



---

# Metodologie di progetto HW

Low Power Design

Last update: 16/03/09

---



## Outline

---

- Motivations
  - Sources of power dissipation in digital CMOS circuits
  - Switching power:
    - Effective capacitance reduction
    - Supply voltage scaling
  - Short-circuit power
  - Leakage power
  - Low-power design methodology
  - Conclusions
-



## Motivations

---

- Why designing low-power circuits/systems?  
Power dissipation is a primary concern due to:
    - Technological reasons:
      - Increasing of clock frequency.
      - Increasing of integration level (large number of transistors per chip).
    - Market reasons:
      - Wide diffusion of battery powered portable systems with high-throughput requirements
    - Financial reasons:
      - Reducing packaging costs and achieving energy savings.
- 

3



## Motivations (cnt'd)

---

- Wide diffusion of battery powered portable systems with high-throughput imposes to consider a *trade-off* between low-power and performance.
  - Low-power design techniques and power estimation methodologies *must* be considered at *different levels of abstraction* during the design process:
    - System Level
    - Behavioral Level
    - RT (Register Transfer) Level
    - Gate Level
    - Transistor Level
- 

4



## Motivations (cnt'd)

---

- Driving forces:
    - Advent of deep sub-micron technologies
    - Increasing market share of mobile applications.
    - Limitations of battery technology.
  
  - Deep sub-micron technologies (smaller geometries) imply higher device densities and clock frequencies
    - Greater power consumption in spite of lower supply voltages.
- 

5



## Mobile Electronics

---

- Semiconductor market for portable applications (Estimates - Source: DATE2001 Tutorial):
    - 1999: 20%
    - 2004: 40%
  - Mobile phone market (Estimates):
    - 600 M units produced in 2001
    - 1.9 B subscribers in 2004
  - PC market (Estimates):

- 1992:	23% Server	7% Mobile	70% Desktop
- 1999:	19% Server	37% Mobile	44% Desktop
- 

6



## Battery Technology

---

- Battery maximum power and capacity increase by 10-15% per year.
- Chip power requirements increase much faster (see 1999 SIA Technology Roadmap)
- Consequence:
  - Larger gap between batter technology enhancements and chip power demand.

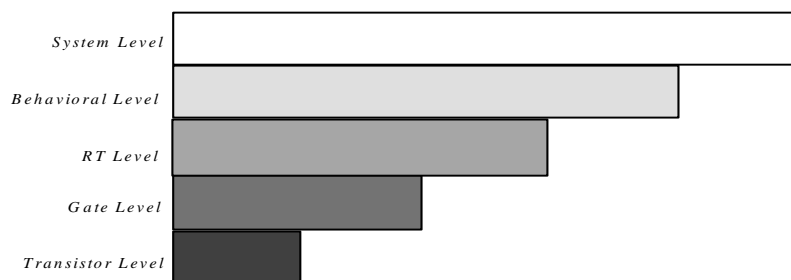
---

7



## Power Savings at Different Abstraction Levels

---



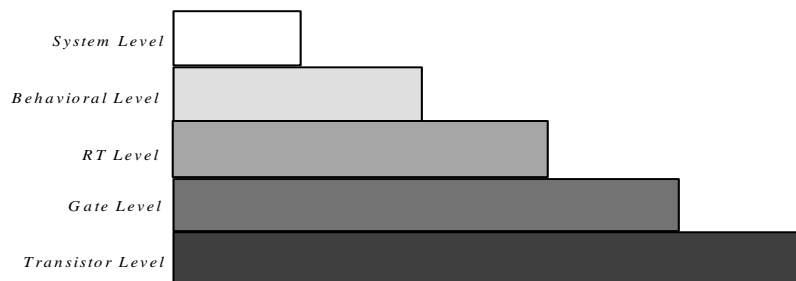
---

8



## Accuracy of Power Estimation at Different Abstraction Levels

---



---

9



## CMOS Circuits

---

- ❑ CMOS technology is predominant in the realization of today's ICs.
- ❑ CMOS devices are intrinsically low-power consuming.
- ❑ CMOS has become the reference technology.

---

10



## Sources of Power Dissipation

---

- Main sources of power dissipation in CMOS circuits:

$$P = P_{Switching} + P_{Short-Circuit} + P_{Leakage}$$

- $P_{Switching}$  due to charging and discharging load capacitors during output transitions.
- $P_{Short-Circuit}$  due to short circuit currents flowing along direct paths from  $V_{dd}$  to  $GND$  during output transitions.
- $P_{Leakage}$  due to leakage currents in diodes and transistors.

