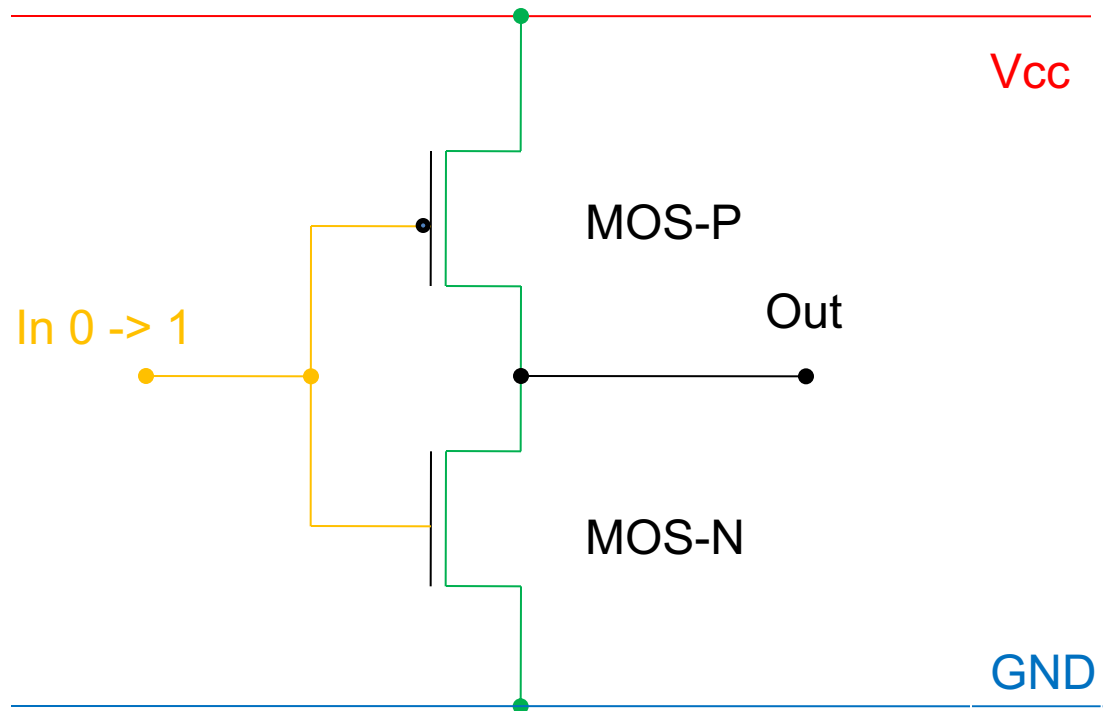
- Input = 0V  $\rightarrow$  MOS-P conduce, MOS-N blocat  $\rightarrow$  Out = 5V

# Exemplu: Inversorul CMOS



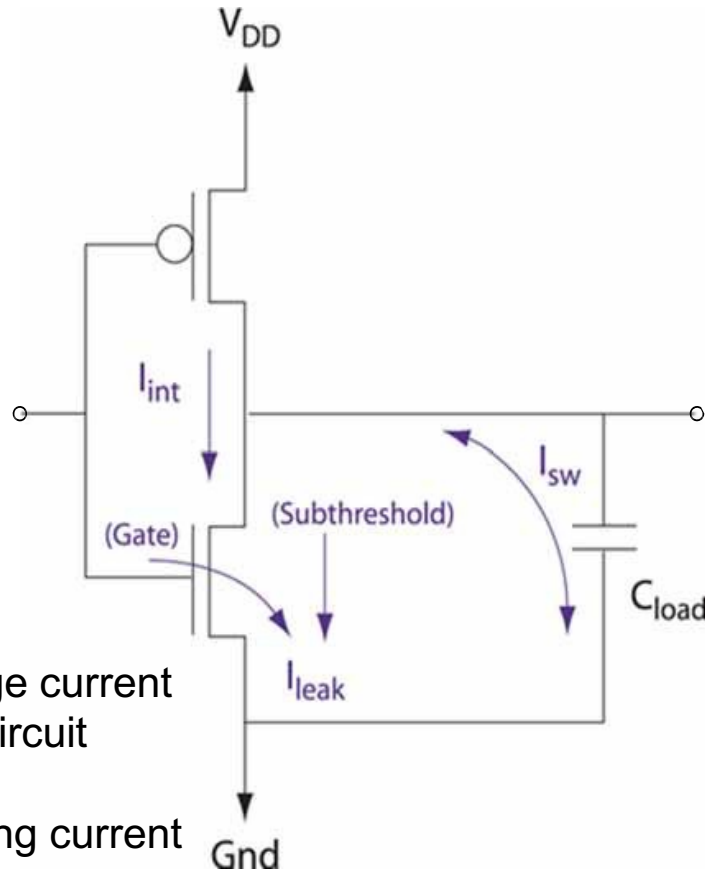
- Input = 5V  $\rightarrow$  MOS-N conduce, MOS-P blocat  $\rightarrow$  Out = 0V

# Exemplu: Inversorul CMOS



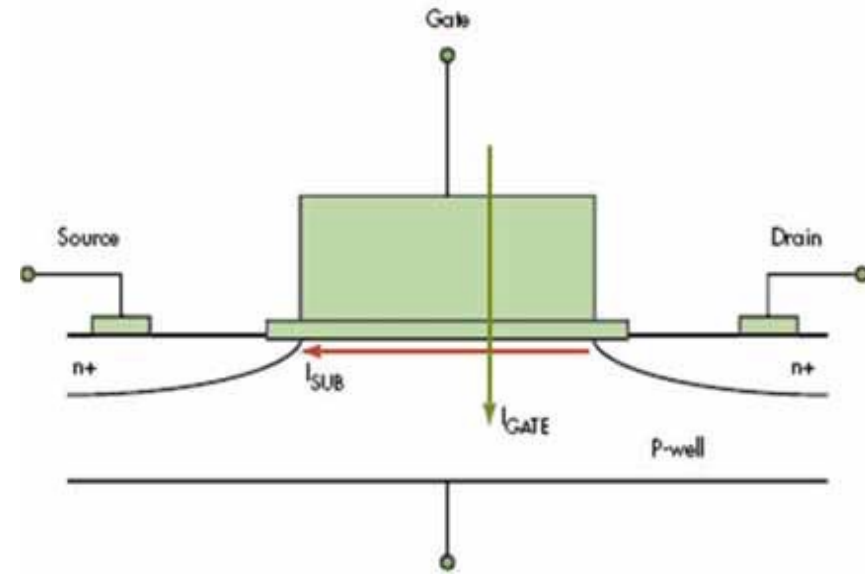
- Ce se întâmplă în regim tranzitoriu? (când intrarea comută)
  - Pentru o scurtă perioadă de timp ( $\sim nS$ ) amândouă tranzistoarele sunt deschise  $\rightarrow$  scurt între  $V_{cc}$  și GND  $\rightarrow$  consum de energie

# Consumul de putere al unei porți CMOS



$I_{leak}$  : leakage current  
 $I_{int}$  : short circuit current  
 $I_{sw}$  : switching current

## Subthreshold and gate-oxide leakage





# Dynamic Voltage Scaling (DVS)

**Consumul de putere CMOS  
(fără leakage):**

$$P \sim \alpha C_L V_{dd}^2 f$$

$V_{dd}$  : supply voltage  
 $\alpha$  : switching activity  
 $C_L$  : load capacity  
 $f$  : clock frequency

**Întârzierea unui circuit  
CMOS**

$$\tau \sim C_L \frac{V_{dd}}{(V_{dd} - V_T)^2}$$

$V_{dd}$  : supply voltage  
 $V_T$  : threshold  
voltage  
 $V_T \ll V_{dd}$

Scăderea  $V_{dd}$  reduce  $P$  quadratic ( $f$  constant).  
Gate delay crește reciproc.  
Frecvența maximă  $f_{\max}$  scade liniar.

# Puterea disipată în CMOS

$$P = \frac{1}{2} ACV^2 f + \tau AVI_{short} f + VI_{leak}$$

$P$  = putere totală

$V$  = tensiunea de alimentare

$f$  = frecvența de ceas

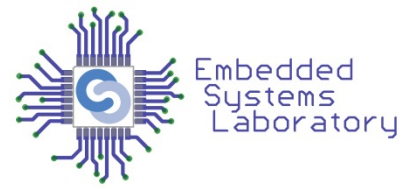
$C$  = capacitatea liniilor de ieșire

$A$  = activitate (tranziții logice pe ciclu de ceas)

$I_{leak}$  = curent de mers în gol       $I_{short}$  = curent de scurt-circuit

$\tau$  = durata curentului de scurt-circuit

# Măsurarea puterii disipate în CMOS



$$P = \underbrace{\frac{1}{2} ACV^2 f}_{\text{Puterea de comutație}} + \underbrace{\tau AVI_{short} f}_{\text{Puterea de scurt-circuit}} + \underbrace{VI_{leak}}_{\text{Puterea de mers în gol}}$$

Puterea de comutație Puterea de scurt-circuit Puterea de mers în gol

Putere dinamică

Putere statică

# Puterea dinamică

---

- Puterea de comutație
  - Puterea dinamică cauzată de încărcarea și descărcarea sarcinilor capacitive de la ieșirea fiecărei porți logice
    - Este “vinovată” pentru majoritatea puterii disipate într-un chip
    - Factori de influență
      - Tensiunea de alimentare (V)
      - Frecvența de ceas (f)
      - Activitatea (A): unele porți logice nu comută la fiecare ciclu
- Puterea de scurt-circuit
  - Putere dinamică datorată existenței curentului de scurt-circuit  $I_{\text{short}}$  care se stabilește un timp  $\tau$  între alimentare și masă în momentul în care intrarea comută dintr-o stare în alta și ambele tranzistoare sunt deschise.

