

# CMOS Digital VLSI Design

Prof. Habil. Dr. Ing. Decebal Popescu

Modulul 5

Metoda efortului logic

# Efortul logic

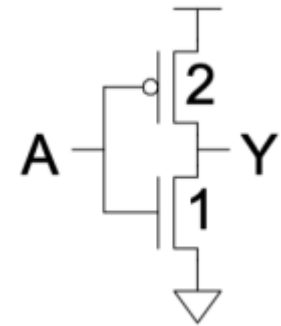
- Introducere
- Noțiuni de bază
- Calculul efortului logic pentru porțile logice
- Rețele logice cu mai multe stagii
- Recapitulare

# Introducere

- O metodă de determinare a întârzierii între intrările primare și ieșirile primare ale unui circuit combinațional
- Cât de mari trebuie să fie tranzistoarele unei porți logice pentru a atinge cel mai mic delay ?
- Câte stagii de logică trebuie utilizate pentru a obține delay-ul minim ?
- Metoda efortului logic este o metodă simplă de minimizare a delay-ului într-un circuit logic. Comparăm delay-urile estimate pentru diferite structuri logice și apoi selectăm structura cea mai rapidă
- Efortul logic specifică cea mai bună cale logică și dimensiunile tranzistoarelor pentru un load dat.

# Efortul logic

- Cunoaștem din cursurile anterioare că *aspect ratio* între pMOS și nMOS în cazul unui inversor este 2:1 pentru că  $tp_{LH} = tp_{HL}$
- Spuneam anterior că  $L$  este fix din cauza fabricantului dar și din cauza tehnologiei folosite. Ca atare doar  $W$  poate fi modificat.
- Proiectarea unui circuit pentru a atinge viteza maximă or pentru a respecta anumite constrângeri de întârzieri
- Având în vedere că creșterea a lui  $\frac{W}{L}$  presupune o scădere a întârzierii am fi tentați să tot creștem aspect ratio



# Noțiuni de bază

- Modelul efortului logic descrie întârzierile cauzate de load-urile capacitive care sunt conduse de porțile logice precum și de topologia porților logice
- Întârzierea absolută a unei porți este  $d_{(abs)} = d * \tau$ 
  - $d$  este unitatea de întârziere a porții – întârzierea porții logice
  - $\tau$  este unitate de întârziere care caracterizează un proces dat. Pentru un procesul de  $0.6\mu$ ,  $\tau = 50ps$ . Pentru un proces de  $0.18\mu$ ,  $\tau = 12ps$ .

# Noțiuni de bază

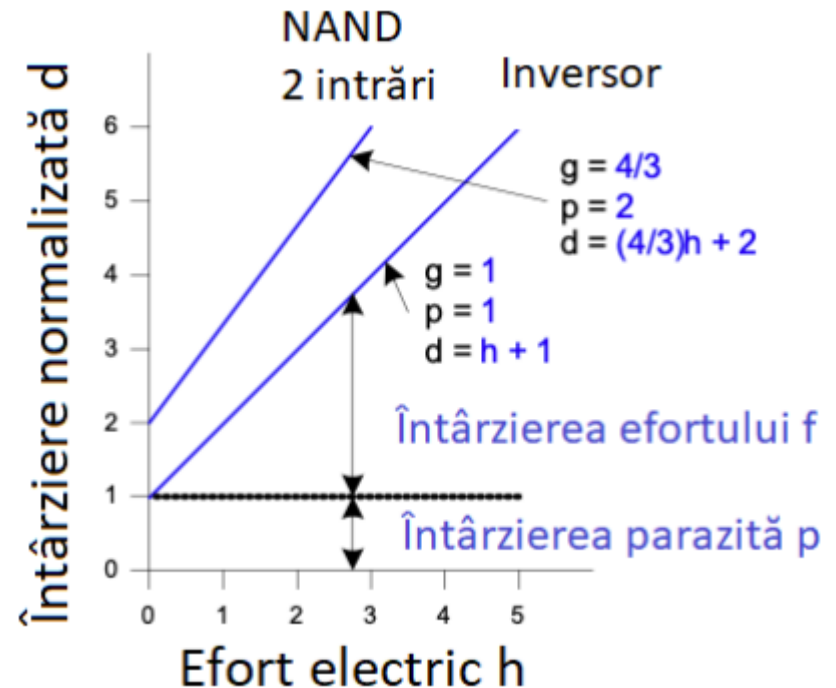
- Întârzierea  $d$  a unei porți logice este compusă din:

$$d = f + p$$

- unde:
  - $f$  = efortul întârzierii sau efortul stagiului
  - $p$  este întârzierea parazită
- Efortul întârzierii are 2 componente:  $f = g * h$ 
  - $g$  = efort logic
  - $h$  = efort electric
- $h = \frac{C_{out}}{C_{in}}$  - fan-out.  $C_{out}$  și  $C_{in}$  sunt capacitățile de intrare și de ieșire

# Noțiuni de bază

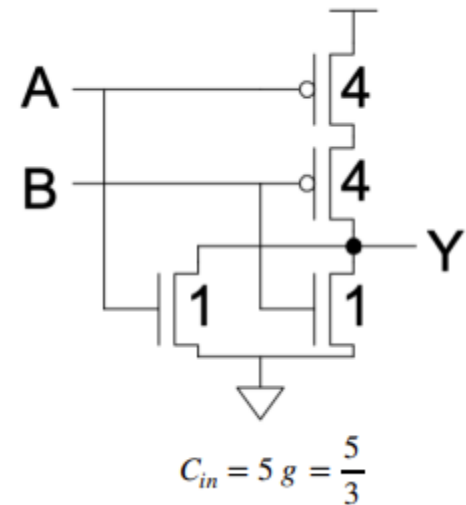
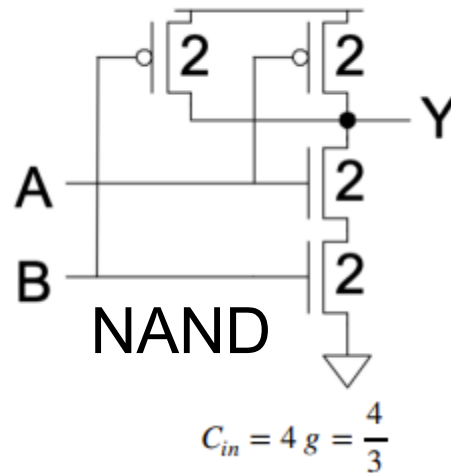
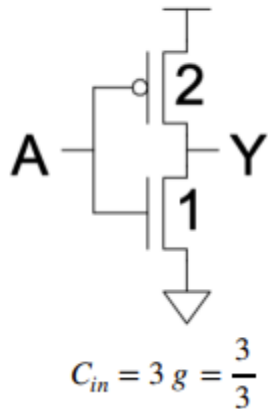
- Efortul logic al unei porți logice ne spune cât de proastă este ieșirea față de un inversor având în vedere că fiecare dintre intrările sale poate conține aceeași capacitanță de intrare ca și un inversor.
- Întârzierea unei porți logice crește cu efortul electric
- Porțile logice complexe au mai mult efort logic și întârziere parazită





# Calcularea efortului logic

- În cazul inversorului capacitanța de intrare este 3 deoarece intrarea A se va duce și pe tranzistorul pMOS și pe tranzistorul nMOS.
- $g=1$  prin definiție și observăm că se respectă faptul că  $tp_{LH} = tp_{HL}$



# Calcularea efortului logic

## Efortul logic pentru porțile elementare

Tipul porții	Numărul de intrări				
	1	2	3	4	$n$
Inversor	1				
NAND		$\frac{4}{3}$	$\frac{5}{3}$	$\frac{5}{3}$	$\frac{n+2}{3}$
NOR		$\frac{5}{3}$	$\frac{7}{3}$	$\frac{9}{3}$	$\frac{2n+1}{3}$
Tristate / mux	2	2	2	2	2
XOR, XNOR		4, 4	6, 12, 6	8, 16, 16, 8	