---

11



## Design for Low-Power

---

- Minimization of  $P_{Short-Circuit}$  and  $P_{Leakage}$  :
  - Technology problem
- Minimization of  $P_{Switching}$  :
  - Design problem
  
- $P_{Switching}$  is predominant  $\Rightarrow$  most part of low-power techniques aims at reducing the switching power

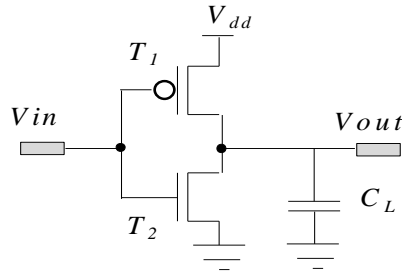
---

12



## Switching Power

### CMOS Inverter:

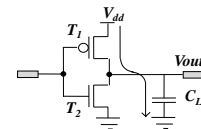


13

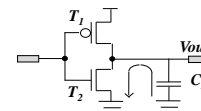


## Switching Power (cnt'd)

- Transition  $0 \rightarrow V_{dd}$  at the output node:
  - Energy dissipated by  $T_1$  (PMOS):  $\frac{1}{2} C_L V_{dd}^2$
  - Energy stored in  $C_L$ :  $\frac{1}{2} C_L V_{dd}^2$
  - Energy drawn from the power supply:  $C_L V_{dd}^2$



- Transition  $V_{dd} \rightarrow 0$  at the output node:
  - Energy dissipated by  $T_2$  (NMOS):  $\frac{1}{2} C_L V_{dd}^2$



- If  $0 \rightarrow V_{dd}$  and  $V_{dd} \rightarrow 0$  transitions occur at the frequency  $f_{CLK}$ , the average power drawn from the power supply due to the switching activity is:

$$P_{SW} = \frac{1}{2} C_L V_{dd}^2 f_{CLK}$$

14



## Switching Power (cnt'd)

---

- In general, switching does not occur at the clock frequency.
- Let  $\alpha$  be the *switching activity* of a logic gate as the probability that the output node switches at each clock cycle .
- The average  $P_{SW}$  for a CMOS gate is given by:

$$P_{SW} = \frac{1}{2} \alpha C_L V_{dd}^2 f_{CLK}$$

---

15



## Effective Capacitance

---

- We define *effective capacitance*  $C_{eff}$  (or switched capacitance) as the term including output load capacitance and switching activity:

$$C_{eff} = \alpha C_L$$

- Design for low-power:
  - Reduction of  $C_{eff}$ 
    - Applicable at all levels of design abstraction
    - Many design techniques proposed.
  - $V_{dd}$  and/or  $f_{CLK}$  scaling:
    - Very effective
    - Big impact on performance

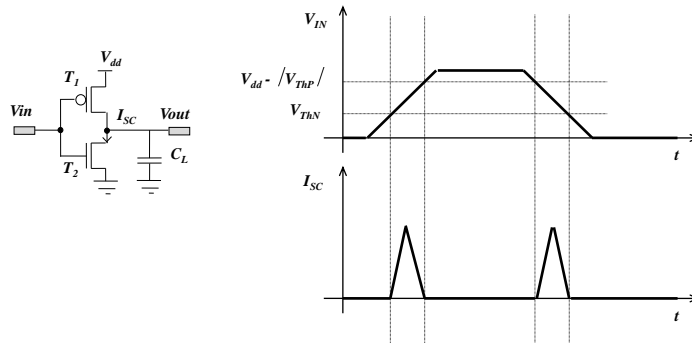
---

16



## Short-Circuit Power

- Ascending and descending signal ramps (of finite length) create a direct current path from  $V_{dd}$  and  $GND$  for short periods during logic transitions.



17



## Short-Circuit Power (cnt'd)

- Let  $V_{ThN}$  and  $V_{ThP}$  be the threshold voltages of the NMOS and PMOS transistors. Let  $V_{IN}$  be the input voltage.
- If the following condition is satisfied:

$$V_{ThN} < V_{IN} < V_{dd} - V_{ThP}$$

there is a conductive path between  $V_{dd}$  and  $GND$  because the NMOS and PMOS transistors are on simultaneously.

- If  $I_{SC}$  is the average short-circuit current flowing from  $V_{dd}$  to  $GND$ , the average power due to  $I_{SC}$  is given by:

$$P_{SC} = I_{SC} V_{dd}$$

18



## Short-Circuit Power (cnt'd)

---

- $I_{SC}$  is significant when rise/fall time of inputs is much larger than the rise/fall time of outputs. This is because the direct path from  $V_{dd}$  to  $GND$  is active for a longer time.
- If  $V_{dd}$  satisfies the condition:

$$V_{dd} < V_{ThN} + |V_{ThP}|$$

then  $I_{SC} = 0$  because NMOS and PMOS transistors are never on simultaneously.

---

19



## Leakage Power

---

- There are two types of leakage power:
  - Reverse-bias diode current through transistor drain:  $I_L$
  - Sub-threshold currents through the channel of an off transistor:  
 $I_{ds}$

- The leakage power dissipated by a CMOS gate due to  $I_L$  and  $I_{ds}$  is:

$$P_{Leakage} = (I_L + I_{ds}) V_{dd}$$

- In general, this value is approximately the 5% of the total power for 350nm technology. It can be around the 50% when you consider 90nm or below.

---

20



## Reduction of $C_{eff}$

---

- Large power savings can be obtained by minimizing the effective capacitance  $C_{eff}$  defined as:  $C_{eff} = \alpha C_L$
- The reduction of  $C_{eff}$  can be done at different abstraction levels.
  