- Leakage current. Este predominant când circuitul este inactiv
  - Există un consum deoarece tranzistoarele nu au rezistența infinită atunci când sunt închise
- Diode leakage (neglijabil)
  - Sursa (și drena) formează o diodă cu substratul
  - În anumite condiții dioda poate fi polarizată, determinând apariția unui curent rezidual.

# Consumul de putere al procesoarelor CMOS

## ► *Principalele surse:*

### ▪ Puterea dinamică

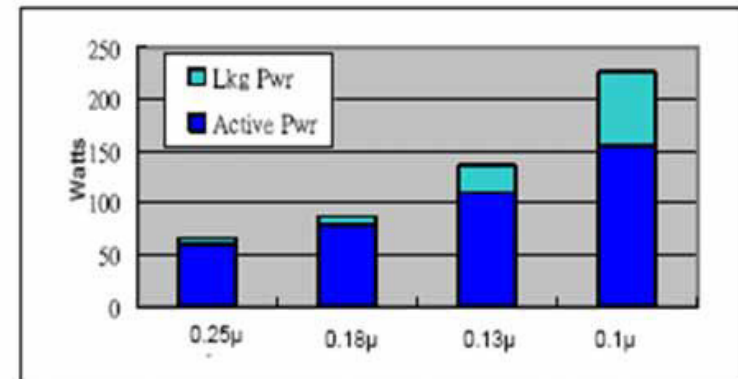
- Încărcarea și descărcarea condensatoarelor parazite

### ▪ Puterea de scurt-circuit

- Scurt-circuit între Vcc și GND la comutare

### ▪ Leakage

- Diode și tranzistoare imperfecte
- Devine unul din factorii majori odată cu micșorarea dimensiunilor



(Micro32 Keynotes by Fred Pollack)

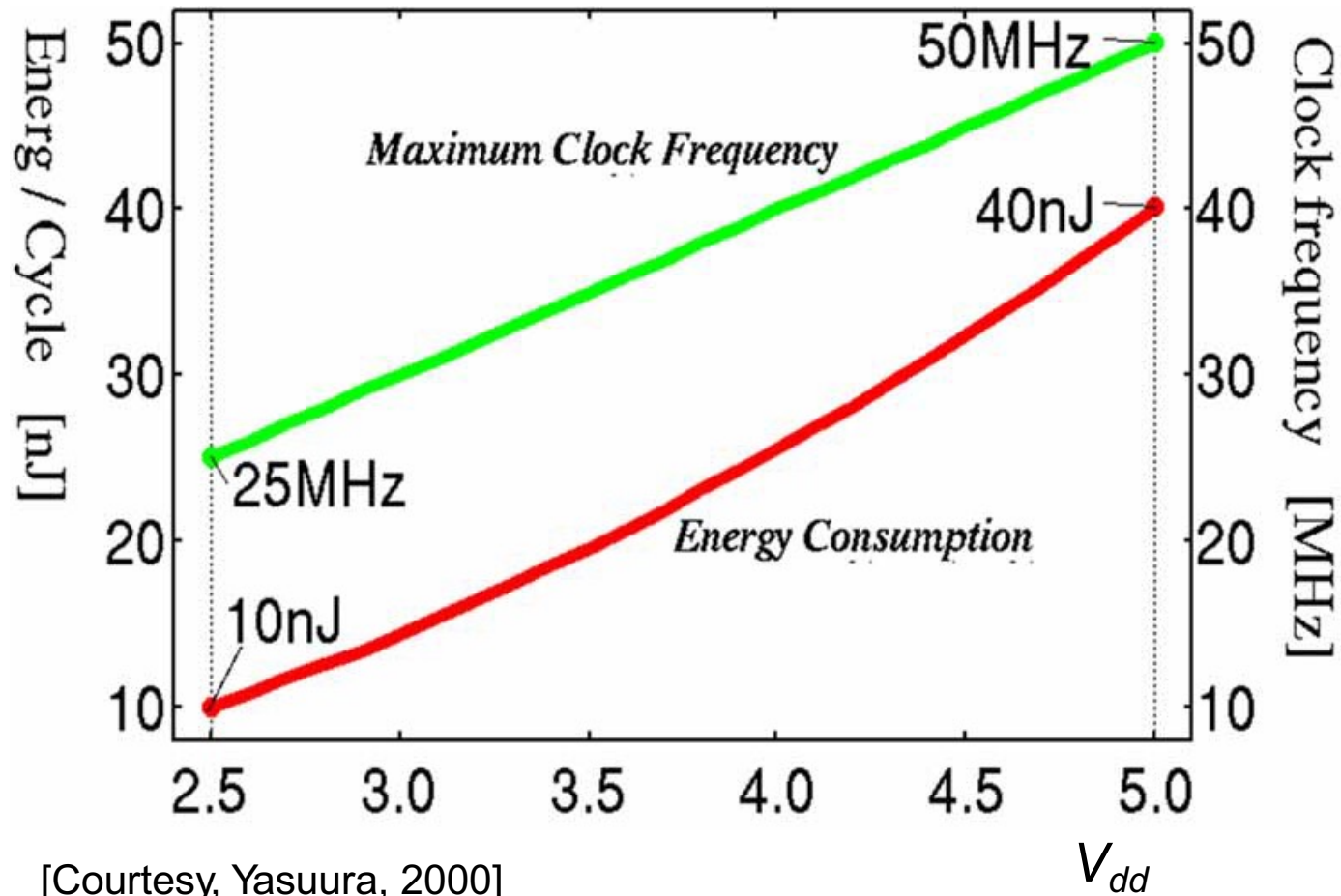
$$P \sim \alpha C_L V_{dd}^2 f$$

$$E \sim \alpha C_L V_{dd}^2 f t = \alpha C_L V_{dd}^2 (\#cycles)$$

Reducerea consumului de energie pentru un task dat:

- Reducerea tensiunii de alimentare  $V_{dd}$
- Reducerea activității  $\alpha$
- Reducerea capacității din sarcină  $C_L$
- Reducerea numărului de cicli  $\#cycles$

