Siemens TI 305 | Siemens TI 405 | Siemens TI 505

And other related componentry

Distributed Control Systems

For Industrial Automation

Siemens - Texas Instruments

Product PDF

Presented by DCScenter.com

For Product Needs

Email: sales@DCScenter.com

Call: 800-793-0630

SIEMENS

SIMATIC TI505

Programming Reference

User Manual

Order Number: PPX:505-8104-5
Manual Assembly Number: 2586546-0051
Fifth Edition

Safety-Related Guidelines

This manual contains the following notices intended to ensure personal safety, as well as to protect the products and connected equipment against damage.

⚠ DANGER

DANGER indicates an imminently hazardous situation that, if not avoided, will result in death or serious injury.

DANGER is limited to the most extreme situations.

⚠ WARNING

WARNING indicates a potentially hazardous situation that, if not avoided, could result in death or serious injury, and/or property damage.

⚠ CAUTION

CAUTION indicates a potentially hazardous situation that, if not avoided, could result in minor or moderate injury, and/or damage to property.

CAUTION is also used for property-damage-only accidents.

MANUAL PUBLICATION HISTORY

SIMATIC TI505 Programming Reference Manual
Order Manual Number: PPX:505–8104–5

Refer to this history in all correspondence and/or discussion about this manual.

Event	Date	Description
Original Issue	12/89	Original Issue (2592436–0001)
Second Edition	03/90	Second Edition (2592436–0002)
Errata Package	05/90	Corrections to Chapters 2, 11, and 13 (2592435–0001)
Third Edition	04/92	Third Edition (2592436–0003)
Fourth Edition	05/93	Fourth Edition (2592436–0004)
Fifth Edition	02/95	Fifth Edition (2592436–0005)

LIST OF EFFECTIVE PAGES

Pages	Description	Pages	Description
Cover/Copyright	Fifth		
History/Effective Pages	Fifth		
iii — xxxvi	Fifth		
1-1 — 1-27	Fifth		
2-1 — 2-7	Fifth		
3-1 — 3-16	Fifth		
4-1 — 4-11	Fifth		
5-1 — 5-43	Fifth		
6-1 — 6-185	Fifth		
7-1 — 7-74	Fifth		
8-1 — 8-15	Fifth		
9-1 — 9-33	Fifth		
A-1 — A-13	Fifth		
B-1 — B-6	Fifth		
C-1 — C-27	Fifth		
D-1 — D-4	Fifth		
E-1 — E-43	Fifth		
F-1 — F-1	Fifth		
G-1 — G-36	Fifth		
H-1 — H-17	Fifth		
Index-1 — Index-10	Fifth		
Registration	Fifth		

Contents

Chapter 2 Data Representation

Chapter 3 I/O Concepts

Chapter 4 Controller Memory

Chapter 5 Programming Concepts

Chapter 6 RLL Instruction Set

Appendix G Status Words

List of Figures

List of Tables

Preface

Introduction

The *SIMATIC® TI505™ Programming Reference Manual* contains the information that you need to design an application program for any of these Series 505™ and Series 500™ programmable controllers:

- SIMATIC® TI525™ /TI535™
- SIMATIC® TI520C™ /TI530C™ /TI530T™
- SIMATIC® TI545™
- SIMATIC® TI555™
- SIMATIC® TI560™ /TI565™ /TI560T™ /TI565P™
- SIMATIC® TI575™

This manual describes the complete instruction set for the complete line of SIMATIC TI500/TI505 controllers. Your controller will not support every feature or instruction described, but it will support all instructions common to the Series 505 and Series 500 families and those particular instructions or features identified by your controller model number.

Additionally, this manual assumes that the controller referenced is at the current firmware release at the time of publication, as listed in Table 1. If your controller is not at the current release, an instruction or feature described in this manual may not be supported. If your controller is at a newer firmware release level, the Release Notes included with your controller or firmware upgrade kit may document new features not covered in this manual.

Table 1 SIMATIC TI500/TI505 Controller Firmware Release Levels

Controller	Release	Controller	Release	Controller	Release
TI545-1101	2.1	TI560	3.2*	TI530T	1.6
TI545-1102	3.0	TI565	3.3	TI525	2.2
TI555	3.0	TI575	3.0	TI535	1.1
TI560T	6.0	TI520C	2.6		
TI565P	2.0	TI530C	2.6		

*TI560 Release 3.2 and earlier does not support features new to Rel. 6.0, listed on page xxxiv.

NOTE: Earlier model controllers (as listed in Table A-4 in Appendix A) have certain restrictions on the memory locations to which they can read and write. Refer to Table A-5 for the memory locations that are valid in each field of an instruction when designing an RLL program for these controllers.

New Features

The following new features are described in this edition of this manual:

- Password protection for application-specific memory areas.
- XSUB VMEbus error bit.
- Text Box, which allows user-supplied text to be stored in L-Memory.
- STW223 through STW225, which represent binary time of day.
- STW226, which provides time of day status.
- STW227 and STW228, which provide the 32-bit VMEbus access address if a VMEbus access error occurs.
- STW229 and STW230, which provide the U Memory offset of the instruction that caused a VMEbus access error.

Refer to the Release Notes included with your controller or upgrade package to determine if your controller model supports these new features.

How to Use This Manual

The RLL instructions that can be used with any of the Series 505/500 controllers are noted by the following tab in the upper left or right corner of the page near the instruction mnemonic.

Series 500
Series 505

The RLL instructions that can be used with specific controllers are noted by controller model, as shown in the example below.

Series 500: TI520, TI530, TI560, TI565
Series 505: TI525, TI535, TI545, TI555, TI575

To help you in your program design tasks, you will find the following additional information in the appendices: Status Words for all controller models and performance data for the TI545, TI555, and TI575.

This manual is not intended to be a primer on RLL or SF programming. If you are not familiar with the techniques of RLL programming or of loop dynamics, you should refer to other documentation or call your Siemens Industrial Automation, Inc., distributor or Sales Office for technical assistance. Training classes in RLL and Special Function programming are available at a number of locations. Contact your distributor for more information. Because there are references to various hardware components, you should review the appropriate hardware and installation manuals for your controller as you design your programs.

Programming Software

To program the controller with the latest features, you need an IBM® PC/AT® compatible personal computer with TISOFT™ Programmable Logic Controller Programming Software (Release 5.0 or later) to enter your RLL, loop, analog alarm, and Special Function programs.

Manual Contents

Topics are listed below by chapter.

- Chapter 1 gives an overview of the components of the Series 505 and Series 500 systems, local, distributed, and remote I/O, the concept of I/O numbering and the hardware/software interface.
- Chapter 2 describes the formats used to represent data types.
- Chapter 3 describes how I/O is read and updated.
- Chapter 4 describes the various controller memory types.
- Chapter 5 presents programming concepts.
- Chapter 6 describes the RLL and box instructions.
- Chapter 7 describes the Special Function Program statements.
- Chapter 8 describes analog alarm programming.
- Chapter 9 describes loop programming.
- Appendix A lists all the variables used by Series 505/500 controllers.
- Appendix B lists the RLL instructions, the amount of memory each requires, and instruction numbering guidelines.
- Appendix C gives information needed to calculate controller program scan times.
- Appendix D provides the formats for the loop and analog alarm flags.
- Appendix E gives application examples for selected RLL instructions.
- Appendix F lists the Special Function Program error codes.
- Appendix G lists the status words supported by the Series 505/500 controllers.
- Appendix H describes how to design an external subroutine, and includes an example subroutine.

Chapter 1

Series 505/500 System Overview

1.1 The TI545 and TI555 Systems

System Components

The programmable controller interacts with your equipment through input/output (I/O) modules that relay information between the equipment and the programmable controller. When you design your program, you need to know the physical and logical configuration of these I/O modules, how your equipment is connected to them, and how they are addressed and accessed. The relationships among the system components of the TI545 and the TI555 systems are illustrated in Figure 1-1. For details about hardware components and installation, refer to the *SIMATIC TI545 System Manual* or the *SIMATIC TI555 System Manual*.

Local and Remote I/O

I/O modules are grouped into local and remote I/O categories depending upon their physical location. The local I/O comprises those modules located in the same base assembly as the programmable controller. The base containing the local I/O is numbered 0. Only Series 505 I/O modules can be installed in the local base.

You can connect up to 15 additional base assemblies to the system, numbered 1–15. The I/O modules in these bases make up the remote I/O as shown in Figure 1-1. Both Series 505 and Series 500 I/O can be connected to a TI545 or TI555 controller as remote I/O.

Individual I/O modules in the remote bases communicate with the controller through Remote Base Controllers (RBC). The RBC in each remote base transmits all information from the I/O modules in that base directly to the controller. The TI545/TI555 remote I/O consists of one channel. A channel comprises up to 15 remote bases.

Assigning I/O Point Numbers

You must assign the I/O point and slot numbers from the I/O Configuration Chart on your programming device. The programmable controller does not update discrete or word I/O points in non-configured I/O modules. Refer to your TISOFT user manual for instructions about configuring the I/O.

For the TI545, a maximum of 2048 I/O points can be assigned. Of these, up to 1024 can be analog or word points, which must be numbered 1–1024. The next 1024 points are discrete only. Up to 32,768 control relays are available.

For the TI555, a maximum of 8192 I/O points can be assigned in any mix of discrete and word I/O. Up to 32,768 control relays are available.

You do not need to assign I/O point numbers consecutively. For example, in a remote system, Base 2 can be assigned I/O points 897–960. If a base is configured and the modules in the base do not match the configuration, the programmable controller logs a non-fatal error. Misconfigured modules are not accessed by your program. Inputs are read as 0; outputs are ignored.

A Special Function Module is divided into the I/O portion and the special function portion. When a Special Function Module is inserted into a TI545 or TI555 system, the special function portion of the module is automatically logged in, and can send data to and receive data from the controller.

NOTE: You must configure the I/O portion so that the controller updates the I/O points. Non-special function modules are not logged in automatically.