# Calcularea efortului logic

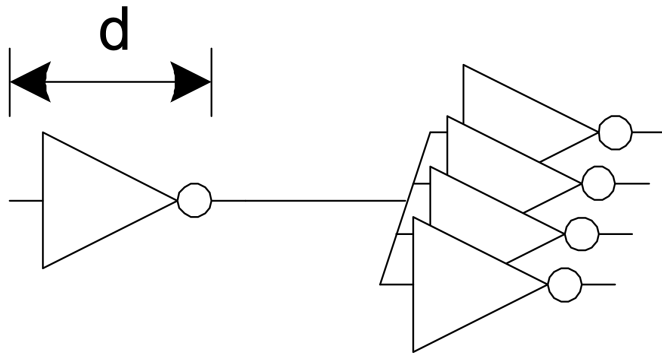
- Întârzierea parazită a porților elementare
  - În multiplii de  $p_{inv}$  ( $\approx 1$ )

Tipul porții	Numărul de intrări				
	1	2	3	4	n
Inversor	1				
NAND		2	3	4	n
NOR		2	3	4	n
Tristate / mux	2	4	6	8	2n
XOR, XNOR		4	6	8	

## Exemplu: Inversorul FO4

- Calculul întârzierii pentru inversorul FO4

$$f = g * h = 1 * 4 = 4$$



$$P_{inv} = 1$$

$$d = f + P_{inv} = 4 + 1 = 5$$

$$d_{(abs)} = d * \tau$$

$\tau$  tipic este 12ps pt tehnologia pe 180nm

$$d_{(abs)} = 5 * 12 = 60ps$$

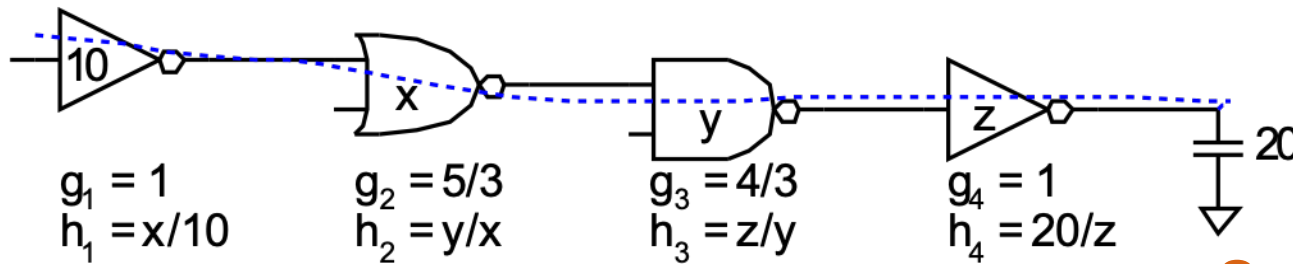
# Concluzii

- Cu cât fan-out-ul este mai mare cu atât întârzierea va crește
- Cu cât avem porți mai complicate (NOR4 vs NOR5) valoarea lui  $g$  va crește deci întârzierea totală va crește

# Efortul logic pentru rețele cu logică pe mai multe stagii

- Efortul logic presupune obținerea celui mai mici întârzieri prin balansarea întârzierii de-a lungul tuturor stagiilor
- Efortul logic de-a lungul unei căi se determină prin înmulțirea eforturilor logice ale tuturor porților logice.
- Efortul logic al căii  $G = \prod g_i$
- Efortul electric al căii  $H = \frac{C_{out}}{C_{in}}$
- Efortul ramurii  $b = \frac{C_{on-path} + C_{off-path}}{C_{on-path}}$
- Efortul căii ramurii  $B = \prod b_i$

