



Circuite integrate digitale

Curs 8



Cuprins

- algoritmul de minimizare (continuare)
- hazardul
- implementarea funcțiilor logice cu porți NAND
- ALU

- automate simple
 - automate bistabile
 - numărătorul

Diagrame Karnaugh și Veitch

		ab			
		00	01	11	10
c	0	m_0	m_1	m_3	m_2
	1	m_4	m_5	m_7	m_6

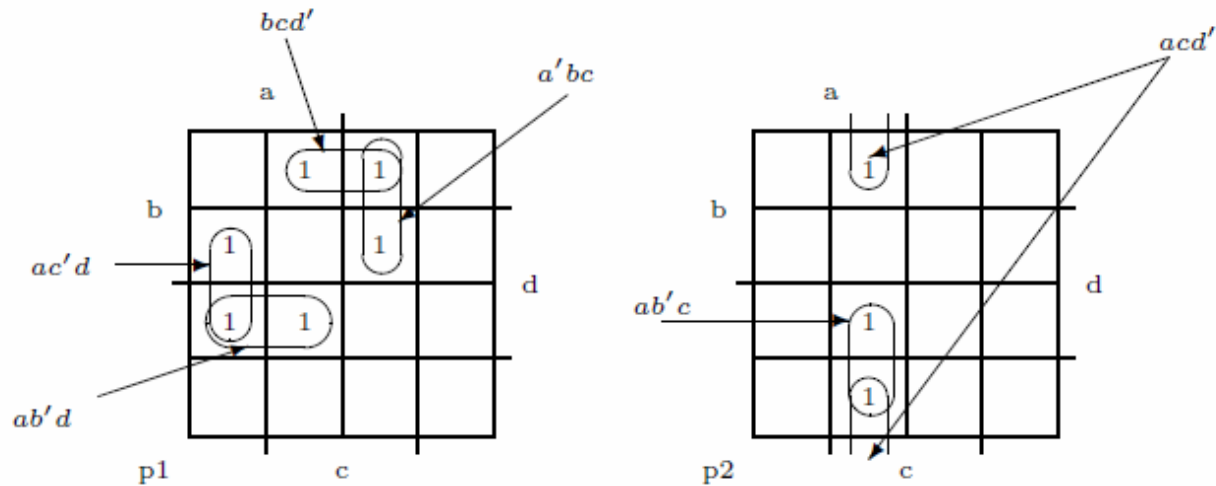
		ab			
		00	01	11	10
cd	00	m_0	m_1	m_3	m_2
	01	m_4	m_5	m_7	m_6
	11	m_{12}	m_{13}	m_{15}	m_{14}
	10	m_8	m_9	m_{11}	m_{10}

		a	
		m_3	m_1
b		m_2	m_0

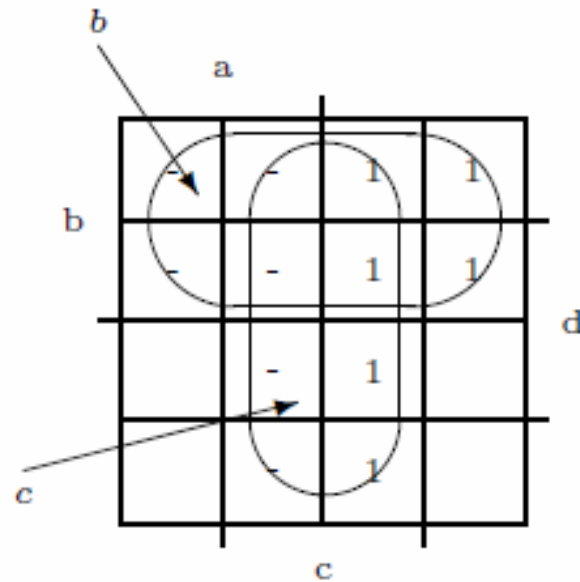
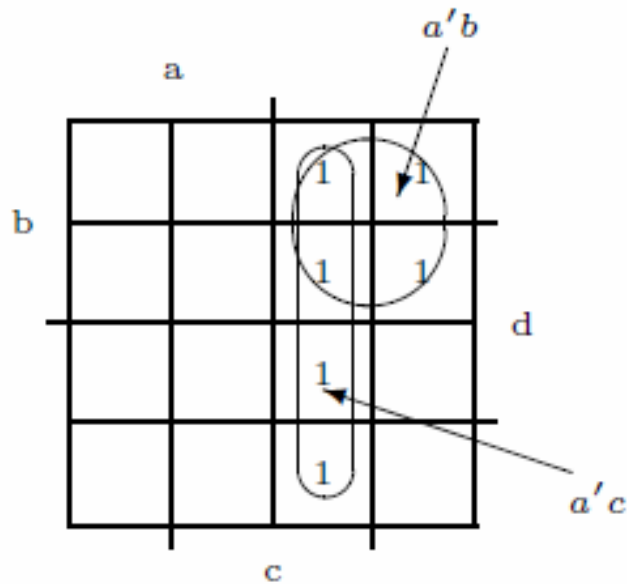
		a			
		m_6	m_7	m_3	m_2
b		m_4	m_5	m_1	m_0
			c		