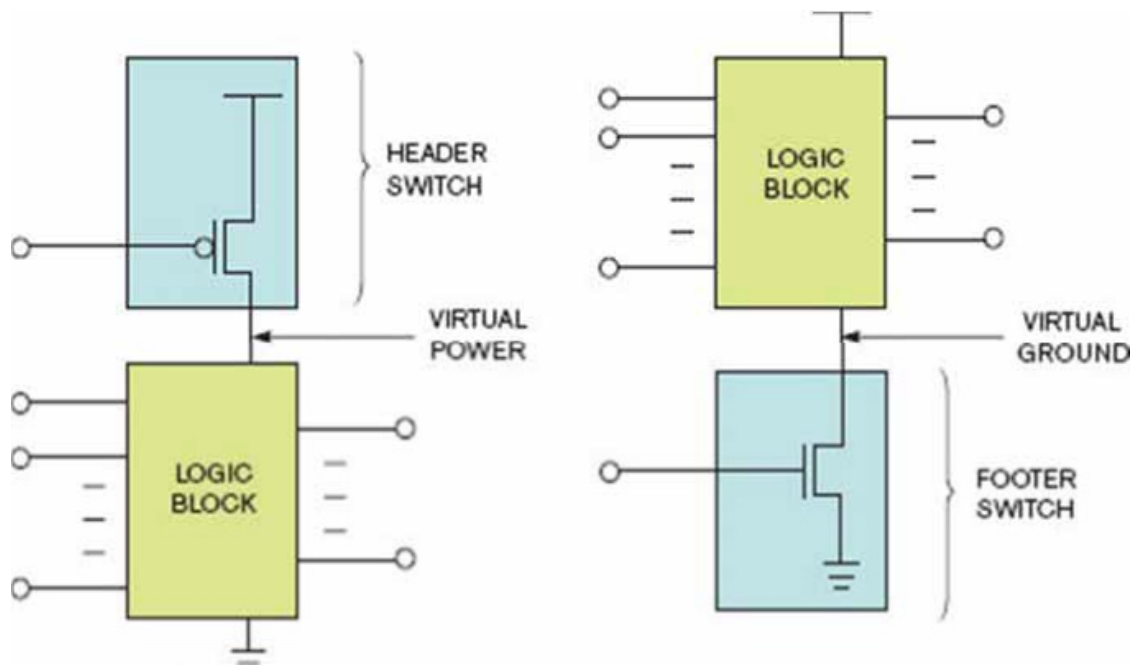
# Exemplu: Voltage Scaling





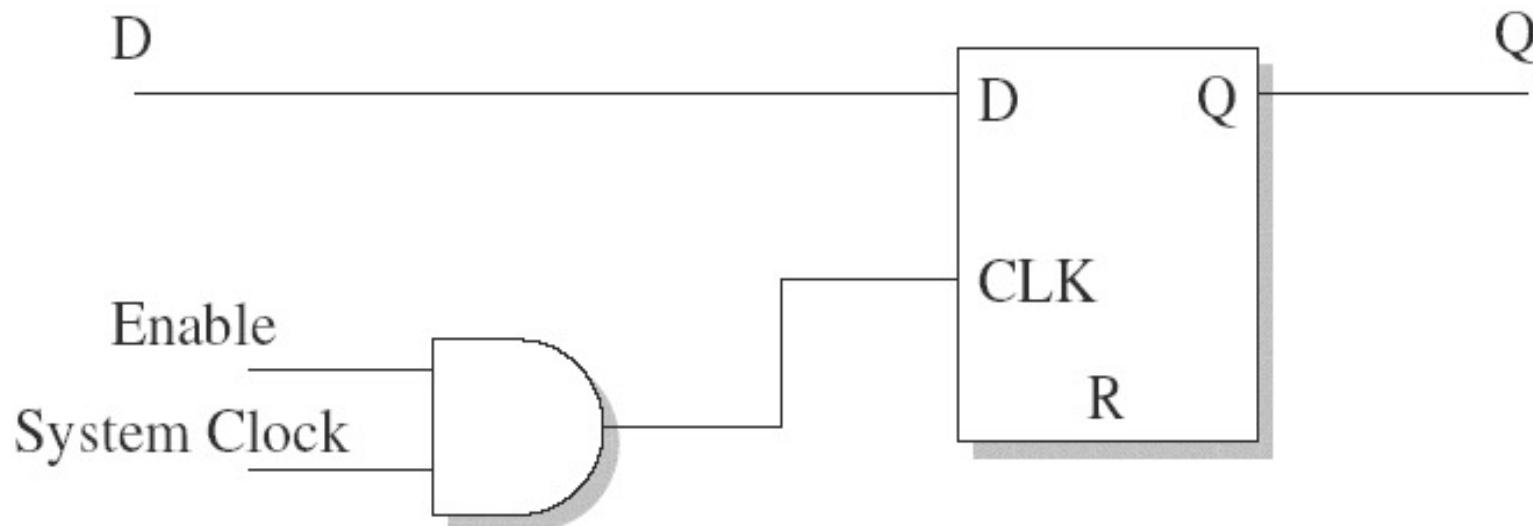
# Power Supply Gating

- ▶ Power gating este una dintre cele mai eficiente metode de a minimiza consumul static de putere (leakage)
  - Taie alimentarea unităților/componentelor inactive
  - Reduce leakage-ul



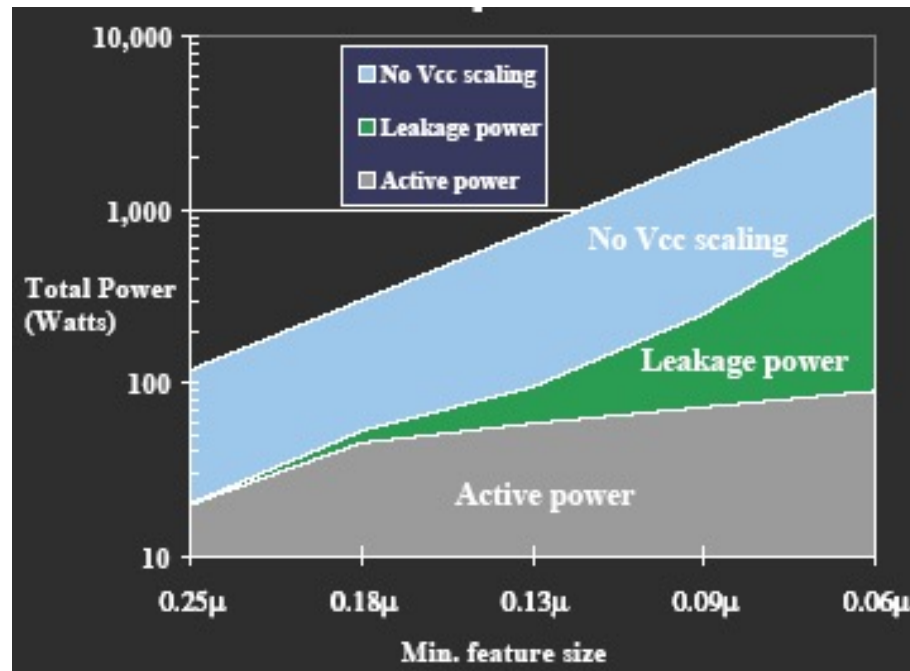
# Clock gating

Deselectează unitațile logice atunci când nu sunt folosite de sistem



Care factor din formula puterii este minimizat?

# Disiparea Puterii



- Puterea activă este factorul cel mai important în ziua de azi
- Puterea de mers în gol va deveni importantă odată cu mărirea integrării
- Puterea de mers în gol crește cu temperatura

# Strategii de reducere a consumului

- Reducerea frecvenței de ceas ( $f$ )
  - Scade puterea medie dar mărește timpul de execuție -> consumul de energie rămâne constant
  - Metrică mai bună pentru un procesor low-power: MIPS/W  $\equiv$  million instructions per sec per watt
- Reducerea tensiunii de alimentare ( $V$ )
  - Puterea de comutație este proporțională cu  $V^2$
  - O scădere a lui  $V$  cu  $\frac{1}{2}$   $\rightarrow$  puterea scade cu  $\frac{1}{4}$
  - De ce nu merge la nesfârșit așa?
- Reducerea activității ( $A$ )
  - Dezactivarea unor blocuri funcționale atunci când nu sunt active
- Reducerea curenților reziduali
  - Dezactivarea alimentării anumitor zone de circuit

# Reducerea lui $V \rightarrow$ Reducerea lui $f$

$$f_{\max} \sim \frac{(V - V_{\text{threshold}})^2}{V}$$

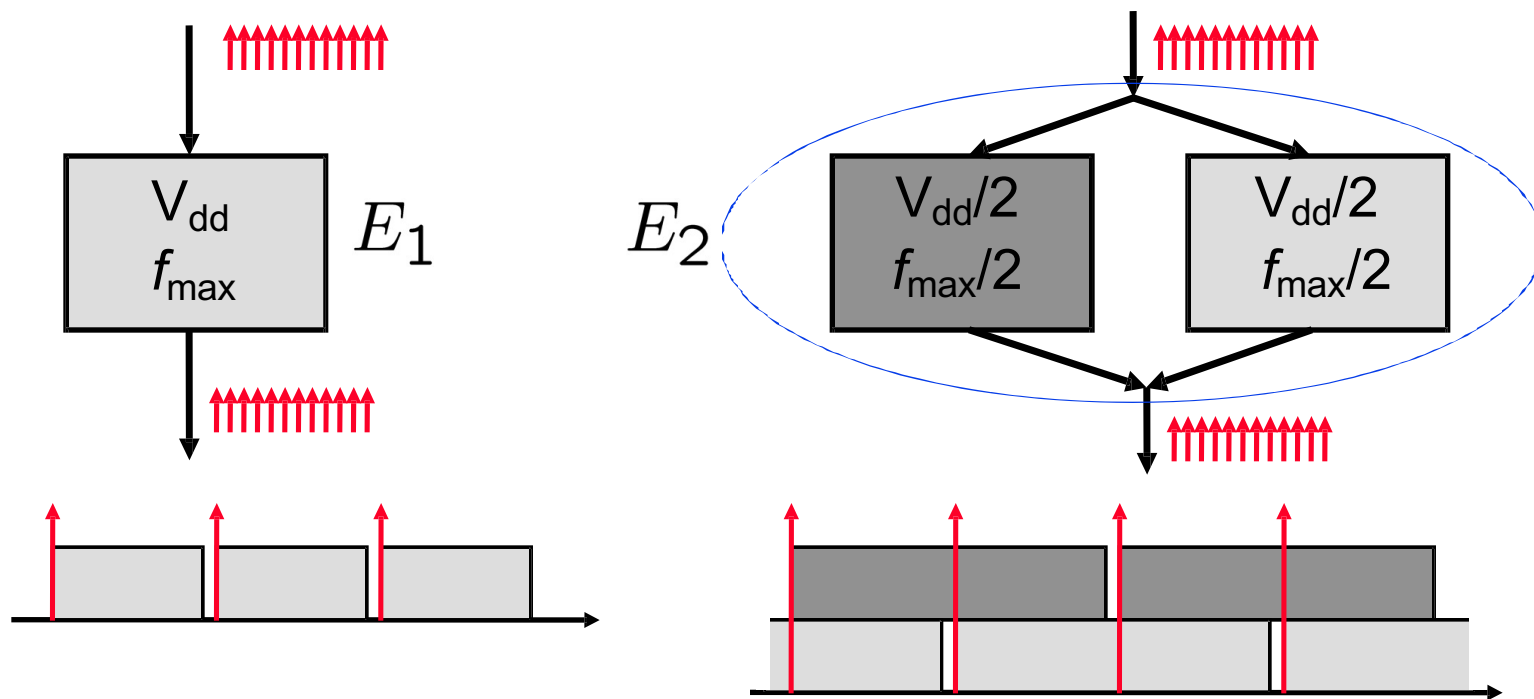
- Frecvența maximă de ceas este direct proporțională cu  $V$
- Reducerea puterii cu  $\frac{1}{4}$  reduce frecvența cu  $\frac{1}{2}$  -> pierderi de performanță -> consum redus
- Implicație indirectă: Paralelismul în procesare este o strategie eficientă de a reduce puterea fără pierderi de performanță

