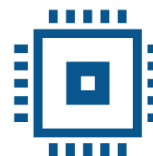




Universitatea *Transilvania* din Braşov
Facultatea de Inginerie Electrică şi Ştiinţa Calculatoarelor
Departamentul Automatică şi Informatică



PROIECT

ANALIZA ŞI SINTEZA CIRCUITELOR NUMERICE II

Echipă:

Pătrăhau Cătălin Ionuţ

Grabovenco Bogdan–Iulian

Grupe:

AIA4LF413

AIA4LF412

Îndrumător:

Prof. Dr. Ing. Moldoveanu Florin Dumitru

Braşov 2024

CUPRINS

1. Tema de proiect.....	3
2. Noțiuni introductive	5
2.1 Circuite logice secvențiale.....	5
2.2 Automate de tipul Mealy.....	6
3. Metoda Huffman	6
3.1 Etapa 1 – funcția de tranziție și funcția de ieșire.....	6
3.2 Etapa 2 – graful de tranziție și matricea primitivă a stărilor	8
3.3 Etapa 3 – reducerea stărilor	10
3.4 Etapa 4 – codificarea stărilor.....	13
3.5 Etapa 5 – matricea de tranziție a stărilor și funcțiile de excitație.....	16
3.6 Etapa 6 – matricele ieșirilor și funcțiile de ieșire.....	19
3.7 Etapa 7 – implementarea funcțiilor de excitație și a funcțiilor de ieșire	20
3.8 Etapa 8 – analiza schemei.....	21
4. Bibliografie.....	23
5. Anexa A.....	24

1. Tema de proiect

Tema de proiect nr. 36

Să se proiecteze un automat secvențial asincron al cărui caiet de sarcini este descris mai jos.

În proiectarea automatului secvențial asincron se va folosi metoda matriceală / Huffman.

Caiet de sarcini:

Se cere să se proiecteze un automat secvențial care să funcționeze după următorul protocol:

Automatul are două intrări, u_1 și u_2 , și două ieșiri y_1 și y_2 . Dacă se aplică mai întâi intrarea u_1 , ieșirea y_1 trece în 1 logic (y_2 rămâne în 0 logic). Ieșirea y_1 își menține valoarea 1 chiar dacă intrarea u_1 trece în 0 logic, și devine 0 numai când se aplică u_2 . Dacă, după aplicarea lui u_1 ($u_1 = 1$) se aplică și u_2 ($u_1 u_2 = 11$), ambele ieșiri trec în 0 și rămân în 0 până când ambele intrări devin 0. Dacă se aplică mai întâi u_2 (din starea inițială, când $u_1 = u_2 = y_1 = y_2 = 0$), y_2 ia valoarea 1 ($y_1 = 0$) și se menține la această valoare chiar dacă u_2 trece în 0. Ieșirea y_2 trece în 0 dacă, după aplicarea lui u_2 ($u_2 = 1$) se aplică și u_1 ($u_1 u_2 = 11$), sau dacă, după ce u_2 a devenit 0, se aplică u_1 .

Se consideră că la un moment dat de timp nu se modifică ambele intrări simultan.

Diagrama de semnale „intrări-ieșiri” este prezentată în figura 1.1.

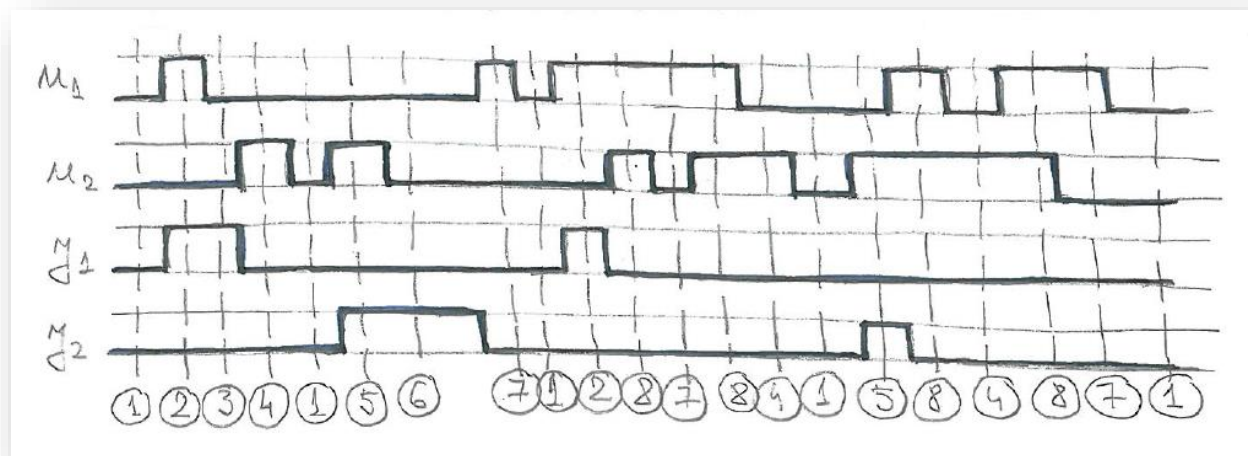


fig.1.1

Cerințe de proiectare:

În rezolvarea temei proiectului se vor trata următoarele probleme:

- Obținerea matricei/tabelei primitive a stărilor, a matricei complete a ieșirilor, precum și a grafului de tranziții.
- Reducerea numărului de stări ale matricei primitive și construirea matricei reduse a stărilor.
- Codificarea stărilor matricei reduse.
- Întocmirea matricelor tranzițiilor stărilor și obținerea expresiilor funcțiilor de excitație ale automatului.
- Întocmirea matricelor ieșirilor și obținerea expresiilor funcțiilor de ieșire ale automatului.
- Implementarea funcțiilor logice obținute cu circuite integrate (se vor utiliza bistabili de tip D pentru implementarea secțiunii de memorie și porți logice pentru structura logică combinațională).
- Analiza schemei logice obținute.

Pe schema logică obținută se vor specifica tipul și gradul de utilizare al fiecărui circuit integrat.

