

# Noțiuni de proiectare a automatelor

---

*Prof. Sorin Larionescu*

*Regulatorul direct centrifugal Watt - 1769*

*Modelul matematic Maxwell - 1868*

*Amplificatorul cu reacție Black, Bode - 1927*

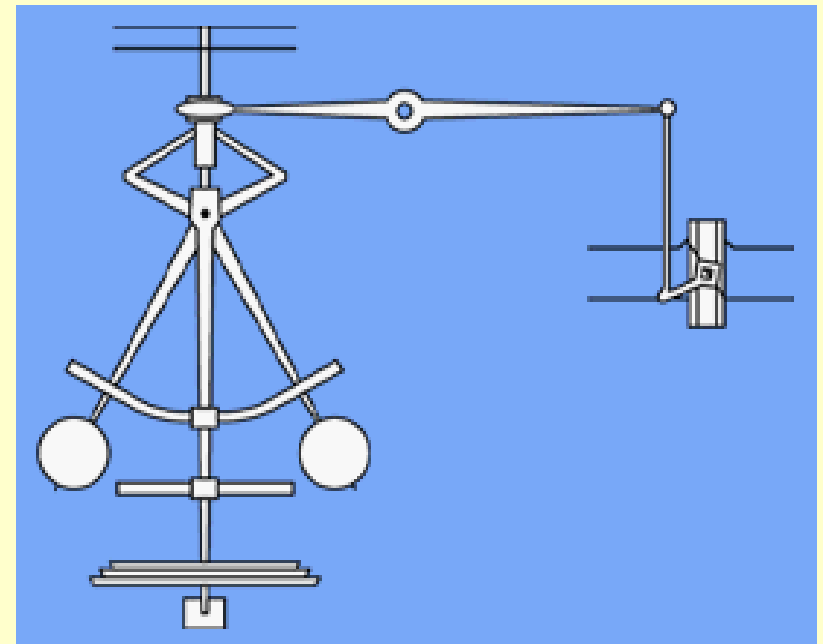
*Regulatorul PID - 1930*

*Teorema de stabilitate Nyquist - 1932*

*Aplicarea transformării Laplace ~ 1950*

*Modelul variabilelor de stare, optimizare ~ 1970*

*Conducerea automata robustă ~ 1980*



# Un sistem este cu conducere automată dacă:

---

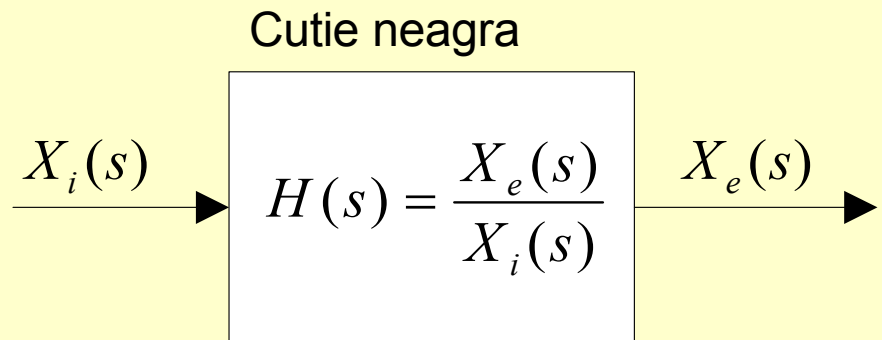
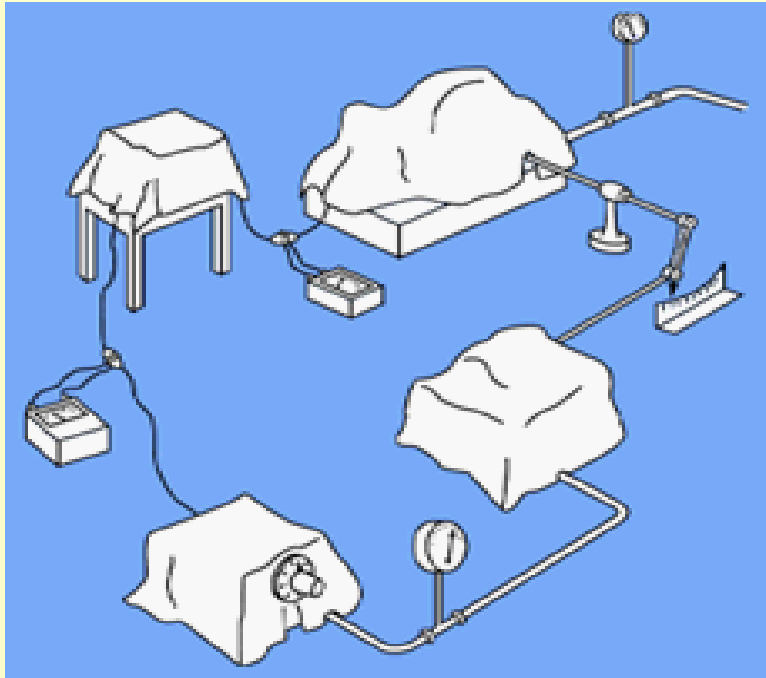
- Folosește metoda memorării.
  - *Elementul conducător se numește automat.*
  - *Automatul este implementat cu contacte și relee, cu automat programabil, cu came și palpatoare, etc.*
- Folosește metoda legăturii inverse.
  - *Elementul conducător se numește regulator. Regulatorul are cel puțin un comparator și un compensator.*
  - *Regulatorul este realizat cu circuite electronice, cu circuite fluidice sau cu calculatoare.*

# Sistemul liniar satisface principiul superpoziției și proprietatea de omogenitate

---

- Cele mai multe sisteme fizice sunt liniare în anumite domenii ale variabilelor (pe porțiuni)
- Majoritatea teoriilor sistemelor automate se referă la sisteme liniare
- Se aplică principiul superpoziției dacă efectul sumei este egal cu suma efectelor
- Proprietatea de omogenitate presupune conservarea factorului de scară de la intrare la ieșirea sistemului

# Conceptul de cutie neagră

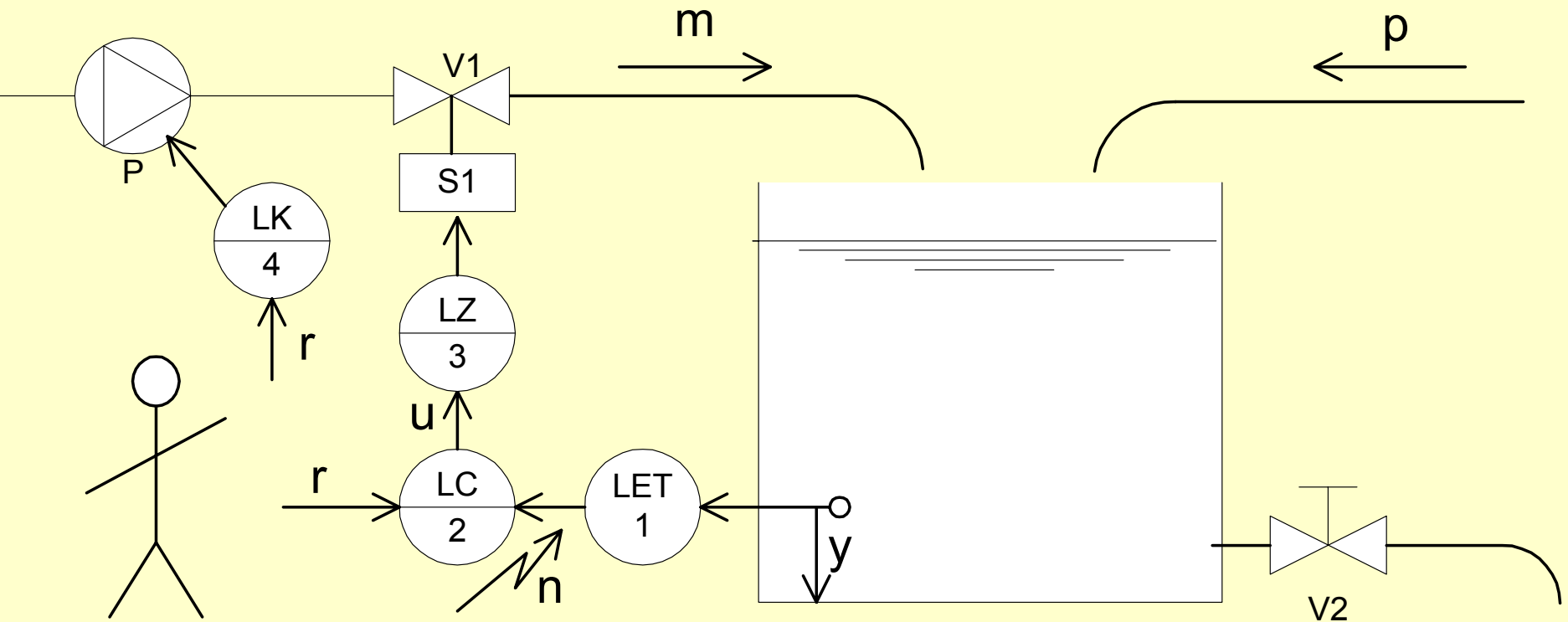


Abstractizare

Ascunderea informației

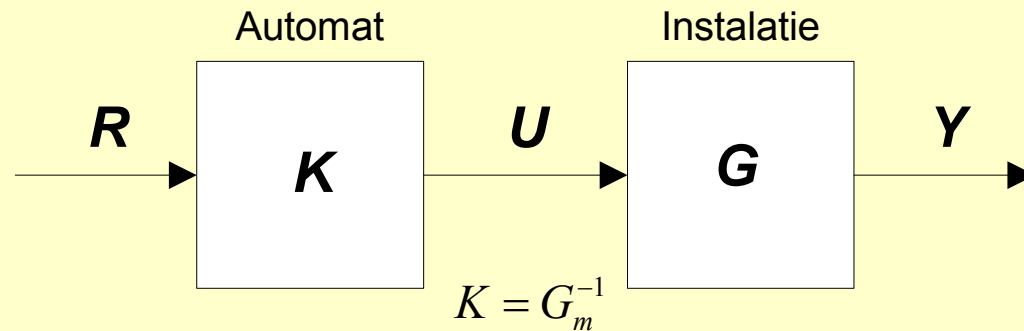
Funcții de transfer

# Schema tehnologică cu aparatura de automatizare pentru conducerea unui rezervor



# Principiul modelului intern

- Automatul  $K$  sau compensatorul  $K$  al regulatorului trebuie să conțină un model  $G_m$  inversat al procesului sau instalației conduse  $G$ .

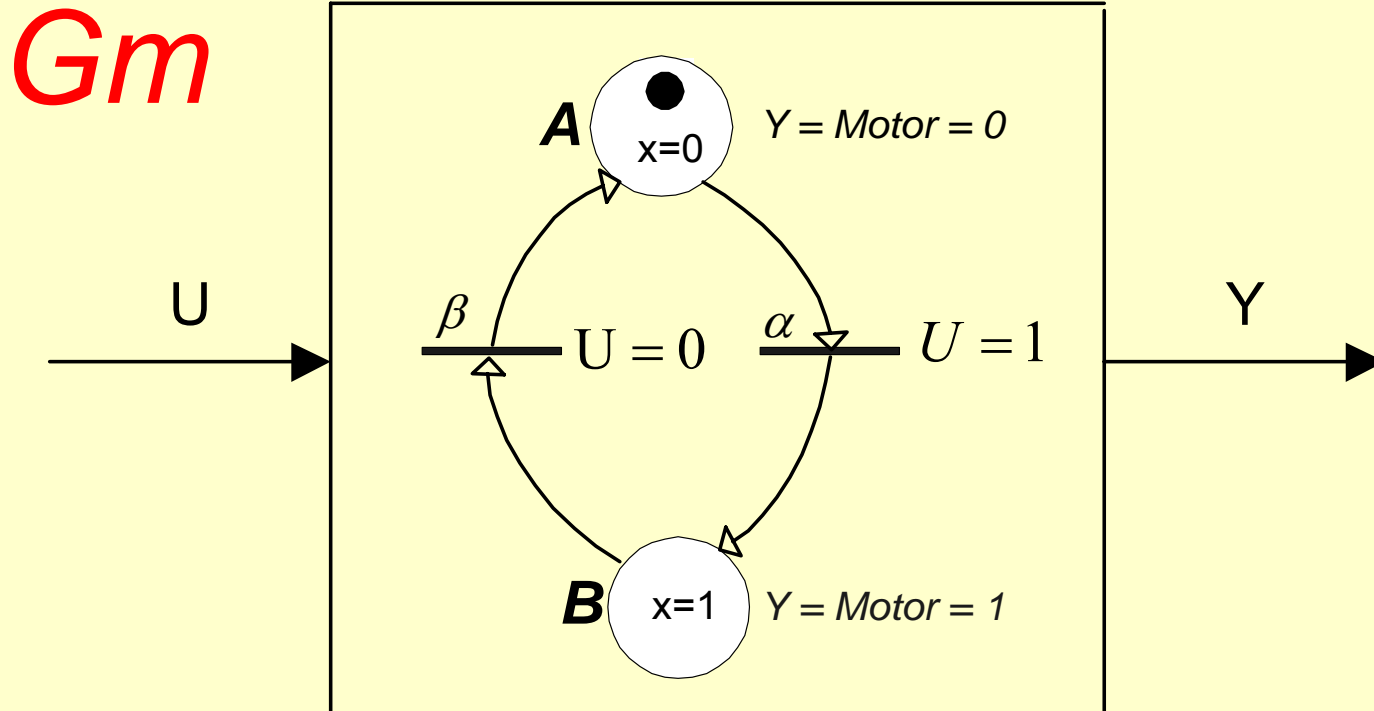


# Trei metode de obținere a lui $G_m$ (modelare) conduc la:

---

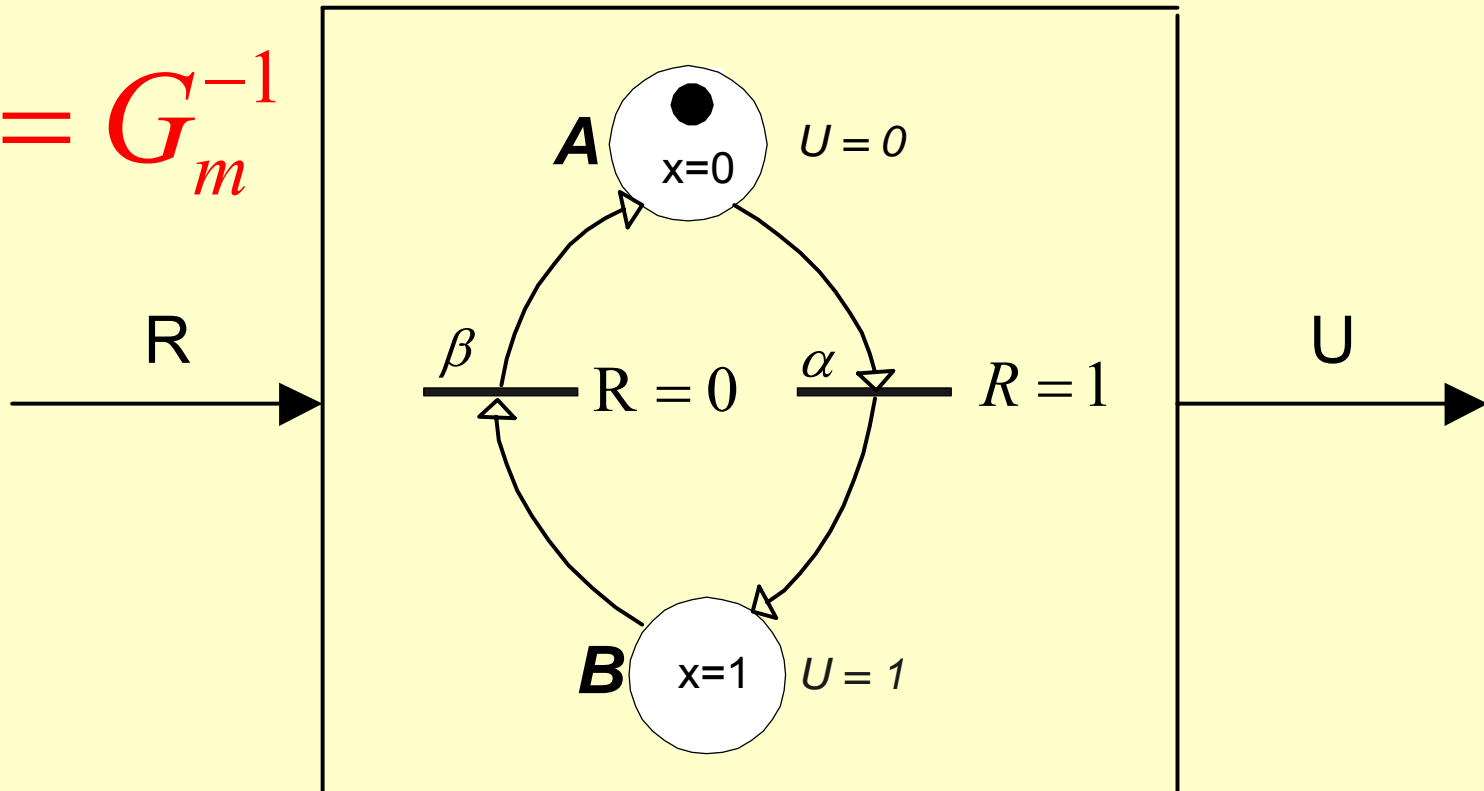
- Sisteme automate discrete logice
- Sisteme automate continue
- Sisteme automate discrete cu eșantionare (numerice)

# Exemplul 1. Modelul unui motor $Gm$ sub forma unei rețele Petri interpretate conforme



# Automatul $K$ pentru pornirea motorului memorează inversul modelului $G_m$ al motorului

$$K = G_m^{-1}$$



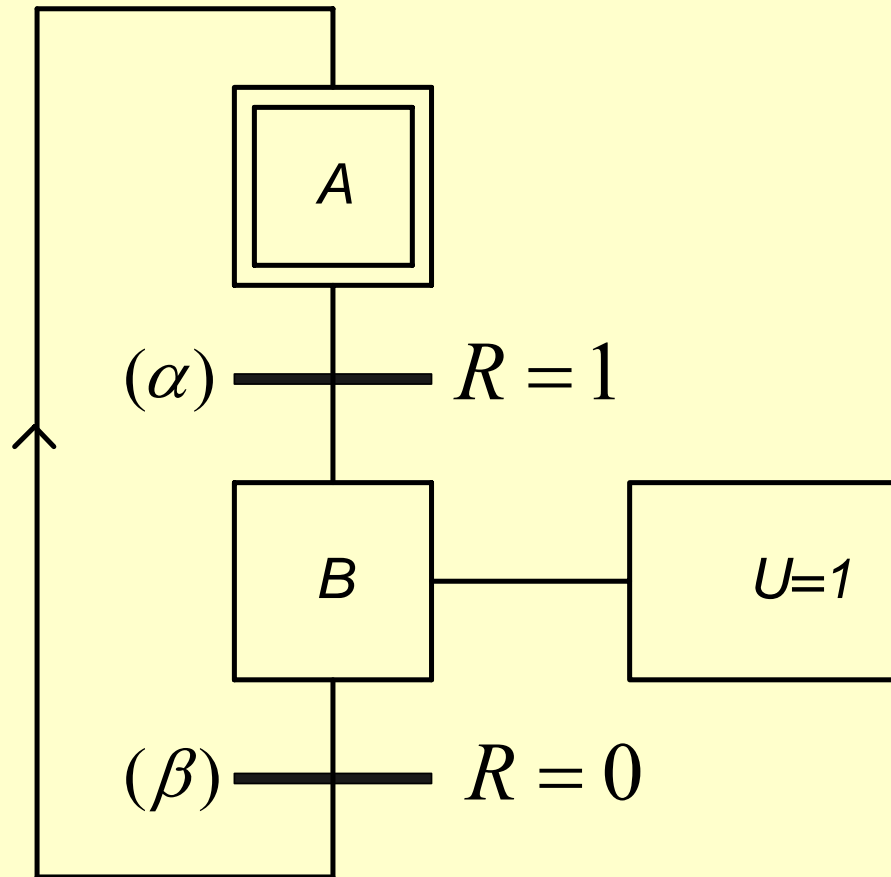
# Performanțele sistemului cu automat modelat cu ajutorul rețelei Petri interpretate

---

- Reversibilitatea - se asigură ciclurile repetitive necesare
- Siguranța - toate pozițiile au maximum o marcă
- Viabilitatea - inexistența blocărilor
- Interblocarea - inexistența conflictelor structurale efective

# Graficul corespunzător rețelei Petri interpretate conforme pentru Automatul Programabil Logic

---

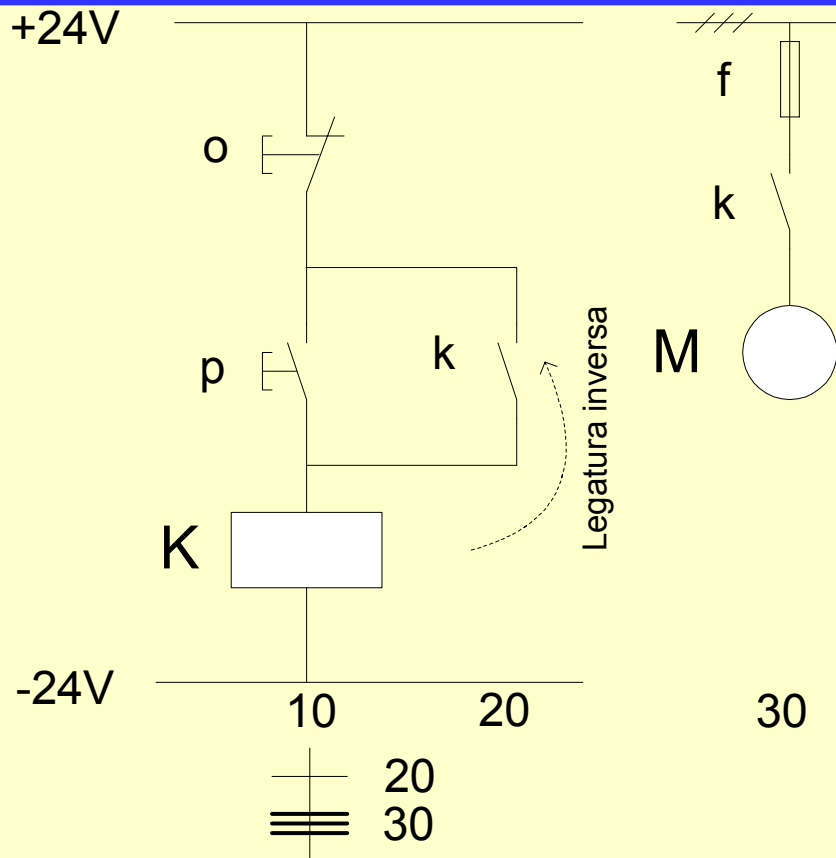


# Sinteza automatului pentru pornirea motorului cu ajutorul diagramei Karnaugh

$$x^{t+\Delta} = \underline{o^t \cdot p^t} + \underline{\cancel{o^t} \cdot \cancel{x^t}} = o^t (p^t + x^t)$$

$x^t \backslash p^t o^t$	00	01	11	10
0	0	0	0	1 /
1	1	0	0	<u>1</u> //

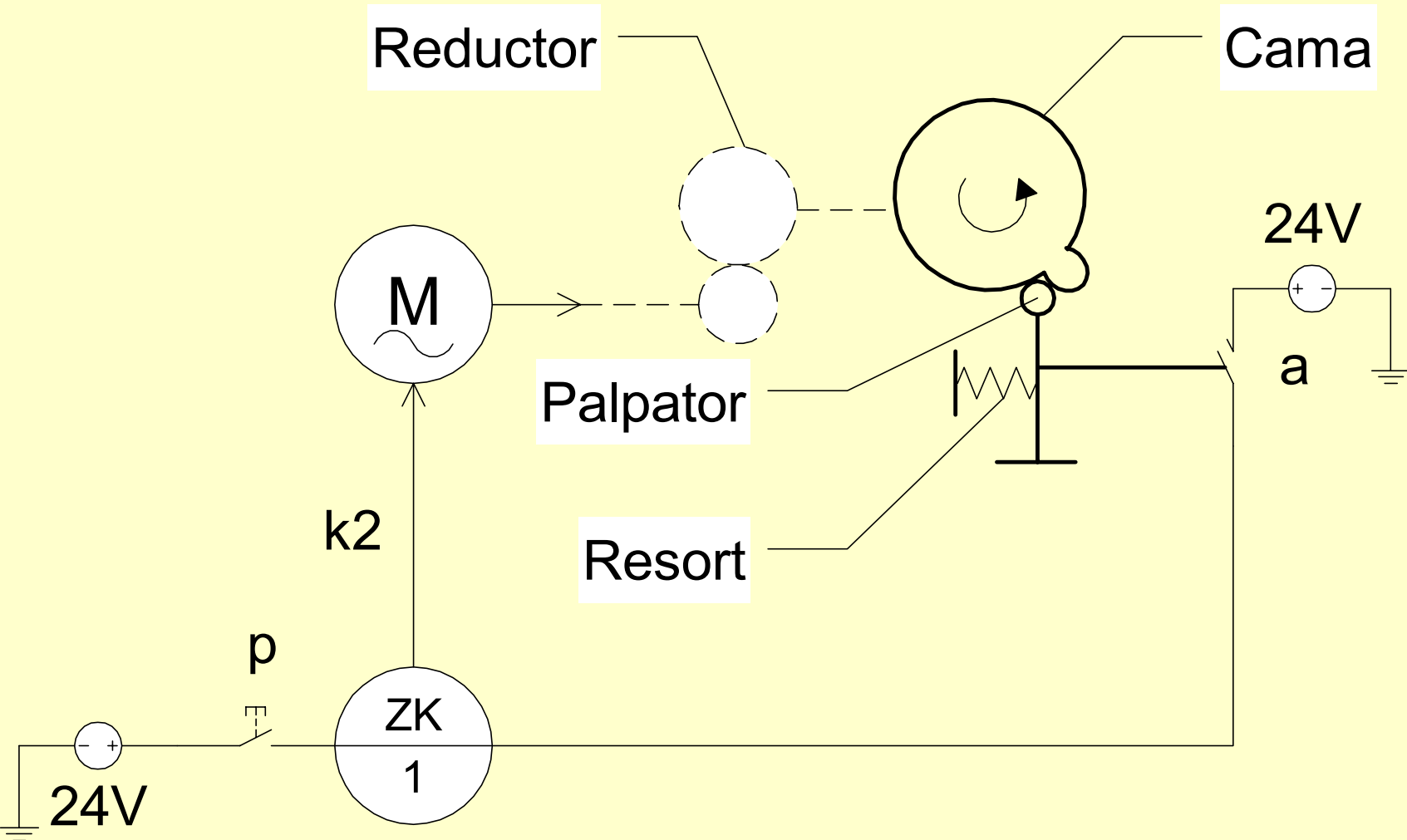
# Schema desfășurată electrică a automatului cu contacte și relee și cu prioritate la oprire



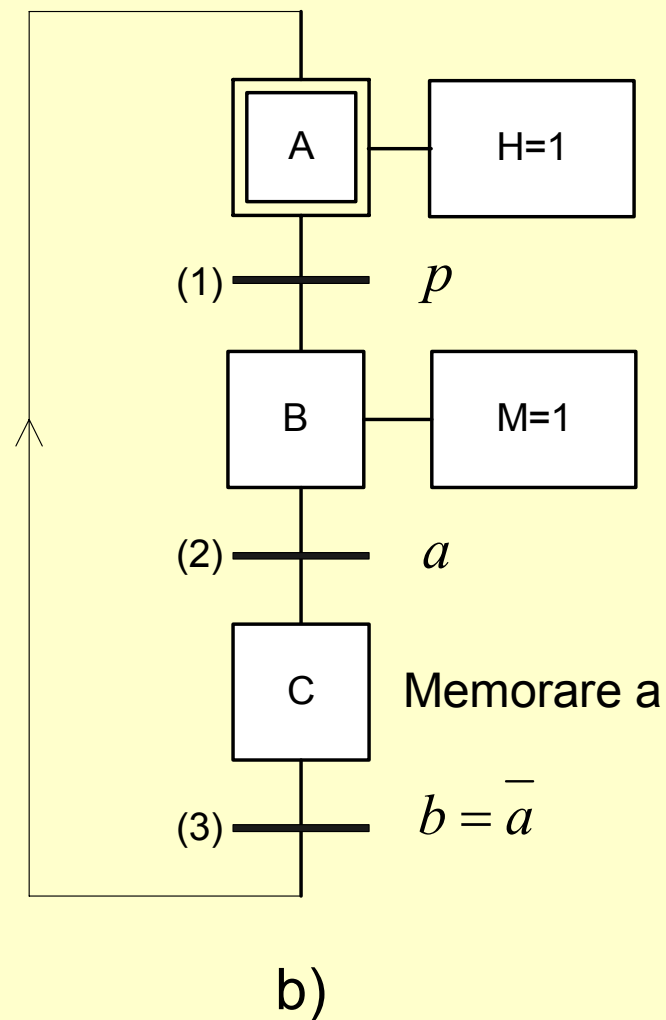
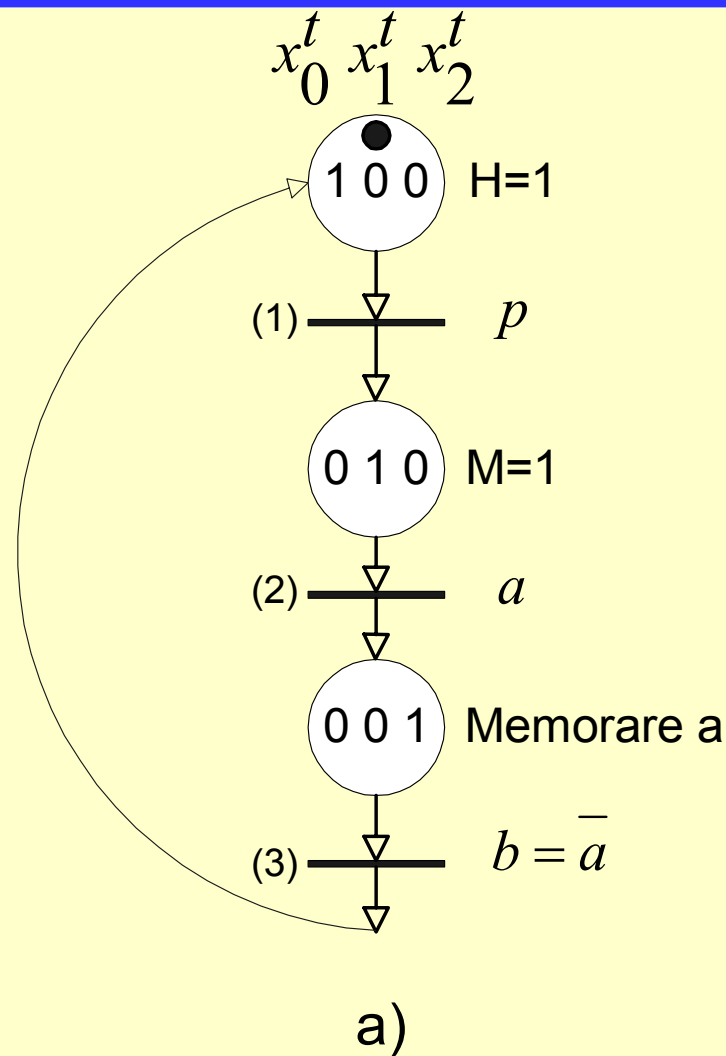
$$k^{t+\Delta} = \overline{o^t} \cdot (p^t + k^t)$$

Automat elementar cu prioritate la oprire	Motor electric
---	----------------

# Exemplul 2. Automat pentru oprirea unui motor după o rotație. Schema tehnologică.



# Rețeaua Petri conformă codificată și graficul pentru programarea APL



Relațiile logice de calcul a biților codului în funcție de condițiile de setare  $s$  și resetare  $r$

---

$$x_n^{t+\Delta} = \left( s_n^t + \bar{r}_n^t \cdot x_n^t \right) \cdot \bar{i}$$

$$x_n^{t+\Delta} = s_n^t + \bar{r}_n^t \cdot x_n^t + i$$

# Implementarea cu contacte și rele

---

- Dacă în locația (poziția) activă inițial nu se realizează acțiuni se renunță la bitul  $x_0$  și se asigură prioritatea la activare. Se rezolvă eventualele conflicte structurale efective.
- În condiția de resetare se introduce o întârziere pentru evitarea nedeterminărilor provocate de setarea și resetarea simultană

# Condițiile de setare $s$ și resetare $r$ pentru implementarea cu APL și cu contacte și rele

---

	<i>Automat Programabil</i>	<i>Contacte si rele</i>
$n=1$	$s_1 = x_0.p$ $r_1 = x_1.a$	$s_1 = x_0.p$ $r_1 = x_1.x_2$
$n=2$	$s_2 = x_1.a$ $r_2 = x_2./a$	$s_2 = x_1.a$ $r_2 = x_2.x_0$
$n=0$	$s_0 = x_2./a$ $r_0 = x_0.p$	$s_0 = x_2./a$ $r_0 = x_0.x_1$

# Relațiile logice ale automatului pentru implementarea cu contacte și relee

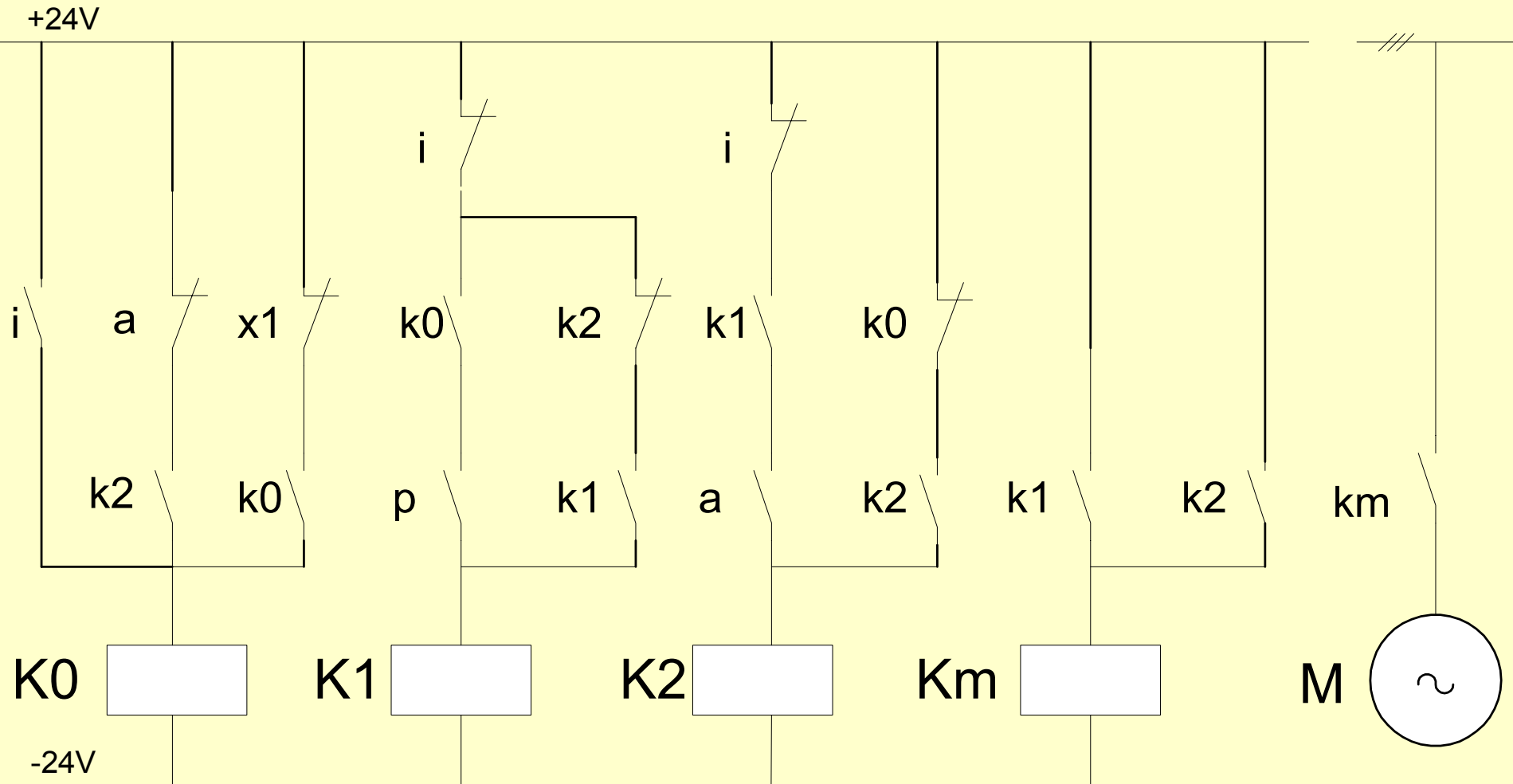
---

$$x_1^{t+\Delta} = (x_0^t \cdot p^t + \bar{x}_2^t \cdot x_1^t) \bar{i}$$

$$x_2^{t+\Delta} = (x_1^t \cdot a^t + \bar{x}_0^t \cdot x_2^t) \bar{i}$$

$$x_0^{t+\Delta} = x_2^t \cdot \bar{a}^t + \bar{x}_1^t \cdot x_0^t + i$$

# Schema desfășurată electrică pentru automatul cu inițializare. Implementare cu contacte și rele



# Programarea Automatului Programabil Logic se poate face cu:

---

- GrafCET (Sequential Function Chart - SFC). Franța, Uniunea Europeană
- Listă de instrucțiuni (Instruction List - IL). Germania
- Schemă desfășurată (Ladder Diagram - LD). SUA
- Schemă bloc funcțională (Function Block Diagram . FBD)
- Program gen Pascal (Structured Text - ST)

# Programul trebuie să conțină:

---

- Configurare intrări, ieșiri, memorii, etc
- Lista instrucțiuni (sau grafic, LD)
  - *Achiziție intrări (Input Scan)*
  - *Logic Scan*
    - Calcul stări
    - Actualizare stări
    - Calcul ieșiri
  - *Furnizare ieșiri (Output scan)*
- Instrucțiunile de bază sunt: LD, ST, AND, OR, negatele lor și S, R

# Configurarea Automatului Programabil Logic

---

- p este conectat la %I0.0 si este memorat in %M0.0
- o este conectat la %I0.1 si este memorat in %M0.1
- contactul k1=y1 exista la %Q1.0
- contactul k2=y2 exista la %Q1.1
- Rezultate intermediare
  - starea actuala xt este memorata in %M0.2
  - starea viitoare xtd este memorata in %M0.3
  - iesirea motor y1 este memorata in %M0.4
  - iesirea lampa y2 este memorata in %M0.5

# Programul Automatului Programabil Logic sub formă de listă de instrucțiuni

---

$$x^{t+\Delta} = \bar{o}^t (p^t + x^t)$$

$$y_1 = x$$

$$y_2 = x$$

LD %I0.0

LD %M0.2

LD %M0.3

LD %M0.2

LD %M0.5

ST %M0.0

OR %M0.0

ST %M0.2

ST %M0.5

ST Q1.1

LD %I 0.1

ANDN %M0.1

LD %M0.2

LD %M0.4

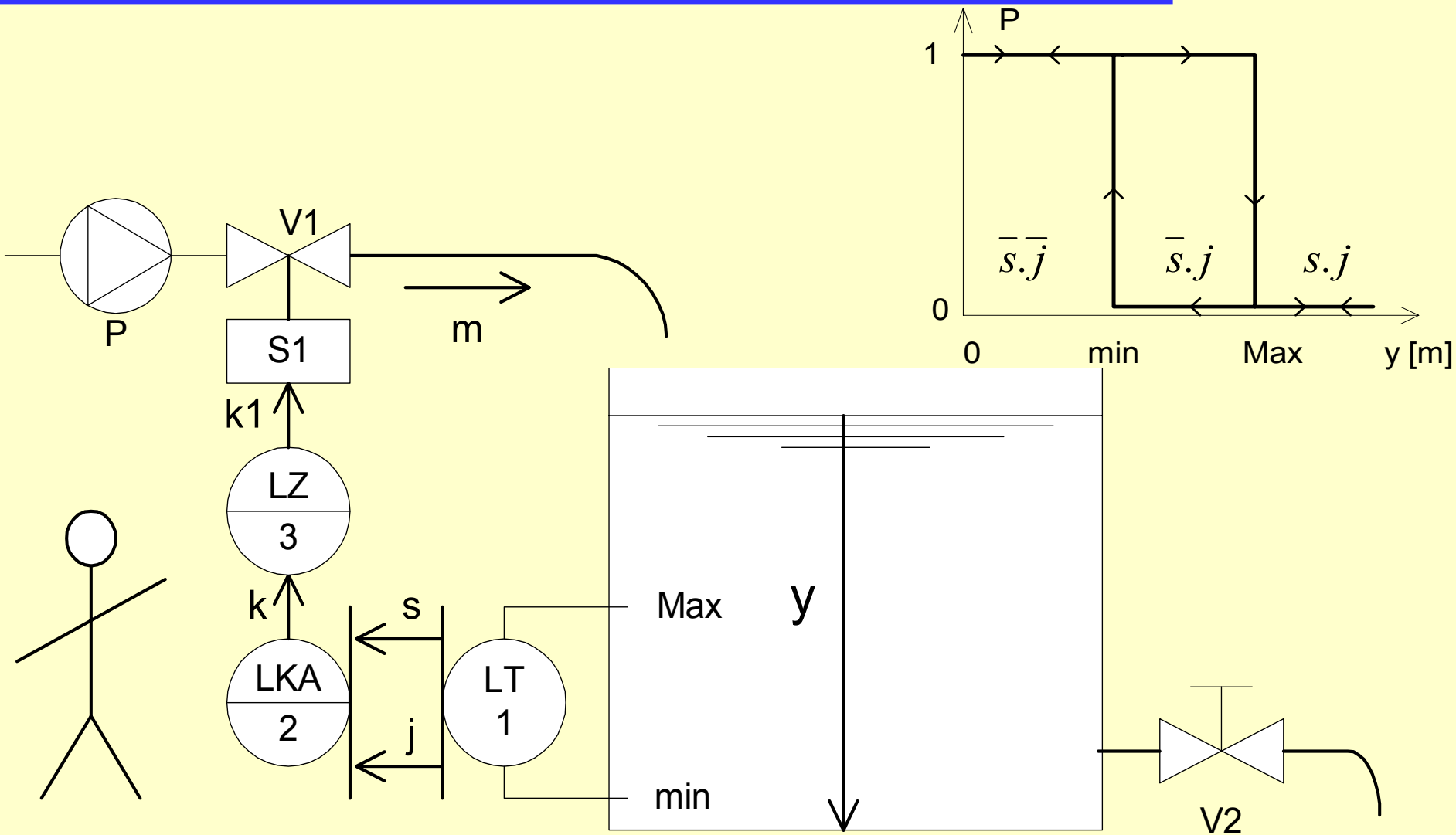
ST %M0.1

ST %M0.3

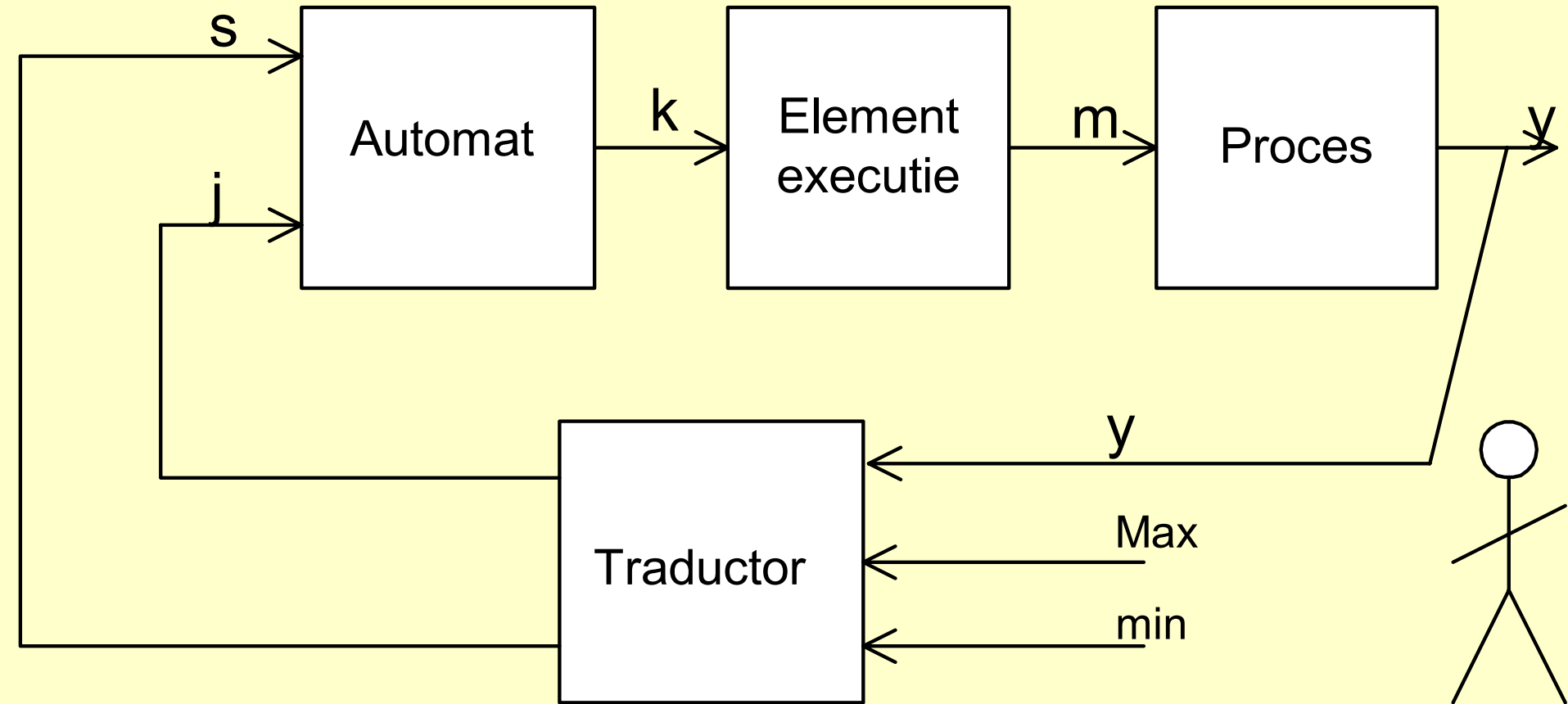
ST %M0.4

ST %Q1.0

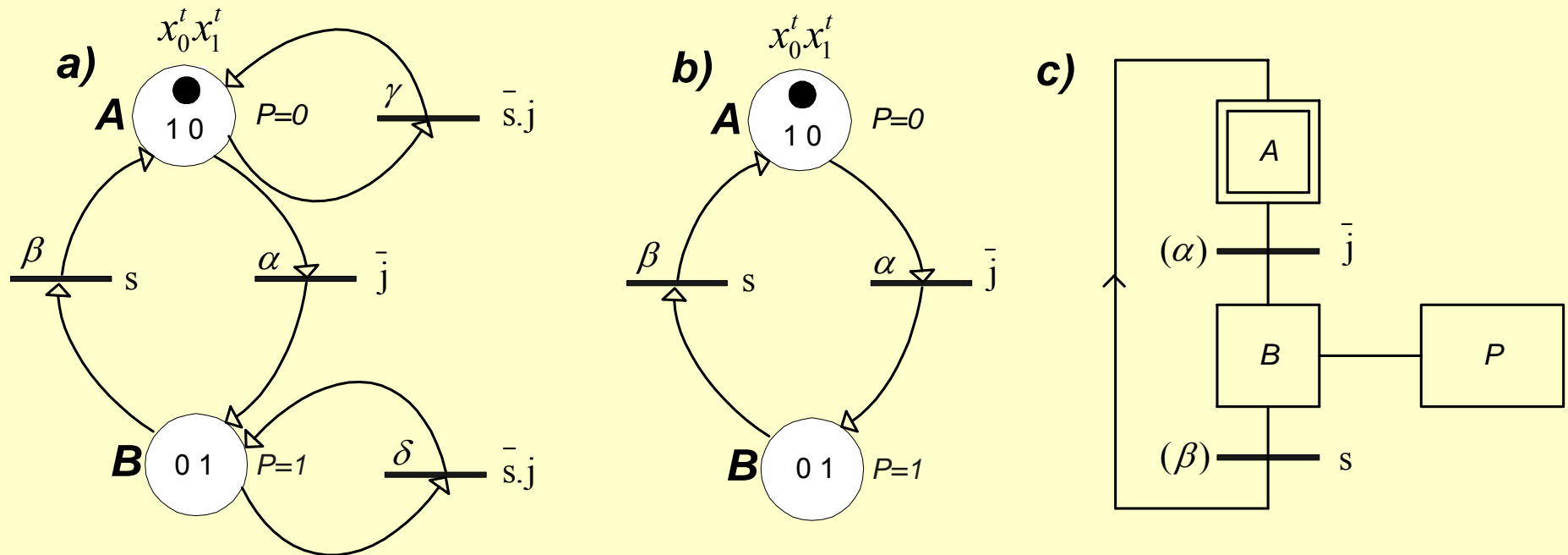
# Exemplul 3. Reglarea 2P a nivelului. Automatul folosit într-o structură cu legătură inversă



# Schema bloc a buclei de reglare automată bipoziționale a nivelului



# Trei variante ale modelului automatului pentru reglarea nivelului



# Calculul relației logice cu ajutorul diagramei Karnaugh

$$x^{t+\Delta} = \overline{j^t} + \overline{s^t} \cdot x^t$$

$x^t \backslash \begin{matrix} s^t & j^t \end{matrix}$	00	01	11	10
0	1	0	0	-
1	1	// 1	0	-

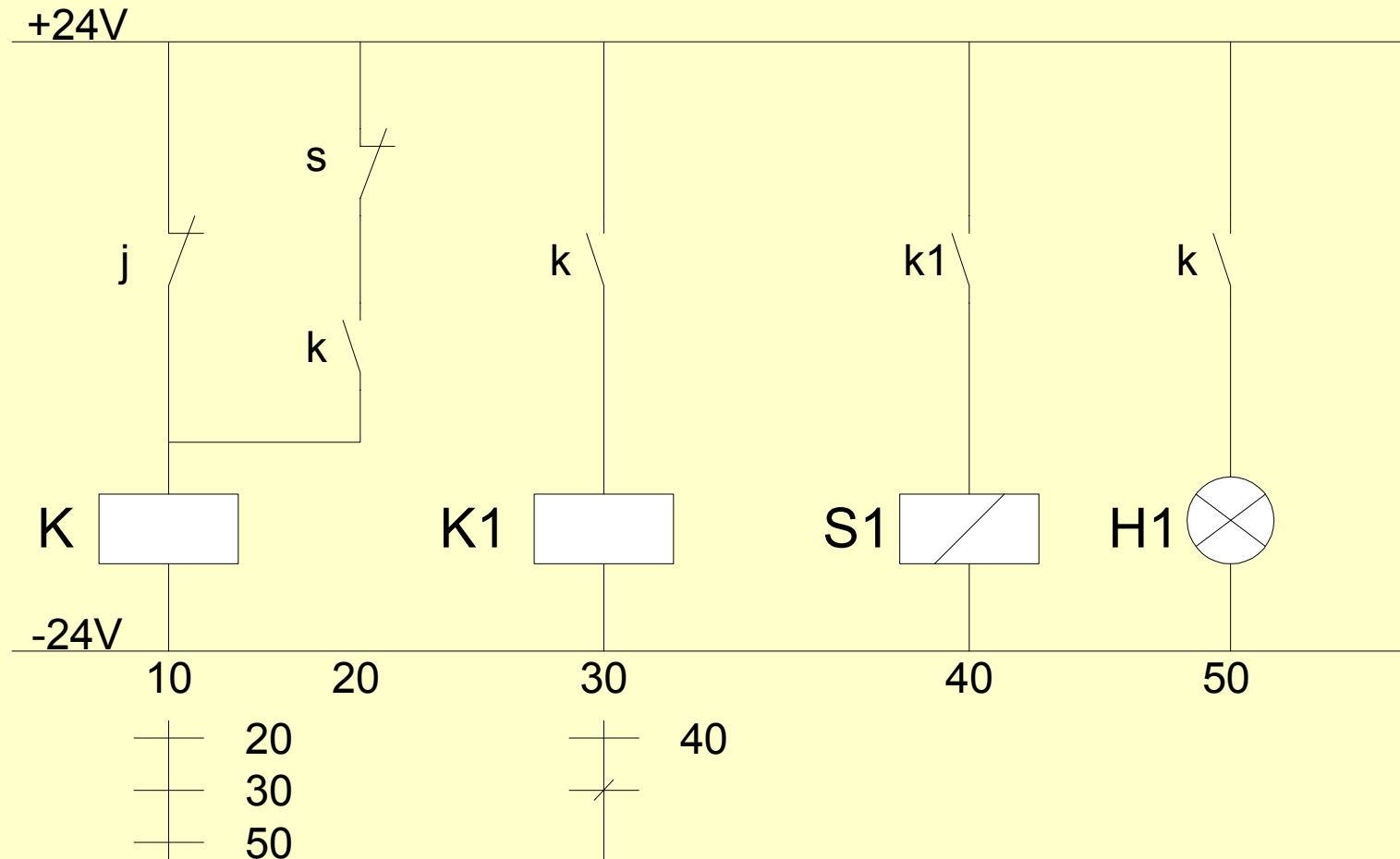
# Performanțele sistemului automat cu regulator bipozițional (2P)

---

$$w = \frac{Max + min}{2}$$

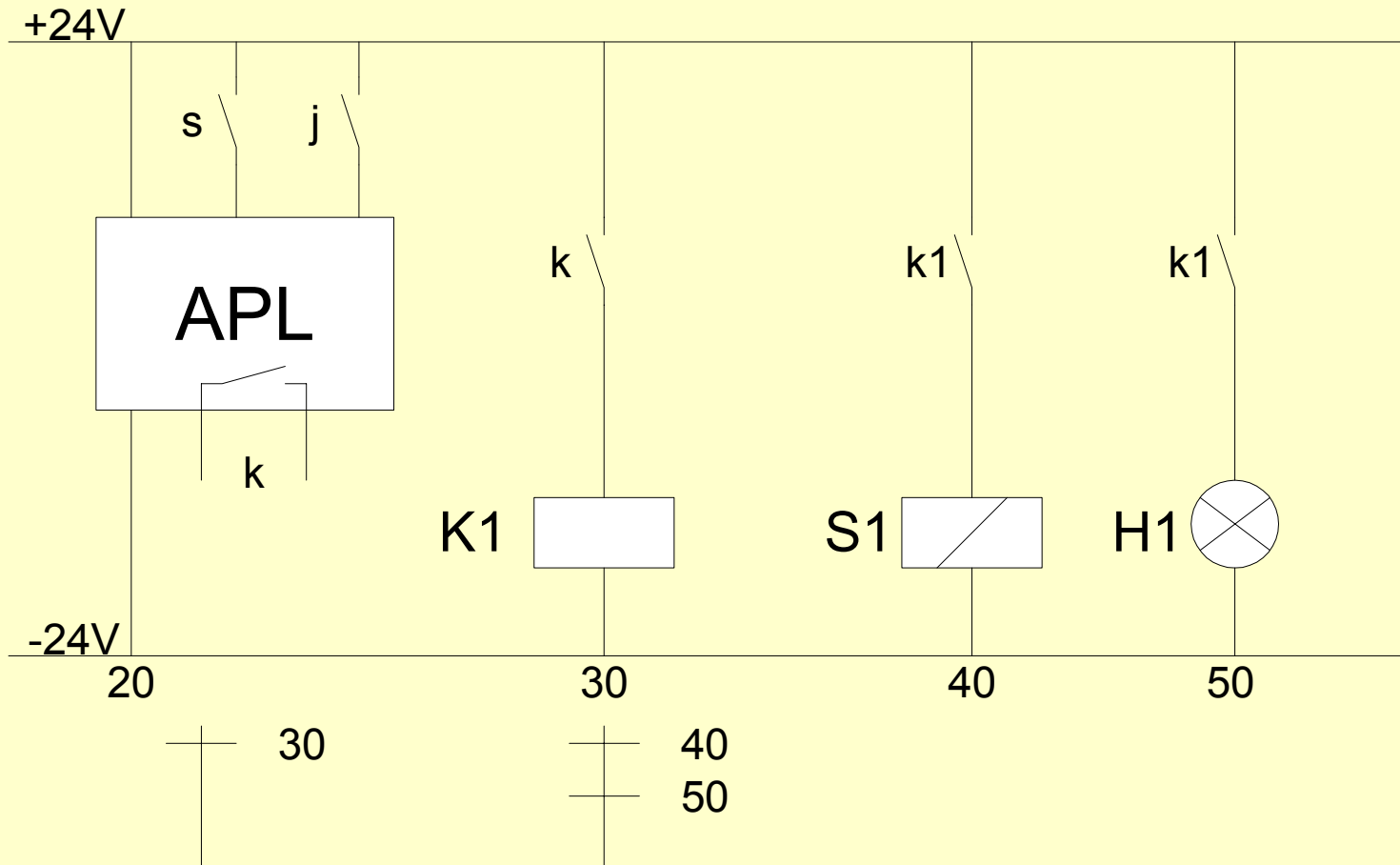
$$e_{max} = \frac{Max - min}{2}$$

# Proiectul constructiv al sistemului automat discret logic - schema desfășurată electrică



Automat	Amplificator	Electromagnet	Semnalizator
---------	--------------	---------------	--------------

# Schema desfășurată electrică cu automat programabil logic



Automat Programabil Logic	Amplificator	Electromagnet	Semnalizator
------------------------------	--------------	---------------	--------------