- Altfel, circuitul nu poate funcționa corect
- Reducerea  $V_{threshold}$  mărește exponențial curentul rezidual
- Puterea de mers în gol începe să fie o problemă

$$I_{leak} \sim \frac{1}{e^{\left(\frac{qV_{threshold}}{kT}\right)}}$$

- ▶ Chestiuni generale
- ▶ Putere vs. Energie
- ▶ *Tehnici de bază*
  - **Paralelism**
  - VLIW (paralelism și overhead redus)
  - Dynamic Voltage Scaling
  - Dynamic Power Management

# Folosirea paralelismului



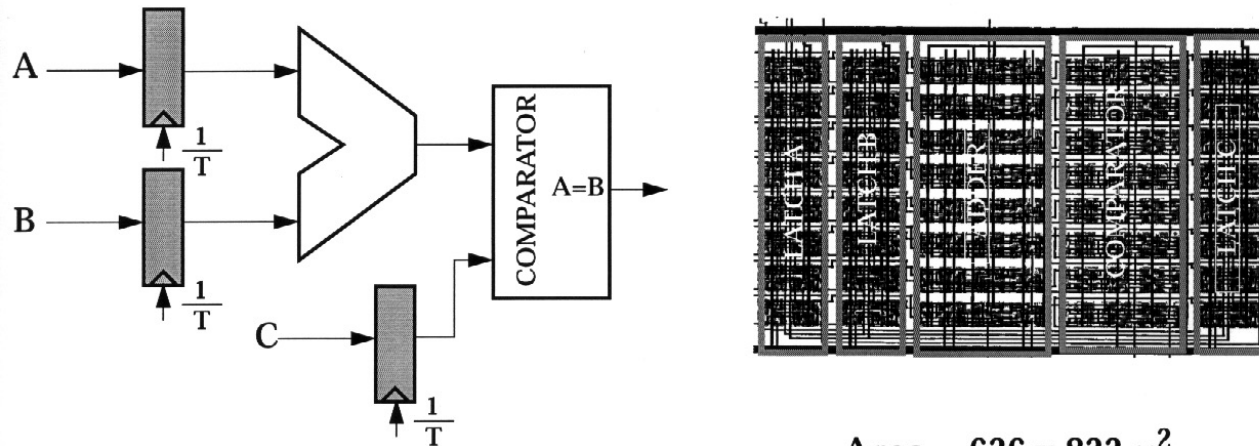
$$E \sim V_{dd}^2 (\#cycles)$$

$$E_2 = \frac{1}{4} E_1$$



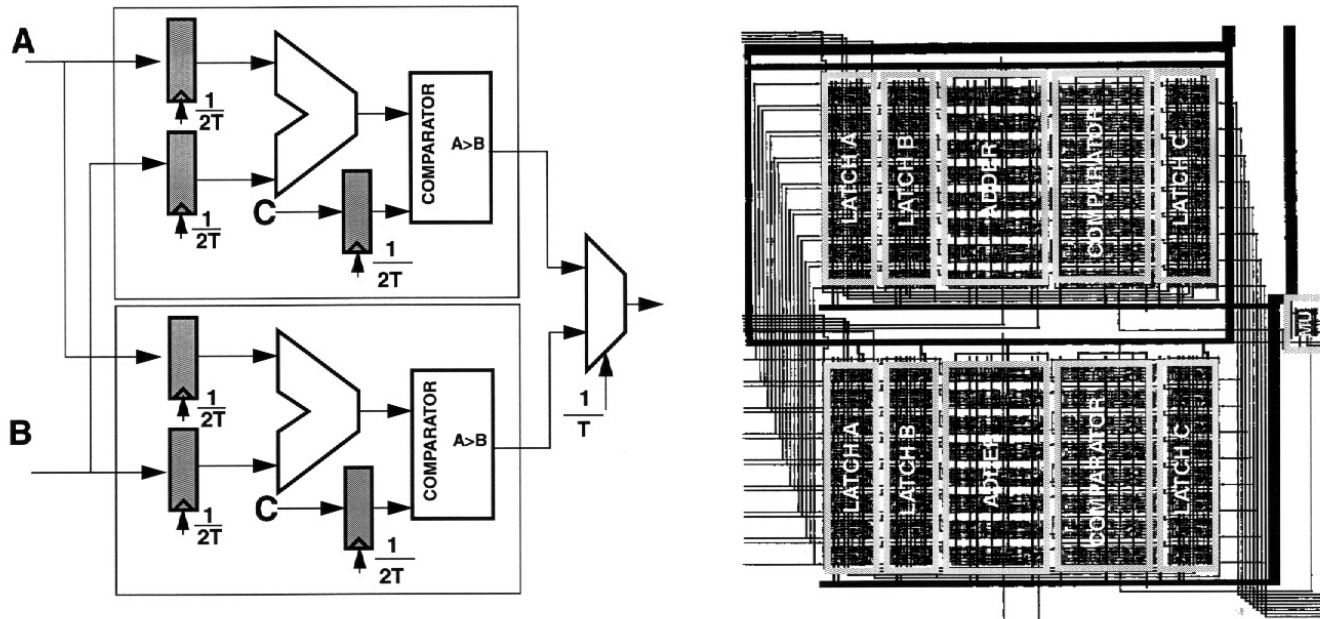


# Exemplu de calcul al puterii



- Întârziere pe calea critică:  $T_{\text{adder}} + T_{\text{comparator}} = 25 \text{ ns}$
- Frecvența:  $f_{\text{ref}} = 40 \text{ MHz}$
- Capacitatea totală de comutație =  $C_{\text{ref}}$
- $V_{\text{dd}} = V_{\text{ref}} = 5\text{V}$
- Puterea de comutație =  $P_{\text{ref}} = C_{\text{ref}} V_{\text{ref}}^2 f_{\text{ref}}$

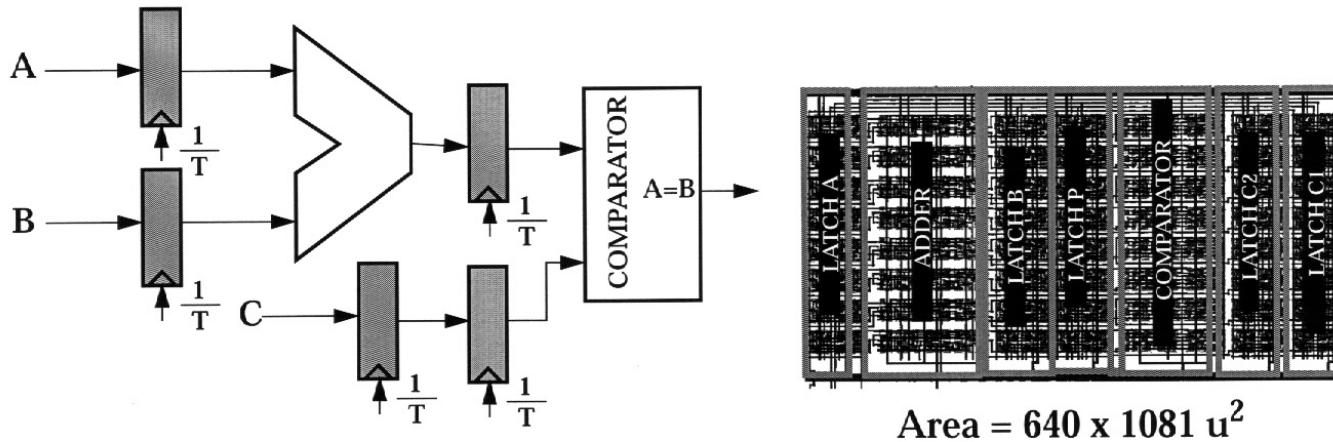
# Paralelizarea căii de date



$$\text{Area} = 1476 \times 1219 \mu^2$$

- Frecvența de ceas poate fi înjumătățită fără a pierde din productivitate:  $f_{\text{par}} = f_{\text{ref}}/2 = 20 \text{ MHz}$
- Capacitate totală de comutație =  $C_{\text{par}} = 2.15C_{\text{ref}}$
- $V_{\text{par}} = V_{\text{ref}}/1.7 (=3V)$
- $P_{\text{par}} = (2.15C_{\text{ref}})(V_{\text{ref}}/1.7)^2(f_{\text{ref}}/2) = 0.37P_{\text{ref}}$

# Pipeline



- $f_{\text{pipe}} = f_{\text{ref}}$   
 $C_{\text{pipe}} = 1.1C_{\text{ref}}$   
 $V_{\text{pipe}} = V_{\text{ref}}/1.7$
- Tensiunea poate fi scăzută păstrându-se aceleași performanțe
- $\text{Pipe} = C_{\text{pipe}} V_{\text{pipe}}^2 f_{\text{pipe}} = (1.1C_{\text{ref}})(V_{\text{ref}}/1.7)^2 f_{\text{ref}} = 0.38P_{\text{ref}}$

