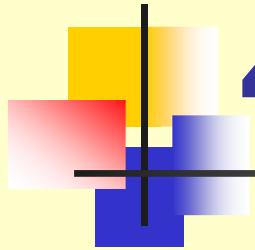


SINTEZA CLC CU LSI.

HAZARDUL COMBINAȚIONAL

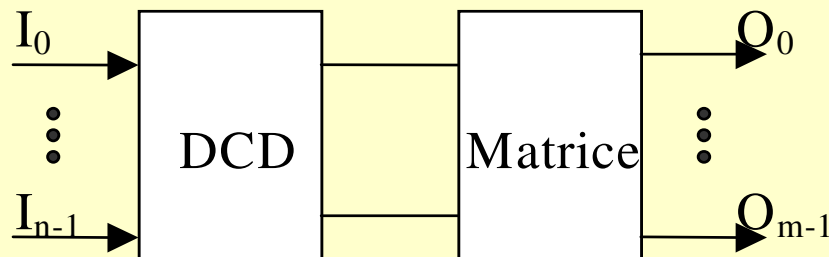


4.3.3. Sinteza CLC cu LSI

- Circuitele integrate de tip LSI (large scale integration) au peste 500 de tranzistoare integrate pe capsulă
- Există mai multe tipuri de astfel de circuite
- Descriem sinteza CLC cu:
 - memorii ROM - Read Only Memory
 - PLA - Programmable Logic Array (au și varianta PAL)

4.3.3.1. Sinteza CLC cu memorii ROM

- **Memoria ROM** - memorează cuvinte binare
 - fixă (permanentă)
 - nevolatilă
 - **conținutul nu se modifică în timpul funcționării**
 - structura stabilită în procesul de fabricație sau stabilită de către utilizator prin **programare**
 - tipuri: PROM, EPROM, EEPROM



4.3.3.1. Sinteza CLC cu memorii ROM

■ Memoria ROM

- formată din 2 niveluri de porți logice:
 - nivelul ȘI (un decodificator)
 - nivelul SAU-NU (matricea de memorie)
- DCD primește codurile de intrare în binar (numărul intrărilor = n) și activează pentru fiecare cod o singură ieșire din cele 2^n posibile
- ieșirile DCD se conectează sau nu se conectează la porțile de tip SAU-NU și astfel se memorează un 0 sau un 1 logic

4.3.3.1. Sinteza CLC cu memorii ROM

- Intrările memoriei ROM
 - se numesc **adrese**
 - reprezintă codurile în binar ale numerelor asociate fiecărui cuvânt de memorie
- Ieșirile memoriei ROM
 - pun la dispoziție datele memorate în cuvintele de memorie
 - sunt “three-state” sau “open collector” pentru a permite legarea în paralel cu ieșirile altor memorii
- Există intrare de Enable care poate inhiba funcționarea memoriei

4.3.3.1. Sinteza CLC cu memorii ROM

■ Notatii:

- n = numărul de biți ai vectorului de adrese (intrări)
- c = numărul de cuvinte memorate în ROM (trebuie să fie putere a lui 2) $\rightarrow c = 2^n$
- b = numărul de biți din fiecare cuvânt de memorie

■ Modul de organizare al memoriei este specificat de produsul $c \times b$

■ Capacitatea memoriei C exprimă **numărul total de biți memorați**: $C = 2^n \times b$

■ Unitatea de măsură pentru capacitate:

- kilobitul $1Kb = 2^{10} = 1024 \text{ biți}$

4.3.3.1. Sinteza CLC cu memorii ROM

■ Aplicații:

- Memorarea instrucțiunilor și datelor în sisteme de calcul și sisteme secvențiale
- Transformări de adresă și înmagazinarea instrucțiunilor în microprogramare
- Conversii de cod
- Generatoare de caractere
- Generare de secvențe de impulsuri
- Implementarea CLC cu un număr mare de variabile de intrare și ieșire

4.3.3.1. Sinteza CLC cu memorii ROM

- Implementarea CLC cu memorii ROM:
 - Se bazează pe structura memoriei
 - La nivelul DCD se decodifică toți termenii canonici formați din variabilele de intrare
 - La nivelul matricei de memorie se adună toți termenii din expresia oricărei funcții și rezultă funcția de ieșire
 - Lista de cuvinte care se memorează reprezintă chiar **tabelul de adevăr al CLC**
 - **Nu mai este necesară minimizarea** (sunt memorați toți termenii canonici și sunt incluse toate posibilitățile de apariție a acestora în funcția de ieșire)
 - Pentru folosire eficientă se folosesc codificarea și MUX

