

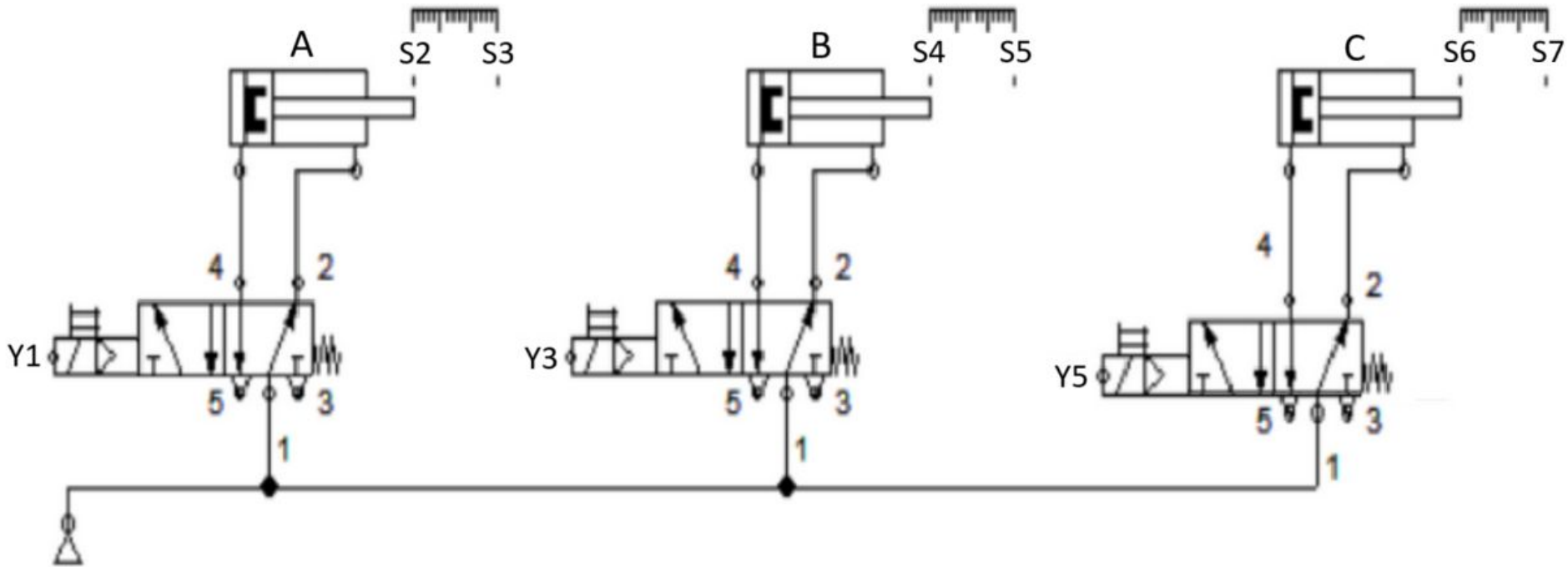
# Controladores Lógicos Programáveis

Técnico em Automação Industrial

# Método de programação por Maximização de contatos

Cadeia estacionária ou passo a passo

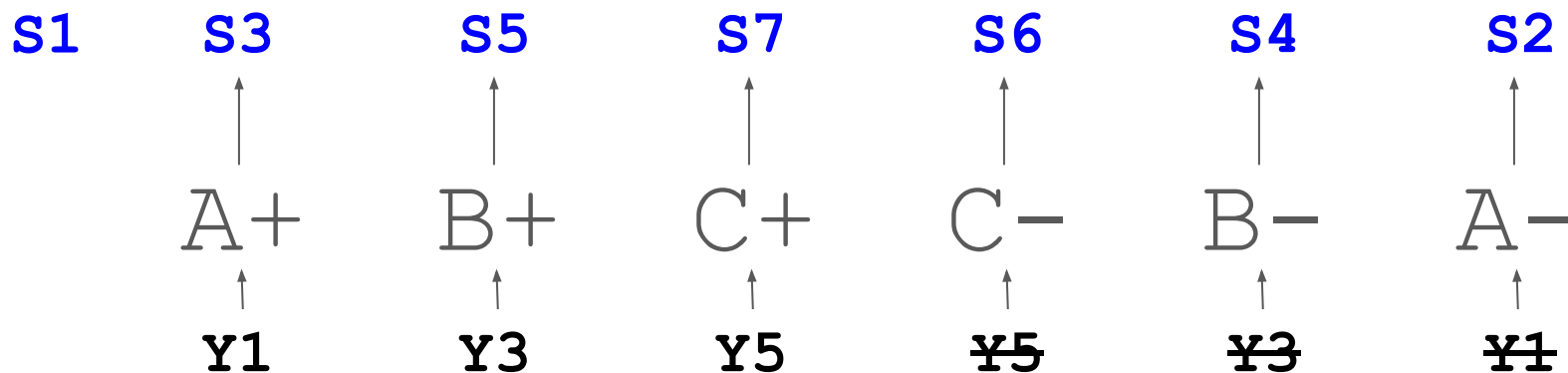
# Circuito



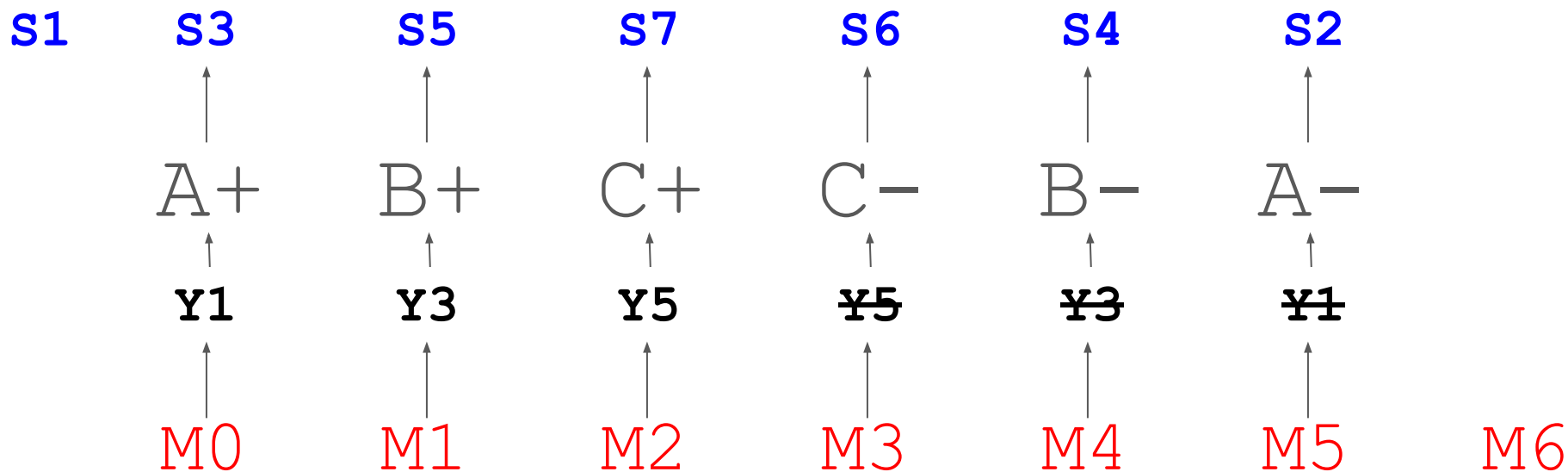
## Passo 0: Sequência de acionamento

A+    B+    C+    C-    B-    A-

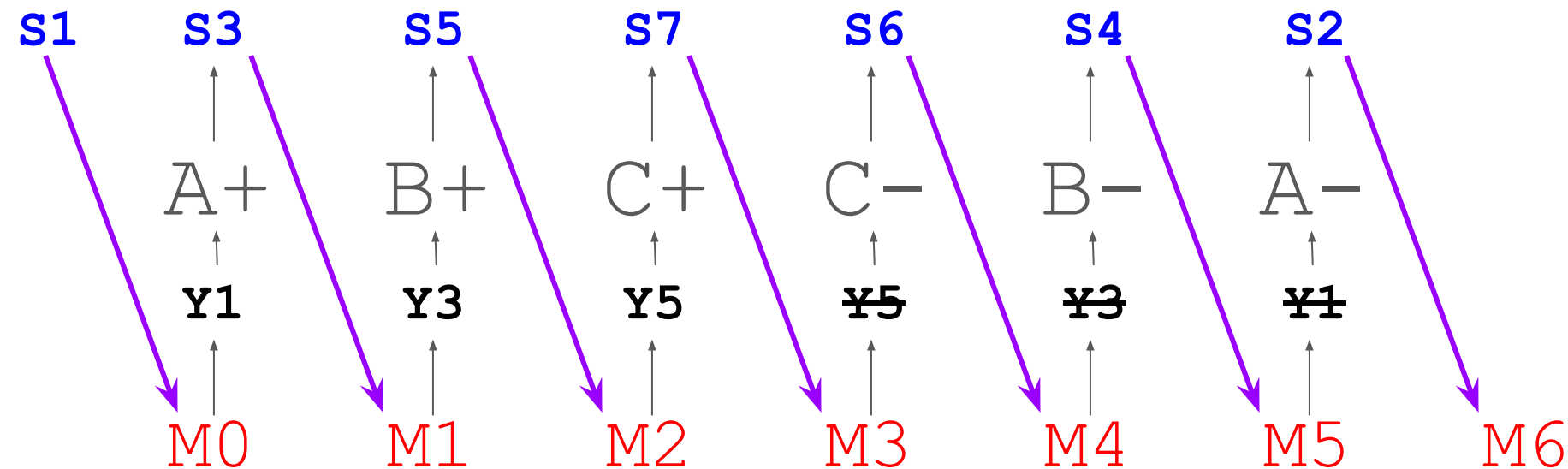
## Passo 1: Atuadores e Fins de Curso



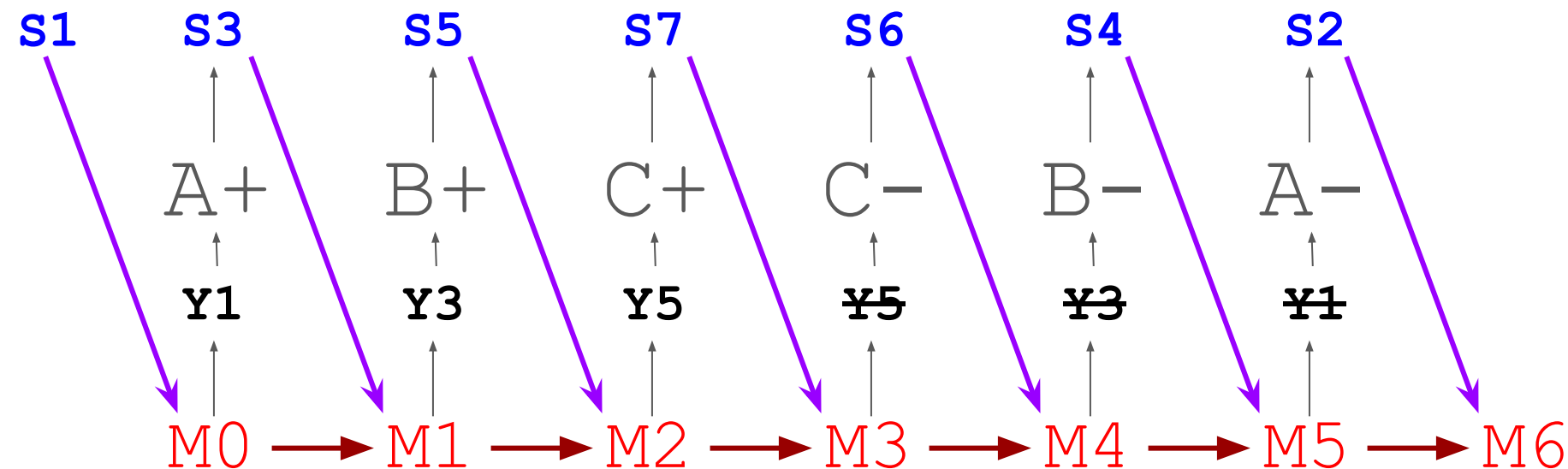
Passo 2: Para cada movimento, uma **memória**, +1



### Passo 3: Acionamento das memórias

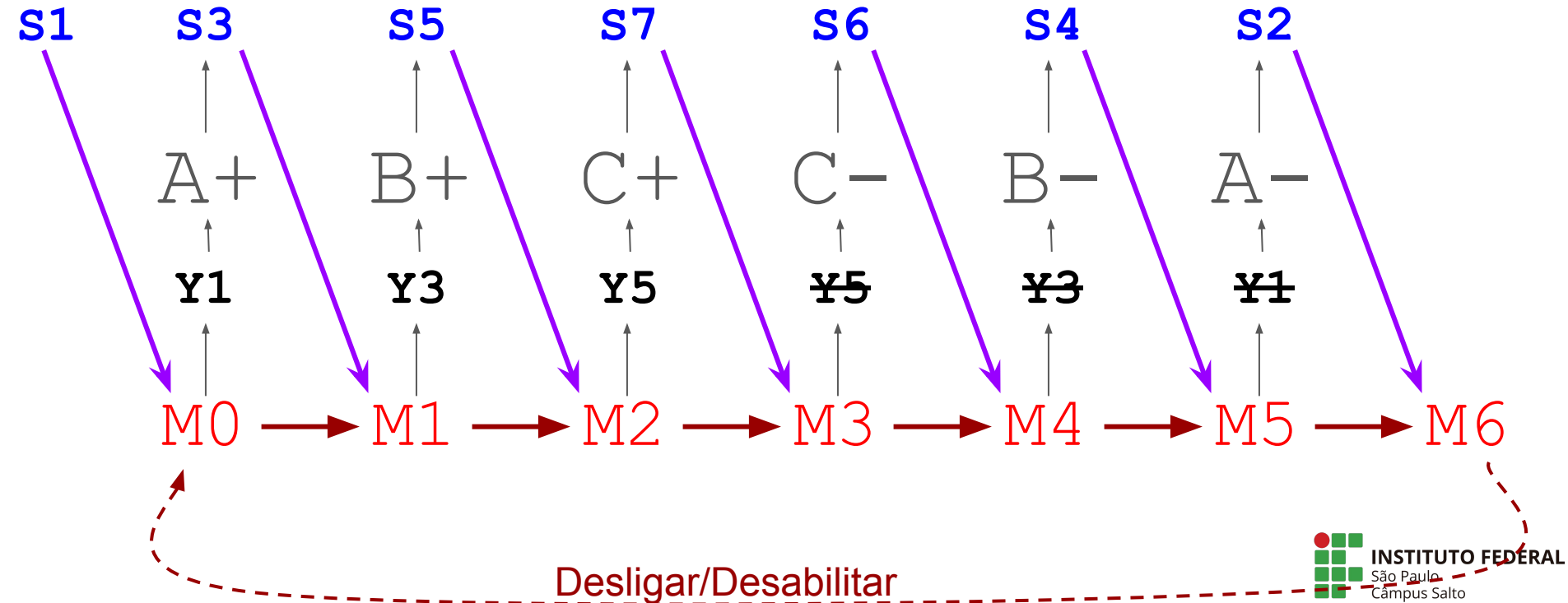


## Passo 4: Habilitação das Memórias





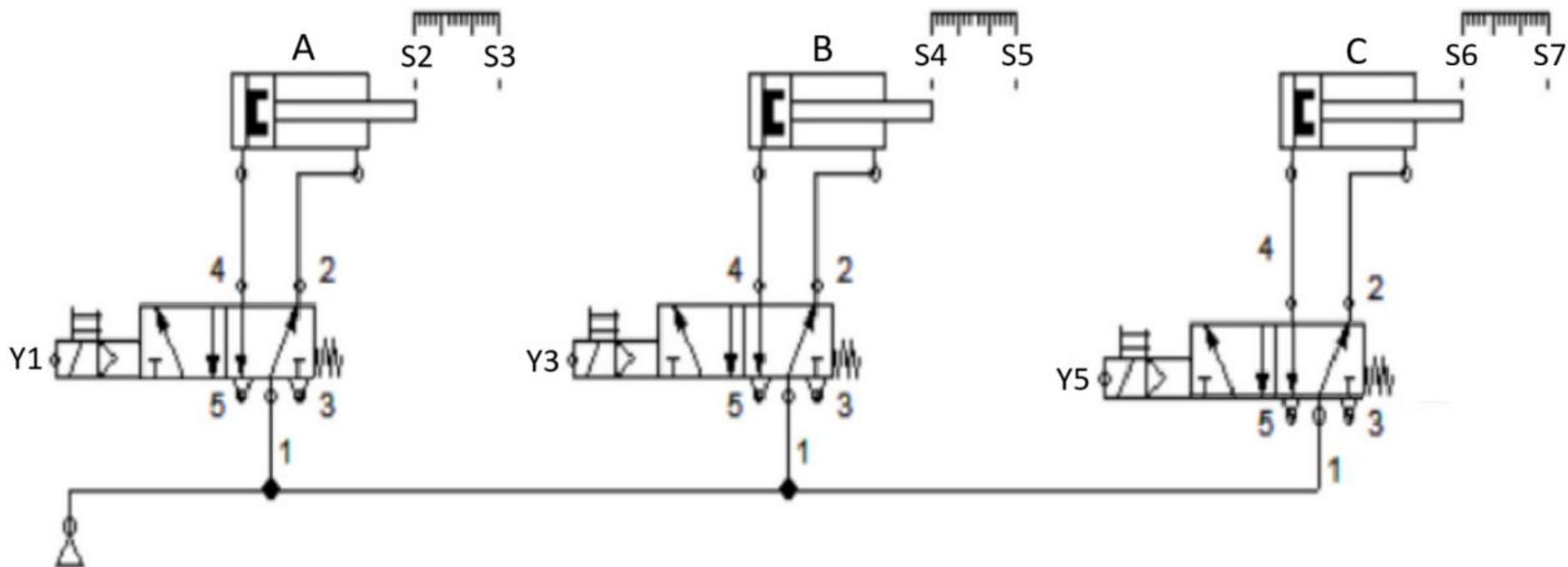
## Passo 5: Desabilitação das Memórias



# Programação no PLC

Linguagem Ladder

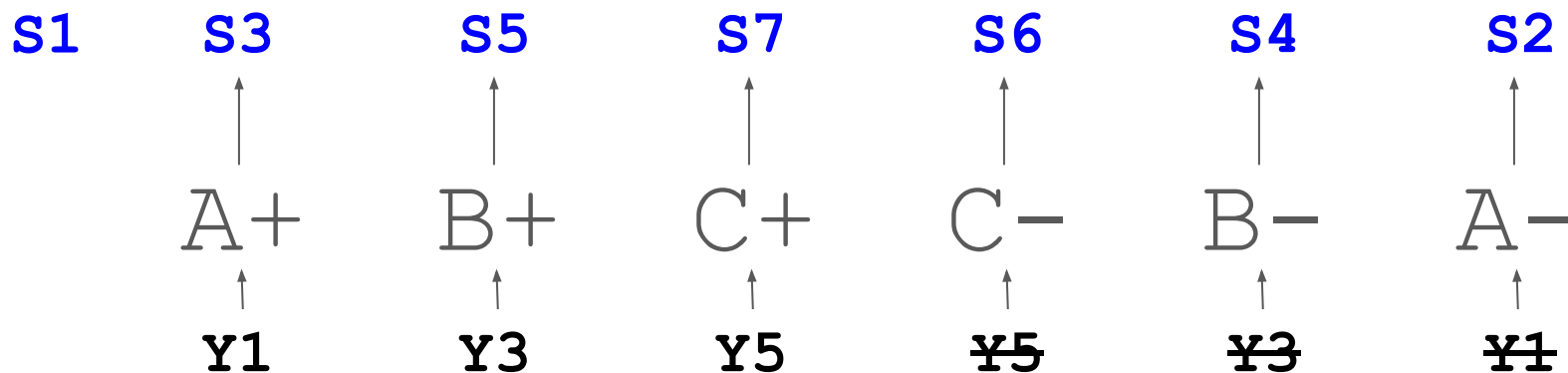
# Circuito



## Passo 0: Sequência de acionamento

A+    B+    C+    C-    B-    A-

## Passo 1: Atuadores e Fins de Curso



# Declaração de Entradas(Fins de curso) e Saídas(Atuadores) de acordo com o diagrama

POUs

- NAVEGA (PRG)
- PLC\_PRG (PRG)
- seq\_ABCcba (FB)

PLC\_PRG (PRG-LD)

```
0001 PROGRAM PLC_PRG
0002 VAR
0003     Exemplo: seq_ABCcba;
0004     (* Entradas *)
0005     S1 AT %IX0.1: BOOL;
0006     S2 AT %IX0.2: BOOL;
0007     S3 AT %IX0.3: BOOL;
0008     S4 AT %IX0.4: BOOL;
0009     S5 AT %IX0.5: BOOL;
0010     S6 AT %IX0.6: BOOL;
0011     S7 AT %IX0.7: BOOL;
0012     (* Saida *)
0013     Y1 AT %QX1.2: BOOL;
0014     Y3 AT %QX1.4: BOOL;
0015     Y5 AT %QX1.6: BOOL;
0016 END_VAR
```

seq\_ABCcba (FB-LD)

```
0001 FUNCTION_BLOCK seq_ABCcba
0002 VAR_INPUT
0003     LIGAR: BOOL;
0004     XA0: BOOL;
0005     XA1: BOOL;
0006     XB0: BOOL;
0007     XB1: BOOL;
0008     XC0: BOOL;
0009     XC1: BOOL;
0010 END_VAR
0011 VAR_OUTPUT
0012     Em_execucao: BOOL;
0013     YA1: BOOL;
0014     YB1: BOOL;
0015     YC1: BOOL;
0016 END_VAR
0017 VAR
0018     M0: BOOL;
0019     M1: BOOL;
0020     M2: BOOL;
0021     M3: BOOL;
0022     M4: BOOL;
0023     M5: BOOL;
0024     M6: BOOL;
0025 END_VAR
```

Diagrama Ladder:

0001

S1

Exemplo

seq\_ABCcba

LIGAR

Em\_execucao

YA1-Y1

YB1-Y3

YC1-Y5

S2-XA0

S3-XA1

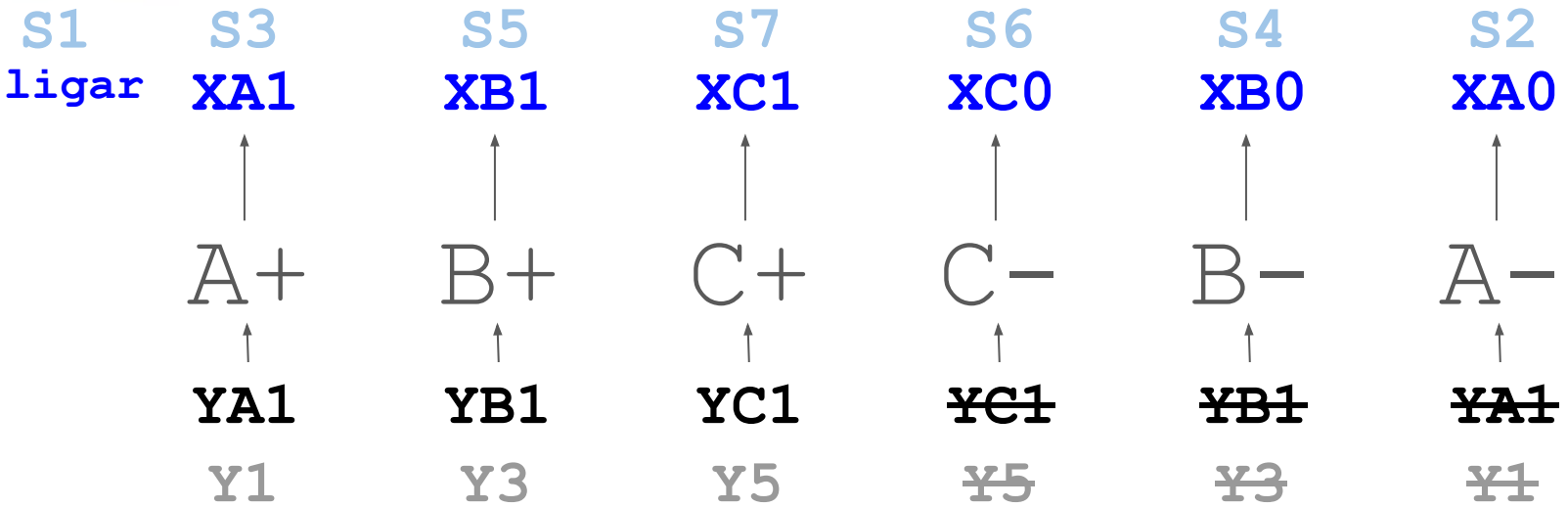
S4-XB0

S5-XB1

S6-XC0

S7-XC1

## Passo 2: Atuadores e Fins de Curso no Bloco Funcional



# Declaração de interface do Bloco Funcional da Sequência

POUs

- NAVEGA (PRG)
- PLC\_PRG (PRG)
- seq\_ABCcba (FB)

PLC\_PRG (PRG-LD)

```
0001 PROGRAM PLC_PRG
0002 VAR
0003     Exemplo: seq_ABCcba;
0004 (* Entradas *)
0005 S1 AT %IX0.1: BOOL;
0006 S2 AT %IX0.2: BOOL;
0007 S3 AT %IX0.3: BOOL;
0008 S4 AT %IX0.4: BOOL;
0009 S5 AT %IX0.5: BOOL;
0010 S6 AT %IX0.6: BOOL;
0011 S7 AT %IX0.7: BOOL;
0012 (* Sidas *)
0013 Y1 AT %QX1.2: BOOL;
0014 Y3 AT %QX1.4: BOOL;
0015 Y5 AT %QX1.6: BOOL;
0016 END_VAR
```

seq\_ABCcba (FB-LD)

```
0001 FUNCTION_BLOCK seq_ABCcba
0002 VAR_INPUT
0003     LIGAR: BOOL;
0004     XA0: BOOL;
0005     XA1: BOOL;
0006     XB0: BOOL;
0007     XB1: BOOL;
0008     XC0: BOOL;
0009     XC1: BOOL;
0010 END_VAR
0011 VAR_OUTPUT
0012     Em_execucao: BOOL;
0013     YA1: BOOL;
0014     YB1: BOOL;
0015     YC1: BOOL;
0016 END_VAR
0017 VAR
0018     M0: BOOL;
0019     M1: BOOL;
0020     M2: BOOL;
0021     M3: BOOL;
0022     M4: BOOL;
0023     M5: BOOL;
0024     M6: BOOL;
0025 END_VAR
```

Diagrama de Ladder (Ligação de Interface):

0001

S1

LIGAR

seq\_ABCcba

Em\_execucao

YA1

YB1

YC1

Y1

Y3

Y5

S2-XA0

S3-XA1

S4-XB0

S5-XB1

S6-XC0

S7-XC1



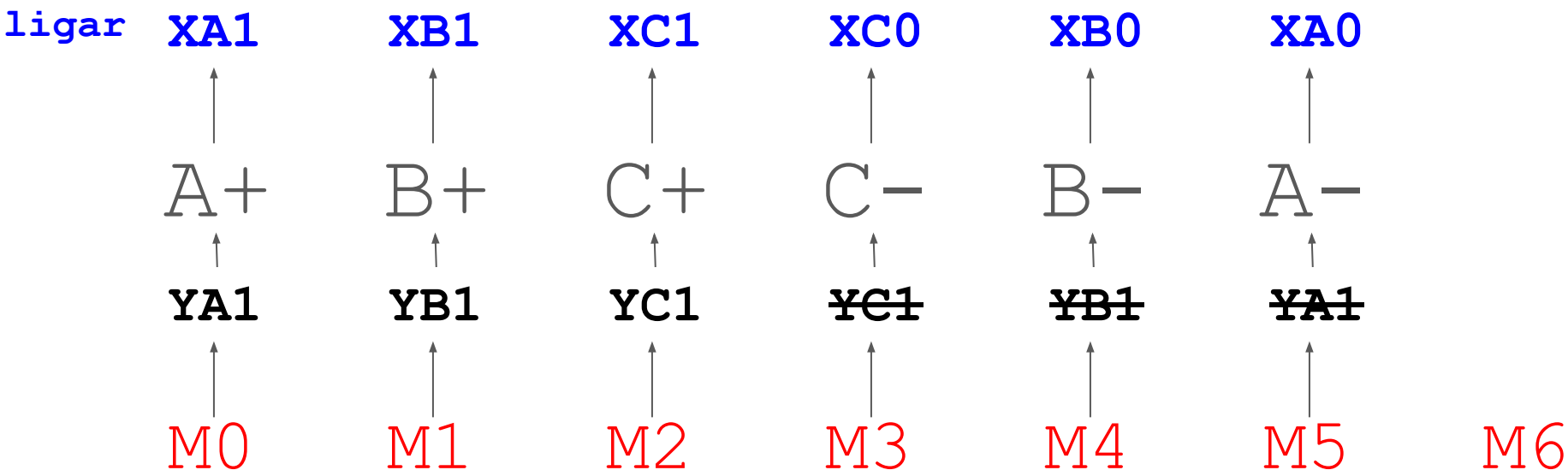
Inserir → Bloco Funcional... → FB def. pelo usuário

The screenshot displays the Siemens STEP 7 interface with three main windows:

- POUs:** A project tree on the left showing the hierarchy: `NAVEGA (PRG)` → `PLC_PRG (PRG)` → `seq_ABCcba (FB)`.
- PLC\_PRG (PRG-LD):** The main editor window showing a ladder logic program. At line 0003, a call to the function block is made: `Exemplo: seq_ABCcba;`. This line is highlighted with a purple box. A purple double-headed arrow points from this text to the function block symbol in the ladder logic diagram below.
- seq\_ABCcba (FB-LD):** A secondary window showing the internal structure of the user-defined function block. It contains two variable declarations:
  - VAR\_INPUT:** `LIGAR: BOOL;` (line 0003)
  - VAR\_OUTPUT:** `Em_execucao: BOOL;` (line 0012)Below these, a list of I/O addresses is provided:
  - Inputs:** `XA0: BOOL;` (0004), `XA1: BOOL;` (0005), `XB0: BOOL;` (0006), `XB1: BOOL;` (0007), `XC0: BOOL;` (0008), `XC1: BOOL;` (0009)
  - Outputs:** `YA1: BOOL;` (0013), `YB1: BOOL;` (0014), `YC1: BOOL;` (0015)The block is terminated with `END_VAR` at line 0016.

The ladder logic diagram in the `PLC_PRG (PRG-LD)` window shows a network starting with a normally open contact `S1` connected to a function block symbol. The symbol is labeled `Exemplo` and `seq_ABCcba`. It has an input `LIGAR` and an output `Em_execucao`. The input `LIGAR` is connected to the `S1` contact. The output `Em_execucao` is connected to a coil (represented by a rectangle). The symbol also lists the internal I/O addresses: `S2-XA0`, `S3-XA1`, `S4-XB0`, `S5-XB1`, `S6-XC0`, `S7-XC1` on the left, and `YA1-Y1`, `YB1-Y3`, `YC1-Y5` on the right.

Passo 3: Para cada **movimento**, uma **memória**, +1

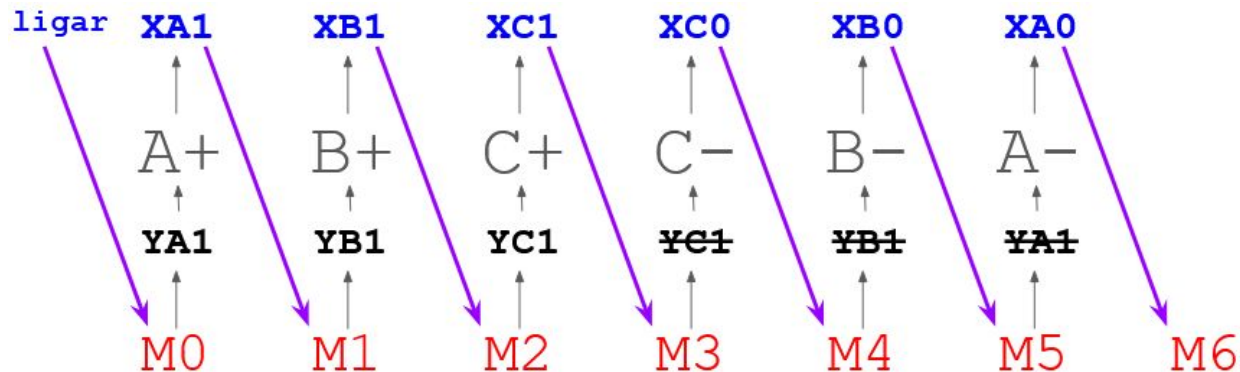
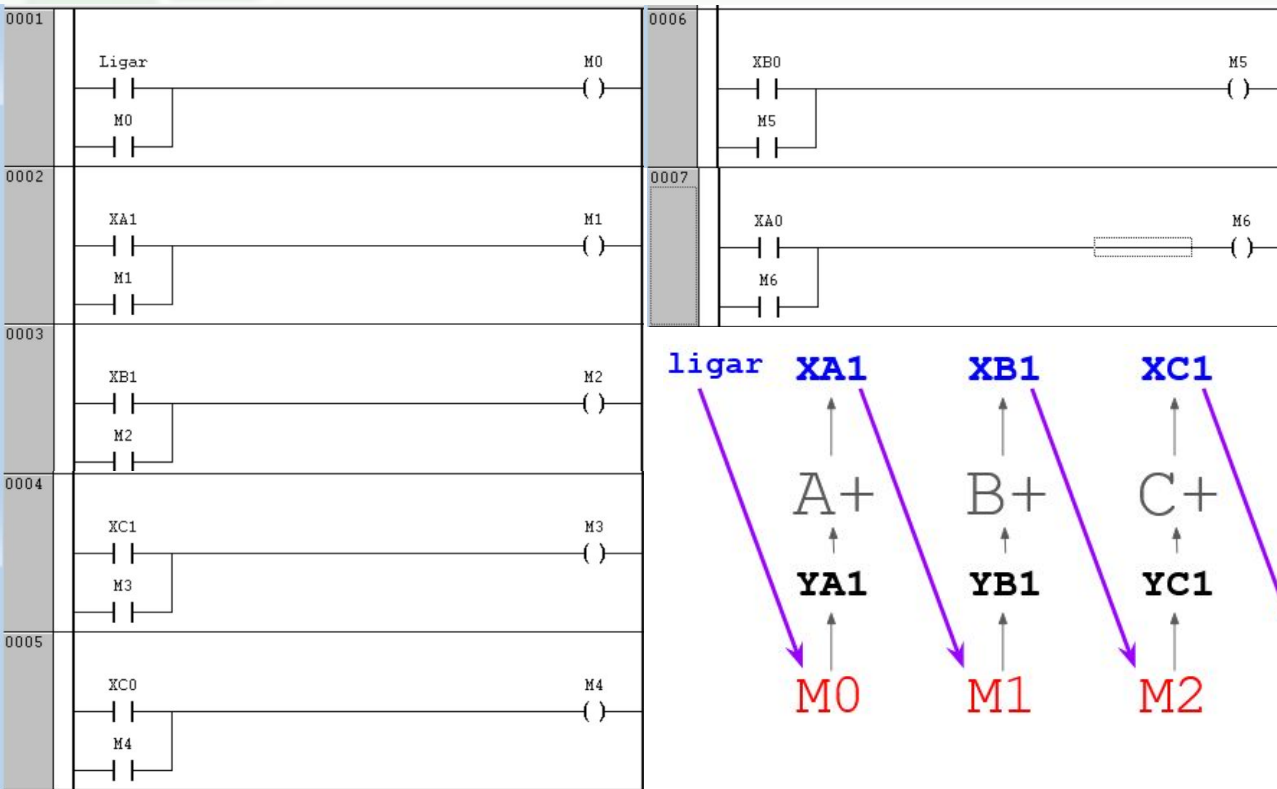


Passo 3: Para cada movimento, uma memória, +1

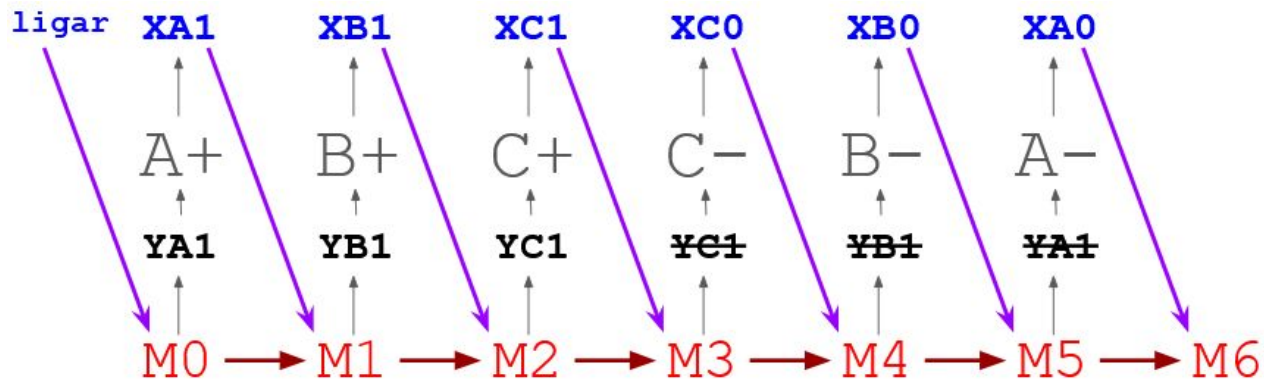
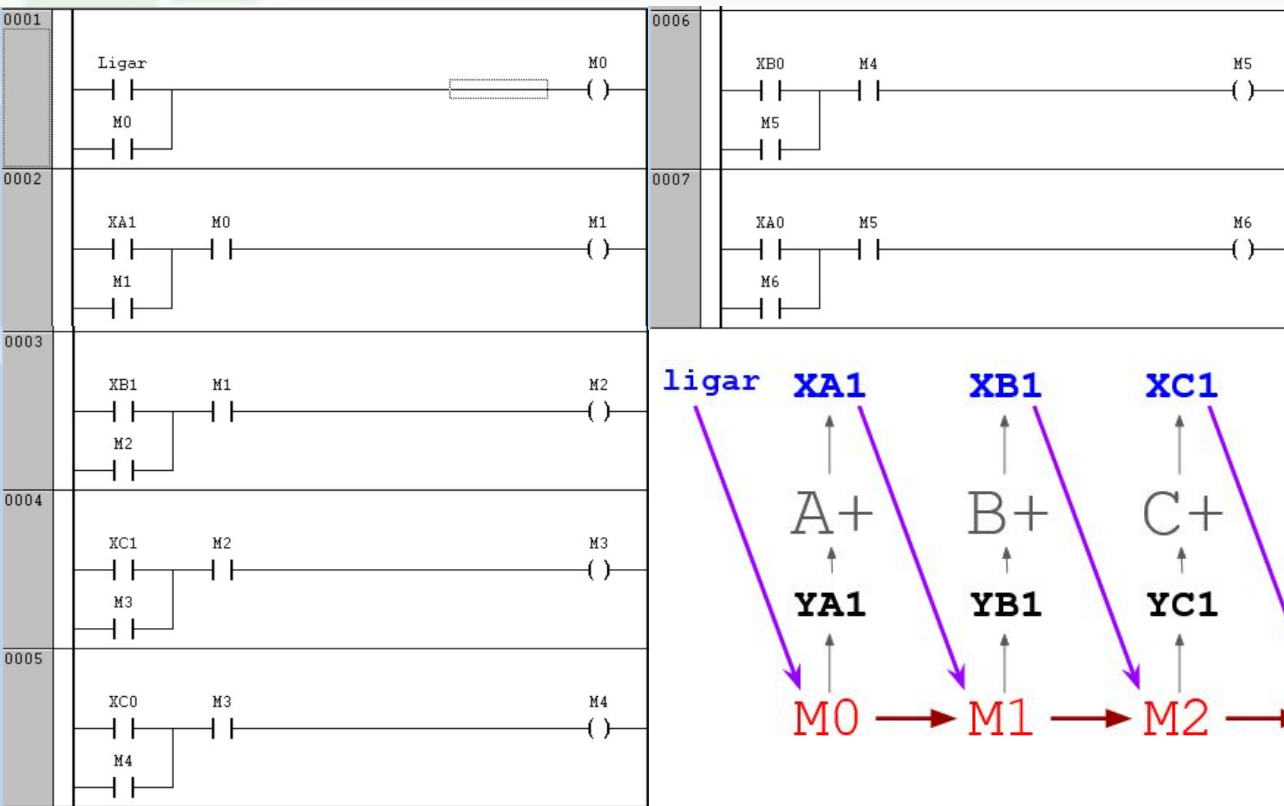
The screenshot displays the Siemens SIMATIC Manager interface with three windows open:

- POUs:** A project tree on the left showing the hierarchy: `NAVEGA (PRG)`, `PLC_PRG (PRG)`, and `seq_ABCcba (FB)`.
- PLC\_PRG (PRG-LD):** The main editor window showing a ladder logic program. It starts with a program declaration and variable declarations for inputs (S1-S7) and outputs (Y1-Y5). A function block `seq_ABCcba` is called with `LIGAR` as the enable input and `Em_execucao` as the output. The output is connected to a coil (represented by a dashed rectangle) in the first step (0001).
- seq\_ABCcba (FB-LD):** The function block editor showing the internal logic of the `seq_ABCcba` block. It defines input variables (`LIGAR`, `XA0`-`XA1`, `XB0`-`XB1`, `XC0`, `XC1`) and output variables (`Em_execucao`, `YA1`, `YB1`, `YC1`). A red bracket on the right side of this window groups the output variables from line 0018 to 0024, with the text "Número de movimentos + 1" next to it, indicating that the number of outputs is one more than the number of movements.

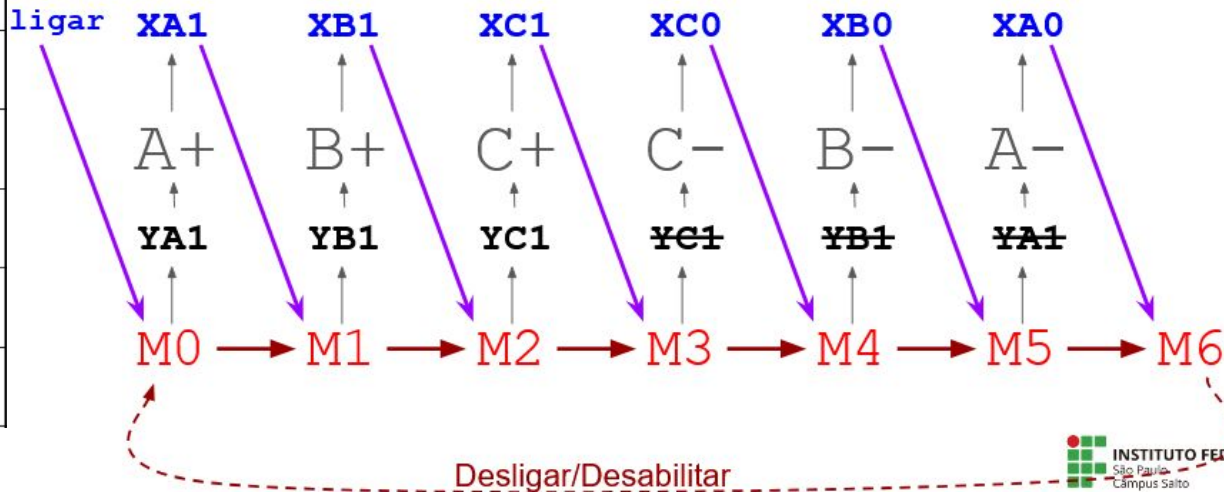
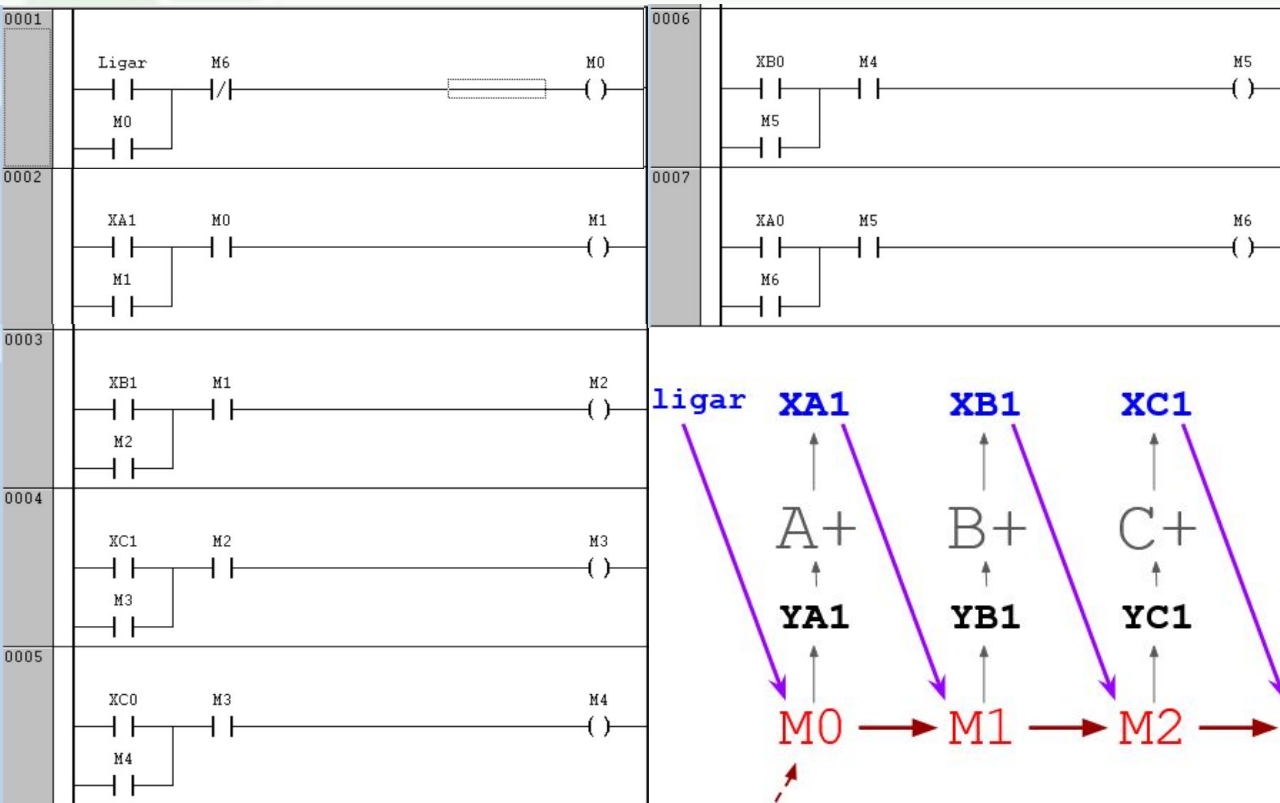
## Passo 4: Acionamento das memórias



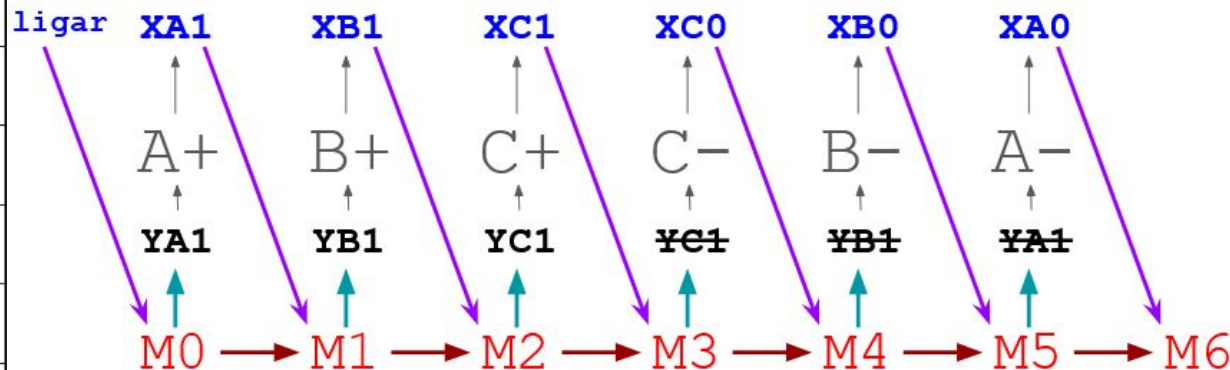
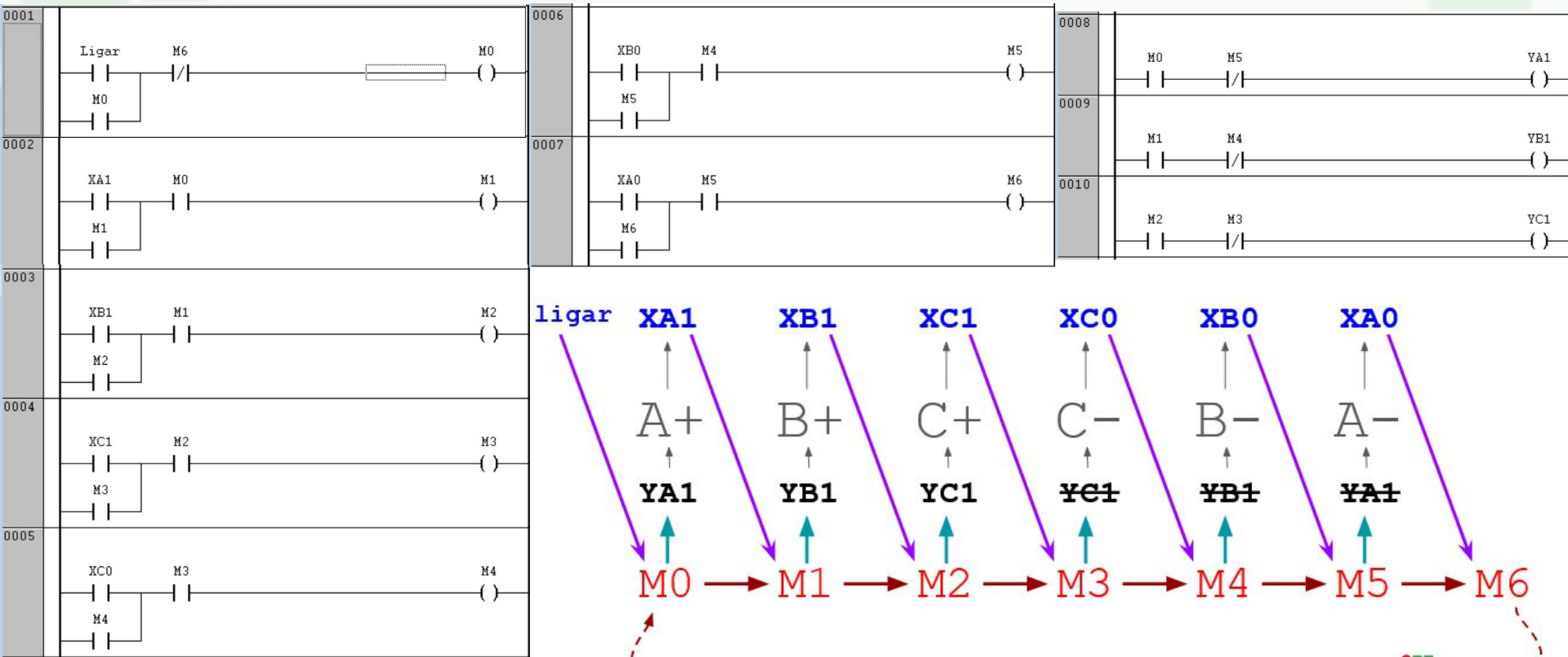
## Passo 5: Habilitação das Memórias



## Passo 6: Desabilitação das Memórias



## Passo 7: **Acionamento de Saídas**



Desligar/Desabilitar

# Exercícios



Aplicar o método de Maximização de contatos, para produzir as seguintes sequências de acionamento:

1) A+ C+ C- B+ A- B-

2) A+ A- B+ B- C+ C-

3) A+ C+ [A- B+] B- C-

4) C+ B+ B- A+ C- B+ B- A-

Obs.: [A- B+]: Movimentos simultâneos