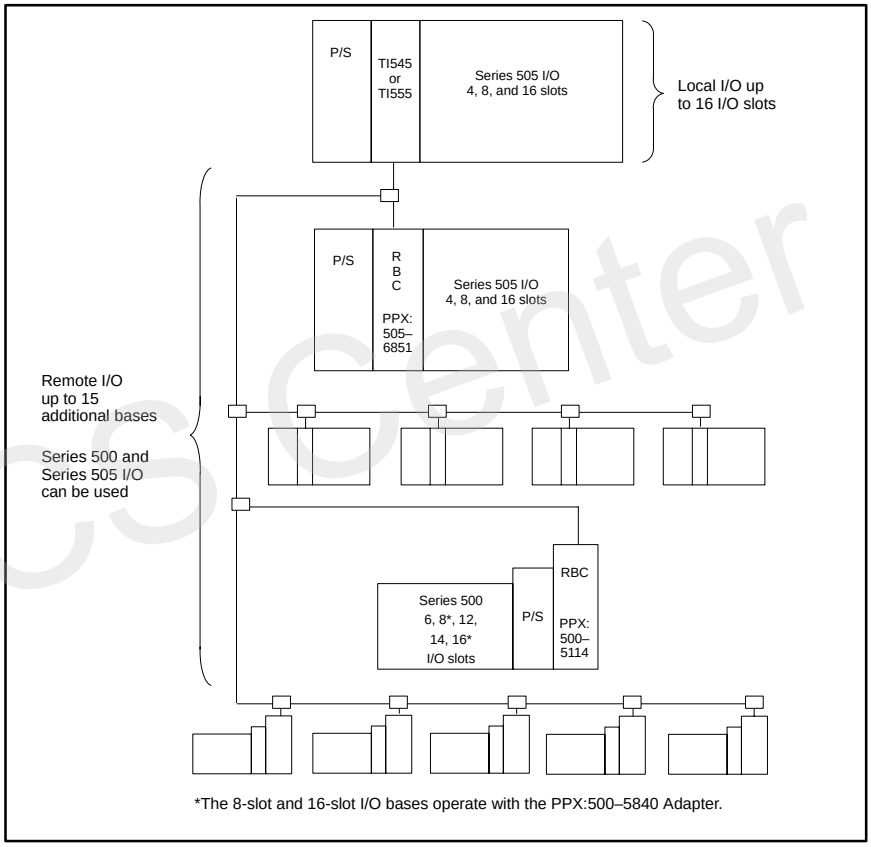


Figure 1-1 Components for the TI545 / TI555 System

Program Execution

The TI545 and TI555 controllers execute four scan operations during the programmable controller scan.

- Interrupt RLL execution*
- Cyclic RLL execution
- Discrete scan
- Analog task processing

Interrupt RLL Execution

The interrupt I/O feature allows you to program a specific response which executes immediately in response to a field input transition (interrupt request) from your application. Interrupt I/O operation requires the use of at least one Interrupt Input Module (PPX:505-4317) installed in the local base. See Section 3.4 for more information on interrupt I/O operation.

Cyclic RLL Execution

A cyclic RLL program consists of a section of ladder logic, usually short for quick cycle times, that runs independently of the main RLL program. Cyclic RLL is executed periodically throughout the entire programmable controller scan, interrupting the discrete scan and the analog scan as necessary. Because the execution of a cyclic RLL task is not synchronized with the I/O update, use the immediate I/O instructions to access the I/O.

Discrete Scan

The discrete scan consists of three primary tasks that are executed sequentially (Figure 1-2) and at a rate that can be user-specified.

Normal I/O Update. During the normal I/O cycle update, the programmable controller writes data from the image registers to the outputs, and stores data from the inputs into the image registers. The length of the I/O update cycle is dependent upon the number of bases and types of modules (analog, discrete, or intelligent). All I/O points are fully updated each scan.

Main Ladder Logic Cycle. The programmable controller executes the main RLL task.

Special Function Module Communication. Communication with special function (SF) modules, e.g., NIM, BASIC, PEERLINK™, etc., consists of the following actions.

- Service requests from a previous scan for which processing has been completed are transmitted to the SF modules.
- Remote bases are polled for initial SF module service requests.
- Remote base communication ports are polled for service requests.
- Service requests from SF modules and remote base communication ports are processed.

*Interrupt RLL operation available on TI555 Release 1.1 or greater.

Each SF module that requires service increases the scan time, depending upon the type of module and task. Each type of module is allowed a certain number of service requests per scan. Once these are completed, this function is terminated. Some service requests can be deferred, and these are processed during the analog task time slice described below.

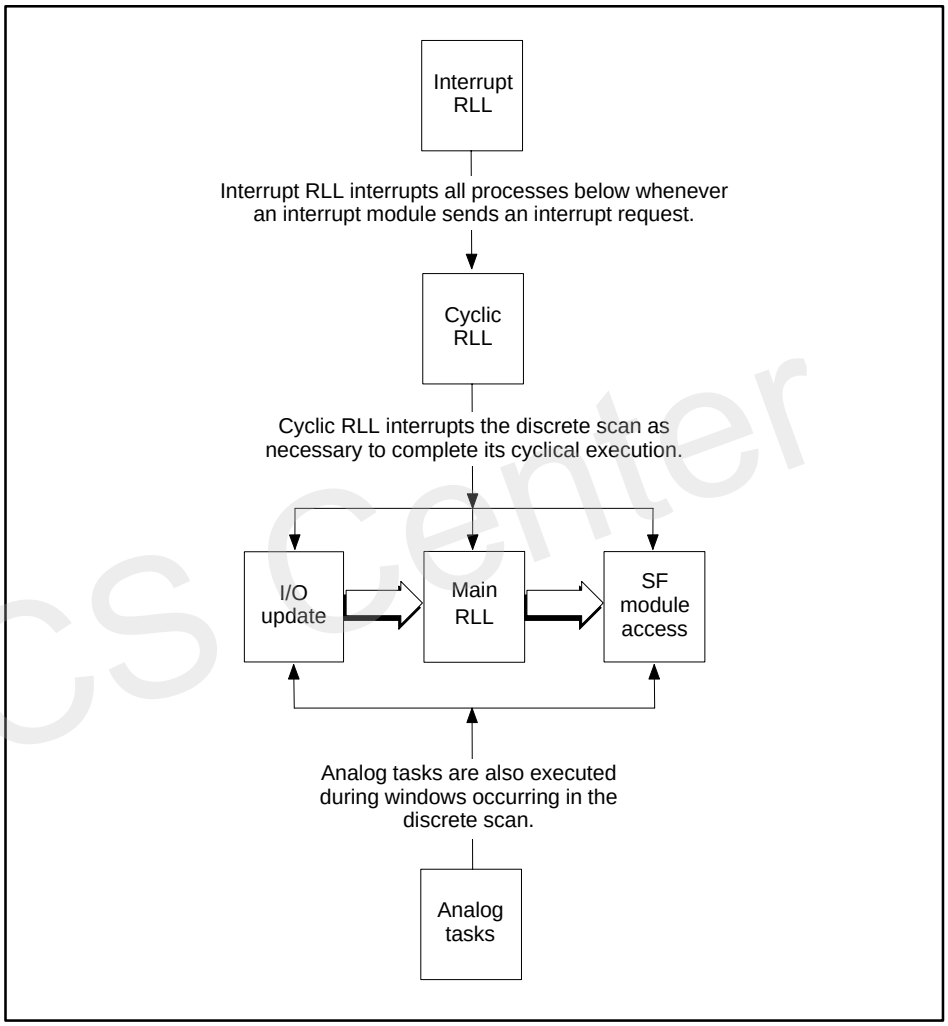


Figure 1-2 Discrete Scan Sequence for the TI545 and TI555 Controllers

Analog Task Processing

The analog portion of the scan is composed of five general types of tasks (Figure 1-3), which are cyclical or non-cyclical in their execution.

Analog tasks are guaranteed execution once per scan, following the discrete scan, provided there is processing to be done. Analog tasks are also processed during windows of suspended activity that occur during the normal I/O and SF portions of the scan. RLL execution is not interrupted by analog tasks.

You can adjust the amount of time spent per controller scan for all analog tasks, except diagnostics, with a programming unit and using AUX Function 19. The time allocation for a given analog task is referred to as its time slice.

Cyclic Analog Tasks

The following types of processes are executed cyclically. Each has a sample rate which determines how often it is executed.

- Loops
- Analog alarms
- Cyclic SF programs

The programmable controller has an analog task that executes each type of cyclic process. When enabled, each cyclic process is placed in the execution queue that is managed by the analog task responsible for executing that type of process.

The cyclic processes are time-ordered in their individual queues according to when each process is rescheduled for execution, relative to the other cyclic processes within the same queues. The process with the highest priority (closest to overrunning) is executed first. The process is executed until it is completed or until the time specified for that particular task's time slice expires. If the executing process is completed before the time slice expires, the process with the next highest priority is executed. If the time slice expires before the process is completed, the process (and the task) is put on hold in its current position.

The programmable controller then advances to the next analog task. When the programmable controller sequences through its operations and returns to an analog task with a cyclic process on hold, the process resumes execution from the hold point, unless a higher priority process was scheduled since the last respective time slice. If a process in a cyclic time slice is not finished executing when it is scheduled to execute again, an overrun flag is set.

Restricted SF programs, which are called by loops or analog alarms, are executed from within the loop or analog alarm tasks. Therefore, their execution time is included within the loop or analog alarm time slice.

SF subroutines, which are called by SF programs or other SF subroutines, are processed during the calling program's time slice.

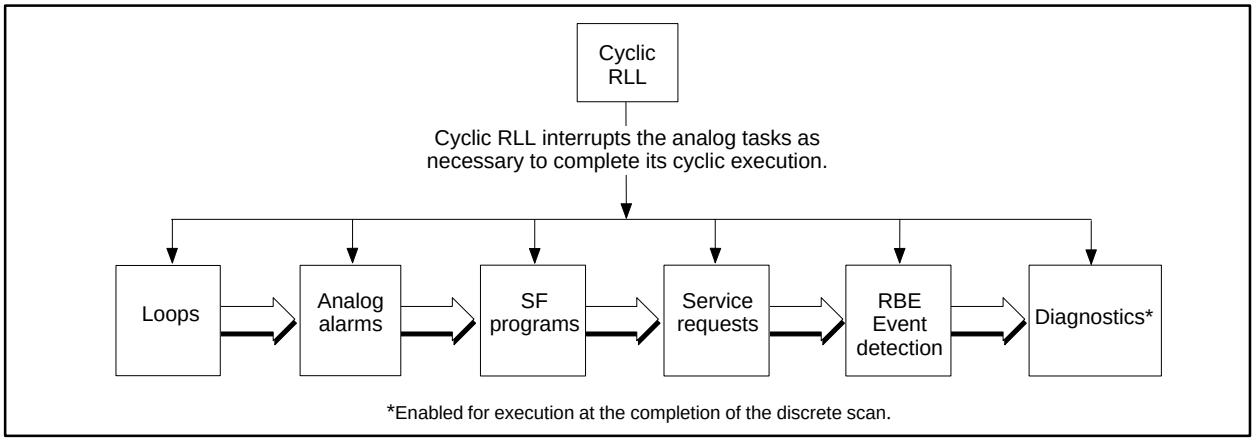


Figure 1-3 Analog Task Scan Sequence for the TI545 and TI555 Controllers

Non-cyclic Analog Tasks

The following types of processes are executed non-cyclically:

- Priority/Non-priority SF programs.
- RLL-requested SF subroutines.
- Service request messages.
- Report by Exception (RBE) event detection.
- Run-time diagnostics.

Priority and Non-Priority SF Programs are non-cyclic processes that are queued when the SFPGM RLL box instruction receives power flow. There is an analog task that executes priority SF programs, and another analog task that executes non-priority SF programs. These processes are executed in the order that they are queued in the appropriate task's execution queue. When the programmable controller completes one of these processes, it removes the process from the respective queue and turns on the SFPGM output. There are no overrun flags associated with these processes.

