Proiectare logică

Curs 13

Proiectarea cu dispozitive logice programabile

Cristian Vancea

https://users.utcluj.ro/~vcristian/PL.html

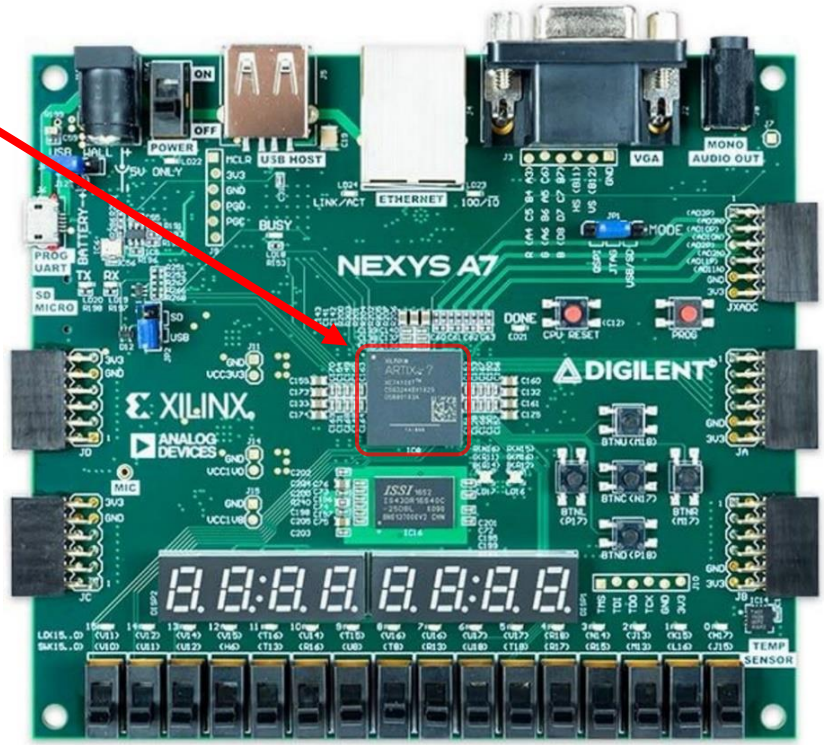
Cuprins

- Proiectarea cu dispozitive logice programabile
  - Circuite VLSI (Very Large Scale Integration) uzuale
  - Proiectare clasică vs. proiectare cu dispozitive logice programabile
  - Completitudine funcțională
  - Funcții universale
  - Celula logică de bază
  - Arhitecturi de dispozitive logice programabile

Proiectarea cu dispozitive logice programabile

Circuite VLSI (Very Large Scale Integration) uzuale

- Prezintă un grad extins de integrare.
- Exemple de circuite VLSI:
  - CPLD (Complex Programmable Logic Device);
  - FPGA (Field Programmable Gate Array).



https://digilent.com/reference/programmable-logic/nexys-a7/reference-manual

Proiectarea cu dispozitive logice programabile

Proiectarea clasică – etape

1. Definirea specificațiilor dispozitivului.
2. Construirea diagramei bloc.
3. Separarea secțiunilor din diagramă.
4. Detalierea până la nivelul logic de implementare.
5. Integrarea pieselor - se poate realiza cu produse software specializate.
6. Crearea prototipului + depanare + corecții (ex: îmbunătățiri de viteză).
7. Implementarea pe placă (Printed Circuit Board – PCB) – la acest nivel pot apărea ajustări care țin de constrângeri fizice ale PCB.

Obs: Se poate reveni la pași anteriori în mod repetat atunci când sunt necesare modificări esențiale => proces lent de durată (**dezavantaj**).

Proiectarea cu dispozitive logice programabile

Proiectarea cu dispozitive programabile - etape

1. Definirea specificațiilor dispozitivului.
2. Separarea în blocuri mari având ca suport fizic diverse categorii de componente: memorii, microprocesoare, dispozitive logice programabile (FPGA – Field Programmable Gate Array, CPLD – Complex Programmable Logic Device) și logica de comunicare între acestea.
3. Descrierea la nivel înalt a sistemului cu editor schematic sau limbaj de descriere hardware (ex: VHDL, Verilog).
4. Simularea sistemului – elimină efortul creării unui prototip fizic.
5. Crearea listei de componente și a interconexiunilor – rezultă o documentație în format specific numită *netlist*.
6. Implementarea pe placă + simulare mai detaliată.

Obs: Cele mai multe modificări apar la nivelul componentelor (FPGA, CPLD) și se pot aplica prin modificarea descrierii la nivel înalt => reducerea timpului de proiectare (**avantaj**).

Proiectarea cu dispozitive logice programabile

Completitudine funcțională

Definiție: Un circuit logic combinațional capabil să genereze o disjuncție (SAU) de conjuncții (ȘI) sau o conjuncție (ȘI) de disjuncții (SAU) peste intrările sale, cu capacitatea de inversare a cel puțin uneia din intrări, are proprietatea de completitudine funcțională.

Obs: Folosind în mod repetat un circuit cu completitudine funcțională se poate realiza orice funcție logică în forma canonică sau minimizată.

Proprietăți de bază ale unui circuit cu completitudine funcțională:

1. Poate construi una din funcțiile logice SAU ori ȘI.
2. Poate construi funcția logică NOT.

} Ex: SAU-NU, ȘI-NU

Obs$_1$: Cu funcțiile SAU și NOT se poate construi ȘI folosind *De Morgan*:

$$\overline{\bar{a} + \bar{b}} = a \cdot b$$

Obs$_2$: Cu funcțiile ȘI și NOT se poate construi SAU folosind *De Morgan*:

$$\overline{\bar{a} \cdot \bar{b}} = a + b$$

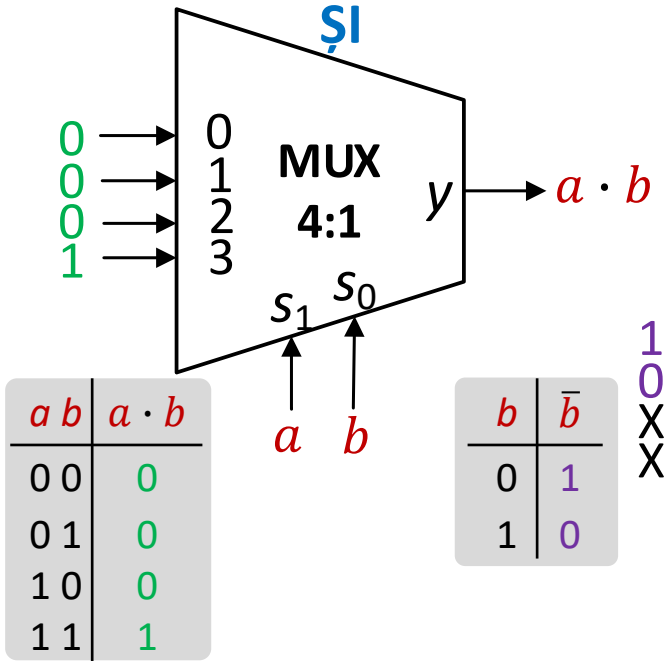
Proiectarea cu dispozitive logice programabile

Funcții universale

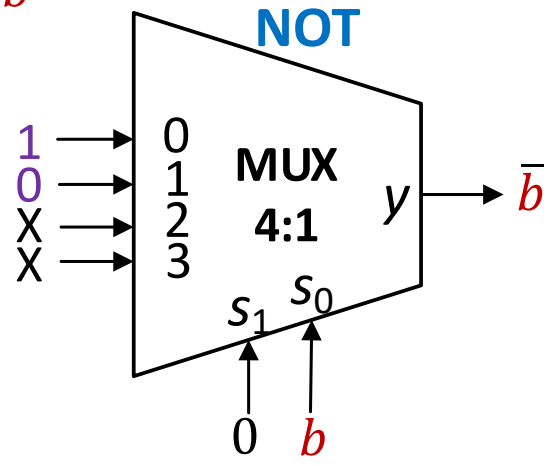
Definiție: Un circuit logic combinațional care poate fi configurat astfel încât să genereze orice funcție logică de intrările sale se numește generator de funcții sau funcție universală.

Familii de circuite funcții universale: memorii (ROM, RAM), MUX.

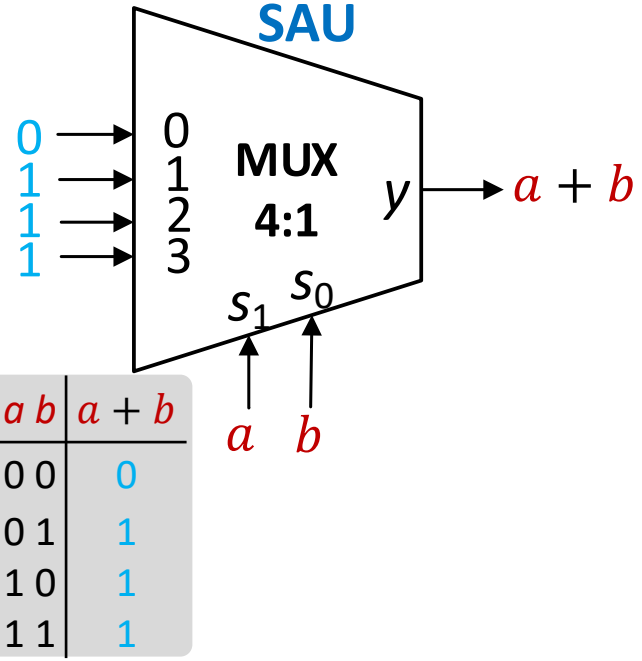
Ex: MUX 4:1 –> $y = x_0 \cdot \bar{a} \cdot \bar{b} + x_1 \cdot \bar{a} \cdot b + x_2 \cdot a \cdot \bar{b} + x_3 \cdot a \cdot b$



a b	$a \cdot b$
0 0	0
0 1	0
1 0	0
1 1	1



b	$\bar{b}$
0	1
1	0



a b	$a + b$
0 0	0
0 1	1
1 0	1
1 1	1

Obs$_1$: MUX are **completitudine funcțională**.

Obs$_2$: Pentru 3 sau 4 variabile se poate folosi MUX 8:1, respectiv MUX 16:1.

Proiectarea cu dispozitive logice programabile

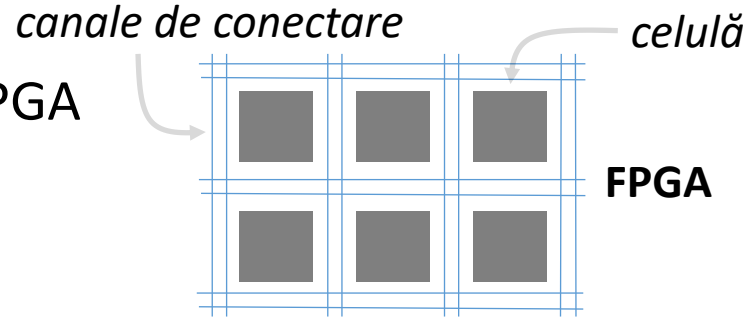
Completitudine funcțională vs. funcții universale în dispozitive programabile

Notă: Orice funcție universală are completitudine funcțională deci se poate realiza orice funcție logică prin utilizarea uneia sau mai multor unități.

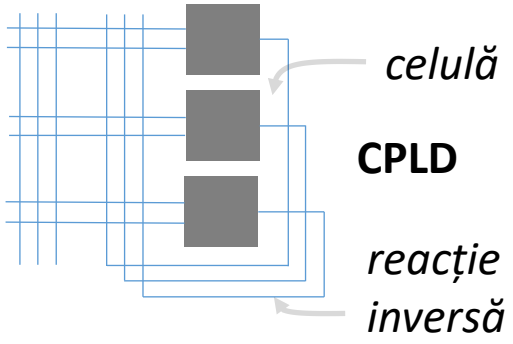
Obs: Implementarea funcțiilor logice complexe poate necesita mai puține circuite de tip funcție universală, decât circuite cu completitudine funcțională.