- Reduction of the load capacitance  $C_L$ :
  - Using dynamic CMOS logic.
  - Using pass-gate logic.
  - Dimensioning transistors and interconnections.
  - Optimizations during the place&route phase.

---

21



## Reduction of $C_{eff}$ (cnt'd)

---

- Reduction of the switching activity  $\alpha$ :
  - Optimization based on parallel and pipelined architectures
  - Encoding techniques and data representation techniques
  - Logic minimization and technology mapping.
  - Shut-down techniques based on clock gating.

---

22



## Reduction of $C_{eff}$ (cnt'd)

---

- Factors influencing  $C_{eff}$ :
  - Logic function
  - Technology
  - Input probability
  - Circuit topology
- We examine each factor in detail

---

23



## Factors Influencing $C_{eff}$ : Logic Function

---

- Example: 2-Input NOR gate with two inputs  $A$  e  $B$  implemented in CMOS static technology.
  - Assume  $A$  and  $B$  independent
  - Assume  $A$  and  $B$  equiprobable  $p(A = 1) = p(B = 1) = 1/2$
  - Assume only one input transition per clock cycle is allowed.
- The probability for the output to be 1 is:

$$p(O = 1) = (1 - p(A = 1)) (1 - p(B = 1)) = 1/4$$

- The probability for the output to be 0 is:

$$p(O = 0) = 1 - p(O = 1) = 3/4$$

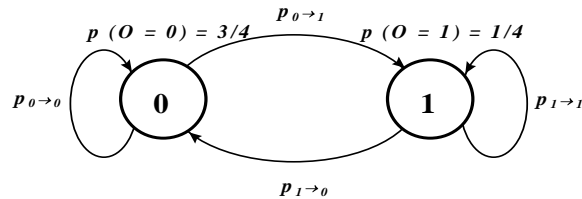
---

24



## Factors Influencing $C_{\text{eff}}$ : Logic Function

- State Transition Diagram for the NOR gate:



- For the output node 0, the transition probability  $p_{0 \rightarrow 1}$  is given by the probability that the current state is 0 times the probability that the next state is 1. We have:

$$p_{0 \rightarrow 1} = p(O=0) p(O=1) = \frac{3}{4} \frac{1}{4} = \frac{3}{16}.$$

25



## Factors Influencing $C_{\text{eff}}$ : Logic Function

- Similarly:

$$p_{1 \rightarrow 0} = p(O=1) p(O=0) = \frac{1}{4} \frac{3}{4} = \frac{3}{16}$$

$$p_{0 \rightarrow 0} = p(O=0) p(O=0) = \frac{3}{4} \frac{3}{4} = \frac{9}{16}$$

$$p_{1 \rightarrow 1} = p(O=1) p(O=1) = \frac{1}{4} \frac{1}{4} = \frac{1}{16}$$

$$\Rightarrow \alpha = p_{0 \rightarrow 1} + p_{1 \rightarrow 0} = \frac{3}{8}$$

26



## Factors Influencing $C_{\text{eff}}$ : Logic Function

- Example: 2-Input XOR gate with two inputs  $A$  e  $B$  implemented in CMOS static technology.
  - Assume  $A$  and  $B$  independent
  - Assume  $A$  and  $B$  equiprobable  $p(A = 1) = p(B = 1) = 1/2$
  - Assume only one input transition per clock cycle is allowed.
- The probability for the output to be 1 is:

$$p(O = 1) = (1 - p(A = 1)) p(B = 1) + p(A = 1)(1 - p(B = 1)) = 1/2$$

- The probability for the output to be 0 is:

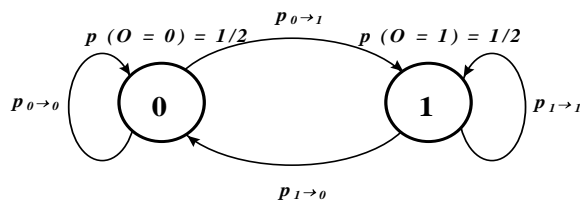
$$p(O = 0) = 1 - p(O = 1) = 1/2$$

27



## Factors Influencing $C_{\text{eff}}$ : Logic Function

- State Transition Diagram of the XOR gate:



- For the output node 0, the transition probability  $p_{0 \rightarrow 1}$  is given by the probability that the current state is 0 times the probability that the next state is 1. We have:

$$p_{0 \rightarrow 1} = p(O = 0) p(O = 1) = 1/2 \cdot 1/2 = 1/4$$

28



## Factors Influencing $C_{\text{eff}}$ : Logic Function

---

- Similarly:

$$p_{1 \rightarrow 0} = p(O = 1) p(O = 0) = \frac{1}{2} \frac{1}{2} = \frac{1}{4}$$

$$p_{0 \rightarrow 0} = p(O = 0) p(O = 0) = \frac{1}{2} \frac{1}{2} = \frac{1}{4}$$

$$p_{1 \rightarrow 1} = p(O = 1) p(O = 1) = \frac{1}{2} \frac{1}{2} = \frac{1}{4}$$