RLL-requested SF Subroutines are queued into one of two SFSUB queues when the SFSUB RLL box instruction receives power flow. One queue handles SFSUB 0 instructions and the other handles all other SFSUB instructions.

Service Requests received from the communication ports are placed on one of two communications queues. Read and write commands are placed on the priority communication queue for fastest response. Commands that may require several scans to complete, e.g., program edits and the TISOFT **FIND** function, are placed in a non-priority communications queue.

Report By Exception event detection task only executes when the programmable controller is used with SIMATIC® PCS™, Release 3.0 or later. The RBE event detection task monitors PCS-defined process events and notifies the PCS when an event is detected.

Run-time Diagnostics are enabled for execution at the completion of the discrete scan. The time slice for diagnostics is 1 ms and cannot be changed.

Setting the Scan

The TI545/TI555 scan is defined as the time between normal I/O updates. You can set the scan for the controller as follows.

- Fixed — The programmable controller starts a new discrete scan at the specified time interval. The controller executes the discrete scan once and then cycles to the analog scan portion, executing the analog tasks at least one time. If the analog tasks are completed within the specified time, the controller goes into a loop mode (processing analog tasks or idling) until time to start the next scan.

  A scan overrun status bit is set (bit 14 in Status Word 1) if the total execution time for the discrete scan portion and the first execution of the analog scan portion exceeds the fixed scan time.

- Variable — The programmable controller executes all tasks once and then starts a new scan. All discrete and analog tasks are guaranteed one execution per scan. Specify variable scan for the fastest possible execution of the discrete scan.

- Variable with upper limit — The programmable controller executes the discrete scan once and then executes the analog tasks. The controller remains in the analog portion of the scan as long as there are analog tasks to be done. When the upper time limit expires, or no analog tasks require processing, a new scan is begun.

  The analog scan portion is executed at least one time. A scan overrun status bit is set if the total execution time for the discrete scan portion and the first execution of the analog scan portion exceeds the upper limit.

Cycle time for the cyclic RLL can be a fixed value or a user-specified variable. As a variable, the cycle time can be changed by logic in your application program. If the cyclic RLL completes execution in less than the specified cycle time, execution does not resume until the next cycle begins. The programmable controller scan time is extended by the amount of time to execute the cyclic RLL multiplied by the number of times the cyclic RLL is executed during the programmable controller scan.

The timing relationship of the scan operations is shown in Figure 1-4. Refer to the Appendix C for details about how to configure the time slices.

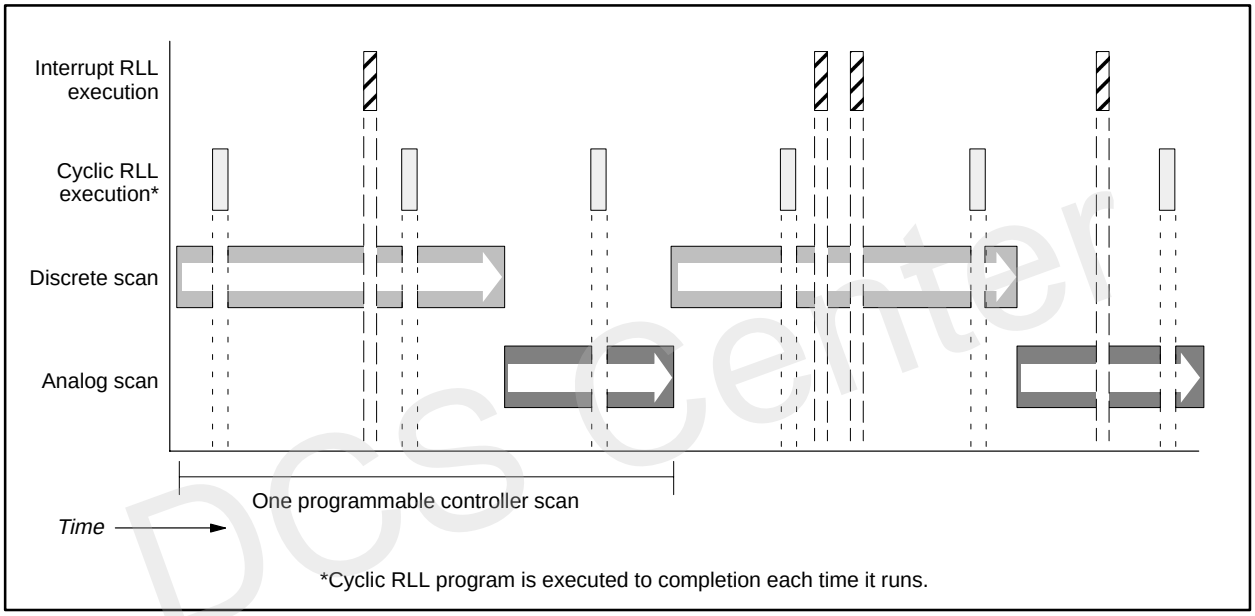


Figure 1-4 Timing Relationship of the TI545 / TI555 Controller Scan Operations

1.2 The TI560/TI565 System

TI560/TI565 System Components

The programmable controller interacts with your equipment through input/output (I/O) modules that relay information between the equipment and the programmable controller. When you design your program, you need to know the physical and logical configuration of these I/O modules, how your equipment is connected to them, and how they are addressed and accessed. The relationships among the system components of the TI560/TI565 controllers are illustrated in Figure 1-5. For details about hardware components and installation, refer to the *SIMATIC TI560/TI565 System Manual*.

TI560/TI565 Remote I/O

The TI560/TI565 chassis holds the main CPU (TI560), the Special Function CPU (TI565), the Remote Channel Controllers (RCC), memory expansion cards, and the Hot Backup Unit.

The I/O modules are housed in I/O base assemblies. An I/O base assembly has slots for a remote base controller (RBC), a power supply, and the I/O modules. Individual I/O modules in the remote bases communicate with the programmable controller through the base controllers. The RBC in each remote base assembly transmits all information from the I/O modules in that base assembly to the RCC.

The RCC serves as the master device in servicing the I/O points. The RCC requests and receives I/O updates from an RBC over the remote link. Each RCC board contains two channels capable of controlling 1024 I/O points per channel. A maximum of eight channels (four RCCs) may be used. Up to 16 base assemblies may be connected to a channel.

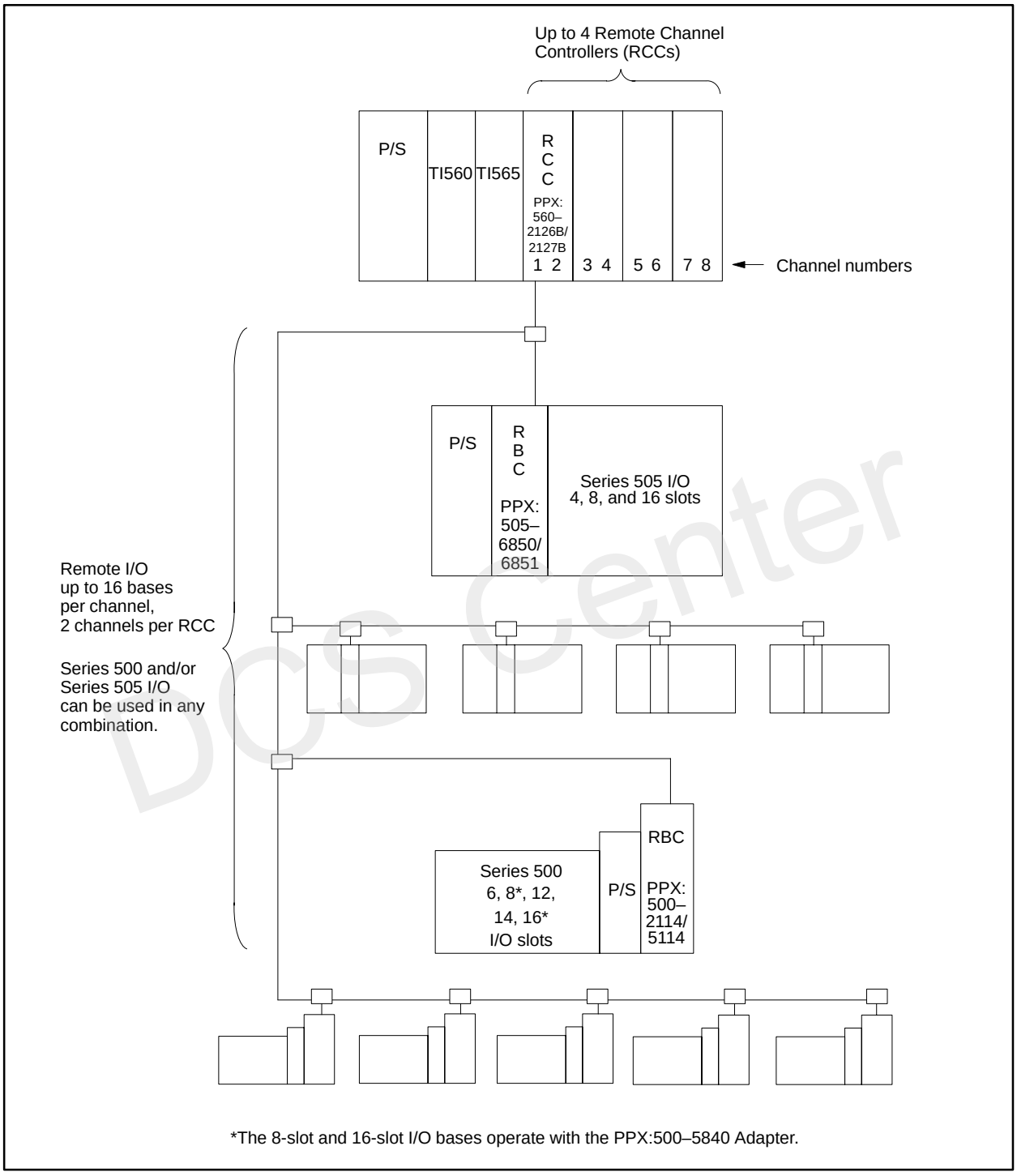


Figure 1-5 Components for the TI560/TI565 System

Assigning I/O Point Numbers

You must assign the I/O point and slot numbers from the I/O Configuration Chart on your programming device. The programmable controller does not update discrete or word I/O points in non-configured I/O modules. Refer to your TISOFT user manual for instructions about configuring the I/O. I/O addresses are restricted to the range of points available in each channel. Each channel has the address range shown in Table 1-1.

A Special Function Module is divided into the I/O portion and the special function portion. When a Special Function Module is inserted into a TI560/TI565 system, the special function portion of the module is automatically logged in, and can send and receive data from the controller.

NOTE: You must configure the I/O portion so that the programmable controller updates the I/O points. Non-special function modules are not logged in automatically.