		a			
		m_{12}	m_{14}	m_6	m_4
b		m_{13}	m_{15}	m_7	m_5
			d		
c		m_9	m_{11}	m_3	m_1
		m_8	m_{10}	m_2	m_0
		c			

Algoritmul de minimizare



Cu valori "don't care" (funcții incomplet definite)

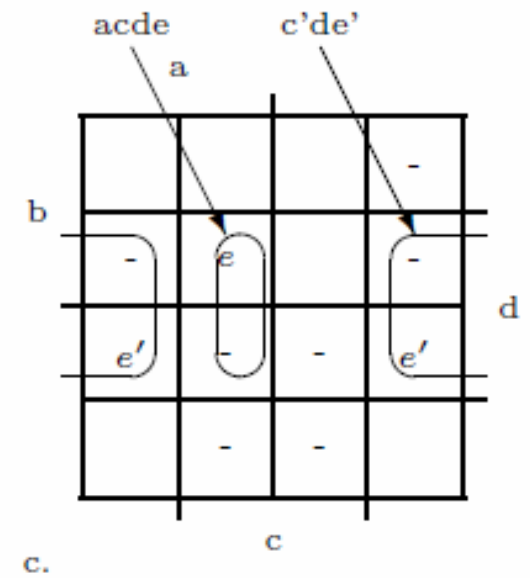
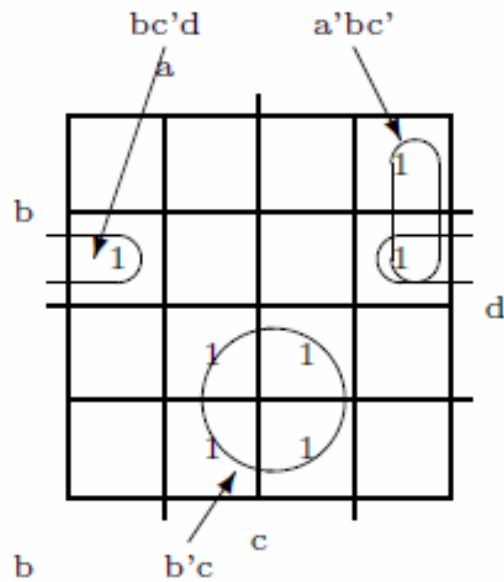
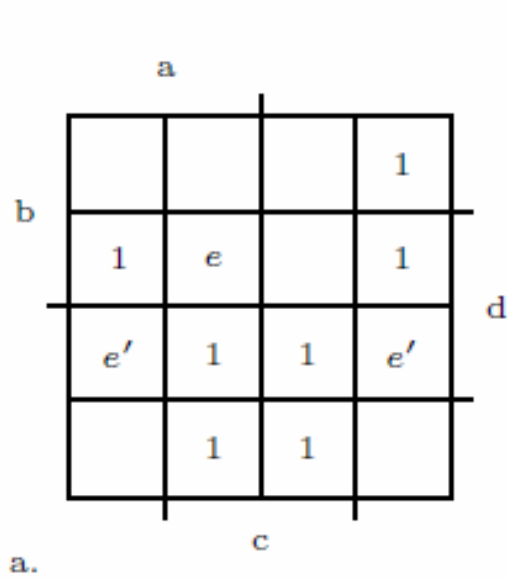




Cu variabile incluse

- pentru mai mult de 4 variabile: diagrame cu variabile incluse
- se imparte funcția în două "părți" și se rezolvă în două etape (diagrame)
 - doar valorile de 1 și "-"
 - diagrama care conține variabilele, valorile de 1 sunt înlocuite cu "-"
- trebuie verificate la final eventualele adiacențe!

