Proiectarea subsistemelor VLSI

Problema generală:
- gestiunea unei celule de bază care poate implementa funcțiile prevăzute ale sistemului

Celula de bază dispune:
- un set de criterii/reguli de interconectare

Geometria celulei și criteriile de interconectare dictează proiectarea circuitului de comandă care influențează arie de punctuare constituentă din celule de bază

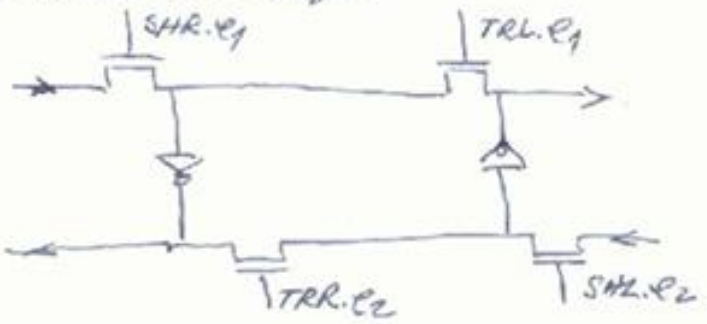
După completarea circuitelor de comandă rezultă un subsistem cu modul funcțional și o interfață bine definită privind:
- specificațiile funcționale;
- " " geometrice
- " " de I/E - date
- " " de comandă și sincronizare.

Setul de circuite de sincronizare are 3 roluri:
- rutarea de date,
- testarea de date,
- rutarea de comandă,

Exemplu: Proiectarea unei memorii de tip Stivă LIFO

Celula de bază este capabilă să realizeze
operațiile: Push - Pop - Nop, în concordanță
cu cele trei faze și al cărui cod de
operație.

Celula de bază



Semnalul de sus se fixează pe durata lui φ_1,
iar semnalul de jos - pe durata lui φ_2
Fluxul de comenzi este vertical,
pe liniile de metalizare, iar fluxul de
date este orizontal pe linii de difuzie