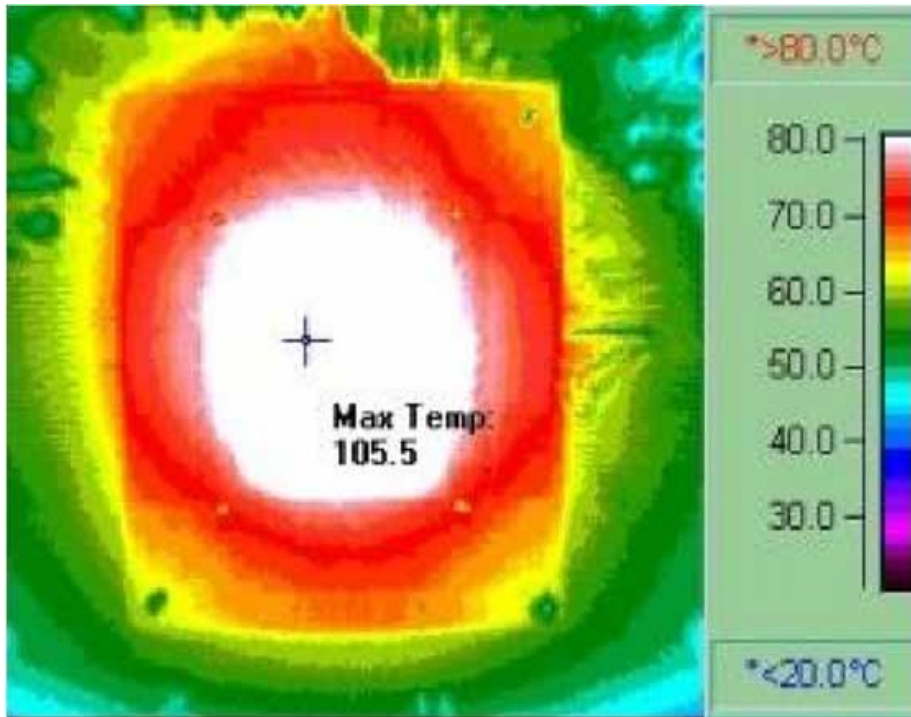
Arhitectura	Tensiune	Arie	Putere
Originala	5V	1	1
Pipeline	3V	1.3	0.38
Paralel	3V	3.4	0.37
Pipeline-Paralel	2.0V	3.7	0.18

- + Prin paralelizare s-a redus puterea dinamică disipată de 5.5 ori.
- Suprafața circuitului a crescut de 4 ori.
- Mărirea numărului de unități logice duce la creșterea puterii disipate în regimul de mers în gol și scurt-circuit.

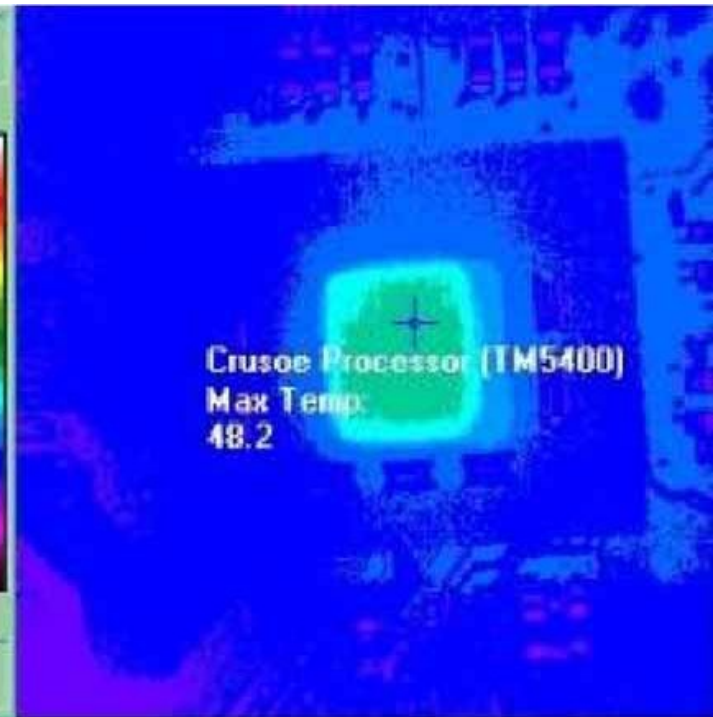
- ▶ Chestiuni generale
- ▶ Putere vs. Energie
- ▶ *Tehnici de bază*
  - Paralelism
  - **VLIW (paralelism și overhead redus)**
  - Dynamic Voltage Scaling
  - Dynamic Power Management

# Ajută foarte mult idelie noi...

Pentium



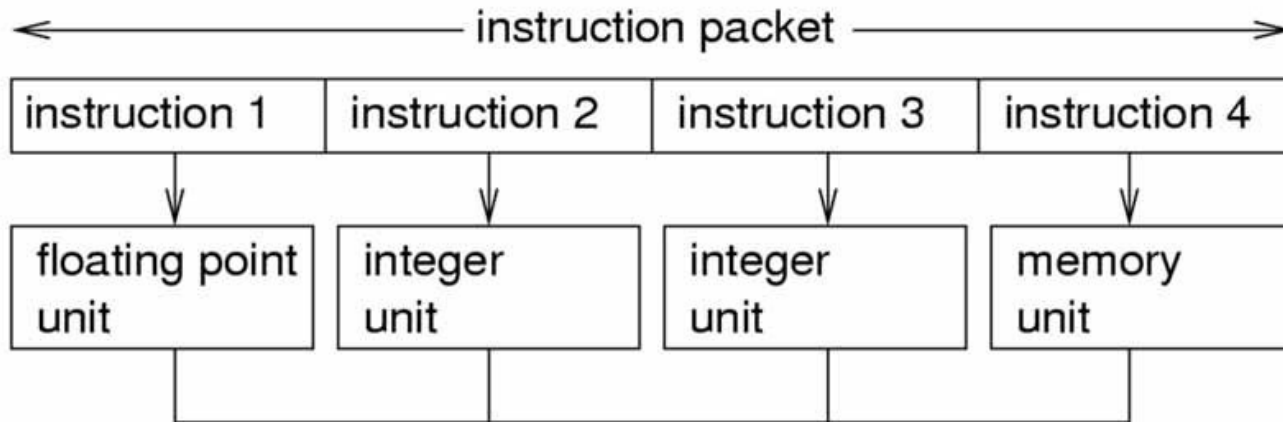
Crusoe



Rulează aceeași aplicație multimedia.

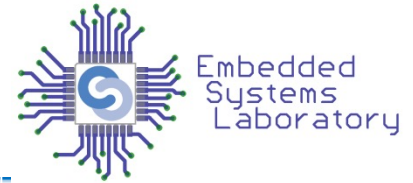
Publicat de Transmeta [\[www.transmeta.com\]](http://www.transmeta.com)

- ▶ ***Un grad foarte mare de paralelism***
  - Multe unități computaționale, (deeply) pipelined
- ▶ ***Arhitectură hardware simplă***
  - Paralelism explicit (parallel instruction set)
  - Paralelizarea este făcută offline (compilator)

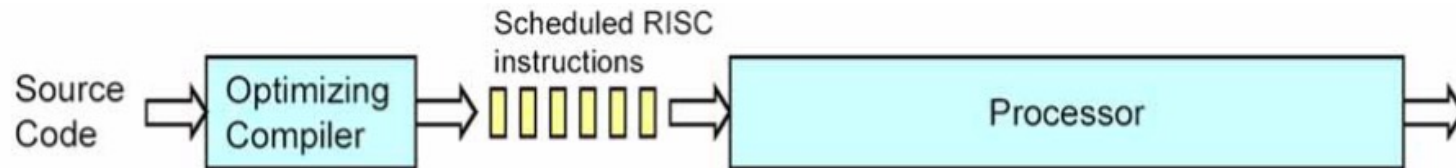




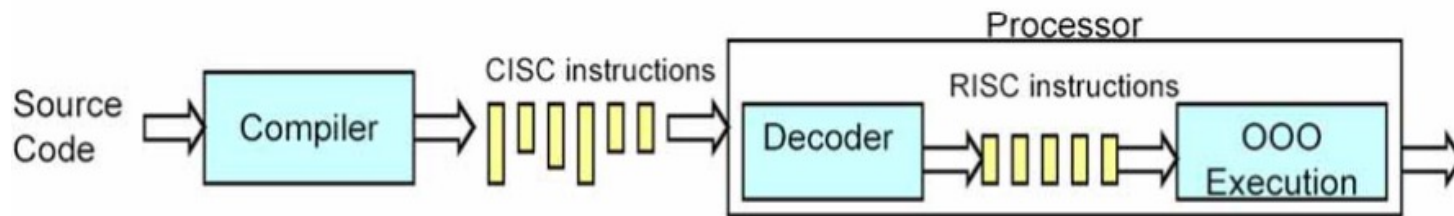
# Transmeta este o arhitectură VLIW tipică



