Chapter 3: Memory Organization and Addressing
Memory Organization & Addressing

Learning objectives

- Understand memory organization and addressing for the following PLCs:
  - IEC 61131-3 Model
  - Allen-Bradley ControlLogix
  - Allen-Bradley PLC-5/SLC-500
IEC 61131-3 Model

Technology Overview

- **IEC 61131-3** Protocol
- International Electrotechnical Commission (IEC)
- Vendor Independent Standardized Programming Language for Industrial Automation

- Specifies Syntax, Semantics and Display for:
  - Ladder Diagram
  - Sequential Function Charts
  - Function Block Diagram
  - Structured Texts
  - Instruction Lists

- IEC 61131-3 refers to **Part 3** of IEC 61131 suite
  - Part 1 – General Overview
  - Part 2 – Hardware
  - **Part 3 – Programming Language**
  - Part 4 – User Guidelines
  - Part 5 - Communication
Memory Organization & Addressing

IEC 61131-3 Model

IEC 61131-3 Protocol: A mutually agreed upon set of rules, conventions, and agreements for the efficient and orderly design of PLCs

Layered Software Architecture
- Each layer hides details of lower layers

Common Elements of IEC 61131-3 Architecture:
- **Configuration** (Top layer)
  - It describes entire software (program + data) for one PLC
  - Each PLC, within an organization, has a separate configuration
  - Includes resources, memory addresses for I/O channels, system capabilities
- **Resource** (within Configuration layer)
  - Provides support function for the execution of IEC programs
  - Configuration may consist of one or more resources
  - Provides an interface between a program and a I/O ports of the PLC
IEC 61131-3 Model
Common Elements of IEC 61131-3 Architecture (continued)

**Tasks** (within Resource layer)
- Control the execution of a set of programs and/or function blocks
- Programs/function blocks can be executed periodically or can be event driven
- Programs must be assigned to a task in order to be executed
  - The task must be configured to either execute periodically or upon a trigger
- Resource layer may consist of one or more Tasks

http://www.personal.kent.edu/~asamba/tech43550/iec61131.pdf
IEC 61131-3 Memory Model
IEC 61131-3 Memory Model

Variables

- **Local Variable**
  - Defined within the software element
  - Accessible only by the software element
  - Software Element: Function Block, Resource or Configuration

- **Global Variables**
  - Defined for a configuration
  - Accessible to all elements contained in the configuration
    - Global Config variable can be accessed by all software elements
    - Global Program Variable can be accessed by all function blocks in the program

- **Directly Represented Variables**
  - Refers to memory and I/O locations
A-B ControlLogix Memory

[Diagram showing project structure with Task1 and Task2.]
Simple Ladder Logic

NOT Operation

Tasks

- Configured to control execution of one or more programs
- Maximum of 32 tasks
- No more than one continuous task
- Others are periodic, scheduled at fixed interval. May be interrupted by a higher-priority periodic task.

Programs

- Consists of Routines
- Max. of 32 per task
- Executed in order as listed in task
Simple Ladder Logic

**NOT Operation**

- Possible Combinations of the 2 Switches: \(2^2\)

### NOT Truth Table

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>Light</td>
</tr>
<tr>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
</tr>
<tr>
<td>OFF</td>
<td>ON</td>
<td>OFF</td>
</tr>
<tr>
<td>ON</td>
<td>OFF</td>
<td>ON</td>
</tr>
<tr>
<td>ON</td>
<td>ON</td>
<td>OFF</td>
</tr>
</tbody>
</table>
ControlLogix Program Organization

Tasks
- Configured to control execution of one or more programs
- Maximum of 32 tasks
- No more than one continuous task
- Others are periodic, scheduled at fixed interval. May be interrupted by a higher-priority periodic task.

Programs
- Consists of Routines
- Max. of 32 per task
- Executed in order as listed in task
ControlLogix Program Organization

Routines
- Encapsulate code in single programming language
- Main Routine is first one executed
  - Must call other routines
- Fault Routine is executed if a program fault is encountered during execution

Controller Tags
- Global directly represented variables
  - Represents global controller memory
  - I/O data
ControlLogix Scan

- Control logic (program) continuously scanned.
- Data is transferred to/from I/O modules any time during the logic scan.
- If input data must remain constant, make a copy at beginning of scan and use the copy throughout scan.