# Efortul logic pentru rețele cu logică pe mai multe stagii



*Se poate scrie  
 $F = GH$ ?*

$$G = 1$$

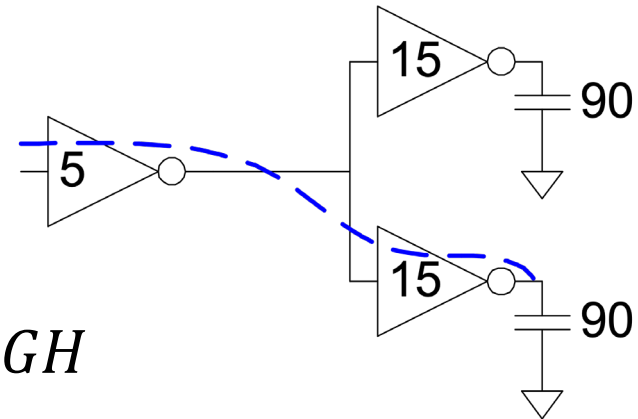
$$H = \frac{90}{5} = 18$$

$$GH = 18$$

$$h_1 = \frac{15+15}{5} = 6$$

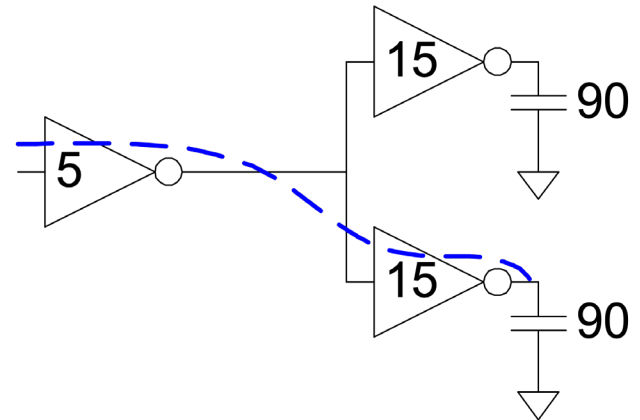
$$h_2 = \frac{90}{16} = 6$$

$$F = g_1 * g_2 * h_1 * h_2 = 36 = 2GH$$



# Efortul ramificației

- Efortul ramurii  $b = \frac{C_{on-path} + C_{off-path}}{C_{on-path}}$
- Efortul căii ramurii  $B = \prod b_i$
- $BH = \frac{C_{out}}{C_{in}} \prod b_i = \prod h_i$
- Efortul căii  $F = GBH$