Bibliografia recomandată:

- [1] Ștefan, Gh. – Circuite și sisteme digitale, Ed. Tehnică, București, 2000.
- [2] Wakerly, J.F. – Circuite digitale, Ed. Teora, București, 2002.
- [3] Wilkinson, B. – Electronică digitală. Bazele proiectării, Ed. Teora, București, 2002.
- [4] Mano, M.M. – Digital Design, Prentice Hall International, London, 2002.
- [5] Moldoveanu, F., Floroian, D. – Circuite logice și comenzi secvențiale. Circuite logice combinaționale, Ed. Universității Transilvania din Brașov, 2003.
- [6] Toacșe, Gh., Nicula, D. – Electronică digital , vol. I, Ed. Tehnică, București, 2005.

2. Noțiuni introductive

2.1. Circuite logice secvențiale

Circuitele logice secvențiale, pe scurt CLS, sunt circuite de comutare la care starea ieșirilor la un moment dat nu depinde doar de starea intrărilor, așa cum este cazul la circuitele logice combinaționale, ci și de stările anterioare ale acestora. Din acest motiv ele, acestea trebuie să aibă memorie în care să se rețină informații despre starea anterioară. Existența memoriei face ca evoluția în timp a acestor circuite să fie definită printr-o succesiune de stări interne în care se poate afla circuitul. Modelul general sau schema bloc generală pentru aceste circuite este dată în figura 2.1.1.

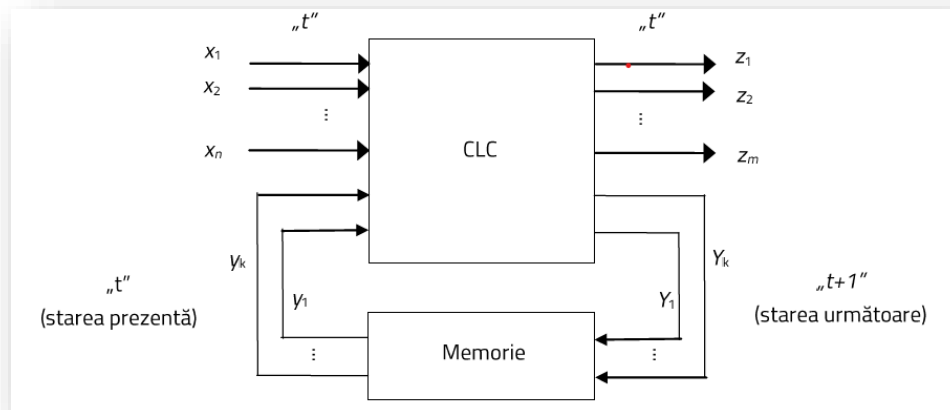


fig.2.1.1

CLS-urile pot fi descrise prin următoarele mulțimi:

- $X = \{x_1, x_2, \dots, x_n\}$ -> setul variabilelor de intrare
- $Z = \{z_1, z_2, \dots, z_m\}$ -> setul variabilelor de ieșire
- $Y = \{y_1, y_2, \dots, y_k\}$ -> setul variabilelor de stare

Se poate aprecia că evoluția oricărui CLS poate fi complet descrisă prin relații între intrări, starea prezentă, starea următoare și ieșiri.

2.2. Automate de tipul Mealy

Funcția f este funcția de tranziție a intrărilor, iar g este funcția ieșirilor, iar cele 2 funcții, f și g , descriu funcționarea automatelor de tipul Mealy.

- $f: Y \times X \rightarrow Y$
- $g: Y \times X \rightarrow Z$

S-au păstrat notațiile de mai sus pentru mulțimi.

Automatele de tipul Mealy se diferențiază de cele de tipul Moore prin faptul că funcțiile ieșirilor depind atât de starea prezentă, cât și de intrările prezente; în schimb, funcțiile de tranziție a intrărilor depind de aceleași argumente: stare și intrări.

3. Metoda Huffman

Sinteza CLS se poate face doar atunci când se cunoaște corespondența dintre secvența semnalelor de intrare și secvența semnale de ieșire a circuitului ce urmează a fi proiectat.

Metoda Huffman este o metodă de sinteză a circuitelor secvențiale asincrone care se folosește în cazul în care nu sunt folosite un număr foarte mare de intrări și stări, în caz contrar se utilizează metoda organigramei. Metoda Huffman se mai numește și metoda matriceală și are următorul algoritm:

3.1. Etapă 1

Descrierea completă a funcționării automatului și stabilirea stărilor în care acesta se poate afla. În acest scop se folosesc datele din enunțul problemei de sinteză, date care trebuie să reflecte corespondența dintre intrările și ieșirile circuitului printr-un ciclu de funcționare complet. În cele mai multe cazuri întâlnite în practică, descrierea funcționării se face fie prin text, fie prin diagrame de

semnal. În cazul diagramei de semnal, aceasta trebuie să conțină toate secvențele posibile ale corespondenței intrare-ieșire deoarece, în caz contrar, interpretarea acestor prime informații ar conduce la structuri fizice incomplete sau cu o funcționare modificată față de cea dorită. Stările se determină în așa fel încât să se țină cont de toate situațiile enunțate, condițiile avute în vedere trebuie să fie clare, necontradictorii și să nu se repete. Etapa se mai numește și etapa de stabilire primară a stărilor.

Conform enunțului temei, automatul ce trebuie proiectat are 2 intrări u_1 și u_2 , 8 stări s_1, s_2, \dots, s_8 care descriu un ciclu de funcționare complet și 2 ieșiri y_1 și y_2 . Automatul asincron este de tip Mealy fiindcă ieșirile depind atât de stări, cât și de intrări, așa cum este descris în protocolul de funcționare, și anume dacă se aplică mai întâi u_1 , iar $u_2 = 0$, atunci y_1 trece în 1 logic. y_1 trece în 0 logic atunci când se aplică și u_2 ; dacă se aplică mai întâi u_2 , iar $u_1 = 0$, atunci y_2 trece în 1 logic. y_2 trece în 0 logic când se aplică și u_1 .

Observație: după starea s_1 poate urma doar starea s_5 sau s_2 (depinde de intrarea aplicată prima).

În continuare, se va scrie funcția de tranziție $f(s, u_1 u_2)$, funcție rezultată din condițiile problemei.