4.3.3.1. Sinteza CLC cu memorii ROM

- Etapele de implementare a CLC cu memorii ROM
 - Se stabilește dimensiunea memoriei necesare pentru aplicație
 - Se aleg tipurile de memorii cu dimensiuni identice sau apropiate cu cele stabilite anterior
 - Se reduce dimensiunea memoriei, dacă este posibil, prin codificarea intrărilor și ieșirilor sau multiplexarea intrărilor
 - Dacă nu există memorii potrivite se fac transformări de dimensiuni - modificarea numărului de cuvinte sau al numărului de biți pe cuvânt
 - Se stabilește tabelul de adevăr al memoriei ROM

4.3.3.1. Sinteza CLC cu memorii ROM

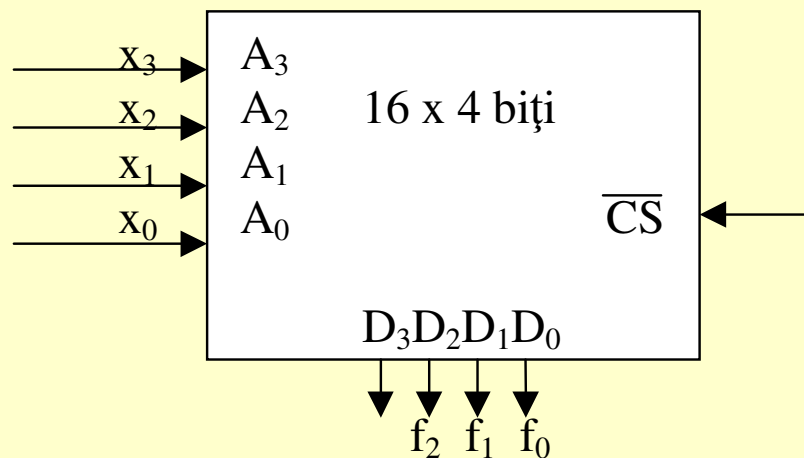
- **Exemplu:** Să se implementeze funcțiile:

$$f_0(x_3, x_2, x_1, x_0) = x_3 \bar{x}_2 \bar{x}_1 \bar{x}_0$$

$$f_1(x_3, x_2, x_1, x_0) = x_2 \bar{x}_1$$

$$f_2(x_3, x_2, x_1, x_0) = \bar{x}_3 \bar{x}_2 \bar{x}_1 x_0 + x_3 \bar{x}_2 \bar{x}_0 + x_3 x_2 x_0 + x_2 x_1$$

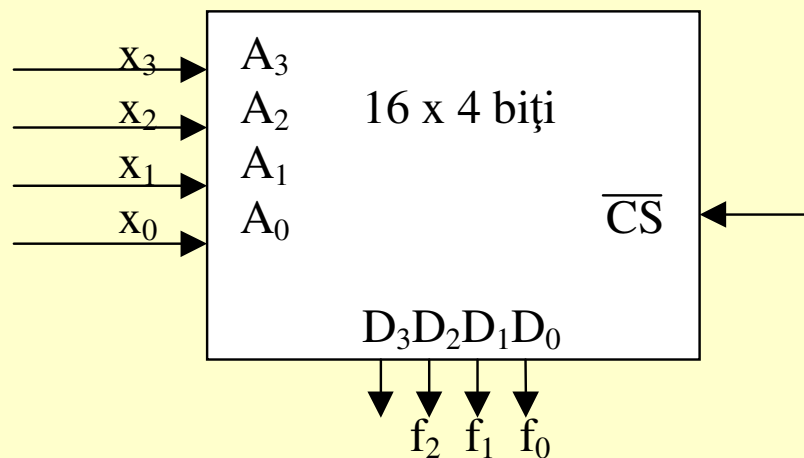
- Vom folosi o memorie de tipul:



4.3.3.1. Sinteza CLC cu memorii ROM

■ Exemplu

- Notatii: $A_3 - A_0$ intrările de adrese; $D_3 - D_0$ ieșirile de date; CS (chip select) intrarea de Enable
- Dimensiunea: $2^4 \times 4 = 16 \times 4$ biți (avem 4 intrări de adresă $\Rightarrow 2^4$ cuvinte și avem 4 biți într-un cuvânt)