Definiție: *Dispozitivele logice programabile* = colecții de celule universale sau cu completitudine funcțională plasate într-o rețea de interconectare:

- arhitecturi cu canale de conectare de tip FPGA
  - matrice de celule cu interconexiuni orizontale și verticale configurabile;



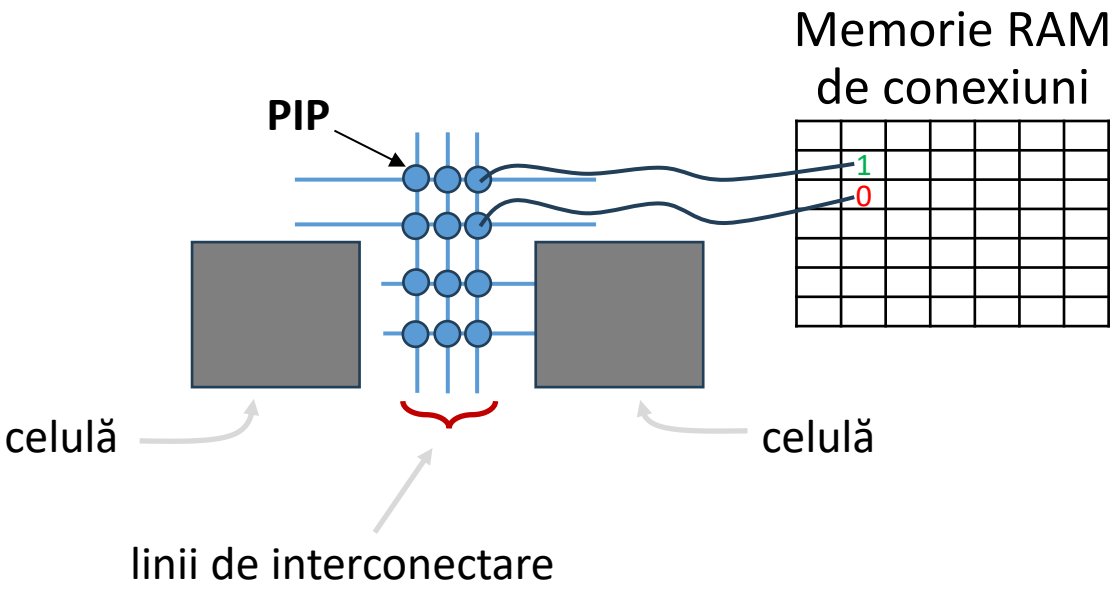
- arhitecturi *foldback* de tip CPLD
  - seturi de celule cu legături configurabile cu reacție inversă.



Proiectarea cu dispozitive logice programabile

Rețeaua de interconectare

- Este compusă din linii de interconectare și puncte de conexiune propramabile (**PIP** – Programmable Interconnection Point), controlate de la o memorie RAM, scrisă corespunzător la programarea dispozitivului.



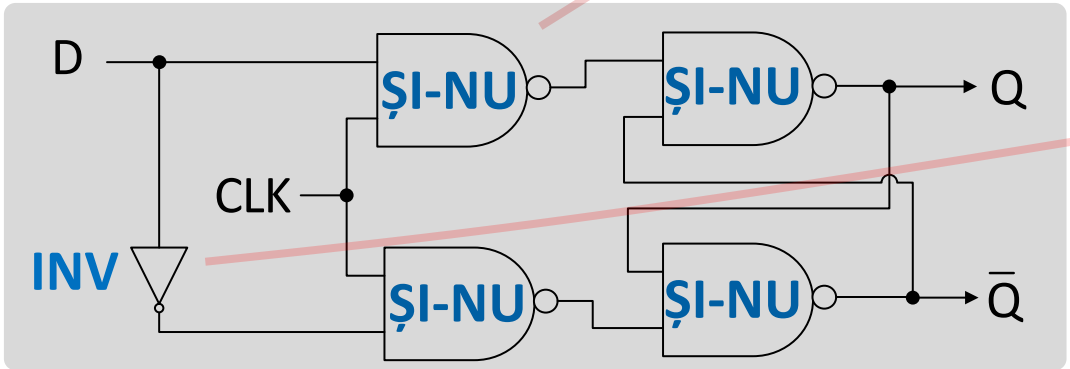
Proiectarea cu dispozitive logice programabile

Celula logică de bază

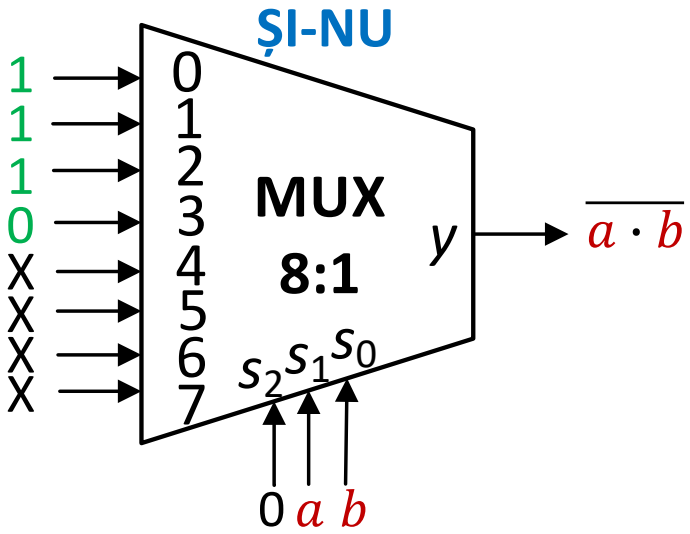
Ex: Poate conține MUX 8:1.