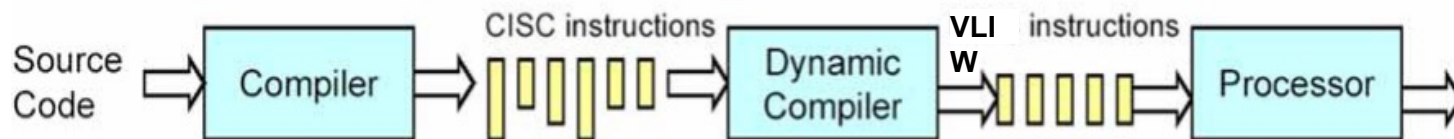
- 128-bit instructions (bundles):
  - 4 operations per instruction
  - 2 combinations of instructions allowed
- Register files
  - 64 integer, 32 floating point
- Some interesting features
  - 6 stage pipeline (2x fetch, decode, register read, execute, write)
  - x86 ISA execution using software techniques
    - Skip the binary compatibility problem!!
    - Interpretation and just-in-time binary translation
  - Speculation support



e.g. Sun, MIPS, ARM



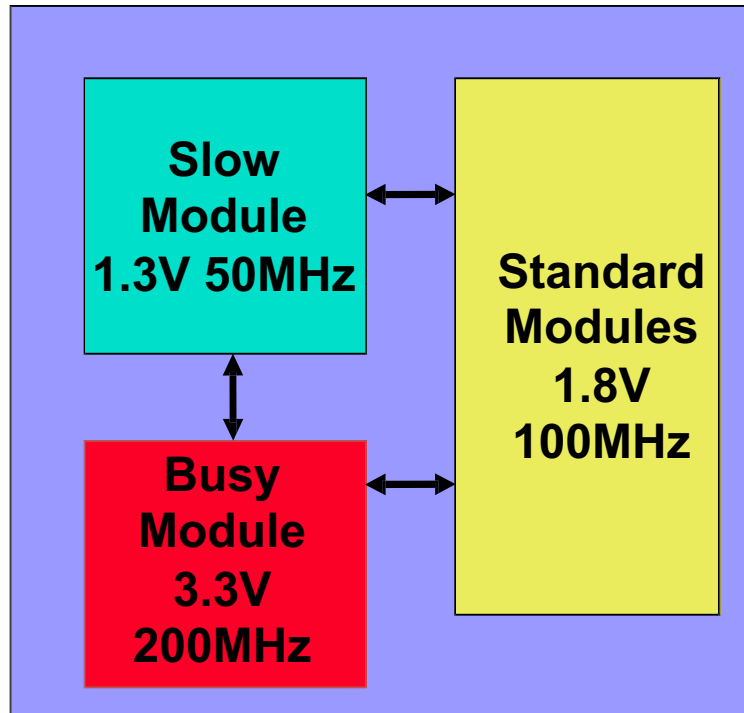
e.g. Intel, AMD



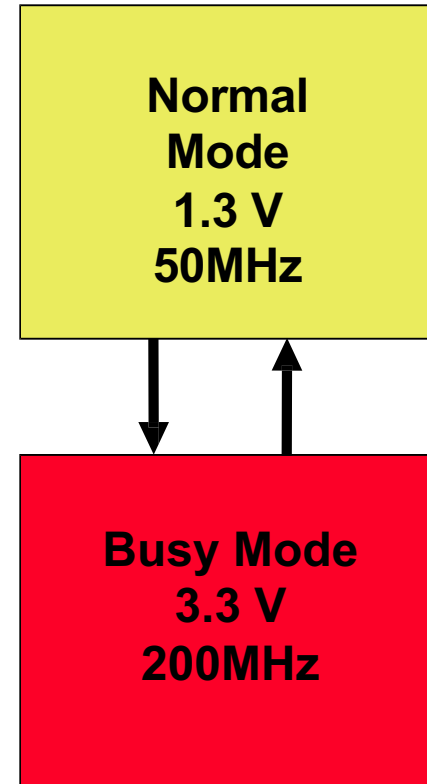
e.g. Transmeta  
(VLIW)

- ▶ Chestiuni generale
- ▶ Putere vs. Energie
- ▶ *Tehnici de bază*
  - Paralelism
  - VLIW (paralelism și overhead redus)
  - **Dynamic Voltage Scaling**
  - Dynamic Power Management

# Management spațial vs. dinamic



Nu toate componentele necesită aceeași performanță.



Performanța cerută poate să varieze în timp

$$P \sim \alpha C_L V_{dd}^2 f$$

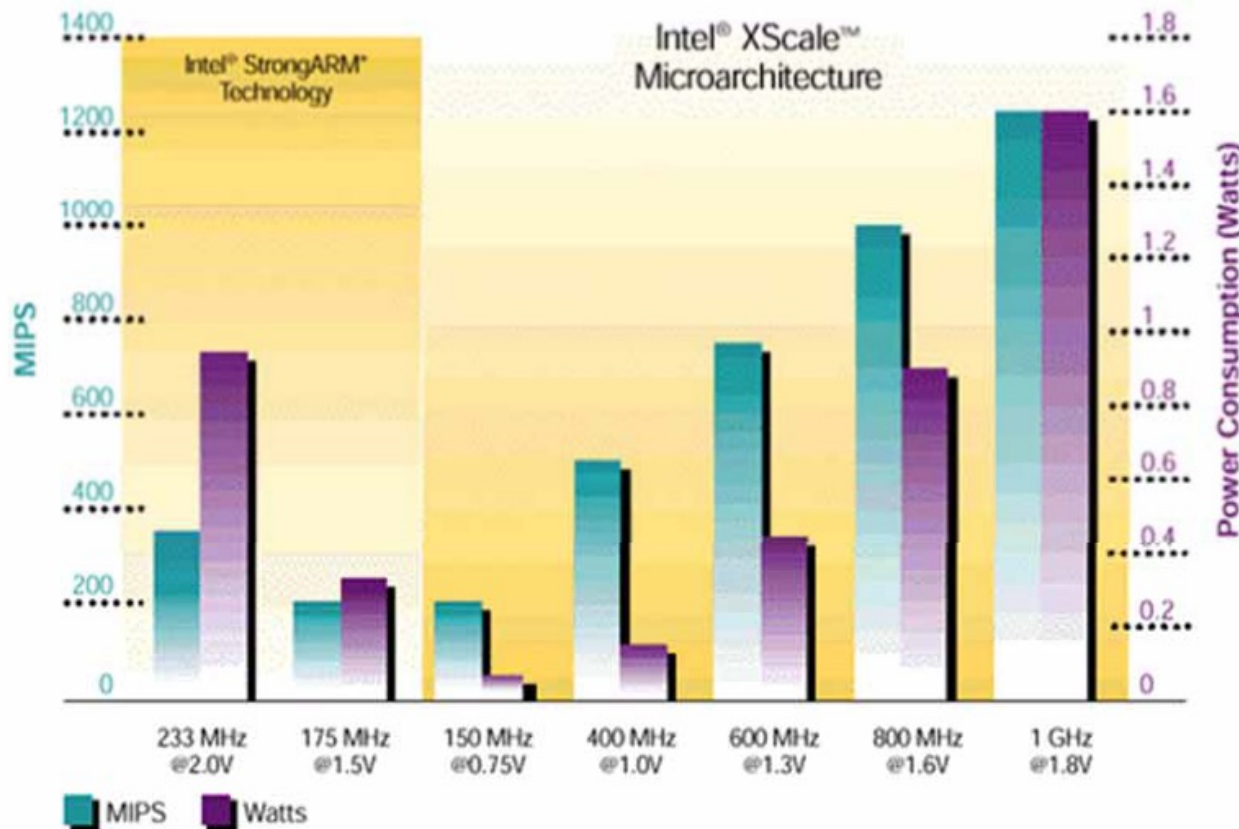
$$E \sim \alpha C_L V_{dd}^2 f t = \alpha C_L V_{dd}^2 (\#cycles)$$

Reducerea consumului de energie pentru un task dat:

- Reducerea tensiunii de alimentare  $V_{dd}$
- Reducerea activității  $\alpha$
- Reducerea capacității din sarcină  $C_L$
- Reducerea numărului de cicli  $\#cycles$

# Exemplu: INTEL Xscale

## POWER-PERFORMANCE COMPARISON

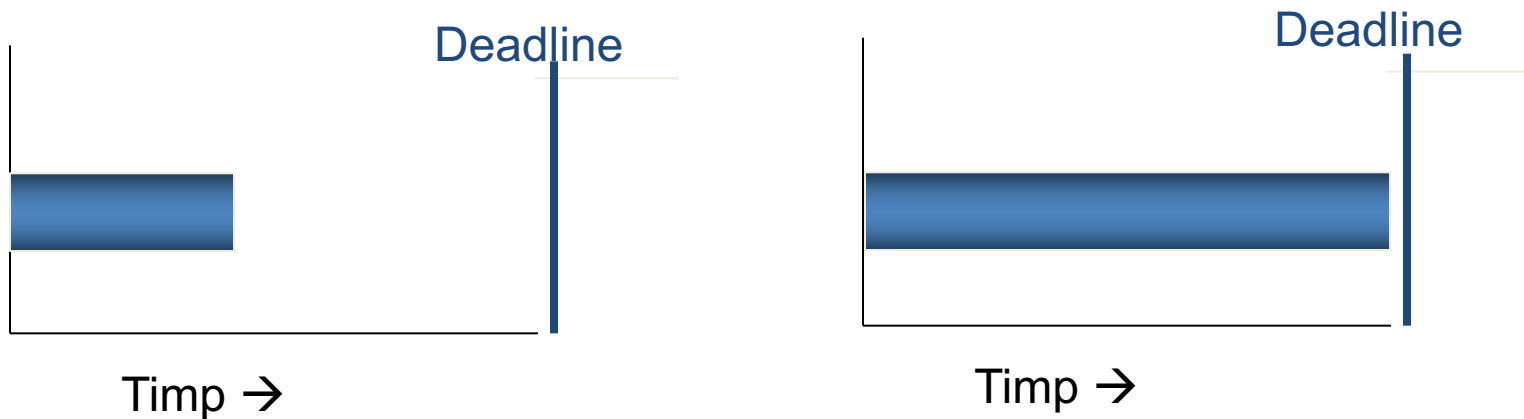


Sistemul de operare trebuie să facă managementul consumului în funcție de necesități și de bugetul energetic dat.