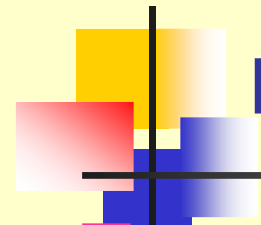


4.3.3.1. Sinteza CLC cu memorii ROM

■ Exemplu

- Tabelul de adevăr pentru funcțiile f_0 , f_1 , f_2 , care se vor obține pe ieșirile de date D_0 , D_1 , D_2 , se va înscrie în memorie
- Ieșirea D_3 nu este utilizată (avem de implementat doar 3 funcții)
- Variabilele de intrare se aplică pe intrările de adrese ale memoriei

4.3.3.1. Sinteza CLC cu memorii ROM



■ Exemplu

A ₃	A ₂	A ₁	A ₀	D ₃	D ₂ = f ₂	D ₁ = f ₁	D ₀ = f ₀
0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	1	0
0	1	0	1	0	0	1	0
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	0
1	0	0	0	0	1	0	0
1	0	0	1	0	0	0	0
1	0	1	0	0	1	0	1
1	0	1	1	0	0	0	0
1	1	0	0	0	0	1	0
1	1	0	1	0	1	1	0
1	1	1	0	0	1	0	0
1	1	1	1	0	1	0	0