The channels on the RCC boards are read and assigned channel numbers in their defined ranges according to where the RCC board is located in the chassis. The RCC board in the slot closest to the TI560 CPU is assigned Channels 1 and 2; the next RCC, Channels 3 and 4, etc. The assignment of I/O identifiers (Xs, Ys, WXs, WYs) to physical points on the bases is limited only to a block of contiguous points required for the particular module, and the points must be within the address range for the particular channel.

Table 1-1 Remote I/O Channel Address Range

Channel	Address Range	Channel	Address Range
1	1–1024	5	4097–5120
2	1025–2048	6	5121–6144
3	2049–3072	7	6145–7168
4	3073–4096	8	7169–8192

TI560 Scan Operation

The TI560 controller requires approximately 8.0 milliseconds (Rel. 1.x) or 11 ms (Rel. 2.0 and later) for overhead tasks. This time is distributed throughout each scan, illustrated in Figure 1-6.

I/O Update. The RCC cards simultaneously write data from the image registers to the outputs, and update the image registers with data from the inputs. The length of the I/O update cycle corresponds to the RCC that requires the longest update time. This is primarily dependent upon the number of bases and types of modules on each channel. Each RCC has two channels which update in parallel. All I/O points are fully updated each scan.

Ladder Logic Cycle. Upon completion of the I/O update, the main CPU starts the execution of the RLL program. While the program is being executed, the RCCs run background tasks: polling for unconfigured bases and servicing operator interfaces connected at the I/O bases if that RBC has requested service. For NIM 4.0 releases, SF module read requests may be processed during RLL execution. The main CPU executes RLL programs in 2.2 milliseconds (Rel. 3.0 or earlier) or 1.5 ms (Rel. 5.0 or later) per k words of program instructions. The entire program is fully executed each scan.

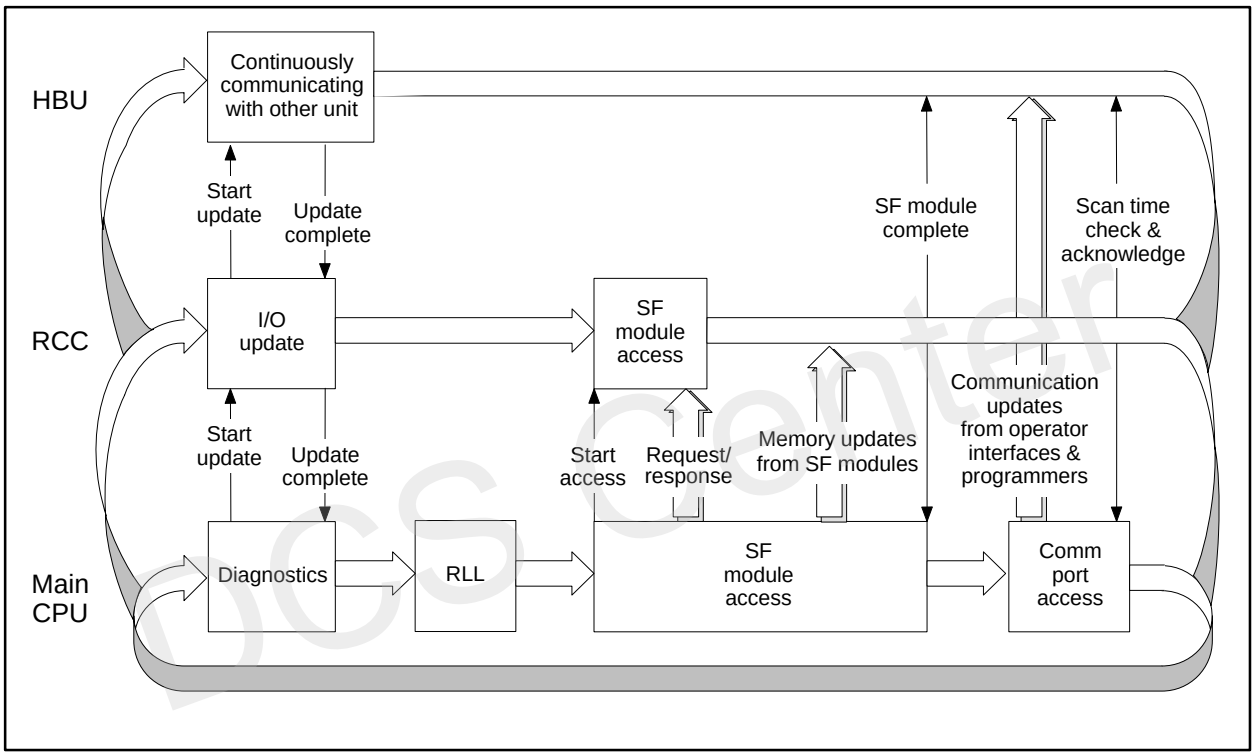


Figure 1-6 Scan Sequence for the TI560 Controller

Special Function Module Communication. Upon completion of the RLL scan, the Special Function (SF) module communications begin. The Main CPU executes task codes, which were gathered from the SF modules by the RCC, and the resulting information is made available to the RCCs for transfer back to the modules.

Each SF module that requires service during this period adds scan time according to the type of module and the type of task. Each type of module is allowed a certain number of task code requests, block transfers, or store-and-forward operations per scan. Once these are completed, the SF cycle is terminated by the RCC.

Communication Port Service. The TI560 services communication port requests from the two (local) ports on the main CPU and the (remote) ports on all the RBCs. The amount of time spent executing the port communication requests depends on the scan type — variable or fixed. For a variable scan, the port communication cycle is suspended as soon as either one of the following conditions is true.

- All requests have been serviced, or
- At least 6 ms have been spent executing requests.

For a fixed scan, the port communication cycle is suspended as soon as the next scan is scheduled to start, and either of the above conditions is true.

Execution of communication requests is time-shared between the ports on a turn-by-turn basis. A request from local port 1 receives 2 ms, if needed, and then a request from local port 2 receives 2 ms, if needed. Then, a request from any remote port receives 2 ms, if needed. This is repeated until execution is suspended for either one of the reasons listed above.

Hot Backup Unit Communications. During all of the above periods, the HBU transmits messages between the TI560 or TI565 systems (with standby unit on line). The HBU adds approximately 9.0 ms for TI560 operation and approximately 1 to 4 ms additional for TI565 updates.

TI565 CPU Functions

The TI565 CPU executes loops, analog alarms, and special function (SF) programs throughout the programmable controller timeline. During the I/O cycle, the TI565 CPU continues to run until it needs to read an I/O point. Once the I/O cycle completes, the TI565 CPU resumes running.

The TI565 can do 32 loop calculations, update 16 analog alarms, and execute up to 1200 additional floating point calculations as called from SF programs in 1 second. This assumes the scan time is equal to, or greater than, 50 milliseconds to allow the TI565 to complete tasks without having to process interrupts from the TI560.

NOTE: The PPX:565–2820 can execute loops, analog alarms, and special function programs approximately three times faster than the rates given above. Actual execution times are not available at time of publication.

A scheduler prioritizes the analog alarm, loop, and cyclic SF program tasks executed by the processor. If there are no tasks of a particular type to be executed, the processor does not idle, but spends the minimum time necessary and then advances to the the next task type. For example, if no analog alarms are programmed, the processor begins executing SF programs after processing loop calculations.

TI565 CPU Communications. The Main (TI560) CPU is idle during TI565 CPU communications and gives the TI565 CPU full access to the bus. This time period typically is less than 1.0 millisecond, but is dependent on the number of SF programs being called from ladder logic and other requests queued for processing by the TI565 CPU. Figure 1-7 shows the relationships of the processors' operations.

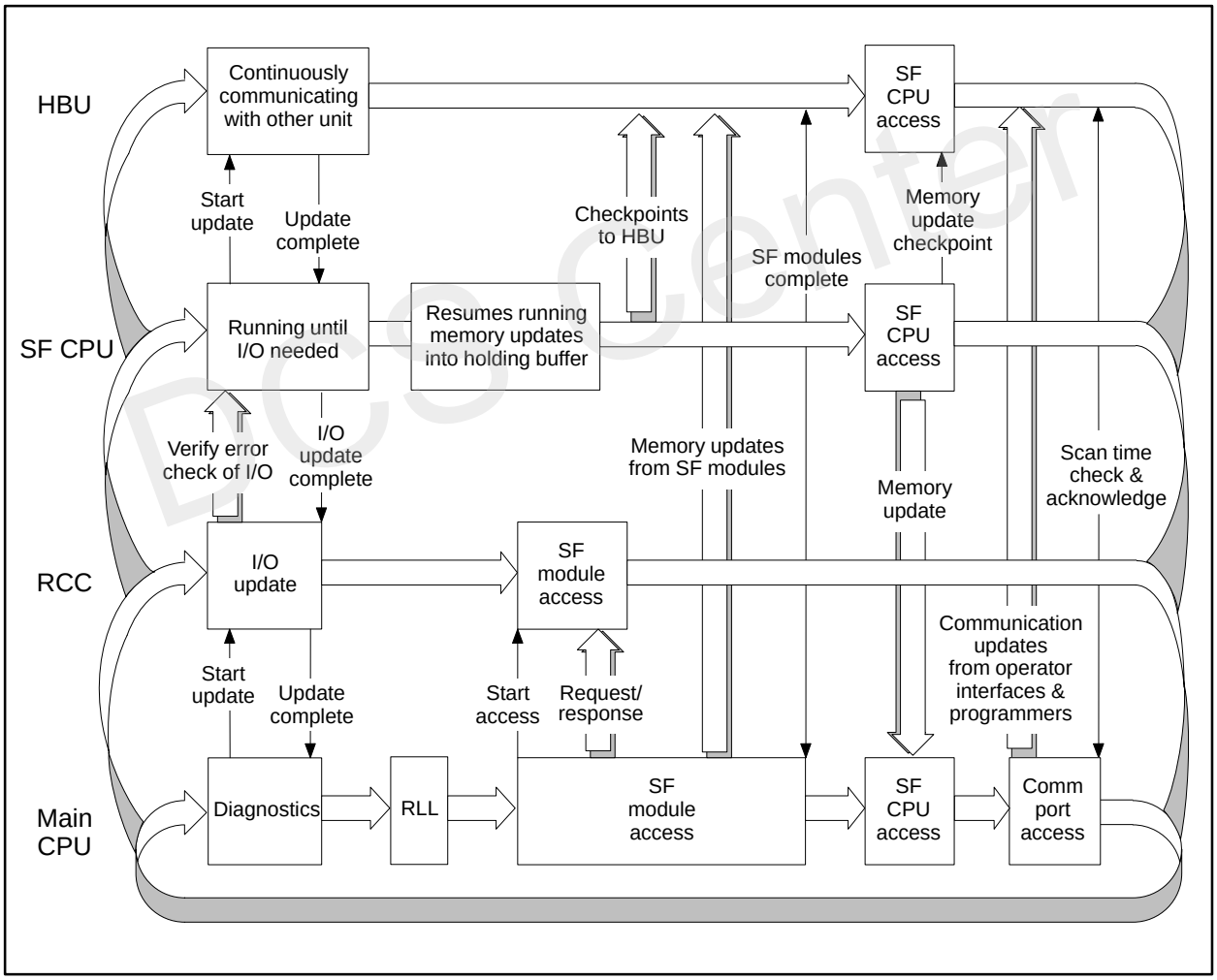


Figure 1-7 Scan Sequence for the TI565 CPU

1.3 The TI575 System

TI575 System Components

The TI575 Control System provides a means by which various control products can communicate over a VMEbus backplane. The TI575 system is a scaleable/flexible control system that can accept multiple TI575 Central Processing Unit (CPU) cards.