$$\Rightarrow \alpha = p_{0 \rightarrow 1} + p_{1 \rightarrow 0} = \frac{1}{2}$$

---

29



## Factors Influencing $C_{\text{eff}}$ : Technology

---

- The choice is between static and dynamic CMOS technology.
  - $C_L$  of a *static* CMOS gate is charged to  $V_{dd}$  any time a  $0 \rightarrow 1$  transition at the output node is required, and it is discharged to  $0$  any time a  $1 \rightarrow 0$  transition at the output node is required.
  - $C_L$  of a *dynamic* CMOS gate is pre-charged to  $V_{dd}$  at each clock cycle, and it is discharged to  $0$  any time a  $1 \rightarrow 0$  transition at the output node is required.
  - Consequence:
$$\alpha (\text{dynamic CMOS}) \geq \alpha (\text{static CMOS})$$
  - For the load capacitance, we usually have:
$$C_L (\text{dynamic CMOS}) \leq C_L (\text{static CMOS})$$
- 

30



## Factors Influencing $C_{\text{eff}}$ : Technology

---

- Example: 2-Input NOR gate with two inputs  $A$  e  $B$  implemented in CMOS dynamic technology.
    - Assume  $A$  and  $B$  independent
    - Assume  $A$  and  $B$  equiprobable  $p(A = 1) = p(B = 1) = 1/2$
    - Assume only one input transition per clock cycle is allowed.
  - The probability for the output node to be discharged is:
$$p(O = 0) = 3/4$$
  - The probability for  $C_L$  to be re-charged at the next clock cycle equals:
$$p(O = 0)$$
  - Therefore:
$$p_{0 \rightarrow 1}(\text{static CMOS}) = 3/16 < p_{0 \rightarrow 1}(\text{dynamic CMOS}) = 3/4$$
- 

31



## Factors Influencing $C_{\text{eff}}$ : Technology

---

- In static CMOS gates, the transition probability depends on both input probability and previous state.
  - In dynamic CMOS gates, the transition probability depends only on input probability.
  - In static CMOS, if the inputs do not change from the previous clock cycle, the gate output does not switch.
  - In dynamic CMOS, the gate output may switch even if the inputs do not change from the previous clock cycle.
- 

32



## Factors Influencing $C_{\text{eff}}$ : Input Probability

- Example: 2-Input NOR gate with two inputs  $A$  e  $B$  implemented in CMOS static technology.
- We assume the inputs to the logic gates are not equiprobable and let  $p(A = 1)$  e  $p(B = 1)$  be the probabilities of inputs  $A$  and  $B$ .

- The probability for the output node to be 1 is:

$$p(O = 1) = (1 - p(A = 1)) (1 - p(B = 1))$$

- The probability for the output node to be 0 is:

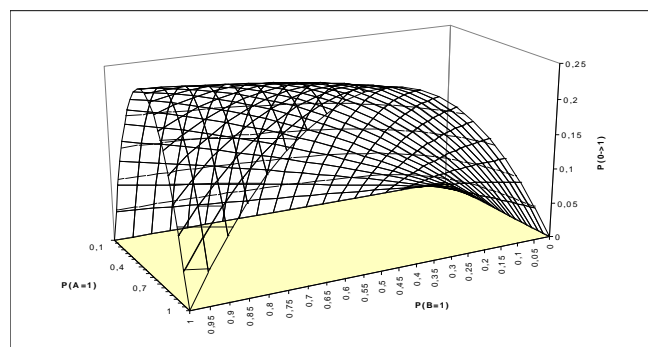
$$p(O = 0) = 1 - p(O = 1)$$

33



## Factors Influencing $C_{\text{eff}}$ : Input Probability

- The probability for the output node to have a  $0 \rightarrow 1$  transition is:  $p_{0 \rightarrow 1} = p(O = 0) p(O = 1) = (1 - (1 - p(A = 1)) (1 - p(B = 1))) (1 - p(A = 1)) (1 - p(B = 1))$

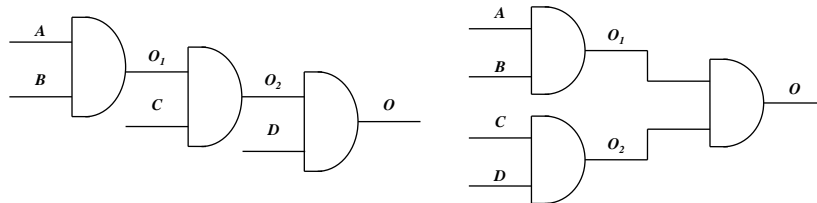


34



## Factors Influencing $C_{\text{eff}}$ : Circuit Topology

- Circuit topology can have a strong impact on the global switching activity.
- Example: Chain and tree implementations of a 4-Input AND gate:



- Assume static CMOS gates.
- Assume  $A, B, C, D$  independent and equiprobable:  

$$p(A = 1) = p(B = 1) = p(C = 1) = p(D = 1) = \frac{1}{2}$$

35



## Factors Influencing $C_{\text{eff}}$ : Circuit Topology

- Probability that the output of a 2-input AND gate is 1:

$$p(O = 1) = p(A = 1) p(B = 1) = \frac{1}{4}$$

- Probability that the output of a 2-input AND gate is 0:

$$p(O = 0) = 1 - p(O = 1) = \frac{3}{4}$$

- Probability that the output of a 2-input AND gate makes a  $0 \rightarrow 1$  transition is given by:

$$p_{0 \rightarrow 1} = p(O = 0) p(O = 1) = \frac{3}{4} \frac{1}{4} = \frac{3}{16}$$

- Switching activity of the output of a 2-input AND gate:

$$\alpha = p_{0 \rightarrow 1} + p_{1 \rightarrow 0} = \frac{3}{8}$$

36



## Factors Influencing $C_{\text{eff}}$ : Circuit Topology

---

- Comparison between chain and tree circuit:

	<i>Cascade</i>	<i>Tree</i>
$p(O_1=1)$	$1/4$	$1/4$
$p(O_1=0)$	$3/4$	$3/4$
$p(O_2=1)$	$1/8$	$1/4$
$p(O_2=0)$	$7/8$	$3/4$
$p(O=1)$	$1/16$	$1/16$
$p(O=0)$	$15/16$	$15/16$
$\alpha(O_1)$	$3/8$	$3/8$
$\alpha(O_2)$	$7/32$	$3/8$
$\alpha(O)$	$15/128$	$15/128$

- Globally, the chain implementation has a lower switching activity than the tree implementation (see  $\alpha(O_2)$ ) for equiprobable inputs.

---

37



## Factors Influencing $C_{\text{eff}}$ : Circuit Topology

---

- We have considered only the static behavior of the circuit, i.e. no timing behavior. In other words, we have considered only the *static* component of the switching activity.
- Timing skews between signals may cause spurious transitions (glitches) resulting in extra power dissipation. The *dynamic* component of the switching activity is due to glitches.

---

38



## Factors Influencing $C_{\text{eff}}$ : Circuit Topology

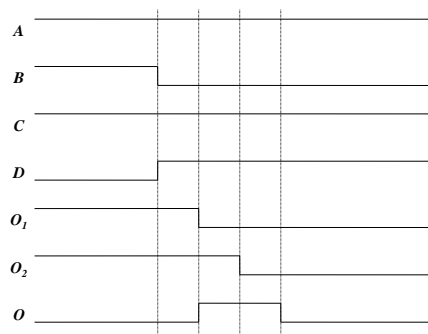
- Example: Chain and tree implementations of 4-Input AND gate.
- Let us consider the input transition:  $1110 \rightarrow 1011$
- If we ignore the timing behavior (*zero delay model* for each logic gate)  $\Rightarrow$  no output transitions.
- If we assume *unit delay model* for each logic gate  $\Rightarrow$  for the chain circuit we obtain the following timing diagram.

39



## Factors Influencing $C_{\text{eff}}$ : Circuit Topology

- Timing diagram for the chain circuit:



40



## Factors Influencing $C_{\text{eff}}$ : Circuit Topology

---

- Due to the timing skews through the logic, the output node makes an extra transition (glitch). This is because the chain circuit is *unbalanced* (paths of different lengths).
  - The tree circuit is glitch free. This is because it is *balanced* (paths of equal lengths)
  - In general, we can reduce the spurious transitions by using balanced paths.
  - The example demonstrates that a given circuit topology can have a smaller *static* component of switching activity, but it can have a larger *dynamic* component.
- 

41



## Supply Voltage Scaling

---

- Large power savings can be obtained by reducing the supply voltage, due to the quadratic dependence of the energy per transition on  $V_{dd}$ .
  - Power savings are relatively independent of:
    - Circuit function
    - Circuit technology
  - Supply voltage scaling is applicable at different abstraction levels of the design development.
  - Supply voltage scaling affects circuit speed.
- 

42



## Supply Voltage Scaling

- Circuit speed decreases as  $V_{dd}$  approaches the threshold voltage  $V_{Th}$
- By using a first-order approximation, the delay  $T_d$  of a CMOS gate is given by:

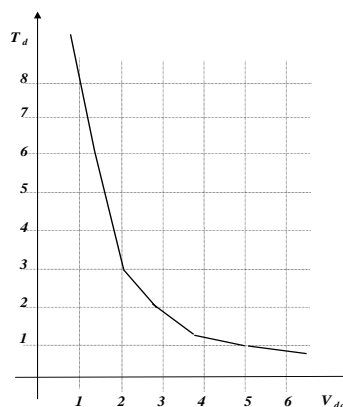
$$T_d = \frac{C_{out} V_{dd}}{I} = \frac{C_{out} V_{dd}}{\eta (W/L)(V_{dd} - V_t)^2}$$

- Where  $\eta$  depends on the technology, and  $W$  and  $L$  are channel width and length of the CMOS transistors.
- It is desirable to operate at the lowest possible speed, since this allows that highest  $V_{dd}$  scaling.