# Întârzierea multinivel

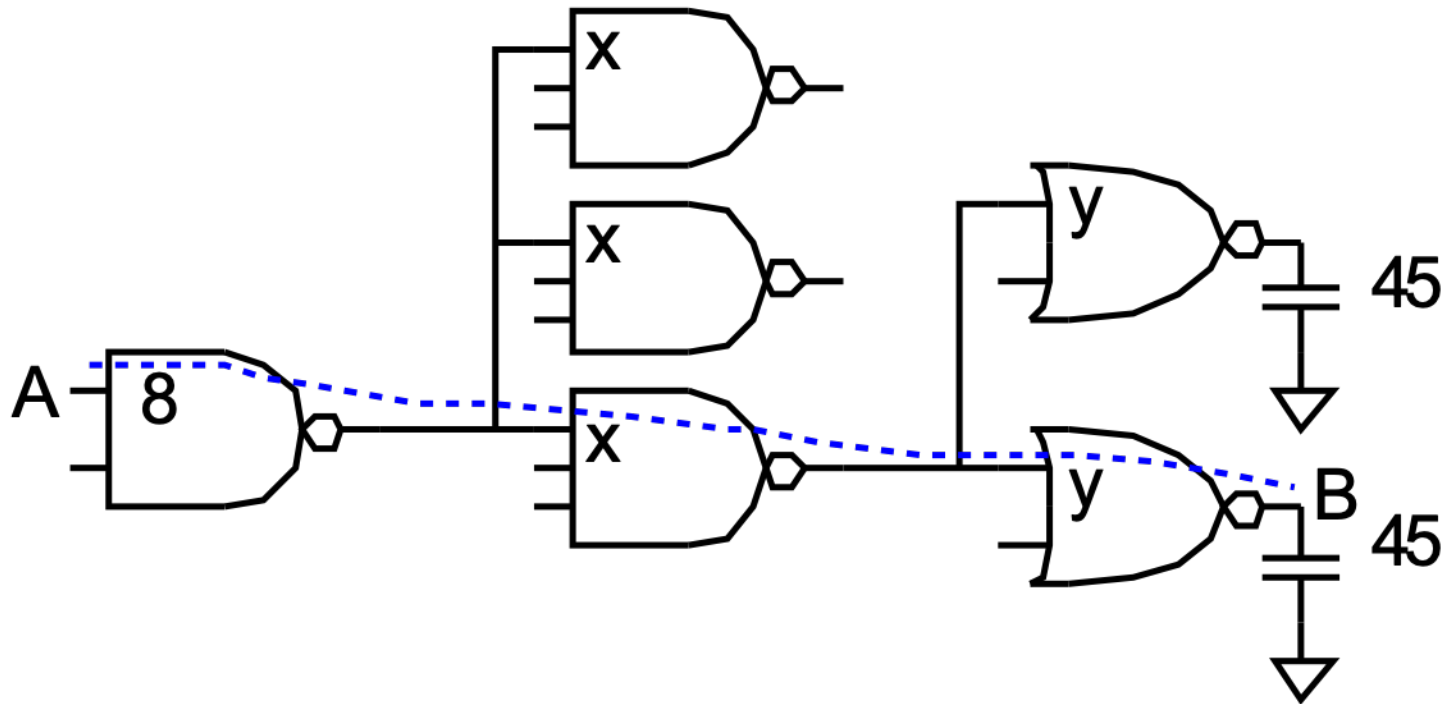
- Întârzierea căii va fi  $D = \sum d_i = D_F + P$  unde
  - $D_F = \sum g_i * h_i$
  - $P = \sum p_i$
- În cazul unei rețele cu N stagii, efortul stagiului pentru fiecare stagiou va fi
  - $\hat{f} = g_i * h_i = \sqrt[N]{F}$

# Proiectarea circuitelor rapide

- Întârzierea minimă ce poate fi obținută de-a lungul unei căi
  - $\hat{D} = N * \sqrt[N]{F} + P$
  - $\hat{h}_i = \frac{\sqrt[N]{F}}{g_i} = \frac{C_{out-i(total)}}{C_{in-i}}$
- Capacitanța văzută la intrarea în fiecare stadiu
  - $C_{in-i} = \frac{C_{out-i(total)}}{\hat{h}_i}$

# Exemplu

Selectăm dimensiunile  $x$  și  $y$  ale porților pentru întârzierea minimă de la  $A$  la  $B$



# Exemplu

Efortul logic:

$$G = \frac{4}{3} * \frac{5}{3} * \frac{5}{3} = \frac{100}{27}$$

Efortul electric:

$$H = \frac{45}{8}$$

Efortul de ramură:

$$B = 3 * 2 = 6$$

Efortul căii:

$$F = GBH = 125$$

Efortul stagiului cel mai bun:

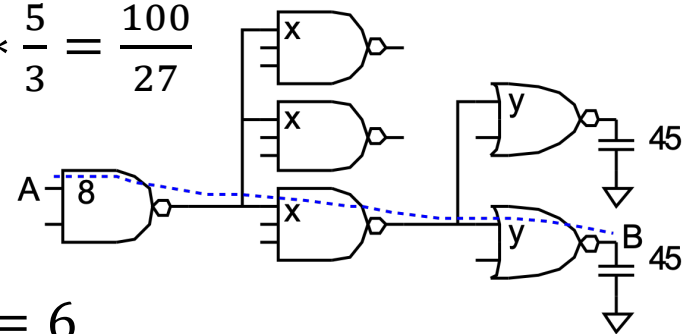
$$\hat{f} = \sqrt[3]{F} = 5$$

Întârzierea parazită

$$P = 2 + 3 + 2 = 7$$

Întârzierea

$$D = 3 * 5 + 7 = 22 = 4.4FO_4$$



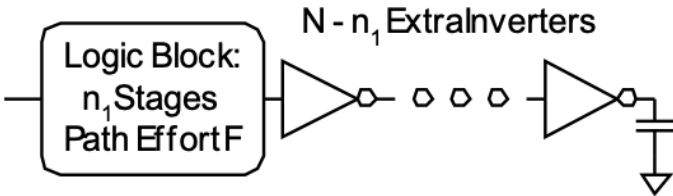
# Derivare

Considerăm inversoare la finalul căii

- Câte ne oferă întârzierea minimă?

$$D = N^{\sqrt[N]{F}} + \sum_{i=1}^{n_1} p_i + (N - n_1)p_{inv}$$

Logic Block:  
 $n_1$  Stages  
 Path Effort  $F$



$$\frac{\partial D}{\partial N} = -\sqrt[N]{F} * \ln(\sqrt[N]{F}) + \sqrt[N]{F} + p_{inv} = 0$$

- Definirea efortului optim pe stagiu  $\rho = \sqrt[N]{F}$

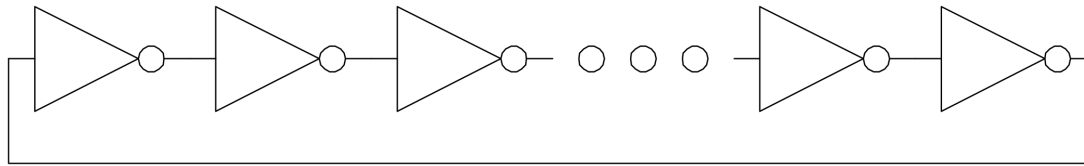
$p_{inv} + \rho(1 - \ln(\rho)) = 0$  nu are o soluție închisă

- Neglijând partea parazită ( $p_{inv} = 0$ ), găsim  $\rho = 2.718(e)$

Pentru  $p_{inv} = 1$ , găsim (numeric)  $\rho = 3.59$

# Exemplu: Oscilator tip inel

Estimăm frecvența unui oscilator de tip inel cu N stagii



Efortul logic:  $g = 1$

Efortul electric:  $h = 1$

Întârzierea parazită:  $p = 1$

Întârzierea stagiului:  $d = 2$

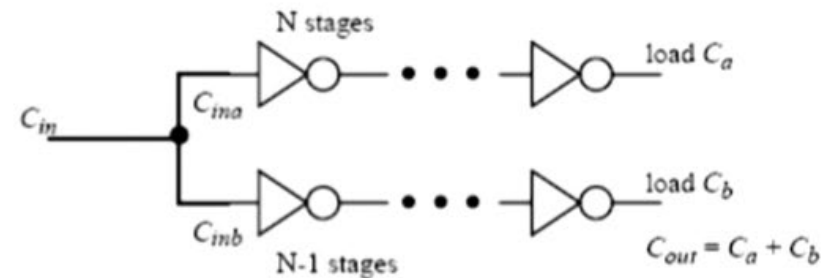
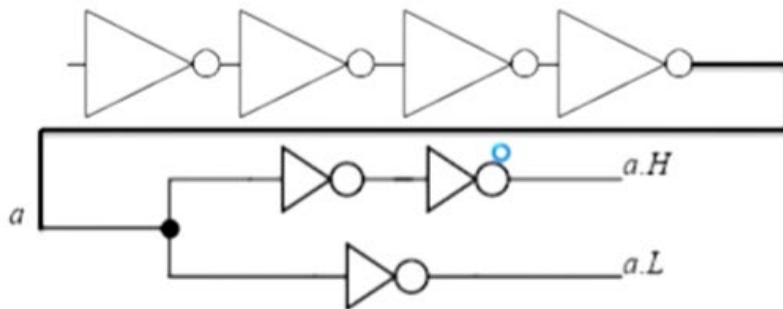
Frecvența: 
$$f_{osc} = \frac{1}{2 * N * d} = \frac{1}{4N}$$

