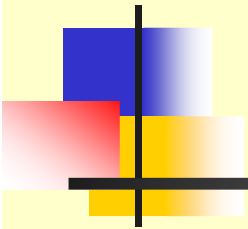
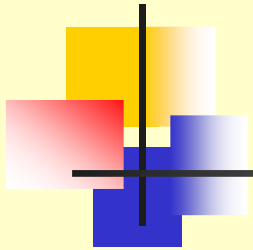


5.5.4. AUTOMAT DE RĂSPUNS LA TELEFON

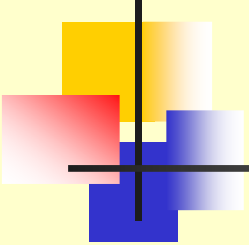




5.5.4. Enunțul problemei

- Să se realizeze sinteza unui automat (circuit secvențial sincron) de răspuns la telefon
 - Se poate programa numărul de apeluri sonore ale soneriei telefonului după care începe să funcționeze automatul
 - Se comandă redarea unui mesaj înregistrat
 - Se comandă înregistrarea mesajului apelantului
 - Programul automatului se încheie în condițiile:
 - După preluarea (înregistrarea) mesajului
 - Dacă apelantul închide
 - Dacă destinatarul răspunde la telefon

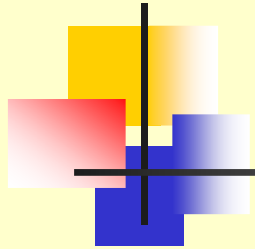
5.5.4.1. Determinarea funcționării

- 
- Se stabilește **schema bloc** a automatului, cu componentele periferice adiționale necesare
 - Componentele adiționale:
 - TEL – telefon propriu-zis
 - NRT – numărător pentru numărul apelurilor sonore ale soneriei telefonului
 - PLAY – casetofon pentru redarea mesajului de întâmpinare înregistrat
 - REC – casetofon pentru înregistrarea mesajului apelantului

5.5.4.1. Determinarea funcționării

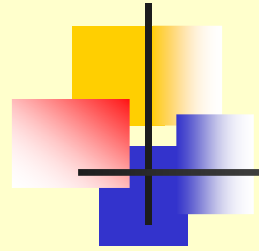
- Se stabilesc **variabilele de intrare și ieșire** și caracterul lor sincron sau asincron
- Variabilele folosite:
 - **Sonerie**
 - Fiecare apel al soneriei telefonului provoacă decrementarea număratorului NRT până se ajunge la valoarea 0
 - Nu este variabilă a automatului!!!
 - **Start**
 - Variabilă de intrare asincronă, de la numărător
 - Determină începerea funcționării automatului, dacă numărătorul a ajuns la valoarea 0
 - **SP** (StartPlay), **SR** (StartRecord)
 - Variabile de ieșire spre casetofon
 - **EP** (EndOfPlay), **ER** (EndOfRecord)
 - Variabile de intrare asincrone, de la casetofon

5.5.4.1. Determinarea funcționării



- Variabilele folosite:
 - **AS** (ApelantStop)
 - Variabilă de intrare asincronă de la telefon (apelantul poate închide oricând telefonul)
 - **DPU** (DestinatarPick-Up)
 - Variabilă de intrare asincronă de la telefon
 - Apare când destinatarul răspunde la telefon
 - **Init**
 - Variabilă de ieșire spre numărător
 - Încarcă paralel numărătorul cu valoarea stabilită pentru numărul de apeluri ale soneriei telefonului până la intrarea în funcțiune a automatului