$f(s_1, 00) = s_1$	$f(s_1, 01) = *$	$f(s_1, 11) = *$	$f(s_1, 10) = s_2$
$f(s_2, 00) = s_3$	$f(s_2, 01) = *$	$f(s_2, 11) = *$	$f(s_2, 10) = s_2$
$f(s_3, 00) = s_3$	$f(s_3, 01) = s_4$	$f(s_3, 11) = *$	$f(s_3, 10) = s_2$
$f(s_4, 00) = s_3$	$f(s_4, 01) = s_4$	$f(s_4, 11) = *$	$f(s_4, 10) = *$
$f(s_5, 00) = s_6$	$f(s_5, 01) = s_5$	$f(s_5, 11) = *$	$f(s_5, 10) = *$
$f(s_6, 00) = s_6$	$f(s_6, 01) = s_5$	$f(s_6, 11) = *$	$f(s_6, 10) = s_7$
$f(s_7, 00) = s_6$	$f(s_7, 01) = *$	$f(s_7, 11) = *$	$f(s_7, 10) = s_7$
$f(s_8, 00) = *$	$f(s_8, 01) = *$	$f(s_8, 11) = *$	$f(s_8, 10) = *$

Funcția de ieșire va fi:

$$g(s_1, u_1 u_2) = (0, 0)$$

$$g(s_2, u_1 u_2) = (1, 0)$$

$$g(s_3, u_1 u_2) = (1,0)$$

$$g(s_4, u_1 u_2) = (0,0)$$

$$g(s_5, u_1 u_2) = (0,1)$$

$$g(s_6, u_1 u_2) = (0,1)$$

$$g(s_7, u_1 u_2) = (0,0)$$

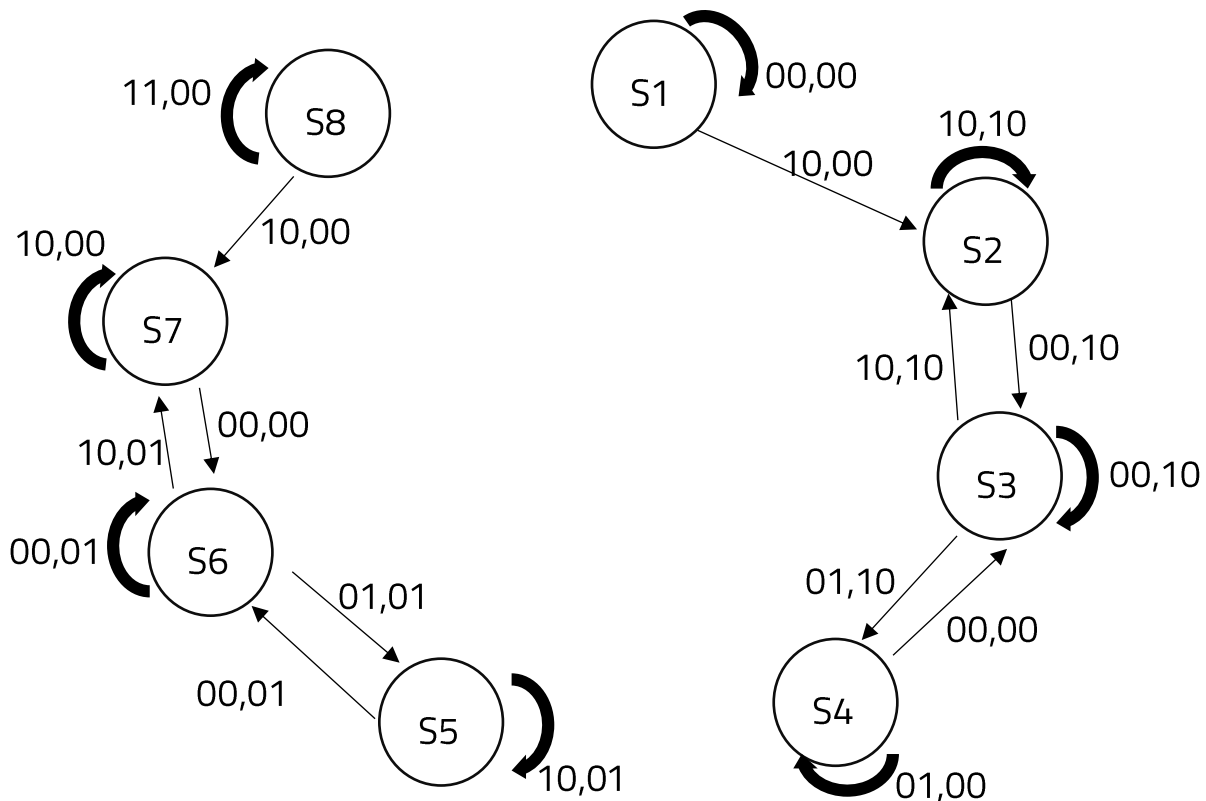
$$g(s_8, u_1 u_2) = (0,0)$$

Pentru y_1 automatul are ieșire doar în s_2 și s_3 , iar pentru y_2 are în s_5 și s_6 .

3.2. Etapa 2

Determinarea matricei sau tabelului primitive a stărilor. În condițiile impuse și având în vedere stările stabilite la etapa precedentă, se întocmește graful de tranziție și apoi tabela/matricea stărilor. În multe situații, sinteza pornește direct de la graful de tranziție. După întocmirea tabelului stărilor care, în acest caz, se numește tabela/matricea primară/primitivă a stărilor, în acesta se definesc stările stabile și instabile ale automatului.

Graful de tranziție va arăta în felul următor:



Matricea primitivă a starilor este următoarea:

$s_k \backslash u_1 u_2$	00	01	11	10	y_1	y_2
s_1	s_1	-	-	s_2	0	0
s_2	s_3	-	-	s_2	1	0
s_3	s_3	s_4	-	s_2	1	0
s_4	s_3	s_4	-	-	0	0
s_5	s_6	s_5	-	-	0	1
s_6	s_6	s_5	-	s_7	0	1
s_7	s_6	-	-	s_7	0	0
s_8	-	-	s_8	s_7	0	0

Cu "-" au fost notate stările nedefinite, iar stările stabile au fost evidențiate cu galben.

Matricea se numește

- primară deoarece este prima matrice concepută
- primitivă deoarece conține toate stările

O caracteristică a matricei primitive este aceea că pe fiecare linie există doar o singură stare stabilă.