$$(bc'd + a'bc' + b'c) + (acde + c'de')$$



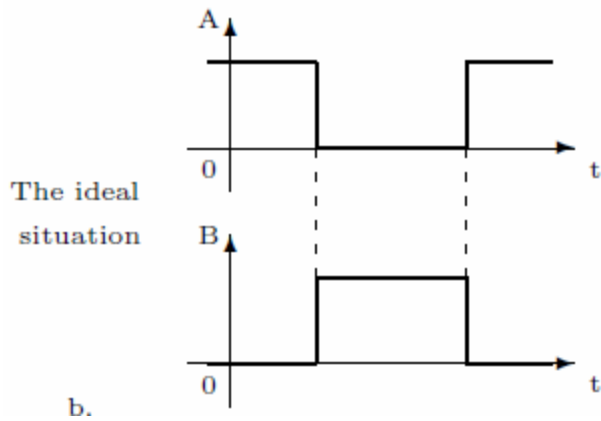
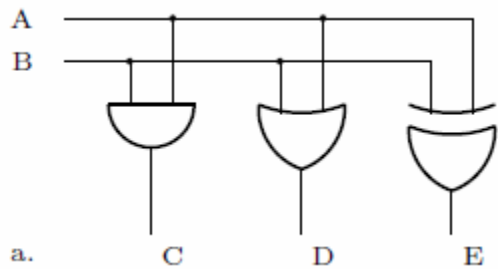


Hazard

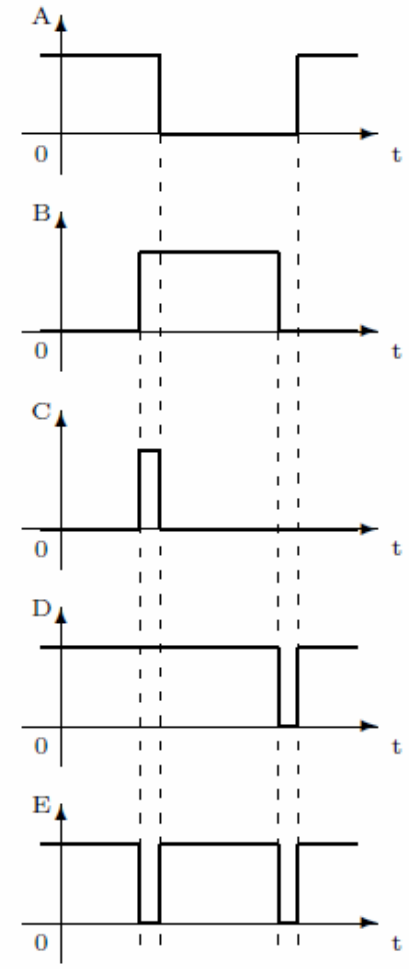
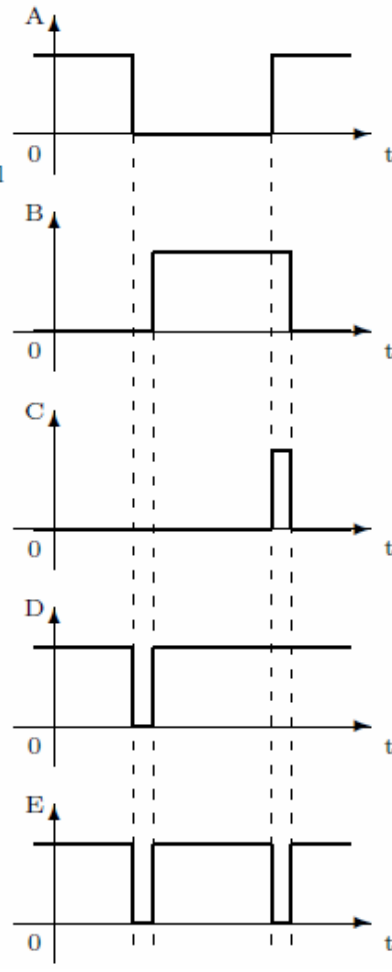
- de comutare – nu există soluții
- de propagare