5.5.4.1. Determinarea funcționării

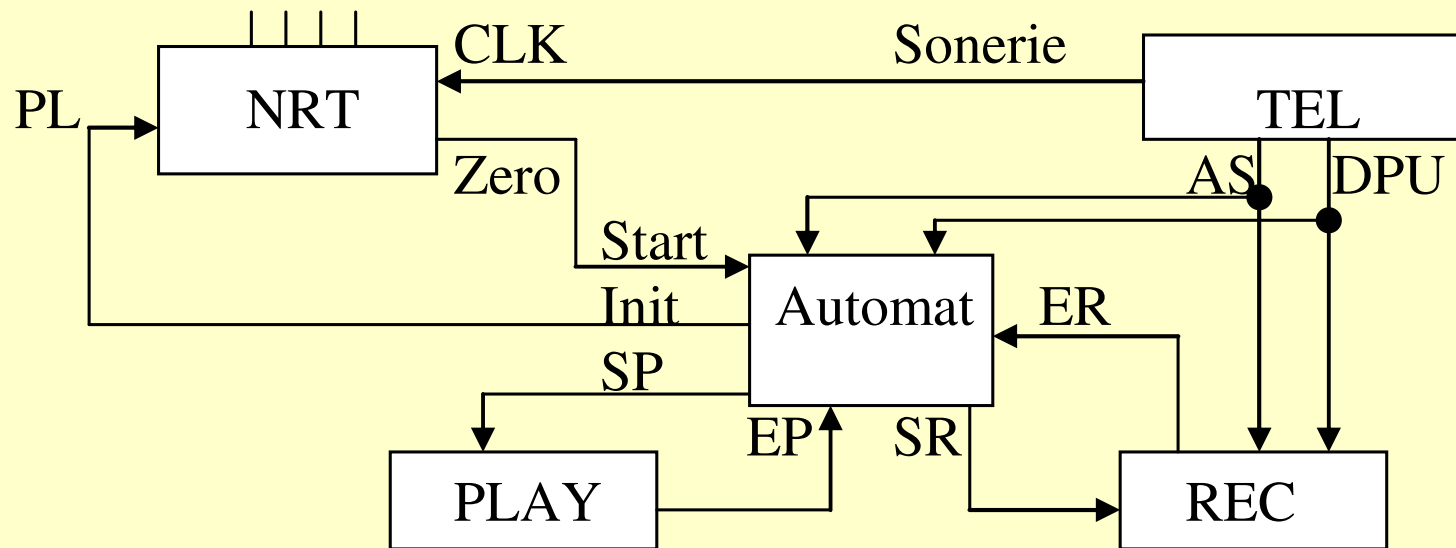


■ Observație:

- Variabilele de intrare asincrone **AS** și **DPU** generează semnalul de **Reset** pentru bistabilii interni ai automatului și opresc înregistrarea pe casetofon