Matricei primitive trebuie să i se atașeze întotdeauna matricea completă a ieșirilor. Dimensiunea acestei matrici este aceeași cu a matricei primitive a stărilor. Această matrice se completează în felul următor – în primul rând se completează ieșirile stărilor stabile, iar apoi se completează ieșirile stărilor instabile, astfel:

- Situația în care stările între care are loc tranziția au aceeași ieșire. În această situație este absolut obligatoriu ca și starea instabilă să aibă aceeași ieșire. O proiectare corectă a automatului secvențial presupune, neapărat, respectarea acestei condiții.
- Când stările între care are loc tranziția au ieșiri diferite: când tranziția are loc la sfârșitul intervalului de tranziție, ieșirea stării instabile va fi ieșirea stării s_i ; când tranziția are loc la începutul intervalului de tranziție, atunci ieșirea stării instabile va fi ieșirea stării s_j ; când nu se cunoaște momentul tranziției (sau nu ne interesează acest lucru) ieșirea stării instabile este

indiferentă. În tabelele următoare vom nota cu „*” această ieșire, tocmai pentru a arăta apartenența acesteia.

		y_1			
$s_k \backslash u_1 u_2$		00	01	11	10
s_1		0	-	-	*
s_2		1	-	-	1
s_3		1	*	-	1
s_4		*	0	-	-
s_5		0	0	-	-
s_6		0	0	-	0
s_7		0	-	-	0
s_8		-	-	0	0

		y_2			
$s_k \backslash u_1 u_2$		00	01	11	10
s_1		0	-	-	0
s_2		0	-	-	0
s_3		0	0	-	0
s_4		0	0	-	-
s_5		1	1	-	-
s_6		1	1	-	*
s_7		*	-	-	0
s_8		-	-	0	0

3.3. Etapa 3

Reducerea numărului de stări ale matricei primitive și întocmirea matricei reduse a stărilor; în această etapă se elimină stările redundante sau echivalente și apoi se minimizează numărul de stări ale automatului. Astfel se obține tabelul redus al stărilor prin unificarea sau fuzionarea stărilor echivalente sau compatibile. Această etapă nu este obligatorie pentru o sinteză ce urmărește doar îndeplinirea unor condiții de funcționare impuse. Ea devine însă obligatorie în momentul în care se pune problema prețului de cost al automatului secvențial, deci găsirea unei configurații minime. În

concluzie, reducerea numărului de stări primare sau primitive se obține prin alipirea sau fuzionarea unui număr de stări după anumite reguli; se vor obține astfel o matrice cu un număr redus de stări care reflectă însă complet funcționarea unui automat secvențial echivalent.

Regula după care se pot alipi 2 linii este următoarea: 2 linii ale matricei primitive caracterizate de i și j se pot alipi și în felul acesta se poate obține o formă redusă dacă tranzițiile din stările stabile i și j ale celor 2 linii conduc prin aplicarea aceleiași secvențe de intrări la softarea unică k . Alipirea sau fuzionarea trebuie să țină cont și de un criteriu suplimentar, acela al concordanței ieșirilor. Rezultă că pentru a alipi conform definiției 2 stări ale matricei primitive caracterizate de stările stabile i și j , trebuie ca pe fiecare coloană a matricei primitive, în locațiile corespunzătoare, să fie una dintre următoarele combinații.

"i"	i	j	i	j	k	i	-	i	-	k	-
"j"	i	j	i	j	k	-	j	-	j	-	-
	i	j	i	j	k	i	j	i	j	k	-

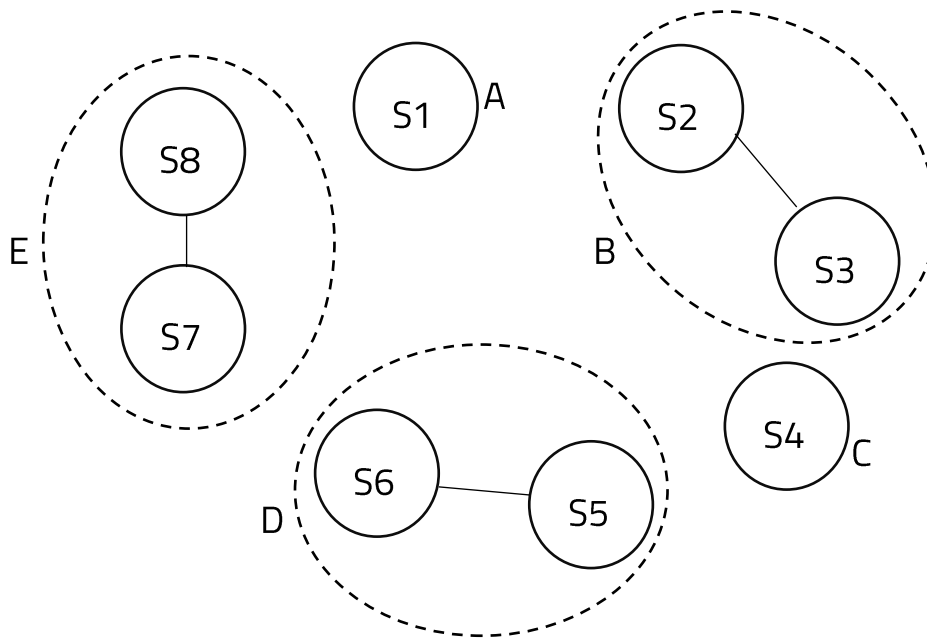
- "i" și "j" se referă la s_i și s_j .
- "i" și "j" înseamnă "i" stabil și "j" stabil.

Pentru a se putea pune în evidență toate alipirile posibile între stările automatului se construiește poligonul alipirilor. În nodurile acestuia se trec stările nereduse, iar pe segmente de dreaptă se vor marca alipirile posibile. După întocmirea poligonului trebuie aleasă soluția de fuzionare optimă. În acest sens, dacă în cazul general k – stări nereduse formează, în poligonul alipirilor, un contur poligonal complet (interior sau exterior), rezultă că cele k stări nereduse se pot alipi obținându-se o singură stare echivalentă. În cazul în care prin alipire s-a obținut din numărul total de stări nereduse singură stare redusă, circuitul logic secvențial degenerază într-unul combinațional.

Atât pentru prima ieșire, cât și pentru cea de-a doua, se va alege următoarea varianta de reducere a stărilor, și anume se vor reduce 2 câte 2, mai puțin s_1 și s_4 .

$A(s_1)$, $B(s_2, s_3)$, $C(s_4)$, $D(s_5, s_6)$, $E(s_7, s_8)$

Poligonul alipirilor va fi următorul:



Concordanța ieșirilor este respectată pentru ambele cazuri.

Prin fuziunea a 2 linii se respectă:

- prioritate are starea "cea mai puternică"
- (opțional) se păstrează indicele cel mai al stărilor fuzionate

Prin urmare, matricea redusă a stărilor vă arăta în felul următor.

$s_k \backslash u_1 u_2$	00	01	11	10
$A(s_1)$	s_1	-	-	s_2
$B(s_2, s_3)$	s_3	s_4	-	s_2
$C(s_4)$	s_3	s_4	-	-
$D(s_5, s_6)$	s_6	s_5	-	s_7
$E(s_7, s_8)$	s_6	-	s_8	s_7

Astfel am redus astfel numărul stărilor de 8 la 5 prin alipirea stării 2 cu starea 3, stării 5 cu starea 6 și stării 7 cu starea 8.

Matricele reduse ale ieșirilor:

		y_1			
$s_k \backslash u_1 u_2$		00	01	11	10
$A(s_1)$		0	-	-	-
$B(s_2, s_3)$		1	-	-	1
$C(s_4)$		-	0	-	-
$D(s_5, s_6)$		0	0	-	0
$E(s_7, s_8)$		0	-	0	0

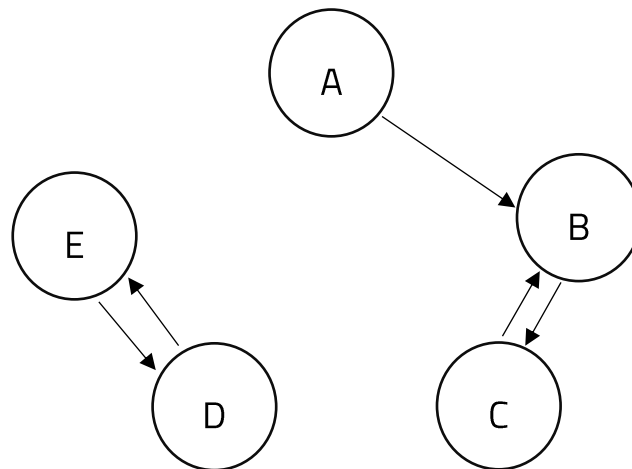
		y_2			
$s_k \backslash u_1 u_2$		00	01	11	10
$A(s_1)$		0	-	-	0
$B(s_2, s_3)$		0	0	-	0
$C(s_4)$		0	0	-	-
$D(s_5, s_6)$		1	1	-	-
$E(s_7, s_8)$		-	-	0	0

3.4. Etapa 4

Codificarea stărilor. În această etapă se determină mai întâi numărul de variabile secundare de stare și se realizează asigurarea sau alocarea stărilor. Această etapă se mai numește și etapa alocării variabilelor secundare de stare; trebuie făcută cu mare atenție pentru a se evita fenomenul de hazard de tranziție determinat de variabilele secundare de stare. Etapa creează premisele obținerii funcțiilor de excitație ale automatului. Ea se referă de fapt la codificarea univocă a stărilor fuzionate ale automatului redus echivalent descris de matricea redusă și codificată a stărilor și ieșirilor.

În cazul general, pentru q stări reduse sunt necesare p variabile de stare $2^p > q$. Codificarea stărilor este o etapă foarte importantă deoarece trebuie evitată apariția hazardului de tranziție. Pentru a se evita apariția acestui fenomen în funcționarea circuitului este necesară să fie interzise tranzițiile care să presupună modificarea la un moment dat a mai mult decât o singură variabilă de stare. În felul acesta se evită apariția curselor critice care pot determina apariția unor tranziții de stare incorecte și semnale de ieșire false. Regula este următoarea:

Codificarea stărilor trebuie făcută în așa fel încât toate tranzițiile să aibă loc numai între stări codificate adiacent. Din acest motiv trebuie construit așa numitul poligon al tranzițiilor. Nodurile acestui poligon reprezintă stările stabile fuzionate, iar laturile orientate ca sens reprezintă tranziții între stările alipite, deci poligonul tranzițiilor va fi următorul.



Se adoptă următoarea variantă de codificare unde toate stările reduse sunt codificate adiacent.

s_r \ cod	c_1	c_2	c_3
A	0	0	0
B	0	0	1
C	0	1	1
D	1	1	1
E	1	0	1

Am preferat să folosim o codificare unde fiecare stare redusă să fie adiacentă următoarei stări prin doar 2 biți, astfel matricea redusă și codificată a stărilor va deveni:

$c_1 c_2 c_3 \backslash u_1 u_2$	00	01	11	10
000	s_1	-	-	s_2
001	s_3	s_4	-	s_2
011	s_3	s_4	-	-
111	s_6	s_5	-	s_7
101	s_6	-	s_8	s_7
010	-	-	-	-
100	-	-	-	-
110	-	-	-	-

Matricele reduse și codificate ale ieșirilor vor deveni la rândul lor după cum urmează.

y_1

$c_1 c_2 c_3 \backslash u_1 u_2$	00	01	11	10
000	0	-	-	-
001	1	-	-	1
011	-	0	-	-
111	0	0	-	0
101	0	-	0	0
010	-	-	-	-
100	-	-	-	-
110	-	-	-	-

y_2

$c_1 c_2 c_3 \backslash u_1 u_2$	00	01	11	10
000	0	-	-	0
001	0	0	-	0
011	0	0	-	-
111	1	1	-	-
101	-	-	0	0
010	-	-	-	-
100	-	-	-	-
110	-	-	-	-

3.5. Etapa 5

Determinarea matricei tranzițiilor stărilor și obținerea funcțiilor de excitație ale automatului.

Matricea tranzițiilor stărilor reduse se obține direct din matricea redusă a stărilor și permite obținerea excitațiilor pentru secțiunea de memorie a automatului, deci, în această etapă se vor determina funcțiile de control sau de excitație pentru automate elementare și, prin urmare, ale reacțiilor schemei secvențiale. În cazul circuitelor basculante bistabile există notații specifice, deci se vor face următoarele notații: $c_1 = Q_1^t$, $c_2 = Q_2^t$, $c_3 = Q_3^t$, $C1 = Q_1^{t+1}$, $C2 = Q_2^{t+1}$, $C3 = Q_3^{t+1}$. Va rezulta tabelul de codificare:

s_r \ cod	Q_1^t	Q_2^t	Q_3^t
A	0	0	0
B	0	0	1
C	0	1	1
D	1	1	1
E	1	0	1

Astfel, matricea tranzițiilor stărilor sau diagrama de excitație va deveni:

$Q_1^t Q_2^t Q_3^t$ \ $u_1 u_2$	00	01	11	10
000	000	-	-	001
001	001	011	-	001
011	001	011	-	-
111	111	111	-	101
101	111	-	101	101
010	-	-	-	-
100	-	-	-	-
110	-	-	-	-

Observație: tabelul s-a completat și cu celelalte combinații de 3 biți care sunt indiferente.

Circuitul basculant bistabil, pe scurt CBB, de tip D are, în general, următoarele proprietăți:

- ecuația după care funcționează este $Q^{t+1} = D^t$, unde D reprezintă intrarea de date la momentul t, iar Q^{t+1} este starea la momentul următor lui t
- tabelul de funcționare:

$D^t Q^t$	Q^{t+1}
00	0
01	0
10	1
11	1

După cum se observă tabelul respectă ecuația după care funcționează bistabilul de tip D.

- tabelul de funcționare în formă restrânsă:

D^t	Q^{t+1}
0	0
1	1

- tabelul de excitație:

$Q^t Q^{t+1}$	D^t
00	0
01	1
10	0
11	1

Particularizând proprietățile bistabilului de tip D pentru matricea tranzițiilor stărilor, funcțiile de excitație vor deveni:

$Q_1^t Q_2^t Q_3^t \backslash u_1 u_2$	00	01	11	10
000	0	-	-	0
001	0	0	-	0
011	0	0	-	-
111	1	1	-	1
101	1	-	1	1
010	-	-	-	-
100	-	-	-	-
110	-	-	-	-

$$C1 = Q_1^t Q_3^t (= Q_1^{t+1})$$

$Q_1^t Q_2^t Q_3^t \backslash u_1 u_2$	00	01	11	10
000	0	-	-	0
001	0	1	-	0
011	0	1	-	-
111	1	1	-	0
101	1	-	0	0
010	-	-	-	-
100	-	-	-	-
110	-	-	-	-

$$C2 = \bar{u}_1 u_2 + \bar{u}_1 \bar{u}_2 (= Q_2^{t+1})$$

$Q_1^t Q_2^t Q_3^t \backslash u_1 u_2$	00	01	11	10
000	0	-	-	1
001	1	1	-	1
011	1	1	-	-
111	1	1	-	1
101	1	-	1	1
010	-	-	-	-
100	-	-	-	-
110	-	-	-	-

$$C3 = u_2 + u_1 \bar{u}_2 + Q_3^t \bar{u}_2 (= Q_3^{t+1})$$

3.6. Etapa 6

Determinarea matricelor ieșirilor și obținerea funcțiilor de ieșire ale schemei

Matricele ieșirilor se obțin din matricea redusă și codificată a ieșirilor, pentru fiecare ieșire redusă se înlocuiește codul ei.

ieșirea y_1

$Q_1^t Q_2^t Q_3^t \backslash u_1 u_2$	00	01	11	10
000	0	-	-	-
001	1	-	-	1
011	-	0	-	-
111	0	0	-	0
101	0	-	0	0
010	-	-	-	-
100	-	-	-	-
110	-	-	-	-

$$Y_1 = \bar{Q}_1^t \bar{Q}_2^t Q_3^t$$

ieșirea y_2

$Q_1^t Q_2^t Q_3^t \backslash u_1 u_2$	00	01	11	10
000	0	-	-	0
001	0	0	-	0
011	0	0	-	-
111	1	1	-	-
101	-	-	0	0
010	-	-	-	-
100	-	-	-	-
110	-	-	-	-

$$Y_2 = Q_1^t Q_2^t Q_3^t$$

3.7.Etapa 7

Implementarea funcțiilor de excitație și a funcțiilor de ieșire obținute la etapele precedente, mai precis la etapa 6 și etapa 7. Această etapă va avea în vedere circuitele fizice ce vor fi utilizate în implementare.

În acest proiect se vor utiliza bistabili de tip D pentru implementare secțiunii de memorie și porți logice pentru structura logică combinațională, mai exact porți de tipul NAND (ȘI-NU). Pentru a putea utiliza schema cu porți logice NAND, se vor prelucra relațiile obținute astfel încât operațiile logice să fie compatibile cu tipul de poartă logică utilizată în circuitul NAND, și anume se va nega de două ori, folosindu-se relația lui De Morgan.

$$C1 = Q_1^{t+1} = Q_1^t Q_3^t = \overline{\overline{C1}} = \overline{Q_1^t Q_3^t}, \text{ aici se va folosi o poartă NAND pe post de inversor}$$

$$C2 = Q_2^{t+1} = \bar{u}_1 u_2 + \bar{u}_1 \bar{u}_2 = \overline{\overline{C2}} = \overline{\overline{u_1 u_2} + \overline{u_1 u_2}} = \overline{\overline{u_1 u_2} \overline{u_1 u_2}}$$

$$C3 = Q_3^{t+1} = u_2 + u_1 \bar{u}_2 + \bar{u}_2 Q_3^t = \overline{\overline{C3}} = \overline{u_2 + u_1 \bar{u}_2 + \bar{u}_2 Q_3^t} = \overline{\overline{u_2} \overline{u_1 \bar{u}_2} \overline{\bar{u}_2 Q_3^t}}$$

$$Y_1 = \bar{Q}_1^t \bar{Q}_2^t Q_3^t = \bar{Y}_1 = \overline{\overline{Q}_1^t \bar{Q}_2^t Q_3^t}, \text{ caz similiar ca la C1}$$

$$Y_2 = Q_1^t Q_2^t Q_3^t = \bar{Y}_2 = \overline{Q_1^t Q_2^t Q_3^t}, \text{ caz similiar ca la C1}$$

Pentru $C1$, Y_1 și Y_2 se vor folosi după formarea termenilor, așa cum s-a menționat mai sus, o poartă NAND cu 2 intrări pentru a avea ieșirea dorită, altfel am avea termenul inițial negat doar o singură dată. Acest caz este similar cu negarea unui intrări unde se folosește o poartă NAND cu 2 intrări cu rol de inversor pentru a minimiza costul și a nu folosi integrate adiționale. Pentru a nega propriu-zis intrarea se aplică semnalul de intrare la un port și 1 logic la celălalt port, astfel la ieșirea porții NAND se obține intrarea negată.

În continuare se va prezenta implementarea funcțiilor de excitație și funcțiilor de ieșire, figura 3.7.1.

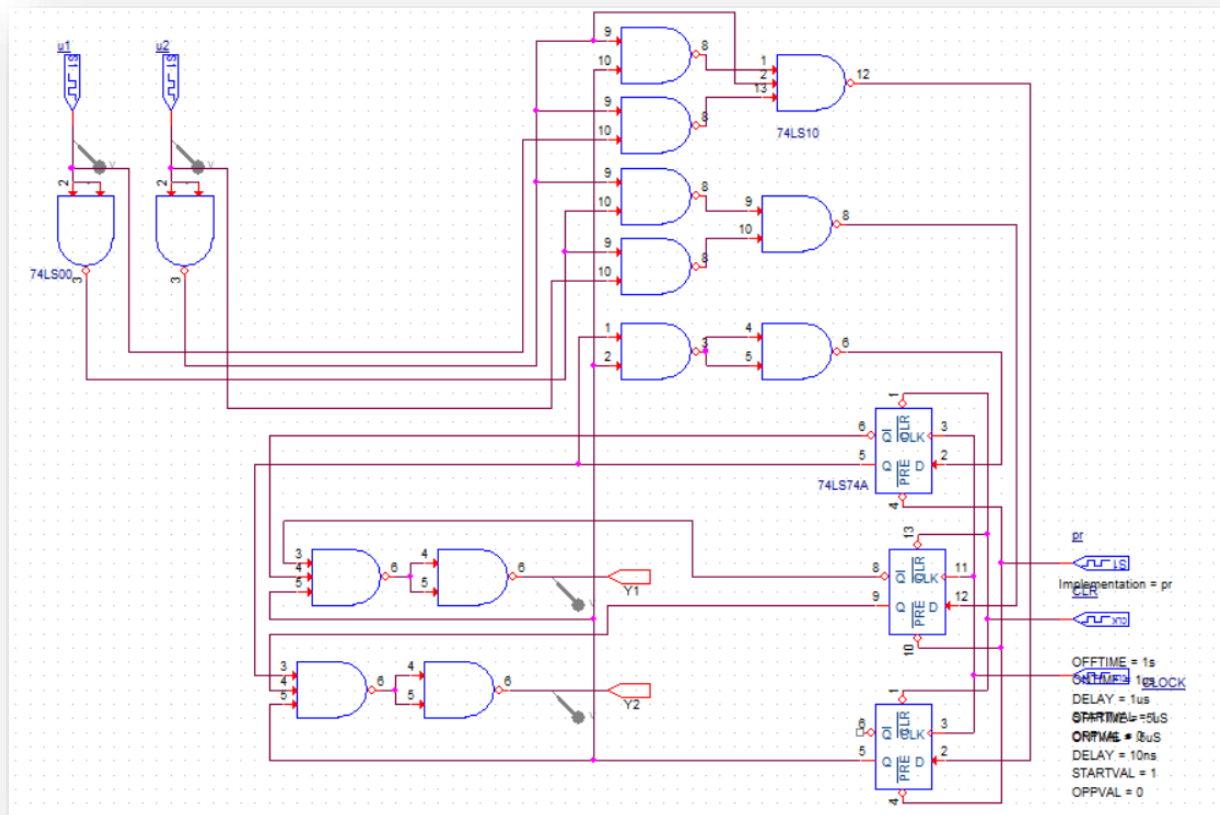


fig.3.7.1

Pentru implementare s-au folosit următoarele circuite integrate:

- 4 x 74LS00
- 74LS10
- 2 x 74LS74(-1)

Anexa A de la final arată cum sunt dispuse porțile și bistabilele în circuitele integrate.

3.8. Etapa 8

Analiza schemelor obținute. Odată implementată schema se trece la testarea funcționării acesteia. Se va analiza secvența de funcționare conform diagramei "intrări-ieșire" prezentată la început în cerința proiectului. Operația de analiză presupune identificarea unor situații de avarii; de-asemeni, rolul acestei analize este, în plus, și acela de a descoperi eventuale simplificări ce se pot aduce schemei

logice, cât și reorganizarea acestora dacă se găsesc structuri folosite în comun. La final, în cursul analizei trebuie rezolvată și problema adaptării schemei secvenței de comandă introdusă la diverse alte opțiuni de comutare. Figura 3.8.1. prezintă analiza schemei în OrCAD.