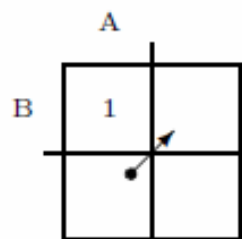
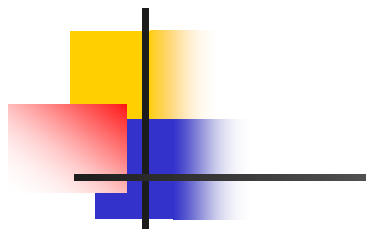
- evidențierea hazardului pe diagramă
- corectarea hazardului

Hazard datorat intrărilor asincrone

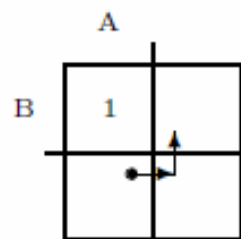


The actual situations

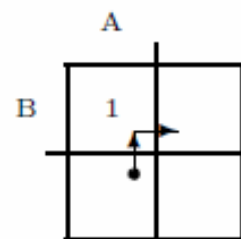




IDEAL

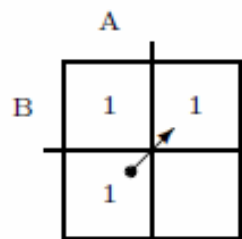


"A < B"

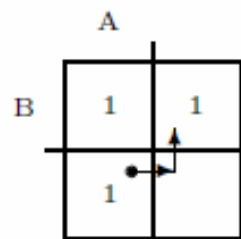


"B < A"

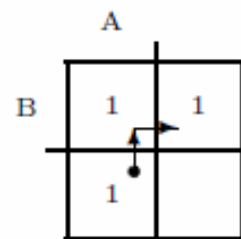
AND



IDEAL

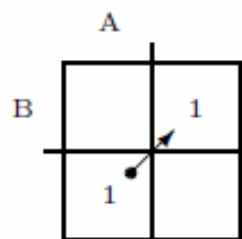


"A < B"

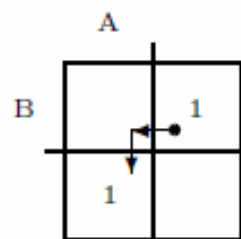


"B < A"

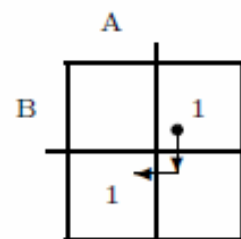
OR



IDEAL



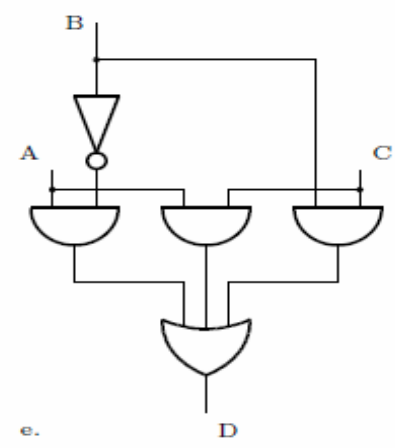
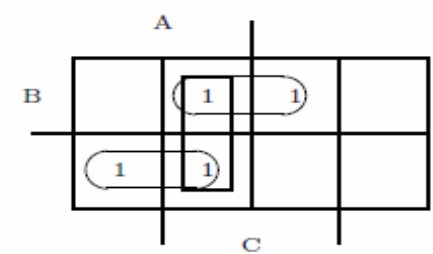
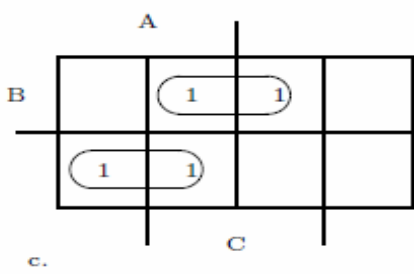
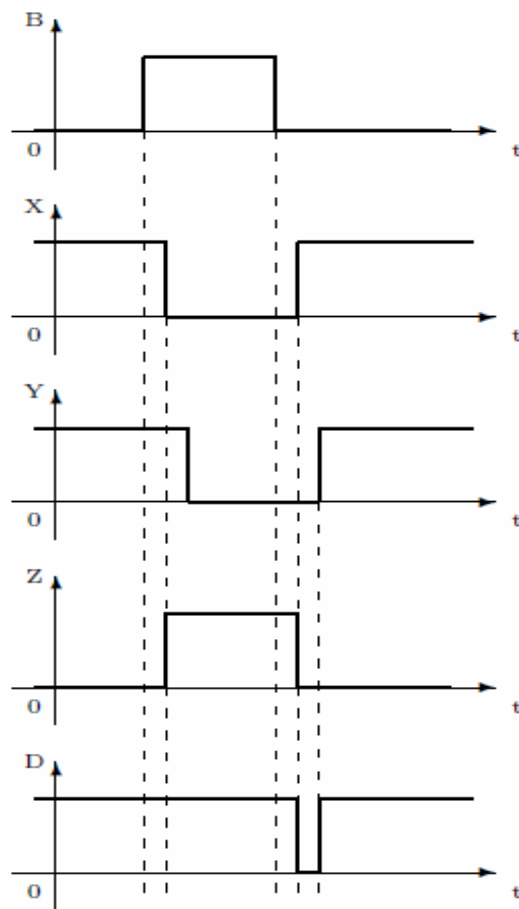
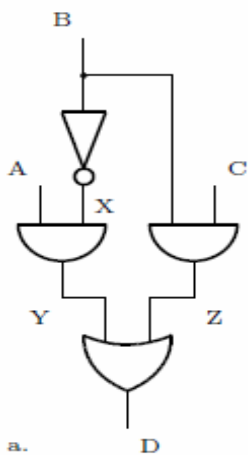
"A < B"