The TI575 programmable controller system interacts with your equipment through input/output (I/O) modules that relay information between the equipment and the CPU. When you design your program, you need to know the physical and logical configuration of these I/O modules, how your equipment is connected to them, and how they are addressed and accessed. The relationships among the system components of the TI575 System are illustrated in Figure 1-8. For details about the hardware components and installation, refer to the *SIMATIC TI575 System Manual*.

TI575 Local and Remote I/O

I/O modules are grouped into local and remote I/O categories depending upon their physical location. The local I/O comprises those modules located in the same base assembly as the CPU.

When you install the optional remote I/O annex card (PPX:575–2126), the TI575 can communicate with Series 505 and Series 500 I/O. You can connect up to 15 Series 505/500 base assemblies to each CPU. The I/O modules in these bases make up the remote I/O as shown in Figure 1-8.

Individual I/O modules in the remote bases communicate with the TI575 through remote base controllers (RBC). The RBC in each remote base transmits all information from the I/O modules in that base directly to the CPU. The TI575 CPU remote I/O consists of one channel. A channel on the TI575 CPU comprises up to 8192 I/O points.

Both Series 505 and Series 500 I/O can be connected to a TI575 CPU as remote I/O. The TI575 controller is capable of addressing directly the PPX:505-6851 RBC in a Series 505 base assembly, or the PPX:500-5114 RBC in a Series 500 base assembly. The I/O numbering scheme for remote I/O is identical to that of the TI545. For a discussion of the Series 505/500 remote I/O numbering, refer to Section 1.1. For information about communicating with VME-compatible I/O, refer to the user documentation for the TI575 system.

TI575 Scan Operation

The TI575 scan operation is identical to that of the TI545 and TI555 controllers. Refer to Section 1.1 for a detailed discussion of the scan functions.

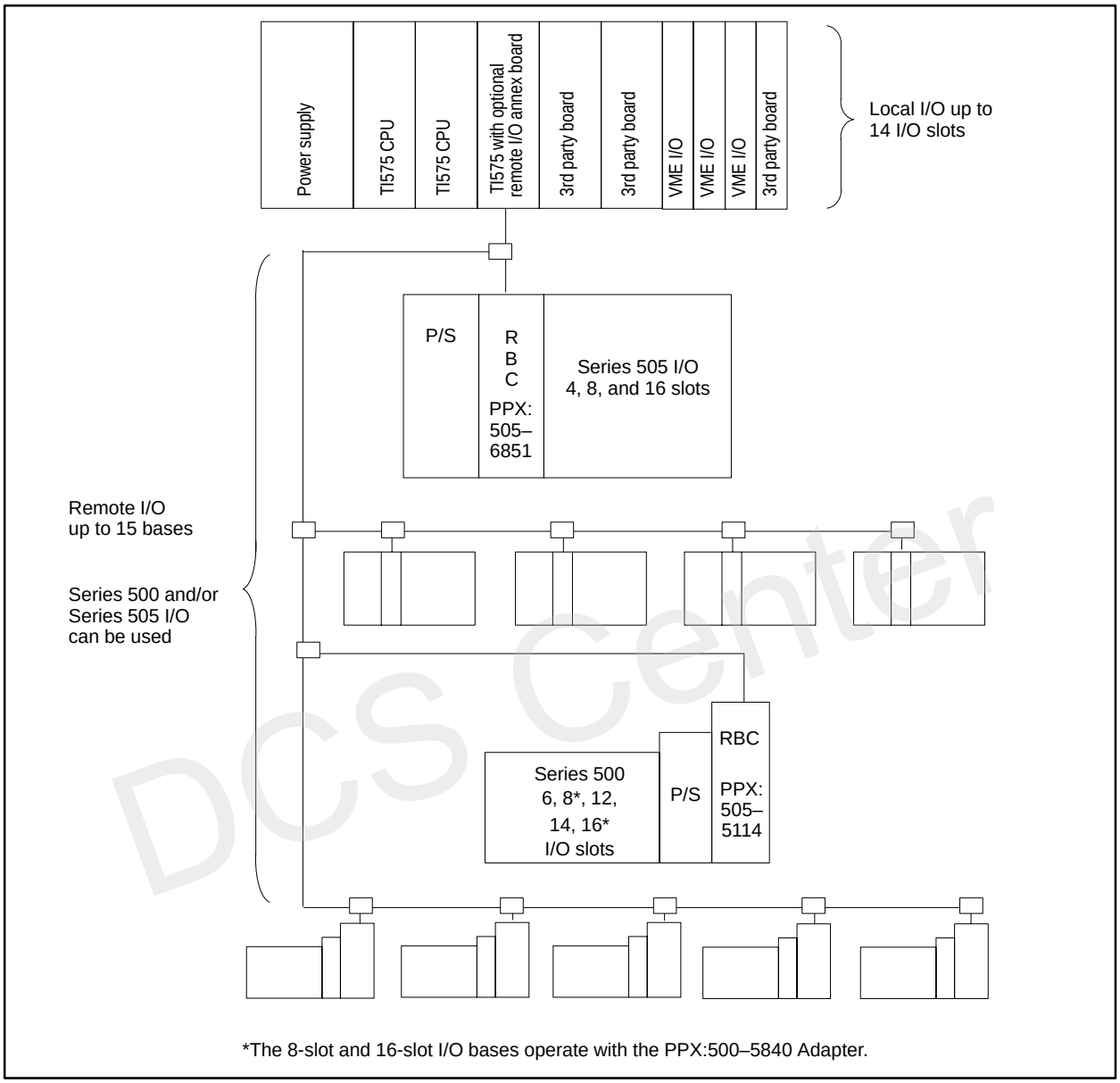


Figure 1-8 Components for the TI575 System

1.4 The TI525/TI535 Systems

System Components

The programmable controller interacts with your equipment through input/output (I/O) modules that relay information between the equipment and the programmable controller. When you design your program, you need to know the physical and logical configuration of these I/O modules, how your equipment is connected to them, and how they are addressed and accessed. The relationships among the system components of the TI525/TI535 controllers are illustrated in Figure 1-9. For details about hardware components and installation, refer to the hardware manual for your system.

Local and Distributed I/O

I/O modules are grouped into local or distributed I/O categories depending upon their physical location. The local I/O comprises those modules located in the same base assembly as the programmable controller. The local I/O includes I/O modules in up to 2 logical (8-slot) bases numbered 0 and 1.

You can connect up to 14 additional logical bases to the system, numbered 2–15. The I/O modules in these bases make up the distributed I/O as shown in Figure 1-9. Except for the PPX:525–1102, the TI525 and TI535 controllers support both local and distributed I/O. Both Series 500 and 505 I/O can be connected to a TI525/TI535 controller as distributed I/O.

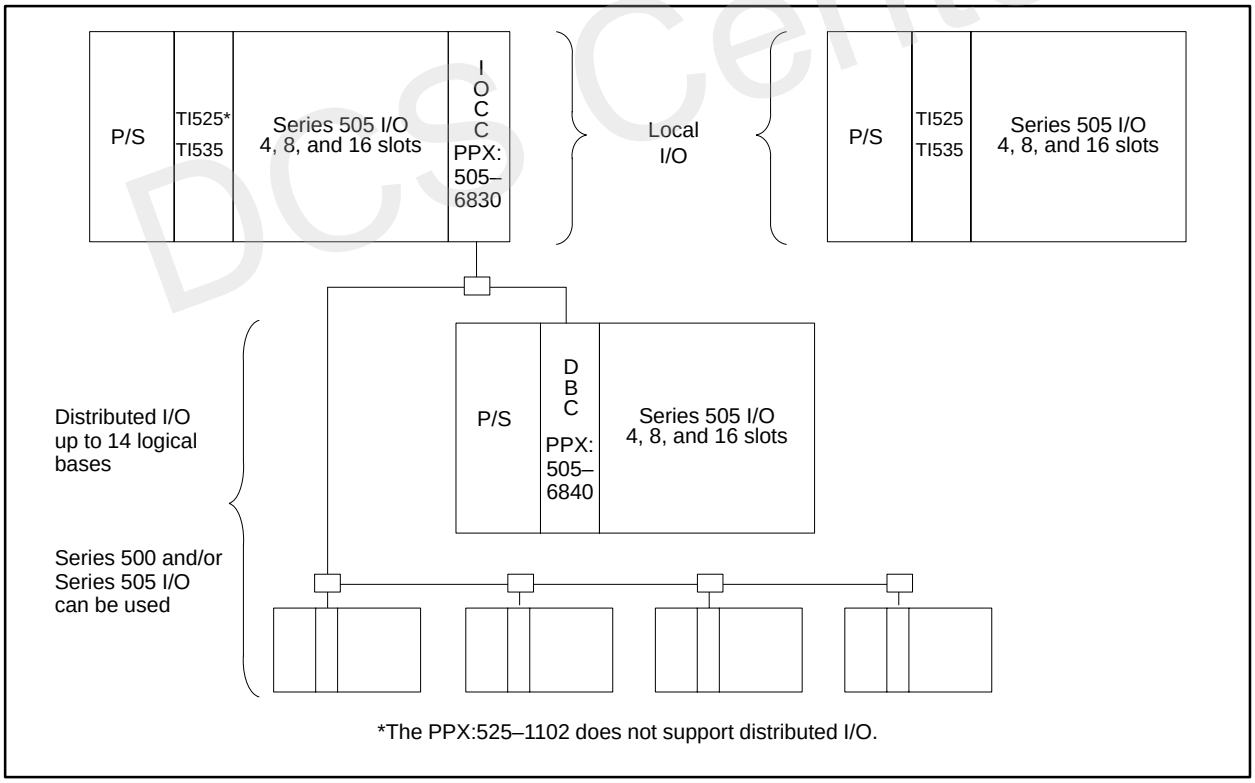


Figure 1-9 Components for the TI525 and TI535 Systems

Series 505 Logical Base

For a Series 505 system, a base assembly is composed of one or two logical bases. A logical base is defined as a group of eight I/O slots, as shown in Figure 1-10. Therefore, the 16-slot base assembly contains two 8-slot logical bases, while the 8-slot base assembly contains one logical base. The four-slot PPX:505–6504 base also contains one logical base, but slots 5–8 do not physically exist.

