Proiect CLP

Tema nr. 40

Funcţionarea unei maşini de stare sincrone este reprezentată prin organigrama ASM din figura de mai jos.

RAS

CAS

F1

STROB

LOAD

F0

RAS

F2

F1

CAS

000

ABC

001

010

011

101

100

110

111

1

0

0

1

1

1

1

1

1. Se numeşte bloc ASM o structură care constă dint-un unic element de stare şi o reţea de elemente de decizie respectiv elememente de ieşiri condiţionate. Blocul ASM se obţine prin conectarea celor trei elemente fundamentale ale organigramei ASM. Un bloc ASM are o singură cale de intrare şi un număr oarecare de căi de ieşire determinate de structura elementelor de decizie. Un bloc ASM descrie funcţionarea automatului pe perioada unei singure stări. Fiecare bloc ASM reprezintă starea prezentă sau curentă (yt(A)), ieşirile comandate pe durata stării A dată de g1(y(t)), ieşirile condiţionate de seturile de intrări date prin funcţia g2(y(t),x(t)) şi explicitează de asemenea tranziţiile automatului în stările următoare. Orice cale sau traseu posibil ce leagă o stare cu o stare următoare se numeşte conexiune de stare/cale de legătură. În general există una sau mai multe căi de legătură asociate fiecărei caăi de ieşire dintr-un bloc ASM în funcţie de modul de conectare a elementelor de decizie. Un bloc ASM arată deci starea curentă, ieşirile sale şi condiţiile pentru fiecare cale de legătură a stării următoare.

Blocurile ASM pentru stări sunt prezentate mai jos:

|  |  |
| --- | --- |
| RAS  CAS  000  ABC  001  L1 | Starea stabilă S0:  Ieșire necondiționată RAS;  Cale de legătură L1, starea următoare S1 |
| CAS  001  ABC  010  L2 | Starea stabilă S1:  Ieșire necondiționată CAS;  Cale de legătură L2, starea următoare S2 |
| 010  LOAD  STROB  F1  101  L2  011  1  0  L3  L4 | Starea stabilă Ss:  Intrare: F1  Căi de legătură:   * L3, când F1 = 0, Starea următoare S5 * L4, când F1= 1, Starea următoare S3 |
| LOAD  011  ABC  100  L5 | Sarea stabilă S3:  Ieșire necondiționată: LOAD  Cale de legătură L5, starea următoare S4 |
| L2  000  0  1  L6  L7  F0  CAS  RAS  100  110 | Starea stabilă S4:  Intrare: F0  Căi de legătură:   * L6, când F0 = 0, Starea următoare S6 * L7 când F0 = 1, starea următoare S0 |
| STROB  RAS  101  ABC  000  L8 | Starea stabilă S5:  Ieșiri necondiționate: V, STROB  Cale de legătură: L8, starea următoare S0 |
| CAS  RAS  111  ABC  110  L9 | Starea stabilă S7:  Ieșire necondiționată: CAS  Cale de legătură: L9, starea următoare S6 |
| 111  110  CAS  F1  F2  RAS  ABC  000  RAS  L12  L10  L11  1  1  0  0 | Starea stabilă S6:  Ieșiri necondiționate: RAS  Intrări: F1, F2  Ieșire condiționată U(F2)  Căi de legătură:   * L9 când F2 = F1 = 1, starea următoare S7 * L10 când F2 = 1 && F1 = 0, starea următoare S6 * L11 când F2 = 0, starea următoare S0 |

Tabelul ASM:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Intrări** | | | **Starea prezentă** | | | | **Starea următoare** | | | | **Ieșiri** | | | | | |
| **F0** | **F1** | **F2** | **sym** | **A** | **B** | **C** | **sym** | **DA** | **DB** | **DC** | **CAS** | **RAS** | **U** | **V** | **STROB** | **LOAD** |
| X | X | X | S0 | 0 | 0 | 0 | S1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| X | X | X | S1 | 0 | 0 | 1 | S2 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| X | 1 | X | S2 | 0 | 1 | 0 | S5 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| X | 0 | X | S2 | 0 | 1 | 0 | S3 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| X | X | X | S5 | 1 | 0 | 1 | S0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| X | X | X | S3 | 0 | 1 | 1 | S4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | X | X | S4 | 1 | 0 | 0 | S0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | X | X | S4 | 1 | 0 | 0 | S6 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| X | X | 0 | S6 | 1 | 1 | 0 | S0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| X | 0 | 1 | S6 | 1 | 1 | 0 | S6 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| X | 1 | 1 | S6 | 1 | 1 | 0 | S7 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| X | X | X | S7 | 1 | 1 | 1 | S6 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

1. Implementare cu MUX-uri

Aflarea funcțiilor generatorului noilor stări:

Pentru ABC = 000 DA = 0 DB = 0 DC = 1

Pentru ABC = 001 DA = 0 DB = 1 DC = 0

Pentru ABC = 010 DA = DB = DC = 1

Pentru ABC = 011 DA = 1 DB = 0 DC = 0

Pentru ABC = 100 DA = DB = DC = 0

Pentru ABC = 101 DA = 0 DB = 0 DC = 0

Pentru ABC = 110 DA = DB = DC =

Pentru ABC = 111 DA = 1 DB = 1 DC = 0

Rezultă următoarele diagrame „VID”

DA:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A\BC | 00 | 01 | 11 | 10 |
| 0 | 0 | 0 | 1 |  |
| 1 |  | 0 | 1 |  |

DB:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A\BC | 00 | 01 | 11 | 10 |
| 0 | 0 | 1 | 0 |  |
| 1 |  | 0 | 1 |  |

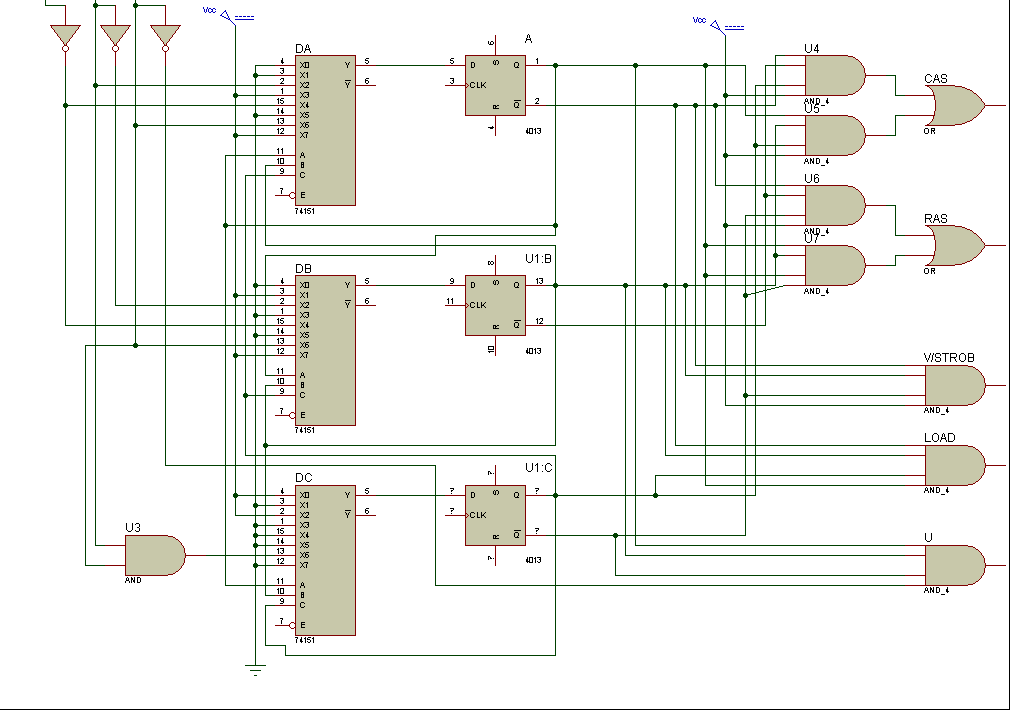
DC:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A\BC | 00 | 01 | 11 | 10 |
| 0 | 1 | 0 | 0 |  |
| 1 | 0 | 0 | 0 |  |

Rezultă funcțiile reduse:

Funcțiile ieșirilor le vom deduce identificând stăriile în care acestea sunt active:

* + CAS e activă în S1 și S7, deci funcția va fi:
  + CAS e activă în S0 și S6, deci funcția va fi:
  + U e activă în S6, numai când F2=1, deci funcția va fi:
  + V e activă în S6, deci funcția va fi:
  + STROB e activă în S6, deci funcția va fi:
  + LOAD e activă în S3, deci funcția va fi:



F2

F1

F0

1. Implementare cu ROM

Un bloc ASM reprezintă o structură care constă dintr-un element de stare unic şi o reţea de elemente de decizie, respectiv de ieşiri condiţionate, reţeaua fiind obţinută prin interconectarea celor 3 elemente de bază. Un bloc ASM are o singură cale de intrare şi un număr oarecare de căi de ieşire determinate de structura elementelor de decizie.

Un bloc ASM descrie funcţionarea automatului pe perioada unei singure stări. Fiecare bloc ASM reprezintă starea prezentă sau curentă şi explicitează de asemenea tranziţiile automatului în stările următoare. Orice cale sau traseu posibil ce leagă o stare cu o stare următoare se numeşte conexiune de stare/cale de legătură. În general, există una sau mai multe căi de legătură asociate fiecărei căi de ieşire dintr-un bloc ASM în funcţie de modul de conectare a elementelor de decizie.

Un bloc ASM arată deci, starea curentă, ieşirile sale şi condiţiile pentru fiecare cale de legătură a stării următoare.

Pentru ASM-urile definite printr-un număr de stări mai mare de 20 se recomandă şi se justifică din punct de vedere economic utilizarea memoriilor ROM în raport cu logica realizată cu circuite integrate pe scară redusă, respectiv medie pentru construcţia GNS. ASM-urile organizate structural ca automate programabile algoritmice bazate pe memorii ROM sunt utilizate pentru aplicaţii la care vitezele de prelucrare ale informaţiilor sunt ridicate.

Starea curentă

Tranziție

Starea urm.

S0 S1 S2 S2 S3 S4 S4 S5 S6 S6 S6 S7

L1 L2 L3 L4 L5 L7 L6 L8 L9 L11 L12 L10

S1 S2 S5 S3 S4 S0 S6 S0 S6 S0 S7 S6

Vom utiliza cuvântul:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| DA | DB | DC | CAS | RAS | U | STROB/V | LOAD |

Din tabela ASM vom comlpeta valoriile acestui cuvânt pentru fiecare adresă de la 0x0 la 0x40

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| adresă | | | | | | Locații ASM | | Cuvânt | Cuvânt (binar) | | | | | | | |
| A | B | C | F0 | F1 | F2 | S | L | DA | DB | DC | CAS | RAS | U | V | LOAD |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 28 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 28 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 28 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 28 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 28 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 28 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 28 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 28 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 2 | 50 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 1 | 2 | 50 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 | 2 | 50 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 1 | 2 | 50 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 2 | 50 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 | 2 | 50 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 | 2 | 50 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 1 | 2 | 50 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 2 | 4 | 60 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 2 | 4 | 60 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 | 2 | 3 | A0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 1 | 2 | 3 | A0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 2 | 4 | 60 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 2 | 4 | 60 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 | 2 | 3 | A0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 | 2 | 3 | A0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 3 | 5 | 81 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 1 | 3 | 5 | 81 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 3 | 5 | 81 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 3 | 5 | 81 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 3 | 5 | 81 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 | 3 | 5 | 81 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 | 3 | 5 | 81 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 3 | 5 | 81 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 4 | 6 | C0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 4 | 6 | C0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 4 | 6 | C0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 | 4 | 6 | C0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 4 | 7 | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 4 | 7 | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 4 | 7 | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 4 | 7 | 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 5 | 8 | 02 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 5 | 8 | 02 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 5 | 8 | 02 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 1 | 5 | 8 | 02 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 5 | 8 | 02 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 | 5 | 8 | 02 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 5 | 8 | 02 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 | 5 | 8 | 02 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 6 | 11 | 0C | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 6 | 10 | C8 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 6 | 11 | 0C | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 1 | 6 | 12 | E8 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 6 | 11 | 0C | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 6 | 10 | C8 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 6 | 11 | 0C | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 | 6 | 12 | E8 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 7 | 9 | D0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 | 7 | 9 | D0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 7 | 9 | D0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 | 7 | 9 | D0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 7 | 9 | D0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 7 | 9 | D0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 | 7 | 9 | D0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 7 | 9 | D0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |

A

B

C

F0

F1

F2

DA DB DC RAS CAS U STROB/V LOAD

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ABC\F0F1F2 | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
| 000 | 28 | 28 | 28 | 28 | 28 | 28 | 28 | 28 |
| 001 | 50 | 50 | 50 | 50 | 50 | 50 | 50 | 50 |
| 010 | 60 | 60 | A0 | A0 | 60 | 60 | A0 | A0 |
| 011 | 81 | 81 | 81 | 81 | 81 | 81 | 81 | 81 |
| 100 | C0 | C0 | C0 | C0 | 00 | 00 | 00 | 00 |
| 101 | 02 | 02 | 02 | 02 | 02 | 02 | 02 | 02 |
| 110 | 0C | C8 | 0C | E8 | 0C | C8 | 0C | E8 |
| 111 | D0 | D0 | D0 | D0 | D0 | D0 | D0 | D0 |

Q D

clk

Q D

clk

Q D

clk

1. Implementarea cu FPLA a maşinii algoritmice de stare în care se utilizează maparea sau tabelarea directă a conexiunilor de stare sau a căilor de legătură. Pentru obţinerea expresiilor necesare programării FPLA-ului se analizează tabelul de stare dat, luându-se în considerare valorile “1” logic din coloanele corespunzătoare ieşirilor (CAS, RAS, U, V, STROB, LOAD) respectiv stărilor următoare (DA, DB, DC).