# Basic PLC

# Course Contents

◎History of Programmable Controllers
◎ Relay Ladder Logic
◎ Central Processing Unit
◎ Input/Output System
◎ Programming and Peripheral Devices
◎ Programming Concepts
◎ Applications
◎ Troubleshooting and Maintenance

# INTRODUCTION TO PLCS

**Advantages of PLCs**

• Less wiring.
• Wiring between devices and relay contacts are done in the PLC program.
• Easier and faster to make changes.
• Trouble shooting aids make programming easier and reduce downtime.
• Reliable components make these likely to operate for years before failure.

# PLC  Origin

- •- Developed to replace relays in the late 1960s

- •- Costs dropped and became popular by 1980s

- •- Now used in many industrial designs

# Historical Background

The **Hydramatic Division** of the **General Motors Corporation** specified the design criteria for the first programmable controller in **1968**

Their primary goal

**To eliminate the high costs associated with inflexible, relay-controlled systems**.

# Historical Background

• The controller had to be designed in modular form, so that sub-assemblies could be removed easily for replacement or repair.

• The control system needed the capability to pass data collection to a central system.

• The system had to be reusable.

• The method used to program the controller had to be simple, so that it could be easily understood by plant personnel.

# Programmable Controller Development

1968      ☐      Programmable concept developed

1969      ☐      Hardware CPU controller, with logic instructions, 1 K of memory and 128 I/O points

1974      ☐      Use of several (multi) processors within a PLC - timers and counters; arithmetic operations; 12 K of memory and 1024 I/O points

1976      ☐      Remote input/output systems introduced

1977      ☐      Microprocessors - based PLC introduced

# Programmable Controller Development

1980      ☐    Intelligent I/O modules developed
                        Enhanced communications facilities
                        Enhanced software features
                        (e.g. documentation)
                        Use of personal microcomputers as
                        programming aids

1983      ☐    Low - cost small PLC's introduced

1985 on   ☐    Networking of all levels of PLC, computer
                        and machine using SCADA software.

# Programmable Logic Controllers
## ( Definition according to NEMA standard ICS3-1978)

A digitally operating electronic apparatus which uses a programming memory for the internal storage of instructions for implementing specific functions such as logic, sequencing, timing, counting and arithmetic to control through digital or analog modules, various types of machines or process.

# Leading Brands Of PLC

AMERICAN
1. Allen Bradley
2. Gould Modicon
3. Texas Instruments
4. General Electric
5. Westinghouse
6. Cutter Hammer
7. Square D

EUROPEAN
1. Siemens
2. Klockner & Mouller
3. Festo
4. Telemechanique

# Leading Brands Of PLC

JAPANESE
1. Toshiba
2. Omron
3. Fanuc
4. Mitsubishi

# Areas of Application

◎ Manufacturing / Machining

◎ Food / Beverage

◎ Metals

◎ Power

◎ Mining

◎ Petrochemical / Chemical

# PLC Size

1.  SMALL      - it covers units with up to 128 I/O's and memories up to 2 Kbytes.
                      - these PLC's are capable of providing simple to advance levels or machine controls.

2.  MEDIUM - have up to 2048 I/O's and memories up to 32 Kbytes.

3.  LARGE      - the most sophisticated units of the PLC family. They have up to 8192 I/O's and memories up to 750 Kbytes.
                      - can control individual production processes or entire plant.

# Tank Used to Mix Two Liquids
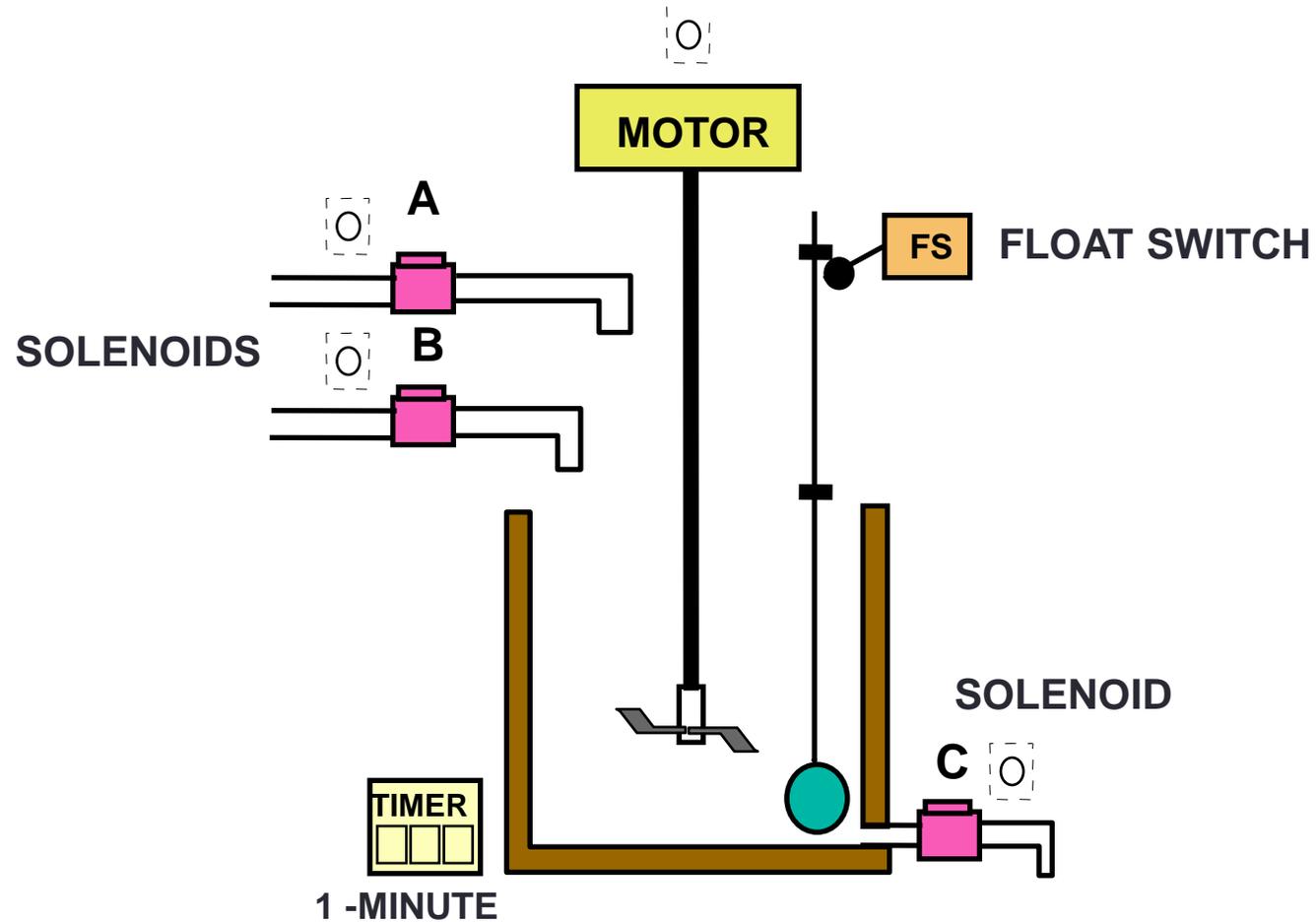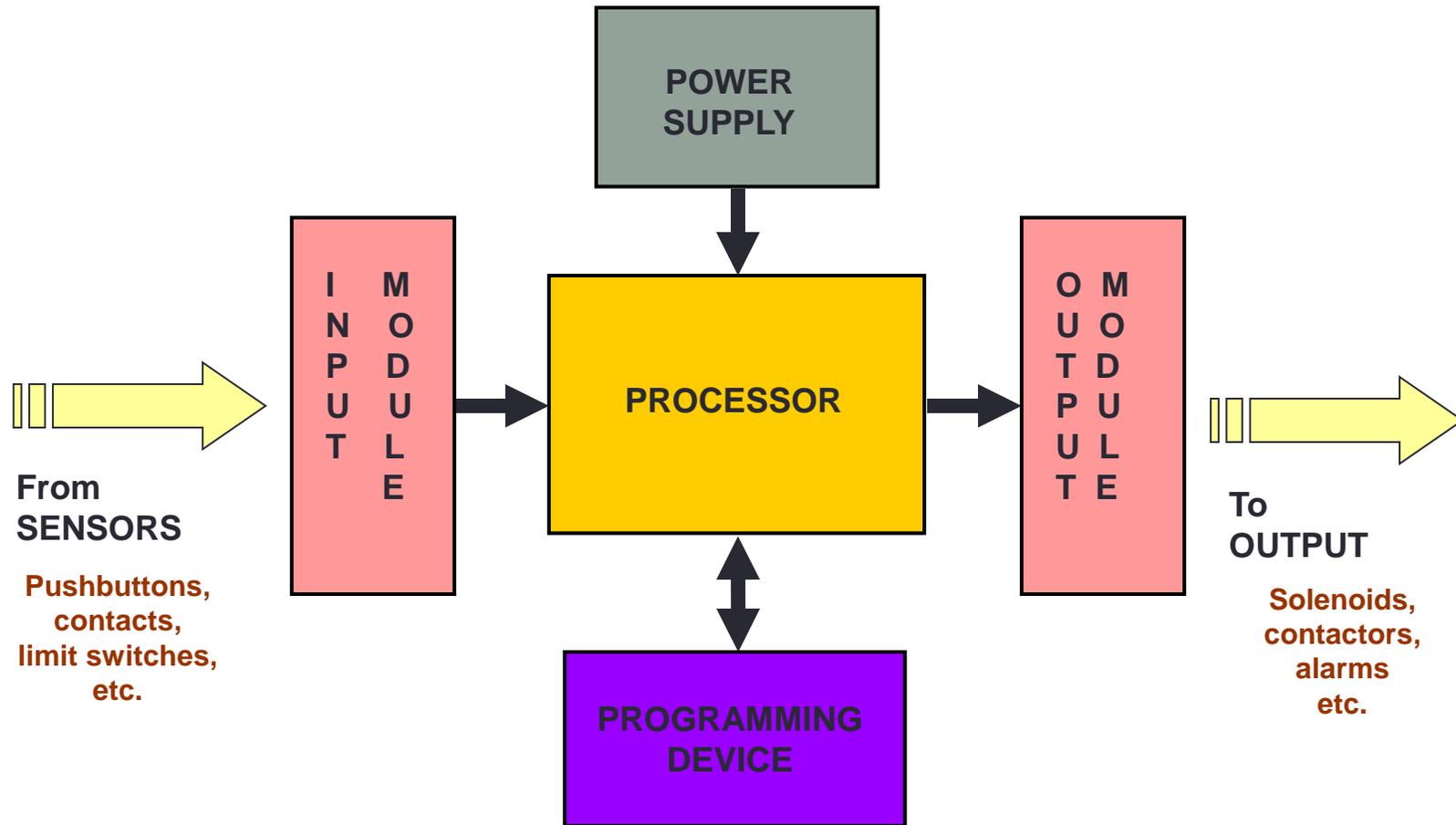
# Tank Used to Mix Two Liquids

A tank is used to mix two liquids. The control circuit operates as follows:

1. When the start button is pressed, solenoids A and B energize. This permits the two liquids to begin filling the tank.

2. When the tank is filled, the float switch trips. This de-energizes solenoids A and B and starts the motor used to mix the liquids together.

3. The motor is permitted to run for one minute. After one minute has elapsed, the motor turns off and solenoid C energizes to drain the tank.

# Tank Used to Mix Two Liquids

4. When the tank is empty, the float switch de-energizes solenoid C.

5. A stop button can be used to stop the process at any point.

6. If the motor becomes overloaded, the action of the entire circuit will stop.

7. Once the circuit has been energized it will continue to operate until it is manually stopped.

# Major Components of a Common PLC



POWER SUPPLY

INPUT MODULE

PROCESSOR

OUTPUT MODULE

PROGRAMMING DEVICE

From SENSORS

Pushbuttons, contacts, limit switches, etc.

To OUTPUT

Solenoids, contactors, alarms etc.

# Major Components of a Common PLC

**POWER SUPPLY**

Provides the voltage needed to run the primary PLC components

**I/O MODULES**

Provides signal conversion and  isolation between the internal logic-  level signals inside the PLC and the field's high level signal.