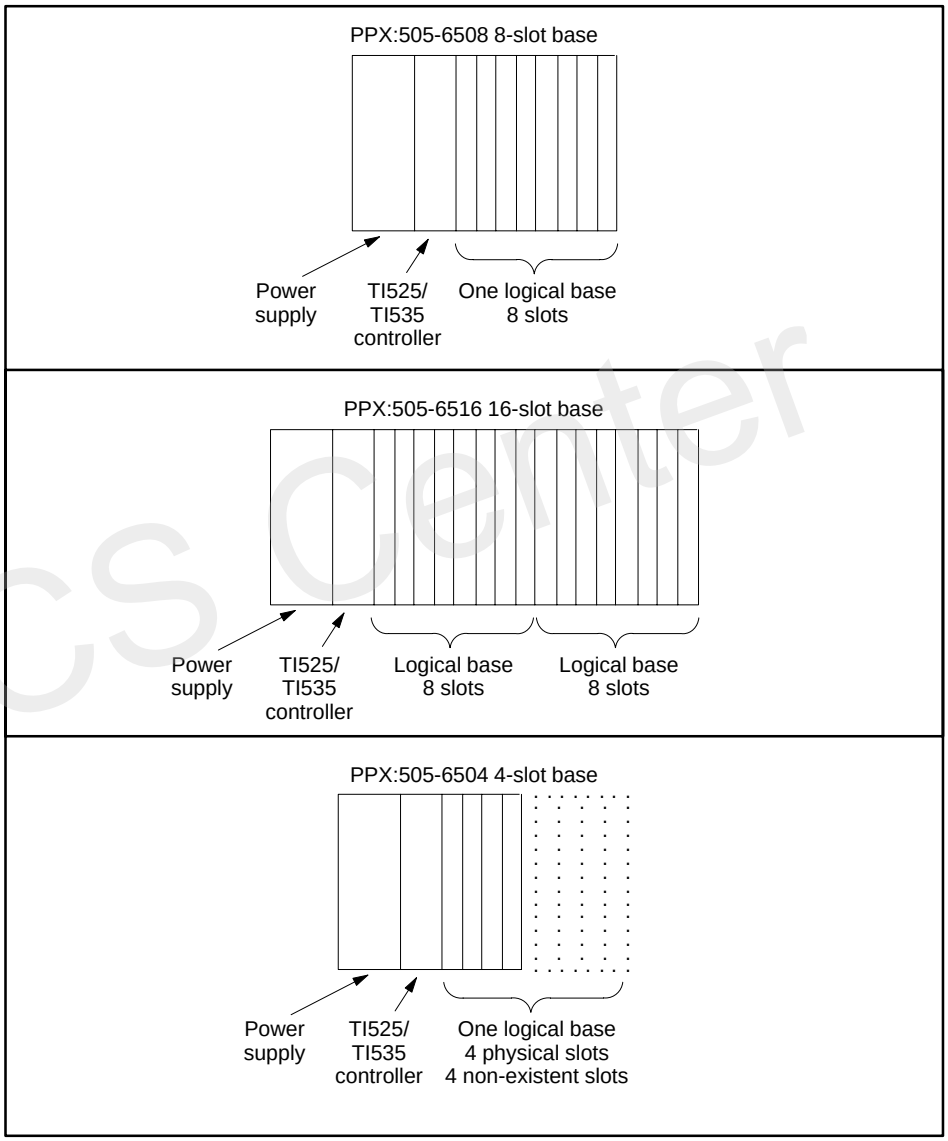


Figure 1-10 Definition of Series 505 Logical Base

Assigning I/O Point Numbers

You must assign the I/O point and slot numbers from the I/O Configuration Chart on your programming device. The TI525/TI535 controller does not update discrete or word I/O points in non-configured I/O modules. Refer to your TISOFT user manual for instructions about configuring the I/O.

A Special Function Module is divided into the I/O portion and the special function portion. When a Special Function Module is inserted into a TI525/TI535 system, the special function portion of the module is automatically logged, and can send and receive data from the TI525/TI535.

NOTE: You must configure the I/O portion so that the programmable controller updates the I/O points. Non-special function modules are not logged in automatically.

You can configure local I/O to a maximum of 512 I/O points. You can obtain this with base PPX:505–6516 (16 slots) and 32-point I/O modules.

The programmable controller permits a maximum number of 1023 points for a distributed system. You do not need to assign the point numbers consecutively. For example, in a distributed system, Base 2 can be assigned I/O points 897–960, but the highest number that you can assign to any point is 1023. After you have entered an I/O configuration, the programmable controller logs a non-fatal error when a module(s) is present but not configured. You cannot use that module until it has been configured.

You do not need to assign I/O point numbers to empty slots or to non-existent slots in logical bases that have fewer than eight slots.

Scan Operation

The functions executed by the TI525/TI535 controllers are described below and illustrated in Figure 1-11.

I/O Update. During the I/O cycle update the programmable controller writes data from the image registers to the outputs, and stores data from the inputs into the image registers. The length of the I/O update cycle depends upon the number of bases and types of modules (analog, discrete or intelligent). All I/O points are fully updated each scan.

Ladder Logic Cycle. The programmable controller executes the relay ladder logic program. The entire RLL program is executed each scan.

Special Function Module Communication. The programmable controller executes task codes that were gathered from the Special Function (SF) modules. The resulting information is transferred back to the SF modules. Each SF module that requires service adds scan time; how much depends on the type of module and the type of task. Each type of module is allowed a certain number of task code requests, block transfers, or store-and-forward operations per scan. Once these are completed, the SF cycle is completed.

Communication Port Service. The programmable controller executes tasks received through the communication ports. A minimum of two milliseconds is allotted for this function. If the scan time has been fixed, then the time remaining after all other functions are finished is devoted to task processing. When the allotted time has expired, task processing is continued on the subsequent scan.

Diagnostics. The programmable controller executes self-diagnostics at the end of each scan.

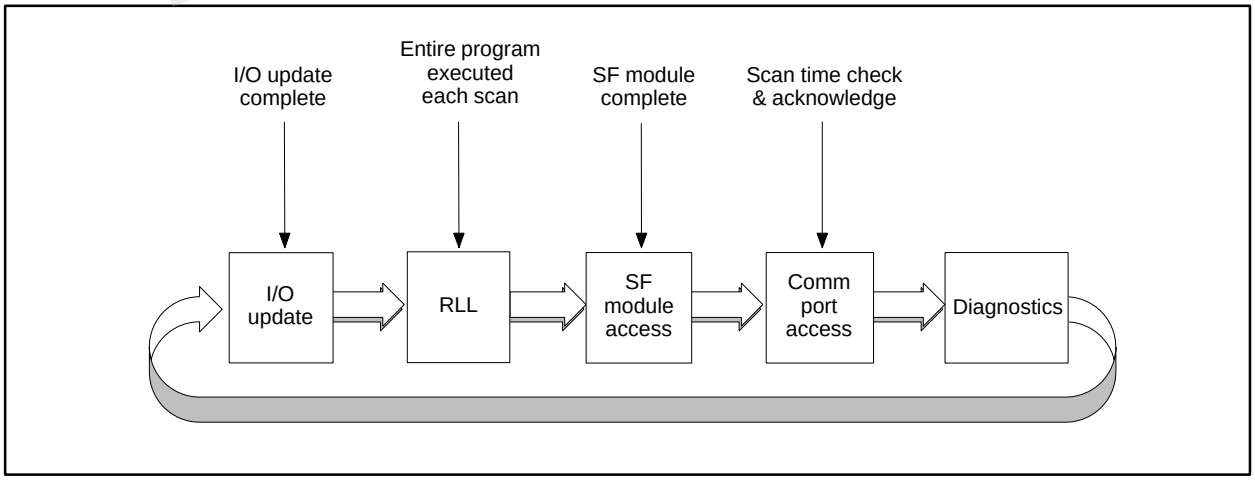


Figure 1-11 Scan Sequence for the TI525/TI535 Controllers

1.5 The TI520C/TI530C/TI530T Systems

System Components

The programmable controller interacts with your equipment through input/output (I/O) modules that relay information between the equipment and the programmable controller. When you design your program, you need to know the physical and logical configuration of these I/O modules, how your equipment is connected to them, and how they are addressed and accessed. The relationships among the system components of the TI520C/TI530C/TI530T controllers are shown in Figure 1-12. For details about hardware and installation, refer to the hardware manual for your system.

Local and Distributed I/O

I/O modules are grouped into local or distributed I/O categories, based on their physical location. The local I/O comprises those modules located in the same base assembly as the programmable controller. The local I/O includes I/O modules in up to two logical (eight-slot) bases numbered 0 and 1.

You can connect up to 14 additional logical bases to the system, numbered 2 to 15. The I/O modules in these bases make up the distributed I/O as shown in Figure 1-12. The TI530C and the TI530T support both local and distributed I/O. Both Series 505 and Series 500 I/O can be connected to a TI530C/TI530T controller as distributed I/O.

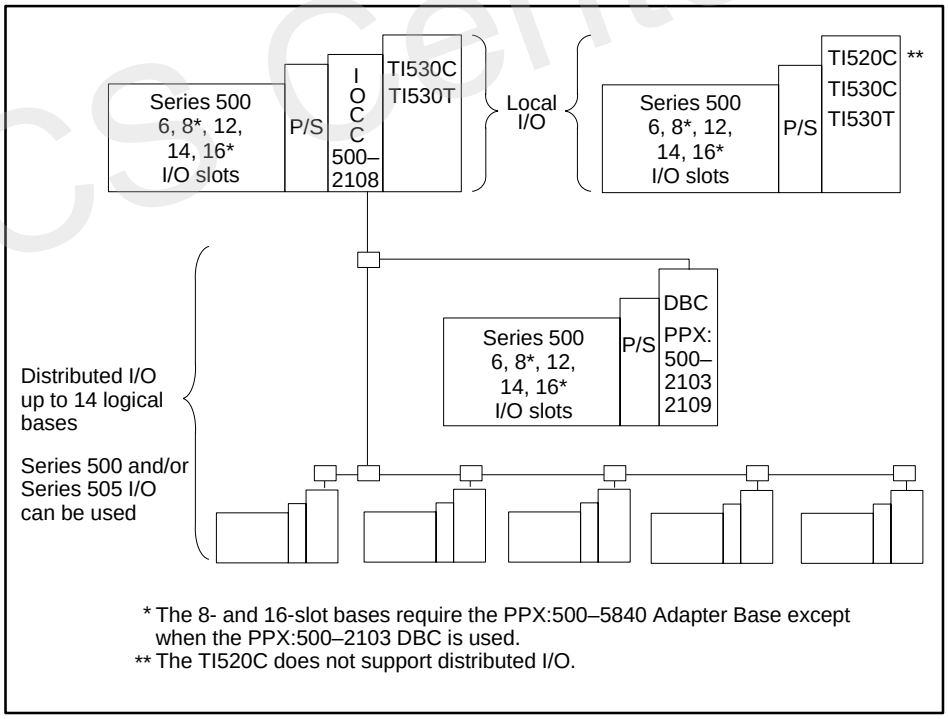


Figure 1-12 Components for the TI520C, TI530C, or TI530T Systems

Series 500 Logical Base

For a Series 500 system, a base assembly is composed of one or two logical bases. A logical base is defined as a group of eight I/O slots, as shown in Figure 1-13. Therefore, the 16-slot base assembly has two 8-slot logical bases. The 14-slot base assembly model also has two logical bases, but slots 7–8 on the second base do not physically exist.