"B < A"

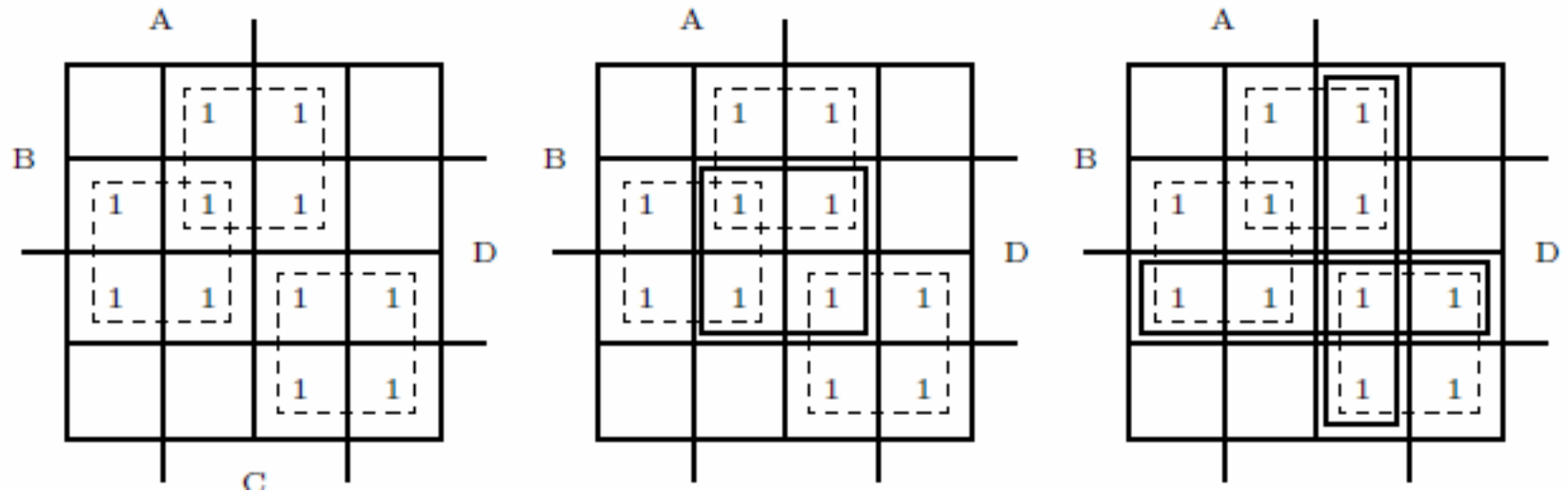
XOR

Hazard de propagare



Eliminarea hazardului de propagare

- eliminarea suprafețelor adiacente prin introducerea unor termeni suplimentari
- introducerea porților logice corespondente în circuit





Implementare cu porți NAND

- efectul propagării prin porți inversoare/neinversoare
- teoremele lui de Morgan:
 - $(a + b)' = a'b'$
 - $(ab)' = a' + b'$



