CSE-3103: Microprocessor and Microcontroller

Dept. of Computer Science and Engineering
University of Dhaka

Prof. Sazzad M.S. Imran, PhD
Dept. of Electrical and Electronic Engineering
sazzadmsi.webnode.com

Processor Memory Organization

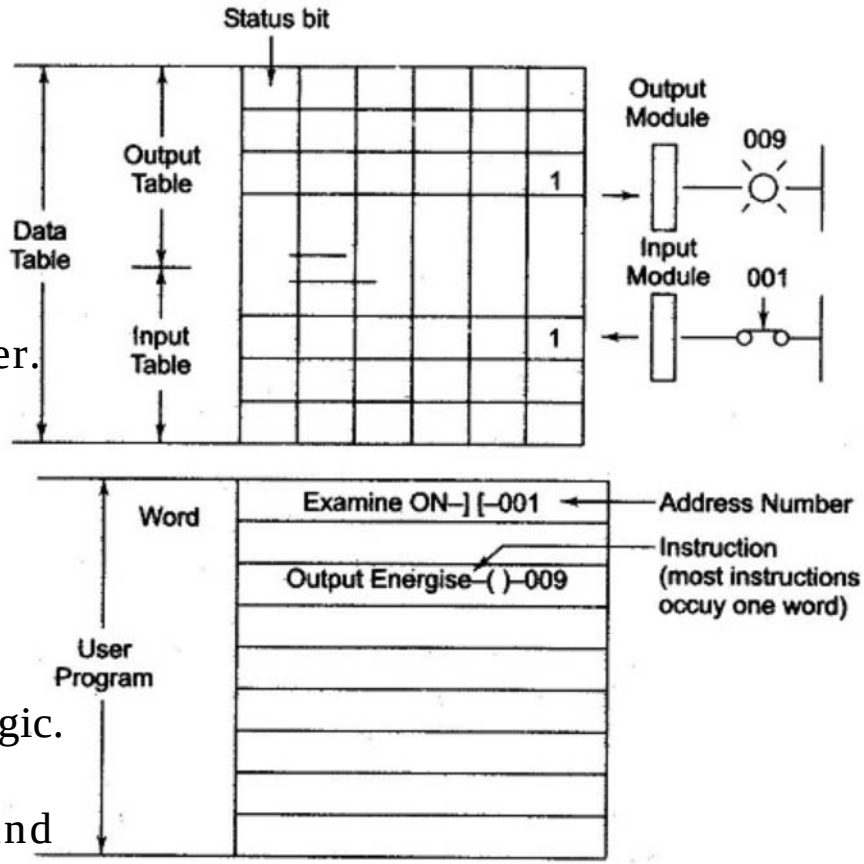
PLC programs are not interchangeable among different models →

- memory organization differences →
  - unique memory layouts,
  - file structures, and
  - addressing methods.
- I/O addressing schemes/formats vary.
- instruction set variations →
  - commands, syntax, and functions differ.

[Allen Bradley PLC Processor]

Memory organization →

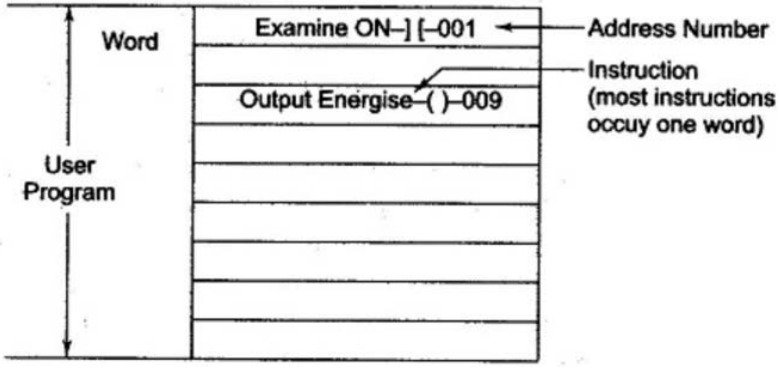
- logically divided into 2 primary sections →
  - 1) program files →
    - contains executable control logic.
  - 2) data files →
    - stores real-time data, status, and values used by program.



Processor Memory Organization

Program files →

stores user ladder logic and system routines,
logic and routines = defines machine operations.
accounts for most of total memory.

