A Survey of Programmable logic Controller Systems

PLC Architectures

There are hundreds of PLC models marketed by scores of different companies, but all have at least three elements in common. The heart of the PLC consists of the CPU with associated memory, the input and output modules providing the necessary interface between the PLC and the machine or system to be controlled, and the programming unit which is required to load the program.

The following sections discussing the various PLC elements - CPU, memory, I/O modules and the programming unit in detail. The operational section of a PLC CPU is show in Figure below.

The Microprocessor

The heart of the microprocessor (or processor) of the PLC is the integrated circuit chip. The microprocessor chips used in PC's are exactly the same as those used in computers generally the Z80, 8080, 8086, 6800, 9900, 80286, 80386 or 80486 family. The microprocessor forms the intelligence of the programmable controller which controls all activities of the PLC.

The microprocessor is controlled by a system program know as the executive program. It is also controlled by application program or user program. The microprocessor can be a factor considered for designing a powerful PLC. Microprocessors are classified as to how powerful they are. Two factors are the bit size and the clock speed which determine how quickly a microprocessor executes instructions. The larger the bit size and the faster the clock speed, the more powerful will be the resulting PLC.

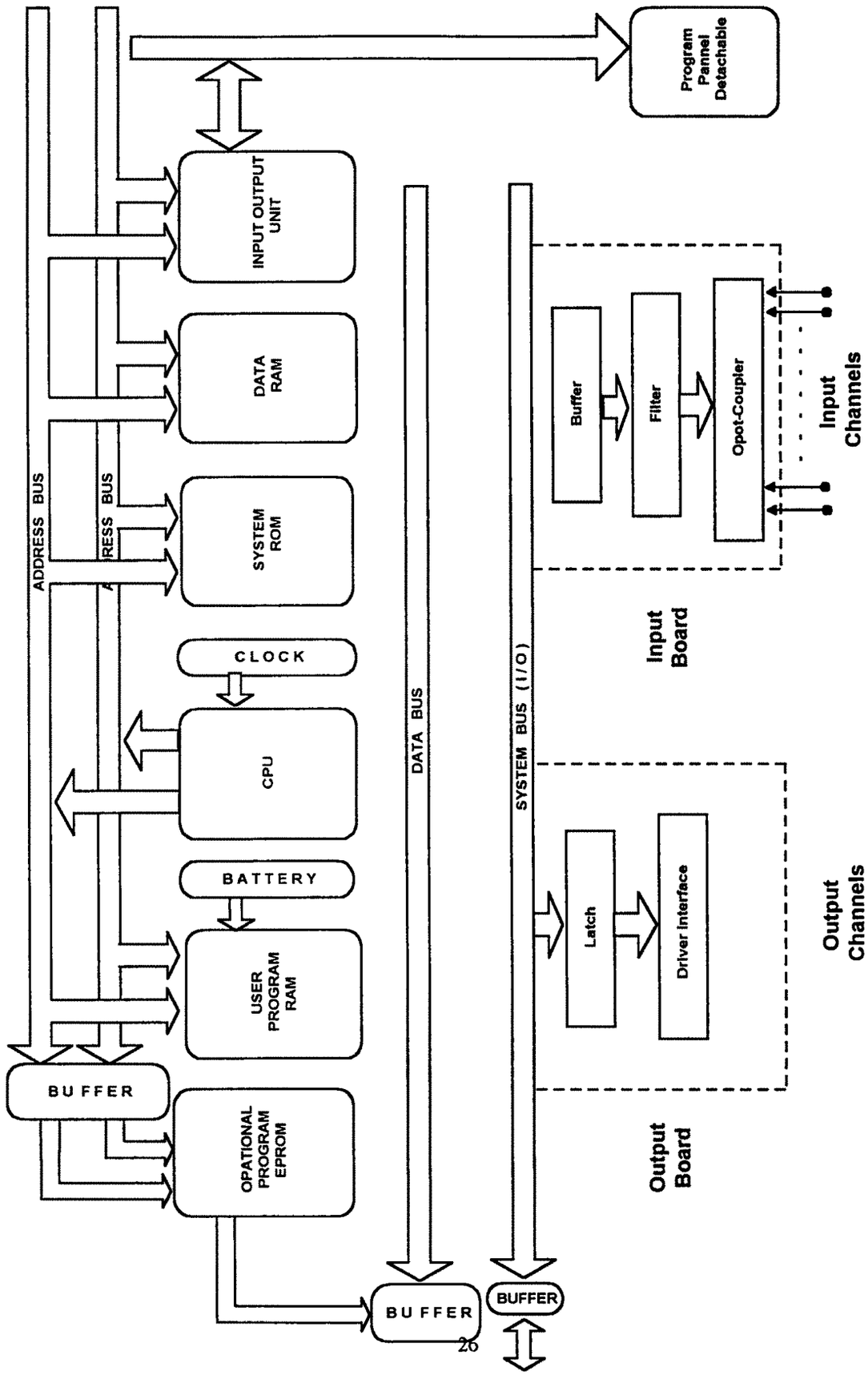

Figure 2 Hardware structure of PLC CPU

The microprocessor is responsible for accepting inputs from the input/output modules, manipulating the data from the inputs and updating the outputs. This process is known as performing a scan. PLCs are also categorized by scan time. Typically the scan time ranges between 1 to 100 milliseconds(ms).

The microprocessor also includes diagnostic programs. These programs detect failures in communication system operation etc.[Ref. (1 - pp. 31-34), (19 - pp. 209-246), (2 - pp. 57-58), (3 - pp. 33-36)]

The Memory

Programmable controllers by definition use a programmable memory for storing instructions used to implement specific function. The memory capacity is usually specified in terms of bytes or words with each byte generally containing 8 bits, sometimes 16 which plays an important role [Ref. 20 - handbook]. Smaller PLCs usually have fixed memory capacity. In the larger PLCs the memory is expandable. With some units this expansion can be carried out by the user using plug-in memory unit.[Ref. 20 - handbook]

It should be realized that the way memory is utilized in a specific PLC is just as important as absolute memory size. For example one or two instructions are required to enter a given ladder diagram element. Furthermore one, two or even more bytes are required to store a given instruction. The answer to these questions have a vital bearing on the memory size required. Some of the memory may be reserved for internal PLC use and is thus not available for storing the user's program. While designing a PLC, it is necessary to pay attention to how much memory is actually allocated for the program and how many bytes are required per element.