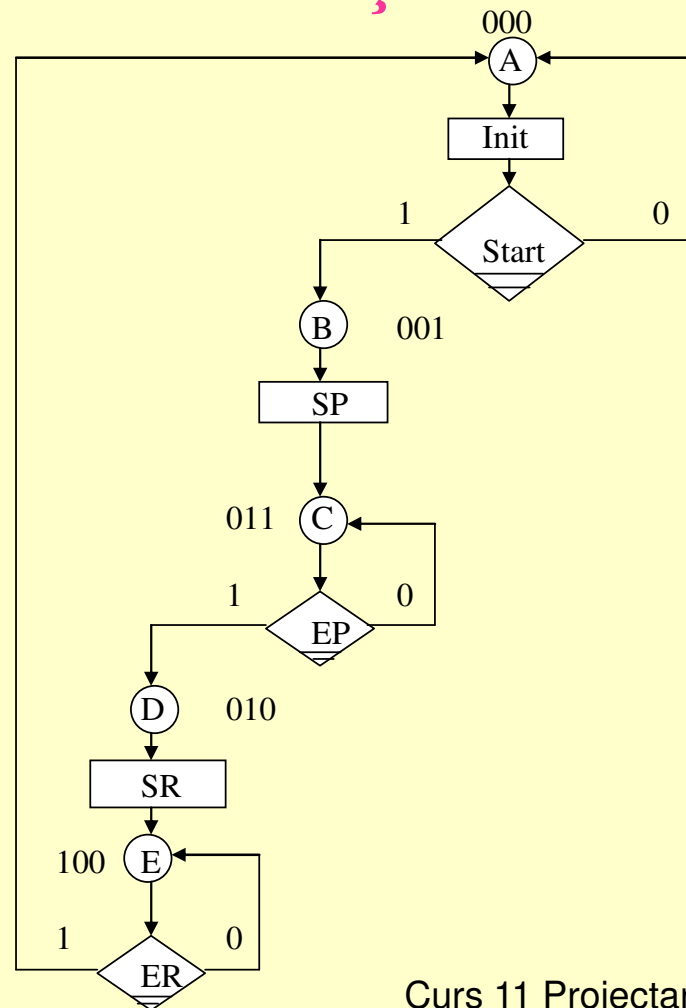
5.5.4.1. Determinarea funcționării

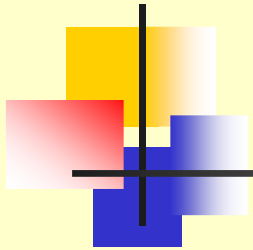
■ Schema bloc



5.5.4.1. Determinarea funcționării

■ Organigrama funcționării





5.5.4.2. Codificarea stărilor

- Deoarece **variabilele de intrare** sunt **asincrone** \Rightarrow pot apărea tranziții false între stări \Rightarrow pentru evitarea lor este nevoie de o **codificare adiacentă** a stărilor automatului: (A,B), (C,D) și (E,A)
- Trebuie codificate 5 stări \Rightarrow 3 variabile de stare
- **Se alege** următoarea codificare, care respectă condițiile de adiacență impuse:
 - A = 000; B = 001; C = 011; D = 010; E = 100
- Nu se poate reduce numărul de stări



5.5.4.2. Codificarea stărilor

- Pe baza codificării făcute se construiește **diagrama Karnaugh pentru stări**

$Q_2 \backslash Q_1 Q_0$	00	01	11	10
0	A	B	C	D
1	E			

- Plecând de la această diagramă a stărilor trebuie obținute cele 3 diagrame pentru stările următoare
- Vom suprapune diagramele pentru starea următoare și vom desena o singură diagramă, înglobând în ea și variabilele de intrare

5.5.4.2. Codificarea stărilor

- Completarea se face urmărind tranzițiile din organigramă și completând în compartimentul fiecărei stări codul stării următoare
- Locațiile necompletate vor fi indiferente, deoarece conținutul lor nu poate fi atins prin funcționare
- Diagramele pentru stările următoare:

$Q_2 \backslash Q_1 Q_0$	00	01	11	10
0	00Start	011	01 \overline{EP}	100
1	\overline{ER} 00	xxx	xxx	xxx

5.5.4.3. Registrul de stări

- Implementarea registrului de stări se face cu bistabile D
- Generatorul noii stări se realizează cu porți logice
- Diagramele Karnaugh ne dau funcțiile pentru intrările bistabilelor D

D_2 :

$Q_2 \backslash Q_1 Q_0$	00	01	11	10
0	0	0	0	1
1	ER	x	x	x

$$D_2 = \overline{ER} \cdot Q_2 + Q_1 \cdot \overline{Q_0}$$

D_1 :

$Q_2 \backslash Q_1 Q_0$	00	01	11	10
0	0	1	1	0
1	0	x	x	x

$$D_1 = Q_0$$

D_0 :

$Q_2 \backslash Q_1 Q_0$	00	01	11	10
0	Start	1	EP	0
1	0	x	x	x

$$D_0 = \text{Start} \cdot \overline{Q_2} \cdot \overline{Q_1} + \overline{EP} \cdot Q_0 + \overline{Q_1} \cdot Q_0$$

5.5.4.4. ieșirile

- Diagramele Karnaugh pentru **ieșiri** se completează ținând cont de organigramă și de diagrama Karnaugh pentru stări

Init:

$Q_2 \backslash Q_1 Q_0$	00	01	11	10
0	1			
1		x	x	x

$$\text{Init} = \overline{Q_2} \cdot \overline{Q_1} \cdot \overline{Q_0}$$

SP:

$Q_2 \backslash Q_1 Q_0$	00	01	11	10
0		1		
1		x	x	x

$$\text{SP} = \overline{Q_1} \cdot Q_0$$

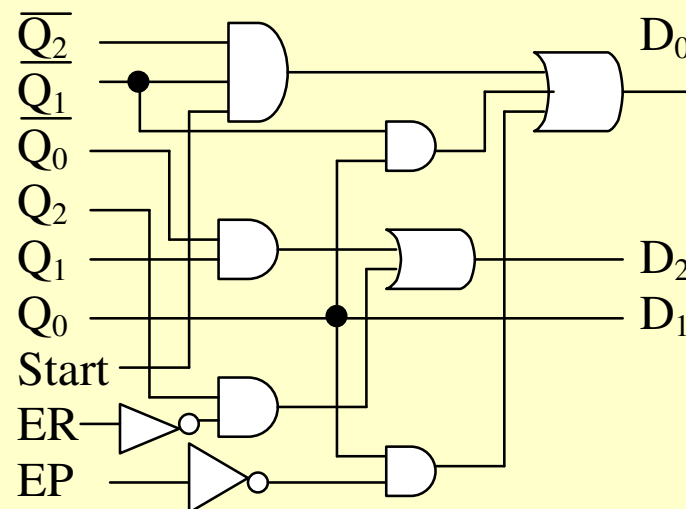
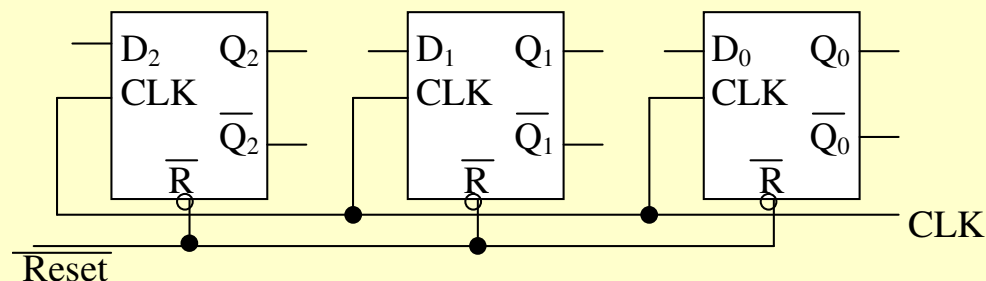
SR:

$Q_2 \backslash Q_1 Q_0$	00	01	11	10
0				1
1		x	x	x

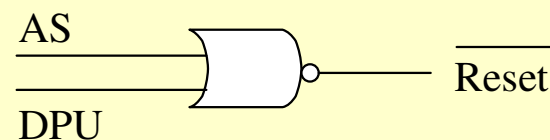
$$\text{SR} = Q_1 \cdot \overline{Q_0}$$

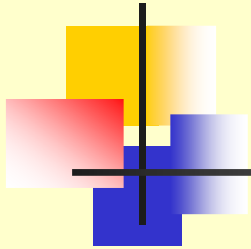
5.5.4.4. Schema circuitului

- Registrul de stări este cu bistabile de tip D
- Funcțiile de excitație secundare D se implementează cu porți logice
- Generarea semnalului de **Reset**, activ pe 0 logic, necesar pentru inițializarea bistabililor registrului de stări, se realizează cu o poartă logică SAU-NU