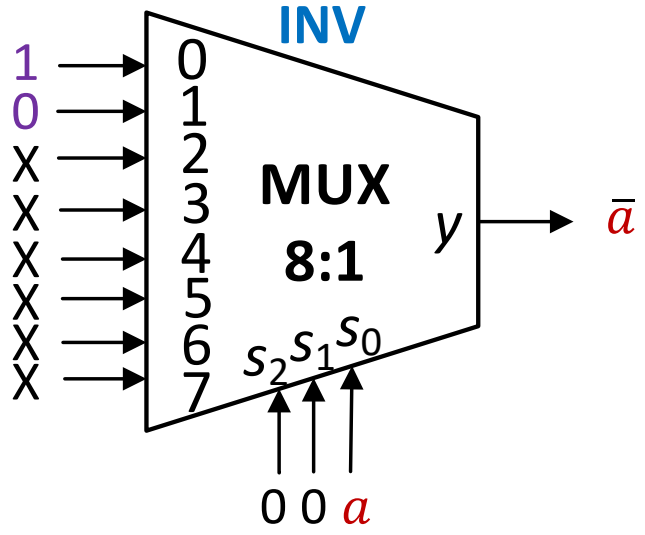
Obiectiv: Să se implementeze un bistabil D.

a b	$\overline{a \cdot b}$
0 0	1
0 1	1
1 0	1
1 1	0



a	$\bar{a}$
0	1
1	0



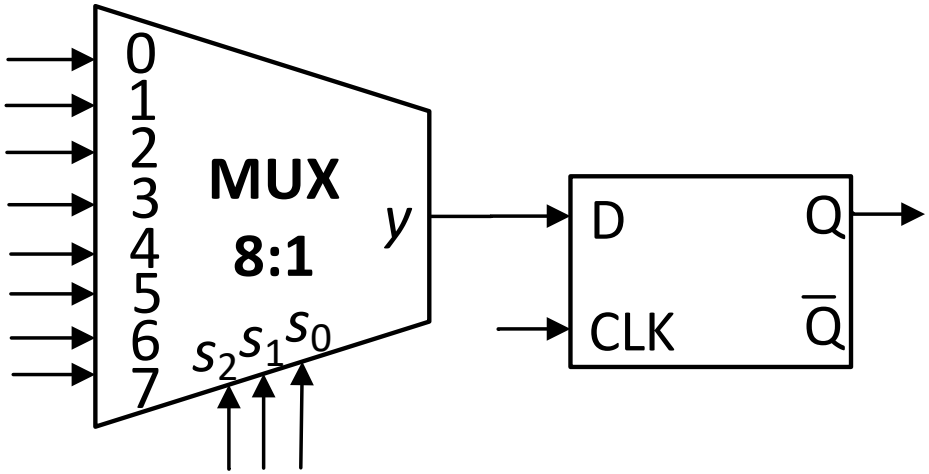


Obs: Un bistabil D ar avea nevoie de 5 x MUX 8:1 => implementare costisitoare.

Proiectarea cu dispozitive logice programabile

Celula logică de bază hibridă

Soluția$_1$: Se leagă MUX 8:1 la un bistabil D => celulă hibridă.

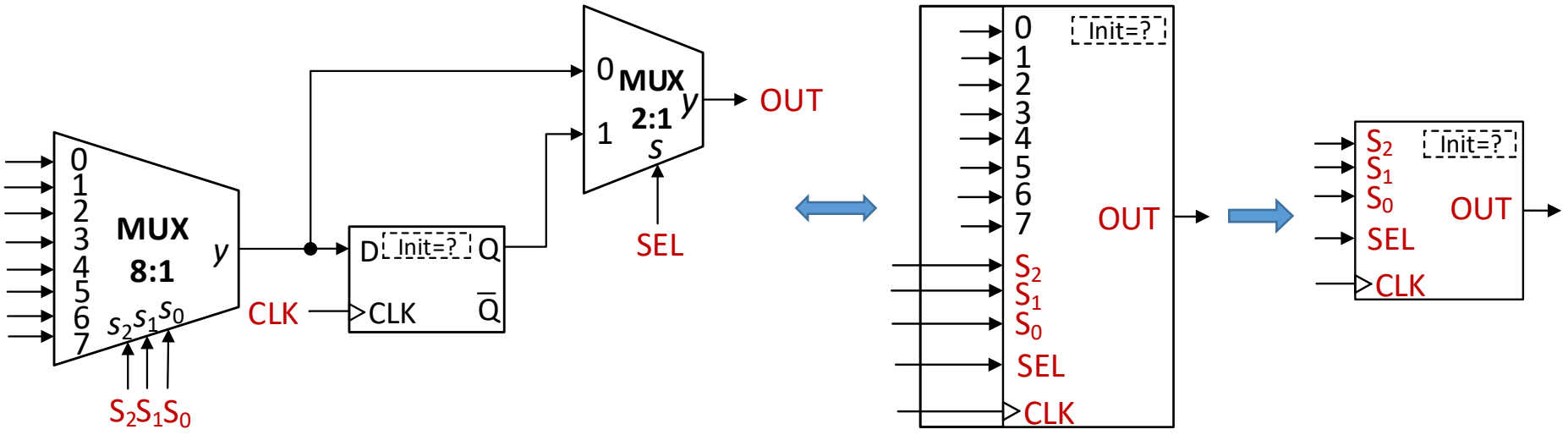


Obs: Toate funcțiile implementate sunt obligate să folosească bistabilul D => dezavantaj.

Proiectarea cu dispozitive logice programabile

Celula logică de bază hibridă

Soluția$_2$: Se adaugă un MUX 2:1 pe ieșire care va decide dacă rezultatul este dat de MUX 8:1 sau de bistabilul D.