From Intel's Web Site

# Just-in-time scheduling

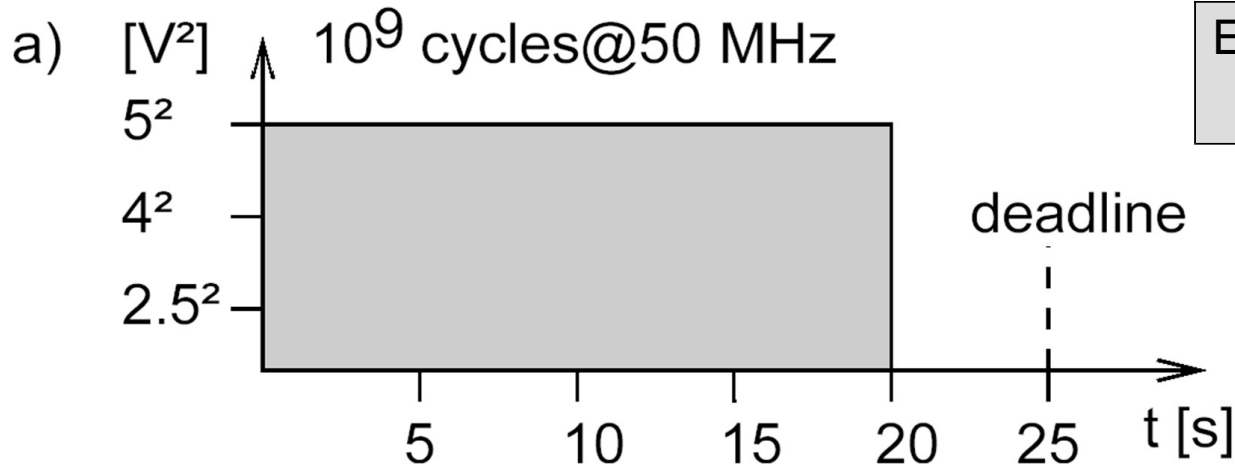
- Exploatează la maxim timpul oferit pentru procesarea unui volum de date.
- Procesorul rulează la o tensiune sau o frecvență redusă pentru a îndeplini sarcina exact în timpul alocat ei.



# Exemplu DVS: a) Complete task ASAP

$V_{dd}$ [V]	5.0	4.0	2.5
Energy per cycle [nJ]	40	25	10
$f_{max}$ [MHz]	50	40	25
cycle time [ns]	20	25	40

Task-ul trebuie să execute  $10^9$  cicli în 25 secunde.



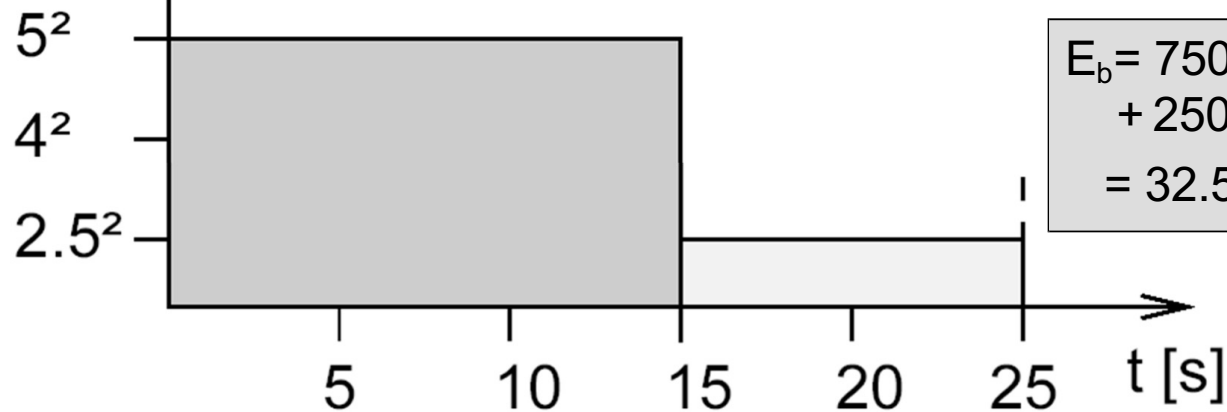
$$E_a = 10^9 \times 40 \times 10^{-9} = 40 \text{ [J]}$$



# Exemplu DVS: b) Două tensiuni

$V_{dd}$ [V]	5.0	4.0	2.5
Energy per cycle [nJ]	40	25	10
$f_{max}$ [MHz]	50	40	25
cycle time [ns]	20	25	40

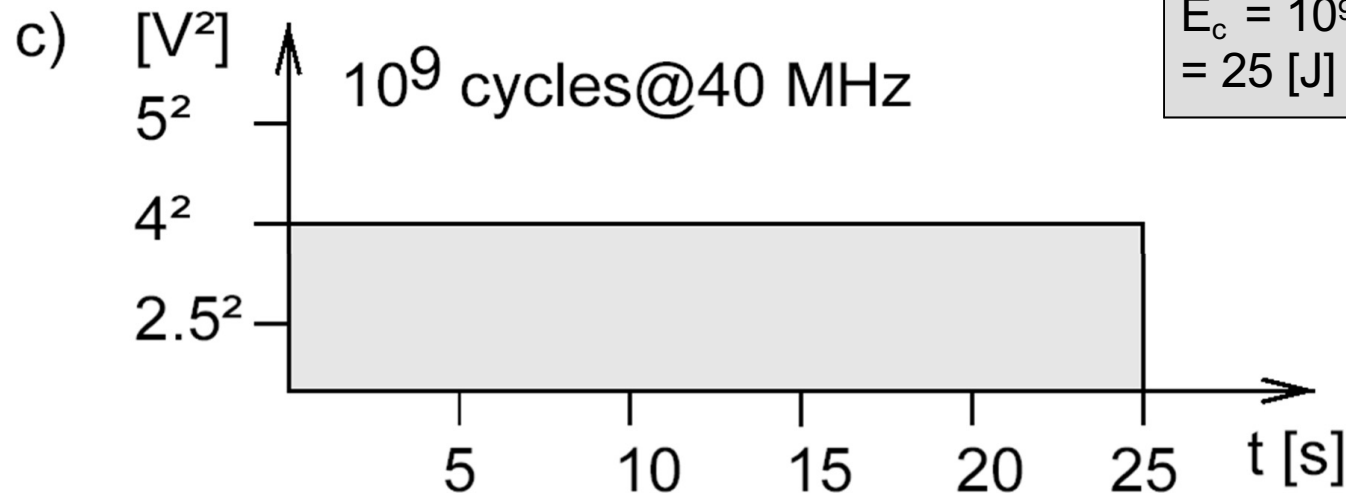
b) [V<sup>2</sup>] 750M cycles @ 50 MHz + 250M cycles @ 25



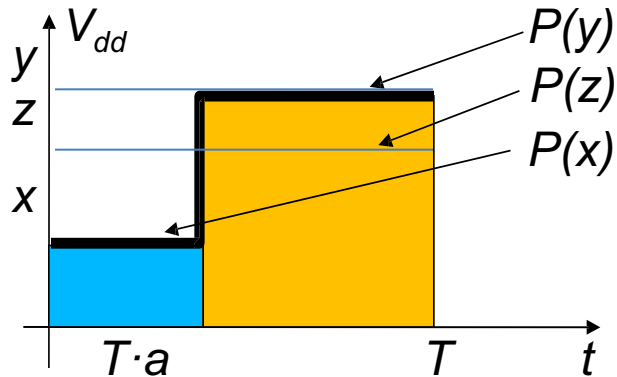
$$E_b = 750 \cdot 10^6 \times 40 \times 10^{-9} \\ + 250 \cdot 10^6 \times 10 \times 10^{-9} \\ = 32.5 \text{ [J]}$$

# Exemplu DVS: c) Tensiune optimă

$V_{dd}$ [V]	5.0	4.0	2.5
Energy per cycle [nJ]	40	25	10
$f_{max}$ [MHz]	50	40	25
cycle time [ns]	20	25	40



# DVS: Strategie optimă



$$z = a \cdot x + (1-a) \cdot y$$

Execută task-ul în timpul fix  $T$   
cu tensiunea variabilă  $V_{dd}(t)$ :