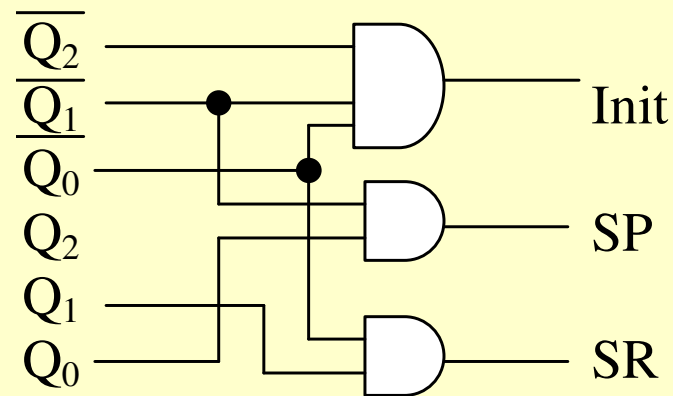
$$\overline{\text{Reset}} = \overline{\text{AS}} + \overline{\text{DPU}}$$





5.5.4.4. Schema circuitului

- Ieșirile se realizează cu porți logice de tip ȘI

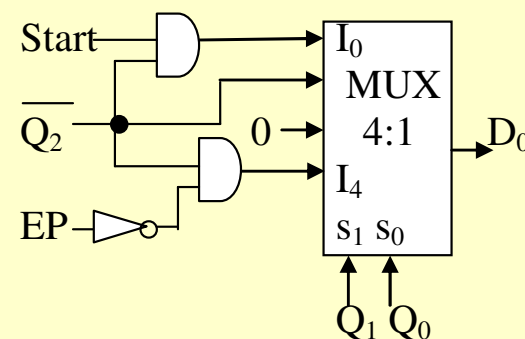
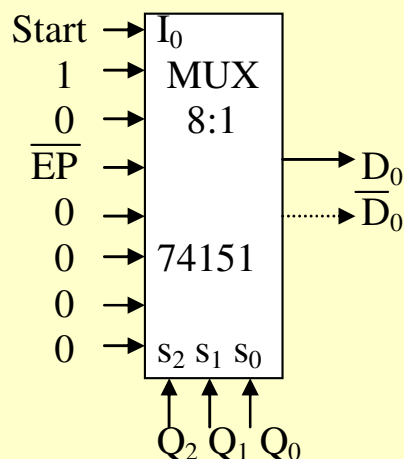
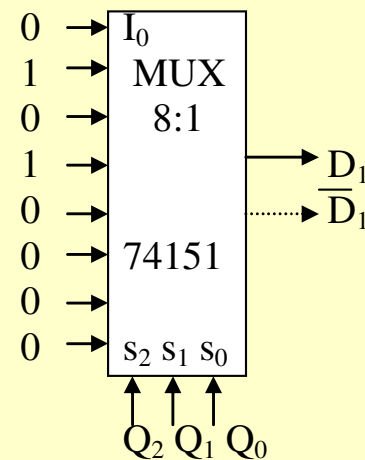
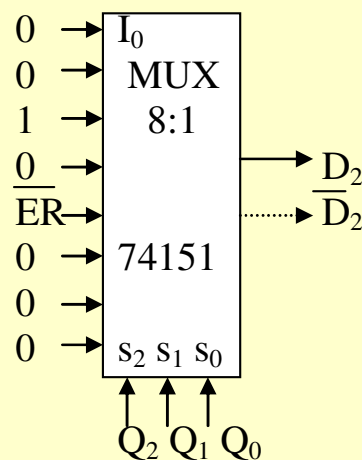


5.5.4.5. Generatorul noii stări cu multiplexoare

- Se poate realiza NUMAI cu multiplexoare dacă numărul intrărilor de selecție a multiplexorului este egal cu numărul variabilelor de stare
- Dacă numărul intrărilor de selecție este mai mic decât cel al variabilelor de stare, la intrările selectate se vor conecta circuite realizate cu porți logice
- Pentru excitațiile secundare se scriu termenii canonici care în diagrama Karnaugh sunt 1 sau conțin variabile înglobate

5.5.4.5. Generatorul noii stări cu multiplexoare

- Implementare cu MUX 8:1
- Implementarea cu MUX 4:1 \Rightarrow pe unele intrări ale multiplexorului am avea ieșiri din porți logice (de exemplu, pentru intrarea D_0)



5.5.4.6. Generatorul noii stări cu decodificatoare

- La intrarea decodicatorului se aplică variabilele de stare
- La ieșirile decodicatorului se obțin stările interne individualizate
- Utilizăm diagrama pentru stări și cele pentru stările următoare
- Pentru obținerea funcțiilor de excitație se vor utiliza porți logice de tip ȘI, SAU și NU

Q₂ Q₁ Q₀	00	01	11	10
0	A	B	C	D
1	E			

Q₂ Q₁ Q₀	00	01	11	10
0	00Start	011	01 $\overline{E}P$	100
1	$\overline{E}R00$	xxx	xxx	xxx

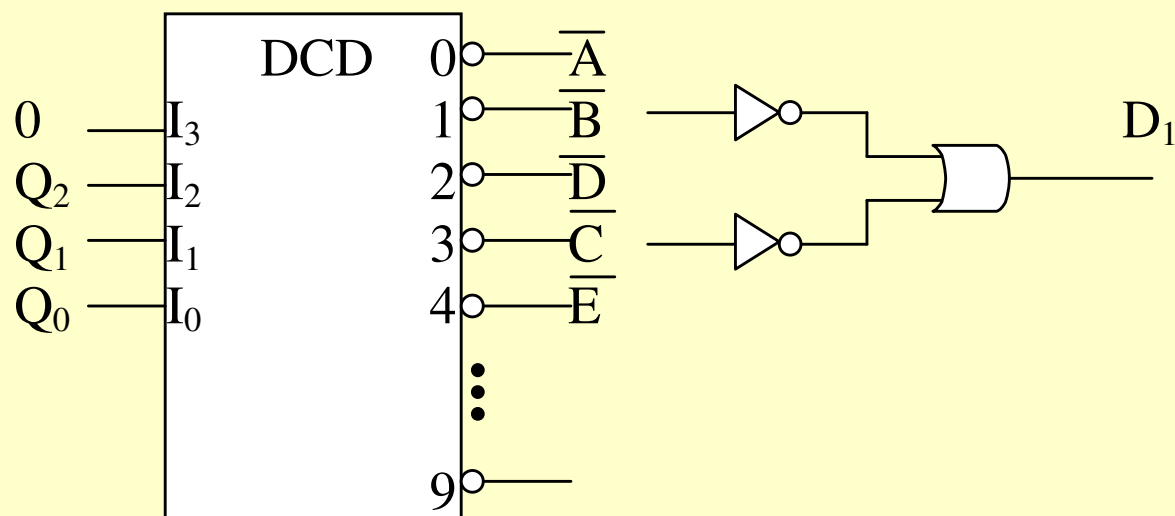
$$D_2 = D + \overline{E}R \cdot E$$

$$D_1 = B + C$$

$$D_0 = \text{Start} \cdot A + B + \overline{E}P \cdot C$$

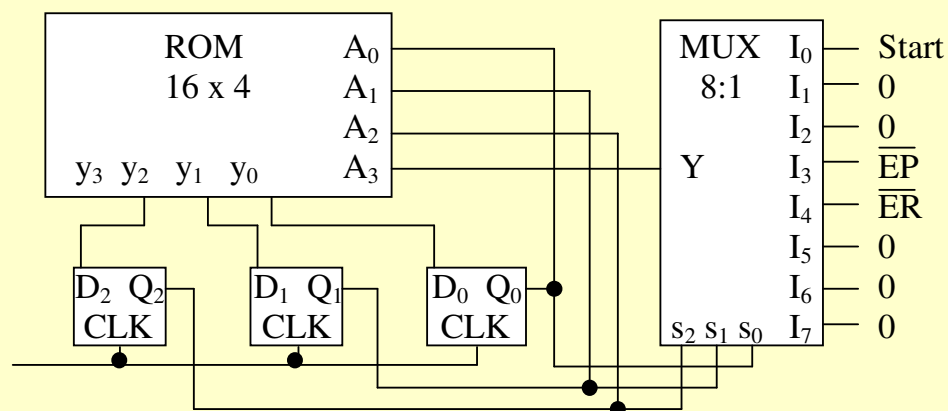
5.5.4.6. Generatorul noii stări cu decodificatoare