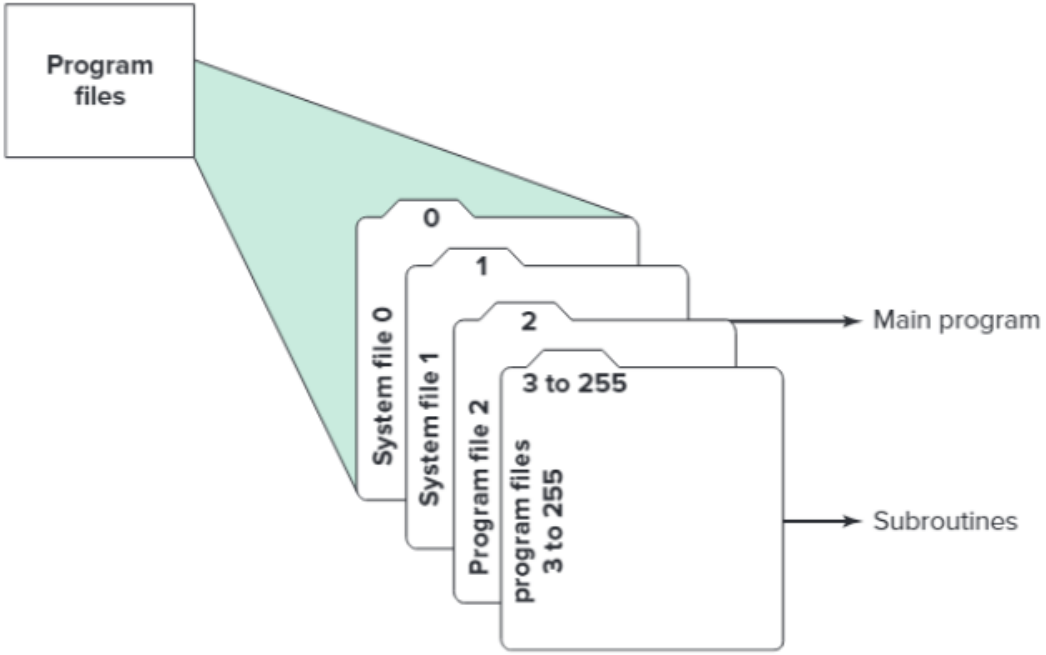


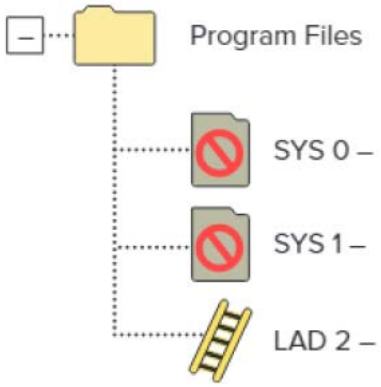
Program files include →

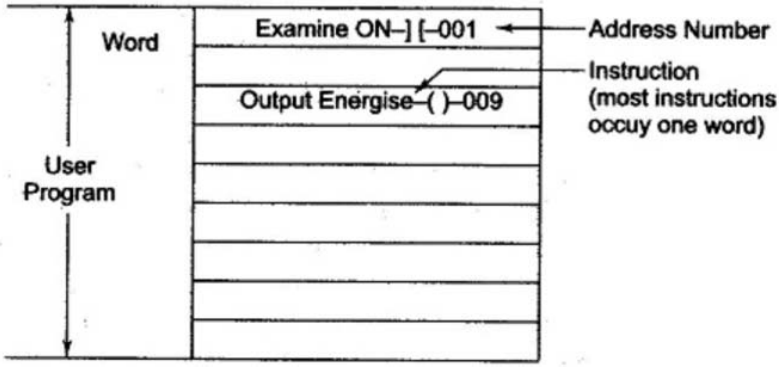
File Number	Type	Description
File 0	System functions	Holds system data: processor type, password, diagnostic info I/O configuration
File 1	Reserved	Used by processor for internal operations; not accessible to programmer
File 2	Main ladder program	Primary user-written logic; scanned repeatedly during operation
Files 3-255	Subroutine programs	Modular subprograms called from File 2 for organized, reusable logic