Deplasare dreapta : POSH

φ_1 : SHR = 1 și φ_2 : TRR = 1

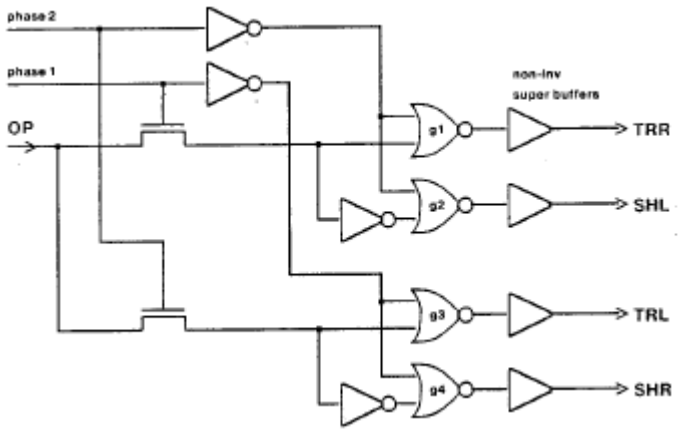
Deplasare stânga : PSH

φ_2 : SHL = 1 și φ_1 : TRL = 1

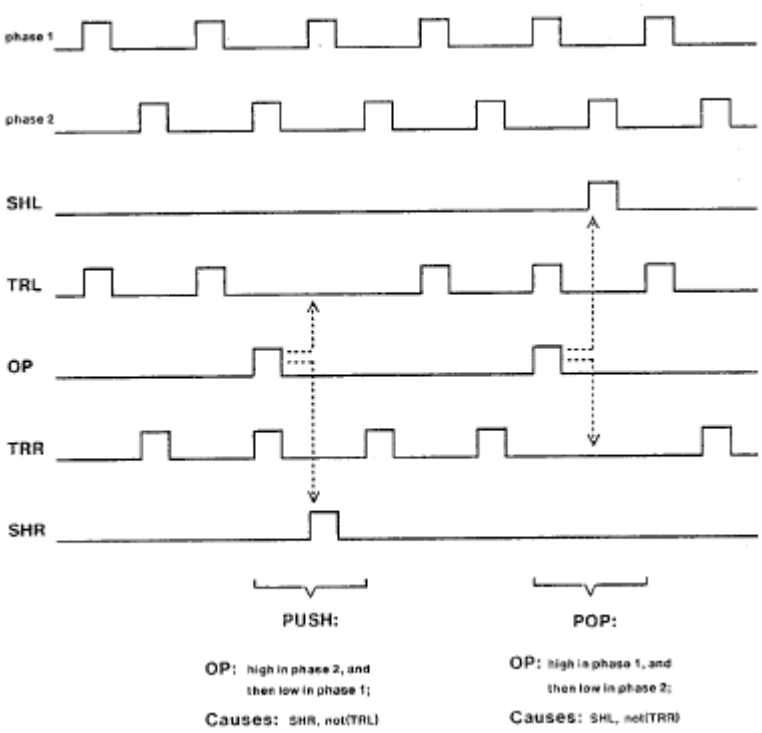
Memorare : NOP

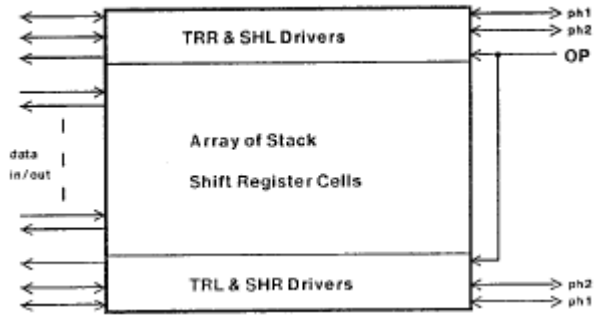
φ_1 : TRL = 1 și φ_2 : TRR = 1

Generarea semnalelor de comandă :
Schema este relativ simplă și poate fi generată prin respectiv.
Se folosește un singur cod de operație de un bit, aplicat fie lui φ_1 (op=1), fie lui φ_2 (op=2).



POSH : op = 1, lui φ_2
op = 0, lui φ_1


phase 1
phase 2
SHL
TRL
OP
TRR
SHR
PUSH:
POP:
OP: high in phase 2, and then low in phase 1;
Causes: SHR, not(TRL)
OP: high in phase 1, and then low in phase 2;
Causes: SHL, not(TRR)

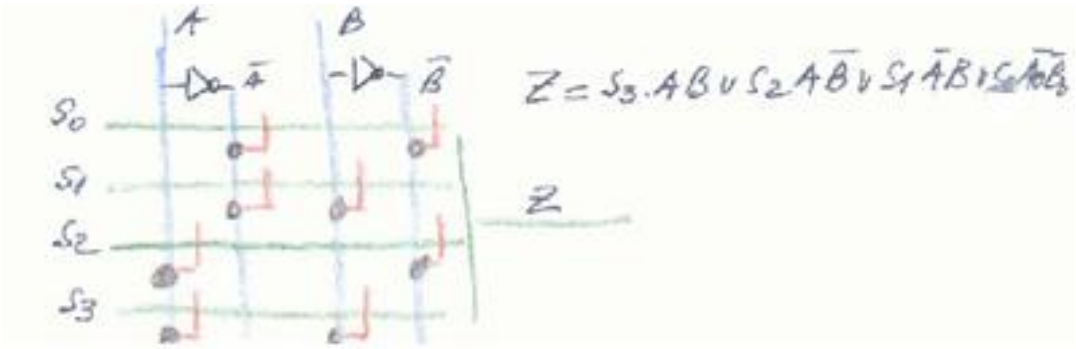

TRR & SHL Drivers
ph1
ph2
OP
Array of Stack
Shift Register Cells
data in/out
TRL & SHR Drivers
ph2
ph1

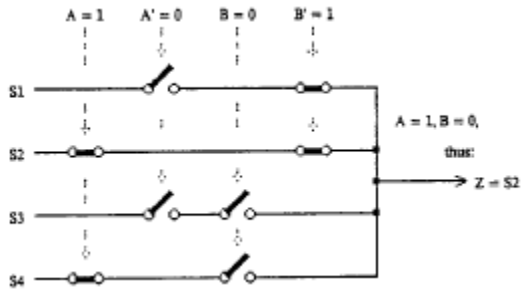
Proiectarea logicii combinationale

Sunt trei categorii de probleme

1. Se cere o cantitate minima de logica
   In acest caz vom avea circuite formate dintr-o singura celula, care poate fi multiplicata
   Se folosesc procedee proiectarii logice traditionale cu circuite NAND, NOR
   Pana recent, fara a mai folosi procedee formale de minimizare
   Utilizarea portilor statice nu este recomandata deoarece
   - nu conduce la forme regulate
   - nu asigura arie minima
   - putere consumata mare.
   - nu realizeaza maximizarea numarului de functii logice pe acelasi circuit

Solutie: Retele formate din tranzistoare de trecere





2. A doua doză de problemă.

Implementarea unei funcții logice oarecare pentru care trebuie să se execute metode de structurare topologică.