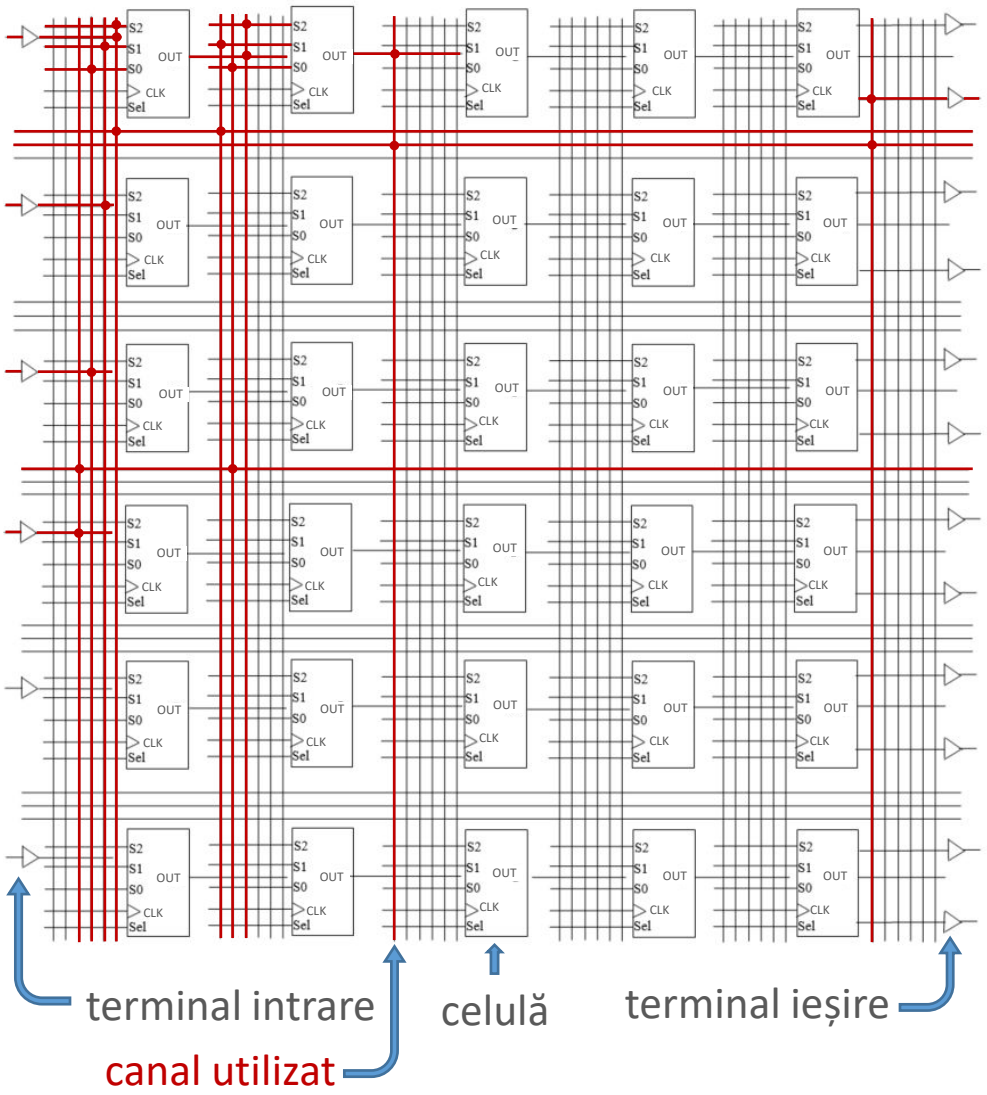


Notă: Se poate implementa orice funcție combinațională sau secvențială de până la 3 variabile. La nevoie se poate configura valoarea de start 0 sau 1 a bistabilului la punerea în funcțiune.

Obs: Intrările de date ale MUX 8:1 se configurează intern => nu apar în afara celulei => se reduce numărul de conexiuni în rețeaua de interconectare.

Proiectarea cu dispozitive logice programabile

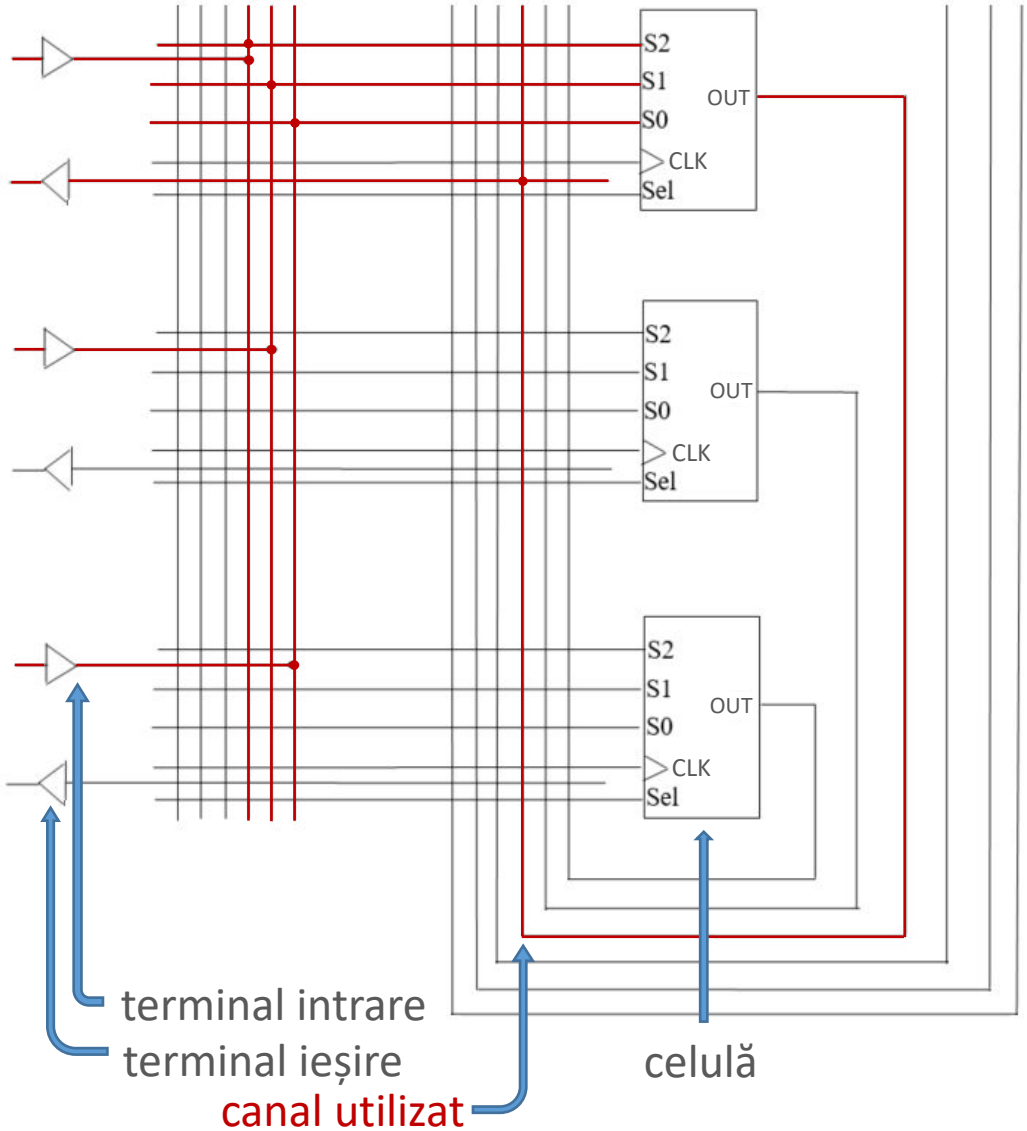
Arhitectură cu matrice de celule și canale de conectare intercelulare – FPGA