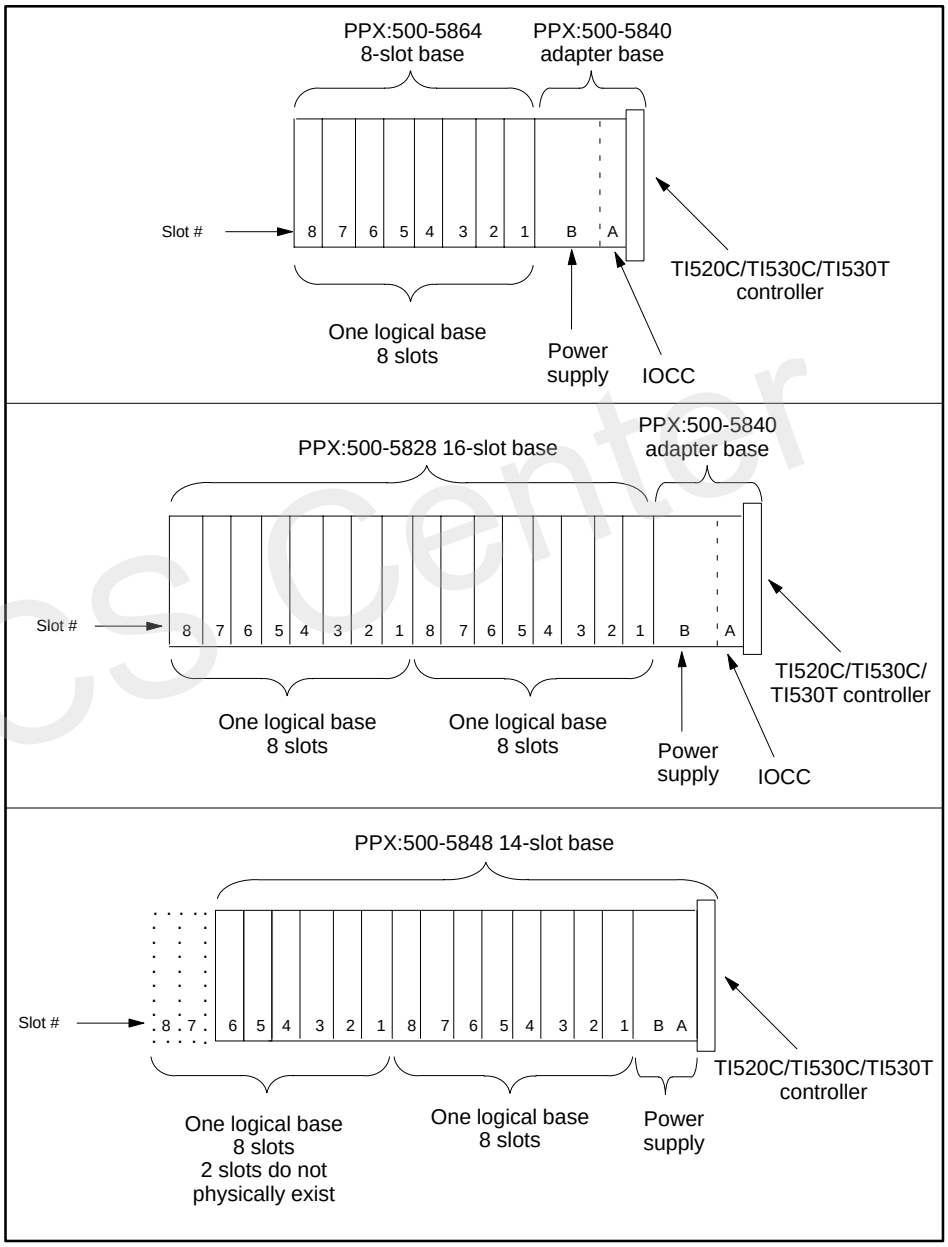


Figure 1-13 Definition of Series 500 Logical Base

Assigning I/O Point Numbers

If you intend to use any modules with more than eight I/O points per physical I/O slot, you must assign the I/O point numbers yourself when you configure the I/O numbering. You do this from the I/O Configuration Chart on your programming device. Refer to your TISOFT user manual for instructions on how to configure the I/O.

You can configure the local I/O to a maximum of 512* I/O points when you assign the I/O point numbers. This is obtained by using base PPX:500–5828 (16 slots) with the PPX:500–5840 adapter base and 32-point I/O modules.

The programmable controller permits a maximum number of 1023 points for a distributed system. You do not need to assign the point numbers consecutively. For example, in a distributed system, Base 2 can be assigned I/O points 897–960, but the highest number that you can assign to any point is 1023. After you have entered an I/O configuration, the programmable controller logs a non-fatal error when a module(s) is present but not configured. You cannot use that module until it has been configured.

You do not need to assign I/O point numbers to empty slots or to non-existent slots in logical bases that have fewer than eight slots.

*I/O Module power consumption requirements may reduce the actual number of I/O points that can be used.

Using Default I/O Numbers

If you do not configure the I/O, the TI520C, TI530C, and TI530T controllers automatically log in all eight-point modules. Modules having greater than eight points are not logged in; however, their presence in the I/O base is indicated when you execute a Read Base function with your programming unit.

When the programmable controller logs in the modules and configures the I/O automatically, the module points are assigned numbers according to the slot and the base in which the module is located. The point number assignments can be determined by referring to Figure 1-14. You can calculate the module starting address by using the following equation:

$$\text{Starting Address} = 1 + \left[\left[(\text{Logical Base \#}) \times 64\right] + \left[(\text{Slot \#} - 1) \times 8\right]\right]$$

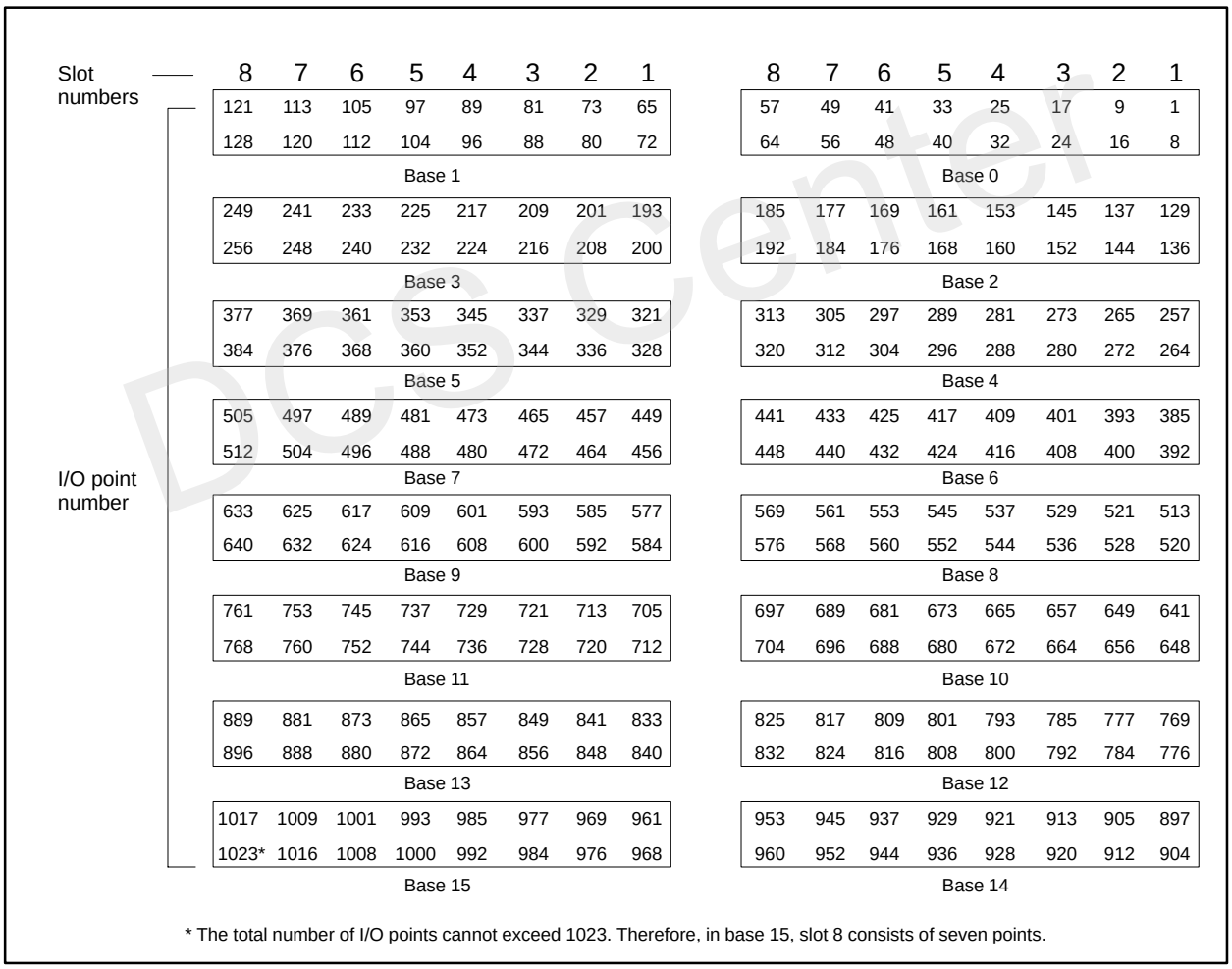


Figure 1-14 Series 500 I/O Default Numbering

Using Default Numbers with 6-, 12-, 14-Slot Bases

The 6-, 12-, and 14-slot I/O base assemblies hold at least 1 logical base with fewer than 8 slots. Because the default numbering is configured on multiples of the 8-slot logical base, I/O point numbers are assigned to points on non-existent slots. For example, in the PPX:500–5848 base assembly with 14 slots, I/O points 113–128 are assigned to the 2 non-existent slots on the second logical base of this assembly. See Figure 1-15.

If you use the 6- 12- or 14-slot base for the programmable controller and install an IOCC as well, the power supply must be installed in Slot B and I/O slot 1 is not available. This slot is covered by the power supply in this situation and cannot accommodate an I/O module. The I/O point numbers (1–8) assigned this slot by the default numbering cannot be used.