Aplicații

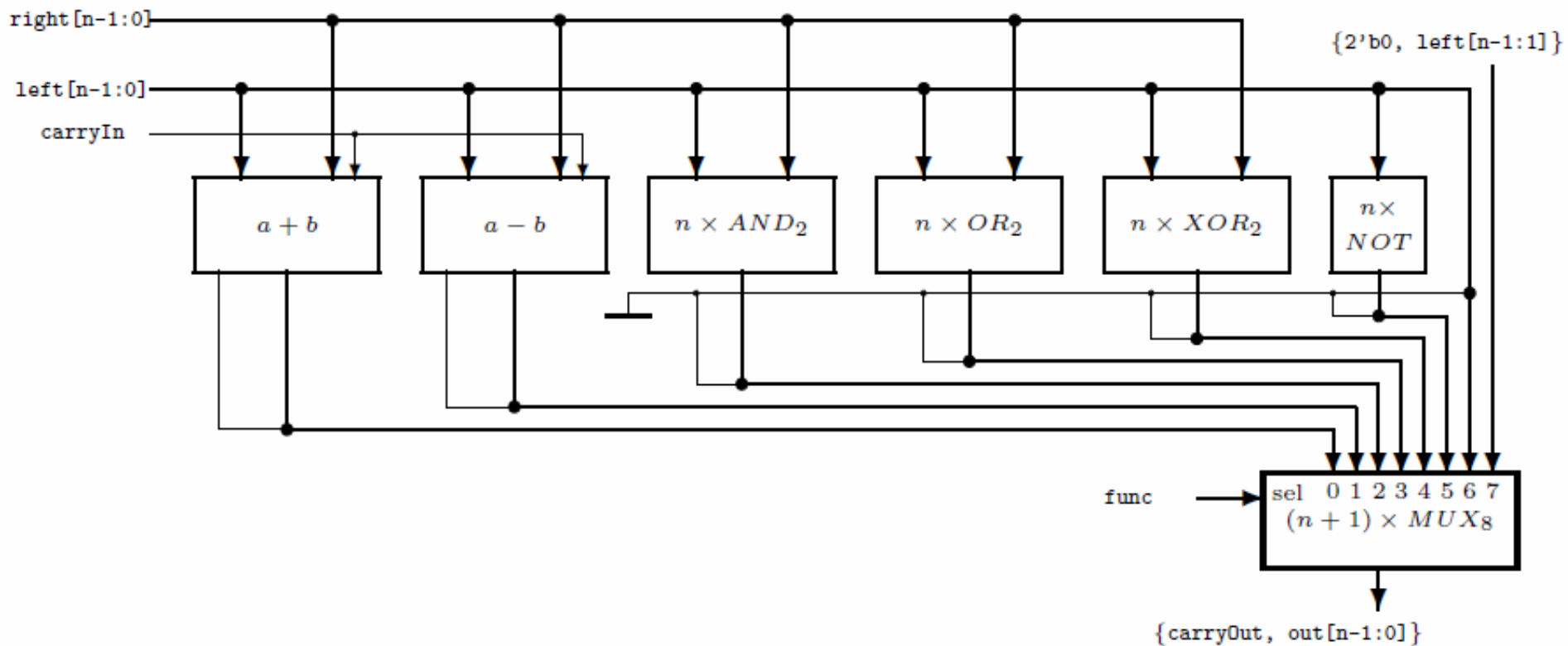
- exemple de minimizare
- eliminarea hazardului
- includerea variabilelor in tabel
- implementare cu MUX



Tema 10

- Se dă funcția $f = ac'd' + a'bc + a'b'd + acd' + a'bc'$
- se cere
 - tabelul de adevăr
 - diagrama
 - minimizare cu ajutorul diagramei
 - implementare cu porți în varianta minimală
 - implementarea anterioară are hazard? de ce? corecții hazardul

Unitate aritmetico-logică





TEMA OBLIGATORIE NR. 1

- Descrieți în Verilog un circuit RALU (o unitate aritmetico-logică cu registre) care realizează 8 funcții pentru operanzi de 4 biți. Proiectul va fi astfel conceput încât să poată fi implementat cu plăcile Nexys 2.
- Data limită a predării temei: 14 mai
- Modalitatea de predare: pe email, la adresa monica.dascalu@upb.ro cu subiectul TEMA 1, numele prenumele, grupa
- Veți trimite un fișier Word cu numele grupa_nume_prenume_TEMA1 care va conține (pentru punctajul maxim) următoarele:
 - schema bloc a circuitului adaptat pentru implementarea pe placi
 - codul sursa Verilog pentru acest circuit
 - modulul de test pentru ALU și modulul de test pentru tot circuitul
 - formele de unda obținute prin simularea modulelor de test
- În cazul în care mai mulți studenți predau teme identice, va fi punctată numai prima temă primită.
- Punctaj maxim: 10 pct din punctajul final (maxim 100)