To know for sure how much memory will be needed for a given automation problem is to write the program then make a calculation based on it. However, a rough idea can be obtained using a rule is thumb. An average of about 20 relay contacts are required in a typical ladder diagram for each output signal. Since the number of outputs is easily determined from a description of the problem, a rough idea concerning the total number suggests multiplying the total number of input and output points by 10 to get the total expected number of contacts. In general, one program instruction is required for each contact. It is recommended to design a PLC with an additional 25-50% memory capacity, compare to it requirement to allow for program modifications and further expansion.

After having estimated the required memory capacity, you must decide whether volatile or nonvolatile memory is required. Volatile memory needs a battery backup to take over during a power break. The batteries are either of the rechargeable nickel-cadmium type or lithium batteries with a shelf life of several years. Note that the battery backup only preserves the program, It is not intended to keep the PLC operating during power break since its voltage and current capacity is insufficient to actuate the I/O modules.

In general random access memory (RAM) is usually preferred for program development and debugging because of the ease of making program changes. Once the program has been finalized it is frequently transferred to EPROM, which is more and insensitive to electrical noise and economical as in [Ref. (21 - pp. 1-4, 14-16), (22 - pp. 460-468)]. PLCs can also be designed with interchangeable EPROM modules. This lets the user store different program or even "recipes" for batch processes, with the program easily

exchanged by plugging in different modules.[Ref. (1 - pp. 52-56), (3 - pp. 30-32), (6 - pp. 19-20)]

Memory Architecture

The architecture of a PC's memory specifics how the various memory system described above are organized and used by the PLC to perform control function. Memory architecture is usually shown schematically by a memory utilization map, such as the one shown in Figure 3 . The memory map indicates that PC memory is divided into three major types: system memory, I/O status memory and application memory.[Ref. 2 - pp. 58-62]

System Memory

The system memory can be subdivided into types of memory: *Execution memory & Scratch pad memory.* Execution memory contains the executive program (or executive operating system). Because the executive program rarely changes, the executive memory should be read only memory (ROM). The executive program controls the operation of the PLC. It provides the translation between high-level programming language or graphical user interface (GUI) and the binary machine language, scans the PLCs to update system status and reads inputs and update outputs. It also helps as the scratch pad memory. This memory is not accessible to PC users.

Input / Output Memory

The input/output memory is accurately described by its name. It is a portion of RAM set aside for storage of current I/O status. Because the executive program requires I/O status updates, the I/O status memory can be thought of as a part of the system memory.

Application Memory

The application memory is the final area of PC memory, which is subdivided into two types: *Data memory and user memory*. These memories hold the data used by the microprocessor to fill its control function along with the user program that directs the microprocessor to perform its control function. [Ref. 4 - pp. 124] The following figure shows the standard memory map.

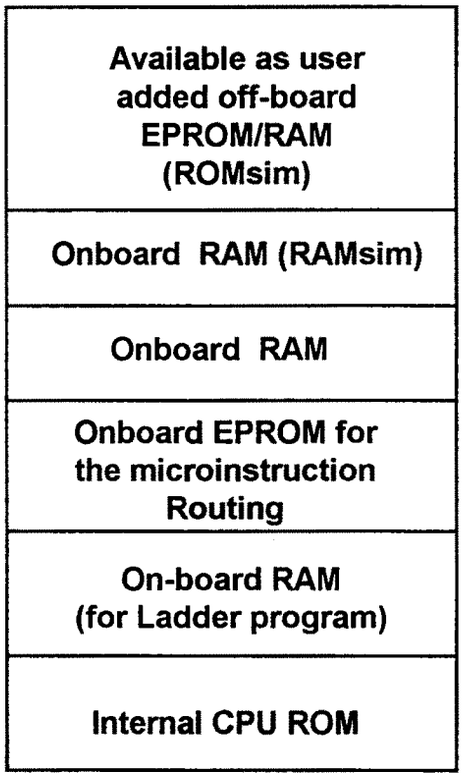

Figure 1 Memory Map Architecture

Input and Output Modules

The function of the I/O modules is to provide the necessary interface between the PLC and the controlled system. The input module performs four tasks electronically.

- First, it senses the presence/absence of an i/p signal at each of its i/ps terminals. The i/p signal tells what switch, sensor or other signal is on/off in the process being controlled.
- Secondly, it converts the input signal for on, or high, to a DC level usable by the module's electronic circuit. For a low, or off, input signal, no signal is converted and it indicates off.
- Thirdly, the input module carries out electronic isolation by electronically isolating the input module output from its input.
- Finally, its electronic circuit must produce an output to be sensed by the PLC CPU.

The output module operates in the opposite manner from the input module. A DC signal from the CPU is converted through each module section (terminal) to a usable output voltage, either AC or DC.

The I/O modules allows a PLC to be directly connected to process actuators and transducers (e.g. Pumps & valves), without the need for intermediate circuitry or relays.