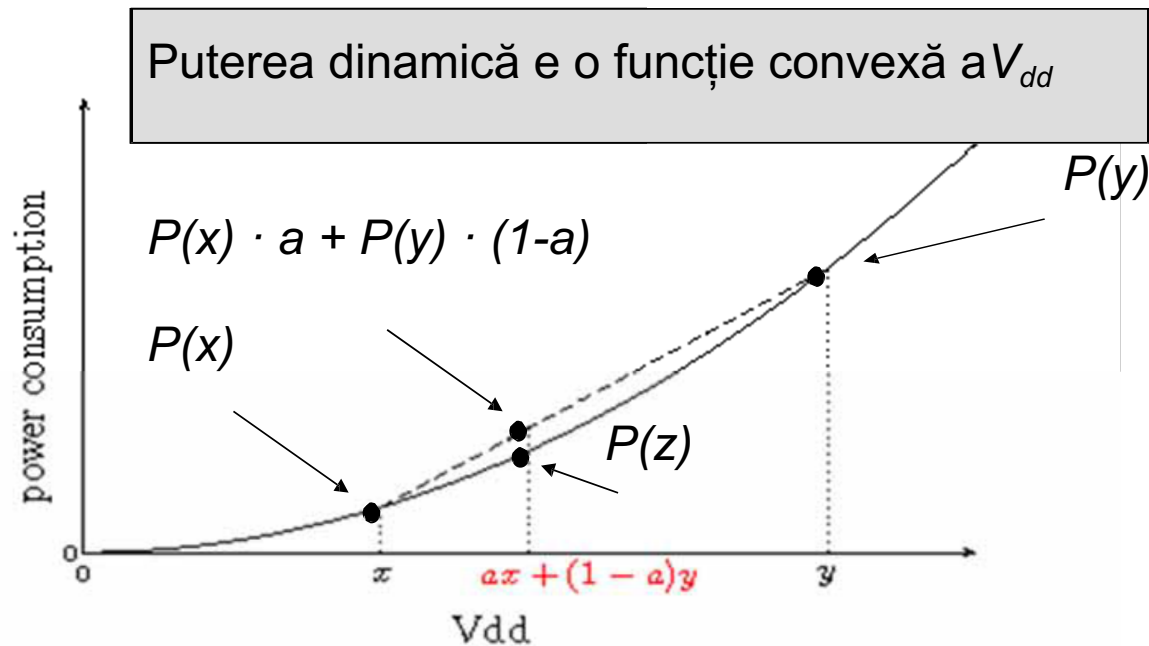
gate delay:  $\tau \sim \frac{1}{V_{dd}}$

execution rate:  $f(t) \sim V_{dd}(t)$

invariant:  $\int V_{dd}(t) dt = \text{const.}$

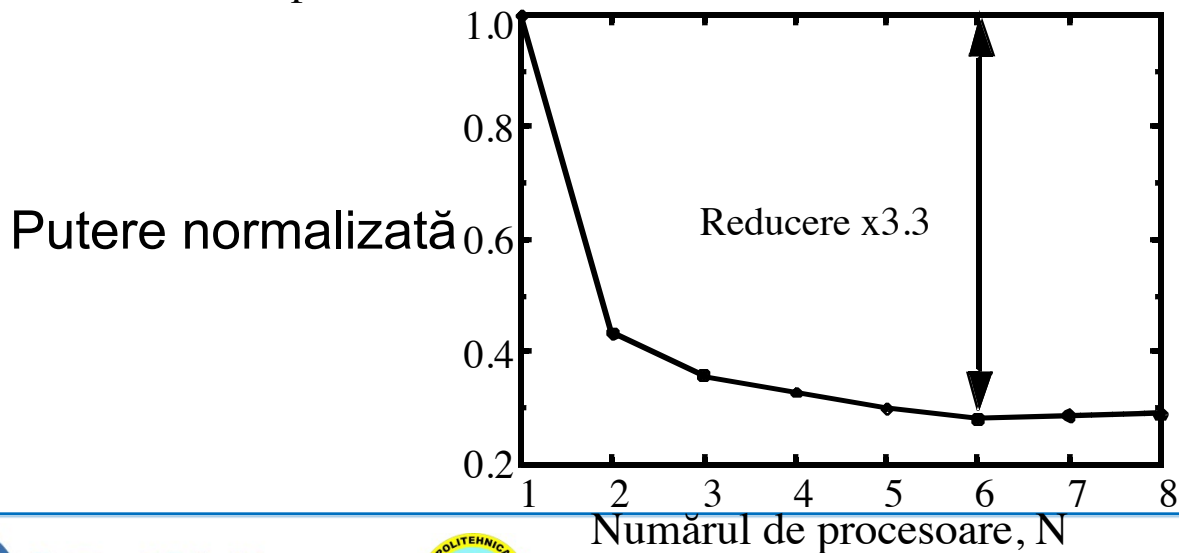
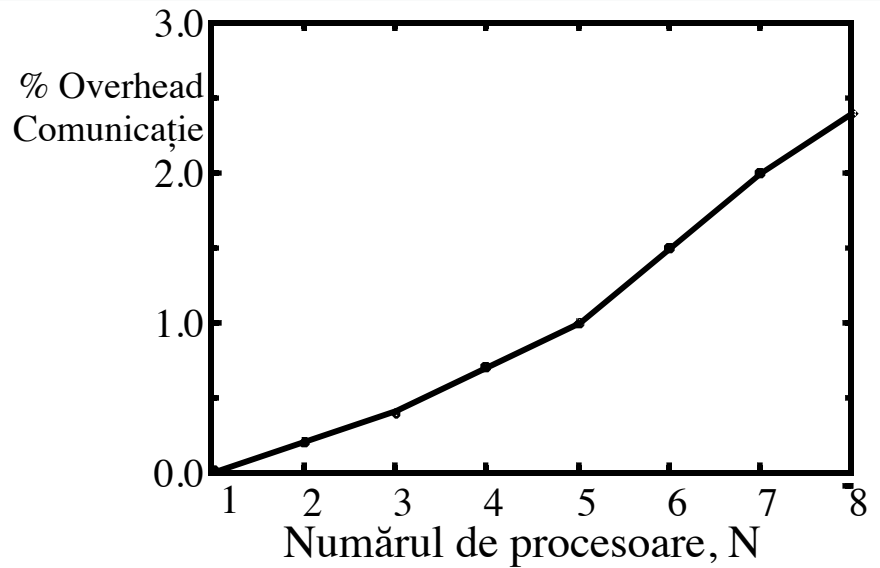
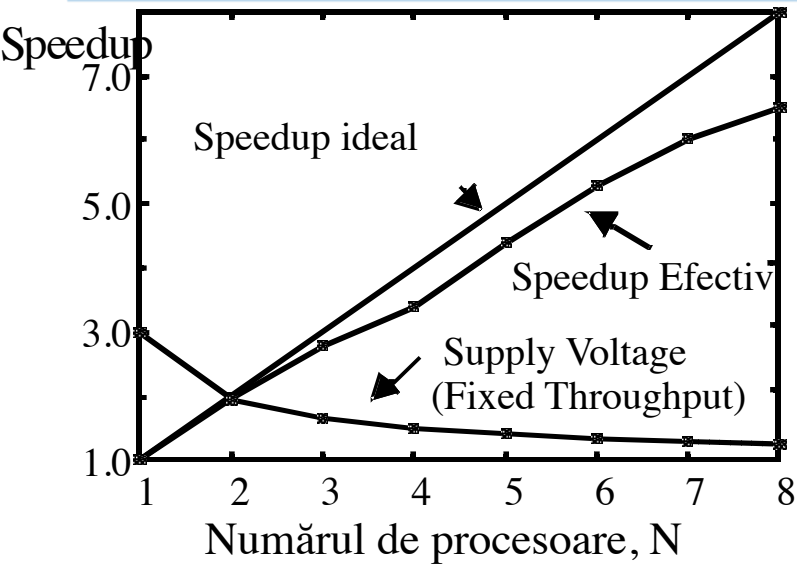
- ▶ **cazul A**: execută la tensiunea  $x$  pentru  $T \cdot a$  unități de timp și la tensiunea  $y$  pentru  $(1-a) \cdot T$  unități de timp;  
Consumul de energie:  $T \cdot (P(x) \cdot a + P(y) \cdot (1-a))$
- ▶ **cazul B**: execută la tensiunea  $z = a \cdot x + (1-a) \cdot y$  pentru  $T$  unități de timp; consumul de energie  $T \cdot P(z)$

# DVS: Strategie optimă

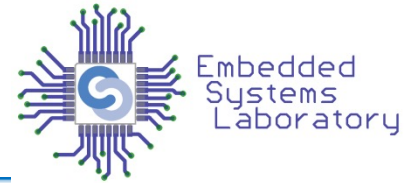


- ▶ Dacă este posibil, rularea la o frecvență constantă (tensiune) minimizează consumul de energie pentru dynamic voltage scaling:
  - **cazul A** este întotdeauna cel mai rău dacă consumul de putere este o funcție convexă a tensiunii de alimentare

# Voltage Scaling - exemplu



# Managementul energetic la nivel de sistem



Aplicație	-> Exportarea job-urilor comp. intensive
Algoritm	-> Procesare variabilă în funcție de calitate
Cod Sursă	-> Structuri de date îmbunătățite
Compilator	-> Compilare energy-aware
Runtime/OS	-> Just-in-time scheduling
ISA	-> ISA redus
Microarhitectura	-> Filter cache
Logica	-> Clock gating
Circuit	-> Reducerea tensiunii de alimentare. Paralelizare

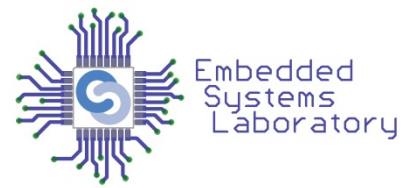
# Exemplu: Filtru Cache

- O memorie cache de mici dimensiuni în fața memoriei cache L1
- Dacă adresa înregistrează un hit în filtru, nu este trimisă mai departe
- Dacă memoria cache e mai mică -> consumul de energie per acces e mai mic.



# Reducerea consumului altor componente

---



- Înlocuirea hard-discului cu memorie flash
  - Consum scăzut și viteză comparabilă cu DRAM
  - Ștergerea are totuși o latență mare
- Echipamente wireless de comunicație
  - Idle mode la transmițător
  - Posibilitatea de modificare a puterii de emisie
- Display și backlight
  - Tehnicile de reducere a consumului pot micșora timpul de viață