Recapitulare

- circuite combinaționale:
 - porți
 - MUX, DCD, DMUX, sumator, carry look-ahead, transcodor...
 - PLD
- circuite de memorare
 - latch-uri
 - bistabile
 - memorii
- numărătorul

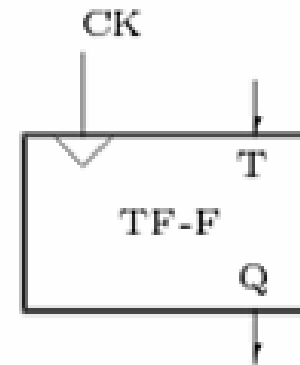


Sisteme de ordin 2 (automate finite)

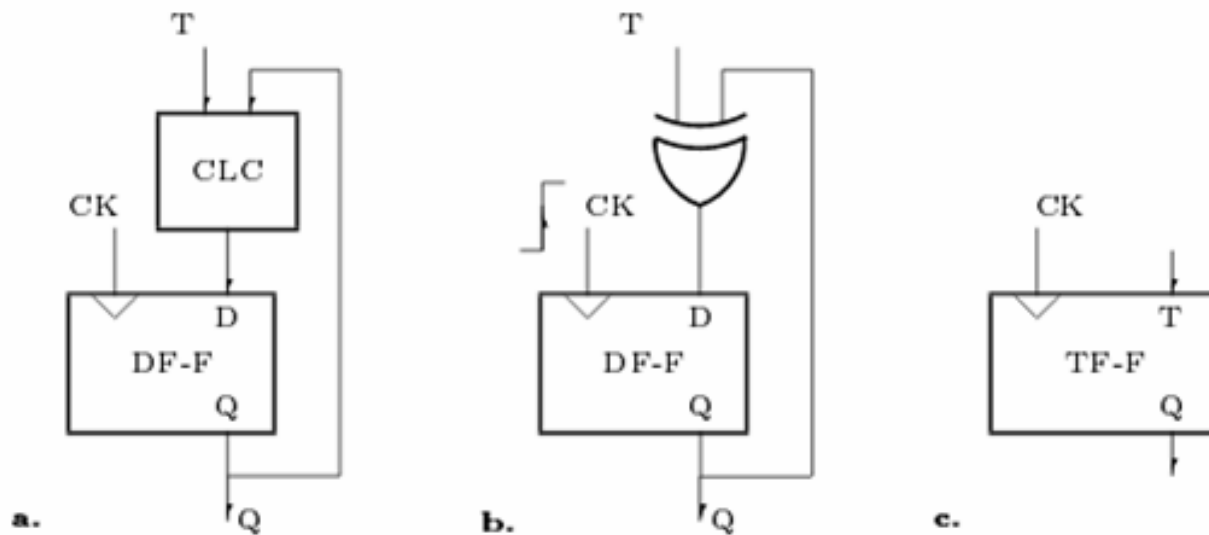
- comportament autonom
- sisteme de comandă și control
- numărătorul este un automat!

Bistabilul T

- cel mai simplu automat
 - T=0: hold
 - T=1: switch
- pentru T=1 are un comportament autonom
- divizor de frecvență



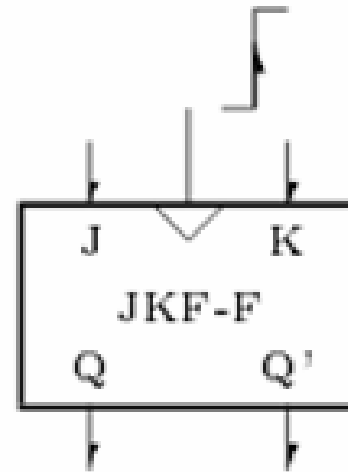
Bistabilul T



Bistabilul JK

- operații

- nop: $J=K=0$
- reset: $J=0, K=1$
- set: $J=1, K=0$
- switch: $J=K=1$





Aplicații

- folosiți un bistabil JK pentru a obține un bistabil T
- proiectarea schemei interne a bistabilului T
- proiectarea schemei interne a bistabilului JK

Bistabilul JK