To provide this signal conversion a programmable controller's I/O unit can be designed to suit different requirements as shown in table below.[Ref. (1 - pp. 56-58), (3 - pp. 36-38), (2 - pp. 63-67) (5 - cp. 6-26), (4 - pp. 101)]

Input module	5V (TTL level) input switch
	24V input switch
	110V input switch
	240V input switch
Output	24V 100mA output switch
	110V 1amp
	240V 1A a.c (triac)
	240V 2A a.c (relay)

Table 1 Input and output modules

It is a standard practice for all I/O channels to be electrically isolated from the controlled processes, using opto-isolators circuit. A typical PLC comes with attached I/O modules and their connecting terminals. The I/O modules are usually packaged in units of 4,8 or even more. Although the voltage levels of the various I/O modules installed in a PLC can be mixed as required, the modules installed in a given unit are generally of the

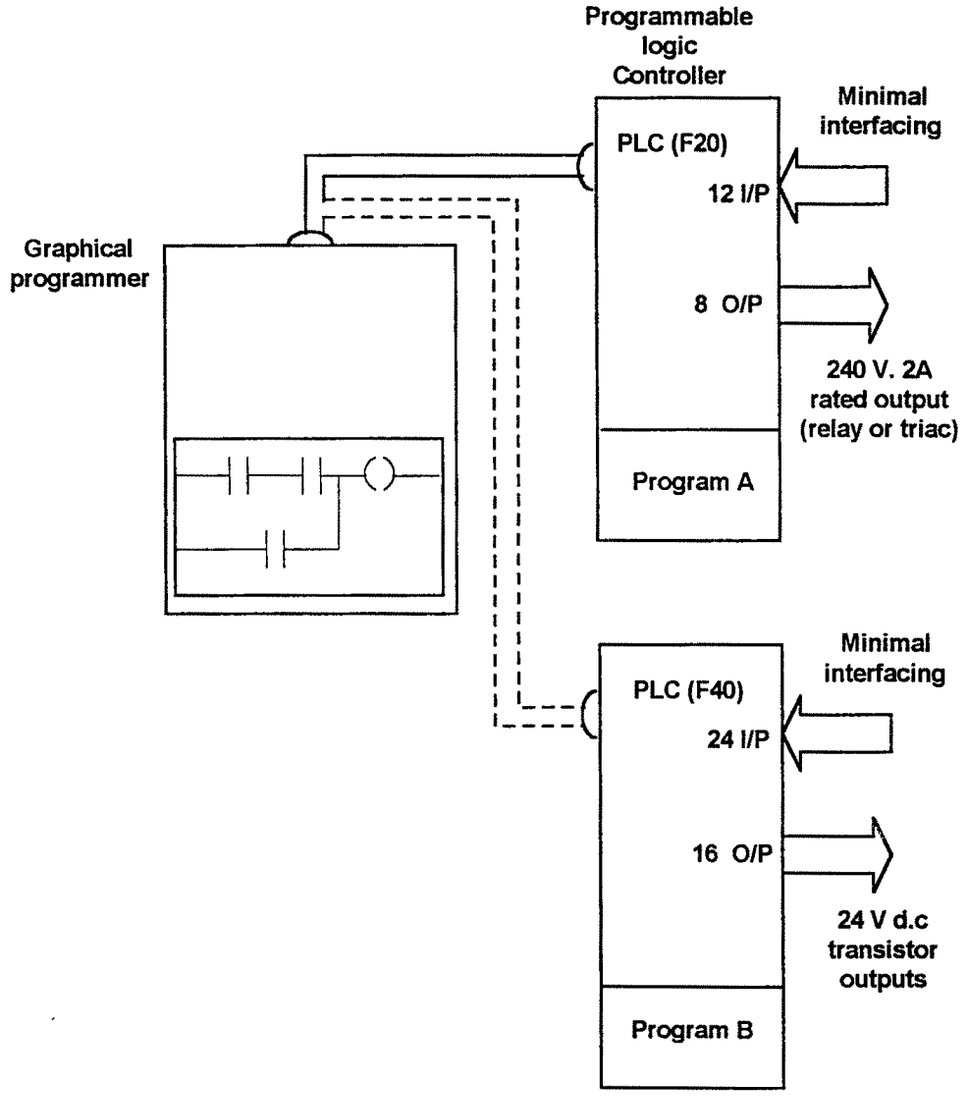

Figure 4 PLC input/output connected to plant equipment

same type. In most cases, the I/O units are designed with the aim of simplifying the connection of process transducers and actuators [Ref.12 - pp.123-142] to the programmable controller.

Every input and output point has a unique address or channel number which is used during program development to specify the monitoring of an input or the activation of a particular output within the program. Indication of the status of I/O unit, making it simple to check the operation of the process inputs and outputs from PLC itself as shown in the figure below.[Ref. 1 - pp. 57].

I/O modules can be divided into three general types, depending on the kind of signal handled.

- Binary (on-off), single bit also called as discrete [Ref. 2 - pp. 64-65].
- Binary multibit also called as numerical [Ref. 2 pp. 65-66].
- Analog interface, Special interface or Proportional integral derivative control.

According to signal the PLC has to handle, the choice can be made or the PLC can be designed to provide the three types of module.

PLC Programming Units

As the name implies, programming and data entry devices allow the PLC operator to enter programs, preset values and such details to the PLCs memory.

These units range from small hand-held portable units sometimes called manual programmers, to Cathode Ray Terminals (CRT) which consist of computer style keyboards with a video screen.

Manual programmers are similar in appearance to large pocket calculators having a number of keys and display, either light emitting diodes (LED's) or liquid crystal display (LCD). Each logic element of the ladder diagram is entered separately with serial or parallel connection achieved by using the AND or OR keys respectively. The NOT key is used specifically for normal closed contacts.[Ref. 5 - pp. 109, showing figure 8-18 programming & editing overlay (courtesy of Allen-Bradley)], [Ref. (1 - pp. 58-68), (4 - pp.29-40)]

Apart from entering the ladder diagram program in the PLC, the programming unit can also be used to check the program, i.e. read back the stored information. In a simpler unit, the information contained in each program word is displayed in alphanumeric form. In more sophisticated programming units, a small portion of the ladder diagram entered in the memory is actually shown in graphical form by means of the LCD.

The programming unit is generally connected to the PLC by a cable which is easily disconnected for portability. With some PLCs, the programming unit can be optionally attached firmly to the PLC remaining there during the operation. In such cases, the unit should be provided with key lock. Removal of the key prevents unauthorized persons from tampering with the program stored in the memory.

Many programming units have a facility for transferring the stored program to tape and conversely, for reading the program stored on a tape and send it back to the PLCs memory. This important feature makes it possible to save a collection of programs for different applications and to load any program without having to retype it.

Some programming units permit online programming i.e. the program can be modified while the PLC is running. Thus the system does not have to be shut down during a

program modification. However, according to some experts, online programming can create some serious safety hazards since certain output signals might undergo unexpected changes during the program modification.