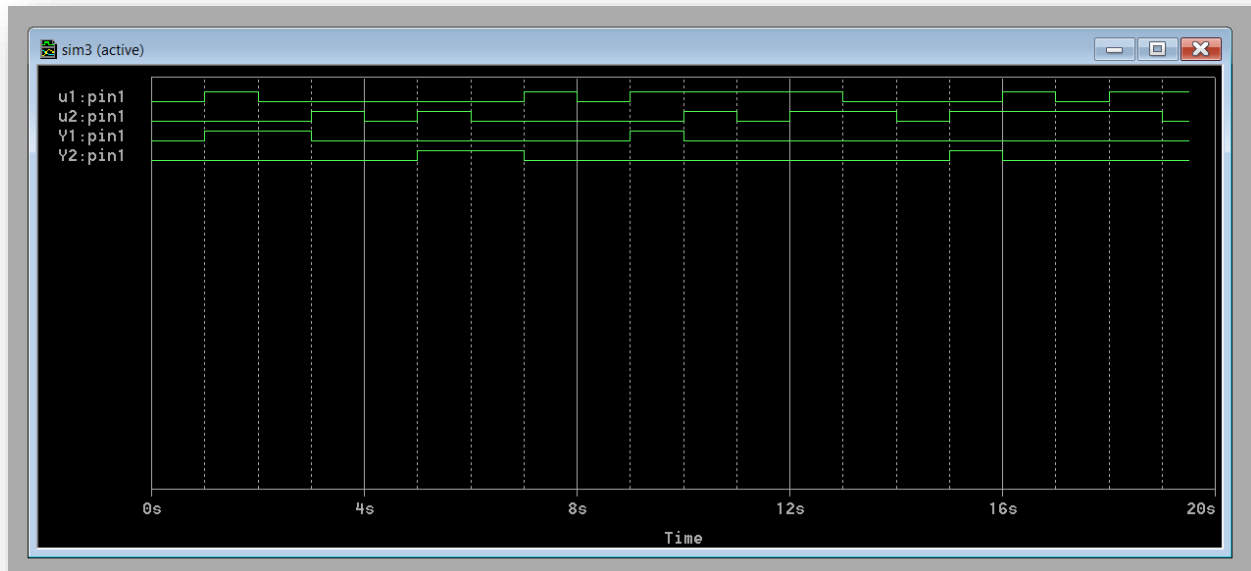


fig.3.8.1

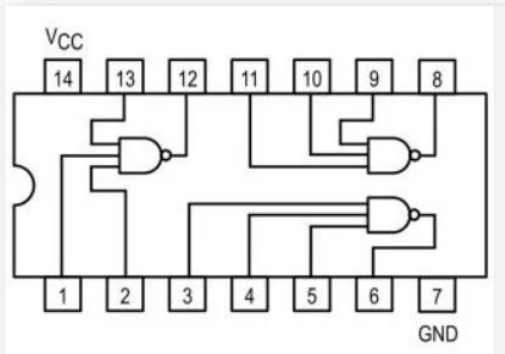
4. Bibliografie

- Moldoveanu F., Floroian D. – Circuite logice și comenzi secvențiale. Circuite logice combinaționale, Ed. Universității Transilvania din Brașov, 2003.
- Toacșe Gh., Nicula D. – Electronică digitală, Ed. Tehnică, București, 2005.
- Wilkinson B. – Electronică digitală. Bazele proiectării, Ed. Teora, București, 2002.
- Wakerly J. F. – Circuite digitale, Ed. Teora, București, 2002.
- Ștefan Gh. M., Bistriceanu V. – Circuite integrate digitale. Probleme. Proiectare, Ed. Albastră, Cluj-Napoca. 2000.
- Maican S. – Sisteme numerice cu circuite integrate. Culegere de probleme, Ed. Tehnică, București, 1980

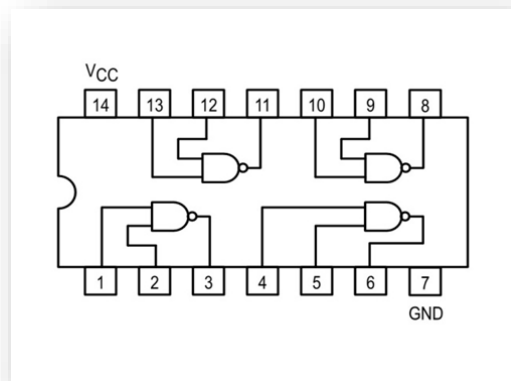
Datasheet-uri:

- <https://www.alldatasheet.com/datasheet-pdf/pdf/15541/PHILIPS/74HC154.html>
- <https://www.alldatasheet.com/datasheet-pdf/pdf/125043/ETC1/74S00.html>
- <https://www.princeton.edu/~mae412/HANDOUTS/Datasheets/74S04.PDF>
- <https://www.alldatasheet.com/datasheet-pdf/pdf/15577/PHILIPS/74HC27.html>
- <https://www.alldatasheet.com/datasheet-pdf/pdf/15563/PHILIPS/74HC21.html>
- <https://www.alldatasheet.com/datasheet-pdf/pdf/5666/MOTOROLA/74LS155.html>
- <https://www.alldatasheet.com/datasheet-pdf/pdf/51038/FAIRCHILD/74LS154.html>
- <https://www.alldatasheet.com/datasheet-pdf/pdf/5661/MOTOROLA/74LS151.html>
- <https://www.alldatasheet.com/datasheet-pdf/pdf/5707/MOTOROLA/74LS32.html>
- <https://www.alldatasheet.com/datasheet-pdf/pdf/5681/MOTOROLA/74LS20.html>
- <https://www.alldatasheet.com/datasheet-pdf/pdf/12605/ONSEMI/74LS10.html>

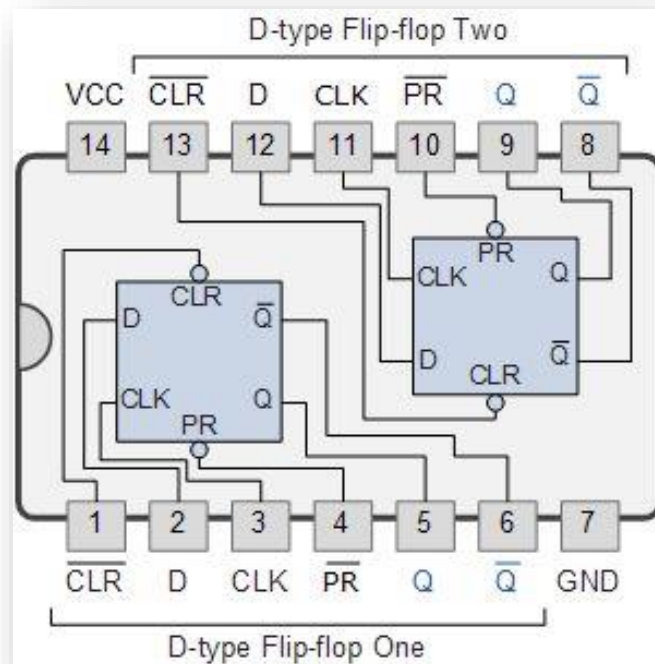
5. Anexa A



CI 74LS10



CI 74LS00



CI 74LS74