- Decodicatorul utilizat va fi de tip zecimal



5.5.4.7. Generatorul noii stări cu memorii și MUX

- Utilizarea memoriilor permite simplificarea logicii generării noii stări



$Q_2Q_1Q_0$ ($s_2s_1s_0$)	$A_3A_2A_1A_0$	$y_3y_2y_1y_0$
000	0000	0000
001	0001	0011
010	0010	0100
011	0011	0010
100	0100	0000
101	0101	0000
110	0110	0000
111	0111	0000

$Q_2Q_1Q_0$ ($s_2s_1s_0$)	$A_3A_2A_1A_0$	$y_3y_2y_1y_0$
000	1000	0001
001	1001	0000
010	1010	0000
011	1011	0011
100	1100	0100
101	1101	0000
110	1110	0000
111	1111	0000

5.5.4.8. Registrul de stare cu numărător

- Numărătorul va fi utilizat pentru funcția de memorare și parțial pentru efectuarea tranzițiilor
- Vom utiliza pentru implementare un numărător sincron (de exemplu, numărătorul zecimal 74162) - acesta poate memora starea circuitului
- Codificarea stărilor se face ținând cont de ordinea de numărare a numărătorului: A = 000; B = 001; C = 010; D = 011; E = 100
- Cu această codificare diagrama Karnaugh pentru stări este:

Q₂ Q ₁ Q ₀	00	01	11	10
0	A	B	D	C
1	E			

5.5.4.8. Registrul de stare cu numărător

- Determinăm funcțiile de numărare f_N și de încărcare (ramificare) f_R și realizăm implementarea lor cu multiplexoare 8:1 (de tipul 74151)
- Intrările de selecție ale MUX vor fi ieșirile numărătorului, $Q_2Q_1Q_0$
- Urmărind organigrama de funcționare, diagramele Karnaugh pentru cele 2 funcții vor fi:

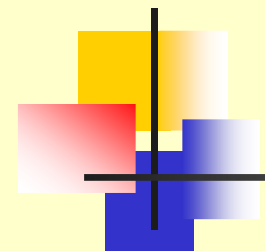
f_N :

$Q_2 \backslash Q_1 Q_0$	00	01	11	10
0	Start	1	1	EP
1				

f_R :

$Q_2 \backslash Q_1 Q_0$	00	01	11	10
0				
1	ER			

5.5.4.8. Registrul de stare cu numărător



- Stările următoare ale numărătorului trebuie specificate numai pentru stările din care au loc ramificări
- Din starea E se face salt la starea A, deci de la codul 100 (E) se trece la codul 000 (A)
- Pentru a obține pe ieșire codul lui A (000) se modifică doar intrarea D_2 de date paralele a numărătorului: $D_2 = \overline{Q_2}$; $D_1 = Q_1$; $D_0 = Q_0$ și se utilizează încărcarea paralelă a numărătorului (obținem un numărător modulo 5)

5.5.4.8. Registrul de stare cu numărător

- Schema circuitului trebuie completată și cu logica pentru determinarea ieșirilor
- Ieșirile Init, SP și SR corespund stărilor A, B, D
- Funcțiile de ieșire sunt:
$$\text{Init} = \overline{Q_2} \cdot \overline{Q_1} \cdot \overline{Q_0}$$
$$\text{SP} = \overline{Q_1} \cdot Q_0$$
$$\text{SR} = Q_1 \cdot Q_0$$
- Implementarea funcțiilor de ieșire se poate realiza cu porți logice de tip ȘI sau cu un decodificator zecimal

5.5.4.8. Registrul de stare cu numărător

- Automatul de răspuns la telefon are următoarea implementare:

