DOWNLOAD PDF PROGRAMMABLE LOGIC

Chapter 1 : Digital Circuits Programmable Logic Devices

A programmable logic device (PLD) is an electronic component used to build reconfigurable digital racedaydvl.com a logic gate, which has a fixed function, a PLD has an undefined function at the time of manufacture.

Overview[edit] PLC system in a rack, left-to-right: Programs to control machine operation are typically stored in battery-backed-up or non-volatile memory. Before the PLC, control, sequencing, and safety interlock logic for manufacturing automobiles was mainly composed of relays , cam timers , drum sequencers , and dedicated closed-loop controllers. Since these could number in the hundreds or even thousands, the process for updating such facilities for the yearly model change-over was very time consuming and expensive, as electricians needed to individually rewire the relays to change their operational characteristics. When digital computers became available, being general-purpose programmable devices, they were soon applied to control sequential and combinatorial logic in industrial processes. However these early computers required specialist programmers and stringent operating environmental control for temperature, cleanliness, and power quality. To meet these challenges the PLC was developed with several key attributes. It would tolerate the shop-floor environment, it would support discrete bit-form input and output in an easily extensible manner, it would not require years of training to use, and it would permit its operation to be monitored. Since many industrial processes have timescales easily addressed by millisecond response times, modern fast, small, reliable electronics greatly facilitate building reliable controllers, and performance could be traded off for reliability. The winning proposal came from Bedford Associates of Bedford, Massachusetts. Modicon, which stood for modular digital controller. One of the people who worked on that project was Dick Morley , who is considered to be the "father" of the PLC. It was presented to Modicon by GM , when the unit was retired after nearly twenty years of uninterrupted service. Modicon used the 84 moniker at the end of its product range until the made its appearance. The automotive industry is still one of the largest users of PLCs. In a parallel development Odo Josef Struger is sometimes known as the "father of the programmable logic controller" as well. These PLCs were programmed in " ladder logic ", which strongly resembles a schematic diagram of relay logic. This program notation was chosen to reduce training demands for the existing technicians. Other early PLCs used a form of instruction list programming, based on a stack-based logic solver. Another method is state logic , a very high-level programming language designed to program PLCs based on state transition diagrams. Many early PLCs did not have accompanying programming terminals that were capable of graphical representation of the logic, and so the logic was instead represented as a series of logic expressions in some version of Boolean format , similar to Boolean algebra. As programming terminals evolved, it became more common for ladder logic to be used, for the aforementioned reasons and because it was a familiar format used for electro-mechanical control panels. Newer formats such as state logic and Function Block which is similar to the way logic is depicted when using digital integrated logic circuits exist, but they are still not as popular as ladder logic. A primary reason for this is that PLCs solve the logic in a predictable and repeating sequence, and ladder logic allows the programmer the person writing the logic to see any issues with the timing of the logic sequence more easily than would be possible in other formats. Programming[edit] Early PLCs, up to the mids, were programmed using proprietary programming panels or special-purpose programming terminals , which often had dedicated function keys representing the various logical elements of PLC programs. Programs were stored on cassette tape cartridges. Facilities for printing and documentation were minimal due to lack of memory capacity. The oldest PLCs used non-volatile magnetic core memory. More recently, PLCs are programmed using application software on personal computers, which now represent the logic in graphic form instead of character symbols. The programming software allows entry and editing of the ladder-style logic. In some software packages, it is also possible to view and edit the program in function block diagrams, sequence flow charts and structured text. Generally the software provides functions for debugging and troubleshooting the PLC software, for example, by highlighting portions of the logic to show current status during operation or via simulation. The software will upload and download the PLC program, for backup and restoration purposes. In some models of programmable controller, the program is transferred from a personal computer to the PLC through a programming board which writes the program into a removable chip such as an EPROM. Functionality[edit] The functionality of the PLC has evolved over the years to include sequential relay control, motion control, process control , distributed control systems , and networking. The data handling, storage, processing power, and communication capabilities of some modern PLCs are approximately equivalent to desktop computers. Desktop computer controllers have not been generally accepted in heavy industry because the desktop computers run on less stable operating systems than do PLCs, and because the desktop computer hardware is typically not designed to the same levels of tolerance to temperature, humidity, vibration, and longevity as the processors used in PLCs. Operating systems such as Windows do not lend themselves to deterministic logic execution, with the result that the controller may not always respond to changes of input status with the consistency in timing expected from PLCs. Desktop logic applications find use in less critical situations, such as laboratory automation and use in small facilities where the application is less demanding and critical, because they are generally much less expensive than PLCs. Discrete inputs are given a unique address, and a PLC instruction can test if the input state is on or off. Just as a series of relay contacts perform a logical AND function, not allowing current to pass unless all the contacts are closed, so a series of "examine if on" instructions will energize its output storage bit if all the input bits are on. Similarly, a parallel set of instructions will perform a logical OR. In an electro-mechanical relay wiring diagram, a group of contacts controlling one coil is called a "rung" of a "ladder diagram ", and this concept is also used to describe PLC logic. Some models of PLC limit the number of series and parallel instructions in one "rung" of logic. The output of each rung sets or clears a storage bit, which may be associated with a physical output address or which may be an "internal coil" with no physical connection. Such internal coils can be used, for example, as a common element in multiple separate rungs. Unlike physical relays, there is usually no limit to the number of times an input, output or internal coil can be referenced in a PLC program. Some PLCs enforce a strict left-to-right, top-to-bottom execution order for evaluating the rung logic. This is different from electro-mechanical relay contacts, which in a sufficiently complex circuit may either pass current left-to-right or right-to-left, depending on the configuration of surrounding contacts. The elimination of these "sneak paths" is either a bug or a feature, depending on programming style. More advanced instructions of the PLC may be implemented as functional blocks, which carry out some operation when enabled by a logical input and which produce outputs to signal, for example, completion or errors, while manipulating variable internally that may not correspond to discrete logic. Timers and counters[edit] The main function of a timer is to keep an output on for a specific length of time. A good example of this is a garage light, where you want power to be cut off after 2 minutes so as to give someone time to go into the house. A Delay-OFF timer activates immediately when turned on, counts down from a programmed time before cutting off, and is cleared when the enabling input is off. A Delay-ON timer is activated by input and starts accumulating time, counts up to a programmed time before cutting off, and is cleared when the enabling input is turned off. A Delay-ON-Retentive timer is activated by input and starts accumulating time, retains the accumulated value even if the ladder-logic rung goes false, and can be reset only by a RESET contact. Counters are primarily used for counting items such as cans going into a box on an assembly line. This is important because once something is filled to its max the item needs to be moved on so something else can be filled. Up counters count up to the preset value, turn on the CTU CounT Up output when the preset value is reached, and are cleared upon receiving a reset. Down counters count down from a preset value, turns on the CTD CounT Down output when 0 is reached, and are cleared upon reset. These small devices are typically made in a common physical size and shape by several manufacturers, and branded by the makers of larger PLCs to fill out their low end product range. Most of these have 8 to 12 discrete inputs, 4 to 8 discrete outputs, and up to 2 analog inputs. Size is usually about 4" wide, 3" high, and 3" deep. Most have a small plug for connecting via RS or RS to a personal computer so that programmers can use simple Windows applications for programming instead of being forced to use the tiny LCD and push-button set for this purpose. Unlike regular PLCs that are usually modular and greatly expandable, the PLRs are usually not modular or expandable, but their price can be two orders of magnitude less than a PLC, and they still offer robust design and deterministic execution of the logics. Features[edit] Control panel with PLC grey elements in the center.

The unit consists of separate elements, from left to right; power supply , controller, relay units for in- and output Control panel with a PLC user interface for thermal oxidizer regulation. PLC input can include simple digital elements such as limit switches , analog variables from process sensors such as temperature and pressure , and more complex data such as that from positioning or machine vision systems. Scan time[edit] A PLC program generally loops i. The program then runs from its first instruction rung down to the last rung. Excessively long scan times may mean the response of the PLC to changing inputs or process conditions is too slow to be useful. As PLCs became more advanced, methods were developed to change the sequence of ladder execution, and subroutines were implemented. Precision timing modules, or counter modules for use with shaft encoders , are used where the scan time would be too long to reliably count pulses or detect the sense of rotation of an encoder. This allows even a relatively slow PLC to still interpret the counted values to control a machine, as the accumulation of pulses is done by a dedicated module that is unaffected by the speed of program execution on the PLC. Process of a scan cycle[edit] There are 5 main steps in a scan cycle:

Chapter 2 : Programmable logic controller - Wikipedia

A programmable logic controller (PLC) or programmable controller is an industrial digital computer which has been ruggedized and adapted for the control of manufacturing processes, such as assembly lines, or robotic devices, or any activity that requires high reliability control and ease of programming and process fault diagnosis.

Programmable Logic Controllers PLC Chapter 6 - Ladder Logic Before the advent of solid-state logic circuits, logical control systems were designed and built exclusively around electromechanical relays. Instead, digital computers fill the need, which may be programmed to do a variety of logical functions. As an acronym, it meant Modular Digital Controller, and later became the name of a company division devoted to the design, manufacture, and sale of these special-purpose control computers. Other engineering firms developed their own versions of this device, and it eventually came to be known in non-proprietary terms as a PLC, or Programmable Logic Controller. The purpose of a PLC was to directly replace electromechanical relays as logic elements, substituting instead a solid-state digital computer with a stored program, able to emulate the interconnection of many relays to perform certain logical tasks. In an effort to make PLCs easy to program, their programming language was designed to resemble ladder logic diagrams. Thus, an industrial electrician or electrical engineer accustomed to reading ladder logic schematics would feel comfortable programming a PLC to perform the same control functions. PLCs are industrial computers, and as such their input and output signals are typically volts AC, just like the electromechanical control relays they were designed to replace. Although some PLCs have the ability to input and output low-level DC voltage signals of the magnitude used in logic gate circuits, this is the exception and not the rule. The following illustration shows a simple PLC, as it might appear from a front view. In this way, the PLC is able to interface with real-world devices such as switches and solenoids. The actual logic of the control system is established inside the PLC by means of a computer program. This program dictates which output gets energized under which input conditions. Although the program itself appears to be a ladder logic diagram, with switch and relay symbols, there are no actual switch contacts or relay coils operating inside the PLC to create the logical relationships between input and output. These are imaginary contacts and coils, if you will. Consider the following circuit and PLC program: When the pushbutton switch is unactuated unpressed , no power is sent to the X1 input of the PLC. They do not exist as real electrical components. They exist as commands in a computer programâ€”a piece of software onlyâ€”that just happens to resemble a real relay schematic diagram. Once a program has been loaded to the PLC from the personal computer, the personal computer may be unplugged from the PLC, and the PLC will continue to follow the programmed commands. Control System Behavior The true power and versatility of a PLC is revealed when we want to alter the behavior of a control system. Since the PLC is a programmable device, we can alter its behavior by changing the commands we give it, without having to reconfigure the electrical components connected to it. For example, suppose we wanted to make this switch-and-lamp circuit function in an inverted fashion: In the following illustration, we have the altered system shown in the state where the pushbutton is unactuated not being pressed: In this next illustration, the switch is shown actuated pressed: One of the advantages of implementing logical control in software rather than in hardware is that input signals can be re-used as many times in the program as is necessary. For example, take the following circuit and program, designed to energize the lamp if at least two of the three pushbutton switches are simultaneously actuated: To build an equivalent circuit using electromechanical relays, three relays with two normally-open contacts each would have to be used, to provide two contacts per input switch. Take for instance this next system, a motor start-stop control circuit: Fail-safe Design in PLC-Controlled Systems An important point to make here is that fail-safe design is just as important in PLC-controlled systems as it is in electromechanical relay-controlled systems. One should always consider the effects of failed open wiring on the device or devices being controlled. In this motor control circuit example, we have a problem: So, we see there is no operational difference between this new design and the previous design. The result, then, for a wiring failure on the X2 input is that the motor will immediately shut off. In this circuit, the lamp will remain lit so long as any of the pushbuttons remain unactuated unpressed. To make the lamp turn off, we will have to actuate press all three switches, like this: Advanced PLC Functionality This section on programmable logic controllers illustrates just a small sample of their capabilities. As computers, PLCs can perform timing functions for the equivalent of time-delay relays , drum sequencing, and other advanced functions with far greater accuracy and reliability than what is possible using electromechanical logic devices. Most PLCs have the capacity for far more than six inputs and six outputs. Fit into a control cabinet, a PLC takes up little room, especially considering the equivalent space that would be needed by electromechanical relays to perform the same functions: Because a PLC is nothing more than a special-purpose digital computer, it has the ability to communicate with other computers rather easily. The actual pumping station is located miles away from the personal computer display:

Chapter 3 : Programmable Logic Controllers

Programmable Logic Controller (PLC) is a digitally operating electronic system or it can be termed as industrial computer which is designed especially for use in industrial environment. It can be easily integrated into the industrial control systems to expand and enhance the performance of industrial processes.

This device, the TMS, was programmed by altering the metal layer during the production of the IC. GE obtained several early patents on programmable logic devices. This was more popular than the TI part but cost of making the metal mask limited its use. The device is significant because it was the basis for the field programmable logic array produced by Signetics in , the 82S Intersil actually beat Signetics to market but poor yield doomed their part. The MMI was completed in and could implement multilevel or sequential circuits of over gates. The device was supported by a GE design environment where Boolean equations would be converted to mask patterns for configuring the device. The part was never brought to market. TI coined the term programmable logic array for this device. Programmable array logic PAL devices have arrays of transistor cells arranged in a "fixed-OR, programmable-AND" plane used to implement "sum-of-products" binary logic equations for each of the outputs in terms of the inputs and either synchronous or asynchronous feedback from the outputs. This made the parts faster, smaller and cheaper. The PAL Handbook demystified the design process. This device has the same logical properties as the PAL but can be erased and reprogrammed. The GAL is very useful in the prototyping stage of a design, when any bugs in the logic can be corrected by reprogramming. GALs are programmed and reprogrammed using a PAL programmer, or by using the in-circuit programming technique on supporting chips. Complex programmable logic device PALs GALs are available only in small sizes, equivalent to a few hundred logic gates. These contain the equivalent of several PALs linked by programmable interconnections, all in one integrated circuit. CPLDs can replace thousands, or even hundreds of thousands, of logic gates. A second method of programming is to solder the device to its printed circuit board, then feed it with a serial data stream from a personal computer. This type of device is based on gate array technology and is called the field-programmable gate array FPGA. The 82S was an array of AND terms. The 82S also had flip flop functions. The term "field-programmable" means the device is programmed by the customer, not the manufacturer. In most larger FPGAs, the configuration is volatile and must be re-loaded into the device whenever power is applied or different functionality is required. In general, CPLDs are a good choice for wide combinational logic applications, whereas FPGAs are more suitable for large state machines such as microprocessors. Other variants[edit] At present,[when? These are microprocessor circuits that contain some fixed functions and other functions that can be altered by code running on the processor. Designing self-altering systems requires engineers to learn new methods, and that new software tools be developed. PLDs are being sold now that contain a microprocessor with a fixed function the so-called core surrounded by programmable logic. These devices let designers concentrate on adding new features to designs without having to worry about making the microprocessor work. The memory is used to store the pattern that was given to the chip during programming. Most of the methods for storing data in an integrated circuit have been adapted for use in PLDs.