Processor Memory Organization

Program files →

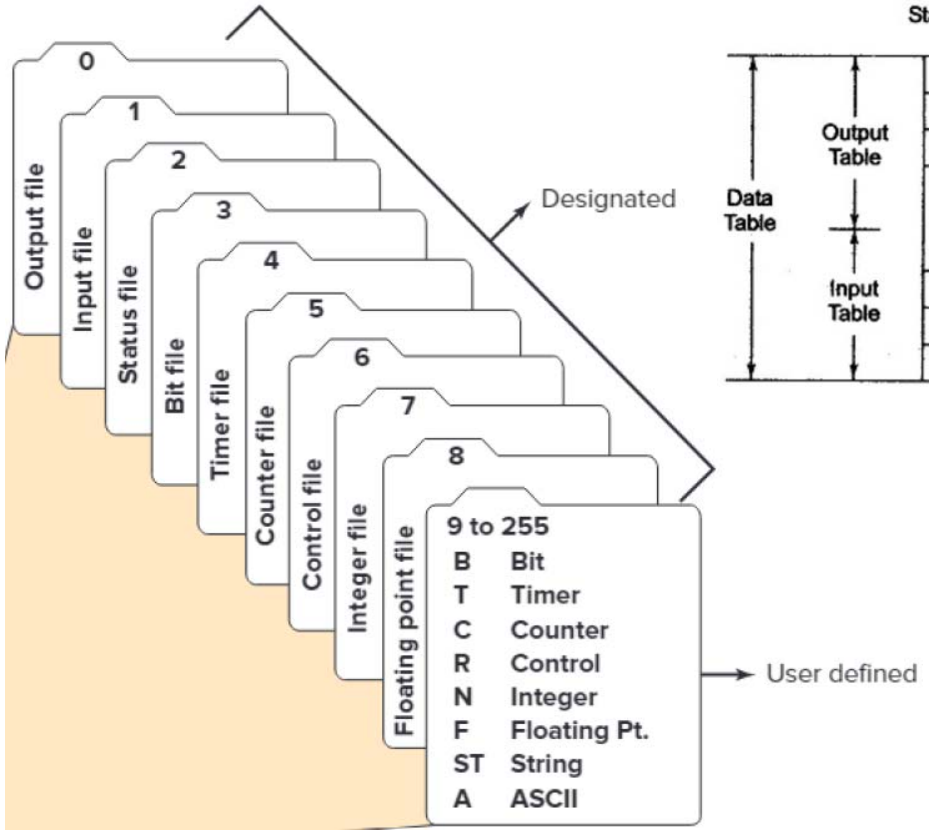


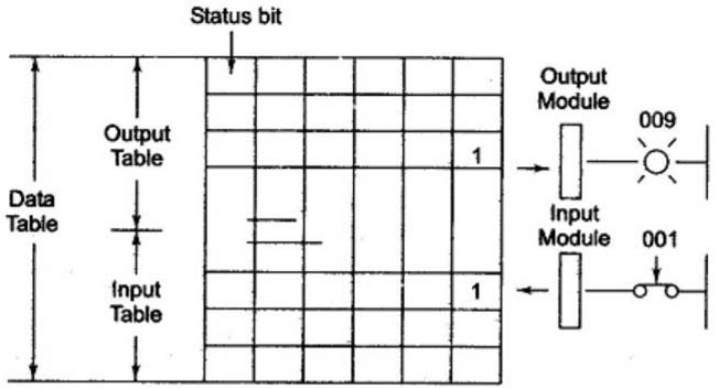


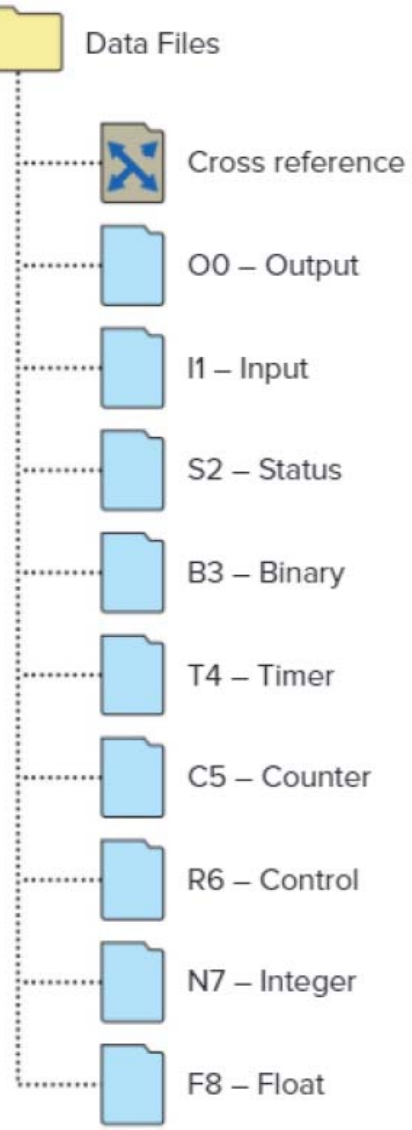


Processor Memory Organiza

Data files →



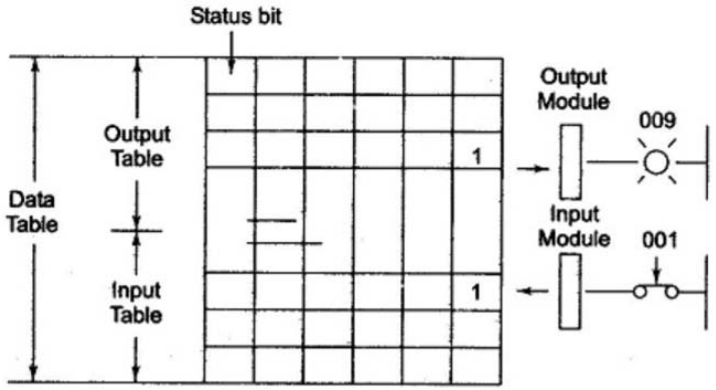




Processor Memory Organization

Data files →

act as PLC's working memory,
hold real-time information that
logic program reads from and writes to.



types of data stored →

input and output status,
timer and counter values,
internal flags (bits),
numerical data (integers, floating-point),
system status information.

access method →

referenced directly in ladder logic instructions.

Processor Memory Organization

Data files include →

File Number	Type	Description
File 0	Output	Stores ON/OFF state of physical output terminals
File 1	Input	Stores ON/OFF state of physical input terminals
File 2	Status	Contains system flags, scan time, error codes, and diagnostic data
Files 3	Bit	Used for internal relay logic; holds intermediate logic states
File 4	Timer	Holds preset, accumulated values, and status bits (EN, DN, TT)
File 5	Counter	Stores preset, accumulated counts, and status bits (CU, CD, DN)
File 6	Control	Tracks position/length for shift registers, sequencers, and FIFOs
File 7	Integer	Stores 16-bit signed integers (-32,768 to 32,767)
File 8	Float	Stores 32-bit floating-point numbers (decimal values)
File 9-255	User-defined data files	Allows programmers to organize custom data types, arrays, structures, and application-specific storage, beyond standard system-defined files (0-8)

Processor Memory Organization

3 parts of I/O address format →

used to identify I/O points in PLC memory.

format → [Type]:[Slot]/[Terminal]

Part 1 →

I for input, O for output.

colon to separate module type from slot.

Part 2 →

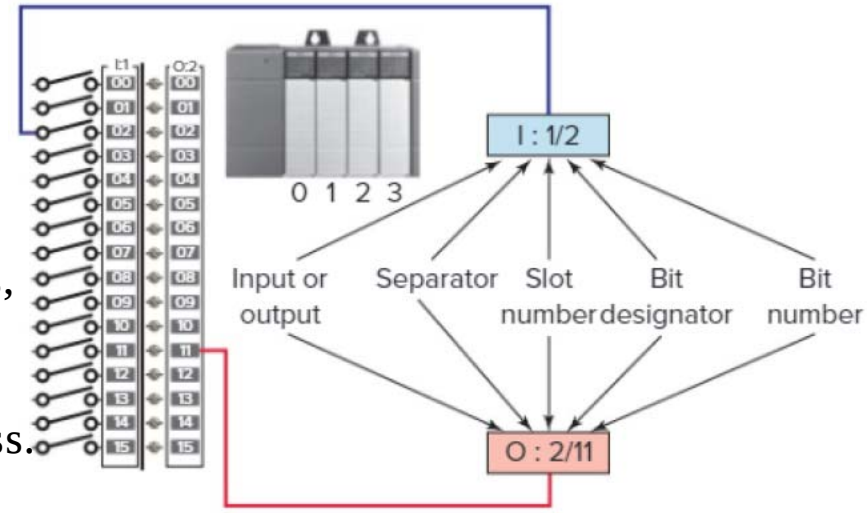
physical or logical slot where I/O module is installed,

forward slash to separate slot from terminal screw.

Part 3 →

specific terminal on module

where device is wired.