- Se pot realiza conexiuni în toate punctele de intersecție dintre linii orizontale și verticale.
- Semnalele externe intră prin terminale (buffer-e) de intrare.
- Prin intermediul conexiunilor semnalele externe se pot conecta la intrările celulelor.
- Prin intermediul conexiunilor ieșirea unei celule poate fi conectată la terminale de ieșire sau la intrările altor celule.
- Rezultatele ies prin terminale (buffer-e) de ieșire.
- Funcțiile de mai multe variabile necesită alocarea mai multor celule.

Proiectarea cu dispozitive logice programabile

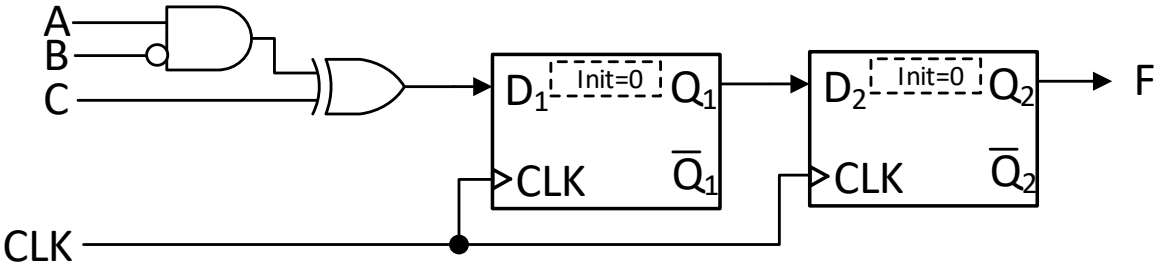
Arhitectură cu set de celule și canale cu reacție inversă (*foldback*) – CPLD



- Se pot realiza conexiuni în toate punctele de intersecție dintre linii orizontale și verticale.
- Semnalele externe intră prin terminale (buffer-e) de intrare.
- Prin intermediul conexiunilor semnalele externe se pot conecta la intrările celulelor.
- Prin intermediul conexiunilor ieșirea unei celule poate fi conectată la terminale de ieșire sau la intrările altor celule.
- Rezultatele ies prin terminale (buffer-e) de ieșire.
- Funcțiile de mai multe variabile necesită alocarea mai multor celule.

Proiectarea cu dispozitive logice programabile

Ex_1: Să se implementeze cu celule logice de bază circuitul:



Rezolvare:

Fiindcă sunt 2 bistabile => se vor folosi 2 celule.

Se determină expresiile logice ale intrărilor bistabilelor:

$D_1 = (A \cdot \overline{B}) \oplus C$; $D_2 = Q_1$

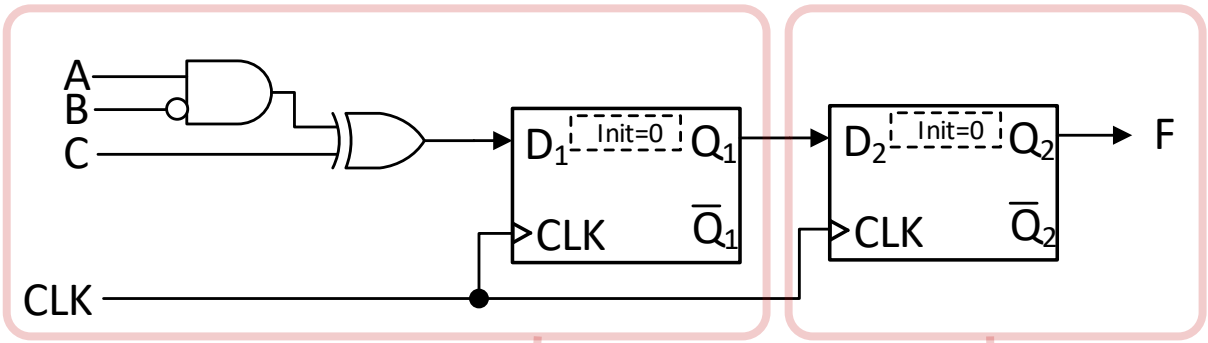
Deoarece implementarea se bazează pe MUX 8:1 se utilizează tabelul de adevăr al expresiilor (forma canonică).

A	B	C	D_1
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Q_1	D_2
0	0
1	1

Proiectarea cu dispozitive logice programabile

Ex$_1$: Să se implementeze cu celule logice de bază circuitul de mai jos.