# Major Components of a Common PLC

**PROCESSOR**

Provides intelligence to command and govern the activities of the entire PLC systems.

**PROGRAMMING DEVICE**

used to enter the desired program that will determine the sequence of operation and control of process equipment  or driven machine.
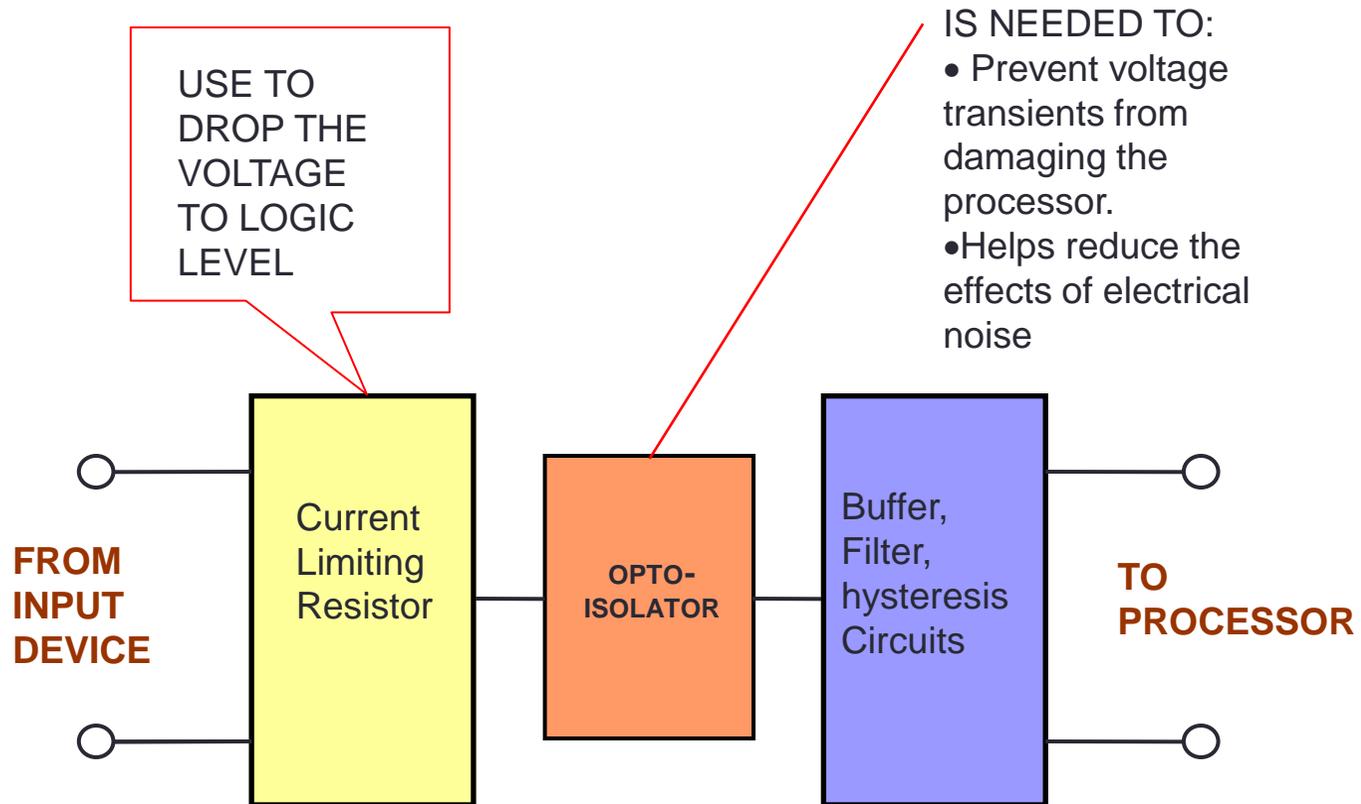
# Programming Device

Types:

◎ Hand held unit with LED / LCD display

◎ Desktop type with a CRT display

◎ Compatible computer terminal

# I/O Module

• The I/O interface section of a PLC connects it to external field devices.

• The main purpose of the I/O interface is to condition the various signals received from or sent to the external input and output devices.

• Input modules converts signals from discrete or analog input devices to logic levels acceptable to PLC's processor.

• Output modules converts signal from the processor to levels capable of driving the connected discrete or analog output devices.
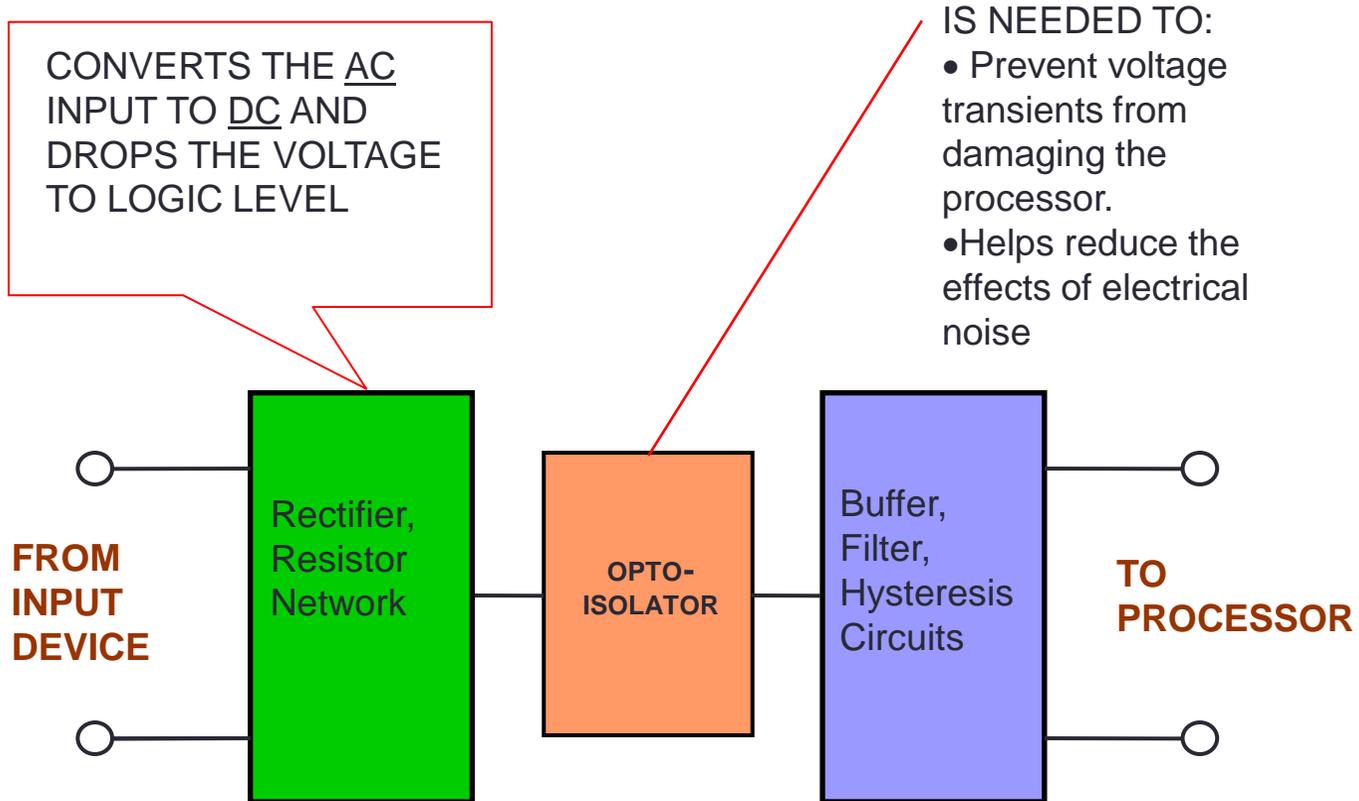
# I/O Module

## DC INPUT MODULE

USE TO DROP THE VOLTAGE TO LOGIC LEVEL
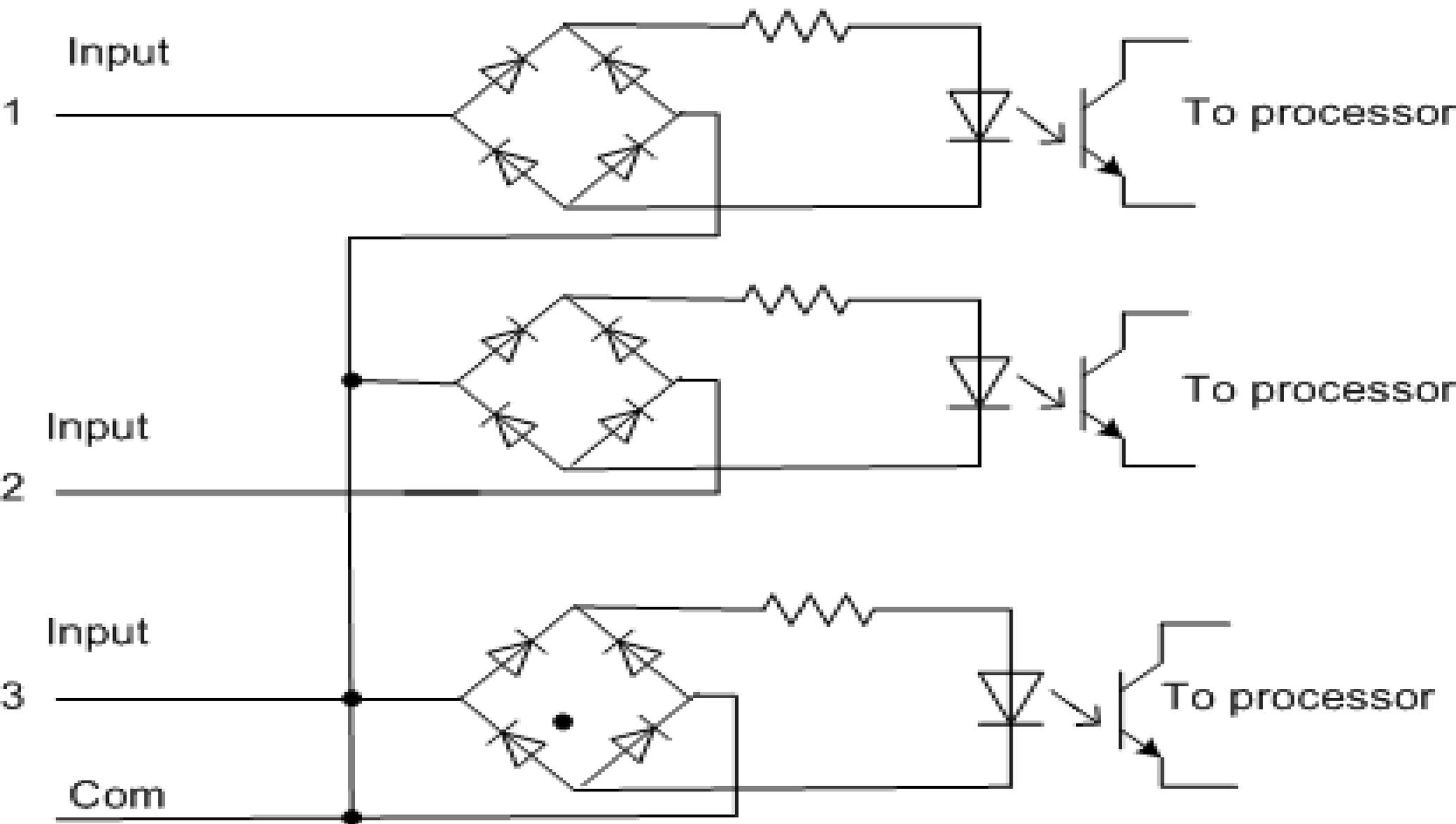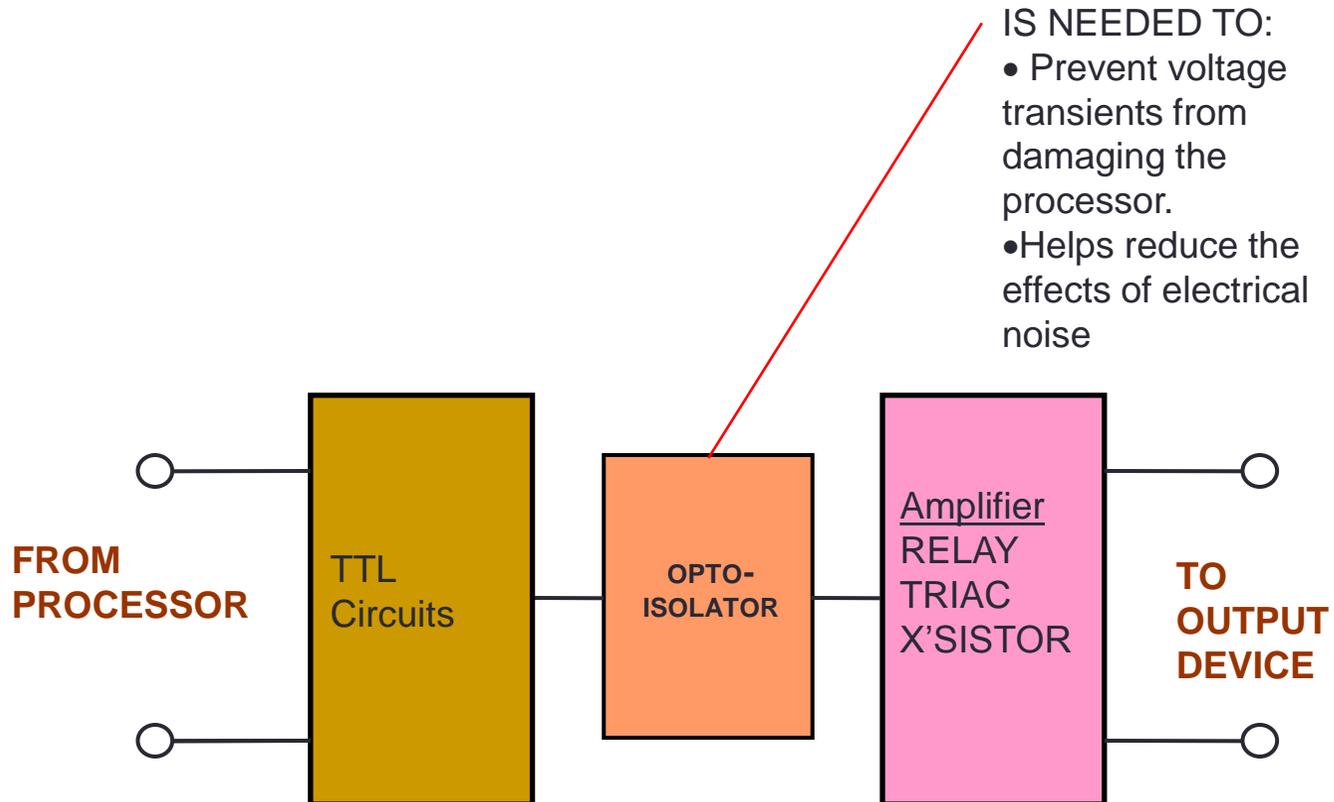
IS NEEDED TO:
• Prevent voltage transients from damaging the processor.
•Helps reduce the effects of electrical noise

**FROM INPUT DEVICE**

Current Limiting Resistor

OPTO-ISOLATOR

Buffer, Filter, hysteresis Circuits

**TO PROCESSOR**

# I/O Module

## AC INPUT MODULE

CONVERTS THE <u>AC</u> INPUT TO <u>DC</u> AND DROPS THE VOLTAGE TO LOGIC LEVEL

IS NEEDED TO:
- Prevent voltage transients from damaging the processor.
- Helps reduce the effects of electrical noise

**FROM INPUT DEVICE**

Rectifier, Resistor Network

**OPTO-ISOLATOR**

Buffer, Filter, Hysteresis Circuits

**TO PROCESSOR**

**Figure 3.1a   3-Input circuit diagram**

# I/O Module

## DC / AC OUTPUT MODULE

IS NEEDED TO:
• Prevent voltage transients from damaging the processor.
• Helps reduce the effects of electrical noise

**FROM PROCESSOR**

TTL Circuits

**OPTO-ISOLATOR**

Amplifier
RELAY
TRIAC
X'SISTOR

**TO OUTPUT DEVICE**

# DIFFERENT TYPES OF I/O CIRCUITS

### 1. General - Purpose Outputs

These are usually low- voltage and low-current and are used to drive indicating lights and other non-inductive loads. Noise suppression may or may not be included on this types of modules.

### 2. Discrete Inputs

Circuits of this type are used to sense the status of limit switches, push buttons, and other discrete sensors. Noise suppression is of great importance in preventing false indication of inputs turning on or off because of noise.

# I/O Circuits

3. Analog I/O

Circuits of this type sense or drive analog signals.

Analog inputs come from devices, such as thermocouples, strain gages, or pressure sensors, that provide a signal voltage or current that is derived from the process variable. Standard Analog Input signals: 4-20mA; 0-10V

Analog outputs can be used to drive devices such as voltmeters, X-Y recorders, servomotor drives, and valves through the use of transducers.
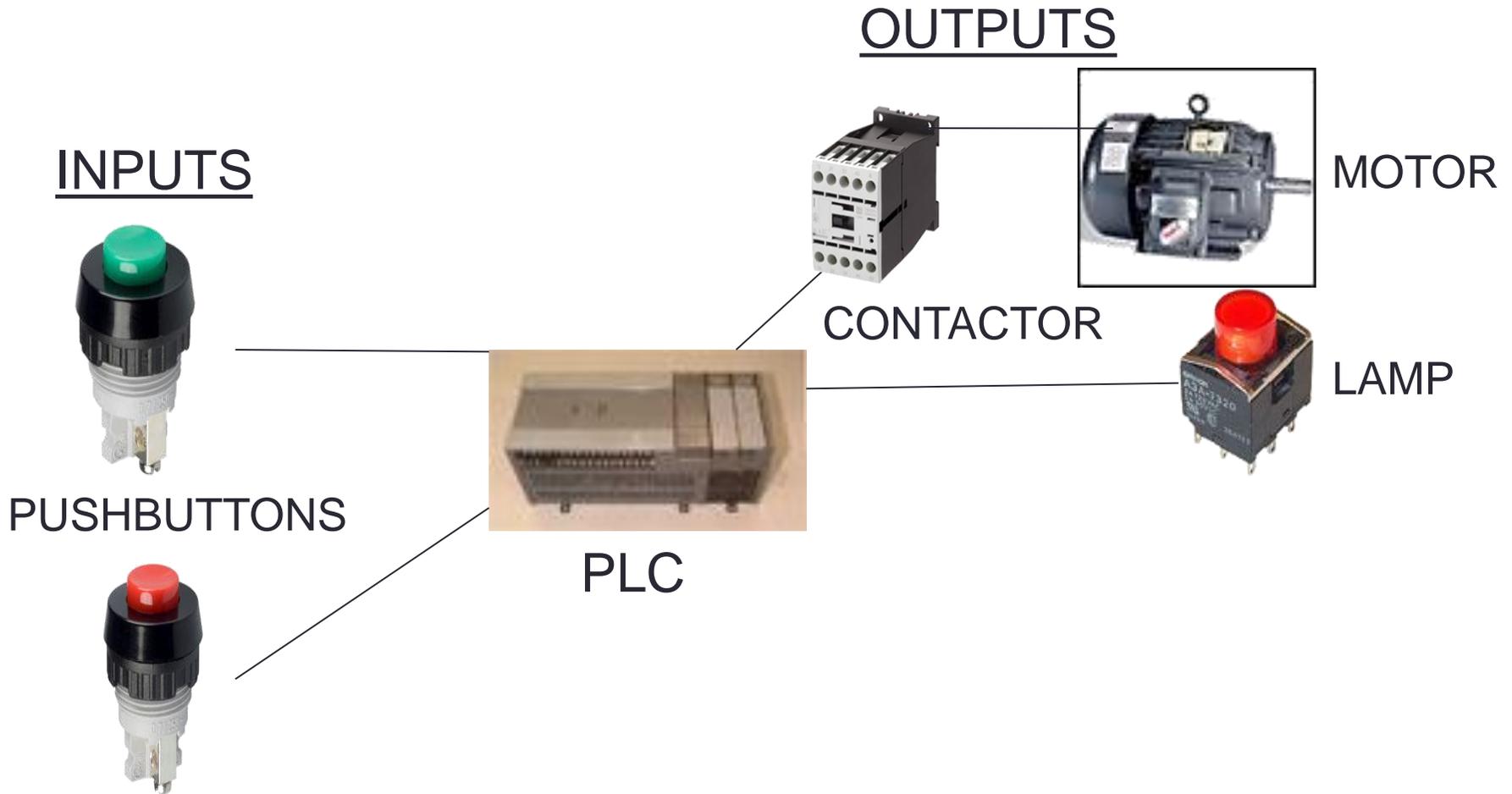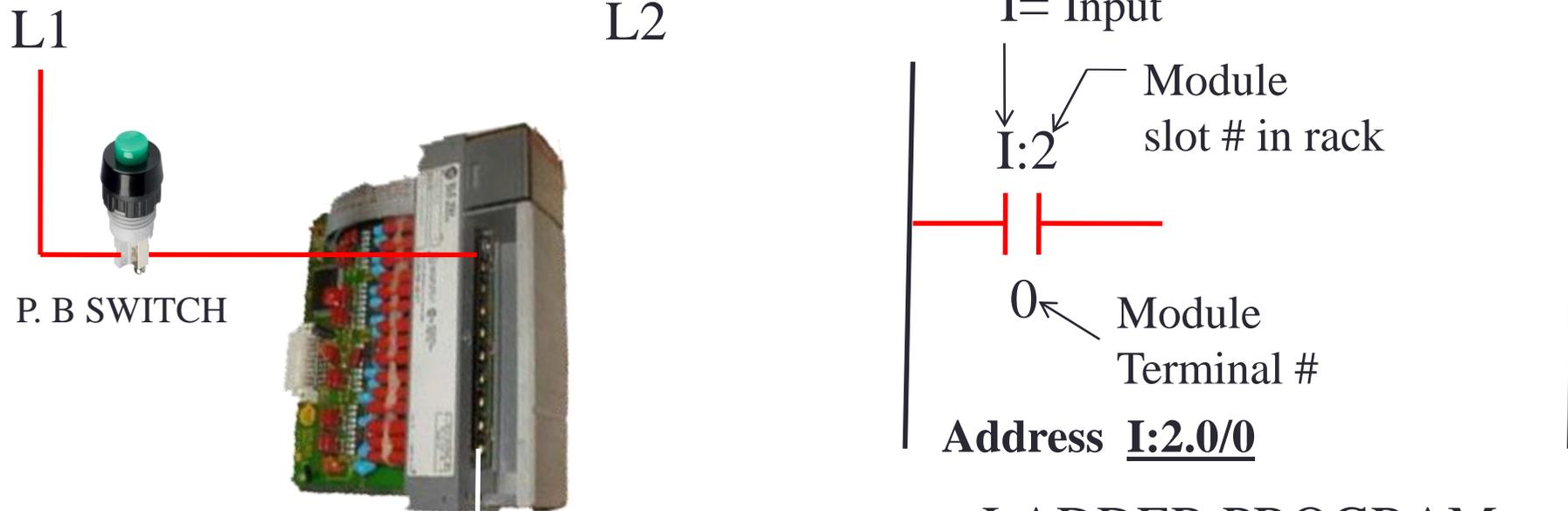Standard Analog Output signals: 4-20mA; 0-5V; 0-10V

# I/O Circuits

4. Special - Purpose I/O

Circuits of this type are used to interface PLCs to very specific types of circuits such as servomotors, stepping motors PID (proportional plus integral plus derivative) loops, high-speed pulse counting, resolver and decoder inputs, multiplexed displays, and keyboards.

This module allows for limited access to timer and counter presets and other PLC variables without requiring a program loader.
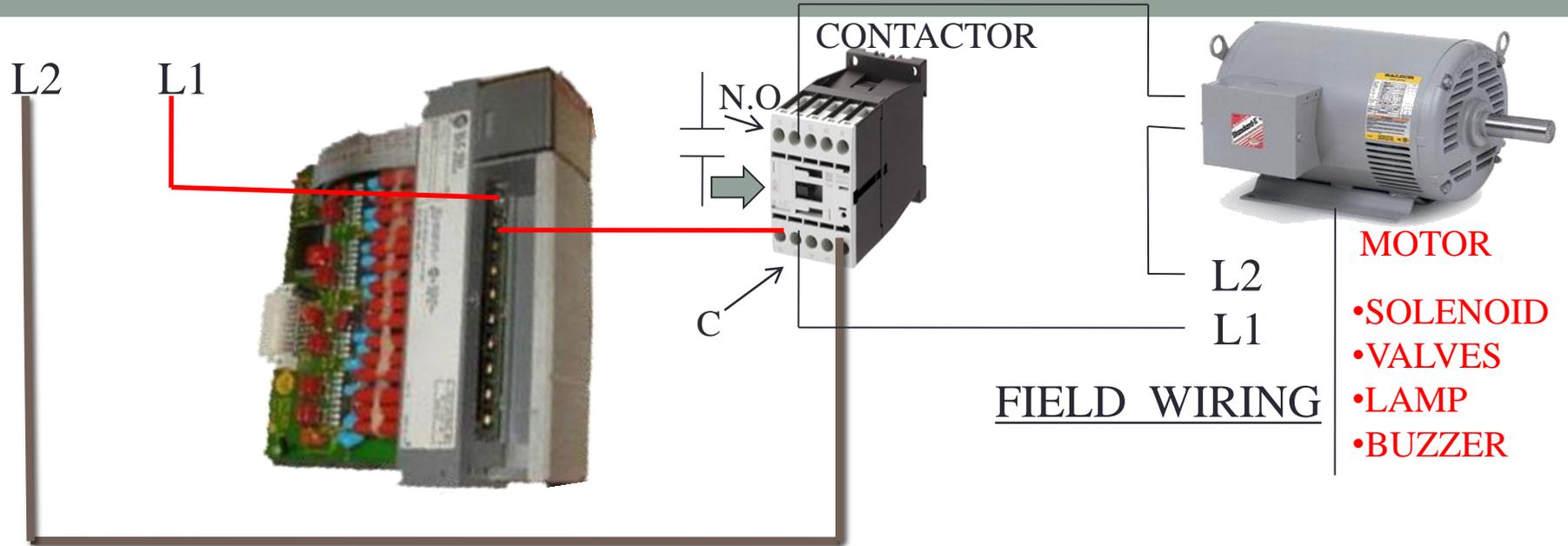
# OUTPUTS

# INPUTS

MOTOR

CONTACTOR
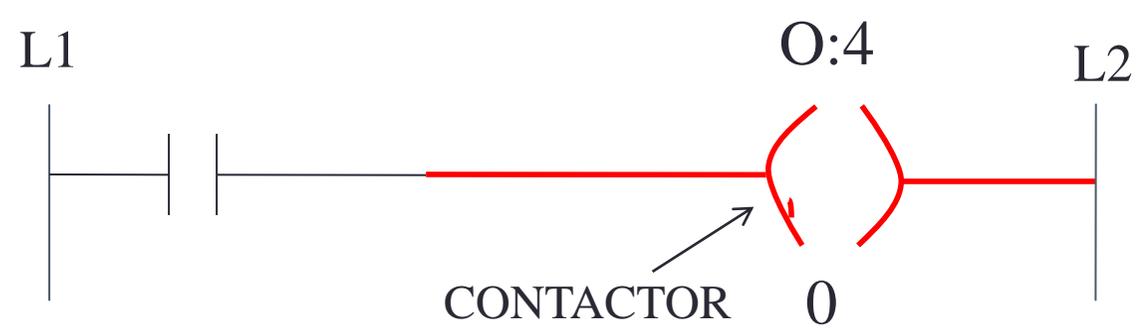
LAMP

PUSHBUTTONS

PLC

# Allen-Bradley 1746-1A16

L1                                          L2

P. B SWITCH

INPUT MODULE

WIRING DIAGRAM

I= Input

Module
slot # in rack

I:2

0    Module
Terminal #

**Address  I:2.0/0**

LADDER PROGRAM

L2    L1

CONTACTOR

N.O

C

L2
L1

MOTOR

•SOLENOID
•VALVES
FIELD  WIRING    •LAMP
•BUZZER

OUTPUT MODULE
WIRING

L1                    O:4              L2

CONTACTOR      0

LADDER PROGRAM

## Discrete Input

A discrete input also referred as digital input is an input that is either ON or OFF are connected to the PLC digital input. In the ON condition it is referred to as logic 1 or a logic high and in the OFF condition maybe referred to as logic o or logic low.
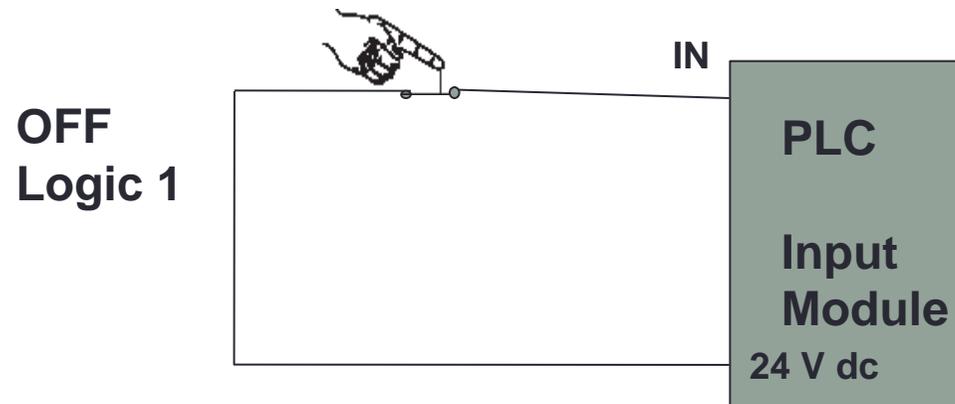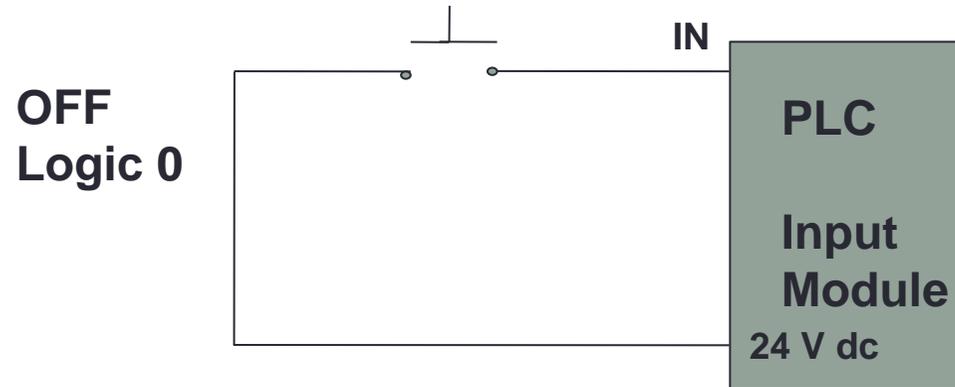
Normally Open Pushbutton

Normally Closed Pushbutton

Normally Open switch

Normally Closed switch

Normally Open contact

Normally closed contact

**OFF**
**Logic 0**

IN

**PLC**

**Input**
**Module**

**24 V dc**

**OFF**
**Logic 1**

IN

**PLC**

**Input**
**Module**

**24 V dc**
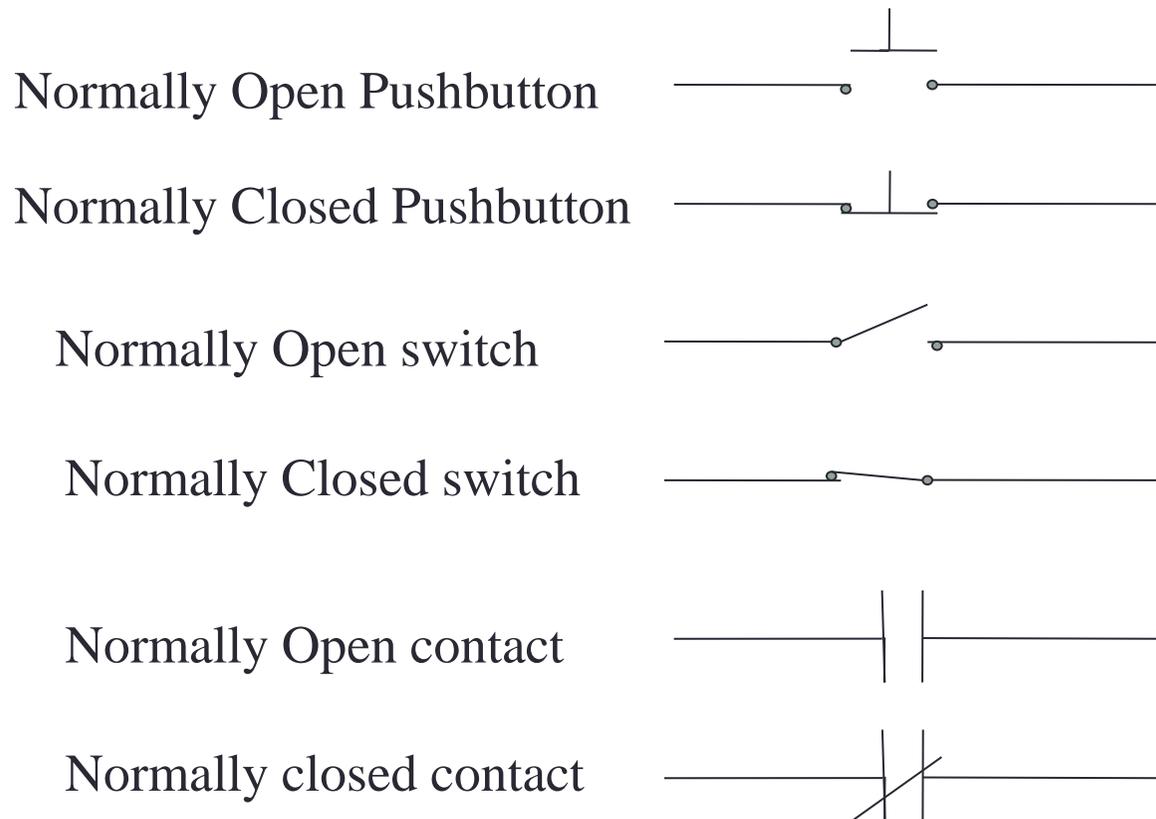
# Analog Input

An analog input is an input signal that has a continuous signal. Typical inputs may vary from 0 to 20mA, 4 to 20mA or 0 to10V. Below, a level transmitter monitors the level of liquid in the tank. Depending on the level Tx, the signal to the PLC can either increase or decrease as the level increases or decreases.

Level Transmitter

IN

Tank

PLC
Analog
Input
Module
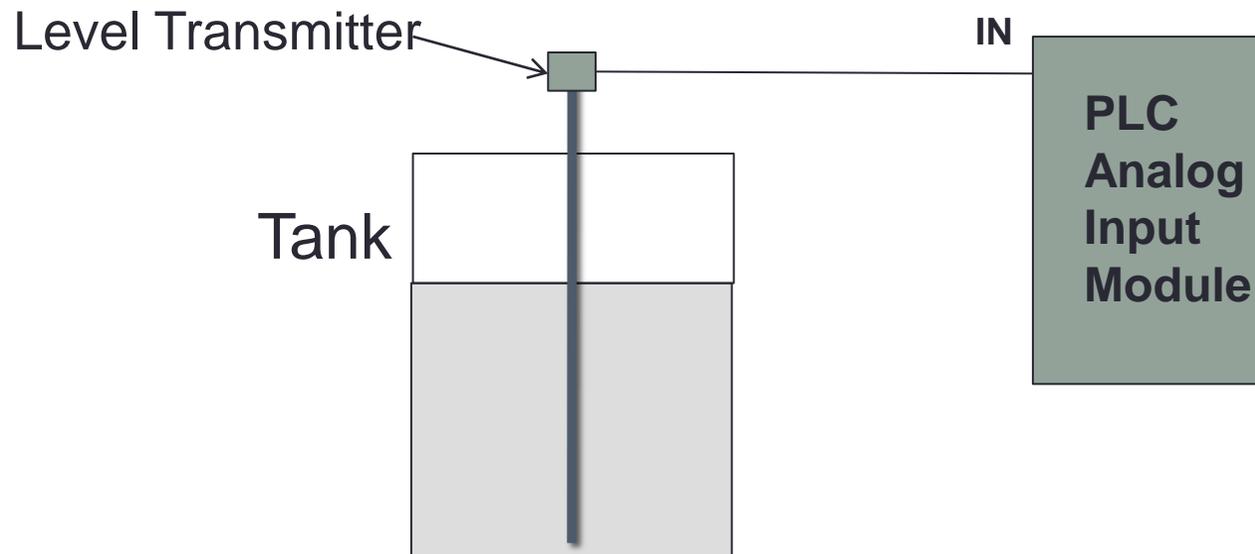
# Digital Output

A discrete output is either in an ON or OFF condition. Solenoids, contactors coils, lamps are example of devices connected to the Discrete or digital outputs. Below, the lamp can be turned ON or OFF by the PLC output it is connected to.

**OUT**

**PLC**

**Digital Output Module**

Lamp
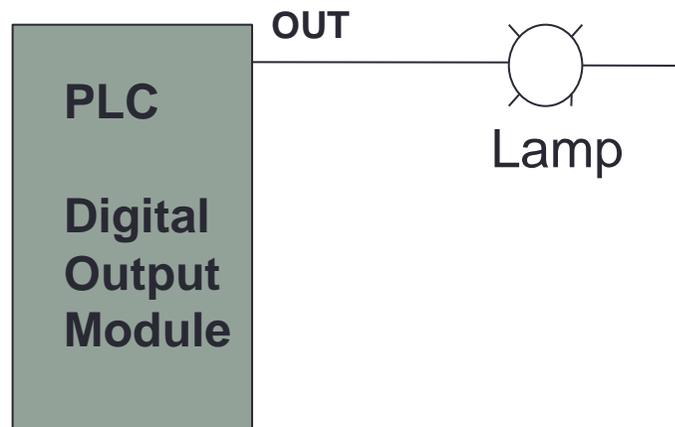
# Analog Output

An analog output is an output signal that has a continuous signal. Typical outputs may vary from 0 to 20mA, 4 to 20mA or 0 to10V.

**Electric to pneumatic transducer**

| PLC | OUT |
| --- | --- |
| | 0 to 10V |
| Analog Output Module | |

E / P

**Supply air**

**Pneumatic control valve**
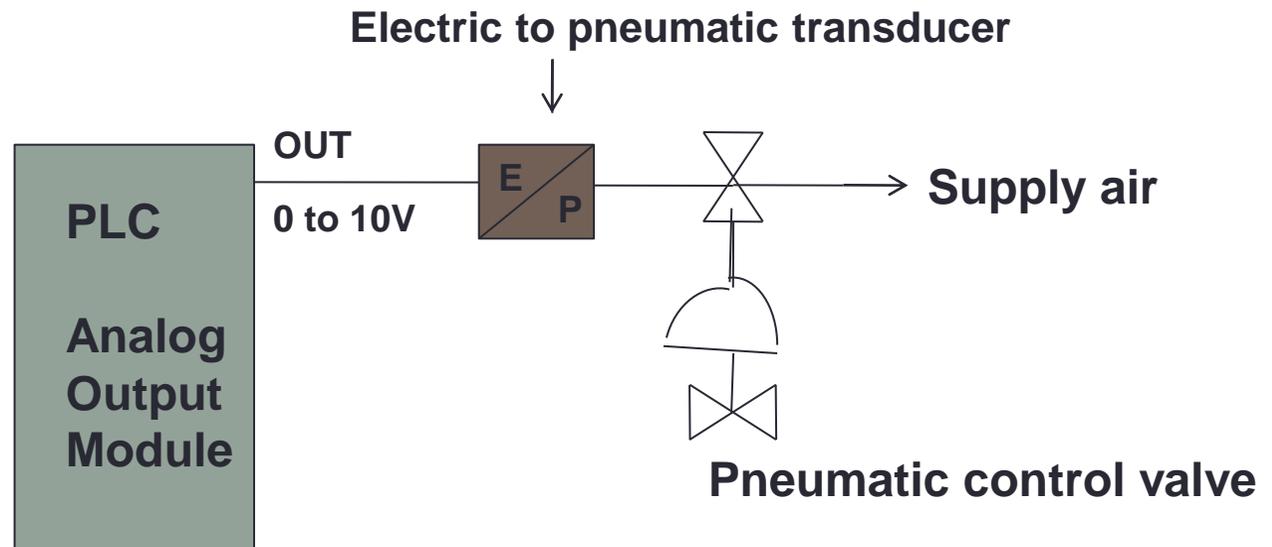
# Processor

The processor module contains the PLC's microprocessor, its supporting circuitry, and its memory system.

The main function of the microprocessor is to analyze data coming from field sensors through input modules, make decisions based on the user's defined control program and return signal back through output modules to the field devices. Field sensors: switches, flow, level, pressure, temp. transmitters, etc. Field output devices: motors, valves, solenoids, lamps, or audible devices.

The memory system in the processor module has two parts: a *system memory* and an *application memory.*

# Memory Map Organization

**SYSTEM**

**APPLICATION**

•Data Table
•User Program
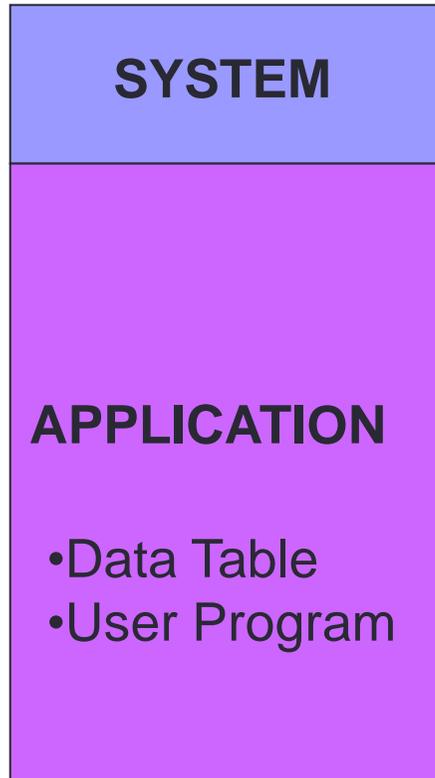
•System memory includes an area called the EXECUTIVE, composed of permanently-stored programs that direct all system activities, such as execution of the users control program, communication with peripheral devices, and other system activities.
•The system memory also contains the routines that implement the PLC's instruction set, which is composed of specific control functions such as logic, sequencing, timing, counting, and arithmetic.
•System memory is generally built from read-only memory devices.

•The application memory is divided into the data table area and user program area.
•The data table stores any data associated with the user's control program, such as system input and output status data, and any stored constants, variables, or preset values. The data table is where data is monitored, manipulated, and changed for control purposes.
•The user program area is where the programmed instructions entered by the user are stored as an application control program.

# Memory Designs

**VOLATILE**.
A volatile memory is one that loses its stored information when power is removed.

Even momentary losses of power will erase any information stored or programmed on a volatile memory chip.

Common Type of Volatile Memory

**RAM. R**andom **A**ccess **M**emory(Read/Write)
Read/write indicates that the information stored in the memory can be retrieved or read, while write indicates that the user can program or write information into the memory.

# Memory Designs

The words **random access**  refer to the ability of any location (address) in the memory to be accessed or used. Ram memory is used for both the user memory (ladder diagrams) and storage memory in many PLC's.

RAM memory must have <u>battery backup</u> to retain or protect the stored program.

# Memory Designs

Several Types of RAM Memory:

1. MOS
2. HMOS
3. CMOS

The CMOS-RAM (**C**omplimentary **M**etal **O**xide **S**emiconductor) is probably one of the most popular. CMOS-RAM is popular because it has a very low current drain when not being accessed (15microamps.), and the information stored in memory can be retained by as little as 2Vdc.

# Memory Designs

**NON-VOLATILE**

Has the ability to retain stored information when power is removed, accidentally or intentionally. These memories do not require battery back-up.

Common Type of  Non-Volatile Memory

**ROM, R**ead **O**nly **M**emory

Read only indicates that the information stored in memory can be read only and cannot be changed. Information in ROM is placed there by the manufacturer for the internal use and operation of the PLC.

# Memory Designs

Other Types of Non-Volatile Memory

**PROM, P**rogrammable **R**ead **O**nly **M**emory
Allows initial and/or additional information to be written into the chip.

PROM may be written into only once after being received from the PLC manufacturer; programming is accomplish by pulses of current.

The current melts the fusible links in the device, preventing it from being reprogrammed. This type of memory is used to prevent unauthorized program changes.

# Memory Designs

**EPROM,** **E**rasable **P**rogrammable **R**ead **O**nly **M**emory

Ideally suited when program storage is to be semi-permanent or additional security is needed to prevent unauthorized program changes.

The EPROM chip has a quartz window over a silicon material that contains the electronic integrated circuits. This window normally is covered by an opaque material, but when the opaque material is removed and the circuitry exposed to ultra violet light, the memory content can be erased.

The EPROM chip is also referred to as **UVPROM.**

# Memory Designs

**EEPROM**, **E**lectrically **E**rasable **P**rogrammable **R**ead **O**nly **M**emory

Also referred to as E$^2$PROM, is a chip that can be programmed using a standard programming device and can be erased by the proper signal being applied to the erase pin.

EEPROM is used primarily as a <u>non-volatile</u> backup for the normal RAM memory. If the program in RAM is lost or erased, a copy of the program stored on an EEPROM chip can be down loaded into the RAM.

# PLC Operation

Basic Function of a Typical PLC

Read all field input devices via the input interfaces, execute the user program stored in application memory, then, based on whatever control scheme has been programmed by the user, turn the field output devices on or off, or perform whatever control is necessary for the process application.

This process of sequentially reading the inputs, executing the program in memory, and updating the outputs is known as scanning.

While the PLC is running, the scanning process includes the following four phases, which are repeated continuously as individual cycles of operation:

PHASE 1
Read Inputs
Scan

PHASE 2
Program
Execution

PHASE 3
Diagnostics/
Comm
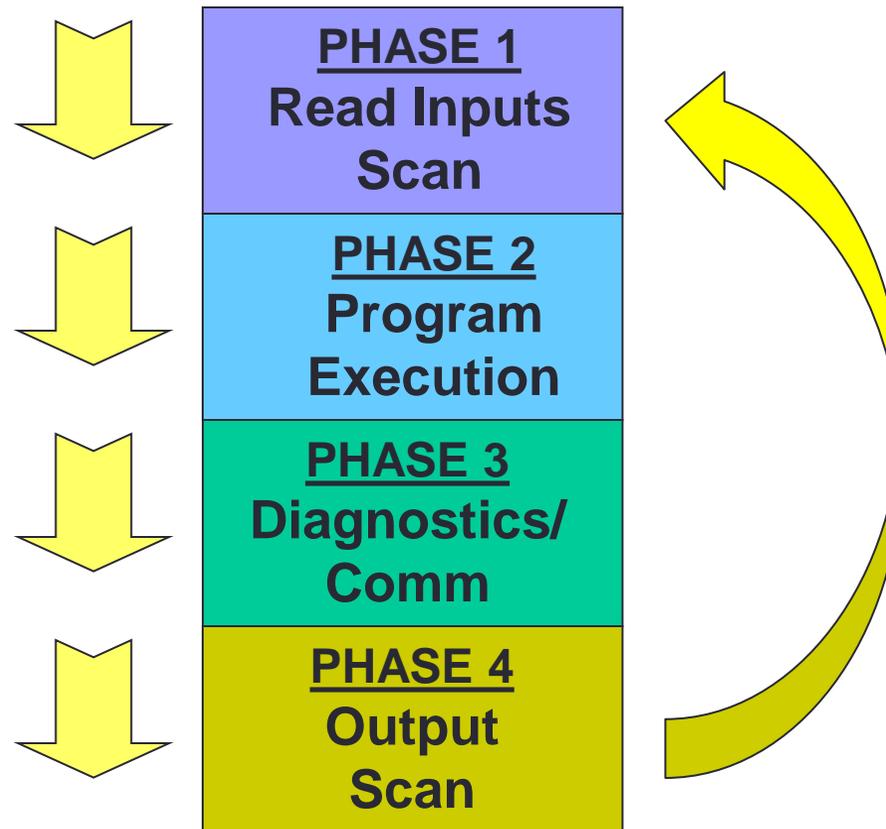
PHASE 4
Output
Scan

# PHASE 1 – Input Status scan

- A PLC scan cycle begins with the CPU reading the status of its inputs.

## PHASE 2– Logic Solve/Program Execution

- The application program is executed using the status of the inputs

## PHASE 3– Logic Solve/Program Execution

- Once the program is executed, the CPU performs diagnostics and communication tasks

## PHASE 4 - Output Status Scan

- An output status scan is then performed, whereby the stored output values are sent to actuators and other field output devices. The cycle ends by updating the outputs.

As soon as Phase 4 are completed, the entire cycle begins again with Phase 1  input scan.

The time it takes to implement a scan cycle is called SCAN TIME. The scan time composed  of the program scan time, which is the time required for solving the control program, and the I/O update time, or time required to read inputs and update outputs. The program scan time generally depends on the amount of memory taken by the control program and type of instructions used in the program. The time to make a single scan can vary from 1 ms to 100 ms.

# Specifications

<u>OUTPUT-PORT POWER RATINGS</u>

Each output port should be capable of supplying sufficient voltage and current to drive the output peripheral connected to it.

<u>SCAN TIME</u>

This is the speed at which the controller executes the relay-ladder logic program. This variable is usually specified as the scan time per 1000 logic nodes and typically ranges from 1 to 200 milliseconds.

# Specifications

MEMORY CAPACITY

The amount of memory required for a particular application is related to the length of the program and the complexity of the control system. Simple applications having just a few relays do not require significant amount of memory. Program length tend to expand after the system have been used for a while. It is advantageous to a acquire a controller that has more memory than is presently needed.

# PLC Status Indicators

- Power On

- Run Mode

- Programming Mode

- Fault

**Troubleshooting**

      1. Look at the process
      2. PLC status lights
          HALT - something has stopped the CPU
          RUN - the PLC thinks it is OK (and probably is)
          ERROR - a physical problem has occurred with the PLC
      3. Indicator lights on I/O cards and sensors
      4. Consult the manuals, or use software if available.
      5. Use programming terminal / laptop.

**List of items required when working with PLCs:**

1. Programming Terminal - laptop or desktop PC.
2. PLC Software. PLC manufacturers have their own specific software and license key.
3. Communication cable for connection from Laptop to PLC.
4. Backup copy of the ladder program (on diskette, CDROM, hard disk, flash memory). If none, upload it from the PLC.
5. Documentation- (PLC manual, Software manual, drawings, ladder program printout, and Seq. of Operations manual.)

**Examples of PLC  Programming Software:**

1. Allen-Bradley – <span style="color:red">Rockwell Software RSLogix500</span>
2. Modicon -   <span style="color:red">Modsoft</span>
3. Omron - <span style="color:red">Syswin</span>
4. GE-Fanuc Series 6 – <span style="color:red">LogicMaster6</span>
5. Square D- <span style="color:red">PowerLogic</span>
6. Texas Instruments – <span style="color:red">Simatic</span>
6. Telemecanique – <span style="color:red">Modicon TSX Micro</span>

# PROGRAMMING



Normally Open
(NO)

Normally Closed
(NC)

Power flows through these contacts when they are closed. The normally open (NO) is true when the input or output status bit controlling the contact is 1. The normally closed (NC) is true when the input or output status bit controlling the contact is 0.

## Coils

—( )—
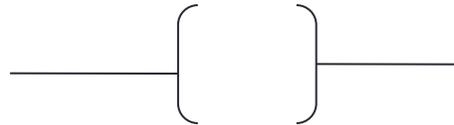
Coils represent relays that are energized when power flows to them. When a coil is energized it causes a corresponding output to turn on by changing the state of the status bit controlling the output to 1. That same output status bit maybe used to control normally open or normally closed contact anywhere in the program.

Boxes
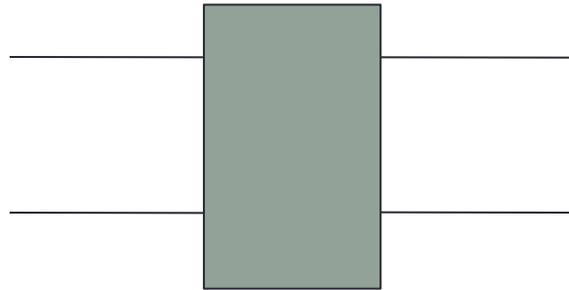


Boxes represent various instructions or functions that are
Executed when power flows to the box. Some of these
Functions are timers, counters and math operations.

# AND OPERATION

A    B    C
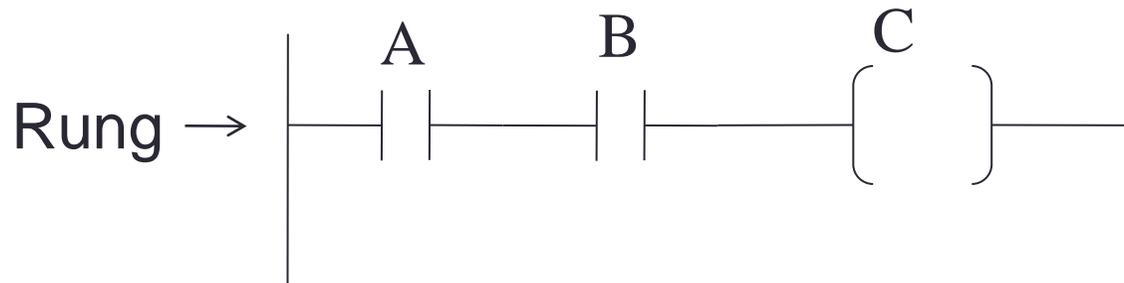
Rung →

Each rung or network on a ladder program represents
a logic operation. In the rung above, both inputs A and B
must be true (1) in order  for the output C to be true (1).

# OR OPERATION
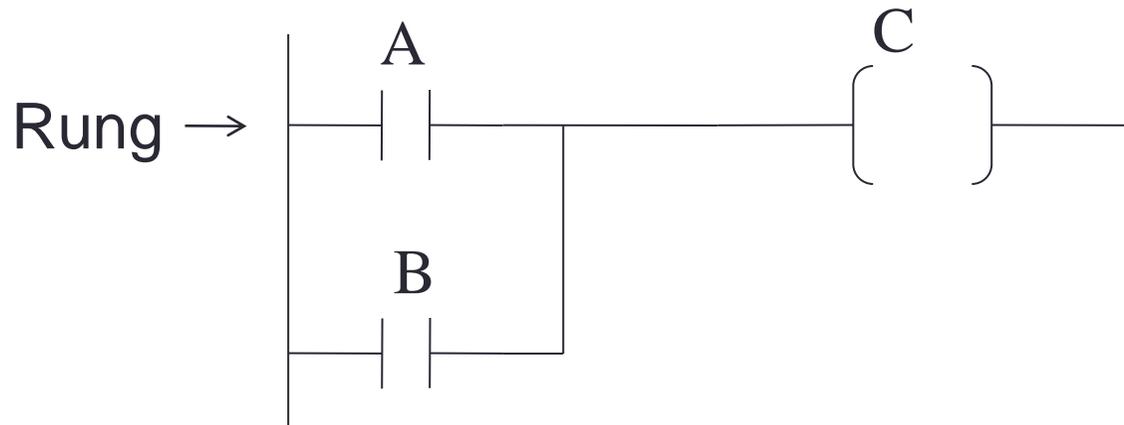
Rung →



In the rung above, it can be seen that either input A or B is be true (1), or both are true, then the output C is true (1).

# NOT OPERATION

Rung →

A     C
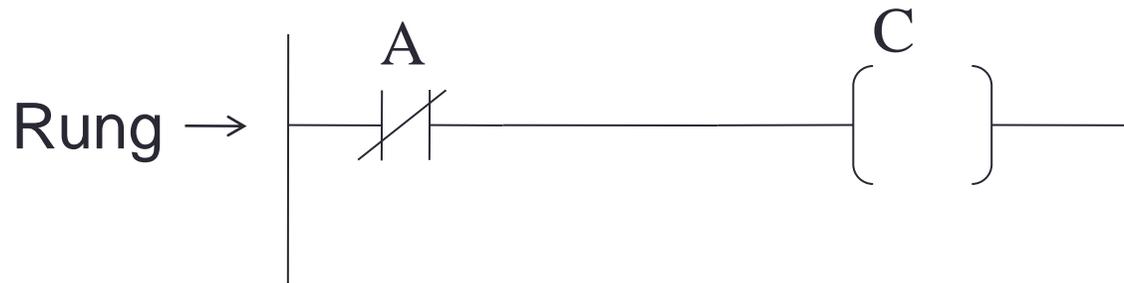
In the rung above, it can be seen that if input A is be true (1), then the output C is true (0) or when A is (0), output C is 1.

# Microcontroller Intel 8051

## [Architecture]

# Pin Description of the 8051



P1.0 — 1          40 — Vcc
P1.1 — 2          39 — P0.0(AD0)
P1.2 — 3          38 — P0.1(AD1)
P1.3 — 4          37 — P0.2(AD2)
P1.4 — 5          36 — P0.3(AD3)
P1.5 — 6          35 — P0.4(AD4)
P1.6 — 7          34 — P0.5(AD5)
P1.7 — 8          33 — P0.6(AD6)
RST — 9           32 — P0.7(AD7)
(RXD)P3.0 — 10    31 — EA/VPP
(TXD)P3.1 — 11    30 — ALE/PROG
(INT0)P3.2 — 12   29 — PSEN
(INT1)P3.3 — 13   28 — P2.7(A15)
(T0)P3.4 — 14     27 — P2.6(A14)
(T1)P3.5 — 15     26 — P2.5(A13)
(WR)P3.6 — 16     25 — P2.4(A12)
(RD)P3.7 — 17     24 — P2.3(A11)
XTAL2 — 18        23 — P2.2(A10)
XTAL1 — 19        22 — P2.1(A9)
GND — 20          21 — P2.0(A8)

8051
(8031)
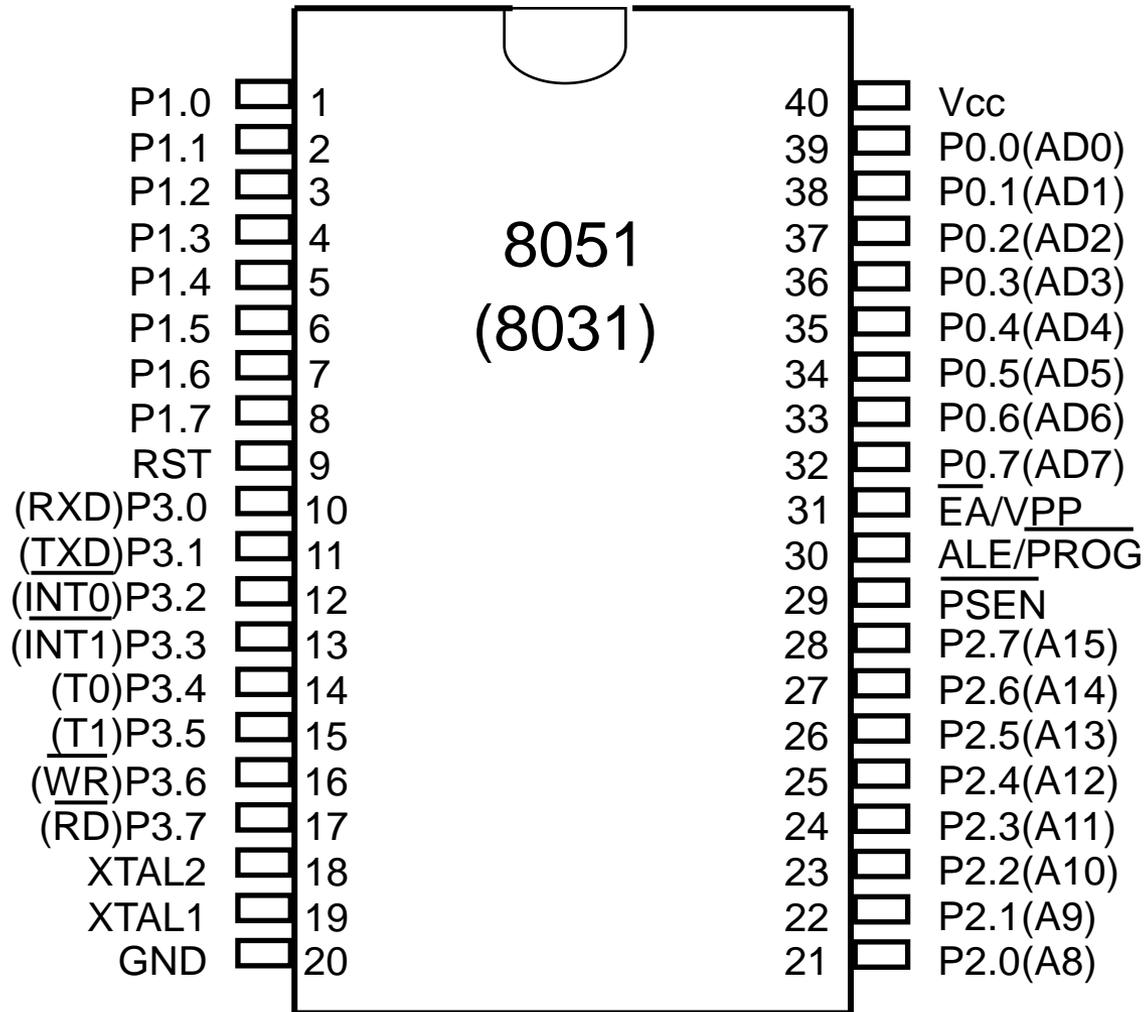
# The Microcontroller

- Microcontrollers can be considered as self-contained systems with a processor, memory and I/O ports.
  - In most cases, all that is missing is the software to define the operation of the embedded system.

- Usually available in several forms:
  - Devices for prototyping
    - Built-in or piggy-back EPROM for storing the software.
  - One Time Programmable (OTP) devices.
    - No window on the package. Therefore, the internal EPROM cannot be erased after being programmed.
  - High Volume Production devices.
    - Use ROM internally to hold the software. Cheaper in large volume.

# The Architectural Needs of a Microcontroller

- Lets consider what architectural features would be needed in a microcontroller.

- What are the expected applications?
  - Sensing the environment (Input)
  - Producing a response (Output)
  - The response may be delayed (Timer/Counter)
  - Prioritized response (Interrupts)
  - Software to control the process (Non-volatile Memory)
  - Temporary data (RAM)

# Major Microcontroller Families

- There are several major "*families*" of microcontrollers available from different manufacturers.

  - A family is defined as a group of products that share the same basic internal architectures and who are code-compatible.

  - Manufacturers usually define an architecture and then make variants of that design producing a family of products.

  - Code written for a member of the family *should* be compatible with all other members of the same family.

# Example Microcontroller Families

- Z80 from Zilog
  - 8-bit microprocessor based on the 8080 architecture.
  - Capable of 1 MIP at 4 MHz.
  - Accumulator, 6 8-bit registers, 2 index registers.
  - Uses external RAM for temporary data.
  - Built-in refresh circuitry for the external RAM.
  - Only port-based I/O.
  - Slowly disappearing.

# Example Microcontroller Families …Cont.

- MC6811 from Motorola
    - 8-bit stack-based architecture.
    - 2 accumulators and 2 index registers.
    - Built in EEPROM and RAM.
    - Digital I/O.
    - Timers.
    - ADC.
    - RS232 communication.
    - *Was* the most powerful and flexible controller available at introduction.
        - Around 1970.
    - *Still* very popular.

# The MCS-51 Family of Microcontrollers

- Originally introduced by Intel in 1981.
  - *Currently*, the most widely used microcontroller.
  - 8-bit processor.
  - 2 distinct separately addressable memory areas.
    - Maximum of 64K on-chip ROM.
      - Usually 0 to 4K.
    - Maximum of 64K external data memory.
    - Maximum of 64K external code memory.
    - Basic version (8051) contains:
      - 4K Bytes of on-chip ROM instruction memory.
      - 128 Bytes of on-chip RAM for temporary data storage and the stack.
  - 2 timers, one serial port, and four 8-bit parallel I/O ports.
  - Speeds starting from12 MHz.

# Features of the 8051 Microcontroller

| Feature | 8051 | 8052 | 8031 |
|---|---|---|---|
| ROM (on-chip program space in bytes) | 4K | 8K | 0K |
| RAM (bytes) | 128 | 256 | 128 |
| Timers | 2 | 3 | 2 |
| I/O pins | 32 | 32 | 32 |
| Serial port | 1 | 1 | 1 |
| Interrupt sources | 6 | 8 | 6 |

- The 8031 requires external instruction memory.
  - It can be as large as 64K Bytes.
  - You lose 2 ports for interfacing to the external memory.
    - You can replace these by interfacing the chip to an I/O port controller like the 8255.

- The 8051 is the original member of the Intel MCS-51 family of Microcontrollers.
  - There are several varieties that differ slightly in the available features.

# The 8051 Microcontroller Internals
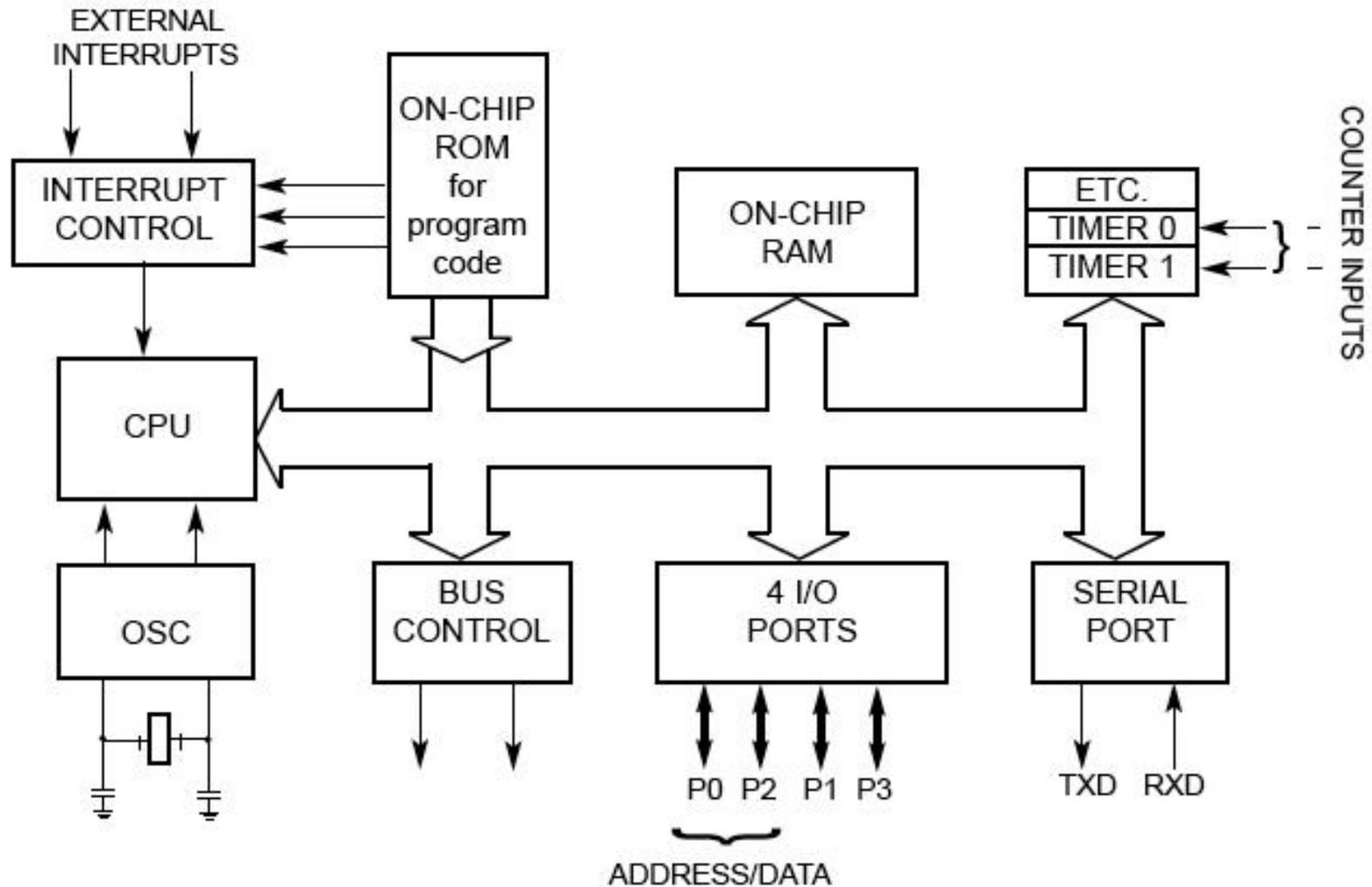
# 8051 Architecture

- Programmer's View
  - Memory Organization
  - Register Set
  - Instruction Set

- Hardware Designer's View
  - Pin-out
  - Timing characteristics
  - Current / Voltage requirements

# Memory Organization

The 8051 has separate address spaces for program storage and data storage.

- Depending on the type of instruction, the same address can refer to two logically and physically different memory locations.

# Program Storage

- After reset, the MCS-51 starts fetching instructions from 0000H.

  - This can be either on-chip or external depending on the value of the $\overline{EA}$ input pin.

    - If EA* is low, then the program memory is external.

    - If EA* is high, then addresses from 0000 to 0FFF will refer to on-chip memory and addresses 1000 up to FFFF refer to external memory.

  - Note that the 8031 must have its $\overline{EA}$ connected low as all of its memory is external.

# Access to External Memory

- Port 0 acts as a multiplexed address/data bus. Sending the low byte of the program counter (PCL) as an address.

- Port 2 sends the program counter high byte (PCH) directly to the external memory.

- The signal ALE operates as in the 8051 to allow an external latch to store the PCL byte while the multiplexed bus is made ready to receive the code byte from the external memory.

- Port 0 then switches function and becomes the data bus receiving the byte from memory.

# Data Storage

- The 8051 has 256 bytes of RAM on-chip.
  - The lower 128 bytes are intended for internal data storage.
  - The upper 128 bytes are the Special Function Registers (SFR).

- The lower 128 bytes are not to be used as standard RAM.
  - Internally 8051's registers default to stack area, and other features. [00-7FH]

# Data Storage [Cont...]

- The lowest 32 bytes of the on-chip RAM form 4 banks of 8 registers each.

- Only one of these banks can be active at any time.

- Bank is chosen by setting 2 bits in PSW
  - Default bank (at power up) is bank 0 (locations 00 – 07).

- The 8 registers in any active bank are referred to as R0 through R7

- Given that each register has a specific address, it can be accessed directly using that address even if its bank is not the active one.

# Data Storage [Cont...]

- The next 16 bytes – locations 20H to 2FH – form a block that can be addressed as either bytes or individual bits.
  - The bytes have addresses 20H to 2FH.
  - The bits have addresses 00H to 7FH.
  - Specific instructions are used for accessing the bits.
- Locations 30H to 7FH are general purpose RAM.

# The SFR (Special Function Register)

- The upper 128 bytes of the on-chip RAM are used to house special function registers.
    - In reality, only about 25 of these bytes are actually used. The others are reserved for future versions of the 8051.

    – These are registers associated with important functions in the operation of the MCS-51.

    – Some of these registers are *bit-addressable* as well as *byte-addressable*.
        - The address of bit 0 of the register will be the same as the address of the register.

# The Elements of SFR

- ACC and B registers – 8 bit each
- DPTR : [DPH:DPL] – 16 bit combined
- PC (Program Counter) – 16 bits
- SP (Stack Pointer) – 8 bit
- PSW (Program Status Word)
- Port Latches
- Serial Data Buffer
- Timer Registers
- Control Registers

| Symbol | Name | Address |
|--------|------|---------|
| ACC* | Accumulator | 0E0H |
| B* | B register | 0F0H |
| PSW* | Program status word | 0D0H |
| SP | Stack pointer | 81H |
| DPTR | Data pointer 2 bytes | |
| DPL | Low byte | 82H |
| DPH | High byte | 83H |
| P0* | Port 0 | 80H |
| P1* | Port 1 | 90H |
| P2* | Port 2 | 0A0H |
| P3* | Port 3 | 0B0H |
| IP* | Interrupt priority control | 0B8H |
| IE* | Interrupt enable control | 0A8H |
| TMOD | Timer/counter mode control | 89H |
| TCON* | Timer/counter control | 88H |
| T2CON* | Timer/counter 2 control | 0C8H |
| T2MOD | Timer/counter mode control | 0C9H |
| TH0 | Timer/counter 0 high byte | 8CH |
| TL0 | Timer/counter 0 low byte | 8AH |
| TH1 | Timer/counter 1 high byte | 8DH |
| TL1 | Timer/counter 1 low byte | 8BH |
| TH2 | Timer/counter 2 high byte | 0CDH |
| TL2 | Timer/counter 2 low byte | 0CCH |
| RCAP2H | T/C 2 capture register high byte | 0CBH |
| RCAP2L | T/C 2 capture register low byte | 0CAH |
| SCON* | Serial control | 98H |
| SBUF | Serial data buffer | 99H |
| PCON | Power control | 87H |

* Bit-addressable

FFFF

FF

00

0000

0000

FFFF

Code
memory

enabled
via $\overline{\text{PSEN}}$

Data
memory

enabled
via $\overline{\text{RD}}$
and $\overline{\text{WR}}$

On-chip
memory

External
memory

## RAM
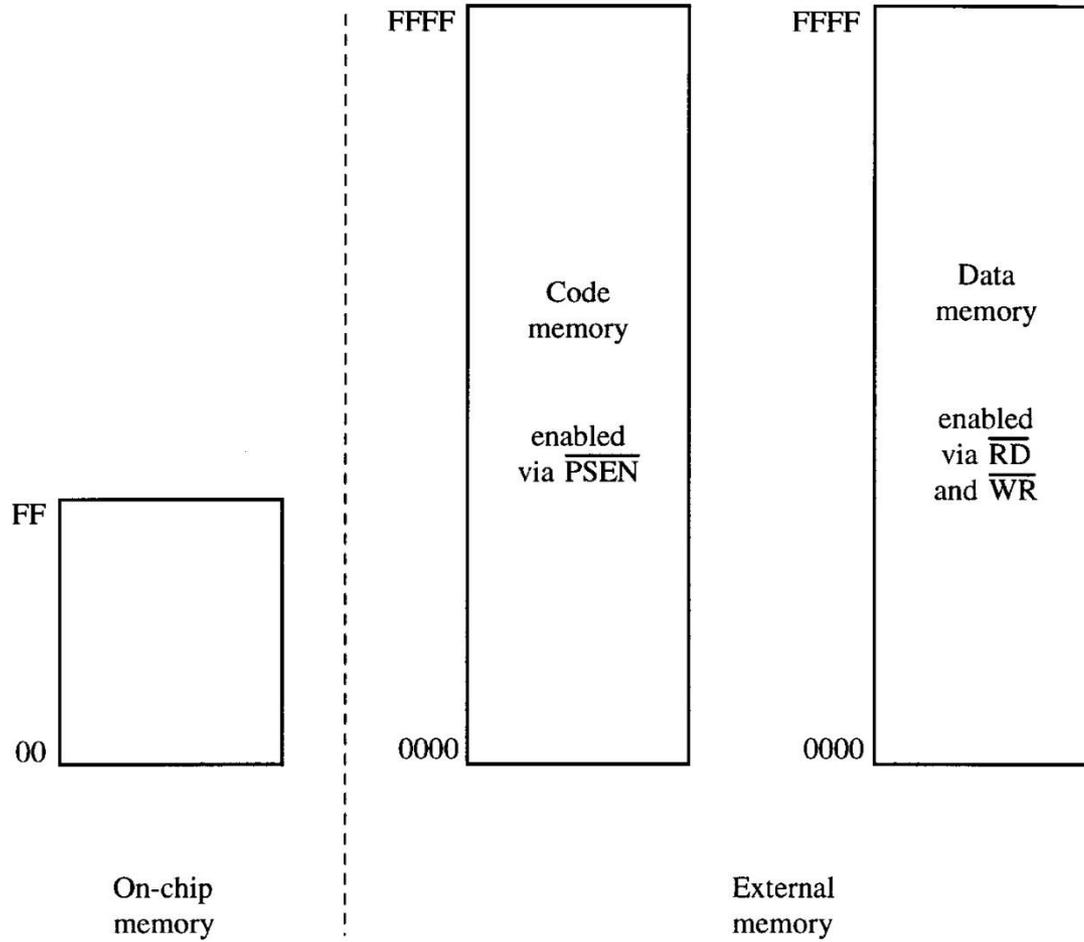
| Byte address | | | Bit address | | | | | |
|---|---|---|---|---|---|---|---|---|
| 7F | General purpose RAM | | | | | | | |
| 30 | | | | | | | | |
| 2F | 7F | 7E | 7D | 7C | 7B | 7A | 79 | 78 |
| 2E | 77 | 76 | 75 | 74 | 73 | 72 | 71 | 70 |
| 2D | 6F | 6E | 6D | 6C | 6B | 6A | 69 | 68 |
| 2C | 67 | 66 | 65 | 64 | 63 | 62 | 61 | 60 |
| 2B | 5F | 5E | 5D | 5C | 5B | 5A | 59 | 58 |
| 2A | 57 | 56 | 55 | 54 | 53 | 52 | 51 | 50 |
| 29 | 4F | 4E | 4D | 4C | 4B | 4A | 49 | 48 |
| 28 | 47 | 46 | 45 | 44 | 43 | 42 | 41 | 40 |
| 27 | 3F | 3E | 3D | 3C | 3B | 3A | 39 | 38 |
| 26 | 37 | 36 | 35 | 34 | 33 | 32 | 31 | 30 |
| 25 | 2F | 2E | 2D | 2C | 2B | 2A | 29 | 28 |
| 24 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 |
| 23 | 1F | 1E | 1D | 1C | 1B | 1A | 19 | 18 |
| 22 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 |
| 21 | 0F | 0E | 0D | 0C | 0B | 0A | 09 | 08 |
| 20 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
| 1F – 18 | Bank 3 | | | | | | | |
| 17 – 10 | Bank 2 | | | | | | | |
| 0F – 08 | Bank 1 | | | | | | | |
| 07 – 00 | Default register bank for R0-R7 | | | | | | | |

*Bit addressable locations* (bracket spanning 2F–20)

**RAM**

## SPECIAL FUNCTION REGISTERS

| Byte address | | | Bit address | | | | | | Register |
|---|---|---|---|---|---|---|---|---|---|
| FF | | | | | | | | | |
| F0 | F7 | F6 | F5 | F4 | F3 | F2 | F1 | F0 | B |
| E0 | E7 | E6 | E5 | E4 | E3 | E2 | E1 | E0 | ACC |
| D0 | D7 | D6 | D5 | D4 | D3 | D2 | - | D0 | PSW |
| B8 | - | - | - | BC | BB | BA | B9 | B8 | IP |
| B0 | B7 | B6 | B5 | B4 | B3 | B2 | B1 | B0 | P3 |
| A8 | AF | - | - | AC | AB | AA | A9 | A8 | IE |
| A0 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 | P2 |
| 99 | not bit addressable | | | | | | | | SBUF |
| 98 | 9F | 9E | 9D | 9C | 9B | 9A | 99 | 98 | SCON |
| 90 | 97 | 96 | 95 | 94 | 93 | 92 | 91 | 90 | P1 |
| 8D | not bit addressable | | | | | | | | TH1 |
| 8C | not bit addressable | | | | | | | | TH0 |
| 8B | not bit addressable | | | | | | | | TL1 |
| 8A | not bit addressable | | | | | | | | TL0 |
| 89 | not bit addressable | | | | | | | | TMOD |
| 88 | 8F | 8E | 8D | 8C | 8B | 8A | 89 | 88 | TCON |
| 87 | not bit addressable | | | | | | | | PCON |
| 83 | not bit addressable | | | | | | | | DPH |
| 82 | not bit addressable | | | | | | | | DPL |
| 81 | not bit addressable | | | | | | | | SP |
| 80 | 87 | 86 | 85 | 84 | 83 | 82 | 81 | 80 | P0 |

**SPECIAL FUNCTION REGISTERS**

# The ACC – Accumulator

- Commonly used for move and arithmetic instructions.
- Can be referred to in several ways:
  - Implicitly in op-codes.
  - Referred to as ACC (or A) for instructions that allow specifying a register.
  - By its SFR address 0E0H.

- Operates in a similar manner to the 8085 accumulator.

- Bit addressable.
  - ACC.2 means bit 2 of the ACC register.

# The B Register

- Commonly used as a temporary register, much like a 9$^{th}$ R register.
- Used by two op-codes
  - MUL AB, div AB
- B register holds the second operand and will hold part of the result
  - Upper 8 bits of the multiplication result
  - Remainder in case of division.

- Can also be accessed through its SFR address of 0F0H.
- Bit addressable.

# The DPL and DPH Registers

- 2 8-bit registers that can be combined into a 16-bit DPTR – Data Pointer.

- Used by commands that access external memory

- Also used for storing 16bit values

  mov DPTR, #data16

  ; setup DPTR with 16bit ext address

  movx A, @DPTR

  ; copy mem[DPTR] to A

- Can be accessed as 2 separate 8-bit registers if needed.

- DPTR is useful for string operations and Look-Up-Table (LUT) operations.

# The SP Register

- SP is the stack pointer.
- SP points to the last used location of the stack.
  - Push operation will first increment SP and then copy data.
  - Pop operation will first copy data and then decrement SP.
- In 8051, stack grows upwards (from low memory to high memory) and can be in the internal RAM only.
- On power-up, SP points to 07H.
  - Register banks 2,3,4 (08H to 1FH) form the default stack area.

- Stack can be relocated by setting SP to the upper memory area in 30H to 7FH.
  - mov SP, #32H

# The PSW Register

- Program Status Word is a "bit addressable" 8-bit register that has all the flags.

MSB                                                                    LSB

| CY | AC | F0 | RS1 | RS2 | OV | - | P |

| Symbol | Position | Function |
|--------|----------|----------|
| CY | PSW.7 | Carry Flag |
| AC | PSW.6 | Auxiliary Carry Flag. For BCD Operations |
| F0 | PSW.5 | Flag 0. Available to the user for general purposes. |
| RS1 | PSW.4 | Register bank select bits. Set by software to determine which register bank is being used. |
| RS2 | PSW.3 | |
| OV | PSW.2 | Overflow Flag |
| - | PSW.1 | Not used |
| P | PSW.0 | Parity Flag. Even Parity. |

# The P0, P1, P2, and P3 Registers

- Port Latches.
- Specify the value to be output on the specific output port or the value read from the specific input port.

- Bit addressable.
  - First bit has the same address as the register.
  - Example: P1 has address 90H in the SFR, so
    - P1.7 or address 97H refer to the same bit.

# The SBUF Register

- Serial Port Data Buffer.
- 2 registers at the same location
  - One is read-only used for reading serial input data.
    - Serial Data Receive Buffer.
  - The other is write-only used for storing serial output data.
    - Serial Data Transmit Buffer.

# Timer Registers – TH0 and TL0

- The high and low bytes of the 16-bit counting register for timer/counter T0.

- There is also a TH1 / TL1 pair for the T1 timer.

- In the 8052, one more pair exists (TH2) / (TL2) for the T2 timer.

- (RCAP2H) and (RCAP2L) exist only in the 8052 and they are copies of the TH2 and TL2 registers.

# Control Registers

- IP – Interrupt Priority.
- IE – Interrupt Enable.
- TMOD – Timer Mode.
- TCON – Timer Control.
- T2CON – Timer 2 Control (8052)
- SCON – Serial Port Control.
- PCON – Power Control (80C51).