Data storage →

PLC stores all data in structured memory areas,

memory areas = Data Tables or Data Files.,

based on 16-bit operations.

data are accessed by specifying memory address.

Processor Memory Organization

Input image table file →

mirror of physical input status,
stored in memory.

Operation →

each connected input →

has bit in input image table,
= terminal to which input is connected.

during I/O scan, PLC reads all input terminals.

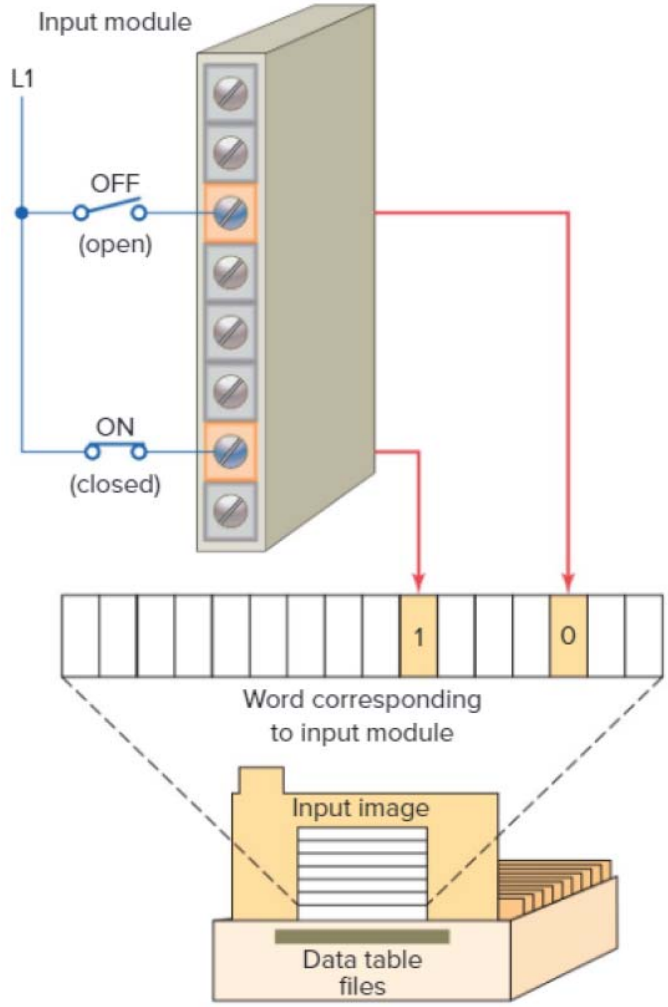
voltage detected →

stores 1 (ON) in corresponding bit,

no voltage →

stores 0 (OFF) in corresponding bit.

table is updated continuously to reflect
real-time input states.



Processor Memory Organization

Output image table file →

holds ON/OFF states that
PLC intends to send to outputs.

Operations →

each connected output →

has bit in output image table,
= terminal to which output is connected.

ladder logic determines desired output states,
program writes 1 (ON) or 0 (OFF) to
corresponding bit in output image table.

during I/O update →

physical outputs are set to match table.

Program Scan

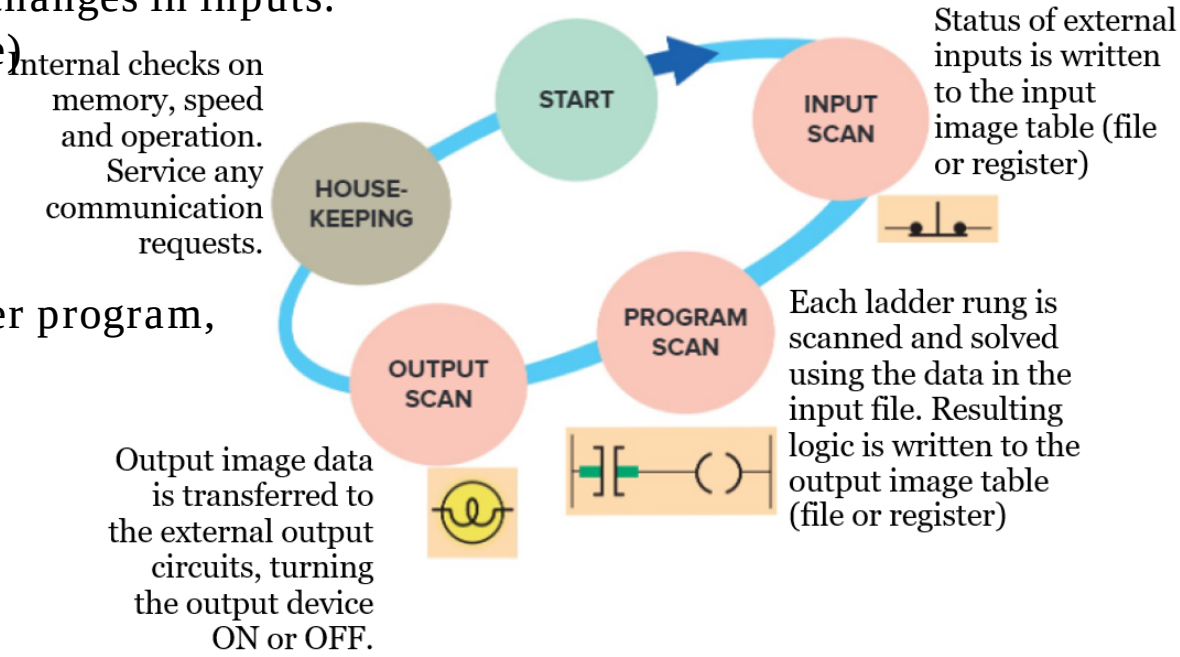
Program scan cycle →
inputs can change at any time.
continually repeats cycle in RUN mode.

scan cycle time →

time to complete scan cycle.
how fast scanner can react to changes in inputs.
< 2×(fastest input change time)
= 1 to 10 ms.

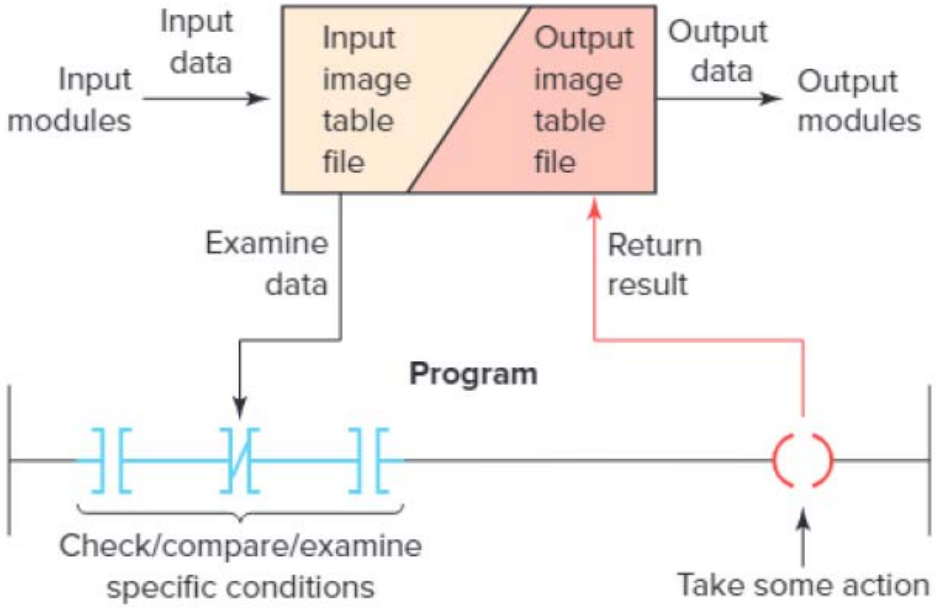
scan time is function of →

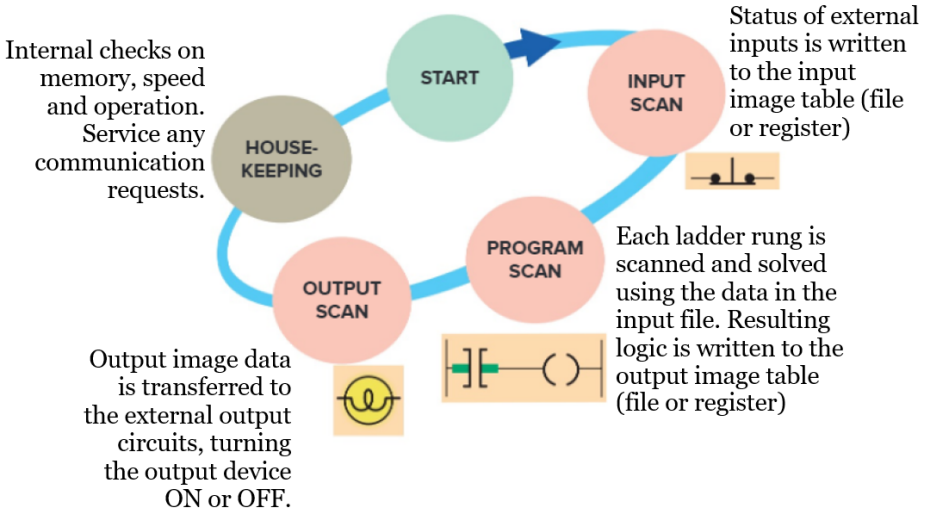
speed of processor module,
length and complexity of ladder program,
type of instructions executed,
number of I/O points.



Program Scan

Overview of data flow during scan process,
= Tasks of PLC processor for each rung execution →





Program Scan

Electrical continuity →

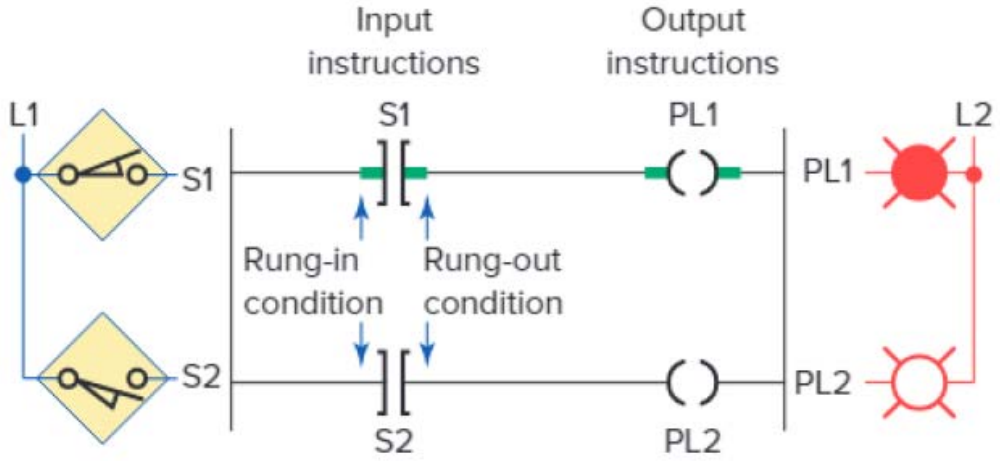
in rung of any hardwired circuit,
required for current to flow and energize load.

Logical continuity →

continuous path of true conditional instructions in rung.
required for output to turn status bit = 1 (ON).

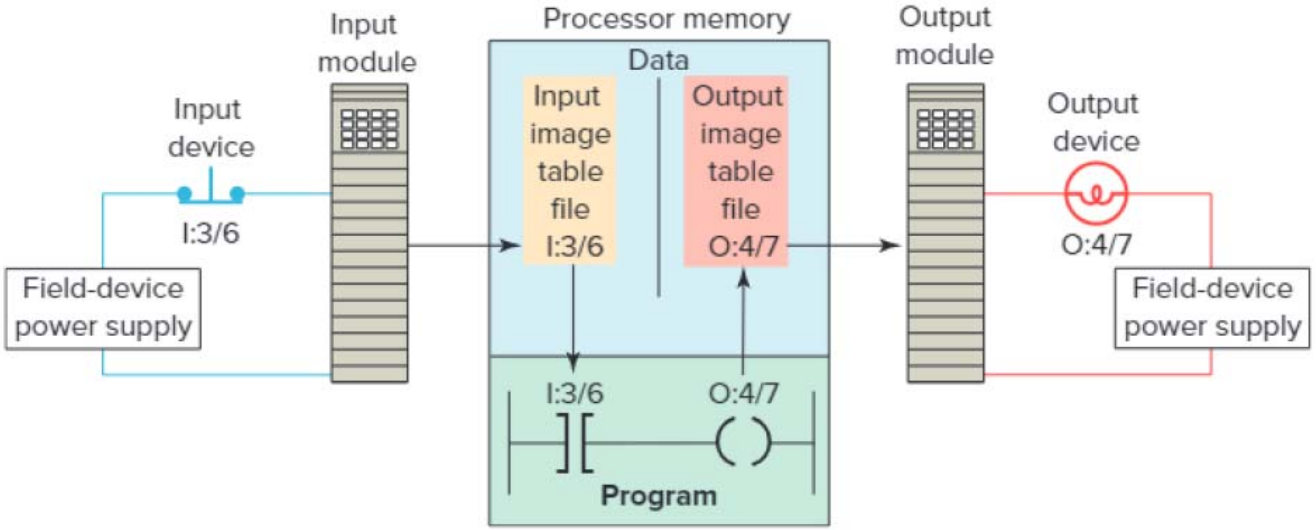
Evaluating ladder logic rung instructions →

controller evaluates instruction rung-condition-in,
sets rung-condition-out to match results of evaluation.



Program Scan

Scan process applied to single rung program →



Step	Action
1	Push button closes, voltage detected at I:3/6
2	Input Scan: I:3/6 bit set to 1 in Input Image Table
3	Program Scan: Evaluate rung, I:3/6 is TRUE
4	Logical continuity exists → set O:4/7 to 1 in Output Image Table

Step	Action	Result
5	Output Scan: Write 1 to O:4/7 terminal	Motor starter energizes
6	Button released → I:3/6 bit set to 0	
7	Next scan: I:3/6 is FALSE → O:4/7 set to 0	Motor turns OFF

Program Scan

Scan process applied to multiple rung program →

PLC processes rungs sequentially from top to bottom.

sense voltage of input terminals,

absence or presence of voltage →

0 or 1 is stored into memory bit location.

location → particular input terminal.

update Input Image Table (all inputs at once).

using input table values →

evaluate each rung in order.

write results to Output Image Table →

conditions controlling output are met =

writes 1 in its memory location,

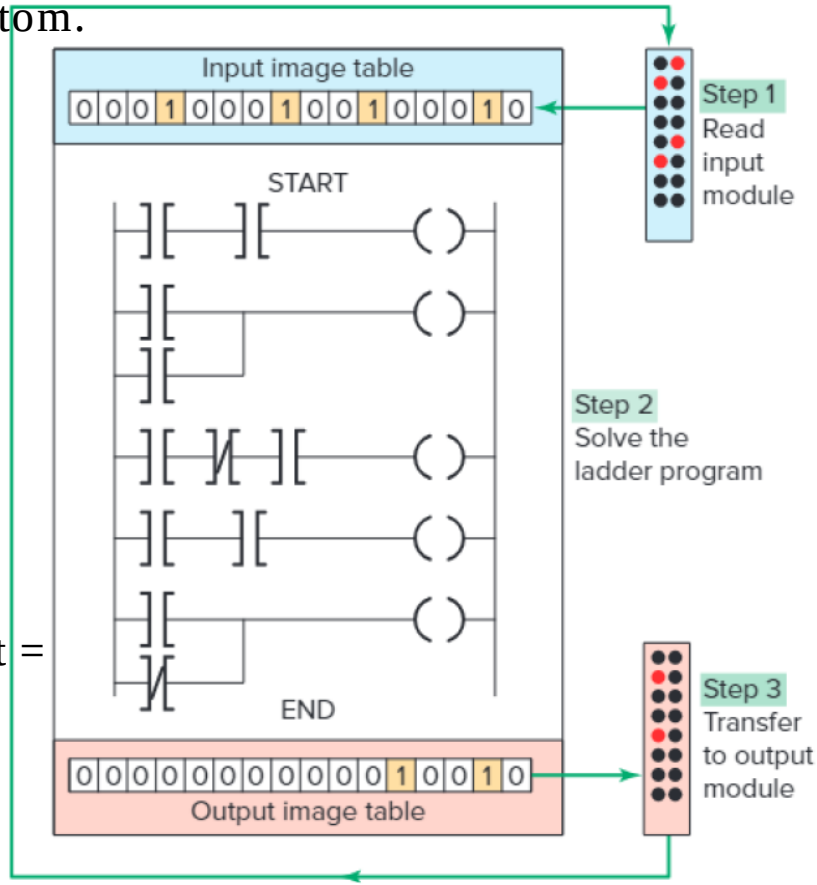
conditions controlling output are not met =

writes 0 in its memory location,

transfer output table results to output module →

update actual states of output devices,

= switching output devices ON (1) or OFF (0).



Program Scan

2 scan patterns to accomplish scan function →

1) horizontal scan by rung method →
processes each rung completely left-to-right
before moving to next rung

2) vertical scan by column method →
processes each column top-to-bottom,
page by page