Modeling and Automation of Industrial Processes

Modelação e Automação de Processos Industriais / MAPI

GRAFCET
(Sequential Function Chart, SFC) 3/3

http://www.isr.tecnico.ulisboa.pt/~jag/courses/mapi2122

Prof. José Gaspar, rev. 2021/2022

GRAFCET   **vs Ladder**

GRAFCET/SFC can be converted directly to ladder logic

Memory variables:
Assign one Boolean variable to each step (s_i) and transition (t_j)

Code to run once:
1.   Initialize steps and transitions
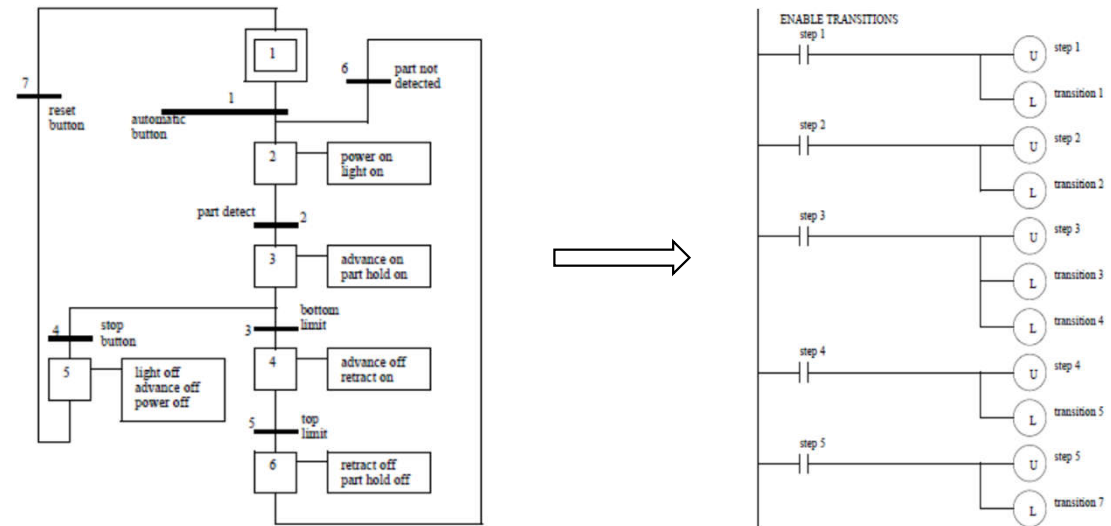Code to run at **every scan cycle**:
2. Check transitions and activate steps
3. Perform activities for steps
4. Enable transitions (jump to 2.)

Ref: [Hugh Jack 2008]

GRAFCET/SFC can be converted directly to ladder logic

Ref: [Hugh Jack 2008]

Memory variables:

Assign one Boolean variable to each step (s_i) and transition (t_j).

Create memories to keep output values.
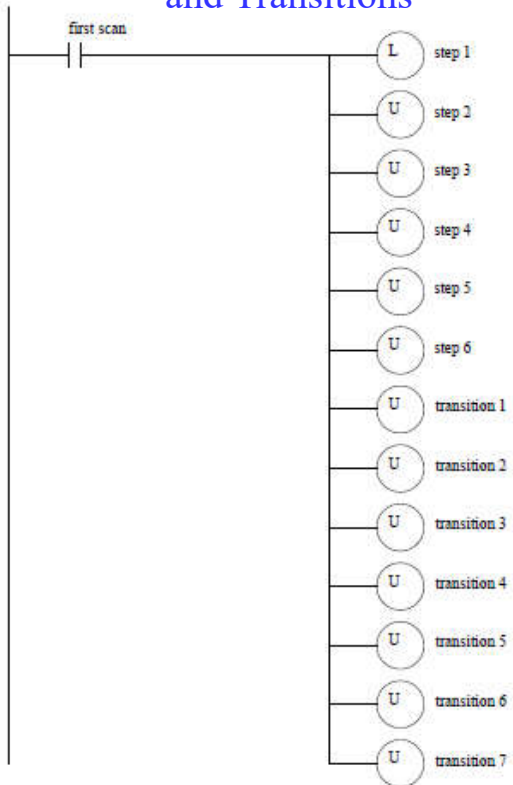
Code to run once:

1. Initialize steps and transitions

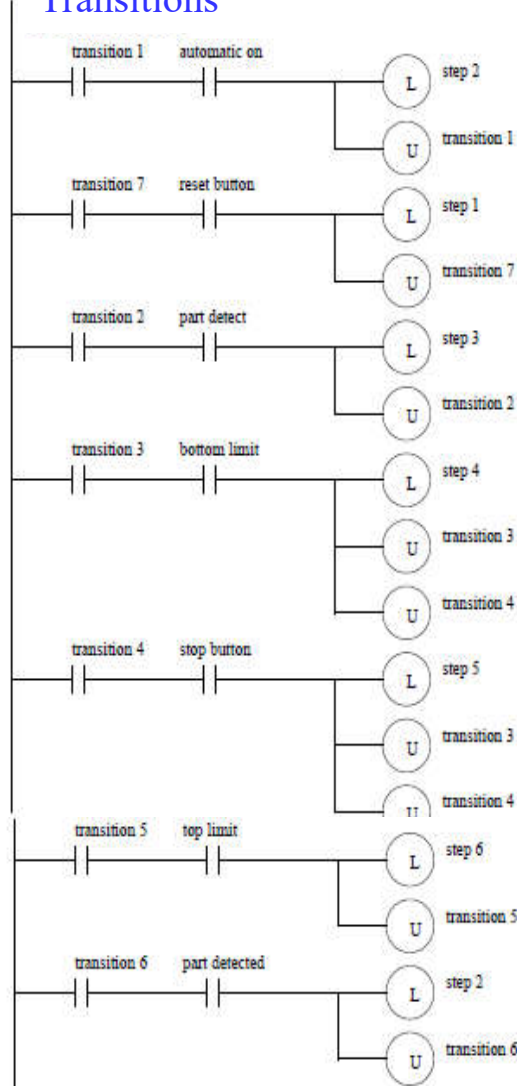Code to run at **every scan cycle**:

2. Check transitions and activate steps
3. Perform activities for steps
4. Enable transitions (jump to 2.)

The complete program, four sections *(to see separately in the next slides)*:
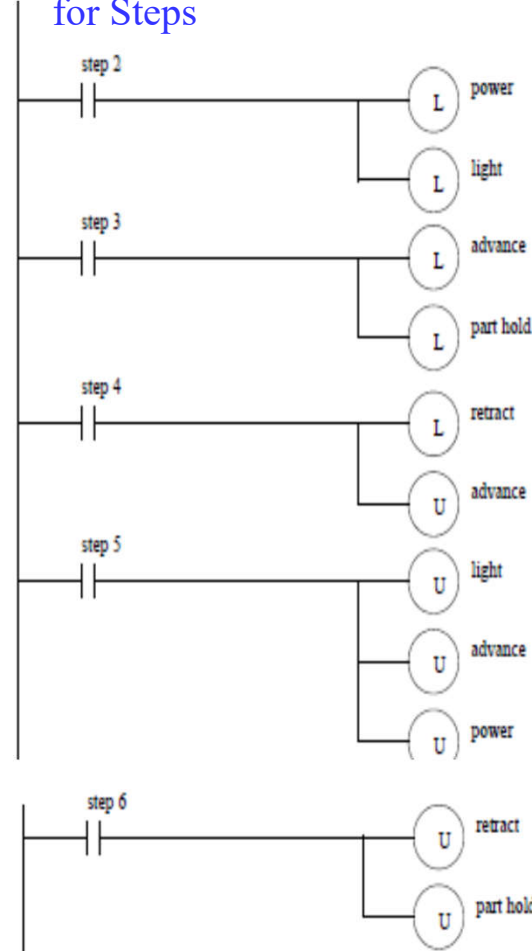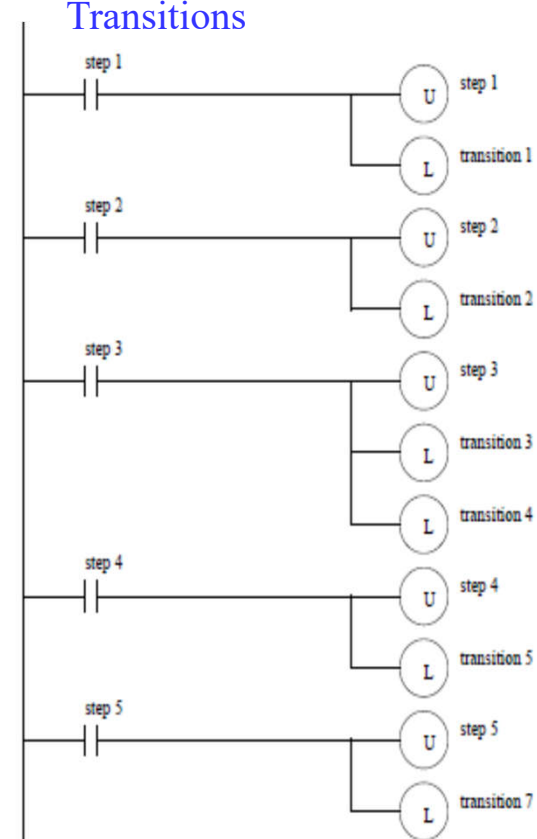
IST / DEEC / MAPI

1. Initialize steps and transitions

INITIALIZE STEPS AND TRANSITIONS

first scan

- L — step 1
- U — step 2
- U — step 3
- U — step 4
- U — step 5
- U — step 6
- U — transition 1
- U — transition 2
- U — transition 3
- U — transition 4
- U — transition 5
- U — transition 6
- U — transition 7
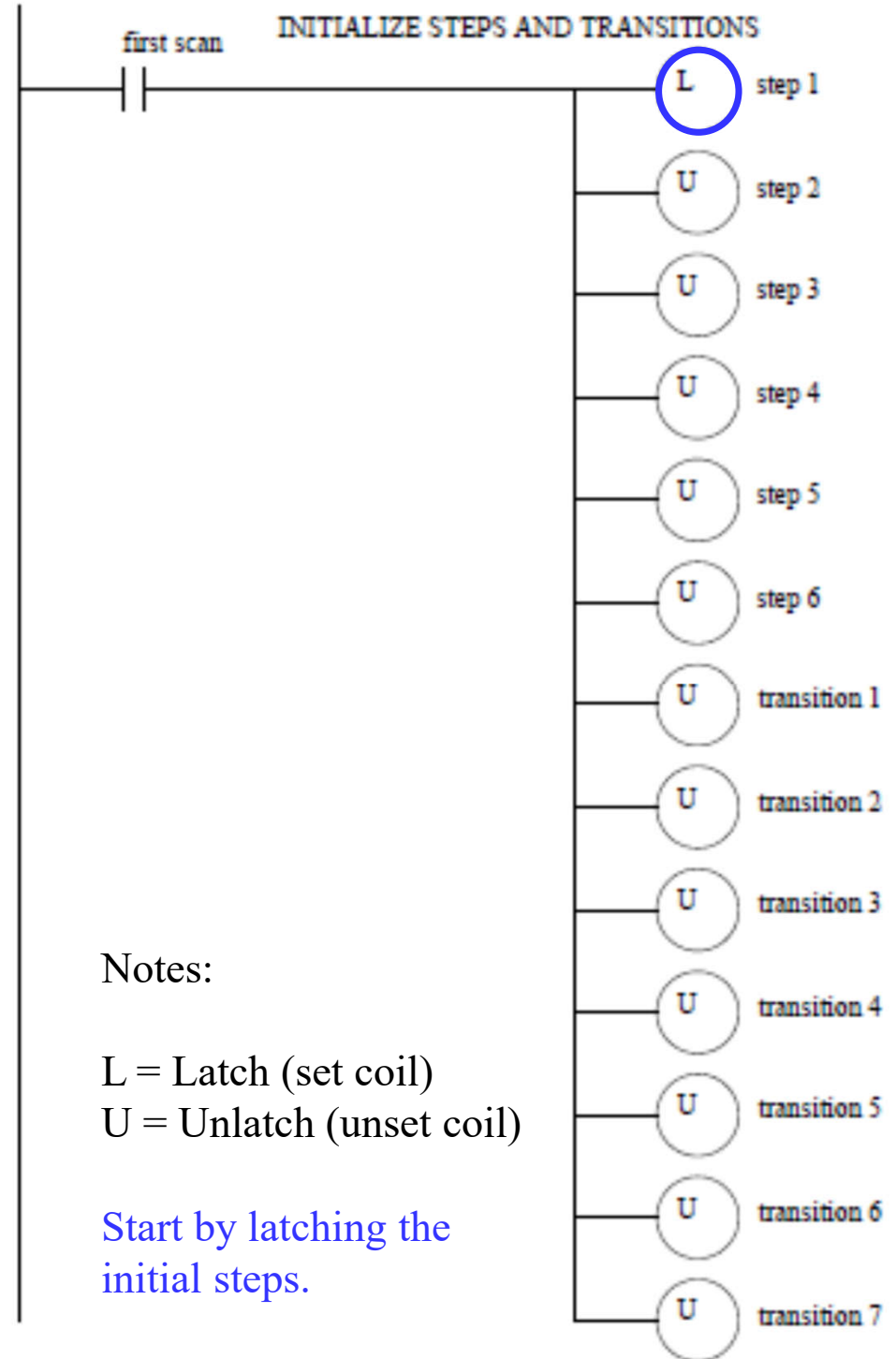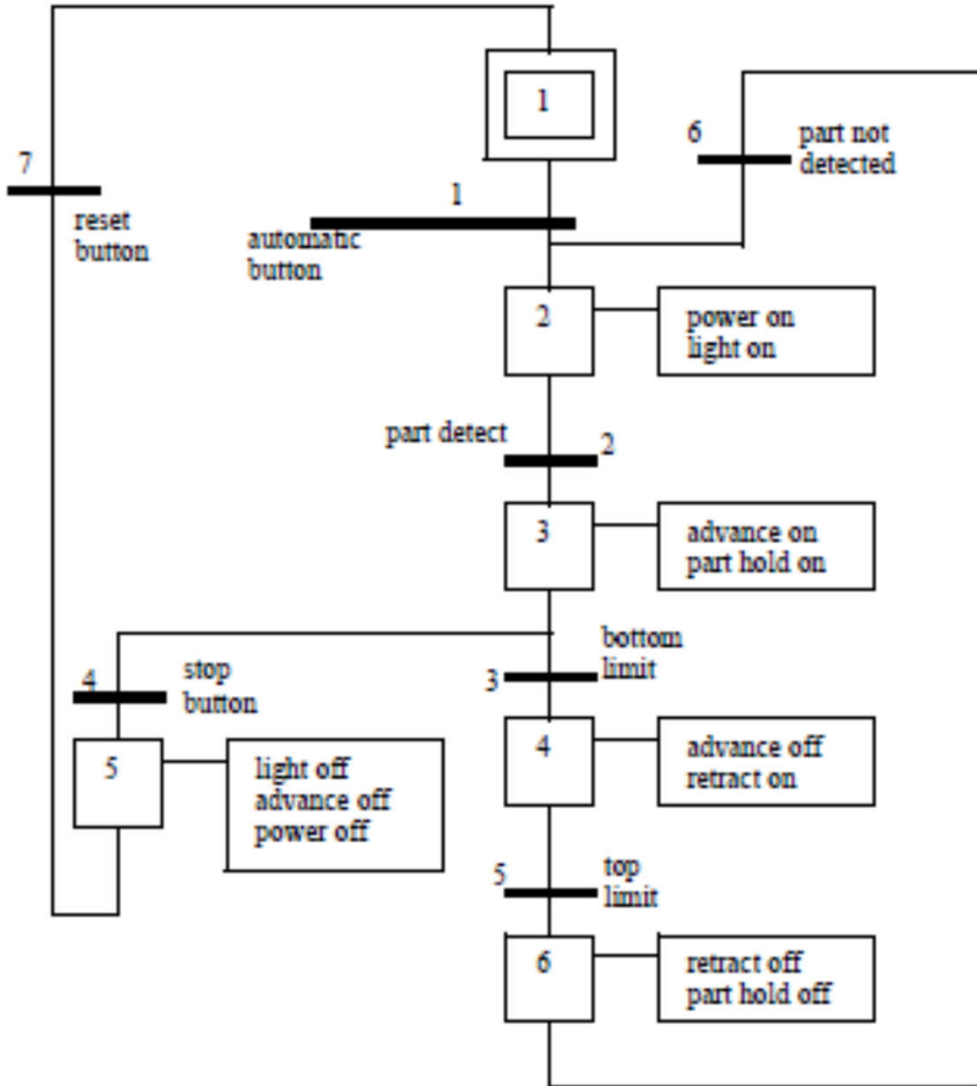
Notes:

L = Latch (set coil)
U = Unlatch (unset coil)

Start by latching the
initial steps.

2. Check transitions & activate steps

CHECK TRANSITIONS

transition 1	automatic on		L	step 2
transition 7	reset button		U	transition 1
transition 2	part detect		L	step 1
			U	transition 7
			L	step 3
			U	transition 2

(GRAFCET diagram on the left with steps 1–6, transitions labelled: reset button (7), automatic button (1), part not detected (6), power on / light on (2), part detect (2), advance on / part hold on (3), bottom limit (3), stop button (4), light off / advance off / power off (5), advance off / retract on (4), top limit (5), retract off / part hold off (6))

Ladder logic rungs:
- transition 1 — automatic on — L step 2 / U transition 1
- transition 7 — reset button — L step 1 / U transition 7
- transition 2 — part detect — L step 3 / U transition 2
- transition 3 — bottom limit — L step 4 / U transition 3 / U transition 4
- transition 4 — stop button — L step 5 / U transition 3 / U transition 4
- transition 5 — top limit — L step 6 / U transition 5
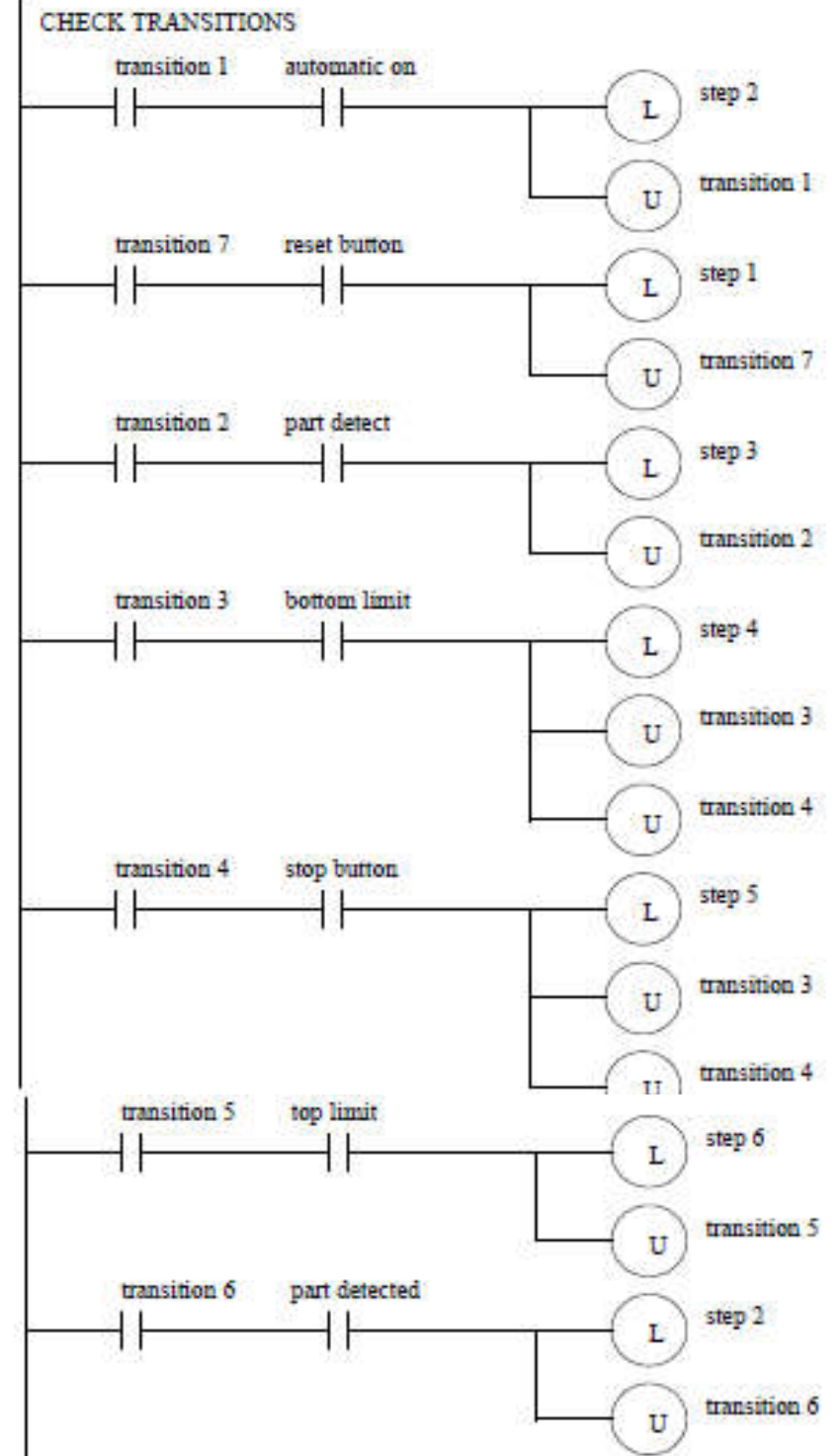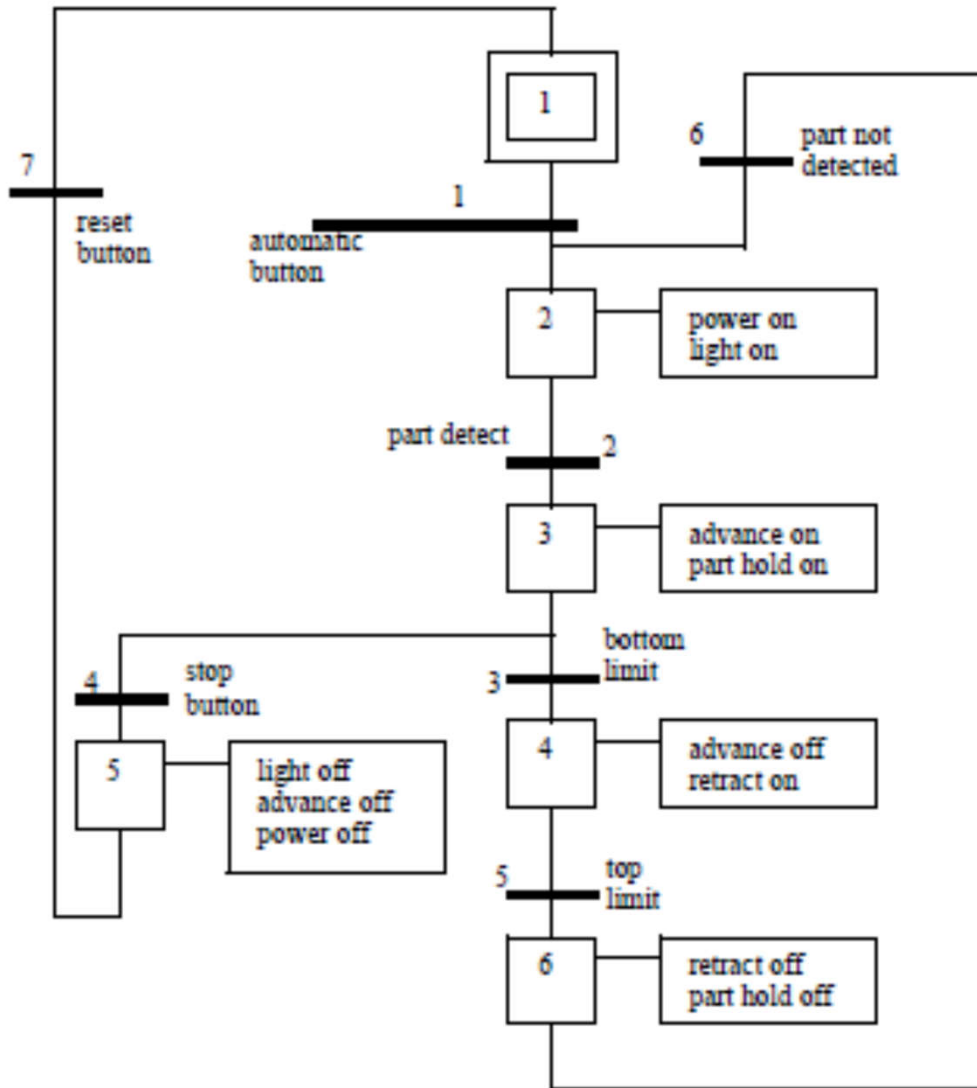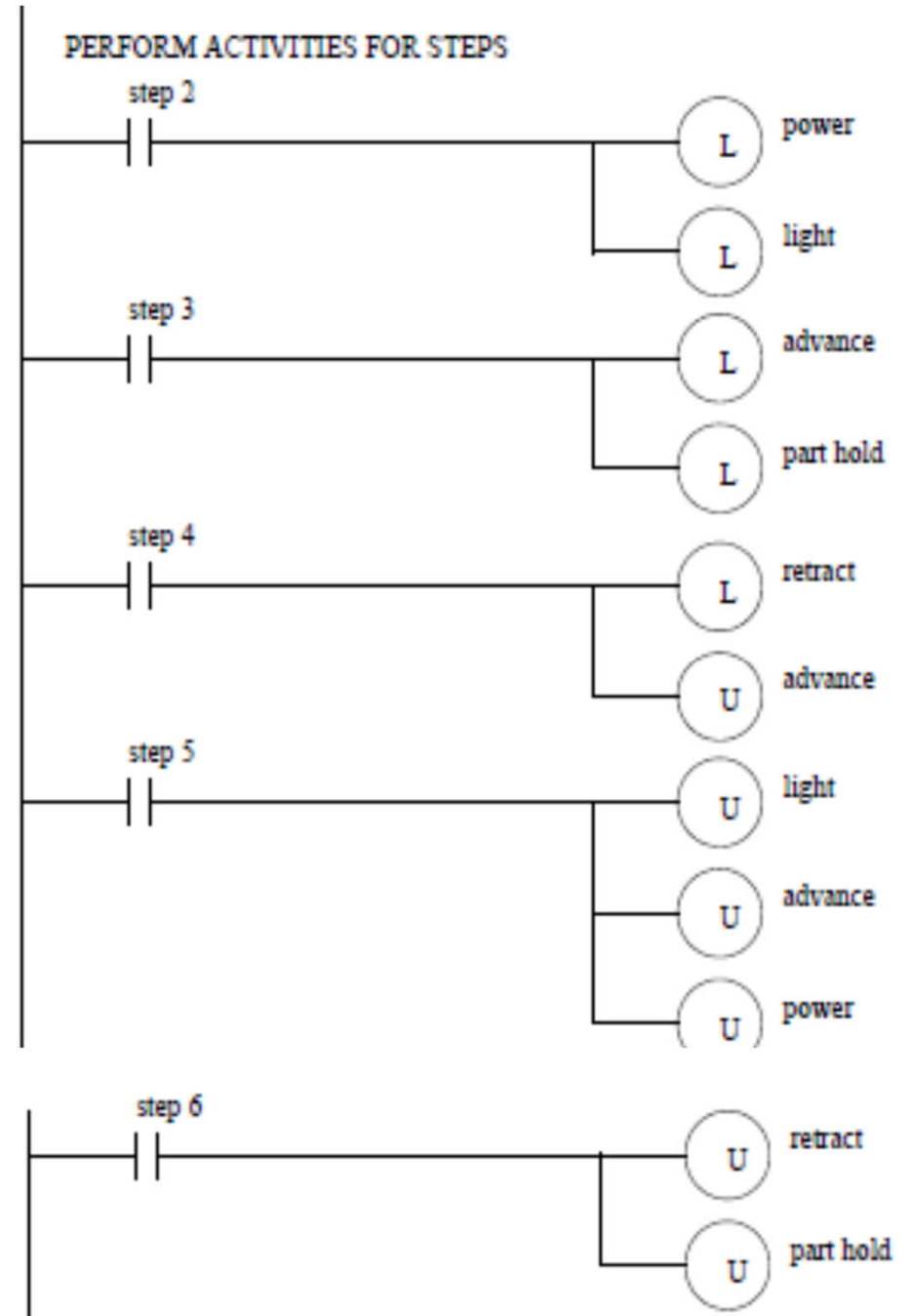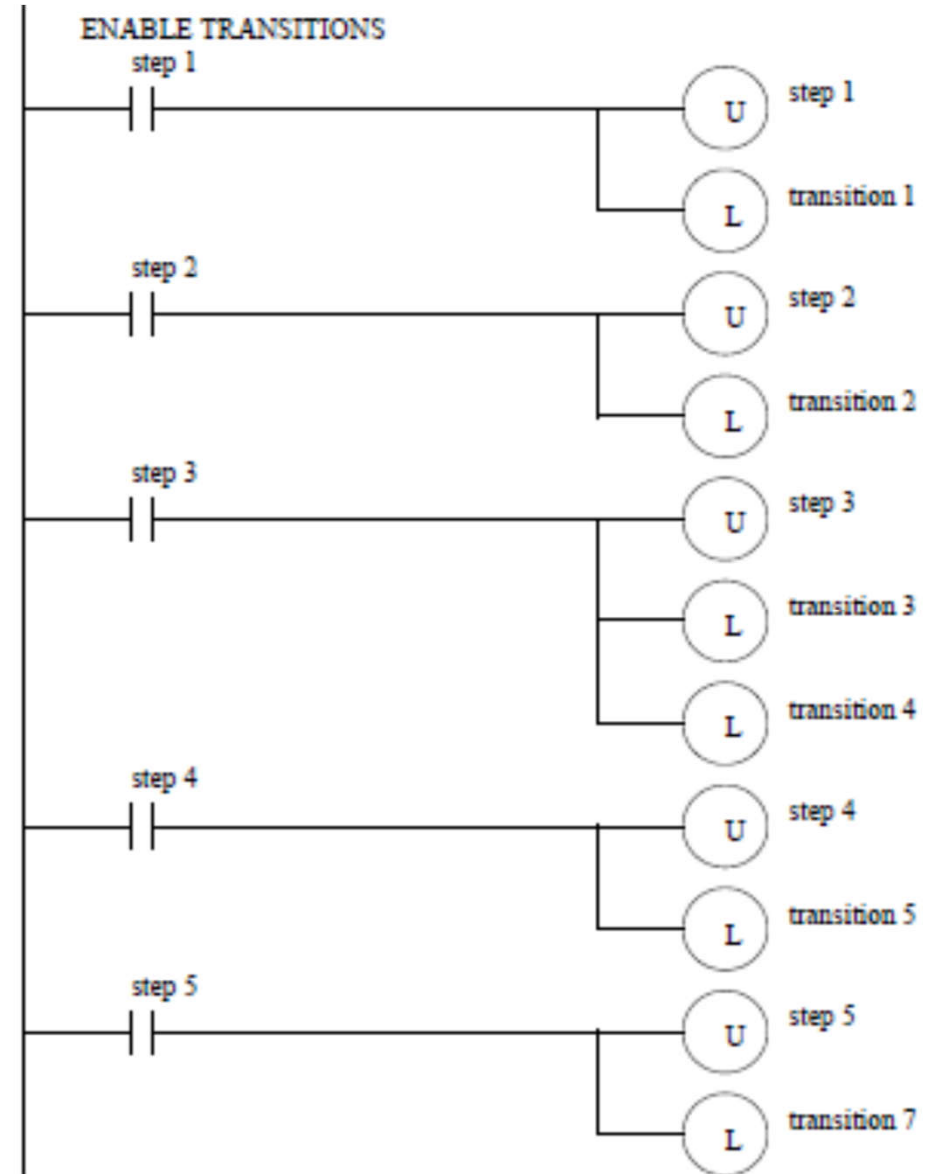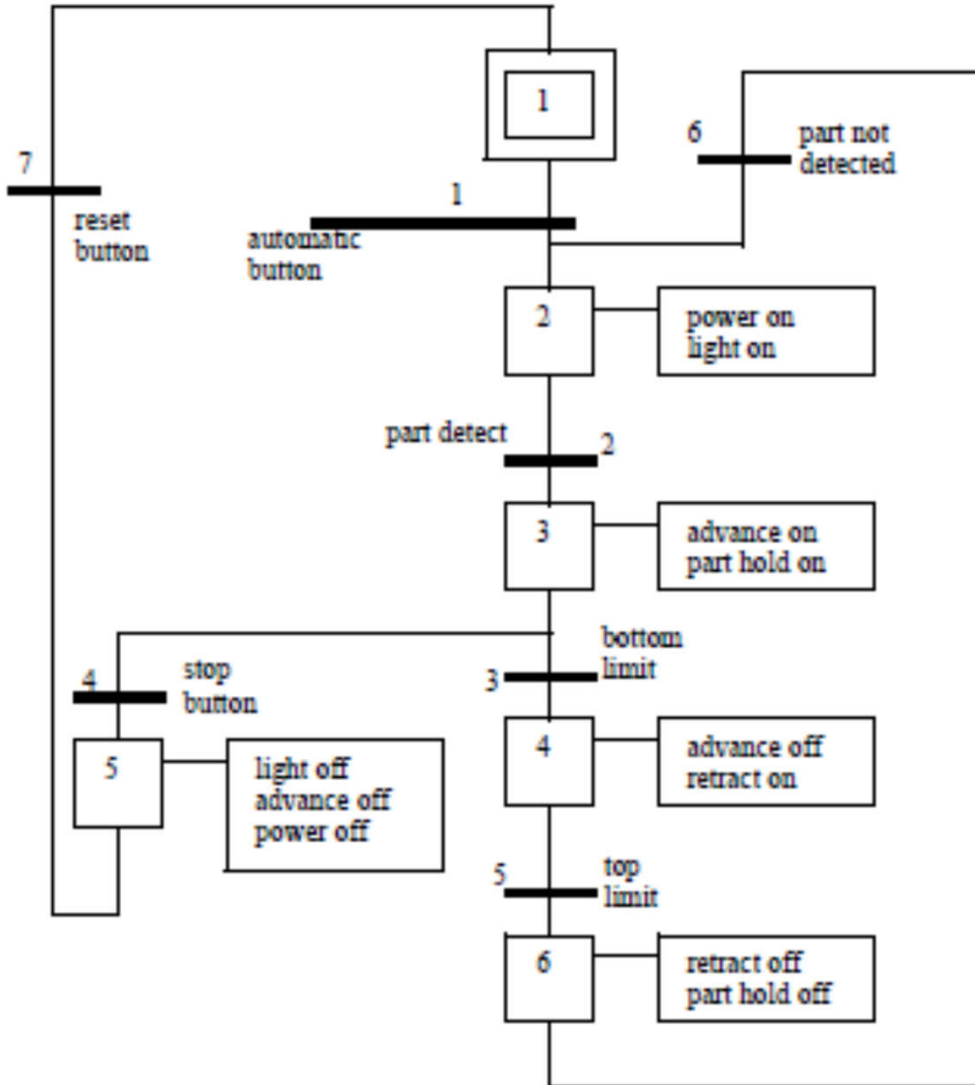- transition 6 — part detected — L step 2 / U transition 6

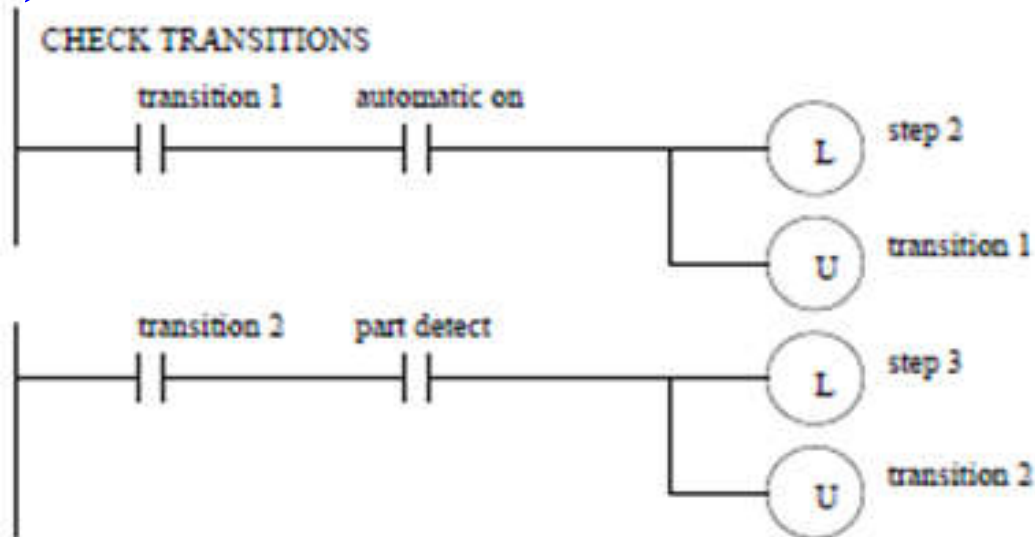## 3. Perform activities for steps

4. Enable transitions

Note: all active steps are made inactive.

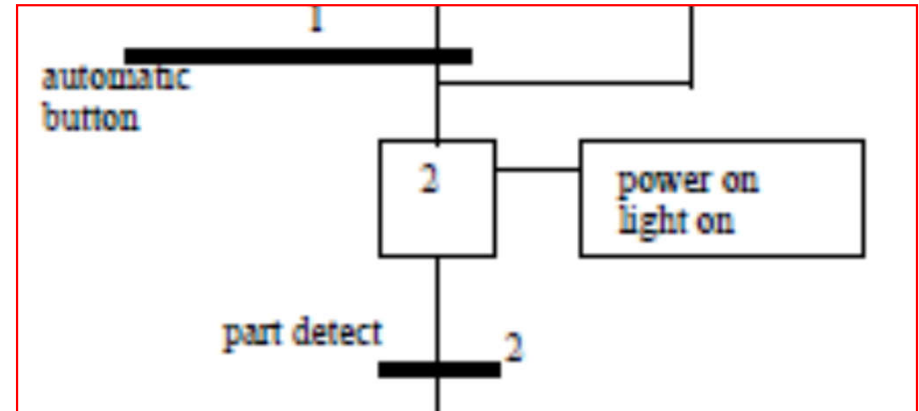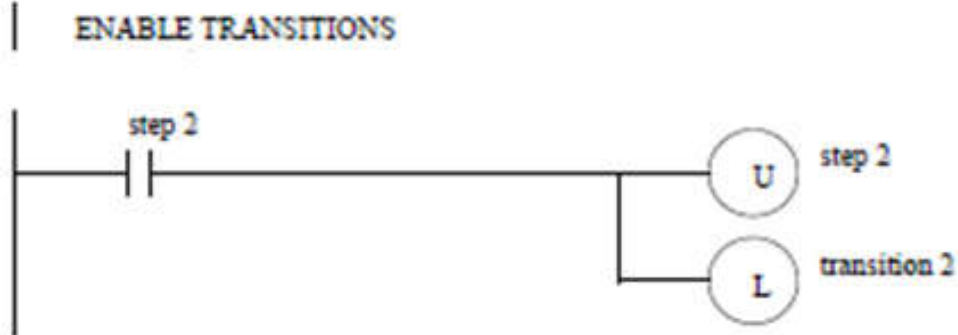# Discussion: not-keeping vs keeping steps active

1)

CHECK TRANSITIONS

transition 1   automatic on

step 2 — L

transition 1 — U

transition 2   part detect

step 3 — L

transition 2 — U

3)

ENABLE TRANSITIONS

step 2

step 2 — U

transition 2 — L

Small part of the Grafcet to analyse

2)

PERFORM ACTIVITIES FOR STEPS

step 2

power — L

light — L
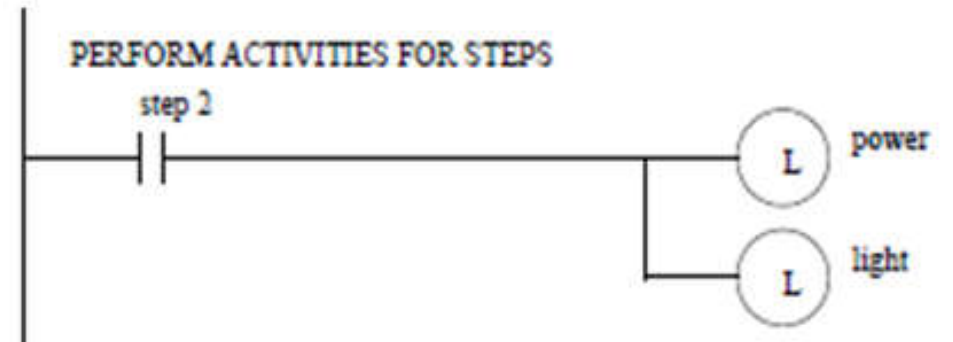
*If step2 is **active in 1** then its work is done in 2 and it is going to be made **inactive (unlatched) in 3**.*

*Note: code parts 1, 2 and 3 run in a single scan cycle; **latched outputs imply they have no spikes**.*

*Note2: **Unity Pro** is not like this, step2 gets inactive only after transposing transitions2.*

Page 9

Homework challenges:

1. Convert the ladder code shown in the previous slides to a structured text program.

2. Consider simulating the ladder diagram, ladder instructions one-by-one, saving all variables:
- Steps (1..6)
- Transitions (1..6)
- Inputs (automatic button, part detect, …)
- Outputs (power, light, advance, …)

Confirm that:
- Step variables are active *at most one scan cycle*
- Outputs are set/unset (latched/unlatched) and therefore *do not need the steps being active all the time*.

GRAFCET **Practice Problem 1**

Draw **one SFC** for one stamping press that can **advance and retract** when a **cycle button** is pushed, and then stop until the button is pushed again. The press has **limit switches** indicating stop advancing and stop retracting.

Further study: discuss the advantages of using SFC as compared with using Ladder in this problem.

From [Hugh Jack 2008]
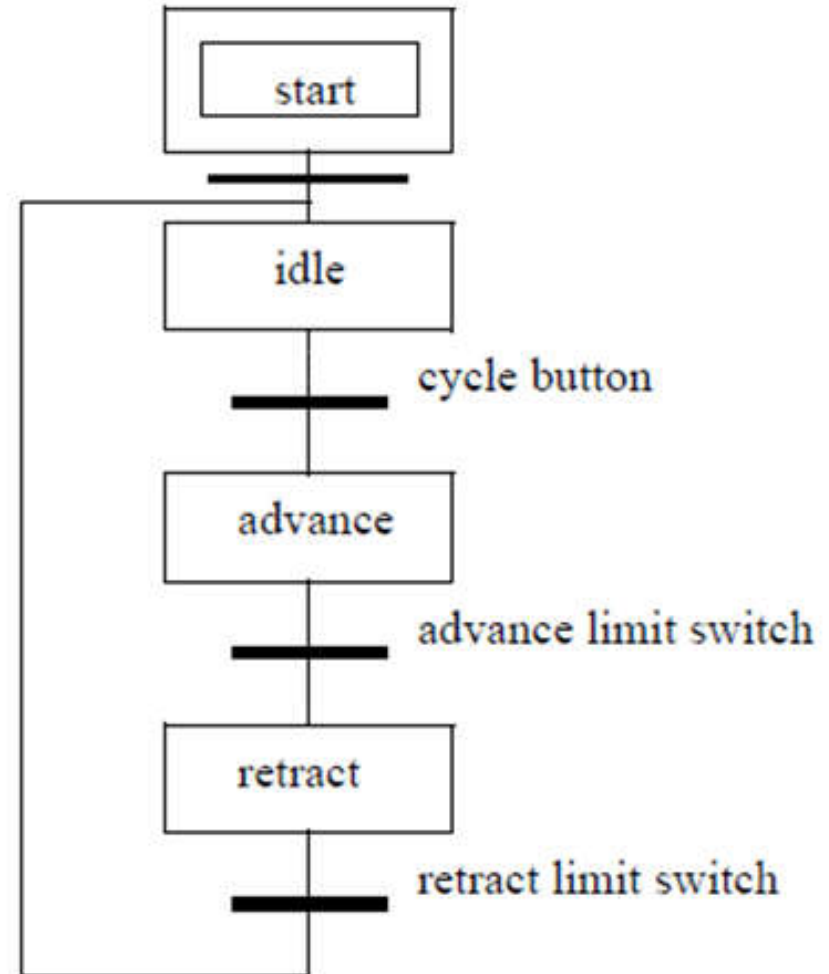
GRAFCET   **Practice Problem 2**

Develop **one SFC** for a two person assembly station. The station has **two presses** that may be used at the same time, **independently**. Each press has a cycle button that will start the advance of the press. A bottom limit switch will stop the advance, and the cylinder must then be retracted until a top limit switch is hit. The two presses are enabled only after a common starting procedure.
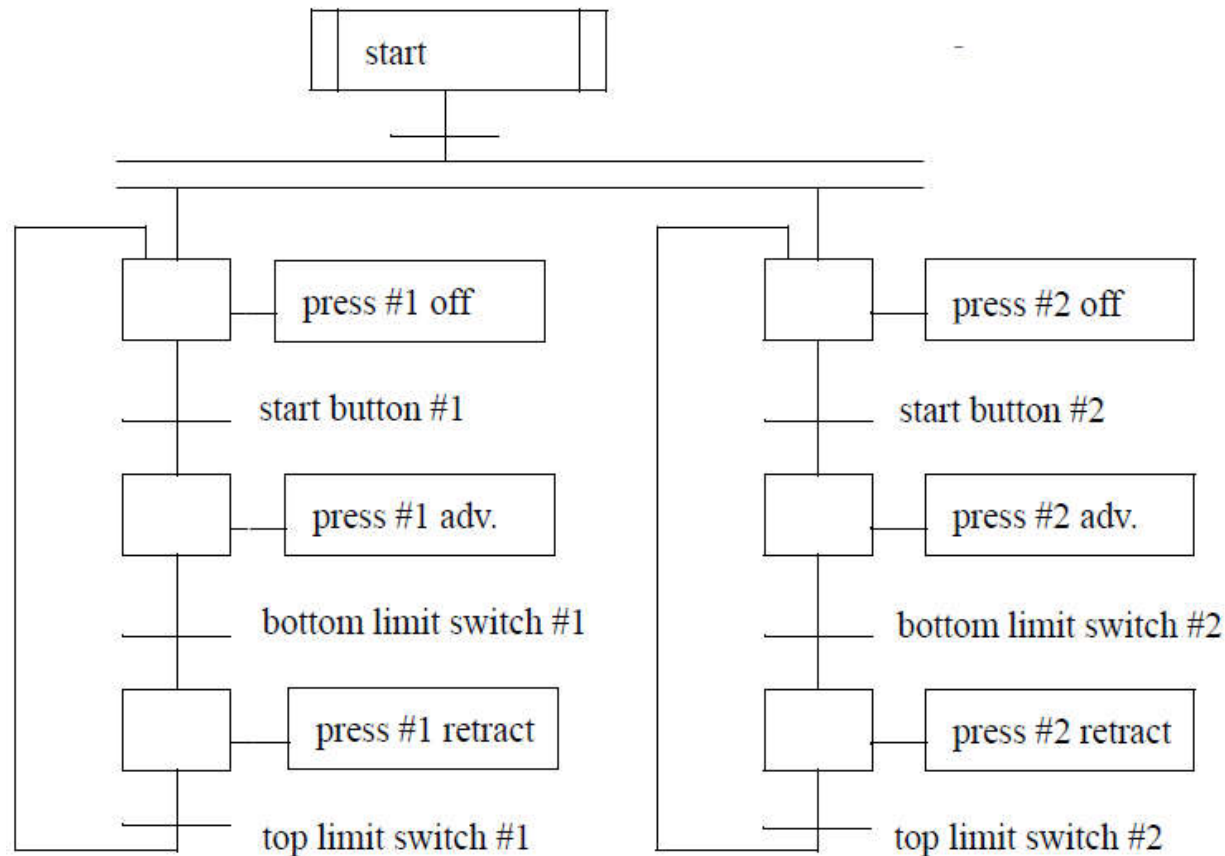
From [Hugh Jack 2008]

GRAFCET   **Practice Problem 3**

Design a garage door controller using an SFC. The behavior of the garage door controller is as follows:

- There is a single button in the garage and a single button remote control. When the button is pushed the door will move up or down.
- There are top/bottom limit switches to stop the motion of the door.

- If the button is pushed once while moving, the door will stop. A second push will start motion again in the opposite direction.
- There is a light beam across the bottom of the door. If the beam is cut while the door is closing the door will stop and reverse.

- There is a garage light that will be on for 5 minutes after the door opens or closes.

From [Hugh Jack 2008]   Page 13