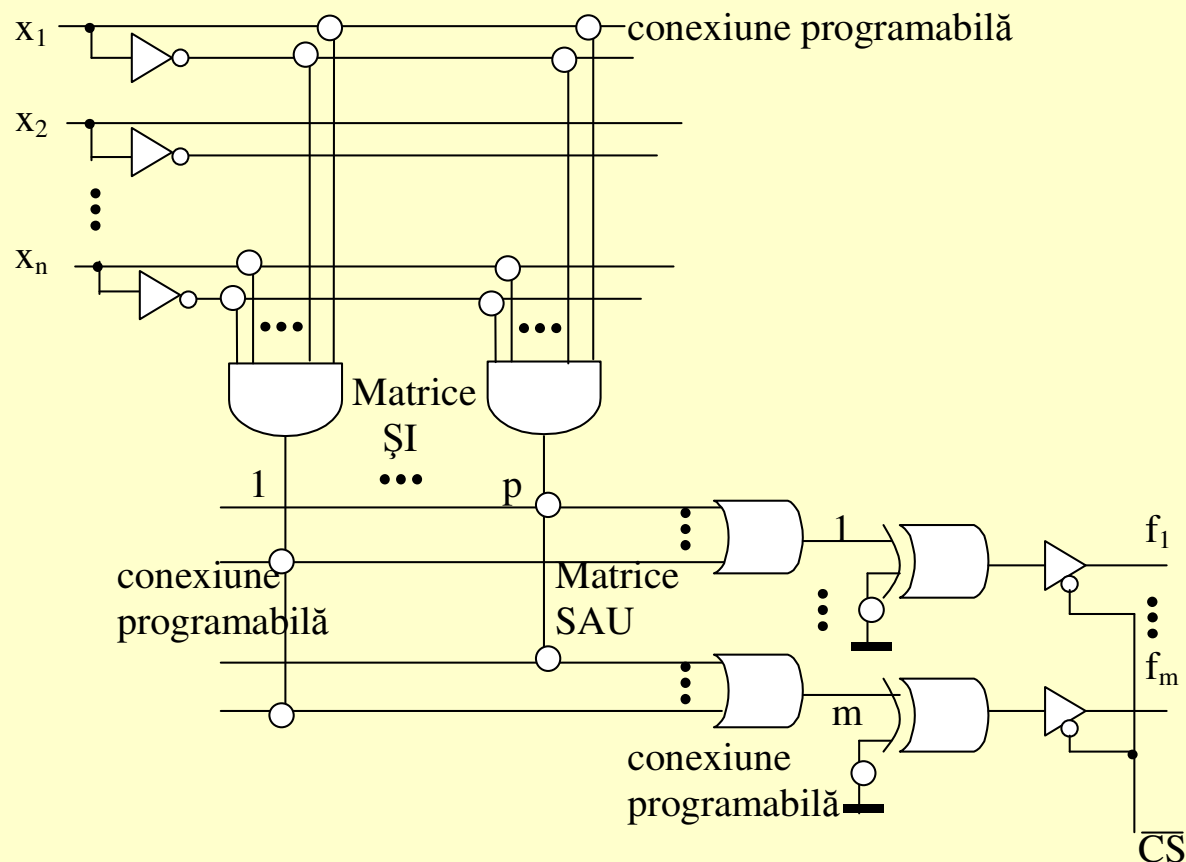


4.3.3.2. Sinteza CLC cu PLA

- **PLA** = Programmable Logic Array
- **Definiție** - PLA este un CLC cu 2 nivele de logică programabilă, o matrice de porți ȘI și o matrice de porți SAU
 - Varianta PAL are matricea de tip SAU neprogramabilă, dar are intrările bidirecționale
- Cele 2 matrici sunt programabile în procesul de fabricație sau de către utilizator, conform aplicației concrete
- Implementarea CLC se face folosind **varianta minimizată a funcțiilor** (termenii elementari)

4.3.3.2. Sinteza CLC cu PLA

- Structura PLA cu “n” intrări, “m” ieșiri și “p” termeni elementari realizabili



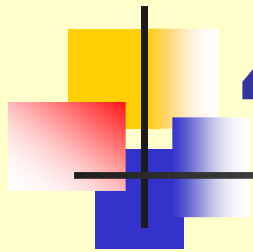


4.3.3.2. Sinteza CLC cu PLA

- Avantajele implementării CLC cu PLA
 - Posibilitatea programării matricei ȘI
 - Posibilitatea complementării ieșirilor (programare individuală ca să fie active pe 0 sau pe 1 logic)
- Aplicații ale PLA
 - Microprogramare
 - Conversii de cod
 - Generatoare de caractere
 - Realizare de tabele de funcții
 - Implementarea automatelor secvențiale

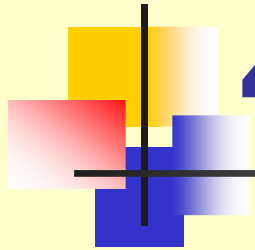
4.3.3.3. Sinteza CLC cu VLSI

- Circuitele **VLSI** (Very Large Scale Integration) - grad extins de integrare
- Capitol special de studiu pentru proiectarea lor



4.4. Hazardul combinațional

- **Definiție** - reprezintă o comportare greșită a CLC, pentru scurt timp
- **Cauze:**
 - Întârzieri produse de circuitele logice
 - Întârzieri produse de firele de legătură
- **Efect** - starea ieșirii circuitului, în momentul modificării variabilelor de intrare, nu coincide cu valoarea funcției corespunzătoare valorilor intrărilor în momentul considerat



4.4. Hazardul combinațional

■ Hazard:

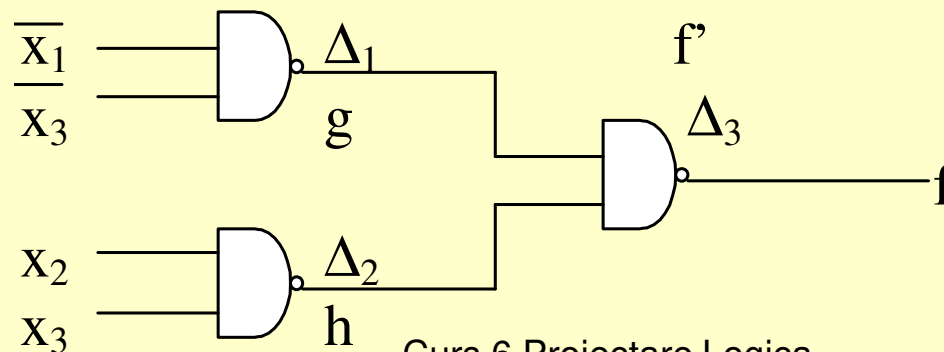
- De funcție - dacă la un moment dat se modifică mai mult decât o singură variabilă de intrare - greu de realizat proiectarea CLC
- Static - la un moment dat se modifică o singură variabilă de intrare - condiție asigurată prin notarea diagramei Karnaugh în cod Gray - se poate elimina

4.4. Hazardul combinațional

■ Exemplu - hazard static

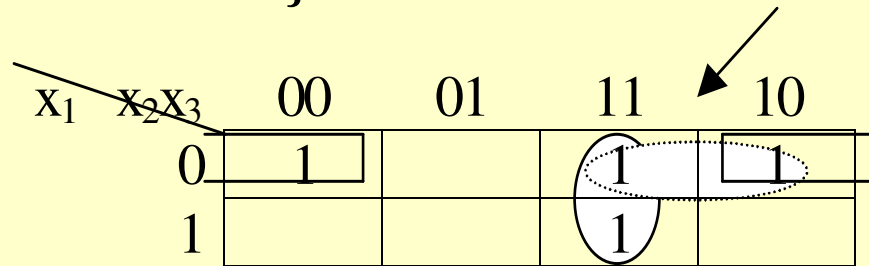
- Funcție de 3 variabile, implementată doar cu porți de tip ȘI-NU pentru a avea aceleași tipuri de porți logice

- $f(x_1, x_2, x_3) = \overline{x_1} \cdot \overline{x_3} + x_2 \cdot x_3 = \overline{\overline{x_1} \cdot \overline{x_3} \cdot x_2 \cdot x_3}$
- $\Delta_1, \Delta_2, \Delta_3$ sunt întârzierile diferite ale porților ȘI-NU
 - Întârziere - timpul dintre momentul aplicării intrărilor și momentul apariției ieșirii



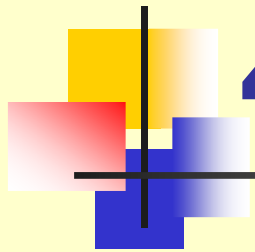
4.4. Hazardul combinațional

- DK pentru funcția f este:



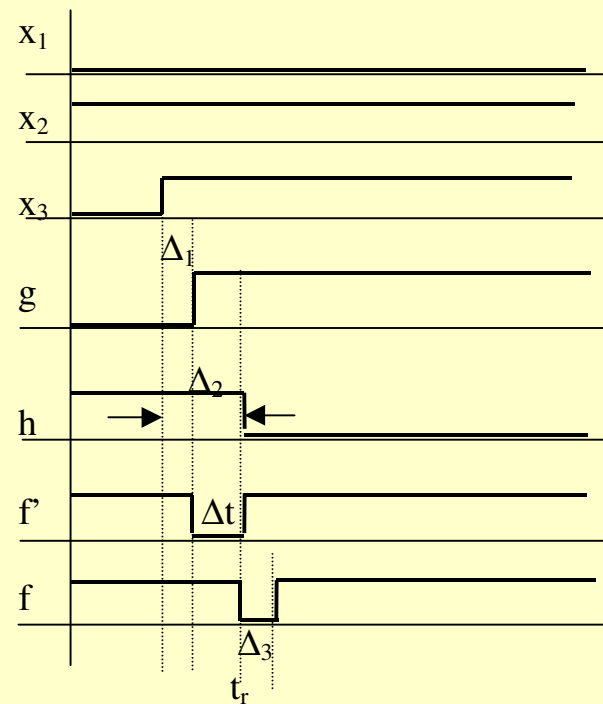
$x_1 \backslash x_2 x_3$	00	01	11	10
0	1		1	1
1			1	

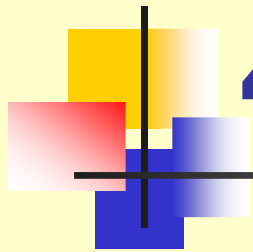
- Întârzieri Δ diferite \Rightarrow hazard static la modificarea intrărilor x_1, x_2, x_3 , din starea 010 în starea 011 sau invers
- După un timp de reacție t_r , datorat întârzierilor, apare la ieșire un impuls negativ $\Delta t = \Delta_2 - \Delta_1$ pe durata căruia ieșirea ia o valoare incorectă



4.4. Hazardul combinațional

- Formele de undă pentru intrări, funcții intermediare și ieșire



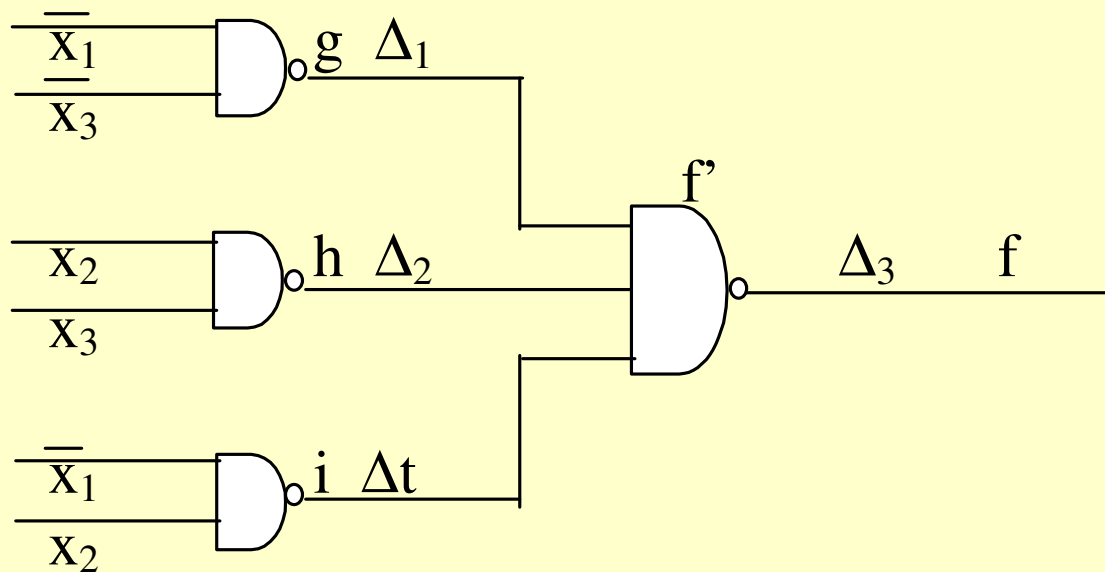


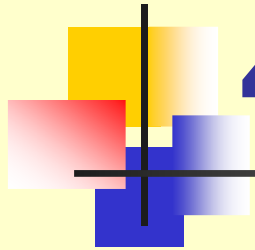
4.4. Hazardul combinațional

- Eliminarea hazardului static - în proiectare, la realizarea funcției se iau în considerare și unii termeni redundanți din DK astfel ca oricare două valori de 1 aflate în căsuțe adiacente să fie incluse cel puțin într-o grupare luată în considerare când se face sinteza
- În exemplul dat se introduce în plus termenul $\overline{x_1} \cdot x_2$
- $f = \overline{x_1} \cdot \overline{x_3} + x_2 \cdot x_3 + \overline{x_1} \cdot x_2$

4.4. Hazardul combinațional

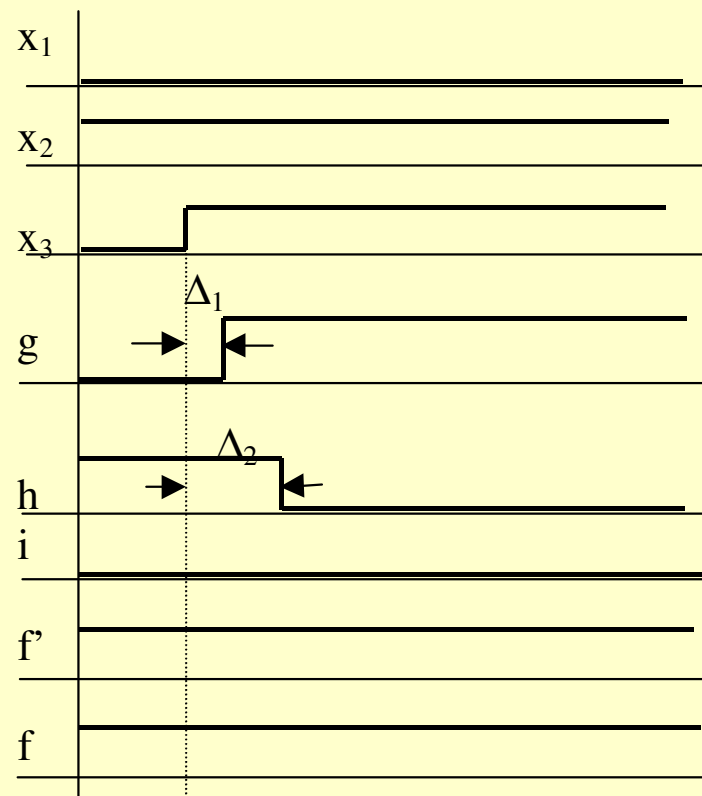
- Schema corectată:

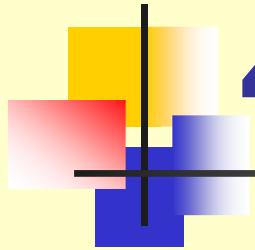




4.4. Hazardul combinațional

- Formele de undă fără hazard combinațional static:





4.4. Hazardul combinațional

- Eliminarea **sigură** a hazardului combinațional static:
 - se ia în considerare ieșirea circuitului numai după un interval de timp mai mare decât întârzierea maximă din circuit!