Chapter 4 : Programmable Logic Controllers (PLC) | Ladder Logic | Electronics Textbook

This report studies the global Programmable Logic Control Systems market status and forecast, categorizes the global Programmable Logic Control Systems market size (value & volume) by manufacturers, type, application, and region.

There are three kinds of PLDs based on the type of array s , which has programmable feature. Basically, users can program these devices or ICs electrically in order to implement the Boolean functions based on the requirement. Here, the term programming refers to hardware programming but not software programming. The user has the flexibility to program the binary information electrically once by using PROM programmer. The block diagram of PROM is shown in the following figure. Here, the inputs of AND gates are not of programmable type. So, we have to generate 2n product terms by using 2n AND gates having n inputs each. We can implement these product terms by using nx2n decoder. Here, the inputs of OR gates are programmable. That means, we can program any number of required product terms, since all the outputs of AND gates are applied as inputs to each OR gate. Therefore, the outputs of PROM will be in the form of sum of min terms. So, we require a 3 to 8 decoder and two programmable OR gates for producing these two functions. The corresponding PROM is shown in the following figure. Here, 3 to 8 decoder generates eight min terms. The two programmable OR gates have the access of all these min terms. But, only the required min terms are programmed in order to produce the respective Boolean functions by each OR gate. The advantage of PAL is that we can generate only the required product terms of Boolean function instead of generating all the min terms by using programmable AND gates. The block diagram of PAL is shown in the following figure. Here, the inputs of AND gates are programmable. That means each AND gate has both normal and complemented inputs of variables. So, based on the requirement, we can program any of those inputs. So, we can generate only the required product terms by using these AND gates. Here, the inputs of OR gates are not of programmable type. So, the number of inputs to each OR gate will be of fixed type. Hence, apply those required product terms to each OR gate as inputs. Therefore, the outputs of PAL will be in the form of sum of products form. There are two product terms present in each Boolean function. The corresponding PAL is shown in the following figure. The programmable AND gates have the access of both normal and complemented inputs of variables. So, program only the required literals in order to generate one product term by each AND gate. Here, the inputs of OR gates are of fixed type. So, the necessary product terms are connected to inputs of each OR gate. So that the OR gates produce the respective Boolean functions. Hence, it is the most flexible PLD. The block diagram of PLA is shown in the following figure. Here, the inputs of OR gates are also programmable. So, we can program any number of required product terms, since all the outputs of AND gates are applied as inputs to each OR gate. The corresponding PLA is shown in the following figure. All these product terms are available at the inputs of each programmable OR gate. But, only program the required product terms in order to produce the respective Boolean functions by each OR gate.

Chapter 5 : Programmable Logic - Wikibooks, open books for an open world

A Programmable Logic Controller, or PLC, is a ruggedized computer used for industrial automation. These controllers can automate a specific process, machine function, or even an entire production line.

Chapter 6 : Programmable logic device - Wikipedia

Marketresearchpro has launched a report on the Programmable Logic Control Systems Market which embraces all-inclusive information on Programmable Logic Control Systems Market based on historical data analysis and crucial future projection for the duration of

Chapter 7 : What is a Programmable Logic Controller (PLC)? (with picture)

Technology Global Programmable Logic Controller (Plc) Market Maxim, Rockwell (A-B), Idec, B&R Industrial, Siemens, IPM, Bosch Rexroth Global Programmable Logic Controller (Plc) Market report highlights the current market size and future potential of the market at the global and regional level with the help of industry trends and market performance.

Chapter 8 : Global Programmable Logic Control Systems Market Research Report

The Field-Programmable Gate Array (FPGA) is a general-purpose semiconductor device containing a large number of digital logic building blocks.

Chapter 9 : Industrial Control Links

A programmable logic controller (PLC) is an industrial solid-state computer that monitors inputs and outputs, and makes logic-based decisions for automated processes or machines.