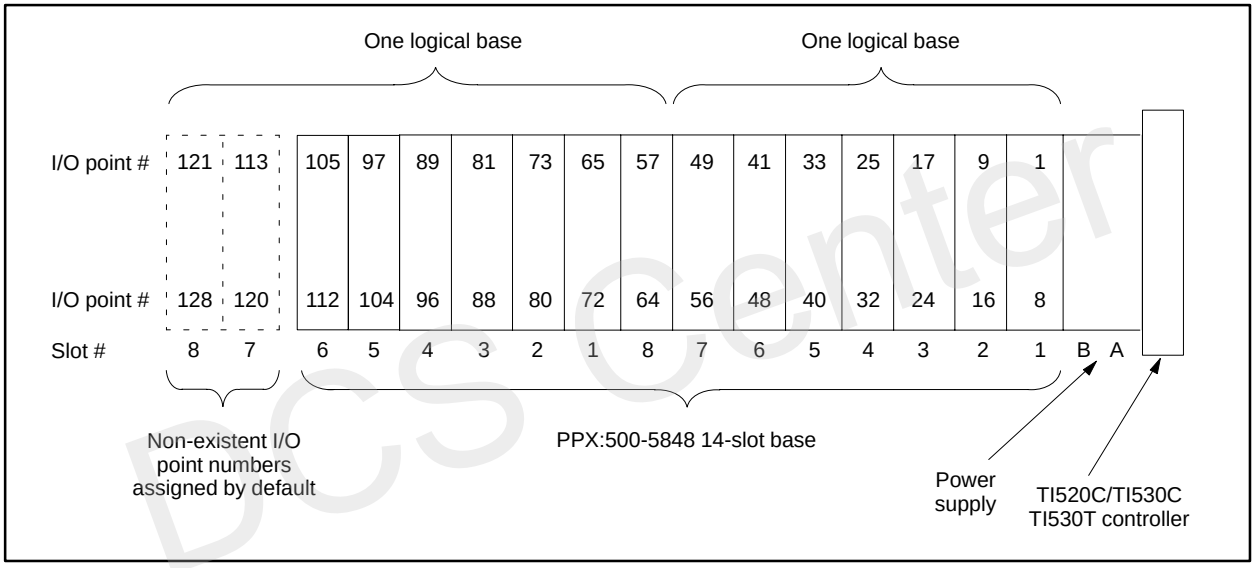


Figure 1-15 Default I/O Point Numbers for 14-Slot Base

Scan Operation

The functions executed by the TI520C/TI530C/TI530T controllers are described below and illustrated in Figure 1-16.

I/O Update. During the I/O cycle update the programmable controller writes data from the image registers to the outputs, and stores data from the inputs into the image registers. The length of the I/O update cycle depends upon the number of bases and types of modules (analog, discrete or intelligent). All I/O points are fully updated each scan.

Ladder Logic Cycle. The programmable controller executes the relay ladder logic program. The entire RLL program is executed each scan.

Special Function Module Communication. The programmable controller executes task codes that were gathered from the Special Function (SF) modules. The resulting information is transferred back to the SF modules. Each SF Module that requires service adds scan time; how much depends on the type of module and the type of task. Each type of module is allowed a certain number of task code requests, block transfers, or store-and-forward operations per scan. Once these are completed, the SF cycle is completed.

Communication Port Service. The programmable controller executes tasks received through the communication ports. A minimum of two milliseconds is allotted for this function. If the scan time has been fixed, then the time remaining after all other functions are finished is devoted to task processing. When the allotted time has expired, task processing is continued on the subsequent scan.

Diagnostics. The programmable controller executes self-diagnostics at the end of each scan.

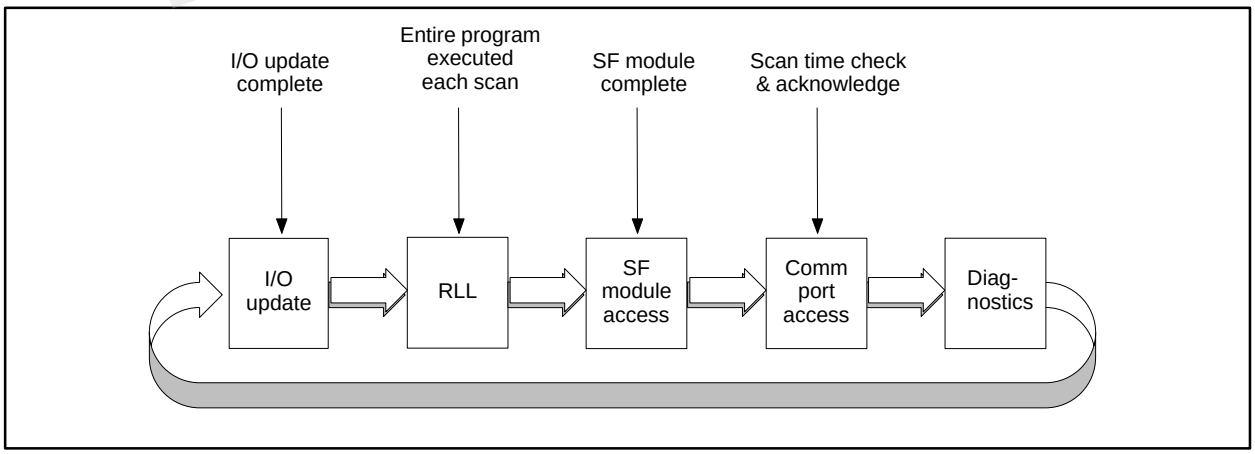


Figure 1-16 Scan Sequence for the TI520C/TI530C/TI530T Controllers

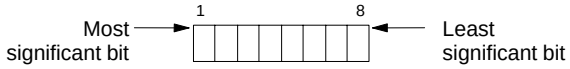
Chapter 2

Data Representation

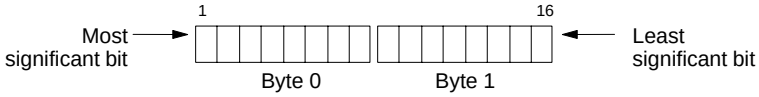
2.1 Definitions

The terms listed below are used throughout this manual and have the following meanings.

Byte. A byte consists of 8 contiguous bits.

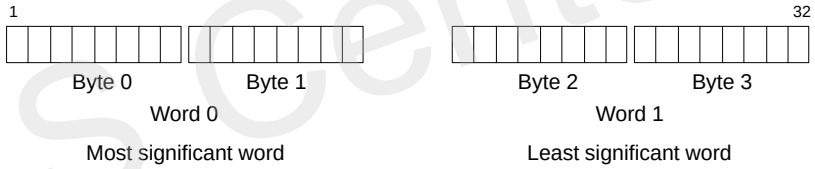


Word. A word consists of 2 contiguous bytes, 16 bits.



For example, the contents of V-Memory word V100 occupy 16 contiguous bits; the word output WY551 occupies 16 contiguous bits.

Long Word. A long word consists of 2 contiguous words, 32 bits, that represent a single value.



For example, the contents of V-Memory long word V693 occupy two contiguous words (32 bits), V693 and V694. The next available address is V695, which can represent a word (16 bits) or another long word (32 bits).

Image Register. The image register is a reserved memory area used to store the value of all discrete (on/off) and word I/O data. Discrete I/O data is contained in the discrete image register. Word I/O data is stored in the word image register. See Section 3.1 for a more complete discussion of the function of the image register.

I/O Point. An I/O point consists of an I/O type and a reference number that represent a location in the image register. An I/O point that represents a discrete bit in the discrete image register is composed of an X or Y I/O type. An I/O point that represents a word in the word image register is composed of a WX or WY I/O type.

2.2 Integers

Signed integers are stored as 16-bit words in the two's complement format as shown in Figure 2-1. The 16-bit format allows you to store values ranging from –32,768 to +32,767 (decimal integer values). When bit 1 (the sign bit) is 0, the number is positive; when bit 1 is 1, the number is negative.

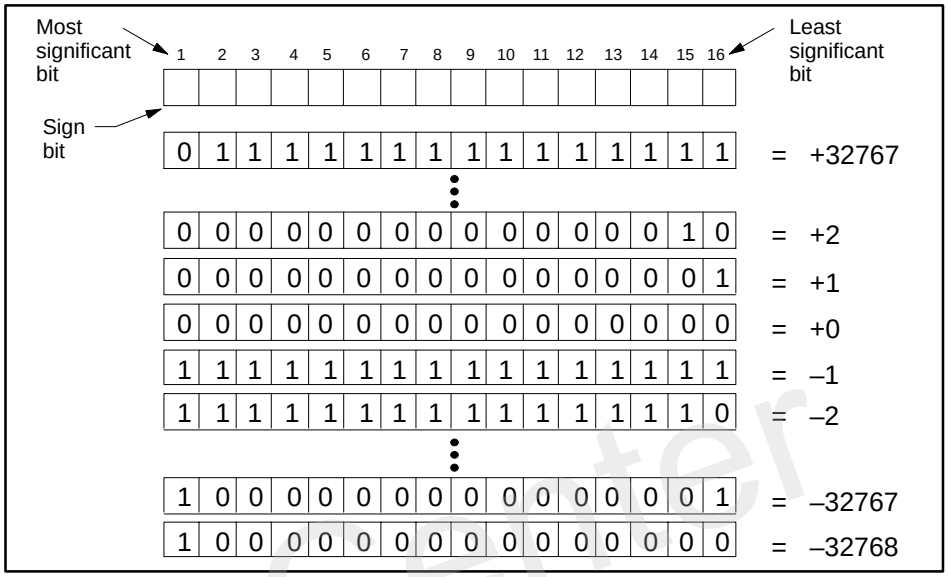


Figure 2-1 Format of Signed Integers

You can display data on your programming unit as an unsigned integer. The 16-bit format allows you to display integer values ranging from 0 to 65535 as shown in Figure 2-2.

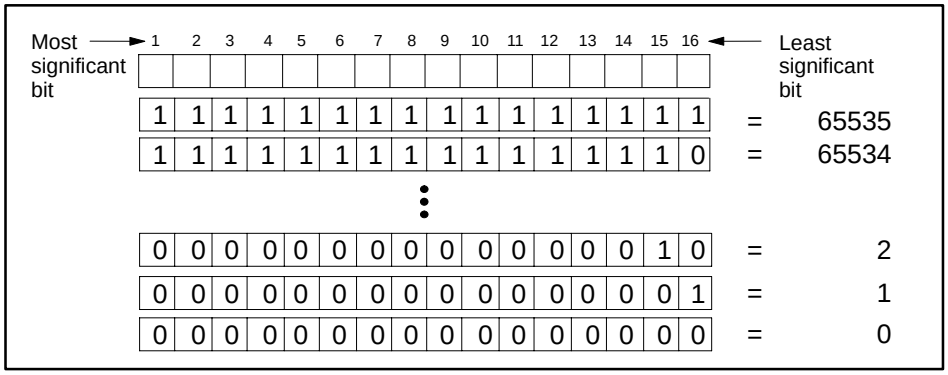


Figure 2-2 Format of Unsigned Integers

Thirty-two bit signed long word integers are stored as 32-bit long words in the two's complement format: