

MINISTERUL EDUCAȚIEI AL ROMÂNIEI  
Universitatea Tehnică „Gheorghe Asachi”  
FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE



# *Proiect*

## *Tema nr. 10*

Automate și microprogramare

INGINERIA SISTEMLOR

Verificat de:

conf.dr.ing. Pricop Andrei-Traian

Realizat de:

Chiriac Mihai - Iulian, 1202A

Iași, 2025



## Cuprins:

### 1. Descrierea procesului

1.1 Tema proiectului .....	pag. - 1 -
1.2 Definirea variabilelor primare .....	pag. 3

### 2. Sinteza comenzii cu circuite SSI

2.1 Graful primar de tranziții .....	pag. 6
2.2 Graful de tranziții redus .....	pag. 7
2.3 Etapele de sinteză	
2.3.1 Alocarea (asignarea) stărilor .....	pag. 8
2.3.2 Stabilirea diagramelor de excitație ale elementelor de memorie	pag. 9
2.3.3 Stabilirea funcțiilor de control .....	pag. 10
2.3.4 Implementarea funcțiilor de control .....	pag. 11
2.4 Schema logică de comandă .....	pag. 12
2.5 Necesari materiale (componente electronice) .....	pag. 15

### 3. Sinteza comenzii cu circuite MSI

3.1 Graful de tranziții .....	pag. 16
3.2 Etapele de sinteză	
3.2.1 Alocarea (asignarea) stărilor .....	pag. 16
3.2.2 Funcțiile logice atașate multiplexoarelor de numărare și de salt	pag. 17
3.2.3 Programarea salturilor .....	pag. 18
3.2.4 Implementarea funcțiilor logice .....	pag. 18
3.3 Schema logică de comandă .....	pag. 19



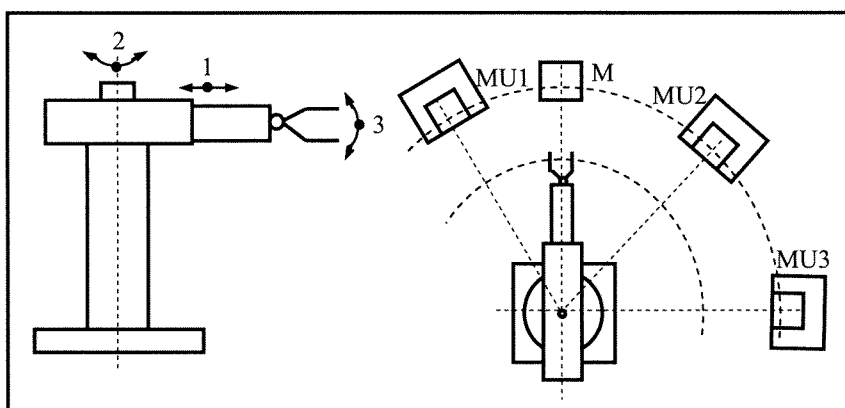
3.4 Necesari materiale (componente electronice) .....	pag. 23
<b>4. Sinteza comenzii cu echipamentul AP MICRO</b>	
4.1 Definirea și alocarea variabilelor procesului .....	pag. 25
4.2 Organizarea generală a programului .....	pag. 27
4.3 Reprezentarea logică a programului funcționării automate .....	pag. 28
4.4 Expresii logice atașate programelor manual și de ieșire .....	pag. 29
4.5 Programul de lucru .....	pag. 31
<b>5. Sinteza comenzii cu releul de control EASY</b>	
5.1 Definirea și alocarea variabilelor procesului (lista de operanzi) .....	pag. 39
5.2 Reprezentarea logică a programului funcționării automate (graf) .....	pag. 40
5.3 Programul de lucru (diagrama de contacte) .....	pag.
<b>6. Concluzii</b> .....	pag.
<b>7. Anexe</b> .....	pag.



## Tema nr. 10

Să se proiecteze comanda automată a unui manipulator cu 2 grade de libertate pentru deservirea a trei mașini-unelte.

### Descrierea procesului:



- 1 - deplasare orizontală braț (avans - retragere)  
2 - rotire corp (stânga - dreapta)  
3 - apucare (închidere - deschidere)  
M - magazie de piese brute (neprelucrate)  
MU1 - prima mașină-unealtă  
MU2 - a doua mașină-unealtă  
MU3 - a treia mașină-unealtă

### Poziția inițială:

Brat retras, rotit la M și cu apucătorul deschis.

**Modul de lucru:**

Piese brute, extrase de la M, se duc la MU1 pentru o primă prelucrare. După această primă prelucrare, piesele se extrag și sunt duse fie la MU2, fie la MU3, pentru prelucrarea finală.

Eliminarea pieselor prelucrate din MU2 sau din MU3 se face automat, fără intervenția manipulatorului.

La comanda START, dată de către operator, brațul alimentează mașina MU1 și așteaptă confirmarea terminării operațiilor corespunzătoare; apoi transferă piesa în MU2 sau în MU3, în funcție de mașina care este liberă la momentul respectiv (testarea se face cu prioritate pentru MU2); după alimentarea mașinii MU2 sau MU3, se va realimenta MU1. Ciclul se continuă cu astfel de transferuri succesive de piese.

O operație de alimentare a mașinii MU1 se desfășoară astfel: brațul avansează (la M), prinde piesa, se retrage, se rotește la MU1, avansează, eliberează piesa, se retrage și comandă inițierea ciclului de lucru al mașinii.

După confirmarea sfârșitului prelucrării în MU1, brațul avansează, prinde piesa prelucrată primar, se retrage, se rotește la MU2 (sau la MU3) și, dacă se confirmă că piesa aflată anterior în mașina respectivă a fost eliminată, avansează, eliberează piesa, se retrage și comandă ciclul de lucru la mașina alimentată.

**Observații:**

1. Se va prevedea în schema de comandă un selector al regimului de lucru, care să permită funcționarea procesului în 3 regimuri:

AUTOMAT CONTINUU - după comanda START, automatul funcționează fără intervenția operatorului, atât timp cât dispune de toate condițiile impuse;

AUTOMAT CICLU CU CICLU - la comanda START, procesul efectuează un ciclu de lucru complet după care așteaptă în poziția inițială o nouă comandă de pornire;

MANUAL (REGLAJ) - fiecare deplasare a elementelor mobile se comandă prin acționarea unor butoane specifice situate pe panoul operatorului.

2. Se vor prevedea 2 butoane de oprire a procesului comandat:

STOP, după acționarea căruia ciclul în curs de execuție este terminat, procesul rămânând în poziția inițială și așteptând o nouă comandă de lucru; după oprirea procesului în urma unei comenzi STOP, tensiunea de alimentare poate fi deconectată;

AVARIE, la apăsarea căruia elementele mobile ale procesului se opresc; reluarea funcționării ansamblului se poate face numai printr-o comandă START, activă în condițiile inițiale stabilite; pentru aceasta, procesul se aduce prin regimul MANUAL în poziția inițială.

3. Zona de acțiune a ansamblului de mașini este protejată printr-o barieră cu ultrasunete cu ieșire discretă. Protecția trebuie declanșată odată cu inițierea unui ciclu de lucru. Pătrunderea unui corp străin în interiorul zonei de lucru în timpul funcționării procesului trebuie să determine blocarea acestuia, simultan cu transmiterea unui semnal de avertizare corespunzător.

4. Deplasările se efectuează la capăt de cursă; pozițiile organelor mobile sunt sesizate de senzori cu ieșire discretă.

5. Inițierea lucrului, sau repunerea în ciclul automat de funcționare după o oprire de avarie poate fi realizată numai de către operator, și numai din poziția inițială stabilită a procesului, printr-o comandă START.



## 1.2 Definirea variabilelor procesului

- Variabilele primare de intrare:

Simbol	Descriere
START	Buton de pornire a ciclului
STOP	Buton de oprire a ciclului
P	Buton de comandă a ciclului automat
SRL	Comutator selectare regim de lucru
PP	Confirmare prezentă piesă
MU1L	Mașină unealtă 1 liberă
MU2L	Mașină unealtă 2 liberă
MU3L	Mașină unealtă 3 liberă
TP1	Confirmare piesă prelucrată la MU1
LA	Confirmare limită avans
LP	Confirmare piesă apucată
LR	Confirmare limită retragere
LE	Confirmare piesă eliberată
LM	Confirmare braț în față la M
L1	Confirmare braț în față la MU1
L2	Confirmare braț în față la MU2
L3	Confirmare braț în față la MU3

- Variabilele primare de ieșire:

Simbol	Descriere
AST	Așteptare semnal pornire
TEST	Comanda efectuării testului
AV	Comanda avansului brațului
PR	Comanda închiderii apucătorului
RET	Comanda retragerii brațului
RS	Comanda rotirii spre stânga
RD	Comanda rotirii spre dreapta
EL	Comanda deschiderii apucătorului
CMU1	Comanda inițierii ciclului automat la MU1
CMU2	Comanda inițierii ciclului automat la MU2
CMU3	Comanda inițierii ciclului automat la MU3

Precizări:

Butonul STOP, este un buton special, de tip impuls, plasat vizibil și ușor accesibil pe panoul operatorului. el are rolul de a opri imediat procesul în orice poziție.

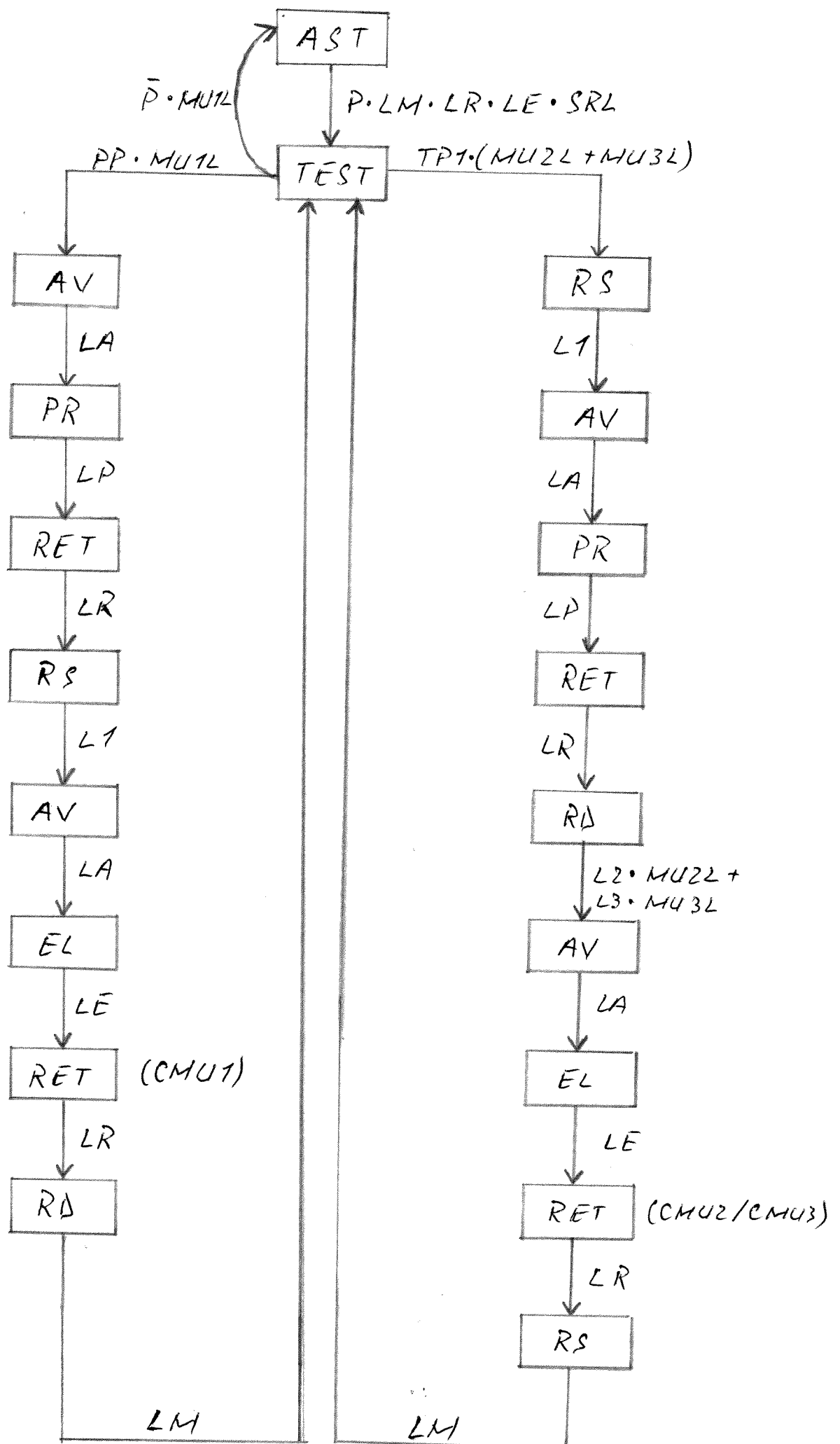
Comutatorul SRL permite selectarea regimului de lucru: Automat, caz în care după lansarea procesului prin apăsarea butonului START, ciclul de lucru continuă până intervenția operatorului; Manual, în care elementele de acționare ale procesului sunt comandate direct de la o serie de butoane specifice

## 2. Sinteza comenzii cu dispozitive SSI

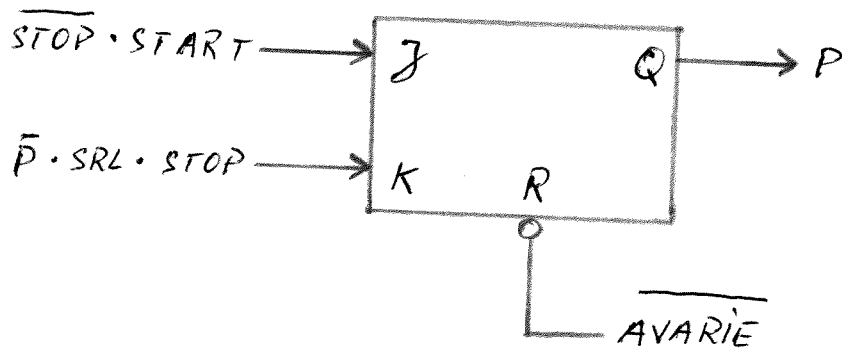
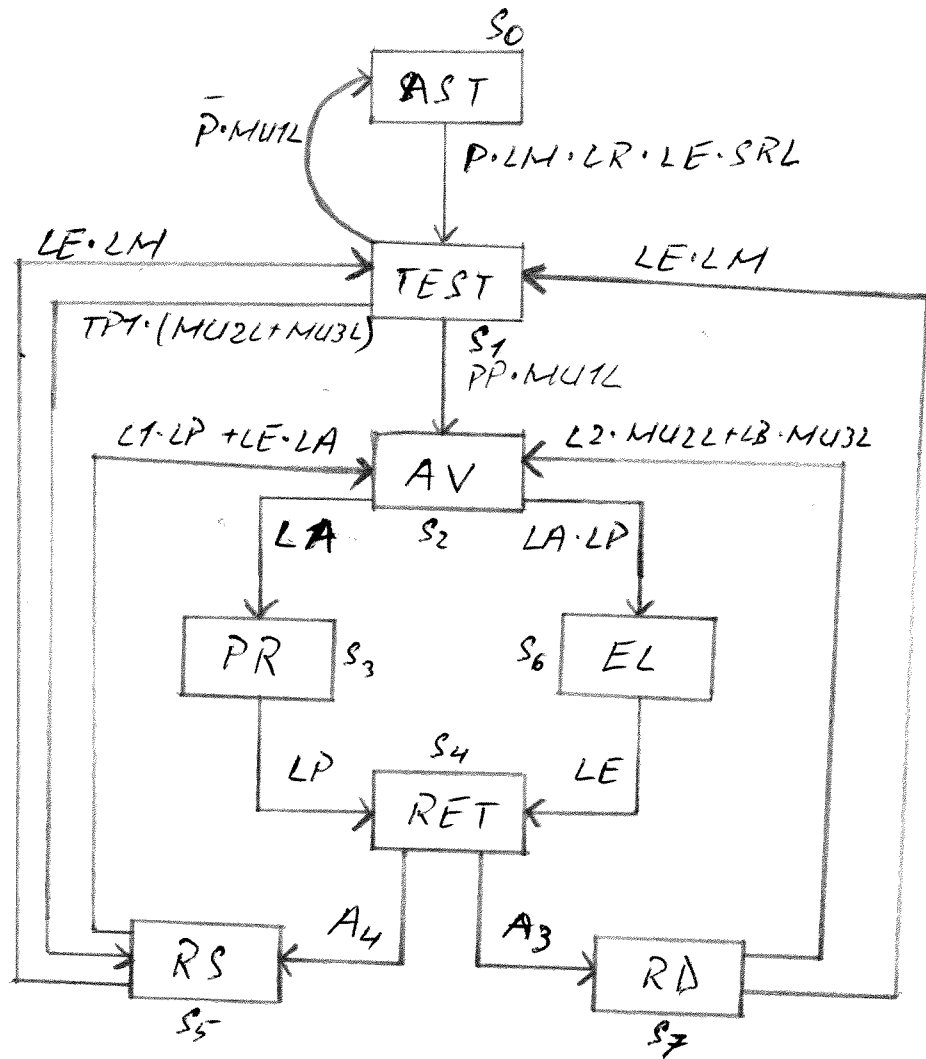
### 2.1. Graful primar de tranziții

În graful primar de tranziții au fost surprinse principalele obiective ale automatizării procesului de deservire, formulate prin tema de proiectare impusă. Procesul demarează numai prin comanda START, dată de către operator și, în funcție de starea comutatorului de regim automat, se repetă ciclic, până când se epuizează stocul de piese de prelucrat furnizat de magazia de piese brute, sau, după efectuarea unui ciclu de alimentare, se oprește în starea inițială, de unde va putea primi o nouă comandă START. Acțiunea butonului de oprire de avarie, selectarea regimului de lucru, precum și alte condiții necesare, dar care nu țin de desfășurarea logică naturală a comenzii în regim automat, vor fi înglobate în cadrul schemei logice de automatizare finale.

Starea inițială (AST) a fost introdusă pentru a selecta regimul de lucru, în funcție de poziția comutatorului SR2, automatul este condus în starea inițială dacă regimul selectat a fost manual, sau în starea de TEST, dacă regimul este **Automat**, iar condițiile de lucru sunt îndeplinite (există piesă în magazie și măcar o mașină - unealtă este liberă).



## 2.2 Graful de tranziții redus



Observații:

După reducerea numărului de stări prin suprapunerea celor echivalente, graficul de tranziții poate deveni nedeterminist, permitând mai multe evoluții pentru aceeași condiție. Pentru a corecta acest lucru, condițiile de tranziție se ajustează astfel încât, în orice moment, să fie posibilă o singură tranziție.

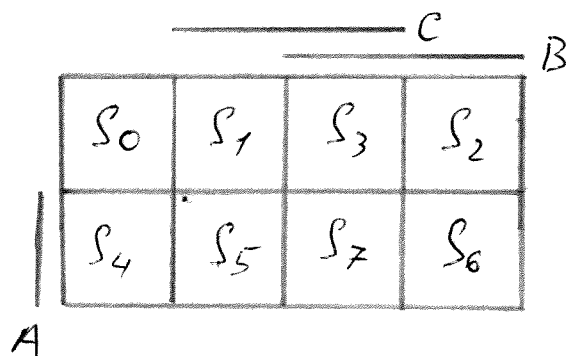
## 2.3 Etapele de sinteză

### 2.3.1 Codificarea (asignarea) stărilor

Deoarece automatul are 8 stări, pentru codificarea acestora se vor folosi 3 variabile secundare,  $A$ ,  $B$  și  $C$ , care vor fi generate de ieșirile a 3 bistabile JK, ca elemente de memorie ale sistemului secvențial.

Funcționarea sistemului va fi sincronă cu un semnal de tact extern, deci nu este necesară asigurarea adiacenței, între stările vecine pe graf.

$S_K$	$A$	$B$	$C$
$S_0$	0	0	0
$S_1$	0	0	1
$S_2$	0	1	0
$S_3$	0	1	1
$S_4$	1	0	0
$S_5$	1	0	1
$S_6$	1	1	0
$S_7$	1	1	1



## 2.3.2 Diagramele de excitatii ale bistabilelor

$S_i$	Condiție	$S_f$
$S_0$	$A_1$	$S_1$
$S_1$	$\bar{P} \cdot MU2L$	$S_0$
	$PP \cdot MU2L$	$S_2$
	$A_2$	$S_5$
$S_2$	$LA$	$S_3$
	$LA \cdot LP$	$S_6$
$S_3$	$LP$	$S_4$
$S_4$	$A_3$	$S_5$
	$A_4$	$S_7$
$S_5$	$A_5$	$S_2$
	$LE \cdot LM$	$S_1$
$S_6$	$LE$	$S_4$
$S_7$	$LE \cdot LM$	$S_1$
	$A_6$	$S_2$

$Q$	$Q_{+1}$	$J$	$K$
0	0	0	*
0	1	1	*
1	0	*	1
1	1	*	0

În tabelul tranzițiilor s-a notat:

$$\begin{cases} A_1 = P \cdot LM \cdot LR \cdot LE \cdot SRL \\ A_2 = TP1 \cdot (MU2L + MU3L) \\ A_3 = LR \cdot LP \cdot LM + LR \cdot LE \cdot (L2 + L3) \\ A_4 = LR \cdot LP \cdot LM \cdot (MU2L + MU3L) \\ A_5 = L1 \cdot LP + LE \cdot LA \\ A_6 = L2 \cdot MU2L + L3 \cdot MU3L \end{cases}$$

	$C$				$B$
	0	$A_2$	$LP$	$LA \cdot LP$	
$A$	*	*	*	*	

$JA$

	$C$				$B$
	*	*	*	*	
$A$	0	$B_1$	$B_2$	0	

$KA$

	$C$				$B$
	0	$PP \cdot MU2L$	*	*	
$A$	$A_4$	$A_5$	*	*	

$JB$

	$C$				$B$
	*	*	$LP$	0	
$A$	*	*	$LE \cdot LM$	$LE$	

$KB$

	$C$				$B$
	$A_1$	*	*	$L1$	
$A$	$B_3$	*	*	0	

$JC$

	$C$				$B$
	*	$B_4$	$LP$	*	
$A$	*	$A_5$	$A_6$	*	

$KC$

În diagramele de excitații s-a notat:

$$\begin{cases} B_1 = L_1 \cdot LP + LE \cdot LA + LE \cdot LM \\ B_2 = A_5 + LE \cdot LM \\ B_3 = A_4 + A_3 \\ B_4 = \bar{P} \cdot MU1L + PP \cdot MU1L \end{cases}$$

### 2.3.3 Funcțiile de control în formă minimă:

• Funcțiile de excitație ale bistabilelor:

$$\begin{cases} JA = A_2 \cdot \bar{B} \cdot C + LP \cdot B \cdot C + LA \cdot LP \cdot B \cdot \bar{C} \\ KA = B_1 \cdot \bar{B} \cdot C + B_2 \cdot B \cdot C \\ JB = PP \cdot MU1L \cdot \bar{A} \cdot C + A_5 \cdot A \cdot C + A_4 \cdot A \cdot \bar{C} \\ KB = LP \cdot \bar{A} \cdot C + LE \cdot LM \cdot A \cdot C + LE \cdot A \cdot \bar{C} \\ JC = A_1 \cdot \bar{A} \cdot \bar{B} + L_1 \cdot B \cdot \bar{A} + B_3 \cdot A \cdot \bar{B} \\ KC = B_4 \cdot \bar{A} \cdot \bar{B} + LP \cdot \bar{A} \cdot B + A_5 \cdot A \cdot \bar{B} + A_6 \cdot A \cdot B \end{cases}$$

• Funcțiile de ieșire (decodificarea stărilor):

- Funcții de comandă:

$$\begin{cases} AV = S_2 = \bar{C} \cdot B \cdot \bar{A} \\ RET = S_4 = C \cdot \bar{B} \cdot \bar{A} \\ PR = S_3 = \bar{C} \cdot B \cdot A \\ RS = S_5 = C \cdot \bar{B} \cdot A \\ EL = S_6 = C \cdot B \cdot \bar{A} \\ RD = S_7 = C \cdot B \cdot A \end{cases}$$

- Funcții auxiliare:

$$\begin{cases} AST = S_0 = \bar{C} \cdot \bar{B} \cdot \bar{A} \\ TEST = S_1 = \bar{C} \cdot \bar{B} \cdot A \end{cases}$$



### 2.3.4 Implementarea funcțiilor de control:

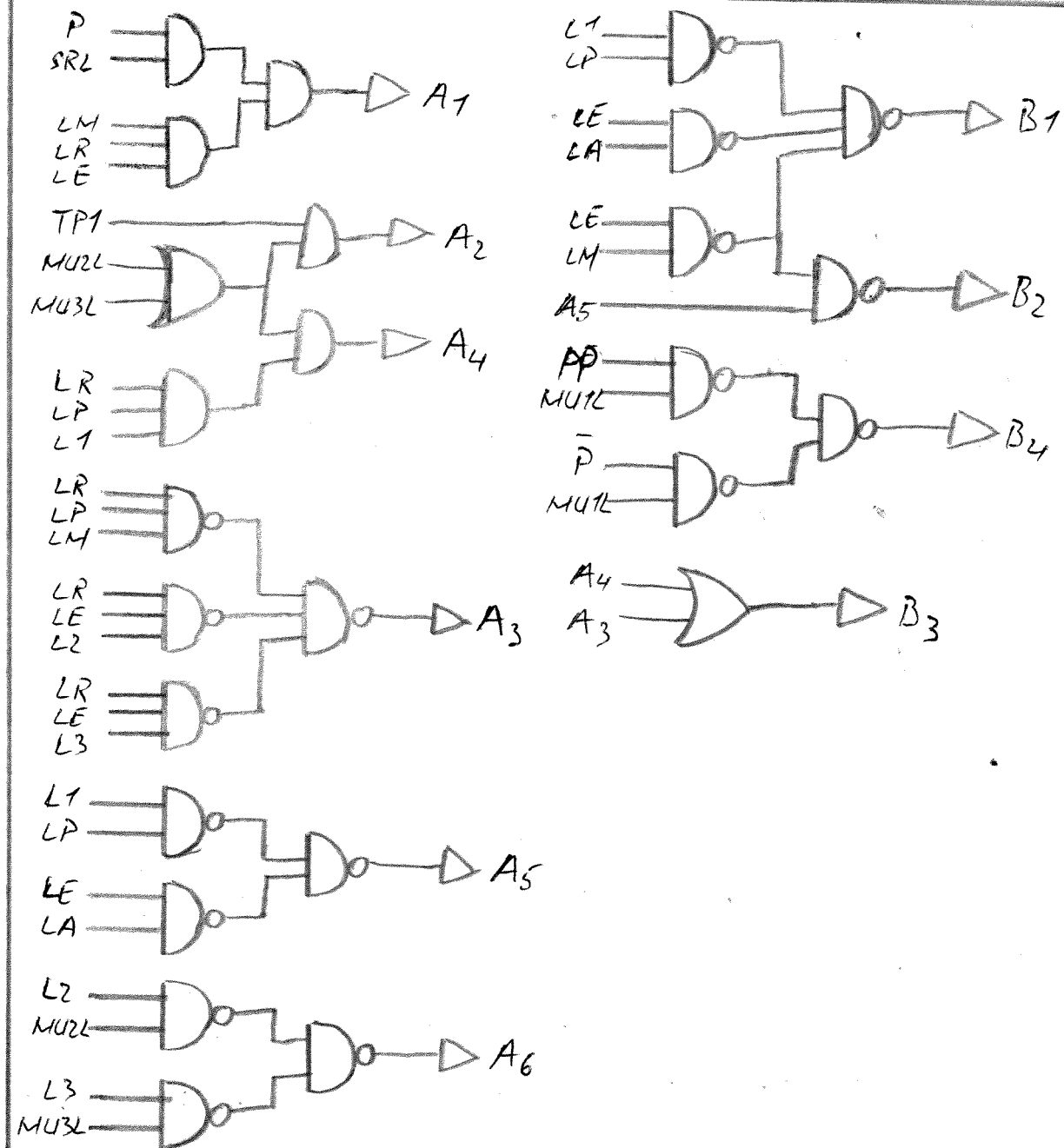
$$\left\{ \begin{array}{l} A_1 = P \cdot (LM \cdot LR \cdot LE) \cdot SRL \\ A_2 = TP1 \cdot (MU2L + MU3L) \\ A_3 = (LR \uparrow LP \uparrow LM) \uparrow (LR \uparrow LE \uparrow L2) \uparrow (LR \uparrow LE \uparrow L3) \\ A_4 = LR \cdot LP \cdot L1 \cdot (MU2L + MU3L) \\ A_5 = (L1 \uparrow LP) \uparrow (LE \uparrow LA) \\ A_6 = (L2 \uparrow MU2L) \uparrow (L3 \uparrow MU3L) \end{array} \right.$$

$$\left\{ \begin{array}{l} B_1 = (L1 \uparrow LP) \uparrow (LE \uparrow LA) \uparrow (LE \uparrow LM) \\ B_2 = A5 \uparrow (LE \uparrow LM) \\ B_3 = A4 + A3 \\ B_4 = (\bar{P} \uparrow MU1L) \uparrow (PP \uparrow MU1L) \end{array} \right.$$

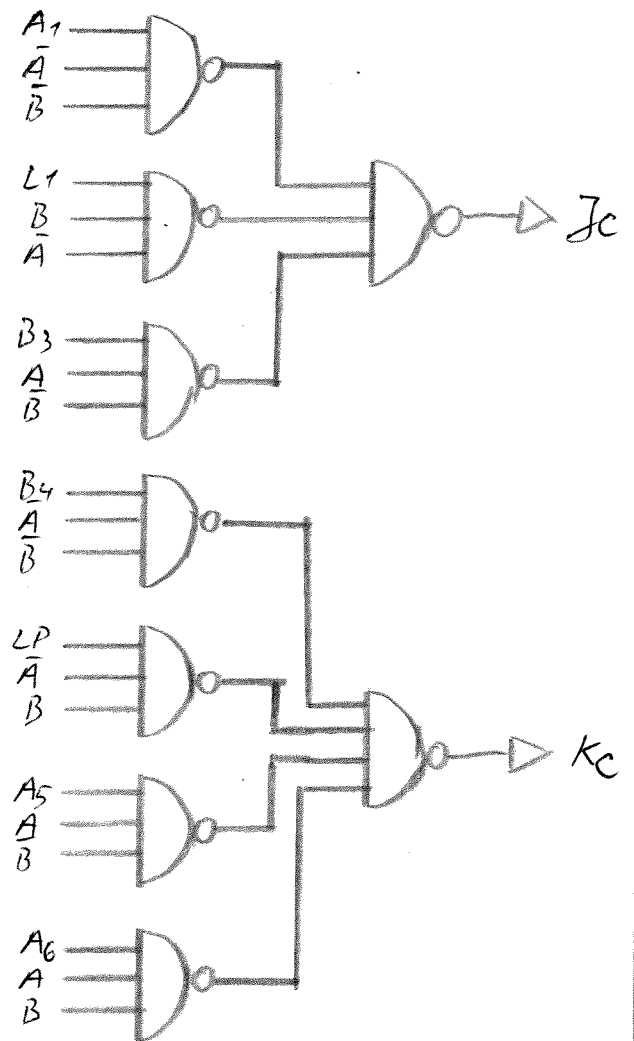
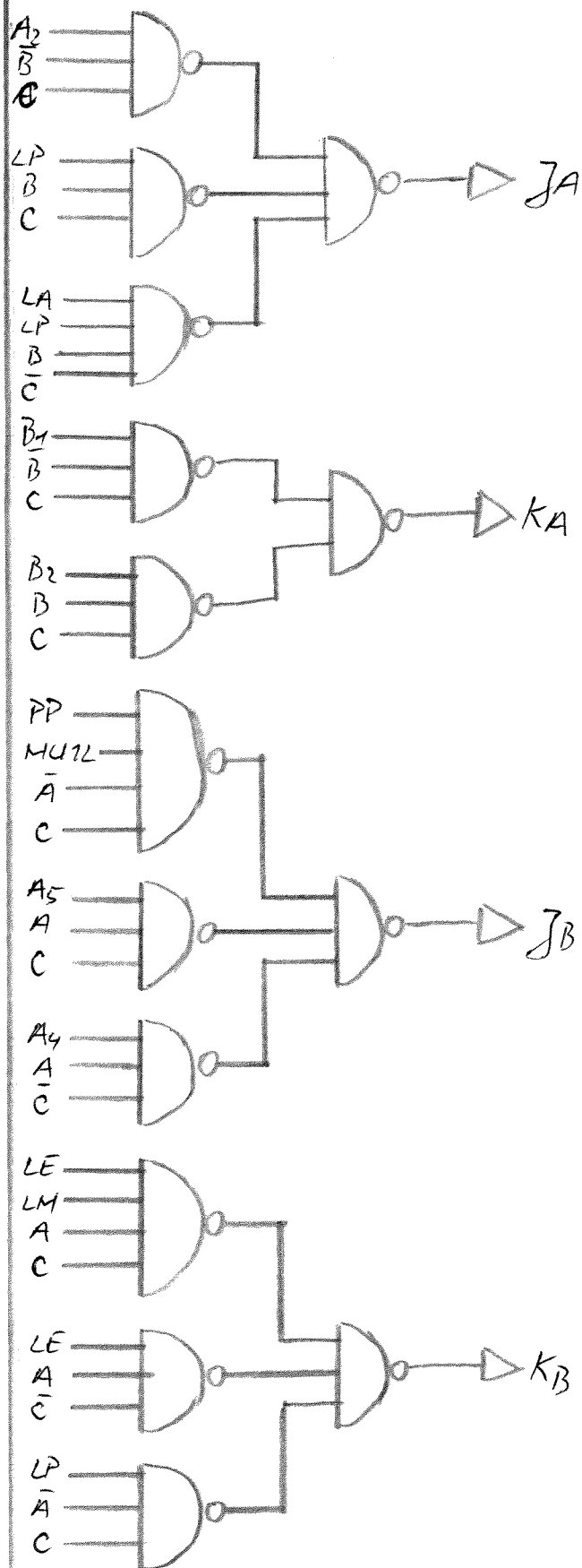
$$\left\{ \begin{array}{l} JA = (A2 \uparrow \bar{B} \uparrow \bar{C}) \uparrow (LP \uparrow B \uparrow \bar{C}) \uparrow (LA \uparrow LP \uparrow B \uparrow \bar{C}) \\ KA = (B1 \uparrow \bar{B} \uparrow \bar{C}) \uparrow (B2 \uparrow B \uparrow \bar{C}) \\ JB = (PP \uparrow MU1L \uparrow \bar{A} \uparrow \bar{C}) \uparrow (A5 \uparrow A \uparrow \bar{C}) \uparrow (A4 \uparrow A \uparrow \bar{C}) \\ KB = (LP \uparrow \bar{A} \uparrow \bar{C}) \uparrow (LE \uparrow LM \uparrow A \uparrow \bar{C}) \uparrow (LE \uparrow A \uparrow \bar{C}) \\ JC = (A1 \uparrow \bar{A} \uparrow \bar{B}) \uparrow (L1 \uparrow B \uparrow \bar{A}) \uparrow (B3 \uparrow A \uparrow \bar{B}) \\ KC = (B4 \uparrow \bar{A} \uparrow \bar{B}) \uparrow (LP \uparrow \bar{A} \uparrow \bar{B}) \uparrow (A5 \uparrow A \uparrow \bar{B}) \uparrow (A6 \uparrow A \uparrow \bar{B}) \end{array} \right.$$

Funcțiile de ieșire nămăn în forma stabilită prin decodificarea stănilon.

## 2.4 Schema logică de comandă



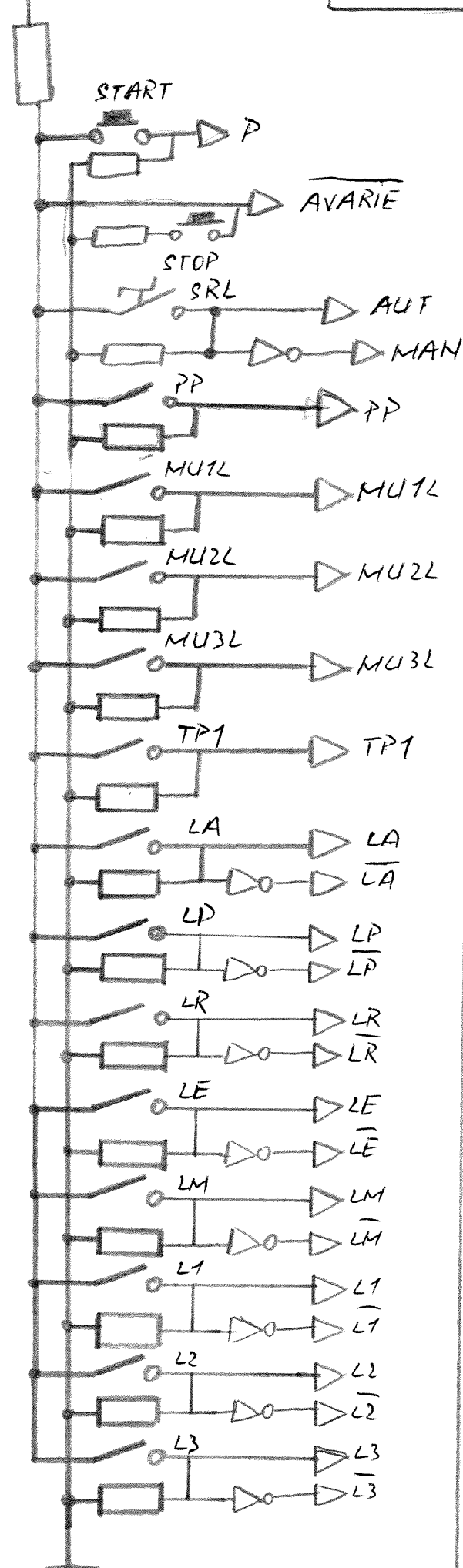
## Functii auxiliare



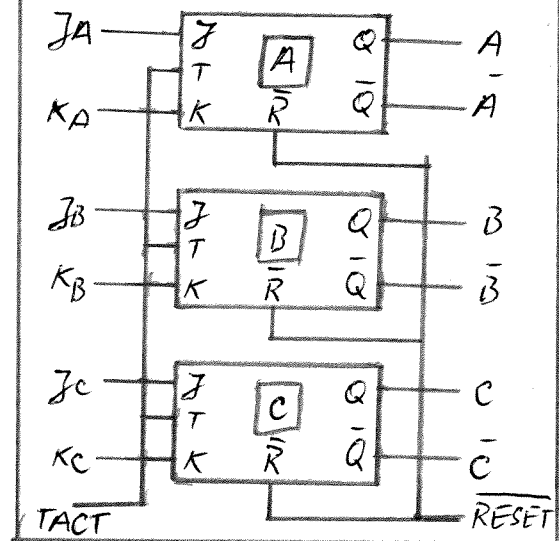
Functii de excitatie

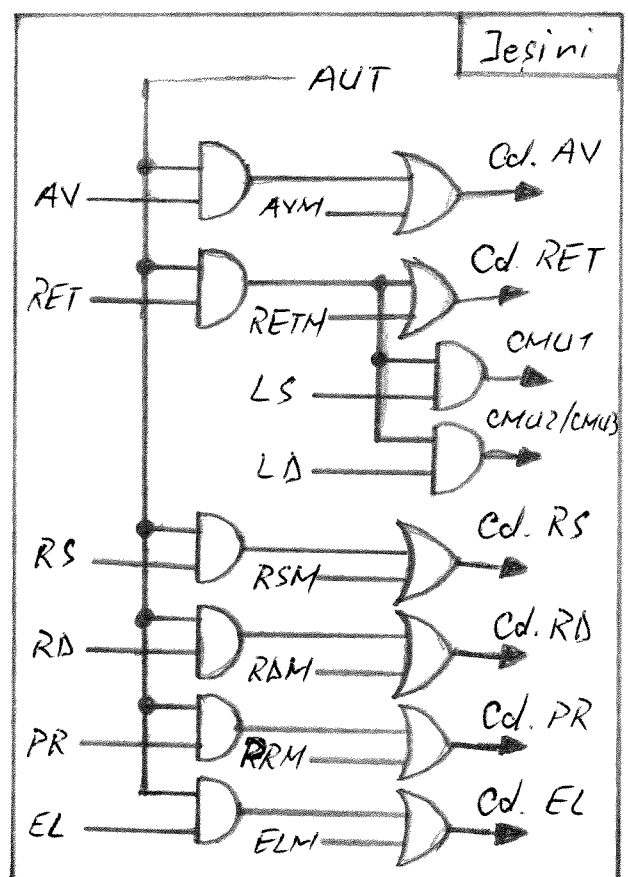
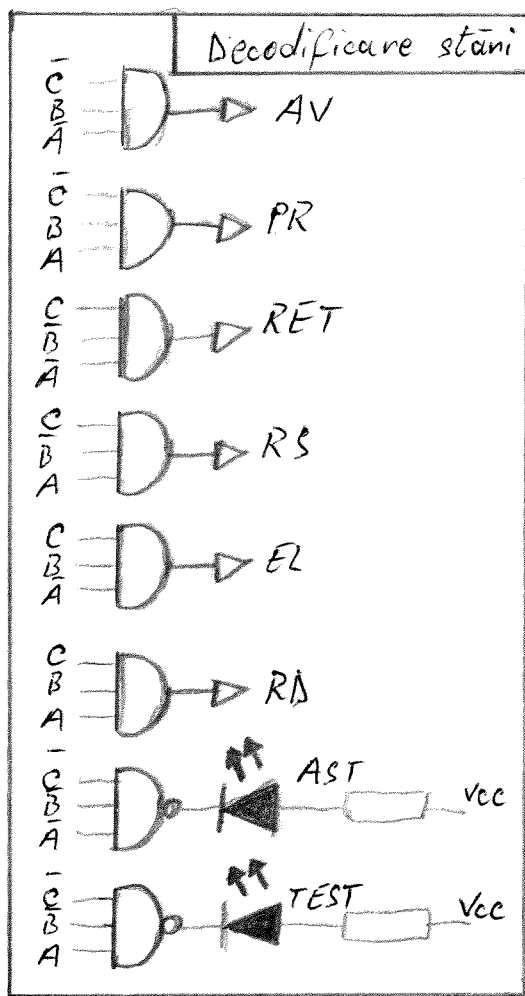
Vcc

Intrani



Memoria





## 2.5 Necesari materiale

Nr. circ.	Poni logic								Bistabile
	NOT	AND-2i	AND-3i	OR-2i	OR-3i	NAND-2i	NAND-3i	NAND-4i	
1	-	4	2	2	-	13	5	-	-
2	-	-	-	-	-	1	13	4	-
3	8	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-	3
5	-	-	6	-	-	-	2	-	-
6	-	8	-	6	-	-	-	-	-
TOTAL	9	12	8	8	-	14	26	4	3

Nr. circuit reprezintă blocurile logice reprezentate:

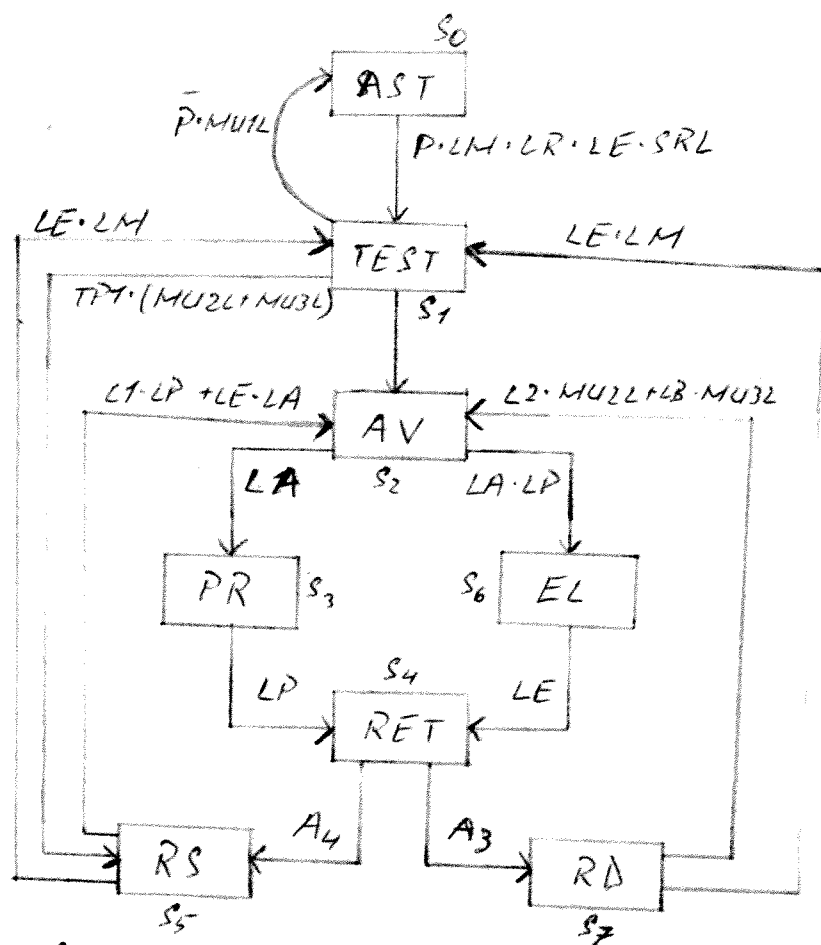
1. Funcții auxiliare
2. Funcții de excitație
3. Intrări
4. Memoria
5. Decodificare stări
6. Ieșiri

Mai sunt necesare următoarele componente:

- Buton impuls. - 2 buc.
- Comutator cu 2 poziții - 1 buc.
- Rezistor de polarizare - 19 buc.
- Rezistor de limitare - 2 buc.
- Diode LED - 2 buc.

### 3. Sinteza comenzii cu circuite MSI

#### 3.1 Graful de tranziții redus



#### 3.2.1 Alocarea (asignarea) stărilor

$S_k$	A	B	C
$S_0$	0	0	0
$S_1$	0	0	1
$S_2$	0	1	0
$S_3$	0	1	1
$S_4$	1	0	0
$S_5$	1	0	1
$S_6$	1	1	0
$S_7$	1	1	1

	C				B
	$S_0$	$S_1$	$S_3$	$S_2$	
A	$S_4$	$S_5$	$S_7$	$S_6$	

### 3.2.2 Funcțiile logice atasate multiplexoarelor

#### Tabelul tranzițiilor

$S_i$	Condiție	$S_f$	Tip
$S_0$	$A_1$	$S_1$	N
$S_1$	$\bar{P} \cdot MU1L$	$S_0$	S
	$PP \cdot MU1L$	$S_2$	N
$S_2$	$A_2$	$S_5$	S
	$LA$	$S_3$	N
	$LA \cdot LP$	$S_6$	S
$S_3$	$LP$	$S_4$	N
$S_4$	$A_3$	$S_5$	N
	$A_4$	$S_7$	S
$S_5$	$A_5$	$S_2$	S
	$LE \cdot LM$	$S_1$	S
$S_6$	$LE$	$S_4$	S
$S_7$	$LE \cdot LM$	$S_1$	S
	$A_6$	$S_2$	S

În tabelul tranzițiilor s-a notat:

$$\begin{cases} A_1 = P \cdot LM \cdot LR \cdot LE \cdot SRL \\ A_2 = TP1 \cdot (MU2L + MU3L) \\ A_3 = LR \cdot LP \cdot LM + LR \cdot LE \cdot (L2 + L3) \\ A_4 = LR \cdot LP \cdot LA \cdot (MU2L + MU3L) \\ A_5 = L1 \cdot LP + LE \cdot LA \\ A_6 = L2 \cdot MU2L + L3 \cdot MU3L \end{cases}$$

$$W_1 = S_0 \cdot A_1 + S_1 \cdot PP \cdot MU1L + S_2 \cdot LA + S_3 \cdot LP + S_4 \cdot A_3$$

$$W_2 = S_1 \cdot (\bar{P} \cdot MU1L + A_2) + S_2 \cdot LA \cdot LP + S_4 \cdot A_4 + S_5 \cdot (A_5 + LE \cdot LM) + S_6 \cdot LE + S_7 \cdot (LE \cdot LM + A_6)$$

$$V_N = (A_1, PP \cdot MU1L, LA, LP, A_3, 0, 0, 0)$$

$$V_S = (0, B_1, LA \cdot LP, 0, A_4, B_2, LE, B_3)$$

unde, s-a notat:

$$\begin{cases} B_1 = \bar{P} \cdot MU1L + A_2 \\ B_2 = A_5 + LE \cdot LM \\ B_3 = A_6 + LE \cdot LM \end{cases}$$

### 3.2.3 Programarea salturilor

	$\overline{C}$				$B$
	X	1	X	1	
A	1	0	0	1	$P_A$

	$\overline{C}$				$B$
	X	0	X	1	
A	1	$A_5$	$A_6$	0	$P_B$

	$\overline{C}$				$B$
	X	1	X	0	
A	1	$LE \cdot LM$	$LE \cdot LM$	0	$P_C$

Formele minime ale funcțiilor de programare a salturilor:

$$\begin{cases} P_A = \overline{C} + \overline{A} \\ P_B = \overline{B} \cdot \overline{C} + \overline{A} \cdot B + A_5 \cdot A \cdot \overline{B} \cdot C + A_6 \cdot A \cdot B \cdot C \\ P_C = \overline{B} \cdot \overline{C} + \overline{A} \cdot C + LE \cdot LM \cdot A \cdot C \end{cases}$$

### 3.2.4. Implementarea funcțiilor logice

• Funcțiile auxiliare:

$$\begin{cases} A_1 = P \cdot (LM \cdot LR \cdot LE) \cdot SRL \\ A_2 = TP1 \cdot (MU2L + MU3L) \\ A_3 = (LR \uparrow LP \uparrow LM) \uparrow (LR \uparrow LE \uparrow L2) \uparrow (LR \uparrow LE \uparrow L3) \\ A_4 = LR \cdot LP \cdot L1 \cdot (MU2L + MU3L) \\ A_5 = (L1 \uparrow LP) \uparrow (LE \uparrow LA) \\ A_6 = (L2 \uparrow MU2L) \uparrow (L3 \uparrow MU3L) \\ B_1 = (\overline{P} \uparrow MU2L) \uparrow A_2 \\ B_2 = A_5 \uparrow (LE \uparrow LM) \\ B_3 = A_6 \uparrow (LE \uparrow LM) \end{cases}$$



- Funcțiile de programare a salturilor:

$$\begin{cases} P_A = \bar{A} + \bar{C} \\ P_B = (\bar{B} \bar{C}) \bar{A} \bar{B} (A_5 \bar{A} \bar{B} \bar{C}) \bar{A} (A_6 \bar{A} \bar{B} \bar{C}) \\ P_C = (\bar{B} \bar{C}) \bar{A} \bar{C} (LE \bar{LM} \bar{A} \bar{C}) \end{cases}$$

- Funcțiile de ieșire (decodificarea stărilor):

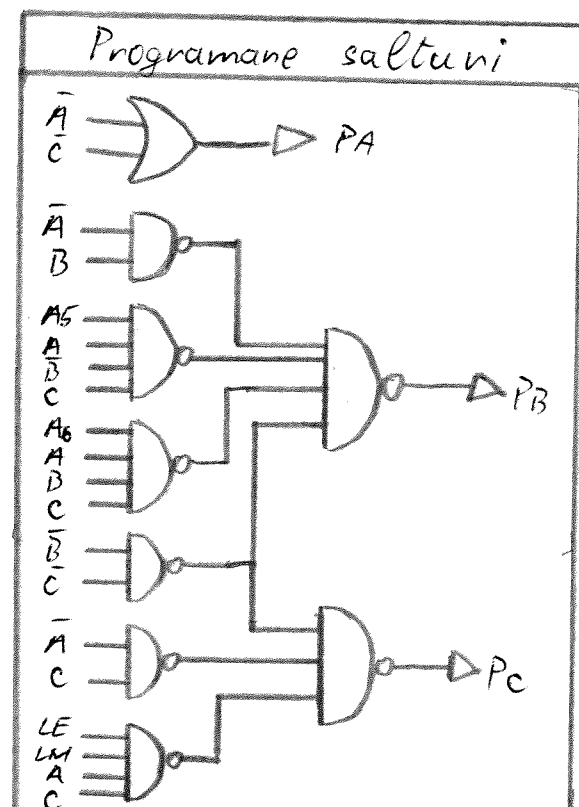
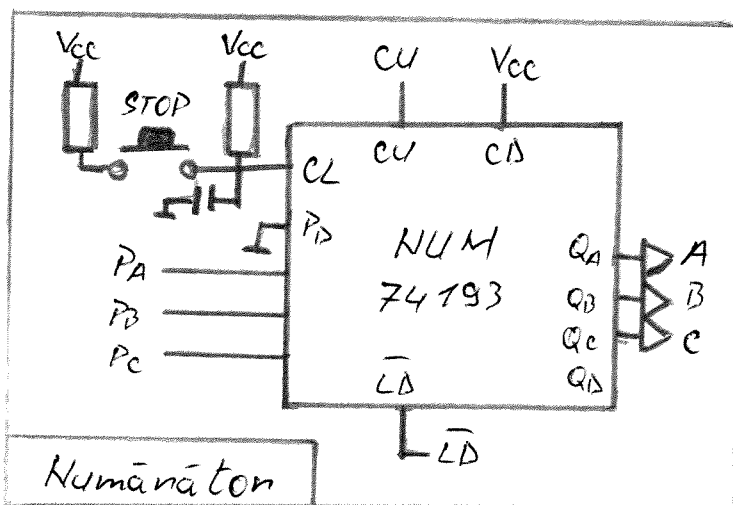
- Funcții de comandă:

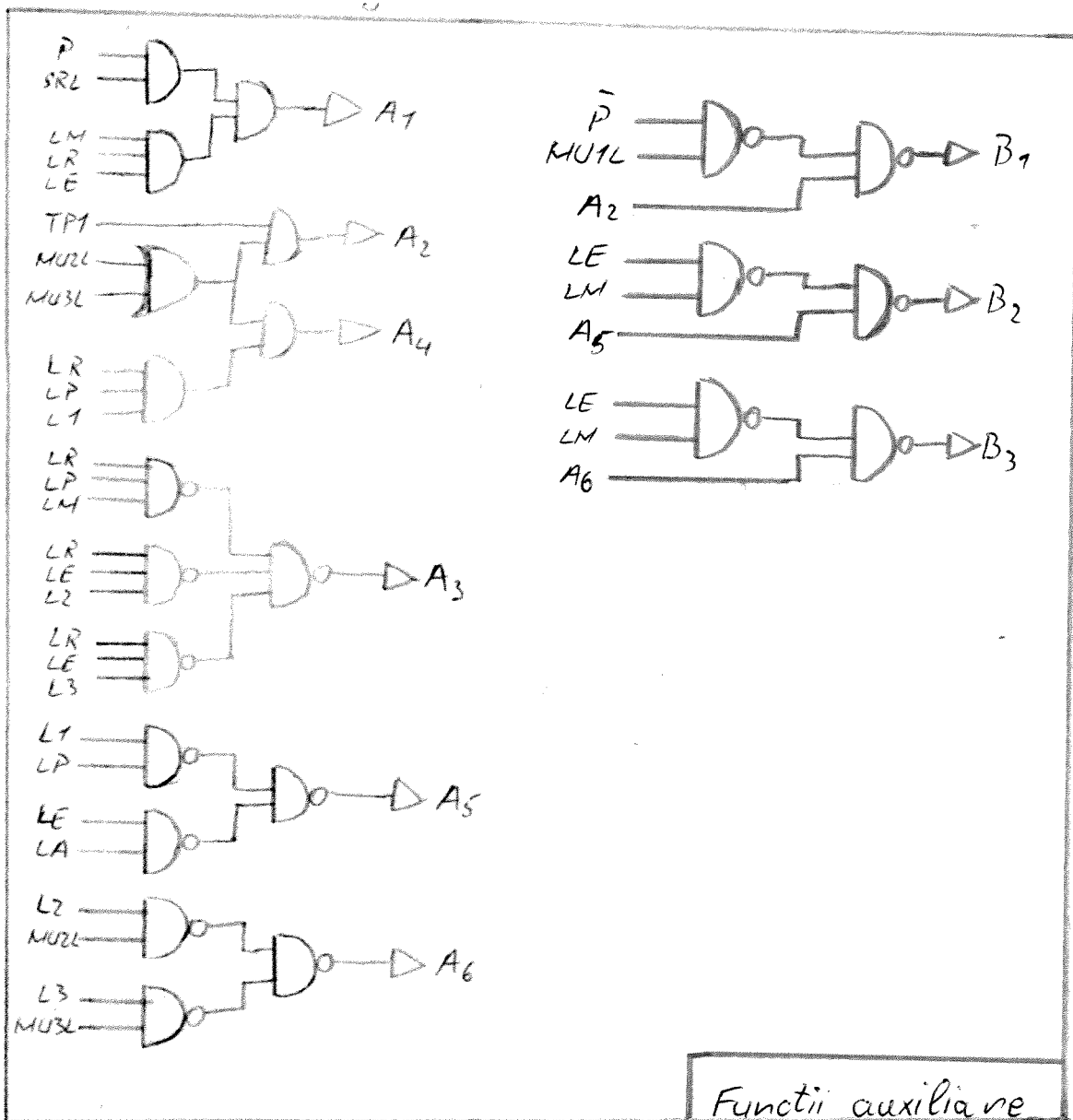
$$\begin{cases} AV = S_2 = \bar{2} \\ PR = S_3 = \bar{3} \\ RET = S_4 = \bar{4} \\ RS = S_5 = \bar{5} \\ EL = S_6 = \bar{6} \\ RD = S_7 = \bar{7} \end{cases}$$

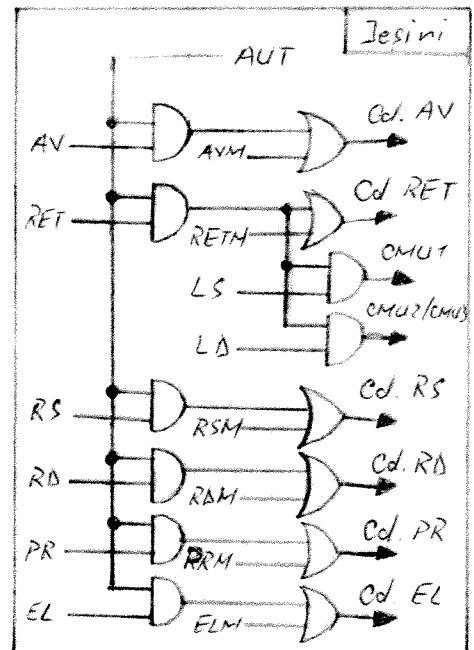
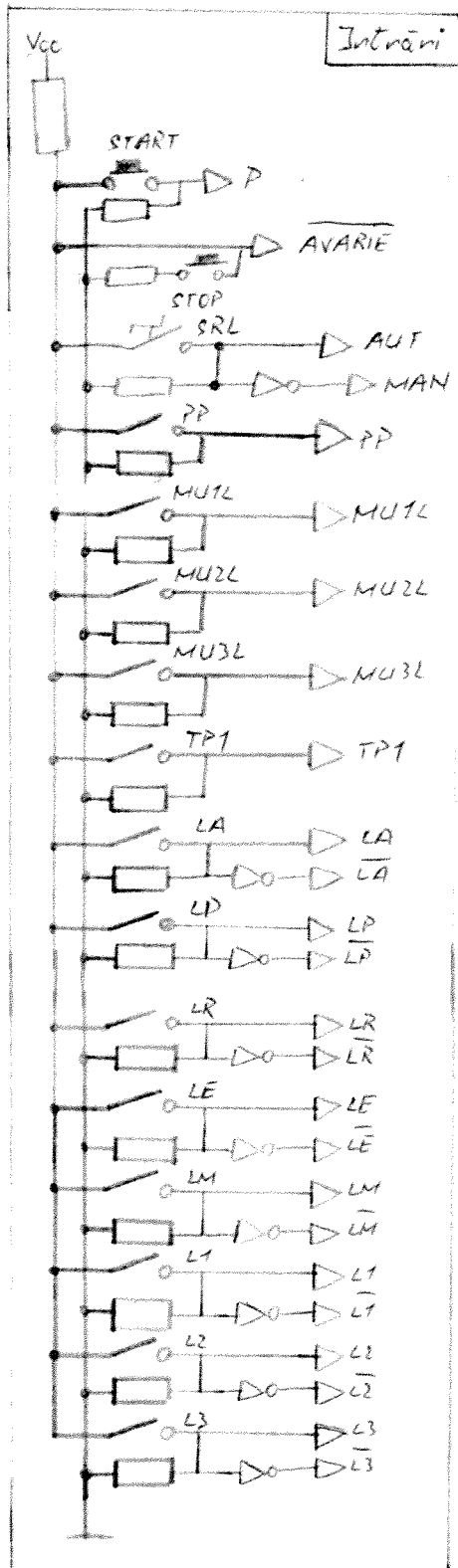
- Funcții auxiliare:

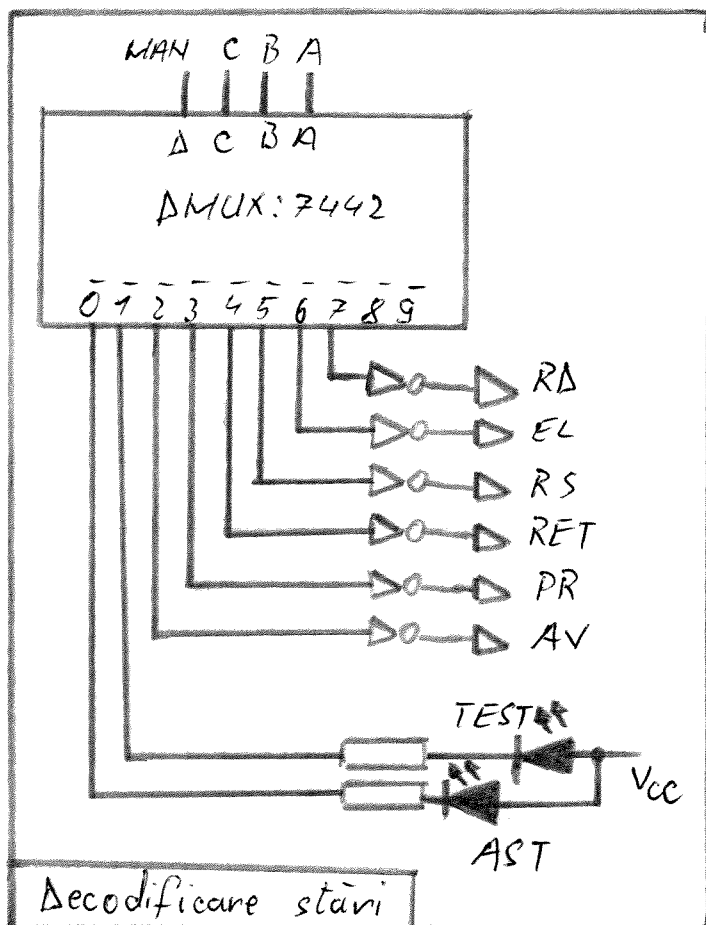
$$\begin{cases} AST = S_0 \rightarrow \bar{AST} = \bar{0} \\ TEST = S_1 \rightarrow \bar{TEST} = \bar{1} \end{cases}$$

### 3.3 Schema logică de comandă.



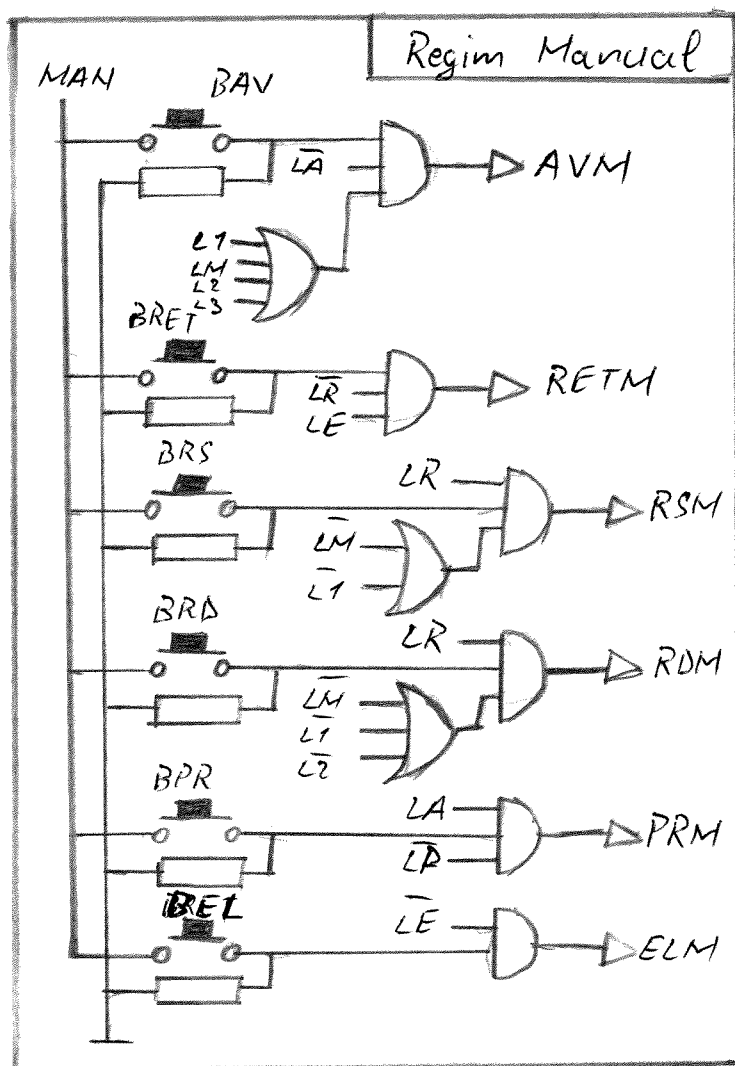


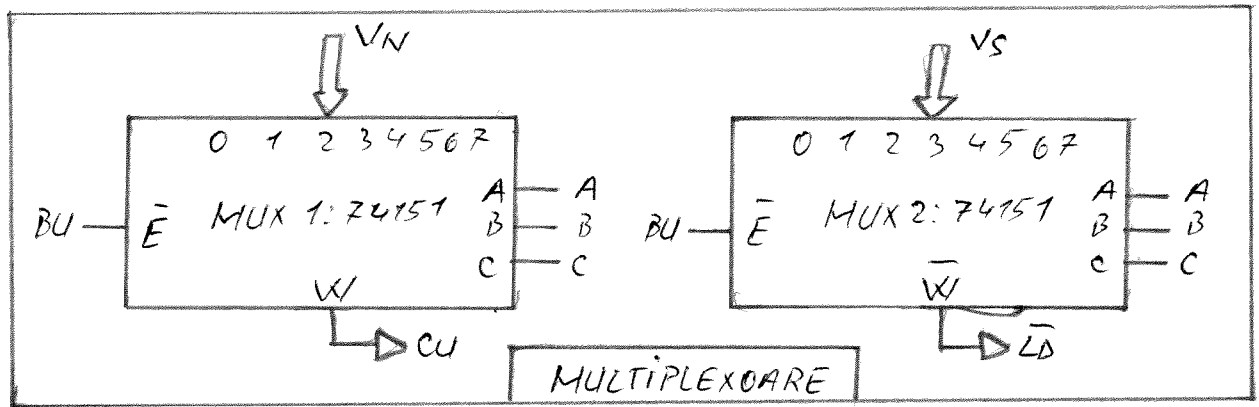




Expresiile comenzilor manuale:

$$\begin{cases} AVM = BAV \cdot \overline{LA} \cdot (L1 + LM + L2 + L3) \\ RETM = BRET \cdot \overline{LR} \cdot LE \\ RSM = BRS \cdot LR \cdot (\overline{LM} + \overline{L1}) \\ RDM = BRD \cdot LR \cdot (\overline{LM} + \overline{L2} + \overline{L3}) \\ PRM = BPR \cdot \overline{LP} \cdot LA \\ ELM = BEL \cdot \overline{LE} \end{cases}$$





### 3.4 Necesari materiale

Nr. circ.	Pondi logice							
	NOT	AND-2i	AND-3i	OR-2i	OR-3i	NAND-2i	NAND-3i	NAND-4i
1	-	4	2	1	-	12	4	-
2	-	-	-	-	-	-	-	-
3	-	-	-	1	-	3	1	4
4	9	-	-	-	-	-	-	-
5	-	1	5	1	2	-	-	-
6	-	-	-	-	-	-	-	-
7	6	-	-	-	-	-	-	-
8	-	8	-	6	-	-	-	-
TOTAL	15	13	7	9	2	15	5	4

Nr. circuit reprezintă blocurile logice reprezentate:

1. Funcții auxiliare
2. Numărător
3. Programare salturi
4. Intrări
5. Regim Manual
6. Multiplexoare
7. Decodificare stări
8. Ieșiri

Mai sunt necesare următoarele componente:

- Numărător binar, sincron, 4 biți, cu presetare, 74193 - 7 buc.
- Multiplexor de 3 biți, 74151 - 2 buc.
- Decodificator zecimal, 7442 - 1 buc.
- Buton impuls - 8 buc.
- Comutator cu 2 poziții - 1 buc.
- Rezistor de polarizare - 23 buc.
- Rezistor de limitare - 2 buc.
- Diode LED - 2 buc.



#### 4. Sinteza comenzii cu echipament AP MICRO

##### 4.1 Definirea și alocarea variabilelor procesului

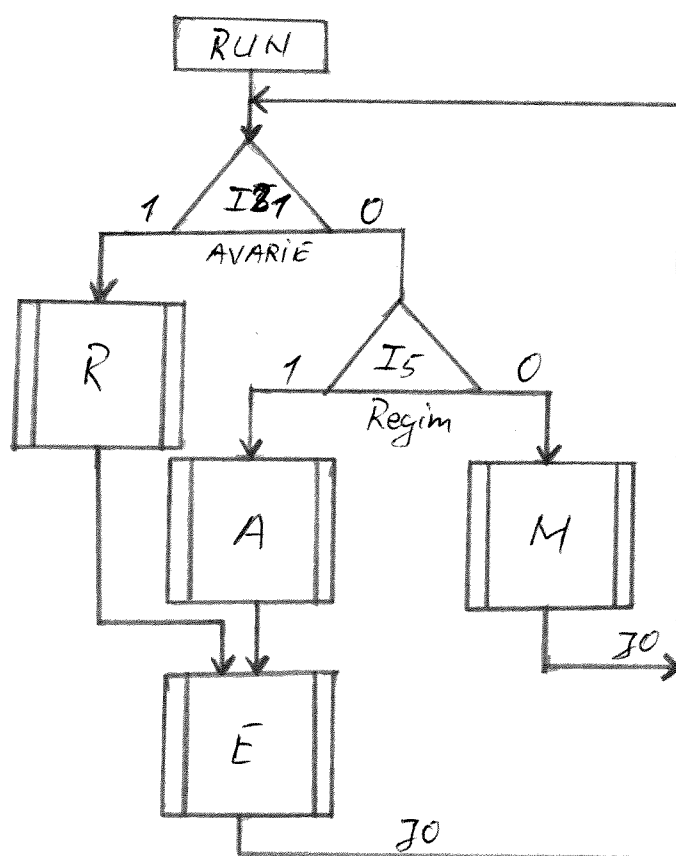
Intrări		
Variabilă AP MICRO	Variabilă Proces	Comentarii
I0	<del>START</del> P	Buton pornire ciclu automat
I1	PP	Prezență piesă
I2	MU1L	MU1 liberă
I3	MU2L	MU2 liberă
I4	MU3L	MU3 liberă
I5	SRL	Select. Regim
I6	LR	Limită retragere
I7	LA	Limită avans
I8	L1	Limită stâng (brat în față MU1)
I9	LM	Limită centru (brat în față M)
I10	L2	Limită dreapta (brat în față MU2)
I11	L3	Limită dreapta (brat în față MU3)
I12	LE	Limită <del>retragere</del> eliberare
I13	LP	Limită prindere
I14	BAV	Buton avans
I15	BRET	Buton retragere
I16	BRS	Buton rotire stânga
I17	BRD	Buton rotire dreapta
I18	BPR	Buton prindere
I19	BEL	Buton eliberare
I20	ANUL	Buton anulare semn.
I21	AVARIE	Oprire de urgență
I22	TP1	Confirmare piesă prelucrată la MU1
I23		
I24		
I25		
I26		
I27		
I28		
I29		
I30		
I31		

Teșini		
Variabilă AP MICRO	Variabilă Proces	Comentarii
E0	AV	Avans
E1	RET	Retnagene
E2	RS	Rotine stanga
E3	RD	Rotine dreapta
E4	PR	Prindene
E5	EL	Eliberare
E6	CMU1	Cd. START MU1
E7	CMU2	Cd. START MU2
E8	CMU3	Cd. START MU3
E9		
E10		
E11		
E12		
E13		
E14		
E15		
E16		
E17		
E18		
E19		
E20		
E21		
E22		
E23		



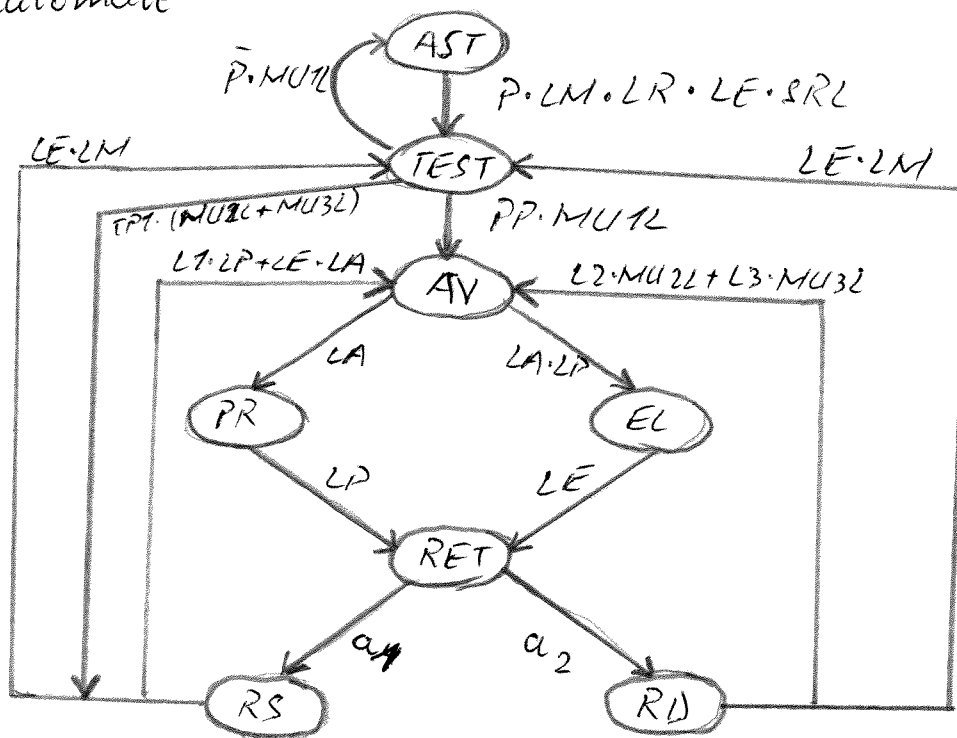
TEMPORIZĂRI		
T <sub>0</sub>	CT	Cd. START MU 1/2/3 (1s)
T <sub>1</sub>		
T <sub>2</sub>		
T <sub>3</sub>		
T <sub>4</sub>		
T <sub>5</sub>		
T <sub>6</sub>		
T <sub>7</sub>		
T <sub>8</sub>		
T <sub>9</sub>		
T <sub>32</sub>	OSC.	Semn. intermitentă

#### 4.2 Organizarea generală a programului



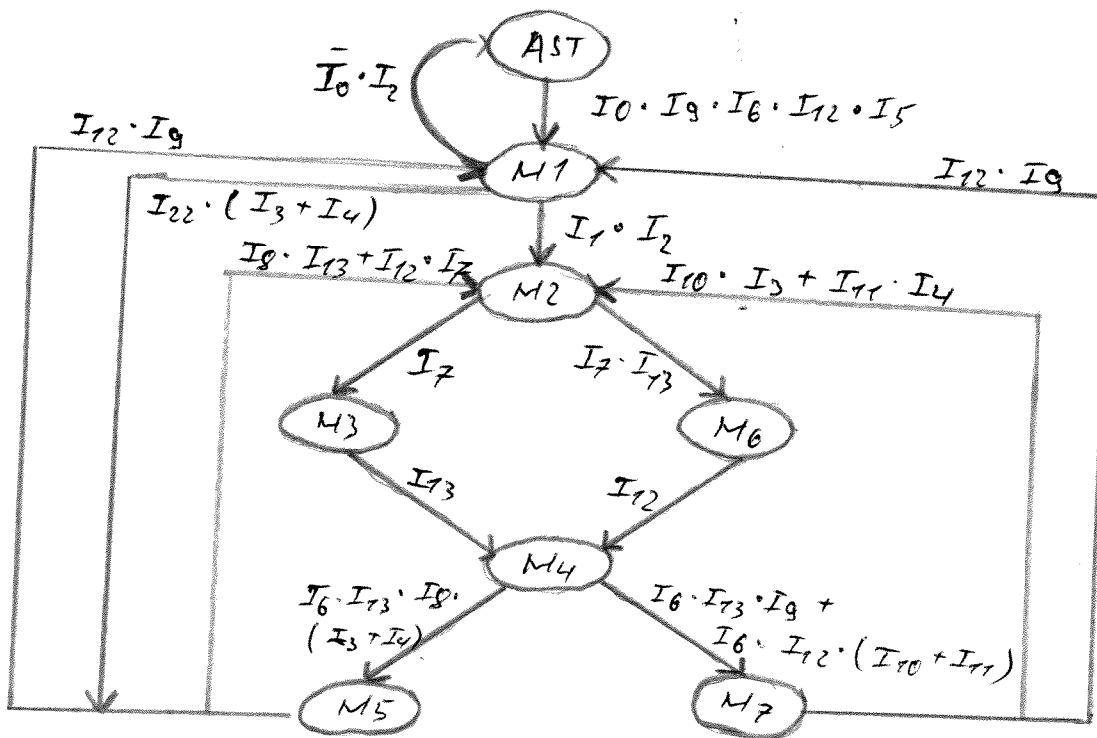
A - Program Automat  
 M - Program Manual  
 R - Program Reset  
 E - Program Ieșiri

### 4.3 Reprezentarea logică a programului funcționării automate



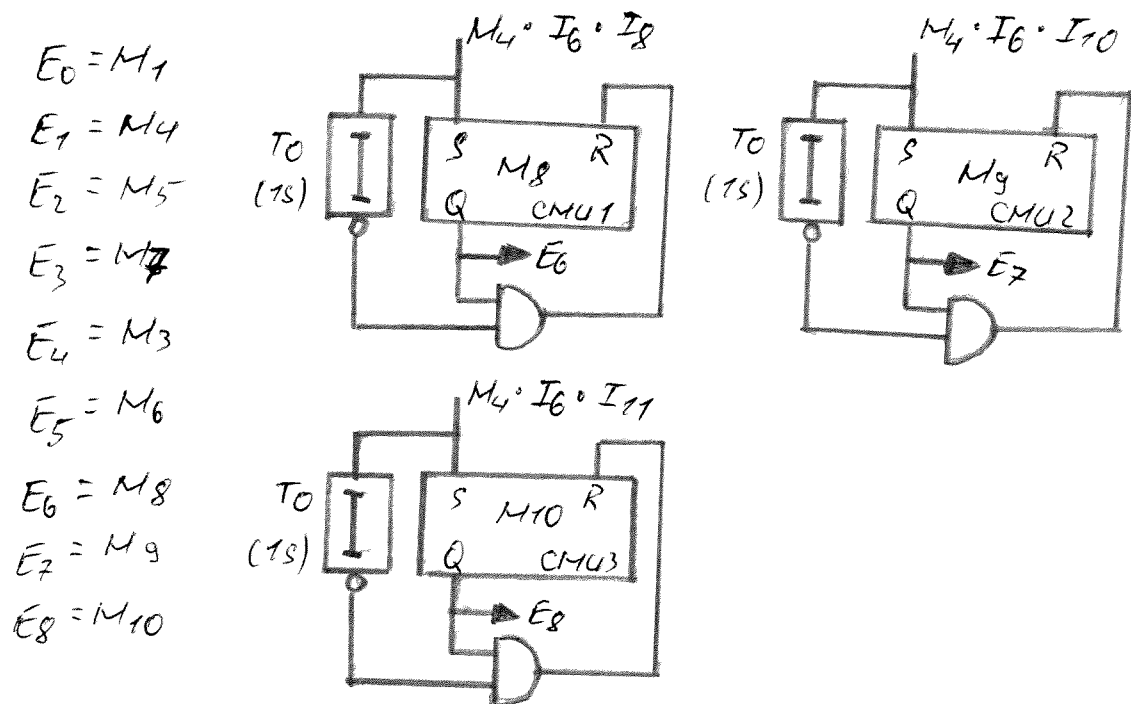
S-a notat:

$$\begin{cases} a_1 = LR \cdot LP \cdot LT \cdot (MU2L + MU3L) \\ a_2 = LR \cdot LP \cdot LM + LR \cdot LE \cdot (L2 + L3) \end{cases}$$



#### 4.4. Expresiile logice atașate programelor manual și de ieșire

##### 4.4.1 Program Ieșiri



##### 4.4.2 Program Manual

$$\begin{aligned}
 AV &= BAN \cdot \bar{L}A \cdot (L1 + LM + L2 + L3) \\
 RET &= BRET \cdot \bar{L}R \cdot LE \\
 RSM &= BRS \cdot \bar{L}R \cdot (\bar{L}M + \bar{L}7) \\
 RID &= BRD \cdot \bar{L}R \cdot (\bar{L}M + \bar{L}2 + \bar{L}3) \\
 PR &= BPR \cdot \bar{L}P \cdot LA \\
 EL &= BEL \cdot \bar{L}E \\
 E_0 &= I_{14} \cdot \bar{I}_7 \cdot (I_8 + I_9 + I_{10} + I_{11}) \\
 E_1 &= I_{15} \cdot \bar{I}_6 \cdot I_{12} \\
 E_2 &= I_{16} \cdot I_6 \cdot (\bar{I}_9 + \bar{I}_8) \\
 E_3 &= I_{17} \cdot I_6 \cdot (\bar{I}_9 + \bar{I}_{10} + \bar{I}_{11}) \\
 E_4 &= I_{18} \cdot \bar{I}_{13} \cdot I_7 \\
 E_5 &= I_{19} \cdot \bar{I}_{12}
 \end{aligned}$$



## 4.5 Program de lucru

Adn. MP	Operatie	Operand	Comentariu
0000	LD	I21	Test apăsare buton Avarie
1	JC	0090	Salt la Reset
2	LDC	I5	Test comutator regim lucru
3	JC	0150	Salt la program Manual
4	LDC	I0	
5	AND	I2	
6	JC	0150	Salt la manual
7	LD	I0	Test conditii initiale
8	AND	I6	
9	AND	I9	
0010	AND	I12	
1	AND	I5	
2	S	M1	Initiere ciclu Automat
3	LD	M1	Trecere M1 - M2
4	AND	I1	
5	AND	I2	
6	S	M2	
7	R	M1	
8	LD	I3	Trecere M1 - M5
9	OR	I4	
0020	AND	I22	
1	AND	M1	
2	S	M5	
3	R	M1	
4	LD	M2	Trecere M2 - M3
5	AND	I7	
6	S	M3	
7	R	M2	
8	LD	M2	Trecere M2 - M6
9	AND	I7	

0030	AND	I 13	
1	S	M6	
2	R	M2	
3	LD	M3	Trecene M3 - M4
4	AND	I13	
5	S	M4	
6	R	M3	
7	LD	I3	Trecene M4 - M5
8	OR	I4	
9	AND	I6	
0040	AND	I8	
1	AND	I13	
2	AND	M4	
3	S	M5	
4	R	M4	
5	LD	<del>I14</del>	Trecene M4 - M7
6	<del>AND</del>	I9	
7	AND	I13	
8	AND	I6	
9	STO	M4	
0050	LD	I11	
1	OR	I10	
2	AND	I12	
3	AND	I6	
4	OR	M4	
5	S	M7	
6	R	M4	
7	LD	M5	Trecene M5 - M7
8	AND	I9	
9	AND	I12	

0060	S	M1	
1	R	M5	
2	LD	M5	Trecere M5 - M2
3	AND	I8	
4	AND	I13	
5	STO	M5	
6	LD	I7	
7	AND	I12	
8	OR	M5	
9	S	M2	
0070	R	M5	
1	LD	M6	Trecere M6 - M4
2	AND	I12	
3	S	M4	
4	R	M6	
5	LD	M7	Trecere M7 - M1
6	AND	I9	
7	AND	I12	
8	S	M1	
9	R	M7	
0080	LD	M7	Trecere M7 - M2
1	AND	I3	
2	AND	I10	
3	STO	M7	
4	LD	I4	
5	AND	I11	
6	OR	M7	
7	S	M2	
8	R	M7	
9	J	O 100	Salt la program "Jesini"

0090	R	M <sub>1</sub>	Program Reset
1	R	M <sub>2</sub>	
2	R	M <sub>3</sub>	
3	R	M <sub>4</sub>	
4	R	M <sub>5</sub>	
5	R	M <sub>6</sub>	
6	R	M <sub>7</sub>	
7	R	M <sub>8</sub>	
8	R	M <sub>9</sub>	
9	R	M <sub>10</sub>	
0100	LD	M <sub>1</sub>	Program İşini
1	STO	E <sub>0</sub>	
2	LD	M <sub>4</sub>	
3	STO	E <sub>1</sub>	
4	LD	M <sub>5</sub>	
5	STO	E <sub>2</sub>	
6	LD	M <sub>7</sub>	
7	STO	E <sub>3</sub>	
8	LD	M <sub>3</sub>	
9	STO	E <sub>4</sub>	
0110	LD	M <sub>6</sub>	
1	STO	E <sub>5</sub>	
2	LD	M <sub>4</sub>	
3	AND	I <sub>6</sub>	
4	AND	I <sub>8</sub>	
5	S	M <sub>8</sub>	
6	STO	T <sub>0</sub>	
7	LD	M <sub>8</sub>	
8	ANNC	T <sub>0</sub>	
9	R	M <sub>8</sub>	



0120	LD	M8	
1	STO	E6	
2	LD	M4	
3	AND	I6	
4	AND	I10	
5	S	M9	
6	STO	T0	
7	LD	M9	
8	ANAC	T0	
9	R	M9	
0130	LD	M9	
1	STO	E7	
2	LD	M4	
3	AND	I6	
4	AND	I11	
5	S	M10	
6	STO	T0	
7	LD	M10	
8	ANAC	T0	
9	R	M10	
0140	LD	M10	
1	STO	E8	
2	J	O	Bucure program
3			Spatii libere in memoria
4			program
5			
6			
7			
8			
9			

0150	LD	I <sub>8</sub>	Program Manual
1	OR	I <sub>9</sub>	
2	OR	I <sub>10</sub>	
3	OR	I <sub>11</sub>	
4	ANBC	I <sub>7</sub>	
5	AND	I <sub>14</sub>	
6	STO	E <sub>0</sub>	
7	LDC	I <sub>6</sub>	
8	AND	I <sub>12</sub>	
9	AND	I <sub>15</sub>	
0160	STO	E <sub>1</sub>	
1	LDC	I <sub>8</sub>	
2	ORC	I <sub>9</sub>	
3	AND	I <sub>6</sub>	
4	AND	I <sub>17</sub>	
5	STO	E <sub>2</sub>	
6	LDC	I <sub>9</sub>	
7	ORC	I <sub>10</sub>	
8	ORC	I <sub>11</sub>	
9	AND	I <sub>6</sub>	
0170	AND	I <sub>17</sub>	
1	STO	E <sub>3</sub>	
2	LD	I <sub>7</sub>	
3	ANBC	I <sub>13</sub>	
4	AND	I <sub>18</sub>	
5	STO	E <sub>4</sub>	
6	LDC	I <sub>12</sub>	
7	AND	I <sub>19</sub>	
8	STO	E <sub>5</sub>	
9	J	O	Buckane Program

0180			<i>Spatii libere in memoria</i>
1			<i>program</i>
2			
3			
4			
5			
6			
7			
8			
9			
0190			
1			
2			
3			
4			
5			
6			
7			
8			
9			
0200			
1			
2			
3			
4			
5			
6			
7			
8			
9			

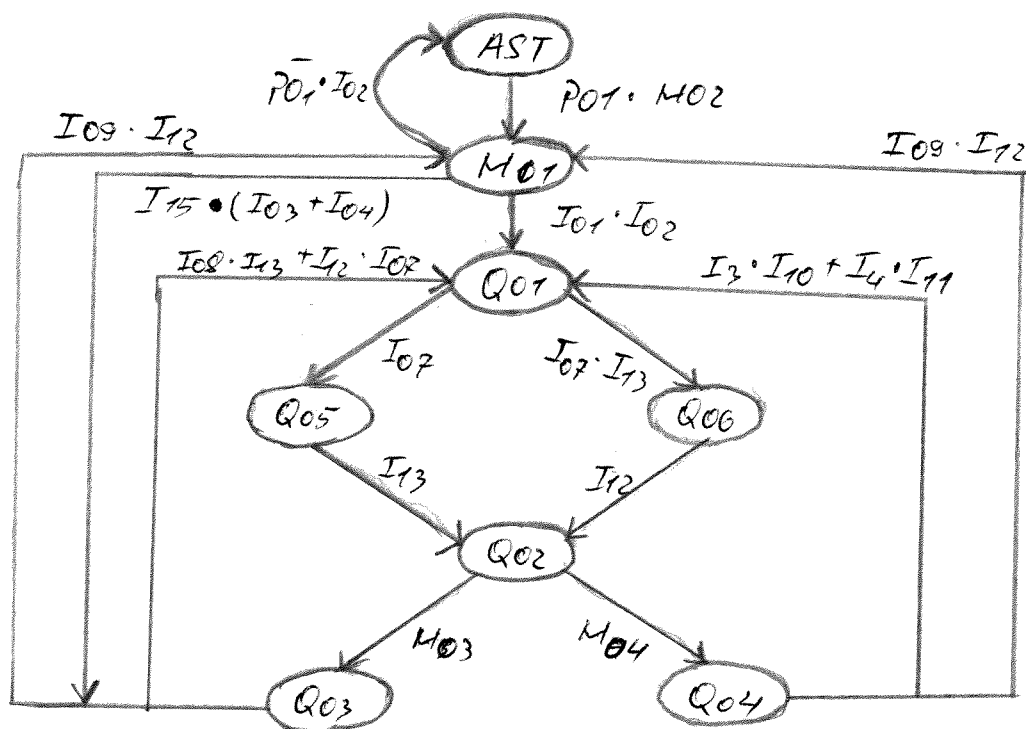
0210			
1			
2			
3			
4			
5			
6			
7			
8			
9			
0220			
1			
2			
3			
4			
5			
6			
7			
8			
9			
0230			
1			
2			
3			
4			
5			
6			
7			
8			
.....			
2047			

## 5. Sinteza comenzii cu Easy 600

### 5.1. Definirea și alocarea variabilelor procesului:

Variabilă EASY	Variabilă Proces	Comentarii
Intrări		
I01	PP	Prezentă piesă
I02	MU1L	MU1 liberă
I03	MU2L	MU2 liberă
I04	MU3L	MU3 liberă
I05	SRL	Select. A/M
I06	LR	Limită retragere
I07	LA	Limită avans
I08	L1	Limită stânga
I09	LM	Limită centru
I10	L2	Limită dreapta MU2
I11	L3	Limită dreapta MU3
I12	LE	Limită eliberare
I13	LP	Limită prindere
I14	AVARIE	Oprire de urgență
I15	TP1	Confirmare piesă preluc.
Ieșiri		
Q01	AV	Avans
Q02	RET	Retragere
Q03	RS	Rotire stânga
Q04	RD	Rotire dreapta
Q05	PR	Prindere
Q06	EL	Eliberare
S01	CMU1	Cd. START MU1
S02	CMU2	Cd. START MU2
S03	CMU3	Cd. START MU3
Butoane		
P01	START	Buton START ciclu
P02	ANUL	Buton anulare semn.
Timene		
T01	CT	Cd. START MU1/2/3 (7s)
T02	OSC	semn. intermitente.

## 5.2 Reprezentarea logică a programului:



S-au utilizat următorii markeri de memorie:

M01	Stare de TEST	-
M02	Condiție normală pentru lansare	$M02 = I_{05} \cdot I_{06} \cdot I_{09} \cdot I_{12}$
M03	Condiție trecere stare RET-RS	$M03 = I_{06} \cdot I_{08} \cdot (I_{03} + I_{04})$
M04	Condiție trecere stare RET-RD	$M04 = I_{06} \cdot I_{08} \cdot \dots$

$$M02 = I_{05} \cdot I_{06} \cdot I_{09} \cdot I_{12}$$

$$M03 = I_{06} \cdot I_{08} \cdot (I_{03} + I_{04}) \cdot I_{13}$$

$$M04 = I_{06} \cdot (I_{13} \cdot I_{09} + I_{12} \cdot (I_{10} + I_{11}))$$

S04 - Lipsă piesă.







Informatii aparat

Documentatie realizată cu: easySoft-Pro V6.98, Build 5425

Desemnare dispozitiv: EASY 619-DC-RC

I debounce: Oprit

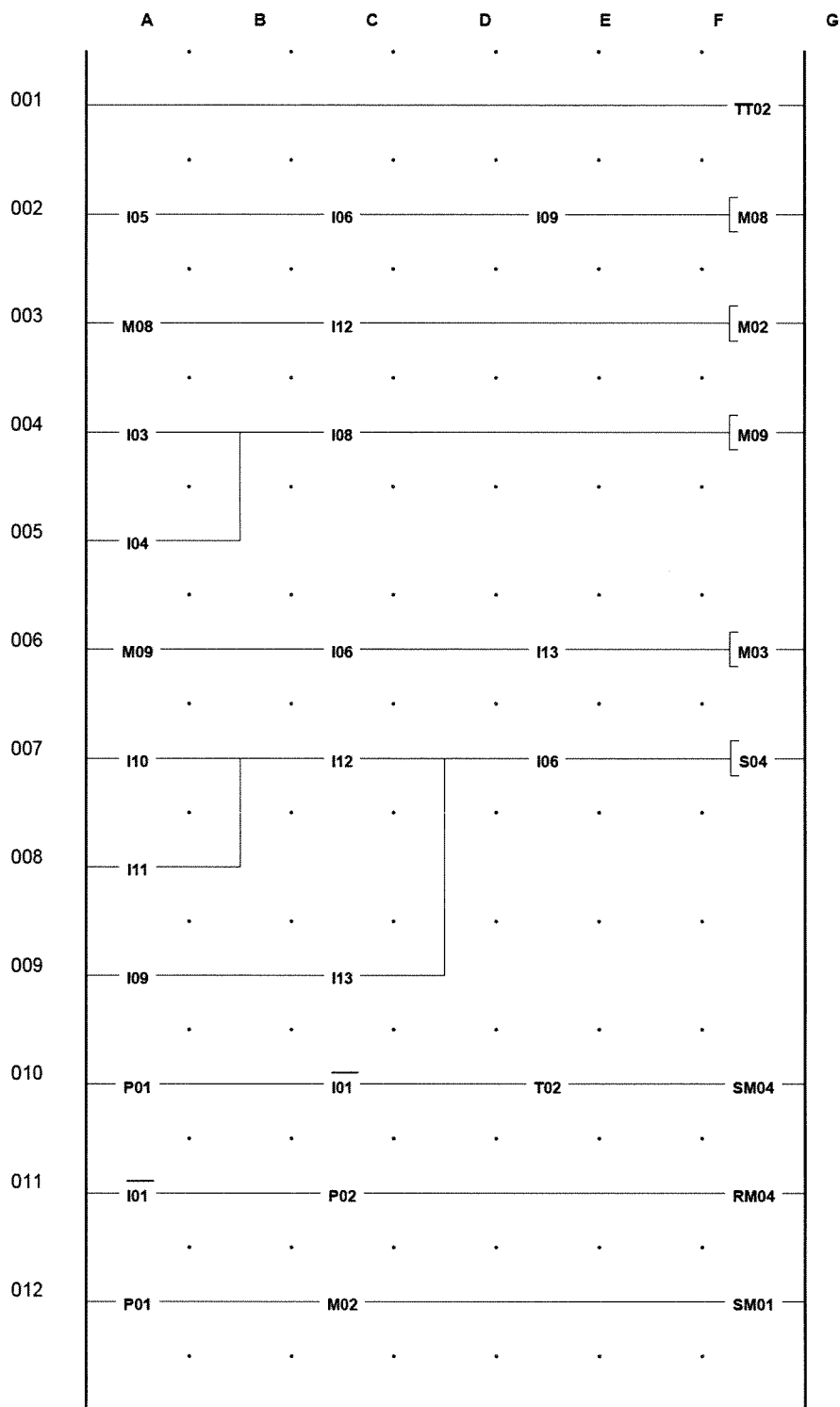
Taste-P: Pornit

Remanenta: Oprit

Parola:: Oprit

Data:	5/26/2025	Client:	Numar comanda:	10
Autor:	I. Chiriac	Cod client:	Numar fabrica:	
Verificat:	A. Pricop		Numar subscriere:	
				Prodicator:
				Pagina: 1 de la 7
				NET-ID: -

## Circuit Diagram



Data: 5/26/2025

Autor: I. Chiriac

Verificat: A. Pricop

Client:

Cod client:

Numar comanda: 10

Numar fabrica:

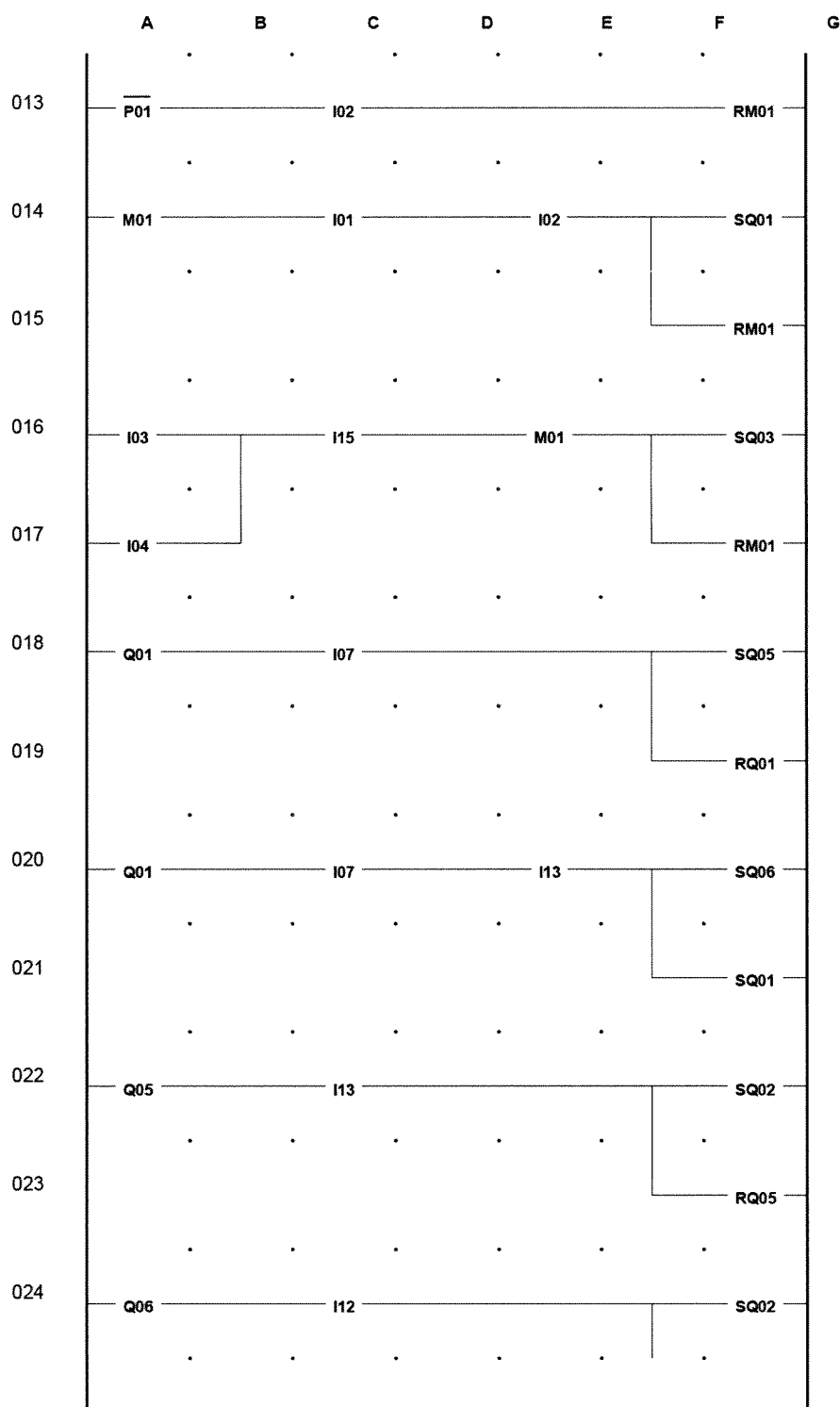
Numar subscriere:

Prodicator:

Pagina: 2 de la 7

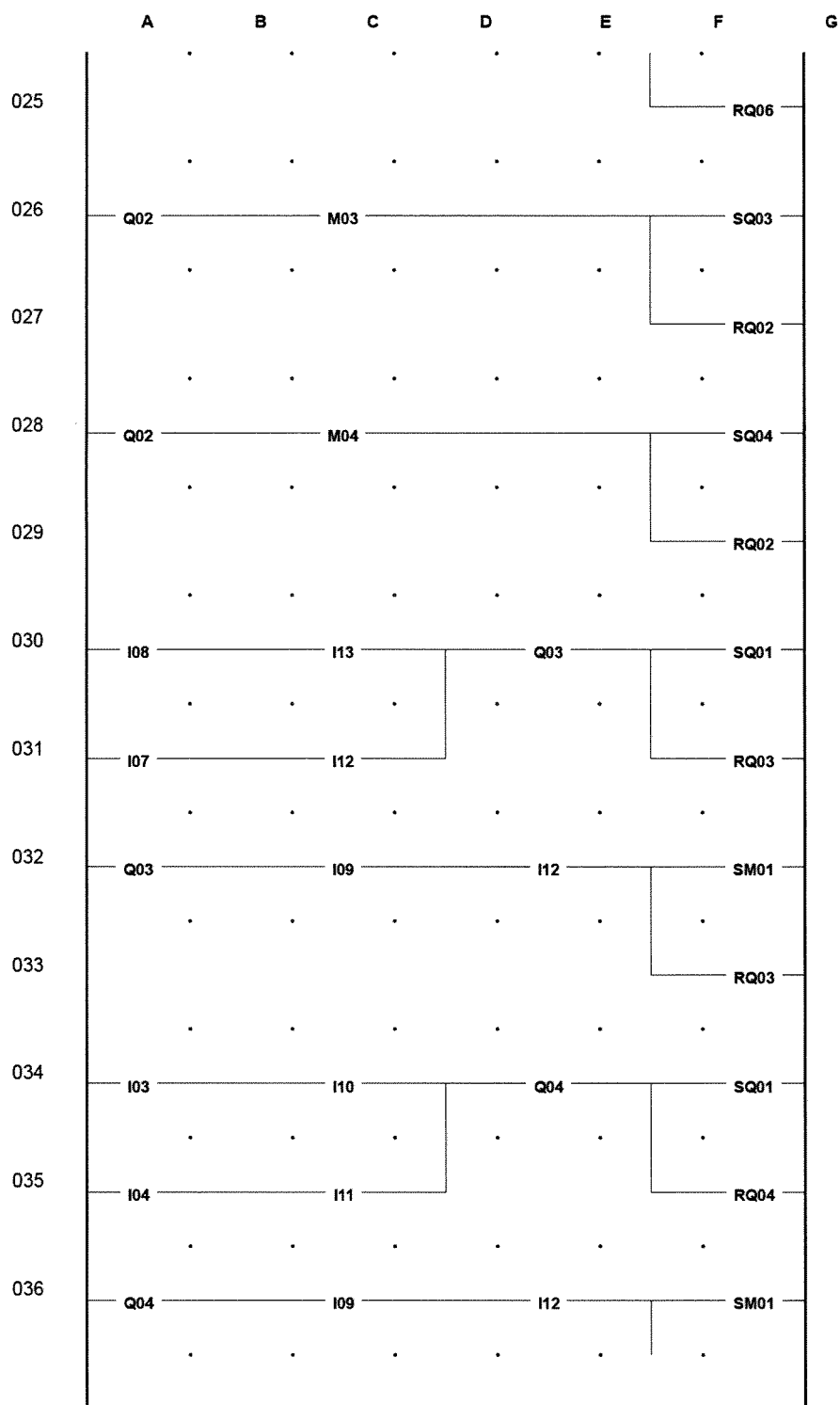
NET-ID: -

## Circuit Diagram



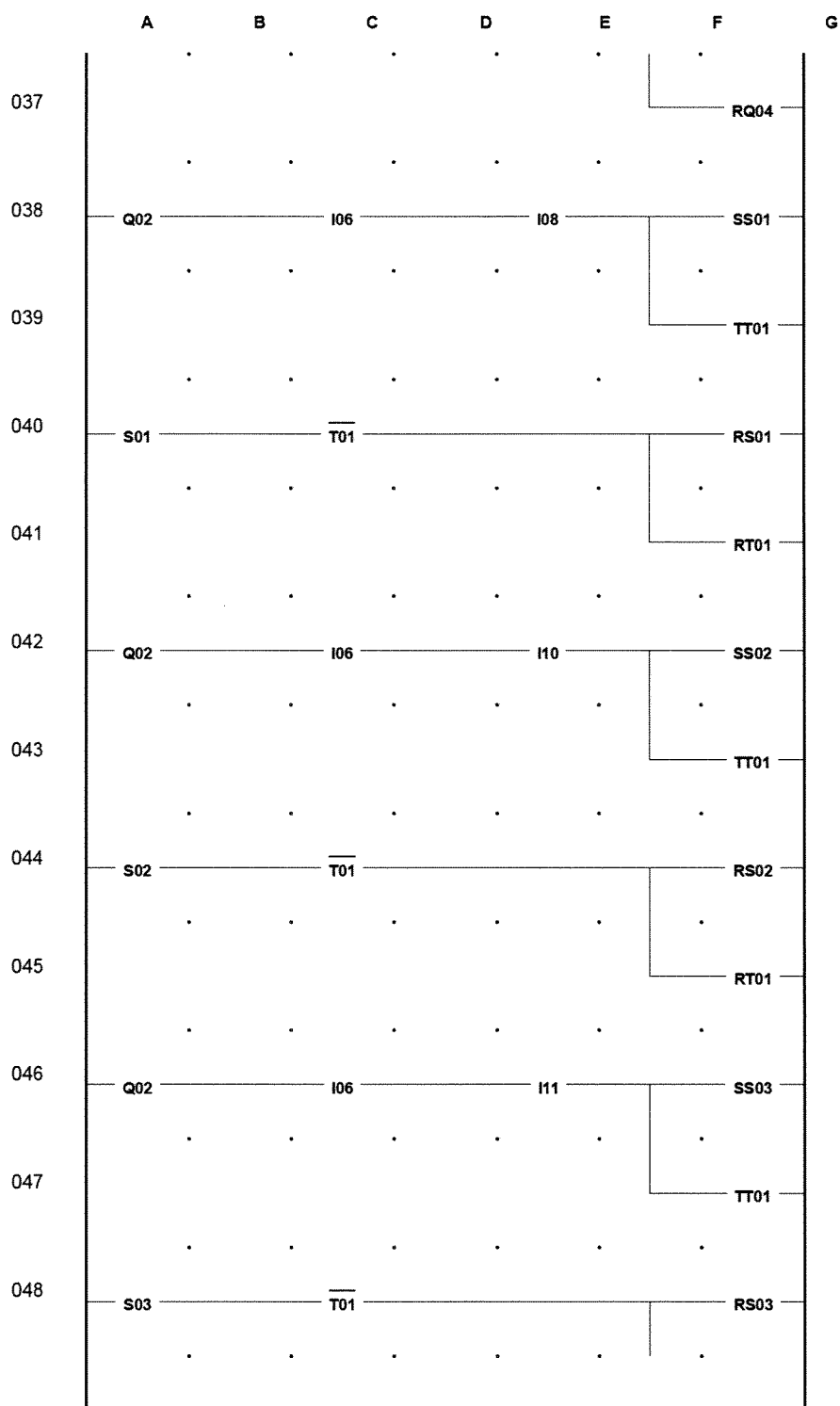
Data:	5/26/2025	Client:	Numar comanda:	10
Autor:	I. Chiriac	Cod client:	Numar fabrica:	
Verificat:	A. Pricop		Numar subscriere:	
				Prodicator:
				Pagina: 3 de la 7
				NET-ID: -

## Circuit Diagram



Data:	5/26/2025	Client:	Numar comanda:	10
Autor:	I. Chiriac	Cod client:	Numar fabrica:	
Verificat:	A. Pricop		Numar subscriere:	
				Prodicator:
				Pagina: 4 de la 7
				NET-ID: -

## Circuit Diagram



Data:	5/26/2025	Client:	Numar comanda:	10
Autor:	I. Chiriac	Cod client:	Numar fabrica:	
Verificat:	A. Pricop		Numar subscriere:	
				Prodicator:
				Pagina: 5 de la 7
				NET-ID: -

## Circuit Diagram

	A	B	C	D	E	F	G
049	.	.	.	.	.	.	
						RT01	
050	.	.	.	.	.	.	
051	.	.	.	.	.	.	
052	.	.	.	.	.	.	
053	.	.	.	.	.	.	
054	.	.	.	.	.	.	
055	.	.	.	.	.	.	
056	.	.	.	.	.	.	
057	.	.	.	.	.	.	
058	.	.	.	.	.	.	
059	.	.	.	.	.	.	
060	.	.	.	.	.	.	

Data:	5/26/2025	Client:	Numar comanda:	10
Autor:	I. Chiriac	Cod client:	Numar fabrica:	
Verificat:	A. Pricop		Numar subscriere:	
			Prodicator:	
			Pagina: 6 de la 7	
			NET-ID: -	

## Function block parameters

T 1:

Parameter display = On

Timing relay type = On-delayed

Setpoint = 1 s . 0 ms

T 2:

Parameter display = On

Timing relay type = Flashing

Setpoint = 1 s . 0 ms

Data:	5/26/2025	Client: Cod client:	Numar comanda:	10
Autor:	I. Chiriac		Numar fabrica:	
Verificat:	A. Pricop		Numar subscriere:	
				Prodicator:
				Pagina: 7 de la 7
				NET-ID: -