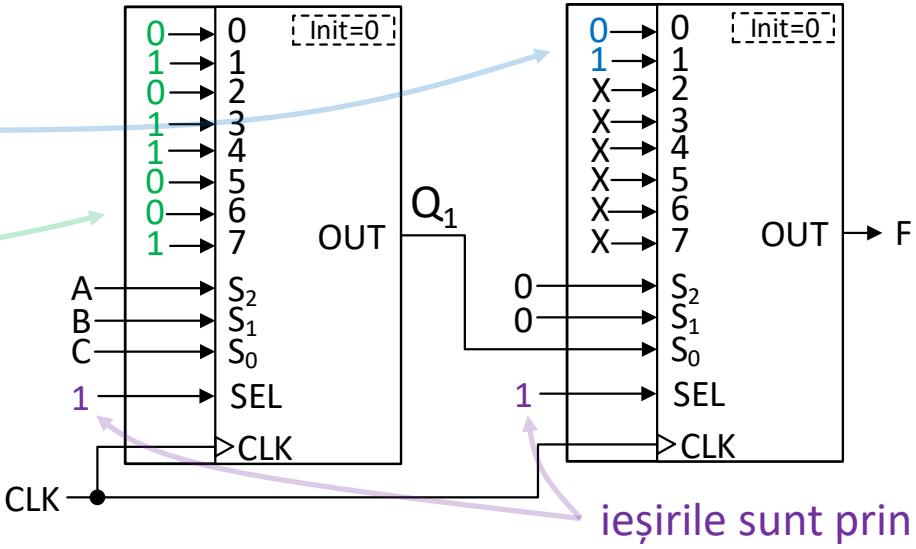


Rezolvare:

Se determină valorile intrărilor de date (0-7) pentru cele 2 celule utilizând tabelele de adevăr și se realizează interconexiunile între celule.

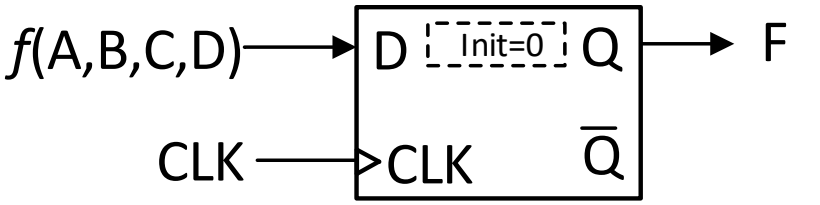
A B C	D_1
0 0 0	0
0 0 1	1
0 1 0	0
0 1 1	1
1 0 0	1
1 0 1	0
1 1 0	0
1 1 1	1

Q_1	D_2
0	0
1	1



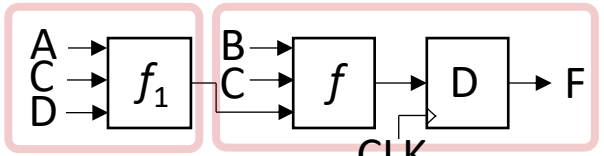
Proiectarea cu dispozitive logice programabile

Ex$_2$: Să se implementeze cu celule logice de bază circuitul de mai jos unde $f(A,B,C,D) = \sum(2,3,7,10,11)$.



Rezolvare: $f(A,B,C,D) = \sum(2,3,7,10,11) \xrightarrow{\text{min. DK}} \overline{B}\cdot C + \overline{A}\cdot C\cdot D$ ⟸ FDM

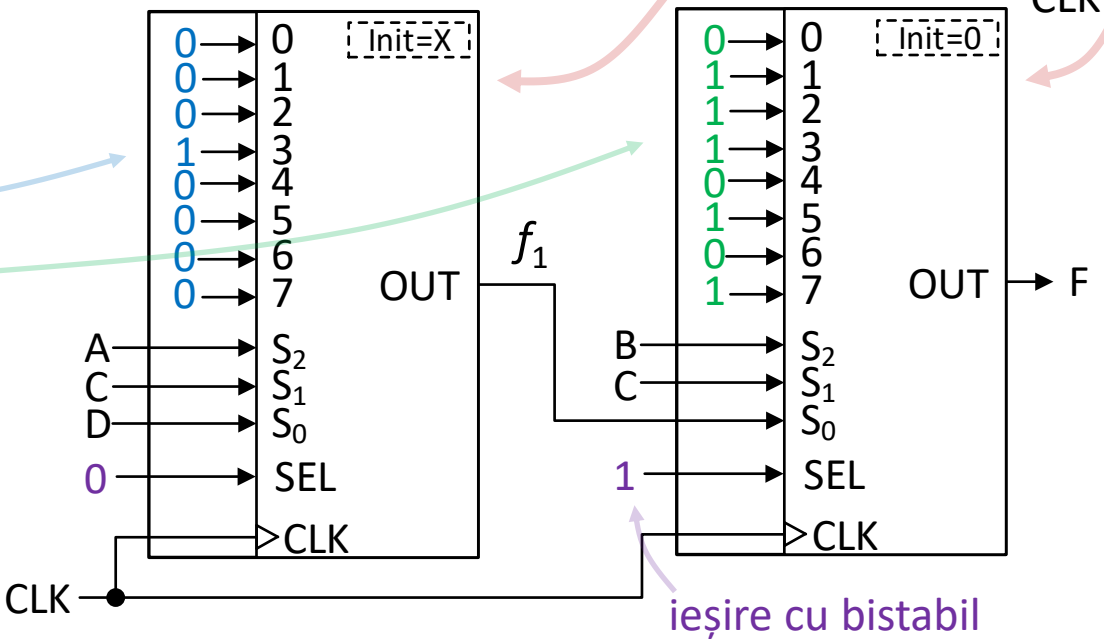
Fiindcă celula are 3 variabile S_2, S_1, S_0 este necesară rescrierea expresiei ***f*** cu funcții de 3 variabile astfel: $f_1 = \overline{A}\cdot C\cdot D$, $f = \overline{B}\cdot C + f_1$ ⟹



Se determină tabelele de adevăr și se planifică circuitul:

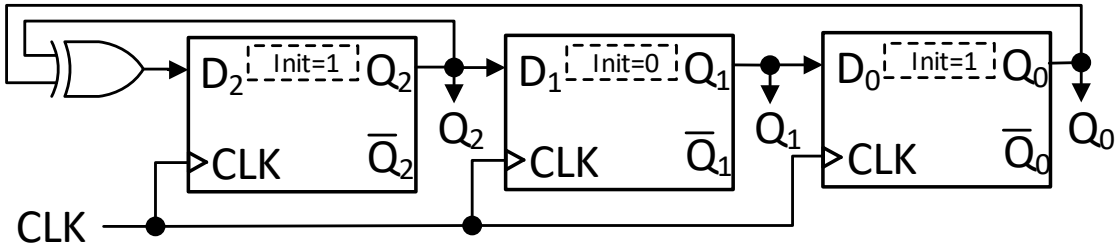
A C D	f_1
0 0 0	0
0 0 1	0
0 1 0	0
0 1 1	1
1 0 0	0
1 0 1	0
1 1 0	0
1 1 1	0

B C f_1	f
0 0 0	0
0 0 1	1
0 1 0	1
0 1 1	1
1 0 0	0
1 0 1	1
1 1 0	0
1 1 1	1



Proiectarea cu dispozitive logice programabile

Ex$_3$: Să se implementeze cu celule logice de bază un generator de secvențe pseudo-aleatoare pe 3 biți:



Rezolvare: Se folosesc 3 celule.

Intrările bistabilelor:

$D_2 = Q_2 \oplus Q_0$

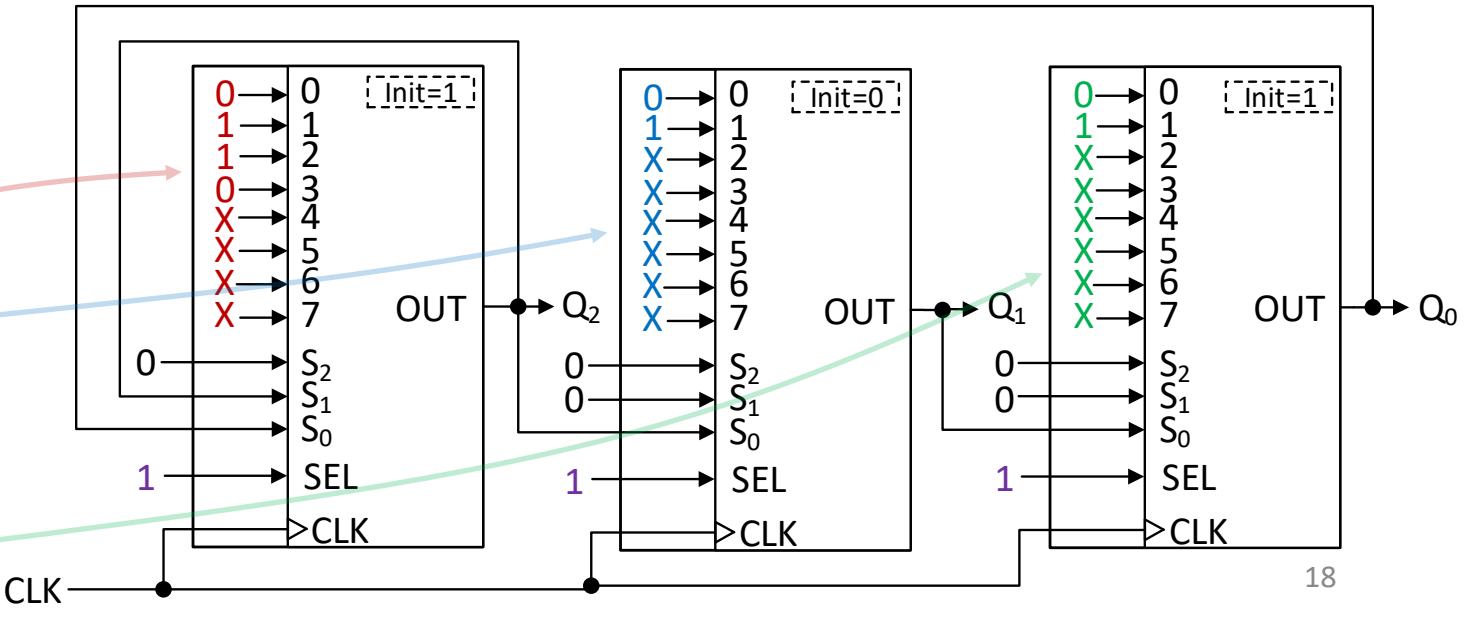
$D_1 = Q_2$

$D_0 = Q_1$

Q_2	Q_0	D_2
0	0	0
0	1	1
1	0	1
1	1	0

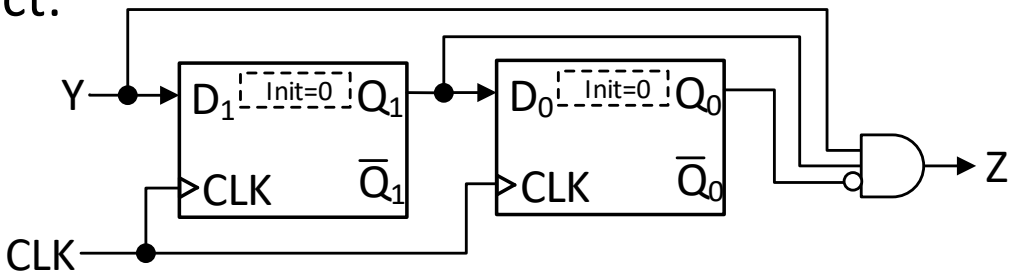
Q_2	D_1
0	0
1	1

Q_1	D_0
0	0
1	1



Proiectarea cu dispozitive logice programabile

Ex$_4$: Să se implementeze cu celule logice de bază un detector al secvenței “110” dintr-un șir de biți citiți serial de la dreapta la stânga pe intrarea Y, sincron cu semnalul de tact:



Rezolvare: Se folosesc 3 celule.

Intrările bistabilelor: $D_1 = Y \quad D_0 = Q_1$

Ieșirea: $Z = Y \cdot Q_1 \cdot \overline{Q_0}$

Y	D_1
0	0
1	1

Q_1	D_0
0	0
1	1

Y	Q_1	Q_0	Z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0