Fie circuitul TALY cu n intrări și m ieșiri

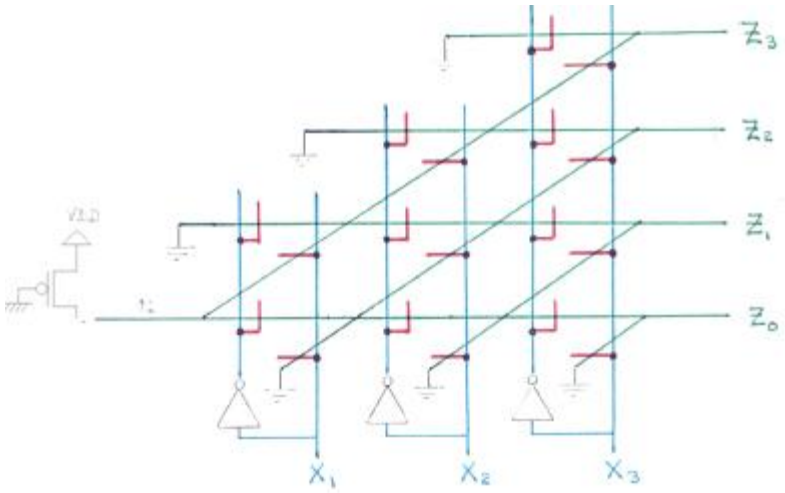
(n = 3)

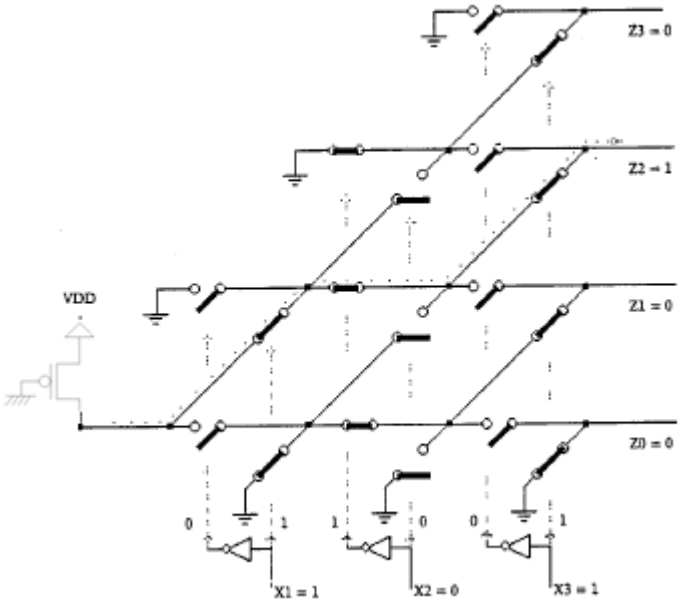
$z_0 = \bar{x}_3 \bar{x}_2 \bar{x}_1$

$z_1 = \bar{x}_3 \bar{x}_2 x_1 \vee \bar{x}_3 x_2 \bar{x}_1 \vee x_3 \bar{x}_2 \bar{x}_1$