Un oscilator inel cu 31 stagii  
în tehnologie 0.6  $\mu\text{m}$  are o  
frecvență de  $\sim 200$  MHz

# Proiectarea FORK

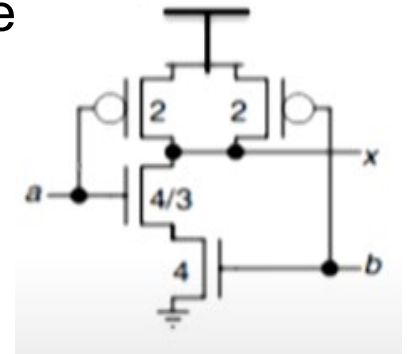
- Semnalele sunt deviate de la calea originală către o cale cu impedanță scăzută
- Ambele semnale primare (provenite dintr-o ieșire) ajung simultan  $\Rightarrow C_{out} = C_a + C_b$
- Circuitul FORK poate fi proiectat utilizând efortul logic !!!!!

$$d_{fork} = (N - 1)^{N-1} \sqrt{\left(\frac{C_b}{C_{inb}}\right)} + (N - 1) * p = N * \sqrt[2]{\left(\frac{C_a}{C_{ina}}\right)} + N * p$$



# Porți asimetrice

- O intrare este favorizată față de restul
- Plasăm intrarea critică cât mai aproape de ieșire
- Selectăm dimensiunile PUN și PDN ca să se potrivească cu dimensiunile inversorului
- Întârzierea porților asimetrice bazate pe circuite logice poate fi calculată





# Recapitulare

Termenul	Stagiul	Calea
Numărul de stagii	1	$N$
Efortul logic	$g$	$G = \prod g_i$
Efortul electric	$h = \frac{C_{\text{out}}}{C_{\text{in}}}$	$H = \frac{C_{\text{out-path}}}{C_{\text{in-path}}}$
Efortul de ramură	$b = \frac{C_{\text{on-path}} + C_{\text{off-path}}}{C_{\text{on-path}}}$	$B = \prod b_i$
Efort	$f = gh$	$F = GBH$
Întârzierea efortului	$f$	$D_F = \sum f_i$
Întârzierea parazită	$p$	$P = \sum p_i$
Întârzierea	$d = f + p$	$D = \sum d_i = D_F + P$

# Metoda efortului logic

- 1) Calculăm efortul căii:  $F = GBH$
- 2) Estimarea numărului de stagii optim:  $N = \log_4 F$

Calea cu N stagii

- 1) Estimarea întârzierii optime:  $D = N * \sqrt[N]{F} + P$
- 5) Determinarea efortului optim pe stagi:  $\hat{f} = \sqrt[N]{F}$
- 6) Determinarea dimensiunii porții:  $C_{in_i} = \frac{g_i * C_{out_i}}{\hat{f}}$

# Limitele Efortului Logic

Problema cu oul și găina

Este necesară o cale pentru determinarea lui G

Dar, nu cunoaștem numărul de stagii fără G

Modelul întârzierii este simplist

Se neglijează efectele frontului pozitiv ale intrărilor.

Interconectarea

Se vor folosi fire în proiectare

Doar viteză maximă

Aria nu este minimă/Puterea ?

# Concluzii

Efortul logic este util pentru a gândi întârzierile în circuite

- Efortul logic numeric caracterizează porțile
- NAND-urile sunt mai rapide decât NORs în CMOS
- Căile sunt mai rapide când întârzierile efortului sunt  $\sim 4$
- Întârzierea căii este slab senzitivă la dimensiunile stagiilor
- Utilizarea de stagii puține nu înseamnă căi mai rapide
- Întârzierea căii este  $\log_4 F_{FO4}$  întârzierile inversorului
- Inversoarele și NAND2 cele mai bune pentru a conduce capacități mari
- Este necesară experiență !!!!!!!