43



## Normalized Delay vs. Supply Voltage



$T_d = \text{Normalized Delay}$   
 $V_{dd} = \text{Supply Voltage (Volt)}$

44



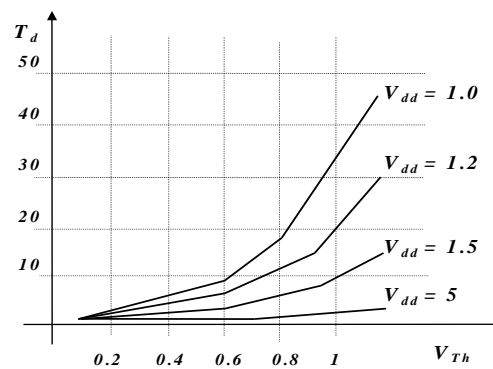
## Supply Voltage Scaling

- Main goal: to reduce power while preserving circuit speed and computational throughput  $\Rightarrow$  we need to compensate the delay increment due to the  $V_{dd}$  scaling.
- Two approaches:
  - Threshold voltage scaling proportionally to  $V_{dd}$  scaling to preserve performance.
  - Architecture-driven supply voltage scaling based on:
    - Parallelization
    - Pipelining

45



## Normalized Delay vs. Threshold Voltage



$T_d = \text{Normalized Delay}$   
 $V_{Th} = \text{Threshold Voltage [Volt]}$   
 $V_{dd} = \text{Supply Voltage [Volt]}$

46



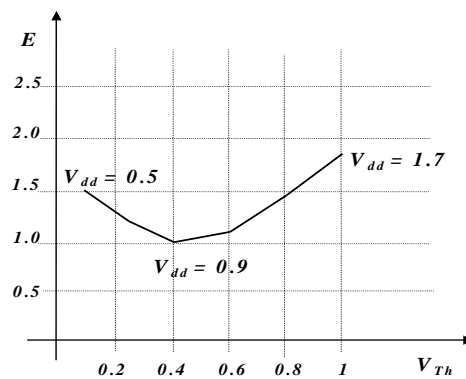
## Threshold Voltage Scaling

- Reducing the threshold voltage allows the supply voltage to be scaled down to lower  $P_{SW}$  without loss in speed.
- If threshold voltage scaling is required, low-threshold CMOS devices must be used for the design.
- The limit on threshold voltage scaling is imposed by:
  - The noise margin
  - The increase of sub-threshold current ( $I_{DS}$ ).
- As  $V_{Th}$  decreases, we need to find a trade-off between  $P_{SW}$  (decreasing as  $V_{Th}$  decreases) and  $P_{Leakage}$  (increasing as  $V_{Th}$  decreases).

47



## Normalized Energy vs Threshold Voltage



$E$  = Normalized Energy  
 $V_{Th}$  = Threshold Voltage [Volt]  
 $V_{dd}$  = Supply Voltage [Volt]

48



## Architecture-Driven Supply Voltage Scaling

---

- Strategy:
    - Modify architecture of the system so as to make it faster
    - Reduce  $V_{DD}$  so as to restore the original speed.
    - Power consumption has decreased.
  - The most common architectural changes rely on the exploitation of parallelization and pipelining.
  - Draw-back:
    - Additional circuitry required to compensate the speed degradation may dominate, and the power consumption may increase
  - Consequence:
    - Parallelism and pipelining do not always pay-off.
- 

49



## Not Only Supply Voltage Scaling

---

- Supply voltage scaling is not the only possible solution to reduce power consumption
  - Remarkable results can be obtained through minimization of  $C_{eff}$
  - Techniques have been developed to reduce both the capacitive load ( $C_L$ ) and the switching activity  $\alpha$
  - $C_{eff}$  minimization must be targeted at all stages of the design process.
- 

50



## Conclusions

---

- The electronic market demands low-power circuits and systems.
- We have analyzed the major sources of power dissipation in CMOS circuits.
- We have presented a quick overview of the the most used approach for low-power design.
- Power optimization is required at all levels of the design process.
- Optimization techniques applied at high abstraction level can reach large power savings.

---

51



---

## Tecniche di Progetto a Livello Logico

- Codifica degli stati.
- Retiming
- Pre-Calcolo.
- Condizionamento del segnale di clock (*clock gating*).



## Tecniche di Codifica degli Stati

---

- Per sistemi orientati al controllo, la specifica iniziale di una Macchina a Stati Finiti (FSM) viene descritta in termini di Grafo di Transizione degli Stati (STG).
- In generale, dato lo STG di una FSM, il problema della codifica degli stati consiste nel trovare un assegnamento di codici binari per gli stati in modo da minimizzare una data funzione di costo.
- La funzione di costo di un algoritmo di assegnamento degli stati finalizzato alla minimizzazione della potenza dissipata deve tener conto della riduzione del numero di transizioni logiche nella codifica degli stati tra due cicli di clock successivi.
- Il problema dell'assegnamento degli stati è un problema NP-completo.

---

- 53 -



## Tecniche di Codifica degli Stati (cont.)

---

- Definizione del problema:  
dato uno STG assegnare dei codici binari agli stati in modo da minimizzare una data funzione di costo.
- In generale, la funzione di costo  $C$  deve tener conto della distanza di Hamming  $H(s_i, s_j)$  tra i codici binari corrispondenti agli stati  $s_i$  e  $s_j$  tra i quali può avvenire una transizione di stato.
- Struttura della Funzione di Costo:  
$$C = \sum_{ij} H(s_i, s_j)$$
  
 $H(s_i, s_j)$  = distanza di Hamming tra i codici binari corrispondenti agli stati  $s_i$  e  $s_j$
- Una funzione di costo più accurata dovrebbe considerare dei fattori di peso corrispondenti ai lati del grafo che tengano conto della probabilità di ogni possibile transizione di stato.

---

- 54 -



## Tecniche di Codifica degli Stati (cont.)

- Si consideri uno STG pesato, i cui pesi rappresentino la probabilità di transizione da uno stato  $s_i$  ad uno stato  $s_j$ , cioè la probabilità che, quando la FSM si trova nello stato  $s_i$ , nel prossimo ciclo di clock si trovi nello stato  $s_j$ .
- Dato uno STG pesato, l'idea base consiste nell'assegnare codici binari *vicini* agli stati connessi da lati con peso più alto.
- Struttura della nuova funzione di costo:

$$C = \sum_j W_{ij} H(s_i, s_j)$$

$W_{ij}$  = peso assegnato al lato del grafo dallo stato  $s_i$  allo stato  $s_j$ .

- Algoritmo:
  - calcolare le probabilità di transizione di stato tra le coppie di stati della FSM ed etichettare con questi valori i corrispondenti lati del grafo;
  - considerare una combinazione lineare tra le probabilità di transizione di stato da  $s_i$  a  $s_j$  e fattori di area come coefficienti di peso  $W_{ij}$  nella funzione di costo.

- 55 -



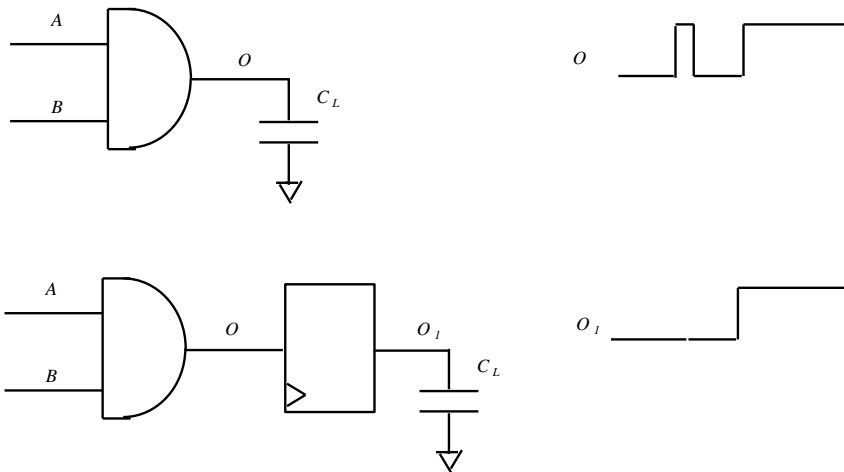
## Tecniche di Retiming

- La posizione dei registri in un circuito sequenziale può impattare fortemente l'area, le prestazioni e la potenza dissipata dal circuito stesso.
- L'osservazione di base è che, in un circuito sequenziale sincrono con un segnale di clock primario, le uscite dei registri possono avere al più una transizione per ogni ciclo di clock.
- Questo concetto può essere usato per ridurre il numero di glitch o transizioni spurie.
- Le tecniche di *ReTiming* si basano sulla selezione di un insieme di porte logiche del circuito tali che, se vengono posti dei registri alle loro uscite, l'attività di transizione globale del circuito venga minimizzata.

- 56 -



## Tecniche di Retiming (cont.)



- 57 -



## Tecniche di Retiming (cont.)

- **Algoritmo:**
  - Selezionare un insieme di porte logiche caratterizzate da:
    - elevato numero di glitch sulle uscite;
    - probabilità elevata che i glitch presenti sulle uscite si possano propagare alle uscite dei nodi successivi.
  - Aggiungere un registro per ogni uscita dell'insieme di porte logiche selezionate.
- Quando i registri sono già presenti nel circuito sequenziale, tale tecnica è finalizzata a spostare i registri tra i vari livelli della logica combinatoria, in modo da modificare la temporizzazione del circuito e diminuire la potenza dissipata, ma senza impattare la funzionalità del circuito.

- 58 -



## Tecniche di Pre-Calcolo e di Clock Gating

---

- Sistemi digitali complessi generalmente contengono porzioni di logica che non eseguono calcoli utili ad ogni ciclo di clock (ad esempio le ALU di un microprocessore).
- L'idea base consiste nello spegnere la logica non usata durante alcuni cicli di clock (**logic shutdown**), con l'obiettivo di limitare il consumo di potenza.
- Disabilitare le porte logiche dallo svolgere transizioni non funzionalmente utili causa una diminuzione nell'attività di commutazione globale di un circuito.
- In passato, questo tipo di tecniche erano applicate manualmente dai progettisti. Con il crescere della complessità dei circuiti integrati, sono state sviluppate tecniche automatiche per lo shutdown di blocchi logici.

---

- 59 -



## Tecniche di Pre-Calcolo

---

- Il metodo si basa sull'idea di duplicare parti della logica con l'obiettivo di:
  - Pre-calcolare i valori di uscita della logica un ciclo di clock prima che vengano richiesti.
  - Usare questi valori pre-calcolati delle uscite per ridurre l'attività di commutazione globale della logica durante il ciclo di clock successivo.
- Conoscere i risultati delle uscite un ciclo di clock in anticipo permette alla logica originale di essere spenta durante il ciclo di clock successivo.
- La dimensione della logica introdotta per effettuare il pre-calcolo deve essere mantenuta sotto controllo. Il vantaggio si ha quando si può considerare solo un sotto-insieme delle condizioni di ingresso per il quale le uscite vengono pre-calcolate.

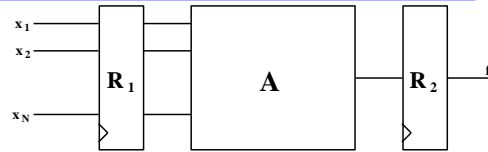
---

- 60 -

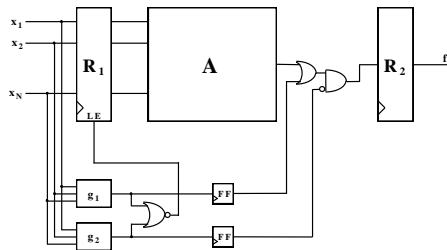


## Tecniche di Pre-Calcolo (cont.)

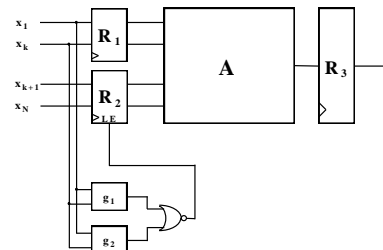
- Architettura di riferimento



### Architettura di pre-calcolo del primo tipo



### Architettura di pre-calcolo del secondo tipo



- 61 -



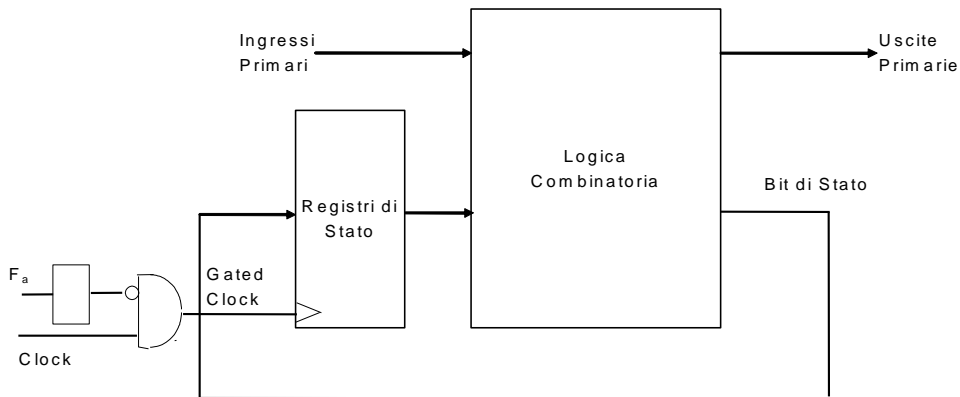
## Tecniche di Clock Gating

- Durante il progetto di sistemi complessi a bassa potenza, è comune fermare selettivamente il segnale di clock del ciclo successivo in parti della logica dove non vengono svolti calcoli utili alla funzionalità globale del circuito.
- In generale, il segnale di clock viene disabilitato in corrispondenza di condizioni di riposo del circuito sequenziale.
- Generalmente la determinazione delle condizioni durante le quali è possibile fermare il segnale di clock è responsabilità del progettista.
- Alcuni metodi automatici sono stati recentemente proposti per ricavare queste condizioni direttamente dalle specifiche del circuito, descritte attraverso STG. Tali metodi si basano sulla ricerca degli auto-anelli dello STG.

- 62 -



## Tecniche di Clock Gating (cont.)



- 63 -



## Conclusioni

- Una considerevole ottimizzazione della potenza dissipata può essere ottenuta ai diversi livelli di astrazione.
- A livello comportamentale, sono stati analizzati approcci basati su trasformazioni finalizzate alla riduzione della tensione di alimentazione e della capacità effettiva.
- A livello architetturale, sono state illustrate tecniche finalizzate a:
  - Riduzione della tensione di alimentazione basate su architetture parallele e pipelined.
  - Riduzione della capacità effettiva basate sulla scelta della rappresentazione dei dati e la codifica, il bilanciamento dei percorsi e la condivisione delle risorse.
- A livello logico, sono state presentate tecniche di codifica degli stati, ri-temporizzazione, pre-calcolo e gated clock.

- 64 -