$z_2 = \bar{x}_3 x_2 x_1 \vee x_3 \bar{x}_2 x_1 \vee x_3 x_2 \bar{x}_1$

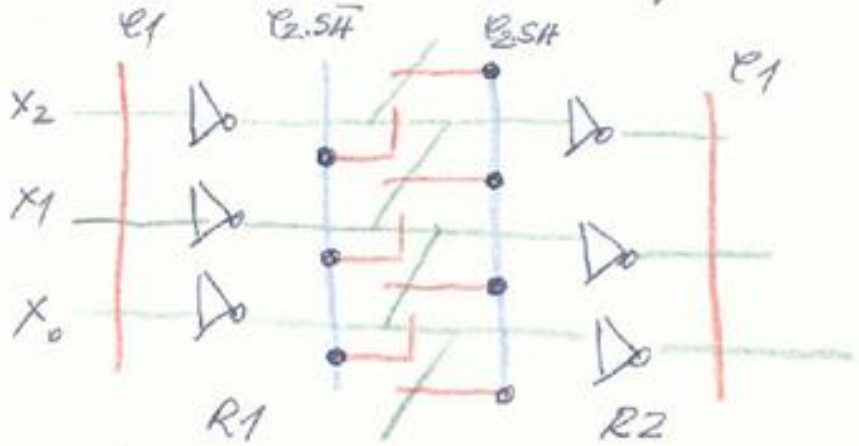
$z_3 = x_3 x_2 x_1$


Z3
Z2
Z1
Z0
X1
X2
X3


VDD
Z3 = 0
Z2 = 1
Z1 = 0
Z0 = 0
X1 = 1
X2 = 0
X3 = 1

Implementarea circuitului la structuri regulate sub aspect topologic.

Realizarea unui circuit de deplasare



φ_1 | $R1 \leftarrow X$

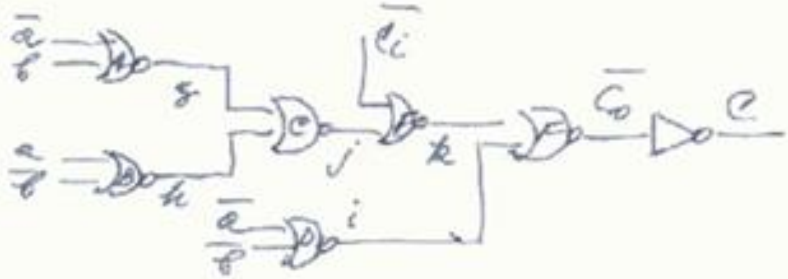
φ_2 | $R2 \leftarrow (R1 \,!\, SH(R1)) \times (\overline{SH}, SH)$

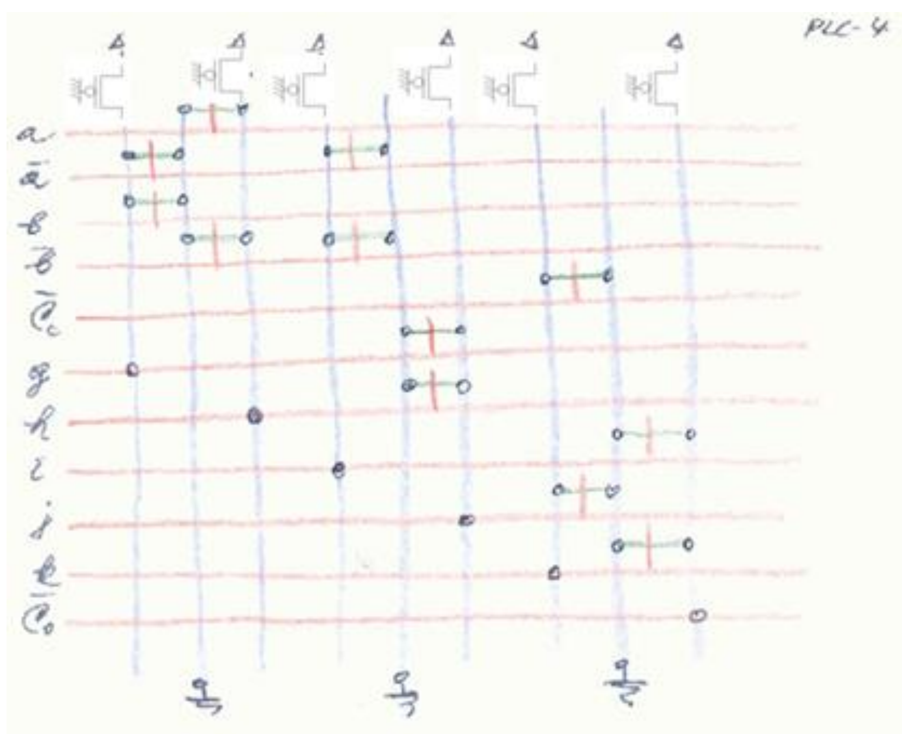
Pentru a manipula mai mult de 3 ieșiri se vor introduce buffere superputere. Schema este generată direct din logica cu relee.

