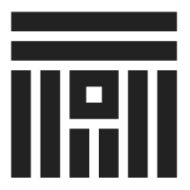
**Universitatea *Transilvania* din Brașov**

**Facultatea de Inginerie Electrică și Știința Calculatoarelor**

**Departamentul Automatică și Informatică**  

PROIECT

ANALIZA ȘI SINTEZA CIRCUITELOR NUMERICE II

Echipă:

Pătrăhau Cătălin Ionuț

Grabovenco Bogdan–Iulian

Grupe:

AIA4LF413

AIA4LF412

Îndrumător:

Prof. Dr. Ing. Moldoveanu Florin Dumitru

Brașov 2024

Cuprins

1. Tema de proiect3
2. Noțiuni introductive5
   1. Circuite logice secvențiale5
   2. Automate de tipul Mealy6
3. Metoda Huffman6
   1. Etapa 1 – funcția de tranziție și funcția de ieșire6
   2. Etapa 2 – graful de tranziție și matricea primitivă a stărilor 8
   3. Etapa 3 – reducerea stărilor10
   4. Etapa 4 – codificarea stărilor13
   5. Etapa 5 – matricea de tranziție a stărilor și funcțiile de excitație16
   6. Etapa 6 – matricele ieșirilor și funcțiile de ieșire19
   7. Etapa 7 – implementarea funcțiilor de excitație și a funcțiilor de ieșire20
   8. Etapa 8 – analiza schemei21
4. Bibliografie23
5. Anexa A24
6. 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, u1 și u2, și două ieșiri y1 și y2. Dacă se aplică mai întâi intrarea u1, ieșirea y1 trece în 1 logic (y2 rămîne în 0 logic). Ieșirea y1 își menține valoarea 1 chiar dacă intrarea u1 trece în 0 logic, și devine 0 numai când se aplică u2. Dacă, după aplicarea lui u1 (u1 = 1) se aplică și u2 (u1 u2 = 11), ambele ieșiri trec în 0 și rămân în 0 până când ambele intrări devin 0. Dacă se aplică mai întâi u2 (din starea inițială, când u1 = u2 = y1 = y2 = 0), y2 ia valoarea 1 (y1 = 0) și se menține la această valoare chiar dacă u2 trece în 0. Ieșirea y2 trece în 0 dacă, după aplicarea lui u2 (u2 = 1) se aplică și u1 (u1 u2 = 11), sau dacă, după ce u2 a devenit 0, se aplică u1.

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

Diagrama de semnale „intrări-ieșiri” este prezentată în figura 1.1.O imagine care conține schiță, diagramă, linie, Plan

Descriere generată automat

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 marticei 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 şi gradul de utilizare al fiecărui circuit integrat.

**Bibliografia recomandată:**

[1] Ştefan, Gh. – Circuite și sisteme digitale, Ed. Tehnică, București, 2000.

[2] Wakerly, J.F. – Circuite digitale, Ed. Teora, Bucureşti, 2002.

[3] Wilkinson, B. – Electronică digitală. Bazele proiectării, Ed. Teora, Bucureşti, 2002.

[4] Mano, M.M. – Digital Design, Prentice Hall International, London, 2002.

[5] Moldoveanu, F., Floroian, D. – Circuite logice şi comenzi secvențiale. Circuite logice combinaționale, Ed. Universității Transilvania din Braşov, 2003.

[6] Toacşe, Gh., Nicula, D. – Electronică digital , vol. I, Ed. Tehnică, Bucureşti, 2005.

1. Noțiuni introductive
   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.

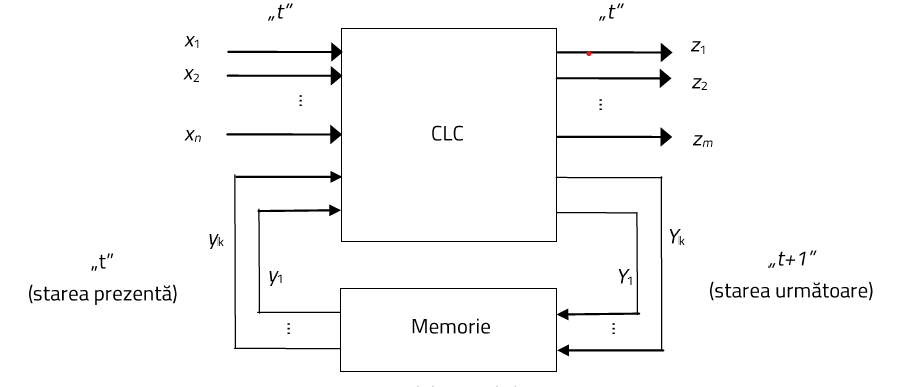


fig.2.1.1

CLS-urile pot fi descrise prin următoarele mulțimi:

* -> setul variabilelor de intrare
* -> setul variabilelor de ieșire
* -> setul variabilelor de stare

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

* 1. 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.

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.

1. 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:

* 1. Etapa 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 acestorr 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  și , 8 stări care descriu un ciclu de funcționare complet și 2 ieșiri și . 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 , iar , atunci trece în 1 logic. trece în 0 logic atunci când se aplică și ; dacă se aplică mai întâi , iar , atunci trece în 1 logic. trece în 0 logic când se aplică și .

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

În continuare, se va scrie funcția de tranziție , funcție rezultată din condițiile problemei.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Funcția de ieșire va fi:

Pentru automatul are ieșire doar în  și , iar pentru are în și .

* 1. Etapa 2

Determinarea matricei sau tabelei 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 tabelei 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:

00,00

11,00

10,10

10,00

10,00

10,00

00,10

10,10

00,00

10,01

00,01

00,10

01,01

01,10

00,01

00,00

01,00

10,01

Matricea primitivă a starilor este următoarea:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 00 | 01 | 11 | 10 |  |  |
|  |  | - | - |  | 0 | 0 |
|  |  | - | - |  | 1 | 0 |
|  |  |  | - |  | 1 | 0 |
|  |  |  | - | - | 0 | 0 |
|  |  |  | - | - | 0 | 1 |
|  |  |  | - |  | 0 | 1 |
|  |  | - | - |  | 0 | 0 |
|  | - | - |  |  | 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 ataşeze întotdeauna matricea completă a ieşirilor. Dimensiunea acestei matrici este aceeaşi cu a matricei primitive a stărilor. Această matrice se completează în felul următor – în primul rând se completează ieşirile stărilor stabile, iar apoi se completează ieşirile stărilor instabile, astfel:

* Situația în care stările între care are loc tranziția au aceeaşi ieşire. În această situație este absolut obligatoriu ca şi starea instabilă să aibă aceeaşi ieşire. O proiectare corectă a automatului secvențial presupune, neapărat, respectarea acestei condiții.
* Când stările între care are loc tranziția au ieşiri diferite: când tranziția are loc la sfârşitul intervalului de tranziție, ieşirea stării instabile va fi ieşirea stării ; când tranziția are loc la începutul intervalului de tranziție, atunci ieşirea stării instabile va fi ieşirea stării ; când nu se cunoaşte momentul tranziției (sau nu ne interesează acest lucru) ieşirea stării instabile este indiferentă. În tabelele următoare vom nota cu „\*” această ieşire, tocmai pentru a arăta apartenența acesteia.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 00 | 01 | 11 | 10 |
|  | 0 | - | - | \* |
|  | 1 | - | - |  |
|  | 1 | \* | - |  |
|  | \* | 0 | - | - |
|  | 0 | 0 | - | - |
|  | 0 | 0 | - | 0 |
|  | 0 | - | - | 0 |
|  | - | - | 0 | 0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 00 | 01 | 11 | 10 |
|  | 0 | - | - | 0 |
|  | 0 | - | - | 0 |
|  | 0 | 0 | - | 0 |
|  | 0 | 0 | - | - |
|  | 1 | 1 | - | - |
|  | 1 | 1 | - | \* |
|  | \* | - | - | 0 |
|  | - | - | 0 | 0 |

* 1. 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 și .
* “i” și ”j” înseamnă “i” stabil și “j” stabil.

Pentru a se putea pune în evidență toate alipirile posibile între stările automatului se construieşte 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 degenerează 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 și .

A(), B(,), C(), D(,), E(,)

Poligonul alipirilor va fi următorul:

A

B

E

C

D

Concordonța 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.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 00 | 01 | 11 | 10 |
| A() |  | - | - |  |
| B(,*)* |  |  | - |  |
| C() |  |  | - | - |
| D(,) |  |  | - |  |
| E(,) |  | - |  |  |

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:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 00 | 01 | 11 | 10 |
| A() | 0 | - | - | - |
| B(,) | 1 | - | - |  |
| C() | - | 0 | - | - |
| D(,) | 0 | 0 | - | 0 |
| E(,) | 0 | - | 0 | 0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 00 | 01 | 11 | 10 |
| A() | 0 | - | - | 0 |
| B(,) | 0 | 0 | - | 0 |
| C() | 0 | 0 | - | - |
| D(,) | 1 | 1 | - | - |
| E(,) | - | - | 0 | 0 |

* 1. Etapa 4

Codificarea stărilor. În această etapă se determină mai întâi numărul de variabile secundare de stare şi se realizează asigurarea sau alocarea stărilor. Această etapă se mai numeşte şi 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 şi codificată a stărilor şi ieşirilor.

În cazul general, pentru q stări reduse sunt necesare p variabile de stare > 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 şi semnale de ieşire false. Regula este următoarea:

Codificarea stărilor trebuie făcută în aşa fel încât toate tranzițiile să aibă loc numai între stări codificate adiacent. Din acest motiv trebuie construit aşa 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.

|  |  |  |  |
| --- | --- | --- | --- |
| cod |  |  |  |
| 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:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 00 | 01 | 11 | 10 |
| 000 |  | - | - |  |
| 001 |  |  | - |  |
| 011 |  |  | - | - |
| 111 |  |  | - |  |
| 101 |  | - |  |  |
| 010 | - | - | - | - |
| 100 | - | - | - | - |
| 110 | - | - | - | - |

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 00 | 01 | 11 | 10 |
| 000 | 0 | - | - | - |
| 001 | 1 | - | - |  |
| 011 | - | 0 | - | - |
| 111 | 0 | 0 | - | 0 |
| 101 | 0 | - | 0 | 0 |
| 010 | - | - | - | - |
| 100 | - | - | - | - |
| 110 | - | - | - | - |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 00 | 01 | 11 | 10 |
| 000 | 0 | - | - | 0 |
| 001 | 0 | 0 | - | 0 |
| 011 | 0 | 0 | - | - |
| 111 | 1 | 1 | - | - |
| 101 | - | - | 0 | 0 |
| 010 | - | - | - | - |
| 100 | - | - | - | - |
| 110 | - | - | - | - |

* 1. 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 acestă 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 următorele notații: =, =, = , ==, =. Va rezultă tabelul de codificare:

|  |  |  |  |
| --- | --- | --- | --- |
| cod |  |  |  |
| 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:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 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 celelate 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 = , unde D reprezintă intrarea de date la momentul t, iar este starea la momentul următor lui t
* tabelul de funcționare:

|  |  |
| --- | --- |
|  |  |
| 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:

|  |  |
| --- | --- |
|  |  |
| 0 | 0 |
| 1 | 1 |

* tabelul de excitație:

|  |  |
| --- | --- |
|  |  |
| 00 | 0 |
| 01 | 1 |
| 10 | 0 |
| 11 | 1 |

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 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 = ( = )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 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 = ( = )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 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 = ( = )

* 1. Etapa 6

Determinarea matricelor ieșirilor și obținerea funcțiilor de ieșire ale schemei

Matricele ieşirilor se obțin din matricea redusă şi codificată a ieşirilor, pentru fiecare ieşire redusă se înlocuieşte codul ei.

Ieșirea

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 00 | 01 | 11 | 10 |
| 000 | 0 | - | - | - |
| 001 | 1 | - | - | 1 |
| 011 | - | 0 | - | - |
| 111 | 0 | 0 | - | 0 |
| 101 | 0 | - | 0 | 0 |
| 010 | - | - | - | - |
| 100 | - | - | - | - |
| 110 | - | - | - | - |

=

Ieșirea

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 00 | 01 | 11 | 10 |
| 000 | 0 | - | - | 0 |
| 001 | 0 | 0 | - | 0 |
| 011 | 0 | 0 | - | - |
| 111 | 1 | 1 | - | - |
| 101 | - | - | 0 | 0 |
| 010 | - | - | - | - |
| 100 | - | - | - | - |
| 110 | - | - | - | - |

=

* 1. Etapa 7

Implementarea funcțiilor de excitație și a funcțiilor de ieșire obținute la etapele precendente, 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.

, aici se va folosi o poartă NAND pe post de inversor

, caz similiar ca la C1

, caz similiar ca la C1

Pentru și 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.

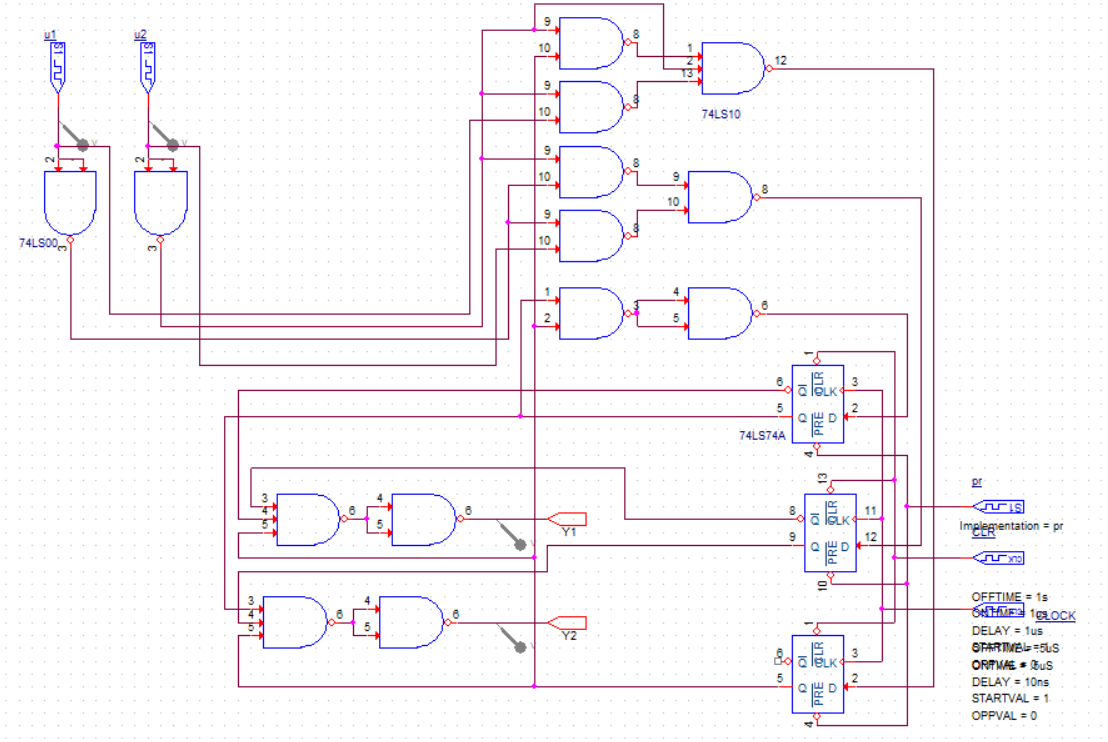


fig.3.7.1

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.

* 1. 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 acesteia dacă se găsesc structurii 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.

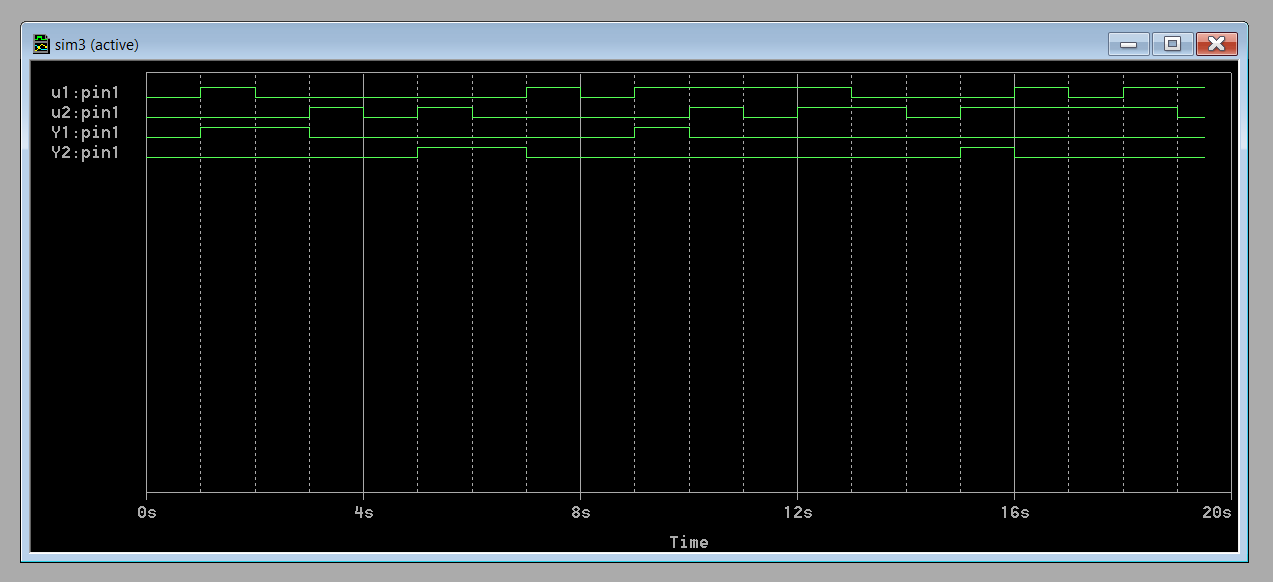


fig.3.8.1

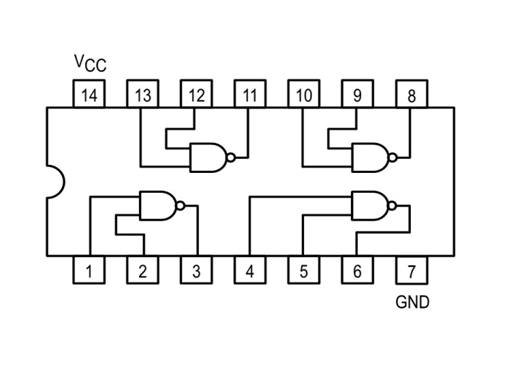
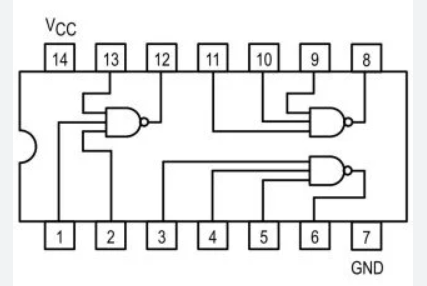
1. Bibliografie

* 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:

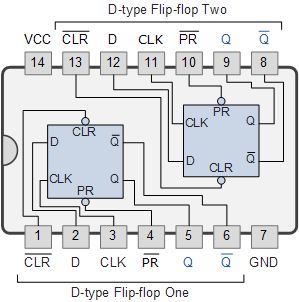
* <https://www.alldatasheet.com/datasheet-pdf/pdf/15541/PHILIPS/74HC154.html>
* <https://www.alldatasheet.com/datasheet-pdf/pdf/125043/ETC1/74S00.html>
* <https://www.princeton.edu/~mae412/HANDOUTS/Datasheets/74S04.PDF>
* <https://www.alldatasheet.com/datasheet-pdf/pdf/15577/PHILIPS/74HC27.html>
* <https://www.alldatasheet.com/datasheet-pdf/pdf/15563/PHILIPS/74HC21.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>

1. Anexa A



CI 74LS10

CI 74LS00



CI 74LS74