The more sophisticated programming units consist of video or CRT terminals. These display several ladder diagram lines at a time and can also directly show power flow with each line during operation, which greatly facilitates troubleshooting. With each unit, the program is usually entered by moving a cursor along the screen. Most such units also have facilities for connecting a printer so that the ladder diagram or the resulting program can be printed and stored in a hard copy form. .[Ref. 5 pp. 113-115]

The cost of a special purpose programming unit designed for a specific PLC is inherently high because of the low sale value involved. A customer purchasing dozens of PLCs might buy only one programming unit.

The use of a microcomputer as a programmer which is used in designing of the PLC gives rise to advanced PLC. It has an additional advantage over a special purpose CRT programmer. The computer can print documentation, help in tasks of writing the PLC program using appropriate software. Indeed it is predicted that most PLCs will be programmed by microcomputer and that small manual programmers will only be used for small or micro PLCs or as portable programming units that operating personnel can easily carry about the plant.

Alternate Programming Languages

The software used with a programmable controller is perhaps the most important part of the PLC system for the everyday user. It is the software that allows the user to instruct the

PLC to perform its functions and that translates the PLC electrical signals into a format the user can understand.

Programming languages vary widely in their details even though they are usually quite similar in their nature. Two general classes of languages used in PLC software are

- Low Level Languages
- High Level Languages

Relay Ladder Diagram and Boolean Language are examples of Low Level Languages. [Ref. 5 - pp. 66-73]

Block diagrams, Computer type languages and Graphical interface are examples of high Level Languages. [Ref. 2 - pp. 99-102]

The Low Level Languages are the most common PLC languages used with most of the small and medium size PLCs. They are convenient for programming sequence type circuits with on-off outputs using contacts of various kinds, timers and counters. Although, logical instructions are easy to learn and use, they can be extremely time consuming to check and relate large program code to the actual circuit's functions. In addition, logic instructions tend to vary between different types of PLCs.

A preferable alternative is to use a graphic program which is discussed in the design of the advanced programmable logic controllers. Graphical programming allows the user to enter his program as a symbolic ladder circuit layout, using standard logic symbols to represent input contacts and output coils. This approach considered here is more user friendly

than programming with mnemonic logic instructions and can be considered as a high level form of language.

The programming panel translates or compiles these graphic symbols into machine (logic) instructions that are stored in the PLC memory, relieving the user of this task.

Relay Ladder Diagrams

These are currently the most popular type, owing to the fact that that electricians, control engineers and maintenance personnel are familiar with them. The ladder diagrams consist of symbols representing normal open or closed contacts and other components to control electric equipment.

The Figure 5 shows the relationship between a physical circuit and its ladder representation. In Figure 5, the motor is connected to power via three switches in series plus an overload switch (a)-(d). The motor will turn on if all the switches are in closed position. Figure 5 (b) shows its equivalent diagram. The ladder diagram shows standard symbols to represent the circuit elements and functions found in a control system. These are the same symbols taken into consideration while designing the graphical interface software. This example is similar to sample ladder diagram showed in [Ref. 32 - handbook] showing interaction of control relay between different rung.

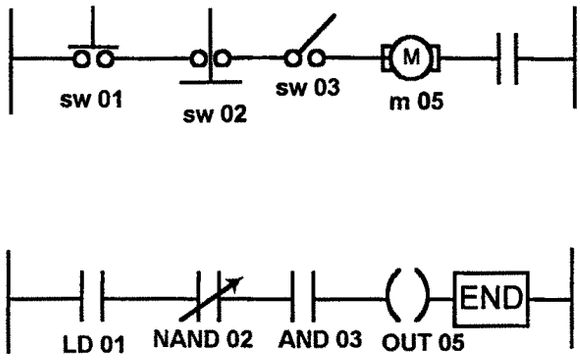

Figure 5 (a) Motor circuit electrical symbols/ladder symbols (b) equivalent relay ladder diagram

The ladder diagram can be represented in a Boolean language form as shown in the following table.

Instruction	Comment
LD 1001	;load the status of input switch 1001
AND 1002	;And the status of input switch 1002 with 1001
ADI 1003	;And the status of input switch 1003 with 1002
ADI 1004	;And the status of input switch 1004 with 1003
OUT 0001	;Output to the 0001 port
END	;End the operation

Table 4 Boolean representation of a ladder diagram

Construction of PLC ladder Diagrams

A PLC programming format must be observed when programming a PLC ladder diagram. Otherwise the PLC CPU will not accept the screen programmed ladder diagram into its memory. In some cases, when correctly formatted ladder diagrams are not received, an error message appears on the screen showing that the program was not entered and why. Why might the ladder diagrams be incorrect for a PLC? Because various ladder construction limitations were probably not observed. These are as follows:

- A contact must always be inserted in the
- A coil must be inserted at the end of a rung.
- All contacts must run horizontally. No vertical oriented contacts are allowed.
- The number of contacts per matrix(network) is limited.
- Only output may be contacted to a group of contacts or a rung.
- Contacts must be "nested" (a branch circuit programmed within a branch circuit) properly as required format.
- Flow must be from left to right.
- Contact progression should be straight across.

[Ref. (1- pp. 62 - 63 & 83 - 120), (14 - pp. 61-66), (15 - pp. 131 - 136), (14 - pp. 61-66), (15 - pp. 131-136)]

Signal Processing in Programmable Controller

Process Scanning Consideration

There two different methods for I/O processing in programming controller, which are *Continuous updating & Mass input/output coping.*

Continuous updating

This involves the CPU in scanning input channels as they occur in program instructions, with a built-in delay to ensure that only valid input signals are read into the processor. The delay-typically 3ms prevents contact bounce pulses and other noise from

entering the PLC. Output channels are driven (directly) when OUT instructions are following a logical operation. Outputs are latched in the I/O unit so they retain their status until the next updating. [Ref. (3 - pp. 48), (5 - pp. 68-70), (1 - pp. 62-63 & 83-120]

Mass input/output copying

In lager PLCs there are several hundred I/O points. Since the CPU can deal with only one instruction at a time during program execution, the status of each input point must be examined individually to determine its effect on the program. Since it requires a 3ms delay on each input, the total cycle time for a continuously sampled system becomes progressively longer as the number of inputs rises.

To allow rapid program execution, input and output updating may be carried out at one particular point in the program. Here a specific RAM within the PLC is used as buffer store between the control logic and the I/O unit. Each input and output has a cell in this I/O RAM. During I/O copying, the CPU scans all the inputs into the I/O units and copies their status into the I/O RAM cells. This happens at the start/end of each program cycle.

As the program is executed, the stored input data is read one location at a time from the I/O RAM. Logic operations are performed on the input data, and the resulting output signals are stored in the output section of the I/O RAM. Then, at the end of each program cycle the I/O copying routine transfers all output signals from the I/O RAM to the corresponding output channel, driving the output stages of the I/O unit. These output stages are latched, that is they retain their status until they are updated by the next I/O routine. [Ref. (1 - pp. 68-70), (3 - pp. 44 - 52), (4 - pp. 126-127)]

PLCs Internal Operation and Signal processing in the CPU

When a program is loaded into PLC, each instruction is placed in an individual code memory location (address). The CPU contains a program counter register which points to the next instruction to be fetch from memory. When the instruction is received by the CPU it is placed in the instruction register for decoding into the internal operation (microinstructions) required by the that particular instruction. This may result in further instructions being read from memory or from a physical device being driven by the CPU.

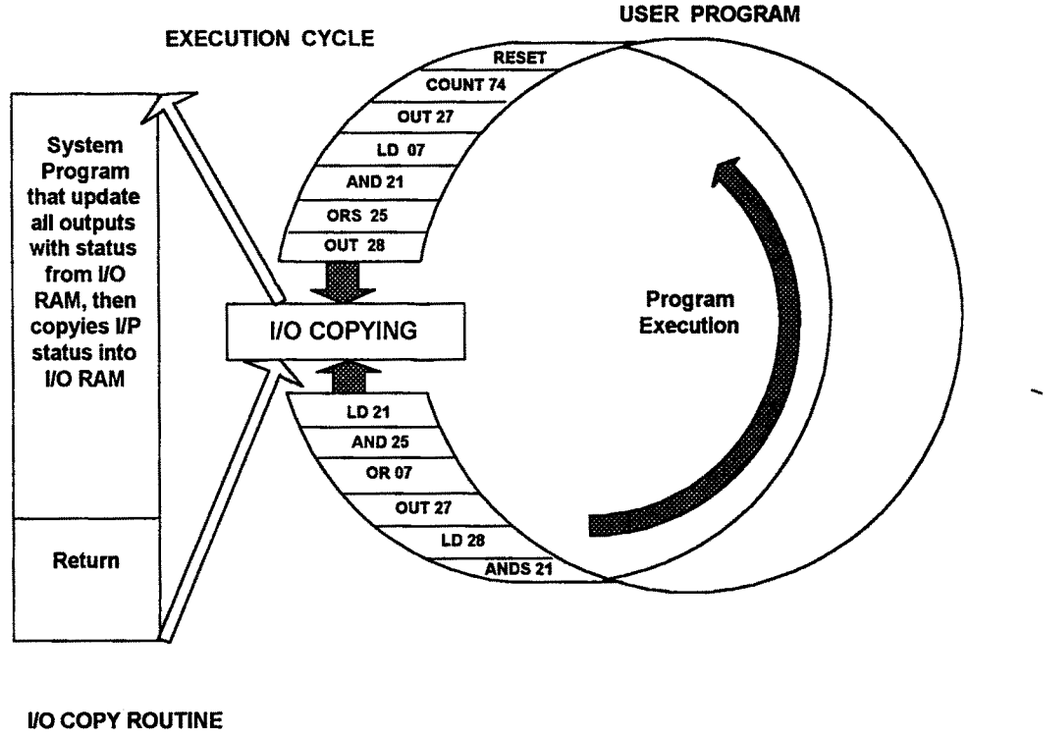

Figure 6 Input Output Copying

When the programmable controller is initially set to run, the program counter will point to address 0000 - the location of the first instruction as shown in **Error! Reference source not found.** and Table 5. The CPU then fetches, decodes and executes this program instructions, The first instruction is the SIC 09 (no. of input/output card). Thus the CPU finds that it has to examine the first element of the logic circuit - it scans 09 input cards and the input status is held in the I/O RAM. The CPU is now executing further instruction to complete the logical circuit. The program automatically increments to point to the next address, where the second instruction is located - AD 21 normal open contact associated with the input switch number 09 card. The input status is held in the I/O RAM, so the CPU scans the RAM location and reads its status from the temporary memory location. Now the CPU fetches the third instruction - AND 01 - and executes it. This causes the CPU to load the input status from the input memory location 01 and then perform a logical AND function with the status in the memory accumulator which contents the status of input switch 21 and the temporary result is stored in the memory accumulator. The program counters stops onto address 0004. Here the instruction - OUT 28 is fetched and decoded instructing the CPU to pass the result to the output memory location of the I/O RAM corresponding to location 28. The sequence then continues until either the last memory location is reached by the program counter, or an END instruction is encountered. If the last instruction is END which denotes that the status stored is output to the output cards. In both cases causes the program counter resets to 0000, causing the program to start at the beginning again and the cycle continues. The program counter can be programmed to jump over one or more address if required, so

that some instructions are not processed. The is done by the use of a conditional jump instruction in a logic program.[Ref. 1 - pp. 55-57]

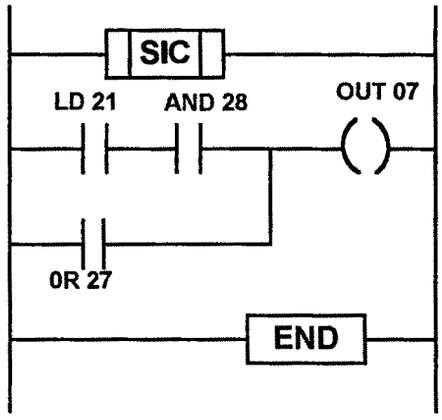

Figure 7 Ladder diagram for CPU Signal Processing

Memory Address	OpCode	Operand
0000	SIC	-
0001	LD 21	A0 21
0002	AND 28	A1 28
0003	ORS 27	A2 27
0004	OUT 07	AA 07
0005	END	AB

Table 5 PLC Program list for ladder diagram

Fetch, decode & execute first instruction	Scan necessary contacts	Next instruction	Scan or actuate devices	Next instruction	Scan I/O or actuate O/P etc.
5ps delay	3ms delay	5ps	3 ms delay	5ps	3ms delay

Table 6 Subroutine Execution

I/O copy	Program execution	End I/O copy	Program
Copy all inputs into RAM	Fetch, decode & execute all instruction in sequence	copy all output from O/P RAM to output unit, & input into RAM	

Table 7 Program Execution

The internal operation is explained here considering the time delay for executing the program in the following table.

Execution time depends on length of total program e.g. 1K program = 5ms	Fixed length delay e.g. 5ms

This task is carried out automatically by the CPU as a subroutine to the normal program. A subroutine is a small program designed to perform a specific function, that may be called by the main program. In this case, the I/O subroutine is automatically accessed by the underlying CPU control program - the monitor. I/O copying takes place between the end of one program cycle and the start of the next.

Effect of Scanning Time

The effect of scanning is important parameter, and should be considered. It is therefore covered here in detail.

The time required for the PLC to carry out one scan is of the order of 5-50 ms, with 10-20 ms being typical. The time depends on the particular PLC and, more important, on the length of program memory. Other things being equal, scanning time is roughly proportional

to memory length. Some PLCs have an "END" command, which should be placed at the end of the program. When "END" is reached, the PLC begins a new scan, rather than continuing to scan the remaining unused memory built into the PLC. Thus the scanning time can be reduced considerably for short programs.

In general, the scanning time should be as short as possible, otherwise the PLC is unable to sense very brief events. For example, suppose we want to count parts passing by on a conveyor belt. The parts are detected by a photo electric cell connected to an input module. Each part passing in front of the cell produces a brief pulse width depending on the part width is shorter than the scanning time, it might fall between two successive scans of the input modules, so that can be solved using special high-speed counter input modules, available with many PLCs.

For systems where a very fast response to some very brief event is required, it might pay to design a special hard-wired electronic switching circuit for handling this particular task, and let the PLC take care of the remainder of the control problem.

With most PLCs, the status of all I/O modules is updated at the beginning of each scan cycle. However, some PLCs provide for a scan interrupt, during which a given I/O module can be updated without waiting for the scan cycle to be completed. This feature helps in "catching" fast-changing inputs.

While short scanning times are desirable in principle, very short times may not provide any actual benefit, because of the relatively slows speed of response of most input modules. Most input modules contain a filter to cut down electrical noise, and this produces an

inherent time delay of the order of 5-10msec. Scanning times shorter than this, therefore, provide little benefit.

PLC functions in Short

So far, we examined the complete PLC system and peeked inside to see how it operates, explored general programming procedures and more specifically the programming of on-off inputs and on-off outputs. Furthermore, we looked at ladder diagram for a process problem. It is time to move into consider the function of PLC.

Some of the function performed by a programmable controller is taken into account while designing are as followed.

Control relay function

These functions involve the generation of an output signal from one or more inputs according to a particular logic rule contained in the PLC memory. This is illustrative of the kind of logic rules used in control relay functions.

Timing function

Most programmable controllers allow the programming of timing functions. This might be used to generate an output signal a specified delay time after an input signal has been received. Another example would be to maintain an output signal for a certain length of time and then shut off.

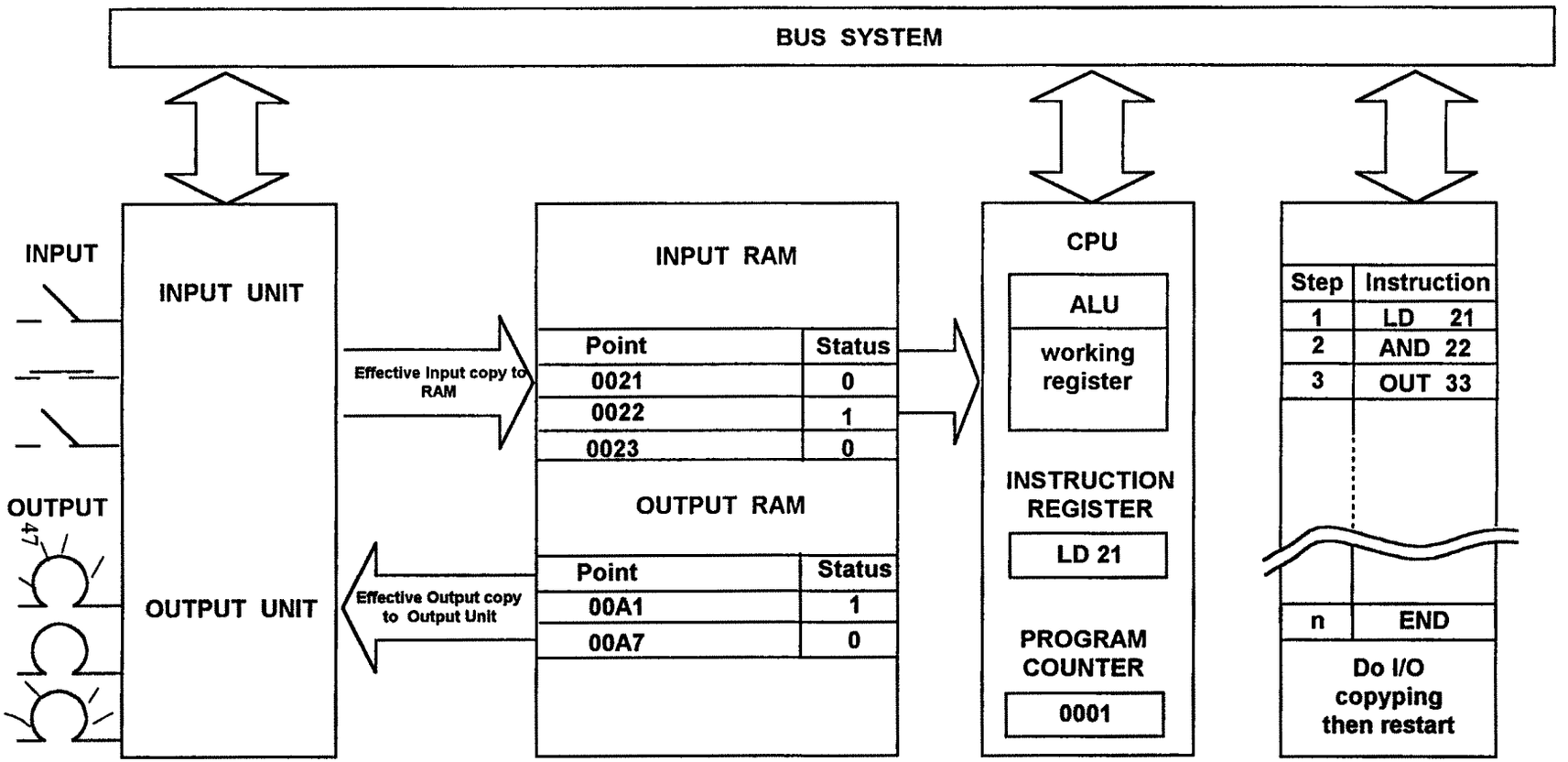

Figure 8 CPU Signal Processing

Counting function

Counting functions are similar to timing functions. The counter adds up the number of input contact closures and generates a programmed output when the sum reaches a certain count. The counter would then be reset to repeat the cycle.

Arithmetic function

The PLCs can be equipped with more features and capabilities. In addition to the basic relay logic function, timers, and counters, some PLCs offer mathematical functions such as addition, subtraction and in certain cases, multiplication and division. Hence it is possible to use the controller for more sophisticated processes where complex calculations are required.

Comparison function

Many PLCs have only two Compare functions *Equal To* and *Greater Than* or *Less Than*. To perform any one of the other four functions (*not Equal To , Less Than, Greater Than, & Less Than or Equal To)*, a combination of the basic functions are used. But rather than having the combinations , you can simply have an instruction of each case. This function is used to perform comparison of the status available.

Analog control function

Analog control devices used to accomplish proportional and derivative control functions are available for PLCs. Thus the PLC can be utilized to regulate analog devices directly.

The first three functions are common features on programmable controllers. Arithmetic and analog control functions are available on requirement making it more powerful PLCs. [Ref. (1 - pp. 87-120), (2 - pp. 81 - 106), (3 - chapter. 5 - 14), (6 - chapter. 7 - 12), (4 - chapter. 5 - 9), (5 - chapter. 10 - 4)]

PLC communication and Automation

PLC communication

There is a need to pass information between PLCs and other devices within an automated plant, this can be provided by a communication facility. In case of small PLCs, the necessary communication hardware and software is incorporated in the programmable controller body, whilst large PLCs have a range of communications modules.

Common uses of PLC communication ports

- Presenting operating data and alarms etc. via printers or VDUs.
- Data logging into archive files or records, to be used for process performance analysis and management information.
- Passing values/parameters into existing PLC program from operator terminals or supervisory controllers.
- Changing resident PLC program - uploading/downloading from a Supervisory controller (microcomputer).
- Forcing I/O points and memory element from a remote terminal.

- Linking a PLC into a control hierarchy containing several size of PLC and computer.

Communication Schemes

Two general schemes are used by which bits of data are transmitted from the peripheral to the PC and vice versa: *Parallel transmission & Serial transmission.*

Parallel Transmission

Data in the form of binary bits, are stored in the PLC logic gates. These are transmitted through the transmission lines. In parallel transmission, each bit from each gate in the PLC is carried by a separate line. This means that for large PCs, several lines are needed for communication between the PLC and computer.

One bit per line is one of disadvantages of it is that, because each line is dedicated, parallel interfaces are not usually interchangeable for different peripherals. On the other hand, the primary advantage of parallel transmission is speed.

Serial Communication

The principle behind serial transmission is that data from the computer is accepted one at a time by the driver of PLC. When each bit of data is received by the driver, it is driven, or transmitted, along a single transmission line to receiver.

In most PLC systems, data is transmitted in the form of ASCII code. The most interesting feature of serial transmission is that only one transmission line is used. Although they are generally slow, they can transmit data over a distance of 50 feet as compared to 6 feet for parallel transmission.

Communication Standards

Several communication standards specify signal levels and physical details of the interface, or transmission cable. The four that find most usage in the PLC system are the RS-232C, the RS449, the 20mA current loop and the IEEE 488.

The RS-232C and RS-449 are officially proclaimed as standards by the Electronic Industries Association (EIA). The IEEE 448 is officially proclaimed as a standard by the institute of Electrical and Electronics Engineers (IEEE). The 20mA current loop is not officially defined by any organization, but has become a standard through common usage.

Standard Communication Requirements.

There are three aspects that must be considered in serial communication. The rated speed of the transmission, that is the number of bits per second that are to be sent over the communication link, and the duration of each of these bits. Secondly, the logic levels should be considered, in other words, what signal represents logic 1 to 0 to transmit information.

Finally, the method for synchronizing the data should be the transmission

Following Table 8 gives the specification of different communication standard:

Standards	Speed & Distance	Logic Levels	Synchronizing
RS 232	75 - 19000 bps, 30m at 9600 band, additional power supply may be necessary, which can extend the operating distance up to 100m, using screen capable at lower data rate.	Voltage between +5V & +25V for two possible signal state (space) & a between -5V & -25V for the other (mark) signal (normally += 12V)	Control signals are available .

20mA Current Loop	Provides long distance, for 9600 bps signal can be transmitted up to a distance of 300m	20mA loop	lack of control (handshake) signals.

Table 8 Standard Communication Requirements

Programmable Controllers & Networks

This chapter addresses communication and emphasize communication between many PLCs or between PLCs and intelligent devices such as computers. This type of network is often called a communication network.

The capability of a PLC to collect data from the manufacturing is inherently provided through its I/O system. How easily the data can be conveyed to other PLC and computer is a function of the communication provided by the PLC

Communication between several PLCs

When several PLCs or machines have to send or receive data to and from each other, it is possible to merge there signals using a concentrator unit. This unit switches between channels under software control from the supervising controller, or in response to a request signal from one of the linked PLCs.

However, when it becomes desirable or necessary to interconnect a large number of terminal or intelligent machines, a communications network is often used. For this purpose, a local area network (LAN) provides the best physical link between all devices plus providing overall data exchange management or protocol, ensuring that each device can 'talk' to the other machines and understand data received from them.

The advent of the local area network offers:

- a data transmission system linking computers and associated devices with a restricted geographical area (up to 10km, although 1km is more typical).
- dispersion of equipment make cabling for point-to-point impractical and prohibitively expensive.
- all device can access and share data and program.
- a flexible base for contributing communications architectures.

LANs are commonly used in business application to allow several users to share costly software packages and peripheral equipment such as printers and hard disk storage.

Type of Communication System in Network

Point - to - Point Communication

Most PLCs have at least one communication port built in named as the program loader interface. These parts typically use unusual protocols that can require considerable effort to implement.

Almost all PLCs offers some from to point-to-point communication via an RS 2332C ASCII based link that utilizes a separate intelligent I/O module for this purpose. This module may support multisrop RS422 and RS485 links as well. Although their modules typically will provide the necessary software function to allow module to communicate with the PLC processor via the I/O system, in some cases these modules are completely user programmable for the external ASCII communication port. This means that all the external communication software would need to be written by the user. Alternatively some modules come

preprogrammed with a fixed protocol that allow I/O and internal memory to be accessed by an external RS232 device. With the ASCII communication modules, it is possible to communicate with a wide variety of devices, such as color graphics terminals, intelligent push buttons, stations, bar-code readers, servo motor controllers and ASCII terminals.

The instruction set of the PLC can have an impact on the how the ASCII information is processed. PLCs with few instructions can be difficult to program for ASCII information processing function, such as search, compare ASCII to binary, binary to ASCII , block transfer function t (to transfer data to or from the ASCII communication module), and flexible condition execution statements are crucial for the efficient handling of ASCII information. If PLC instruction set is insufficient, the most of the processing will need to be done either in the external device or in the communication module itself.

Network Communication.

Most of the networks provide two basic functions

- Reading and writing variables
- Uploading and Downloading programs.

However, because these networks are designed to provide communication functions, not necessarily control function, using a network inside of a control loop requires careful planning and evaluations. Some networks have difficulty transporting information from one point on the network is such a manner as to allow the use of this information in a time critical

control application. The following point should be considered when evaluating a network for PLC application.

Response Time:

The length of time required for an input changing state on one node of a network to a second node receiving information that the input has changed is a critical parameter when trying to implement control of a process over a network. Some networks give only a probabilistic response time based on some hypothetical installation. However, one always should know the precise response time limits before putting control information on a network.

Throughput:

Applications that require a great deal of data to be communication will find the throughput characteristics, typically measured in the bytes per second, of the network more important. Although some networks that exhibit a fast response time may offer high throughput, this is not necessarily true. It is possible for a network with long response times to exhibit a high throughput rate. Also, a high bit signaling rate on the wire (sometimes referred to as large packets of data) to be sent over the networks will exhibit a higher throughput than networks that only support small packet sizes.

Error Checking :

Any network that is used for transferring control information should utilize extensive error checking on the information sent over it. Both ends of the network, the sender and the receiver, should be capable of detecting error and should also perform specific and know error recovery mechanisms, such as retransmission. At a minimum both the sender and

receiver should be notified that there was an error so that the control engineers can program their own recovery scheme.

Access Mechanisms:

Because a network consists of multiple devices, some methods for determining who has access to the network at any given time must be used. Two of the popular access mechanisms are master-slave and peer-to-peer.

On master-slave system there is one master PLC. The master sends commands outs to the slave PLC and they respond appropriately. The slaves on the network never initiate their on commands. They only respond to commands from the master.

The peer-to-peer mechanism allows any device on the network to initiate message. However, as in the case of people talking, if everybody talks at once, nothing intelligible can be heard. Peer-to-peer networks need some mechanism for determining access between all the devices, not just between the master and the slave. Various mechanism for determining access have been implemented, such as token passing and carrier-sense multiple access with collision detection (CSMA/CD).

If control information is to sent over the network, the user should make certain that either the access mechanism is deterministic, that the access time is known and is not probabilities, or that the network characteristics are such that this information can be used under all conditions. Control signals, such as end of travel and emergency stop, should be hard wired. Depending on a network to send safely interlocks and related information

without hard wired backups can result in a hazardous system if the network fails or if access times degrade.

Network protocols:

The protocols used by the network will dictate the message formats and functionality of the network. Because there is considerable variation between the protocols that each PLC manufacture uses on its own network, care must be taken to ensure that the network is capable of performing the functions that are required for the application. Most PLC manufacturers offer their network a set of protocols that are unique to them. These networks typically are referred to as proprietary network. In recent years, networks based on initially recognized standards e.g. manufacture automation model have been introduction by some leading manufactures. These MAP/OSI networks make use of a message specification. This is a common form of communication using high level commands (Read, Write, Upload, Download, Start, Stop etc.) which obviates the need to write a separate communication driver for all the different bands and type of devices in system. Although these networks may not be completely suited for all types of control applications, they can be used in many applications.

In C/M environments designed as integrated manufacturing processes composed of a wide variety of devices consisting of multiple brands of PLCs and multiple brands of computers, a standards based network such as MAPI/OSI, may have a significant advantage over proprietary network because MAP/OSI network can be procured from many different computer and PLC vendors. Propriety networks in these environments may require extensive

use of gateways and custom software drives which increases the cost of system implementation. This makes propriety networks inappropriate for enterprise CIM.