Rețele Weinberger

Fie funcția

$C_o = ab \vee C_i(a\bar{b} \vee \bar{a}b)$ implementată cu circuite NOR





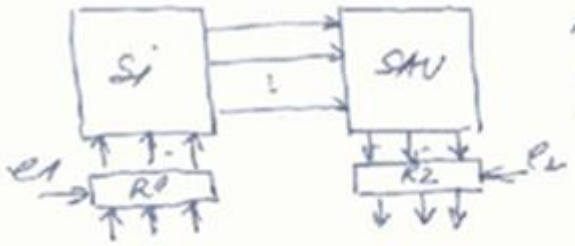
3. Retele logice programabile

O functie logica complexa trebuie implementata fara a recurge la o schema structurata regulata. De exemplu logica retelei negative

Retelele programabile permit aplicarea functiilor logice nestructurate si structurii regulate. FL. pot fi modificate ulterior fara a fi necesara modificarea macheta si montarea PCA

Utilizarea memoriilor pentru stocarea tabelelor de adevar nu este convenabila deoarece ocupa un volum mare.

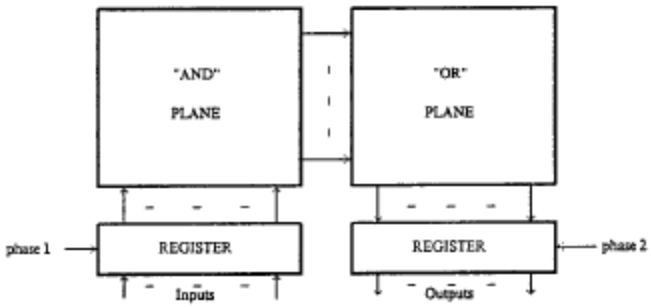
Schema bloc a unui PLA

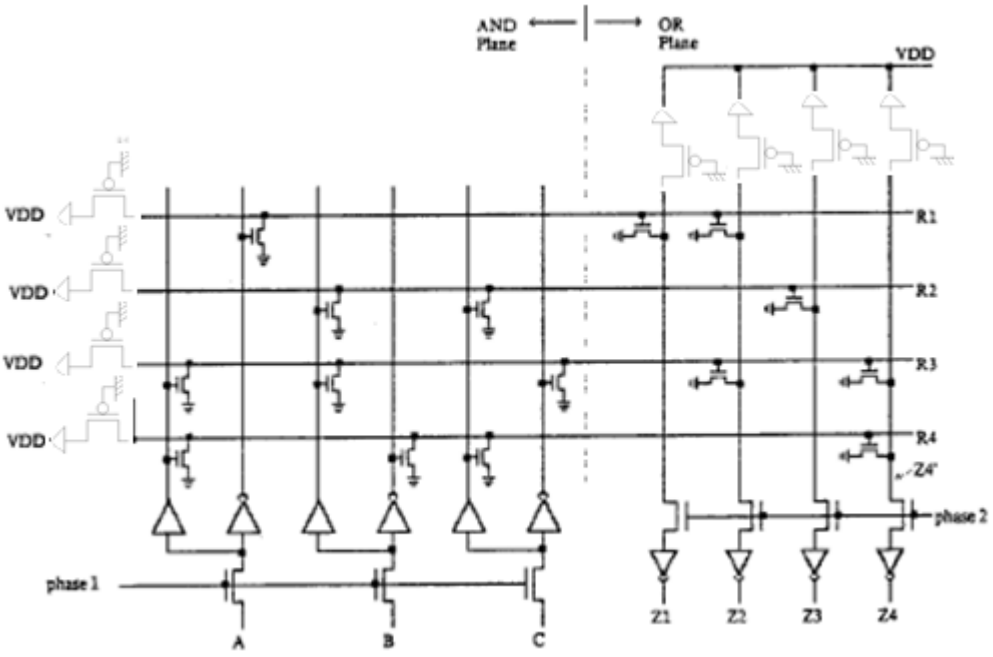


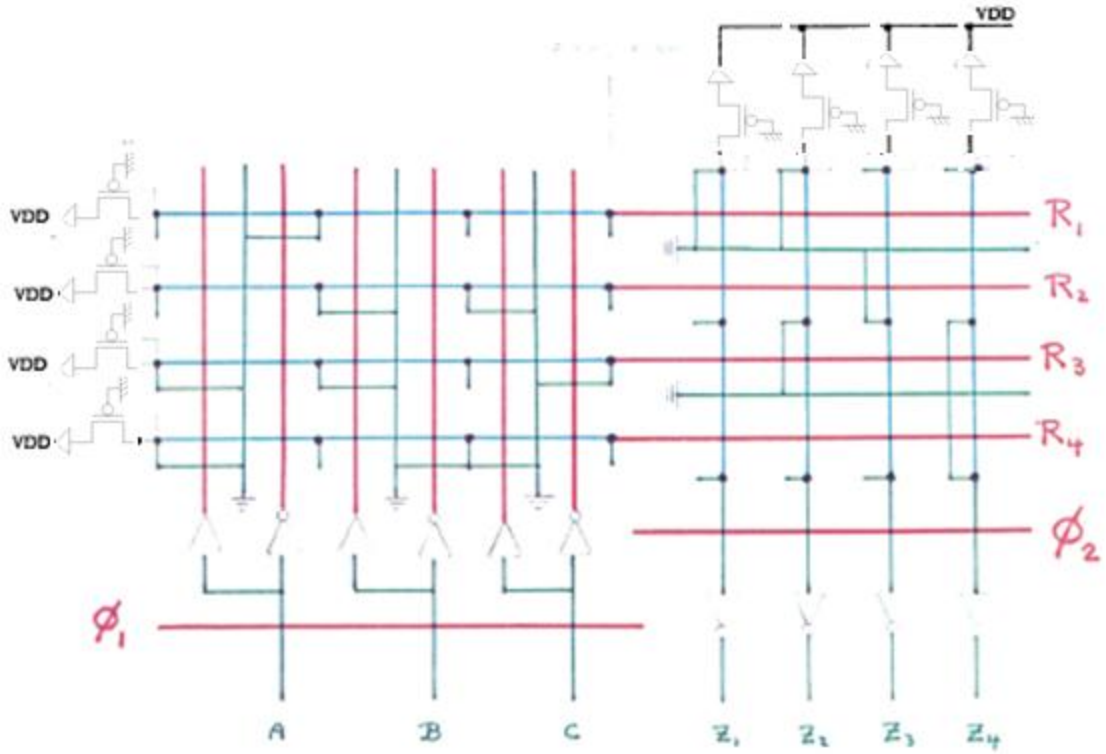
Anzii Si-SAU sunt realizati cu circuite NOR.

$Z1 = A$
$Z2 = A \vee \bar{A}\bar{B}C$
$Z3 = \bar{B}\bar{C}$
$Z4 = \bar{A}\bar{B}\bar{C} \vee \bar{A}B\bar{C}$

implicanti primi:
$A, \bar{A}\bar{B}C, \bar{B}\bar{C}, \bar{A}B\bar{C}$


"AND"
PLANE
"OR"
PLANE
phase 1
REGISTER
REGISTER
phase 2
Inputs
Outputs


AND
Plane
OR
Plane
VDD
VDD
VDD
VDD
VDD
R1
R2
R3
R4
Z4'
phase 2
phase 1
A
B
C
Z1
Z2
Z3
Z4



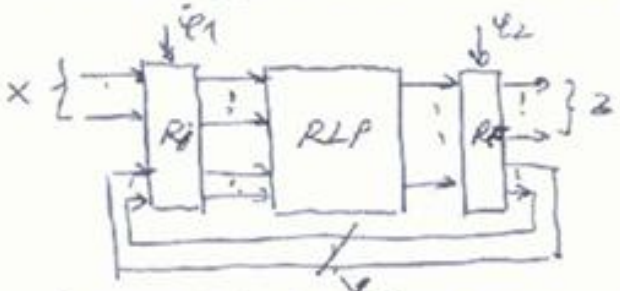
O rețea logică programabilă reactă constă dintr-o linie de elemente de circuit numite porți care produce ieșiri în forma canonică a funcției

Dimensiunile în formă generală depind de următorii parametri

- numărul de intrări,
- numărul termenilor de tip produs,
- numărul de ieșiri
- mărimea de lungime λ.

AUTOMATE FINITE (AF)

AF se realizează pe baza unei rețele logice programabile într-un registru și asigurând o reacție de la ieșire spre intrare



Semnalele de intrare X împreună cu semnalele de stare Y sunt stocate în registrul de intrare Ri, pe durata lui ℓ_1. Semnalele se propagă prin RPL până la intrarea lui RE, care le stochează pe durata lui ℓ_2

Trebuie avută în vedere, la calculul perioadei ceasului întârzierea în RLP

Nr. biților de reacție determină un număr de stări

Automatul este sincron

$$\left.\begin{aligned} Y[(k+1)T] &= f[Y(kT), X(kT)] \\ Z[kT] &= g[Y[kT], X[kT]] \end{aligned}\right\}$$

Dacă bucla de reacție ar fi permanent activă automatul s-ar comporta asincron

Mașinile sincrone funcționează pe rând dacă întârzierile prin circuitele de reacție

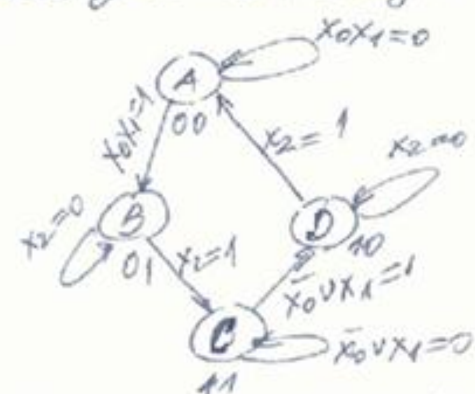
sunt suficient de scurte în raport cu perioada ceasului

Exemplu: Sinteza unui automat cu:

- trei intrări: x_0, x_1, x_2
- cinci ieșiri: $z_0, \ldots, z_4$
- patru stări A, B, C, D codificate prin variabilele de stare

Y_0	Y_1	Stare
0	0	A
0	1	B
1	1	C
1	0	D

Automatul funcționează după următoarea organigramă / diagramă de tranziții:



Realizarea tabelei de tranziții pentru stări și a tabelei pentru ieșiri

Semnale memorate cu Ri la t_1 — Semnale memorate cu Re la t_2 — PLA-9

Intrări	Stări prezente	Stări următoare	Ieșiri	
$x_0\ x_1\ x_2$	$y_0\ y_1$	$y_0\ y_1$	$z_0\ z_1\ z_2\ z_3\ z_4$	
0 * *	0 0	0 0	0 0 0 1 0	R1
* 0 *	0 0	0 0	0 0 0 1 0	R2
1 1 *	0 0	0 1	1 0 0 1 0	R3
* * 0	0 1	0 1	0 0 1 1 0	R4
* * 1	0 1	1 1	1 0 1 1 0	R5
1 0 *	1 1	1 1	0 1 0 0 0	R6
0 * *	1 1	1 0	1 1 0 0 0	R7
* 1 *	1 1	1 0	1 1 0 0 0	R8
* * 0	1 0	1 0	0 1 0 0 1	R9
* * 1	1 0	0 0	1 1 0 0 1	R10

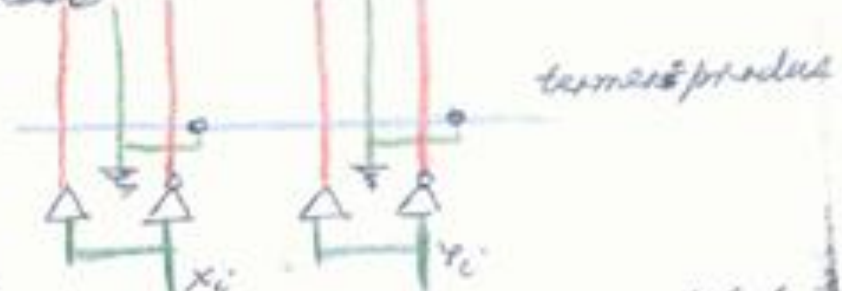
Reguli pentru programarea rețelei logice.

1. Pentru fiecare 1 logic din tabela pentru starea următoare și pentru ieșiri se va lăsa o cale de difuzie între liniile corespunzătoare a rețelei (verticală metal), peste traseul de polisiliciu și traseul orizontal de difuzie conectat la masă

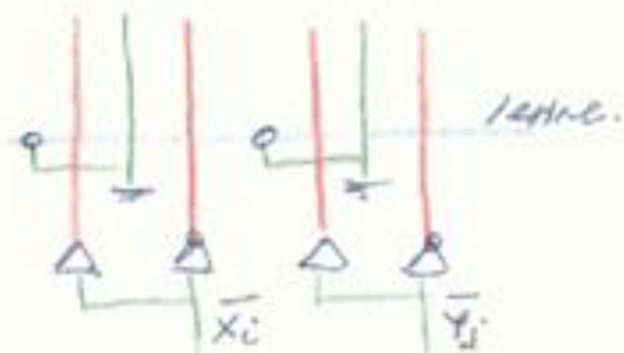
ARIA ȘI

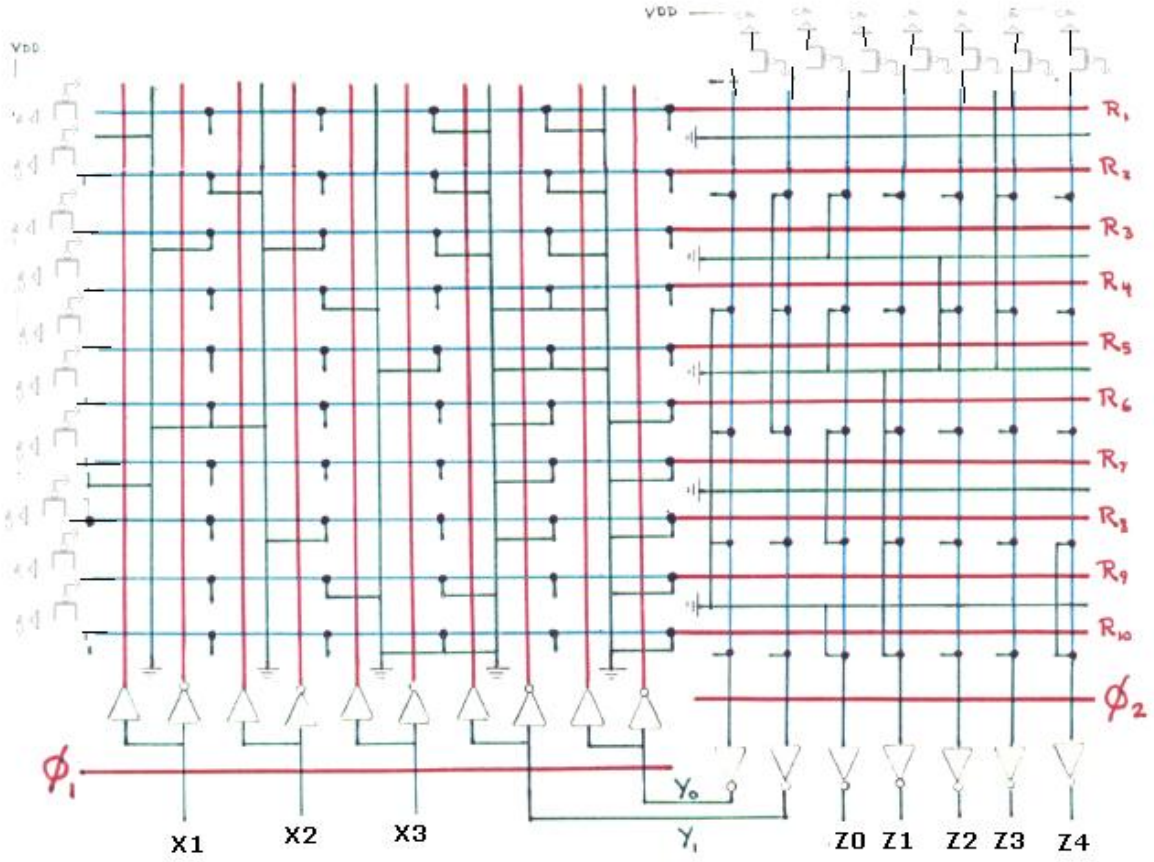
Pentru planul AND sunt două reguli: PCL-10

② Pentru fiecare 1 logic din tabelele pentru intrări și starea prezentă se trasează o cale de difuzie de la linia corespunzătoare a termenului produs peste rețeaua orizontală la traseul vertical de difuzie conectat la masă



③ Pentru fiecare termen egal cu 0 în tabela intrărilor și a stării prezente se va trasa o linie neîntreruptă de difuzie de la termenul produs la masă intersectând traseul de polisiliciu corespunzător intrării directe




VDD
VDD
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
ϕ2
ϕ1
Y0
Y1
X1
X2
X3
Z0
Z1
Z2
Z3
Z4

Mastrea secondaria se poate realiza pe o arie de $(150\lambda)^2$

Pentru $\lambda = 3\,\mu m$ (1978) (ochiul $\lambda = 0{,}3\,\mu m$)
arria ar fi de $(450\,\mu m)^2 \approx 0{,}002\,cm^2$

Mastrea contine peste 150 tranzistori
Si ocupă 1/25 din aria acest chip
integrat complet - ($0{,}25\,cm^2$)
Se tinde ca prezent le 1/25.000 din
aria cipului complet integrat