GFK-1192

GE Fanuc Manual Series 90-70

System Manual for Control Software Users

GE Fanuc Automation

Programmable Control Products

Series 90™-70 System Manual for Control Software Users

GFK-1192B *November 1999*

Warnings, Cautions, and Notes as Used in this Publication

Warning

Warning notices are used in this publication to emphasize that hazardous voltages, currents, temperatures, or other conditions that could cause personal injury exist in this equipment or may be associated with its use.

In situations where inattention could cause either personal injury or damage to equipment, a Warning notice is used.

Caution

Caution notices are used where equipment might be damaged if care is not taken.

Note

Notes merely call attention to information that is especially significant to understanding and operating the equipment.

This document is based on information available at the time of its publication. While efforts have been made to be accurate, the information contained herein does not purport to cover all details or variations in hardware or software, nor to provide or every possible contingency in connection with installation, operation, or maintenance. Features may be described herein which are not present in all hardware and software systems. GE Fanuc Automation assumes no obligation of notice to holders of this document with respect to changes subsequently made.

GE Fanuc Automation makes no representation or warranty, expressed, implied, or statutory with respect to, and assumes no responsibility for the accuracy, completeness, sufficiency, or usefulness of the information contained herein. No warranties of merchantability or fitness for purpose shall apply.

The following are trademarks of GE Fanuc Automation North America, Inc.

Alarm Master	GEnet	ProLoop	Series Three
CIMPLICITY	Genius	PROMACRO	VersaMax
PowerTRAC	Genius PowerTRAC	Series Five	VersaPro
CIMPLICITY 90–ADS	Helpmate	Series 90	VuMaster
CIMSTAR	Logicmaster	Series One	Workmaster
Field Control	Modelmaster	Series Six	

Content of This Manual

Chapter 1. **Introduction:** provides an overview of the Series 90™-70 PLC and of the layout of this manual.

Chapter 2. **System Operation:** provides general information about the 90-70 PLC sweep cycle and handling of programs, blocks, and interrupts. This includes a discussion of PLC system sweep sequences, the system power-up and power-down sequences, clocks and timers, and security.

Chapter 3. **Fault Explanation and Correction:** Provides trouble-shooting information for the Series 90-70 PLC. It explains fault descriptions in the PLC fault table and I/O faults.

Appendix A. **CPU Performance Data:** lists memory size in bytes and execution time in microseconds for each programming instruction. It also contains timing information for other tasks that can be used in conjunction with instruction timings to predict CPU sweep times.

Appendix B. **Interpreting Faults:** describes how to interpret the message structure format when reading the fault tables.

Appendix C. **Memory Allocation:** provides a worksheet for determining total number of bytes of user data used and how much is still available for the user program.

Appendix D. **Using Floating-Point Numbers:** describes considerations for using floating-point math operations.

Related Publications

GFK-0262 *Series 90-70 Programmable Controller Installation Manual*

GFK-0646 *C Programmer's Toolkit for Series 90™ PLCs User's Manual*

GFK-1292 *Control User's Manual*

At GE Fanuc Automation, we strive to produce quality technical documentation. After you have used this manual, please take a few moments to complete and return the Reader's Comment Card located on the next page.

Henry Konat
Technical Writer

Contents

Chapter 1 Introduction

The Series 90™-70 PLC is a member of the GE Fanuc Series 90 PLC family of programmable logic controllers (PLCs). It is easy to install and configure, offers advanced programming features, and is designed for compatibility with other PLCs offered in the Series 90 family of PLCs. Through the use of the latest design and manufacturing technology, open architecture VME bus, and the ability to connect to Genius and FIP I/O and several CPU models, the Series 90-70 PLC provides a powerful, cost-effective platform for small applications through the very largest.

Software Architecture

The programming software architecture provides a platform upon which to build structured control programs. Programs may be built from many program blocks, each of which is related to a control function. Structured programs permit parallel development of a complete program as a collection of program blocks developed independently by many different individuals or OEMs. Structured programs are also easier to understand and debug. A control program may be built of many smaller program blocks, each of which can relate to a specific machine function. This approach makes it easier to isolate and associate control logic with machine functions.

The programming language and representation are based on IEC working draft 65A standard. Eventual adoption of this standard will make it easier to create programs that can be understood globally. It establishes the Series 90-70 PLC in the vanguard of the movement toward recognized international standards.

Beginning with Release 6 PLC CPUs, it has been possible to incorporate multiple programs into a folder. All of these programs can be written in C or one program can be an RLD or SFC program with the remaining programs written in C. In addition, Release 6, and later CPUs have built-in debugging capabilities for C programs and external blocks. For more information on this feature, refer to the *C Programmer's Toolkit* manual (GFK-0646C or later).

Terminology Used in This Manual

The following terms are used with their defined meanings throughout this manual:

User program: any user-generated code, that is, an RLD program, an SFC program, or a standalone C program

Block: any RLD block, Parameterized Subroutine Block, or external block—an external block being either a C block or an C function block (CFBK)

Fault Handling

Faults are handled by a software alarm processor function which time-stamps and logs I/O and system faults in two tables (the PLC fault table and the I/O fault table). These tables can be displayed on the programming software screen or uploaded to a host computer or other coprocessor. Application programs can also gain access to the fault information.

Hardware Configuration

Configuration is the process of assigning physical locations, logical addresses, and parameter values to the hardware modules in the system. It may be done either before or after programming; however, it is recommended that configuration be done first.

Using This Manual

This manual is distributed with Control programming software, and describes the PLC hardware and programming features available in the CPU. Refer to the current IPI distributed with Control for CPU and programming features not described in this version of the manual.

Reference information is available in this manual, as described below:

Appendix A, *CPU Performance Data*, lists the memory size in bytes and the execution time in microseconds for each of the programming instructions. Appendix A also contains timing information for other PC tasks which, when used in conjunction with the instruction timings, can be used to predict CPU sweep times.

Appendix B, *Interpreting Faults*, describes how to interpret the message structure format when reading the PLC and I/O fault tables.

Use the worksheet in Appendix C, *Memory Allocation*, to determine the total number of bytes of user data used and how much is still available for the user program.

Refer to Appendix D, *Using Floating-Point Numbers*, for IEEE formats when dealing with floating-point math operations.

Revisions to This Manual

Appropriate changes have been made to this manual to reflect feature changes, corrections, and updates to existing information. The changes are:

- Name of manual changed to reflect that this reference manual is for Control software users
- References made to CPX and CGR model CPUs, where appropriate, throughout the manual
- Bulk Memory Access information added (chapter 2)
- Value for Constant Sweep timer corrected (chapter 2, pg. 2-46)
- Note added after Table 2-18 regarding CPU Mode switch and description of privilege level 1 updated in table
- Description of System Faults updated (chapter 3, pg. 3-2)
- Appendix A, CPU Performance Data, tables revised (all information not available, will be added to a future version)
- Paragraph added , beginning with “Each Ethernet Global . . . . “, page A-23
- Section titled “Relative CPU Test Performance” added at end of Appendix A

Chapter 2 System Operation

This chapter describes certain system operations of the Series 90-70 PLC system. The table displayed below summarizes the content of each section in this chapter.

Section	Title	Description	Page
1	Basic PLC Sweep Summary	Describes the major steps in a typical PLC sweep, including application program task execution, Programmer Communications Window, System Communications Window, and Background Window.	2-2
2	User Reference Data	Describes user reference data, system status/fault references, and data types.	2-11
3	Program Organization	Describes the structure and use of LD blocks, PSB blocks, external blocks, and standalone C programs.	2-24
4	PLC Sweep Modes and Program Scheduling Modes	Explains Normal Sweep, Constant Sweep, Constant Window, Microcycle Sweep, and Stop modes. Also describes Triggered Interrupt blocks/programs and timed interrupts.	2-45
5	Run/Stop Operations	Describes the four modes of operation supported by the 90-70 PLC: Run/Outputs Enabled, Run/Outputs Disabled, Stop/IO Scan, and Stop/No IO Scan.	2-66
6	Power-Up and Power-Down Sequences	Describes the three parts of system power-up (including power-up self-test, PLC operation initialization, and system configuration), the power-down sequence, and the retention of data memory.	2-68
7	Clocks and Timers	Describes the elapsed time clock, time-of-day clock, and watchdog timers.	2-72
8	System Security	Describes protection level request from the programmer, including password assignment and block lock, OEM protection and password, and the write protect keyswitch.	2-74
9	Series 90-70 PLC I/O System	Describes I/O data mapping and diagnostic data.	2-77

Section 1: Basic PLC Sweep Summary

The user program(s) in the Series 90-70 PLC execute in a repetitive fashion until stopped by a command from the programmer or a command from another device. In addition to executing the user program(s), the PLC obtains data from input devices, sends data to output devices, performs internal housekeeping, services the programmer, services other communications, and performs self-tests. The sequence of operations necessary to execute these components one time is called a sweep. This section summarizes the sweep phases; for more detailed information, refer to section 4 of this chapter.

Basic PLC Sweep

There are seven major phases in a typical PLC sweep as shown in the following figure:

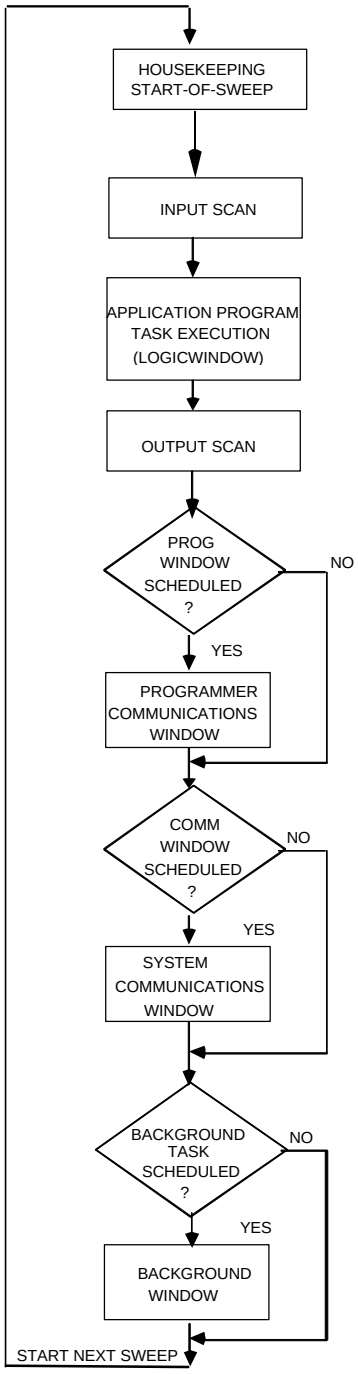


Figure 2-1. Phases of a Typical PLC Sweep

Table 2-1. Major Phases in a Typical PLC Sweep

Step	Description
Housekeeping	Updating %S bits, determining timer update values, and determining the sweep mode occur in this phase.
Input Scan	The CPU reads input data from Bus controllers and input modules during this phase.
Application Program Task Execution (Logic Window)	The CPU solves the logic program(s), using data obtained from the input devices and sets bits to affect the state of output devices.
Output Scan	The CPU writes output data to Bus controllers and output modules during this phase. The user program checksum is computed during this phase of the sweep. Polling for faulted boards also occurs during this phase.
Programmer Communications Window	Communication with the programmer when using serial occurs here with data and/or status transfer in both directions. In addition, reconfiguration of a module or rack also occurs during this portion of the sweep.
System Communications Window	Communications with all intelligent devices except the programmer when using a serial occur during this window. For example, supplying data to a PCM* that is driving a process display would occur during this window.
Background Task Window	CPU self-tests occur in this window.

*For information about the PCM, refer to the Series 90™ Programmable Coprocessor Module and Software Support (GFK-0255).

Housekeeping

The housekeeping portion of the sweep performs all of the tasks necessary to prepare for the start of the sweep. This includes updating %S bits, determining timer update values, and determining the mode of the sweep (Stop or Run).

Input Scan

The scanning of the inputs occurs just prior to the logic solution. During the input scan, the CPU reads input data from the Genius Bus Controllers, FIP Bus Controllers, and Series 90-70 input modules.

Note

Beginning with Release 7.0, Ethernet global data is also read during this scan.

When referring to FIP in this scan, only periodic VCOM (MPS) services are affected. Messages are received in the System Communication Window.

Series 90-70 I/O modules are scanned from lowest to highest I/O reference address. There is no guaranteed scanning order for Bus Controllers.

Note

The input scan will not be performed if a program has an active Suspend I/O function on the previous sweep.

Application Program Task Execution (Logic Window)

The Logic Window is the phase of the sweep where user programs execute. Immediately following the completion of the input scan, the PLC Executive determines which user program(s) are to be run. Programs are then resumed and/or invoked as necessary. Solving the logic provides a new set of outputs.

Interrupt programs and blocks can execute during any phase of the sweep. Refer to section 4 for further details.

There are many ways in which program execution can be controlled to meet the system's timing requirements. The following is a partial list of the commonly used methods:

- Mask Compare and JUMP functions can be used to skip portions of the logic.
- The Suspend I/O function can be used to stop both the input scan and output scan for one sweep. I/O can be updated, as necessary, during the logic execution through the use of DO I/O instructions.
- The Service Request function can be used to suspend or change the time allotted to the window portions of the sweep.
- Program logic can be structured so that blocks and programs are called more or less frequently, depending on their importance and on timing constraints.
- Microcycle sweep mode can be used to phase programs, which need to run less often while limiting the logic window execution time.

A list of execution times for instructions can be found in Appendix A.

Note

In Microcycle Sweep mode, the Logic Window can be skipped or preempted as necessary by the PLC Executive.

Output Scan

Outputs are scanned immediately following logic solution. During the output scan, the CPU sends output data to the Genius Bus Controllers, FIP Bus Controllers, and Series 90-70 output modules. Beginning with Release 7.0, Ethernet global data is also produced in this phase.

Series 90-70 output modules are scanned from lowest to highest I/O reference address. Bus Controllers are scanned from rack 0 to rack 7 and lowest to highest slot number within each rack.

Note

The output scan will not be performed if a program has an active Suspend I/O function on the current sweep.

When referring to FIP in this scan, only periodic VCOM (MPS) services are affected. Messages are received in the System Communication Window.

Polling for faulted boards also occurs during the output scan phase of the sweep. Faulted board polling recognizes replacement boards for faulted ones and reconfigures them into the system. If a board that was previously in the system or configured by the user to be in the system is listed as faulted, it must be polled periodically to determine if a new board has replaced it. Once a previously faulted board is detected as no longer faulted, reconfiguration is run in the Programmer Communications Window until the board(s) are reconfigured into the system.

The background checksum calculation also occurs during the output phase of the sweep. During each output scan phase of the sweep, the configured amount of words of user program is included in the checksum calculation. This checksum helps to ensure the integrity of the user logic while the CPU is executing. If the CPU is configured to perform a background checksum calculation (16 is the default), then this part of the output phase is performed; otherwise, it is skipped.

There are other tests performed during the Output Scan: Processor test checks basic operation of the microprocessor and BCP Opcode test checks basic operation of all BCP instructions.

Note

Beginning with the Release 7 CPUs, for Microcycle Sweep only, the background checksum calculation will occur during the input phase of the sweep.

Programmer Communications Window

This part of the sweep is dedicated to communicating with the programmer and performing faulted board reconfiguration. This is also when communication with the C debugger occurs. If there is a programmer attached, a debugging session is active, or if there is a board in the system that requires reconfiguration (as detected during the faulted board polling portion of the sweep), the CPU executes the Programmer Window. The Programmer Window will not execute if there is no programmer attached, no active debug session occurring, and no board to be configured in the system. During the Programmer Window, highest priority is given to board configuration. Boards are configured as needed, up to the total time allocated to the Programmer Window.

The Programmer Window is used for communication between the CPU and the two dedicated programmer ports: the built-in SNP connection and the parallel programmer connection. The CPU will complete any previously unfinished requests and then begin to process any pending requests in the queue. When the time allocated for the window expires, processing stops.

The Programmer Window time defaults to 10 milliseconds. This value can be configured and stored to the PLC or it can be changed online using your programming software.

Time and execution of the Programmer Window can also be dynamically controlled from the user program using Service Request function #3. The Programmer Communications Window time can be set to a value from 0 to 255 milliseconds.

Note

Even if the Programmer Window is skipped, the PLC can still respond to commands to change mode or state, or to redefine the Programmer Window if the programmer is attached through the parallel port on the Bus Transmitter Module (BTM), or by manually putting the PLC into stop mode.

System Communications Window

The System Communications Window is the part of the sweep used for communication between the CPU and intelligent modules such as the PCM, Genius Bus Controller, FIP Bus Controller, and TCP/IP Ethernet modules.

At the start of the System Communications Window, the CPU will complete any previously unfinished request before executing any pending requests in the queue. When the time allocated for the window expires, processing stops.

The System Communications Window defaults to “Run to Completion” mode. This means that all currently pending requests on all intelligent option modules are processed every sweep. A different mode can be configured and stored to the PLC, or it can be changed online using your programming software.

Time and execution of the System Communications Window can also be dynamically controlled from the user program using Service Request function #4. This allows communications functions to be skipped during certain time-critical sweeps. The System Communications Window time can be set to a value from 0 to 255 milliseconds.

Background Window

A CPU self-test is performed in this window. Included in this self-test is a verification of the checksum for the 90-70 CPU operating system software.

The Background Window time defaults to 0 milliseconds. A different value can be configured and stored to the PLC, or it can be changed online using your programming software.

Time and execution of the Background Window can also be dynamically controlled from the user program using Service Request function #5. This allows background functions to be skipped during certain time-critical sweeps.

Window Modes

The previous sections have described the phases of a typical PLC sweep. The Programmer Window, System Communications Window, and Background Window phases of the PLC sweep can be run in various modes, based on the PLC Sweep mode. (PLC sweep modes are described in detail in section 4.) The following three window modes are available:

Run-to-Completion — In Run-to-Completion mode, all requests made when the window has started are serviced. When all pending requests in the given window have completed, the PLC will transition to the next phase of the sweep.

Constant — In Constant Window mode, the total amount of time that the Programmer Communications Window and Background Window run is fixed. If the time expires while in the middle of servicing a request, these windows are closed, and communications will be resumed the next sweep. If no requests are pending in this window, the PLC will cycle through these windows the specified amount of time polling for further requests. If any window is put in constant window mode, all will be in constant window mode.

Limited In Limited mode, the maximum time that the window runs is fixed. If time expires while in the middle of servicing a request, the window is closed, and communications will be resumed the next time that the given window is run. If no requests are pending in this window, the PLC will proceed to the next phase of the sweep.

Data Coherency in Communications Windows

When running in Constant or Limited Window mode, the Programmer and System Communications Windows may be terminated early in all PLC sweep modes. If an external device, such as a GBC (Genius Bus Controller), is transferring a block of data, the coherency of the data block may be disrupted if the communications window is terminated prior to completing the request. The request will complete during the next sweep; however, part of the data will have resulted from one sweep and the remainder will be from the following sweep. When the PLC is in Normal Sweep mode and the Communications Window is in Run-to-Completion mode, the data coherency problem described above does not exist.

CPU Sweep in STOP Mode

The 90-70 PLC has two modes of operation while it is in Stop mode: Stop/NoIO and Stop/IOScan.

When the PLC is in Stop/NoIO mode the Input Scan, Logic Window, and Output Scan phases of the PLC sweep are skipped.

When the PLC is in Stop/IOScan mode the Logic Window phase of the PLC is skipped but the Input Scan and Output Scan phases are performed each sweep.

In both Stop/NoIO and Stop/IOScan modes, the two Communications Windows run in Run-to-Completion mode and the Background Window runs in Limited mode with a 10 millisecond limit.

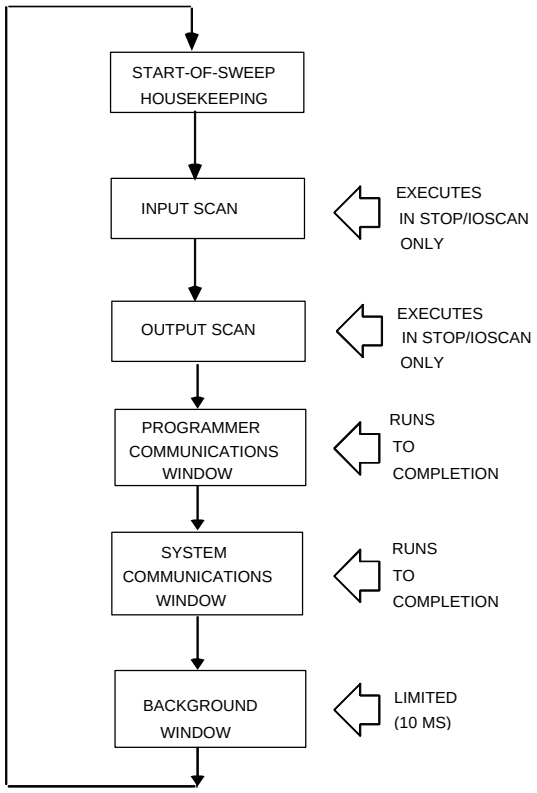


Figure 2-2. CPU Sweep in Stop/NoIO and Stop/IOScan Mode

Note

Stop/IOScan is not supported in Microcycle Sweep mode.

PLC Sweep Modes

The 90-70 PLC supports four PLC sweep modes:

Normal Sweep In Normal Sweep mode, each PLC sweep can consume a variable amount of time. The Logic Window is executed in its entirety each sweep. The Communications and Background Windows can be set to execute in a Limited or Run-to-Completion mode.

Constant Sweep In Constant Sweep mode, each PLC sweep begins at a user-specified Constant Sweep time after the previous PLC sweep began. The Logic Window is executed in its entirety each sweep. If there is sufficient time at the end of the sweep, the PLC will alternate among the Communications and Background Windows, allowing them to execute until it is time for the next sweep to begin.

Constant Window In Constant Window mode, each PLC sweep can consume a variable amount of time. The Logic Window is executed in its entirety each sweep. The PLC will alternate among the Communications and Background Windows, allowing them to execute for a time equal to the user-specified Constant Window timer.

Microcycle Sweep In Microcycle Sweep mode, like Constant Sweep mode, each PLC sweep takes a fixed amount of time. The user specifies the total sweep time (base cycle time) and the total time for the Communications and Background Windows. The Logic Window can be preempted in order to maintain the total sweep time and the Communications Windows and Background Window times. To satisfy the specified window times, the PLC alternates among the Programmer Communications Window, the System Communications Window, and the Background Window, allowing them to execute until it is time for the next sweep to begin.

Note

The information presented above summarizes the different sweep modes. For detailed information on PLC Sweep Modes, refer to “PLC Sweep Modes and Program Scheduling Modes” in section 4 of this chapter.

Section 2: User Reference Data

User References

The PLC data used in an application program is stored as either discrete or register references.

Table 2-2. Register References

Type	Description
%R	Use the prefix %R to assign system register references, which will store program data such as the results of calculations.
%AI	The prefix %AI represents an analog input register. This prefix is followed by the register address of the reference (for example, %AI0015). An analog input register holds the value of one analog input or another value.
%AQ	The prefix %AQ represents an analog output register. This prefix is followed by the register address of the reference (for example, %AQ0056). An analog output register holds the value of one analog output or another value.
%P*	Use the prefix %P to assign program register references that will store program data from the _MAIN block. This data can be accessed from all program blocks. The size of the %P data block is based on the highest %P reference in all blocks. (For more information, refer to the Appendix C, "Memory Allocation.")
%L*	Use the prefix %L to assign local register references that will store program data unique to a block. The size of the %L data block is based on the highest %L reference in the associated block. (For more information, refer to Appendix C, "Memory Allocation.")

* These reference types are scoped at a program level and are therefore only visible to LD programs.

Note

All register references are retained across a power cycle to the CPU.

Table 2-3. Discrete References

Type	Description
%I	The %I prefix represents input references. This prefix is followed by the reference's address in the input table (for example, %I00121). %I references are located in the input status table, which stores the state of all inputs received from input modules during the last input scan. A reference address is assigned to discrete input modules using your programming software. Until a reference address is assigned, no data will be received from the module. %I memory is always retentive.
%Q	The %Q prefix represents physical output references. The coil check function checks for multiple uses of %Q references with relay coils or outputs on functions. Beginning with Release 4 of the software, you can select the level of coil checking desired (Single, Warn Multiple, or Multiple). Refer to "Rung Acceptance" in the online help for additional information. The %Q prefix is followed by the reference's address in the output table (for example, %Q00016). %Q references are located in the output status table, which stores the state of the output references as last set by the application program. This output status table's values are sent to output modules at the end of the program scan. A reference address is assigned to discrete output modules using your programming software. Until a reference address is assigned, no data is sent to the module. A particular %Q reference may be either retentive or non-retentive. *
%M	The %M prefix represents internal references. The coil check function of your programming software checks for multiple uses of %M references with relay coils or outputs on functions. A particular %M reference may be either retentive or non-retentive. *
%T	The %T prefix represents temporary references. These references are never checked for multiple coil use and can, therefore, be used many times in the same program even when coil use checking is enabled. %T may be used to prevent coil use conflicts while using the cut/paste and file write/include functions. Because this memory is intended for temporary use, it is never retained through power loss or Run-to-Stop-to-Run transitions and cannot be used with retentive coils.
%S %SA %SB %SC	The %S, %SA, %SB, and %SC prefixes represent system status references. These references are used to access special PLC data such as timers, scan information, and fault information. For example, the %SC0012 bit can be used to check the status of the PLC fault table. Once the bit is set on by an error, it will not be reset until after the sweep. • %S, %SA, %SB, and %SC can be used on any contacts. • %SA, %SB, and %SC can be used on retentive coils -(M)-. • %S can be used as word or bit-string input arguments to functions or function blocks. • %SA, %SB, and %SC can be used as word or bit-string input or output arguments to functions and function blocks. %S, %SA, %SB, and %SC references are non-retentive.
%G %GA %GB %GC %GD %GE	The %G, %GA, %GB, %GC, %GD, and %GE prefixes represent global data references. These references are used to access data shared among several PLCs. %G, %GA, %GB, %GC, %GD, and %GE references can be used on contacts and retentive coils because the memory is always retentive. %G, %GA, %GB, %GC, %GD, and %GE cannot be used on non-retentive coils.

* Retentiveness is based on the type of coil. For more information, refer to "Retentiveness of Logic and Data" later in this section.

User Reference Size and Default

Maximum user references and default reference sizes for each CPU are listed in the tables below.

Table 2-4. User Reference Sizes

Item	CPU Model					
	935/928 925/915/790	924/914	788	780/781 782/789	771/772	731/732
Maximum %I reference	12288 points	12288 points	352 points[1]	12288 points[2]	2048 points[2]	512 points[2]
Maximum %Q reference	12288 points	12288 points	352 points[1]	12288 points[2]	2048 points[2]	512 points[2]
Maximum %M reference	12288 points	12288 points	12288 points	12288 points	4096 points	2048 points
Maximum %T reference	256 points	256 points	256 points	256 points	256 points	256 points
%S total (S, SA, SB, SC)	512 points	512 points	512 points	512 points	512 points	512 points
%G (GA, GB, GC, GD, GE)	7680 points	7680 points	7680 points	7680 points	7680 points	1280 points
User RAM	1024K bytes (6MB for 928)	512K bytes	512K bytes	512K bytes (CPX 782 has 1024 KB)	CPU771/2: 64, 128, 256, 512 KB depending on expansion memory board purchased CPX772: 512 KB	32K bytes
Maximum %AI reference	8K words	8K words	8K words	8K words	8K words	8K words
Maximum %AQ reference	8K words	8K words	8K words	8K words	8K words	8K words
Maximum %R, 1K word increments	16K words	16K words	16K words	16K words	16K words	16K words
Maximum %L (per block)	8K words.	8K words	8K words	8K words	8K words	8K words
Maximum %P	8K words	8K words	8K words	8K words	8K words	8K words

[1] Total number of physical input and output points together cannot exceed 352 points. This corresponds to approximately 100 redundant points. Refer to Chapter 1 of GFK-1277 for more information.

[2] Prior to Release 6 of Logicmaster, the programming software restricted the total %I and %Q to the limit shown individually for each. For example, when using previous programming packages with a 782 CPU, there was a maximum of 12288 points of %1 and %Q ***combined***. This restriction no longer exists with the newer versions of Logicmaster.

Table 2-5. Default Memory Sizes

Memory Type	CPU Model					
	935/928 925/915/790	924/914	780/781/782 788/789	781/782	771/772	731/732
%AI	64 words	64 words	64 words	64 words	64 words	64 words
%AQ	64 words	64 words	64 words	64 words	64 words	64 words
%R	1024 words	1024 words	1024 words	1024 words	1024 words	1024 words
%P	0 words	0 words	0 words	0 words	0 words	0 words
%L	0 words	0 words	0 words	0 words	0 words	0 words

%G User References and CPU Memory Locations

The Series 90-70 CPU contains only one data space for all of the global data references (%G, %GA, %GB, %GC, %GD, and %GE). The internal CPU memory for this data is 7680 bits long. Your programming software provides the user with a subdivided representation by using %G, %GA, %GB, %GC, %GD, and %GE prefixes; allowing each of these prefixes to be used with bit offsets in the range 1-1280. Your programming software interprets the requested global reference type (%G, %GA, %GB, %GC, %GD, or %GE) and converts it to the %G memory type and correct bit offset for use by the CPU. The actual mapping is shown in the table displayed below.

Table 2-6. %G References and Memory Locations

Global Data Type	%G	%GA	%GB	%GC	%GD	%GE
References Used by the Programming Software	%G1–1280	%GA1–1280	%GB1–1280	%GC1–1280	%GD1–1280	%GE1–1280
Memory Locations Used by the CPU	%G1–1280	%G1281–2560	%G2561–3840	%G3841–5120	%G5121–6400	%G6401–7680

This information is useful when programming 90-70 CPU applications in C language using the C Programmer's Toolkit. For more information about using the C Programmer's Toolkit, refer to the *C Programmer's Toolkit for Series 90™ PLCs User's Manual* (GFK-0646).

Note

A model 731 CPU supports **only** %G since it has only 1280 points of global data.

Bulk Memory Access (BMA)

Bulk Memory Access provides a large block of memory that can be accessed from your application program. Possible uses include recipe files, bulk storage, memory allocation, temporary memory blocks, and so forth. Use of this feature requires Control programming software version 2.2, or later. On-line help for the BMA feature are available in Control software version 2.3.

The BMA is a contiguous block of memory (up to 4 Mbytes) that can be used by logic programs and external devices. BMA is available in the CPX CPUs (CPX772, 782, 928, 935) with CPU release 7.80, or later. A service request (SVCREQ #36) is used to read from the BMA to a local buffer area in the PLC, write from the local buffer to the PLC, and retrieve the address of the BMA in the PLC.

Genius Global Data

The Series 90-70 PLC supports the sharing of data among multiple PLC systems that share a common Genius I/O bus. This mechanism provides a means for the automatic and repeated transfer of %G, %I, %Q, %AI, %AQ, and %R data. No special application programming is required to use global data since it is integrated into the I/O scan. All GE Fanuc PLCs that have Genius I/O capability can send and receive global data from a Series 90-70 PLC.

Transitions and Overrides

The %I, %Q, %M, and %G user references have associated transition and override bits. %T, %S, %SA, %SB, and %SC references have transition bits but not override bits. The CPU uses transition bits for counters, transitional contacts, and transitional coils. Note that counters do not use the same kind of transition bits as contacts and coils. Transition bits for counters are stored within the locating reference.

Caution

Do not override transitional coils. If a transitional coil is overridden and the override is then removed, the coil will come on for one sweep. This can cause unexpected consequences in the PLC ladder logic and in field devices attached to the PLC.

When override bits are set, the associated references cannot be changed from the program or the input device; they can only be changed on command from the programmer.

Retentiveness of Logic and Data

Data is defined as retentive if it is saved by the PLC when the PLC is stopped. The Series 90-70 PLC preserves program logic, fault tables and diagnostics, checksums for program logic, overrides and output forces, word data (%R, %L, %P, %AI, %AQ), bit data (%I, %S, %G, fault locating references, and reserved bits), %Q and %M data (unless used with non-retentive coils). %T data is not saved.

%Q and %M references are non-retentive (that is, cleared when the PLC transitions from Stop to Run, including power-up in Run mode) whenever they are used with non-retentive coils. Non-retentive coils include coils -()-, negated coils -(/)-, SET coils -(S)-, and RESET coils -(R)-.

When %Q or %M references are used with retentive coils or are used as function block outputs, the contents are retained through power loss and Run-to-Stop-to-Run transitions. Retentive coils include retentive coils -(M)-, negated retentive coils -(/M)-, retentive SET coils -(SM)-, and retentive RESET coils -(RM)-.

The last time a %Q or %M reference is programmed on a coil instruction determines whether the %Q or %M reference is retentive or non-retentive based on the coil type. For example, if %Q00001 was last programmed as the reference of a retentive coil, the %Q00001 data will be retentive. However, if %Q00001 was last programmed on a non-retentive coil, then the %Q00001 data will be non-retentive.

Note

When only standalone C programs are used, the retentive nature of data is based solely on the memory type since there are no coil instructions. In this case %Q and %M memory types are retentive.

Data Scope

Each of the user references has “scope”; that is, it may be available throughout the system, available to all programs, restricted to a single program, or restricted to local use within a block.

Table 2-7. Data Scope of User Reference Data

User Reference	Range	Scope
%I, %Q, %M, %T, %S, %SA, %SB, %SC, %G, %R, %AI, %AQ, convenience references, fault locating references	System	From any program, block, or host computer
%P	Program	From any block, but not from other programs
%L	Local	From within a block only

In an LD block:

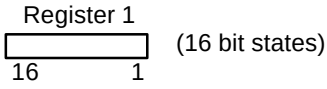
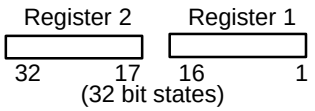
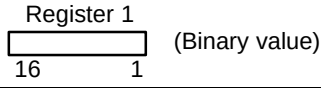
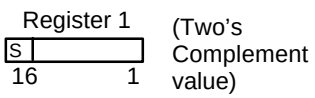
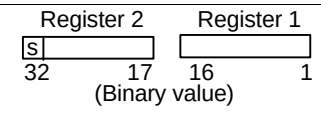
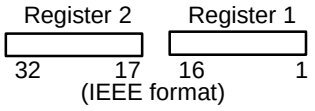
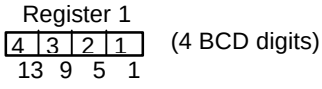
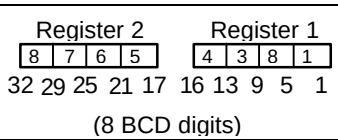
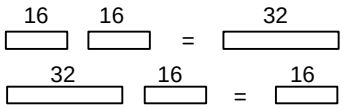
- %P should be used for program references that are shared with other blocks.
- %L are local references which can be used to restrict the use of register data to that block. These local references are not available to other parts of the program.

%I, %Q, %M, %T, %S, %SA, %SB, %SC, %G, %R, %AI, and %AQ references are available throughout the system.

Appendix C contains a Memory Allocation worksheet for determining the total number of bytes of user data used and how much is still available for the user program.

Data Types

Table 2-8. Data Types

Type	Name	Description	Data Format
BOOL	Boolean	A Boolean data type is the smallest unit of memory. It has two states, 1 or 0. A BOOL string may have length N.	
BYTE	Byte	A Byte data type has an 8-bit value. It has 256 values (0–255). A BYTE string may have length N.	
WORD	Word	A Word data type uses 16 consecutive bits of data memory. The valid range of word values is 0000 hex to FFFF hex.	
DWORD	Double Word	A Double Word data type has the same characteristics as a single word data type, except that it uses 32 consecutive bits in data memory instead of only 16 bits.	
UINT	Unsigned Integer	Unsigned integers use 16-bit memory data locations. They have a valid range of 0 to +65535 (FFFF hex).	
INT	Signed Integer	Signed integers use 16-bit memory data locations, and are represented in 2's complement notation. The valid range of an INT data type is –32768 to +32767.	
DINT	Double Precision Integer	Double precision integers are stored in 32-bit data memory locations (two consecutive 16-bit memory locations) and are always signed values (bit 32 is the sign bit.) The valid range of a DINT data type is –2147483648 to +2147483867.	
REAL	Floating-Point	Real numbers use 32 consecutive bits (two consecutive 16-bit memory locations). The range of numbers that can be stored in this format is from ± 1.401298E-45 to ± 3.402823E+38. Refer to Appendix D "Using Floating-Point Math," for IEEE format.	
BCD-4	Four-Digit Binary Coded Decimal	Four-digit BCD numbers use 16-bit data memory locations. Each BCD digit uses four bits and can represent numbers between 0 and 9. This BCD coding of the 16 bits has a legal value range of 0 to 9999.	
BCD-8	Eight-Digit Binary Coded Decimal	Eight-digit BCD data types use two consecutive 16-bit data memory locations (32 consecutive bits). Each BCD digit uses 4 bits per digit to represent numbers from 0 to 9. The complete valid range of the 8-digit BCD data type is 0 to 99999999.	
MIXED	Mixed	A Mixed data type is available only with the MUL and DIV functions. The MUL function takes two integer inputs and produces a double integer result. The DIV function takes a double integer dividend and an integer divisor to product an integer result.	
ASCII	ASCII	Eight-bit encoded characters. A single reference is required to make up 2 (packed) ASCII characters. The first character of the pair corresponds to the low byte of the reference word. The remaining 7 bits in each section are converted. Command codes and non-displayable characters appear on the screen as non-alphanumeric characters (for example, @).	

S = Sign bit (0 = positive, 1 = negative).

Note

Using function blocks that are not explicitly bit-typed will affect transitions for all bits in the written byte/word/dword.

Also for information about using floating-point numbers, refer to Appendix D, *Using Floating-Point Numbers*.

System Status References

System status references (formerly called "Convenience" references) in the Series 90-70 PLC are assigned to %S, %SA, %SB, and %SC memory. They each have a system-supplied nickname that enables you to enter the nickname rather than the exact %S reference. Examples of time tick references include T_10MS, T_100MS, T_SEC, and T_MIN. Examples of other system status references include FST_SCN, ALW_ON, and ALW_OFF.

Note

%S bits are read-only bits; do not write to these bits. You may, however, write to %SA, %SB, and %SC bits.

Listed below are available system status references that may be used in an application program. When entering logic, either the reference or the nickname can be used. Refer to Chapter 3, "Fault Explanation and Correction," for more detailed fault descriptions and information on correcting faults.

While it is possible to use these special names in another context, their use is restricted (for example, you cannot use them as a block name or folder name).

Note

Most references not listed in the following table (for example, %S0002) are not used for the Series 90-70 PLC. Products that have Genius Modular Redundancy (CPU788, CPU789, and CPU790) have additional references, as do the CPU780, CGR772, and CGR935 (with CPU redundancy). Refer to GFK-1277 for status references for Genius Modular Redundancy (GMR), GFK-0827 for Hot Standby CPU Redundancy, and GFK-1427 for Enhanced Hot Standby CPU Redundancy.

Table 2-9. System Status References

Reference	Name	Definition
%S0001	FST_SCN	Current sweep is the first sweep in which the LD or standalone C program executed. Set the first time the user program is executed after a Stop/Run transition and cleared upon completion of its execution.
%S0003	T_10MS	0.01 second timer contact.
%S0004	T_100MS	0.1 second timer contact.
%S0005	T_SEC	1.0 second timer contact.
%S0006	T_MIN	1.0 minute timer contact.
%S0007	ALW_ON	Always ON.
%S0008	ALW_OFF	Always OFF.
%S0009	SY_FULL	Set when the PLC fault table fills up (size configurable with a default of 16 entries). Cleared when the PLC fault table is cleared.
%S0010	IO_FULL	Set when the I/O fault table fills up (size configurable with a default of 32 entries). Cleared when the I/O fault table is cleared.
%S0011	OVR_PRE	Set when an override exists in %I, %Q, %M, or %G memory.
%S0012	FRC_PRE	Set when force exists on a Genius point.
%S0013	PRG_CHK	Set when background program check is active.
%S0014	PLC_BAT	Set to indicate a bad battery in a Release 4 or later CPU. The contact is updated when a change in the battery status occurs.
%S0121	FST_EXE	Current sweep is the first time this block has been called. Set when transitioning from Stop to Run. FST_EXE is not available to standalone C programs.
%SA0001	PB_SUM	Set when a checksum calculated on the application program does not match the reference checksum. If the fault was due to a temporary failure, the discrete bit can be cleared by again storing the program to the CPU. If the fault was due to a hard RAM failure, then the CPU must be replaced. Cleared when the PLC fault table is cleared.
%SA0002	OV_SWP	Set when the PLC detects that the previous sweep took longer than the time specified by the user. Cleared when the PLC detects that the previous sweep did not take longer than the specified time. It is also cleared during the transition from Stop to Run mode. Only valid if the PLC is in Constant Sweep or Microcycle Sweep mode. Also cleared when the PLC fault table is cleared.
%SA0003	APL_FLT	Set when an application fault occurs. Cleared when the PLC fault table is cleared.
%SA0009	CFG_MM	Set when a configuration mismatch is detected during system power-up or during a store of the configuration. Cleared by powering up the PLC when no mismatches are present or during a store of configuration that matches hardware. Also cleared when the PLC fault table is cleared.
%SA0010	HRD_CPU	Set when the diagnostics detects a problem with the CPU hardware. Cleared by replacing the CPU module. Also cleared when the PLC fault table is cleared.
%SA0011	LOW_BAT	Set when a low battery fault occurs. Cleared by replacing the battery and ensuring that the PLC powers up without the low battery condition. . Also cleared when the PLC fault table is cleared.
%SA0012	LOS_RCK	Set when an expansion rack stops communicating with the PLC CPU. Cleared by fixing the problem and power cycling the rack. Also cleared when the PLC fault table is cleared

Table 2-9. System Status References (Continued)

Reference	Name	Definition
%SA0013	LOS_IOC	Set when a Bus Controller stops communicating with the PLC. Cleared by replacing the module and cycling power on the rack containing the module, or when the I/O fault table is cleared.
%SA0014	LOS_IOM	Set when an I/O module stops communicating with the PLC CPU. Cleared by replacing the module and cycling power on the rack containing the module, or when the I/O fault table is cleared.
%SA0015	LOS_SIO	Set when an option module stops communicating with the PLC CPU. Cleared by replacing the module and cycling power on the rack containing the module, or when the PLC fault table is cleared.
%SA0017	ADD_RCK	Set when an expansion rack is added to the system. Cleared by cycling power on the rack containing the module and the configuration matches the hardware after a store, or when the PLC fault table is cleared.
%SA0018	ADD_IOC	Set when a Bus Controller is added to a rack. Cleared by cycling power on the rack containing the module and when the configuration matches the hardware after a store, or when the I/O fault table is cleared, or when the I/O fault table is cleared.
%SA0019	ADD_IOM	Set when an I/O module is added to a rack. Cleared by cycling power on the rack containing the module and when the configuration matches the hardware after a store.
%SA0020	ADD_SIO	Set when an option module is added to a rack. Cleared by cycling power on the rack containing the module and when the configuration matches the hardware after a store, or when the PLC fault table is cleared.
%SA0022	IOC_FLT	Set when a Bus Controller reports a bus fault, a global memory fault, or an IOC hardware fault. Cleared by cycling power on the rack containing the module and when the configuration matches the hardware after a store, or when the I/O fault table is cleared.
%SA0023	IOM_FLT	Set when an I/O module reports a circuit or module fault. Cleared by cycling power on the rack containing the module and when the configuration matches the hardware after a store, or when the I/O fault table is cleared.
%SA0027	HRD_SIO	Set when a hardware failure is detected in an option module. Cleared by replacing the module and cycling power on the rack containing the module, or when the PLC fault table is cleared.
%SA0029	SFT_IOC	Set when there is a software failure in the I/O Controller, or when the I/O fault table is cleared.
%SA0031	SFT_SIO	Set when an option module detects an internal software error. Cleared by cycling power on the main rack and when the configuration matches the hardware, or when the PLC fault table is cleared.
%SA0032	SBUS_ER	Set when a bus error occurs on the VME bus backplane. Cleared by cycling power on the main rack.
%SA0081 – %SA0112		Set when a user-defined fault is logged in the PLC fault table. These bits are cleared when the PLC fault table is cleared.
%SB0001	WIND_ER	Set when there is not enough time to start the Programmer Window in Constant Sweep or Microcycle Sweep mode, or when there is not enough time to start the Logic Window in Microcycle Sweep mode. Cleared when the PLC detects that the previous sweep did have enough time to perform the window.
%SB0009	NO_PROG	Set when the PLC CPU powers up with memory preserved, but no user program is present. Cleared when the PLC powers up with a program present, or when the PLC fault table is cleared.

Table 2-9. System Status References (Continued)

Reference	Name	Definition
%SB0010	BAD_RAM	Set when the CPU detects corrupted RAM memory at power-up. Cleared when the CPU detects that RAM memory is valid at power-up, or when the PLC fault table is cleared.
%SB0011	BAD_PWD	Set when a password access violation occurs. Cleared when the PLC fault table is cleared.
%SB0012	NUL_CFG	Set when an attempt is made to put the PLC in Run mode when there is no configuration data present. Cleared when configuration data is present and the PLC is in Run mode, or when the PLC fault table is cleared.
%SB0013	SFT_CPU	Set when the CPU detects an error in the CPU operating system software. Cleared by cycling power to the CPU, or when the PLC fault table is cleared.
%SB0014	STOR_ER	Set when an error occurs during a programmer store operation. Cleared when a store operation is completed successfully.
%SB0016	MAX_IOC	Set when more than 32 IOCs are configured for the system. Cleared by modifying the configuration and storing it to the PLC CPU, or when the PLC fault table is cleared.
%SB0017	SBUS_FL	Set when the PLC fails to gain access to the bus. Cleared by cycling power on the main rack, or when the PLC fault table is cleared.
%SC0009	ANY_FLT	Set when any fault occurs. Cleared when both fault tables are cleared.
%SC0010	SY_FLT	Set when any fault occurs that causes an entry to be placed in the PLC fault table. Cleared when the PLC fault table is cleared.
%SC0011	IO_FLT	Set when any fault occurs that causes an entry to be placed in the I/O fault table. Cleared when the I/O fault table is cleared.
%SC0012	SY_PRES	Set as long as there is at least one entry in the PLC fault table. Cleared when the PLC fault table is cleared. Once the bit is set on by an error, it will not be reset until after the sweep.
%SC0013	IO_PRES	Set as long as there is at least one entry in the I/O fault table. Cleared when the I/O fault table is cleared.
%SC0014	HRD_FLT	Set when a hardware fault occurs. Cleared when both fault tables are cleared.
%SC0015	SFT_FLT	Set when a software fault occurs. Cleared when both fault tables are cleared.

Fault References

For a detailed discussion of fault references, please refer to Chapter 3.

Table 2-10. System Fault References

System Fault Reference	Description
ANY_FLT	Any fault in the system
SY_PRES	Any system fault in the Series 90 PLC
IO_PRES	Any I/O fault
SY_FLT	Indicates a new entry in the PLC fault table
IO_FLT	Indicates a new entry in the I/O fault table
HRD_FLT	Any hardware fault
SFT_FLT	Any software fault

Table 2-11. Configurable Fault References

Configurable Fault References (Default Action)	Description
SBUS_ER (diagnostic)	System bus error. (The BSERR signal was generated on the VME system bus.)
HRD_CPU (fatal)	PLC CPU hardware fault, such as failed memory device or failed serial port.
HRD_SIO (diagnostic)	Non-fatal hardware fault on any module in the system, such as the failure of a serial port on a PCM.
SFT_IOC (diagnostic)	Non-recoverable software error in a Genius Bus Controller.
SFT_SIO (diagnostic)	Non-recoverable software error in a PCM or LAN interface module.
PB_SUM (fatal)	Program or block checksum failure during power-up or in Run mode.
LOW_BAT (diagnostic)	Low battery signal from CPU or another module in the system.
OV_SWP (diagnostic)	Constant sweep time exceeded.
SY_FULL, IO_FULL (diagnostic)	PLC fault table full (configurable with a default of 16 entries). I/O fault table full (configurable with a default of 32 entries).
APL_FLT (diagnostic)	Application fault.
LOS_RCK (diagnostic)	Loss of rack (BRM failure, loss of power) or missing a configured rack.
LOS_IOC (diagnostic)	Loss of Bus Controller channel or missing a configured Bus Controller.
LOS_IOM (diagnostic)	Loss of I/O module (does not respond) or missing a configured I/O module.
LOS_SIO (diagnostic)	Loss of option module (does not respond) or missing a configured module.
ADD_RCK (diagnostic)	New rack added or previously faulted rack has returned.
ADD_IOC (diagnostic)	Previously faulted Bus Controller is no longer faulted.
ADD_IOM (diagnostic)	Previously faulted I/O module is no longer faulted.
ADD_SIO (diagnostic)	New option module is added or previously faulted module no longer faulted.
IOC_FLT (diagnostic)	Non-fatal bus or Bus Controller error—more than 10 bus errors in 10 seconds (error rate is configurable).
IOM_FLT (diagnostic)	Point or channel on an I/O module—a partial failure of the module.
CFG_MM (fatal)	Wrong module type detected during power-up or Run mode. The PLC does not check the configuration parameters set up for individual modules such as Genius I/O blocks.

Table 2-12. Non-Configurable Faults

Non-Configurable Faults	Description
SBUS_FL (fatal)	System bus failure. The PLC CPU was not able to access the VME bus. BUSGRT-NMI error.
NO_PROG (information)	No application program is present at power-up. Should only occur the first time the PLC is powered up or if the battery-backed RAM containing the program fails.
BAD_RAM (fatal)	Corrupted program memory at power-up. Program could not be read and/or did not pass checksum tests.
WIND_ER (information)	Window completion error. Servicing of Programmer or Logic Window was skipped. Occurs in Constant Sweep or Microcycle Sweep mode.
BAD_PWD (information)	Change of privilege level request to a protection level was denied; bad password.
NUL_CFG (fatal)	No configuration present upon transition to Run mode. Running without a configuration is equivalent to suspending the I/O scans.
SFT_CPU (fatal)	CPU software fault. A non-recoverable error has been detected in the CPU. May be caused by Watchdog Timer expiring.
MAX_IOC (fatal)	The maximum number of bus controllers has been exceeded. The Series 90 PLC supports 32 bus controllers.
STOR_ER (fatal)	Download of data to PLC from the programmer failed; some data in PLC may be corrupted.

Note

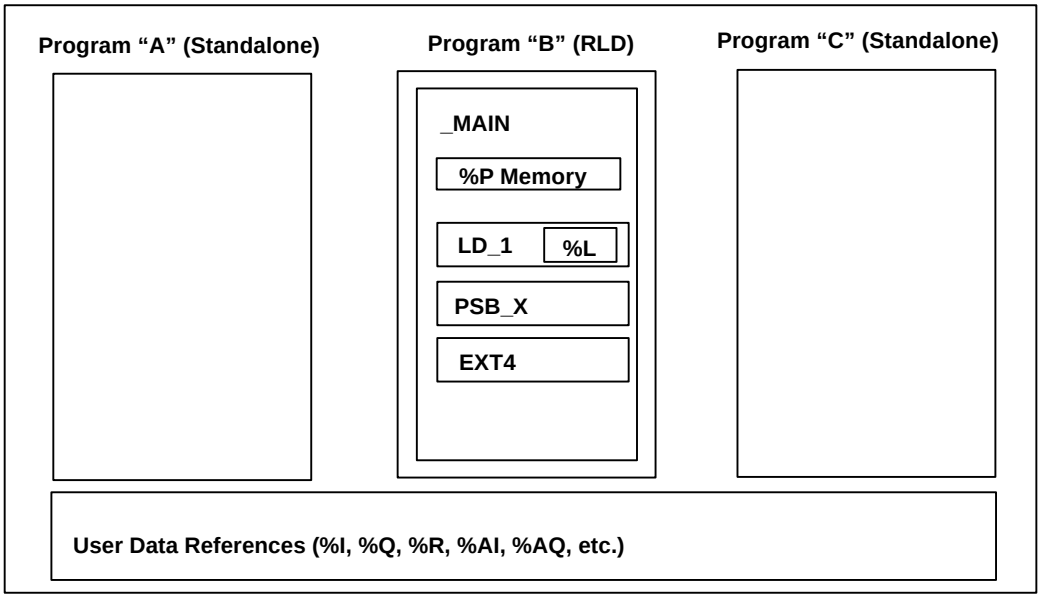
Fault and FIP locating references are discussed in Chapter 3, section 1 of this manual

Refer to the *Series 90™ Sequential Function Chart Programming Language User's Manual* (GFK-0854) for SFC references.

Section 3: Program Organization

The user program(s) contains the logic that is used to process input data and control output data. Program logic is executed repeatedly by the PLC. The Series 90-70 PLC allows up to 16 user programs, with a maximum of 1 LD program. Refer to Tables 2-4 and 2-5 for a listing of program sizes and reference limits for each CPU model.

The following figure depicts 3 user programs, two of which are standalone C programs. The LD program consists of four blocks (_MAIN, LD_1, PSB_X, and EXT4). The figure further illustrates the scoping of various memory types: all references except %P and %L are visible to the standalone programs; %P memory is visible to all of the program blocks, and the LD_1 block has its own local data, %L. Details of standalone programs and blocks are described later in this section.



Ladder Logic Programming

An LD program for the Series 90-70 PLC consists of one or more units called blocks. Four types of blocks are supported by the Series 90-70 PLC:

Table 2-13. Block Types

Block Type	Programming Language	Size Limit	Number of Parameter Pairs	Notes
LD	Ladder Logic	32 KB in Control*	n/a	
SFC	Ladder Logic/SFC	32 KB in Control*	n/a	SFC blocks cannot be used as Interrupt blocks.
PSB	Ladder Logic	32 KB in Control*	0–7	
External	C	64,000 bytes	0–7	External blocks cannot call any other blocks. External blocks are created using the C Programmer's Toolkit.

*When running in Logicmaster compatibility mode, the limit is 16 KB.

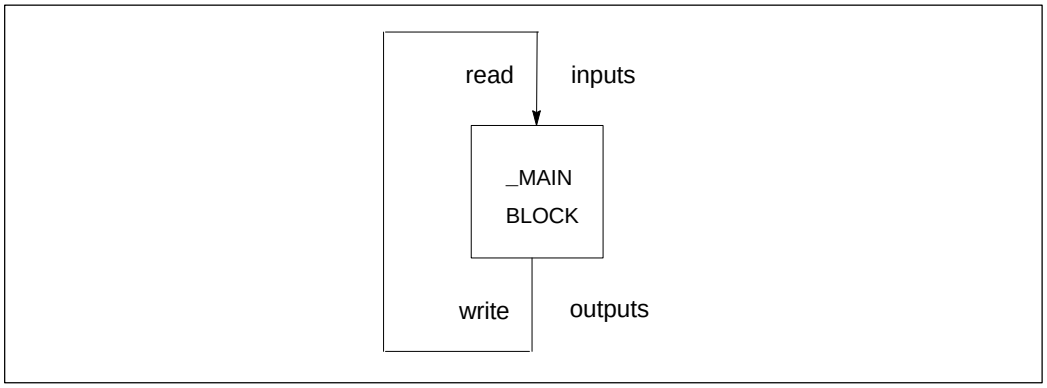
- SFC programming is described in detail in the *Series 90 Sequential Function Chart Programming Language User's Manual*, GFK-0854. Sequential Function Chart (SFC) is an IEC-compliant, graphical, state language specifically designed for controlling sequential processes.
- LD (also known as Relay Logic Diagram language) is the language used in the LD Editor window within Control and in Logicmaster.
- PSB (Parameterized Subroutine Blocks) are LD blocks that have input and output parameters.

Note

Up to 255 blocks can be used. The maximum number of block calls that can be programmed within a given block is 64. The maximum number of programmed calls to a particular block is 255. (A block can be executed any number of times, but there cannot be more than 255 explicit calls to any given block.)

Main Block

When using an LD program there is always a _MAIN block. LD program execution begins with the _MAIN block.

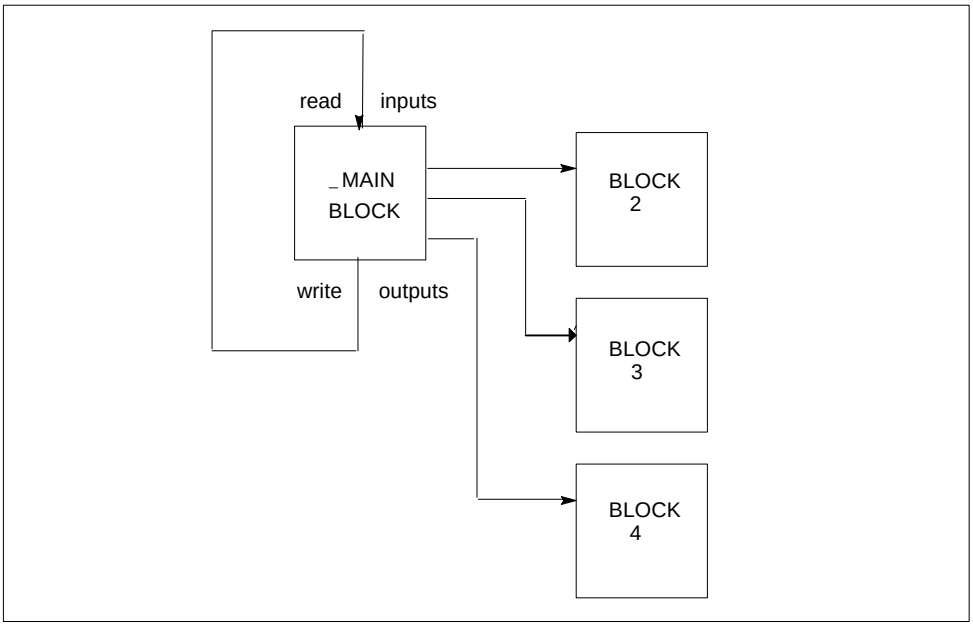


Blocks

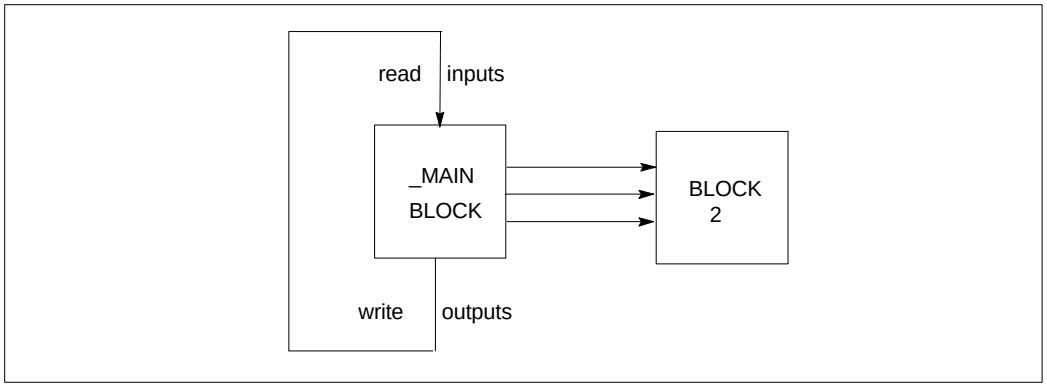
Structuring a program as blocks enables you to reuse logic. Logic that needs to be repeated can be entered in a block. Calls would then be made to that block to execute the logic. In this way, total program size is reduced. Dividing a program into smaller blocks also simplifies programming and reduces the overall amount of logic needed for the program.

Examples of Using Blocks

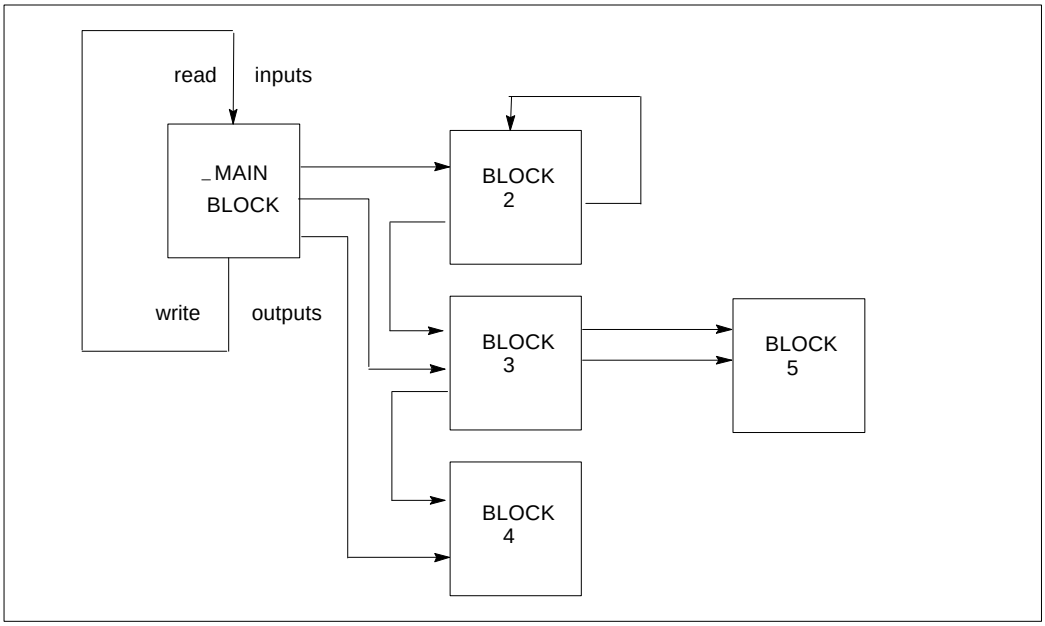
As an example, the logic for an LD program could be divided into three blocks, each of which could be called as needed from the _MAIN block. (A block cannot call the _MAIN block.) In this example, the _MAIN block might contain little logic, serving primarily to sequence the other blocks.



A block can be used many times as the program executes. Logic that needs to be repeated several times in a program could be entered in a block. Calls would then be made to that block to access the logic.



In addition to being called from the _MAIN block, blocks can also be called by other blocks. A block may even call itself.



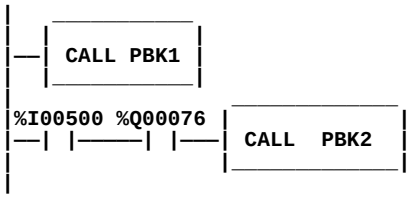
There may be no limit to the number of levels of calls to blocks that your programming software will allow. However, the PLC will only allow a certain number of nested calls before an "Application Stack Overflow" fault is logged and the PLC transitions to Stop/Fault mode. The call depth is guaranteed to be at least four on the 731 and 732 CPUs and eight on all other models. The actual call depth allowed depends on the amount of data (non-boolean) flow used in the blocks. If less than the 171 word data flow limit is used, then more nested calls may be made. The call level nesting counts the _MAIN block as level 1. The illustration above shows three levels of calls.

Note

Before a block can be used, you need to define it in the block declarations. For information on block declarations, refer to the documentation of your programming software.

How Blocks Are Called

A block executes when called from the program logic in the _MAIN block or another block.

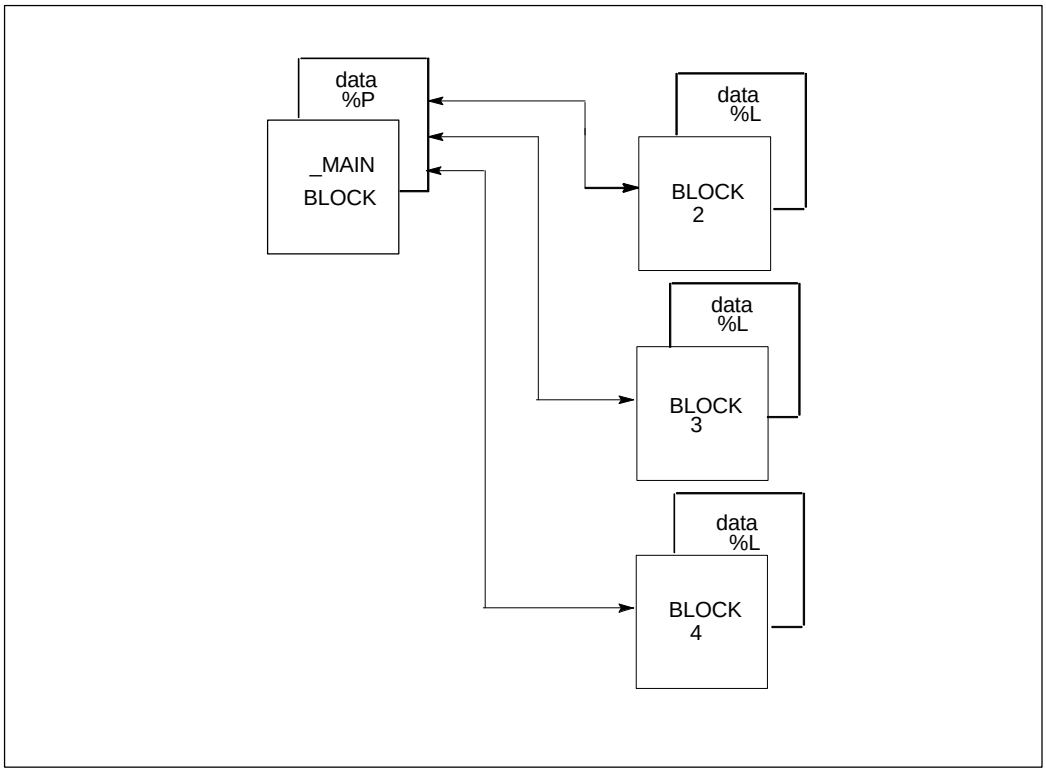


In the example above, PBK1 will always be called. Conditional logic can be used to control calling the block. In order for PBK2 to be called, both input %I00500 and output %Q00076 must be ON.

Blocks and Local Data

Each block in the LD program can have an associated data block. The _MAIN data block is referenced by %P; all other data blocks are referenced by %L.

The size of the data block is dependent on the highest reference in its block for %L and in all blocks for %P. Appendix C, “Memory Allocation,” provides a worksheet for determining the total number of bytes of user data used and how much is still available for the user program.



All blocks within the LD program can use data associated with the _MAIN block (%P). Blocks can use their own %L references as well as the %P references that are available to all blocks. The _MAIN block cannot use %L.

Note

External blocks and Parameterized Subroutine Blocks do not have their own %L data; instead, they inherit the %L data of the calling block.

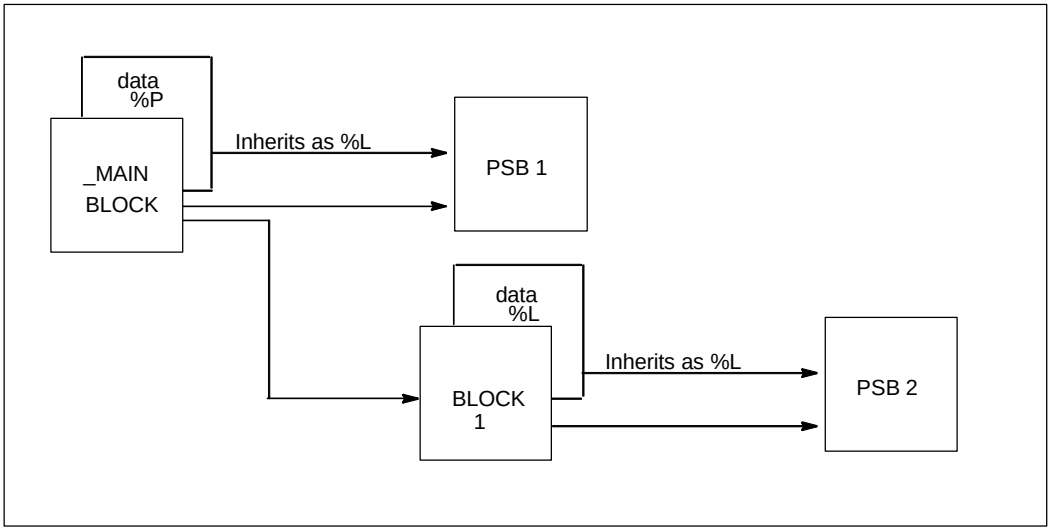
Parameterized Subroutine Blocks

A Parameterized Subroutine Block (PSB) is an optional, user-defined function block, configured with between zero and seven input/output parameter pairs.

As with other blocks, Parameterized Subroutine Blocks can be called by the _MAIN block, other blocks, or itself. The calling block may pass parameters to the Parameterized Subroutine Block. When a Parameterized Subroutine Block is declared, it must be assigned a unique block name along with the number, type, and length of the parameters. Each parameter, other than the Enable and Enable Out parameters, is designated as a BOOL, WORD, or NWORD type, along with a specified length. BOOL lengths range from 1 to 256; WORD and NWORD lengths range from 1 to 512. Default is one bit for BOOL lengths or one word for WORD and NWORD lengths. In addition, you may also declare an optional three-character formal parameter reference name.

Parameterized Subroutine Blocks and Local Data

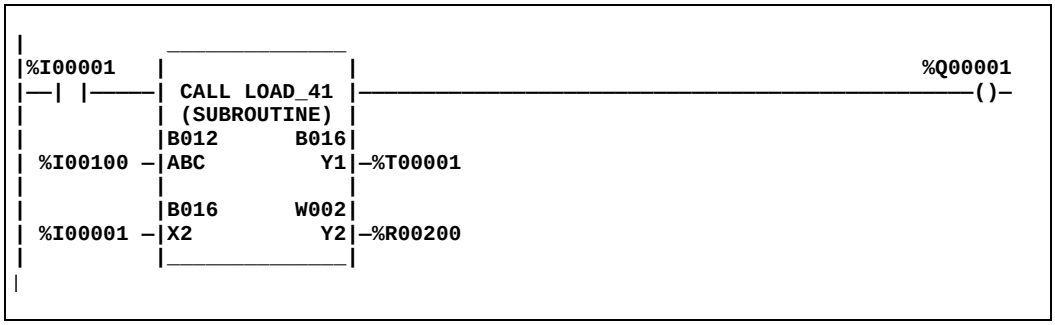
Parameterized Subroutine Blocks support the use of %P global data. Parameterized Subroutine Blocks do not have their own %L data, but instead inherit the %L data of the calling block. Parameterized Subroutine Blocks also inherit %S contacts, such as FST_EXE, from the calling block. If %L references are used within a Parameterized Subroutine Block and the block is called by _MAIN, %L references will be inherited from the %P references wherever encountered in the Parameterized Subroutine Block (for example, %L0005 = %P0005).



How Parameterized Subroutine Blocks Are Called

A Parameterized Subroutine Block executes when called from the program logic in the _MAIN block, another block, or itself.

In the following example, if %I00001 is set, the parameterized subroutine named LOAD_41 is executed. The LOAD_41 subroutine block operates on the input data (located at reference addresses %I00100 – %I00111 and %I00001 – %I00016) and produces values in the block of output data (located at reference addresses %T00001 – %T00016, and at register memory addresses %R00200 – %R00201). The logic within the subroutine can also control the OK output of the Parameterized Subroutine Block. This example shows the subroutine CALL instruction as it will appear in the calling block.



Referencing Formal Parameters Within a Parameterized Subroutine Block

Formal parameters are those parameters used within Parameterized Subroutine Block that are passed from and to the calling block. They are either BOOL, WORD, or NWORD types. NWORD type parameters may be used on any multiword type operands, but not on discrete types. (An NWORD is a number of words passed into a Parameterized Subroutine Block).

The formal parameters are identified as X input parameters or Y output parameters, followed by the number of the input or output parameter, respectively. For example, X2 indicates the parameter used at location X2 in the parameterized subroutine declaration. The X2 label could be followed by a value of 1 to 16 to the length provided in the subroutine declaration (B016).

Up to seven formal parameter pairs may be declared in a Parameterized Subroutine Block. The formal parameter type, number, and length use the form:

ab[ccc]

where:	**a = X**	denotes an input formal parameter.
	a = Y	denotes an output formal parameter.
	b	is a parameter number between 1 and 7.
	c	is a valid BOOL, WORD, or NWORD index.

The labels X1 through X7 and Y1 through Y7 may be assigned a nickname of up to three characters.

Assigned parameters are PLC references or data flow that pass their address or data into or out of a Parameterized Subroutine Block. An assigned parameter may pass either the value of the data in the assigned parameter (BOOL type parameters) or the address of the assigned parameter (WORD or NWORD type parameters). Assigned parameters are defined in a parameter assignment table.

Restrictions on Formal Parameters within a Parameterized Subroutine Block

In general, formal BOOL parameters are allowed on all contacts, coils, and function block parameters that allow discrete references (%I, %Q, %M, %T, %S, %G, and %U). Formal WORD and NWORD parameters are allowed on all function block parameters that allow register references (%R, %AI, %AQ, %P, %L, and %UR). NWORDS should not be used in Control. The type exists only to maintain compatibility with folders created through Logicmaster and imported into a Control folder. In Logicmaster, NWORD parameters are allowed only on multiword type parameters (that is, DINT, DWORD, or REAL). (An NWORD is a number of words passed into a Parameterized Subroutine Block.)

The following list contains several exceptions and restrictions that have been identified when using formal parameters within a Parameterized Subroutine Block:

1. Transitional contacts, transitional coils, and retentive coils are not allowed with formal parameters. The editor (that is, the editor tool within the programming package) will substitute the non-retentive equivalent of these functions, **+——(M)**, **+——(SM)**, and **+——(/M)** and display an appropriate warning message.
2. Formal BOOL input parameters cannot be used as output parameters on a function block.
3. The DO I/O function is not allowed with formal parameters.
4. Multiword type function block parameters (that is, DINT, DWORD, or REAL) are only allowed with formal NWORD parameters.
5. Formal parameters are not allowed on the following function block parameters:

Function	Parameter
Service Request (Service Request)	PARMS input parameter.
Communications Request (COMMREQ)	IN input parameter.
DATA_INIT DATA_INIT_COMM DATA_INIT_PID DATA_INIT_ASCII	Q output parameter.

6. A Parameterized Subroutine Block's BOOL type formal parameters may not be passed to another Parameterized Subroutine Block.
7. WORD formal parameters cannot be passed into another Parameterized Subroutine Block's NWORD input parameter.

External Blocks

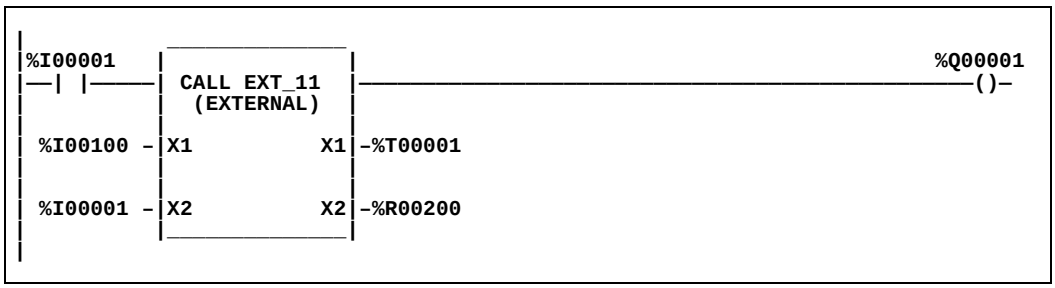
External blocks are created using the C Programmer's Toolkit. Refer to the *C Programmer's Toolkit for Series 90™ PLCs* (GFK-0646) for detailed information regarding external blocks.

How External Blocks Are Called

External blocks are added to a user program by using the Librarian function in your programming software.

An external block executes when called from the program logic in the _MAIN block or from another block. To facilitate the passing and returning of data, an external block may have 0 to 7 parameter pairs.

In the following example, if %I00001 is set, the external block named EXT_11 is executed. The block operates on the input data, located at reference addresses %I00100 – %I00111 and %I00001 – %I00016, and produces values in the block of output data, located at reference addresses %T00001 – %T00016, and at register memory addresses %R00200 – %R00201. The logic within the block can also control the Enable Out output of the external block.

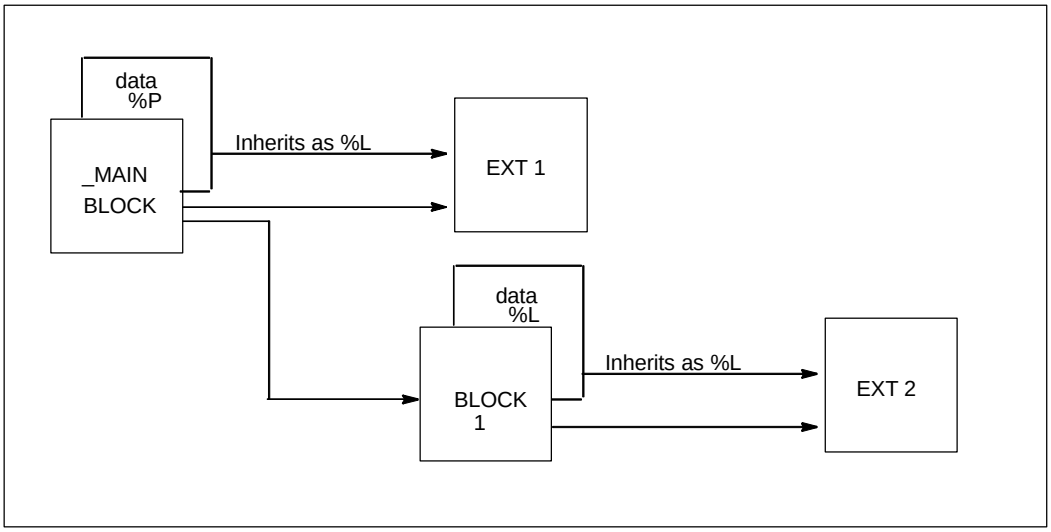


Note

Unlike other block types, external blocks cannot call any other blocks.

External Blocks and Local Data

External blocks support the use of %P global data. External blocks do not have their own %L data, but instead inherit the %L data of the calling block. External blocks also inherit %S contacts, such as FST_EXE, from the calling block. If %L references are used within an external block and the block is called by _MAIN, %L references will be inherited from the %P references wherever encountered in the external block (for example, %L0005 = %P0005).



Local Data Initialization

When an external block is stored to the PLC, a copy of most of its internal data is saved. Global and static initialized data are saved, but if static variables are declared without an initial value, the initial value is undefined and must be initialized by the C application. (Refer to the "Global Variable Initialization" and "Static Variable" parts in Section 6 of Chapter 2 of the *C Programmer's Toolkit for Series 90™ PLCs* (GFK-0646). This data is used to reinitialize the block's data area whenever the PLC transitions from Stop to Run.

External/standalone programs do not use %L data, but the internal data they use is somewhat similar to local data, as discussed above. Internal data used in a standalone program should not be confused with %L data.

Standalone C Programs

Like external blocks, standalone programs are developed using the C Programmer's Toolkit. Unlike external blocks, however, standalone C Programs can be up to 512 KB in size. Standalone C programs cannot call other standalone programs, nor can they call blocks within an LD program. Similarly, blocks within an LD program cannot call a standalone program. Instead, standalone C programs are scheduled for execution using one of several possible program scheduling modes, described in section 4 of this chapter.

A maximum of 16 standalone programs can be used at one time. If an LD program is used, only 15 standalone programs are allowed, for a total of 16 programs.

Note

Since standalone C programs are truly separate programs, they do not have access to memory types local to an LD program (%L and %P). The internal or local data used in a standalone C program should not be confused with %L and %P data.

Data Encapsulation

Each standalone C program is provided with a means of obtaining its own local copy of user data references. Instead of operating on the global set of user references, each standalone program can operate on its own local set of data. This feature is supported through the use of an input/output specification.

The following steps occur when using an I/O specification with a standalone C program:

1. When the program is scheduled for execution, any corresponding input specification is copied from the global user reference data area(s) to an area local to the program.
2. As the program executes, it can operate on its local set of input and output data. Any interruptions during the execution phase will not affect this program's local copies of input or output data.
3. When the program completes, its local output specification is copied back to the specified set of user data reference areas.

Two particular concerns are addressed by using an input/output specification. First, if a program is suspended in mid-execution and an output scan is performed before execution is resumed, output values will remain consistent since the output scan values are obtained from the user data reference locations. Second, if a program is interrupted mid-execution by another program, the first program is unaffected by changes to global data caused by the second program, since it has its own local copy of data.

Another benefit to using an input specification is to provide a more accurate sampling of input values. If a program's execution is postponed due to higher priority programs, the input specification may provide the program with a set of data that more accurately represents the state of that data when the program was scheduled since global user reference data may have been modified by higher priority programs or by the scanning of input values.

Input/Output Specifications

Unlike external blocks, standalone C and LD programs cannot have any input or output parameters. Standalone C programs may utilize an input/output specification that lists a maximum of eight input and eight output ranges. The input ranges will be copied to the program at the start of program execution. The output ranges are copied from the standalone C program to the global reference on the completion of program execution. Note that this differs from external block parameters that are passed by reference, not by value. This operation is especially important for programs that may be time-sliced over multiple sweeps, which can occur when using Microcycle Sweep mode[1].

Caution

When the PLC runs in Microcycle Sweep mode, programs can be suspended in the middle of execution, possibly in the middle of a line of C code or in the middle of a rung of logic or function block. If the program uses global references such as %Q, %R, etc., a possibly inconsistent set of reference values may be present at the time an Interrupt program or output scan occurs. This inconsistency could even be within a given reference value if the value is not accessed according to its type.

[1] Incoherent data can result if a program uses global data and is suspended across multiple sweeps. The data referenced will be from two successive sweeps. Although data cannot be incoherent within a byte or word, global data should only be accessed using its basic type (byte, word, etc.), otherwise incoherency can apply to individual elements as well.

To illustrate the possibility of inconsistent output data, consider the following example program which updates a given output value in several locations. For this example, assume that the user has configured %AQ1 as an output specification from the second program, and that user reference %AQ1 contains the value 0 when the two programs begin execution.

Differences Between Accessing Global and Local (Internal) Data

Program Accessing Global Data	
<program begins>	
%AQ1 = 0	An output scan occurring after this line of code would output a 0 to AQ1.
%AQ1 = %AQ1 + 2	An output scan occurring after this line of code would output a 2 to AQ1.
%AQ1 = %AQ1 + 12	An output scan occurring after this line of code would output a 14 to AQ1.
%AQ1 = %AQ1 / 2	An output scan occurring after this line of code would output a 7 to AQ1.
<program completes>	
Further output scans will output a 7 to AQ1.	

Program Using I/O Specification	
<program begins>	
LOC[1] = 0	An output scan occurring after this line of code would output a 0 to AQ1.
LOC[1] = LOC[1]+2	An output scan occurring after this line of code would output a 0 to AQ1.
LOC[1]=LOC[1] + 12	An output scan occurring after this line of code would output a 0 to AQ1.
LOC[1]=LOC[1]/2	An output scan occurring after this line of code would output a 0 to AQ1.
<program completes> The PLC copies the output specification from a local area back to the global areas when the program completes.	
Further output scans will output a 7 to AQ1.	

Note

Remember that standalone C programs are truly separate programs. They do not have access to memory types local to an LD program (%L and %P). The internal or "local" data used in a standalone C program should not be confused with %L and %P data.

Standalone C Programs and Local Data

Standalone C programs do not have a local data copy provided by the PLC. Similarly, they are not able to access %P memory of an LD program, nor can they access any %L memory associated with a block within an LD program. Standalone C programs do have local data that is declared within the C source file(s) used to create the standalone C program. Refer to the *C Programmer's Toolkit User's Manual* (GFK-0646) for further information.

Local Data Initialization

When a standalone C program is stored to the PLC, a copy of its internal data is saved. This data is used to reinitialize the program's data area whenever the PLC transitions from Stop to Run.

Referencing I/O Specification Data Within a Standalone C Program

Several new C macros used to define and access input and output specification data are defined for standalone C programs. Refer to the *C Programmer's Toolkit User's Manual* (GFK-0646, revision C or later) for information regarding referencing I/O specification data within a standalone C program.

Data Coherency of I/O Specifications

Since standalone C programs can be interrupted by other programs and Interrupt blocks, data incoherency within an I/O specification can occur. Each individual I/O specification is limited to 2048 bytes, for a maximum of 16KB of input data and 16KB of output data. The 90-70 PLC will ensure the following:

- Each byte within an individual I/O specification is coherent with respect to that individual specification.
- If the total length of all input specifications is no more than 2048 bytes, the entire input specification will be coherent.
- If the total length of all output specifications is no more than 2048 bytes, the entire output specification will be coherent.

If the total length of an input or output specification exceeds 2048 bytes, groups of individual specifications whose combined lengths do not exceed 2048 bytes will be coherent. The following table indicates coherency in this case.

Table 2-14. Coherency of I/O Specification

I/O specification	Length (bytes)
1	2000
2	48
Interrupts may occur	
3	1024
Interrupts may occur	
4	1026
Interrupts may occur	
5	10
6	20
7	20
8	20

Using LD vs. Standalone C Programs

Several options need to be considered when determining which type of program is to be used. The following list summarizes many of the features supported in each of the types of programs:

- Interrupt Blocks are preferred over standalone programs when interrupt latency is a concern. The overhead to process an Interrupt Standalone program is much larger than that of an Interrupt block.
- LD programs may operate only on global user reference data. This can introduce data coherency problems when the LD program is run in Microcycle Sweep mode and the program is suspended over multiple sweeps.
- Standalone C programs incur an 8K overhead per program.
- An LD program is preferred over a standalone C program when you are using large amounts of boolean instructions. LD programs are better suited for relay type logic than are standalone C programs.

Table 2-15. LD vs. Standalone C Program Tradeoffs

	Programming Language	Program Size Limit	Data Types Accessible	Local Data Size	Scheduling Modes	Data Encapsulation	Block Types Supported
LD	Ladder Logic	smaller of 512K or available memory size, organized into16K blocks.	All	8k %P, 8K %L per block	All*	No	LD SFC PSB External
Standalone	C, using the C Programmer's Toolkit.	smaller of 512K or available memory size	All except %P, %L	unlimited, counts as part of program size	All *	Yes	n/a

*Using Microcycle Sweep mode with an LD program is not recommended. LD programs always operate on global data directly, which can lead to inconsistent output values if the LD program is suspended mid-execution. Refer to section 4 of this chapter for more information on Microcycle Sweep mode.

Differences in Operation: LD and Standalone C Programs

Retentiveness of Data

When only standalone C programs are used, the retentive nature of data is based solely on memory type since there are no coil instructions. In this case %Q and %M are retentive. If both LD and standalone C programs are used, the retentive property of memory types is driven by their use in the LD program. For more information about retention of logic and data, refer to the "Retentiveness of Logic and Data" discussion in section 2 of this chapter. For more information on retentive properties of specific memory types, refer to the Table 2-3.

Global Data

LD programs only have access to global data areas since they do not have the ability to use input/output specifications. This can lead to inconsistent output values if an LD program is used in Microcycle Sweep mode.

Interrupt Execution

Interrupt blocks within the LD program have the highest priority in the system. In addition, they cannot be preempted, while standalone C programs can be.

Queuing of Interrupts

The 90-70 PLC can queue the invoking of Interrupt blocks within the LD program. A standalone C program triggered by an interrupt will not be queued should another interrupt occur during the processing of the first interrupt. In this case a fault will be logged in the PLC fault table.

System Status References

The following differences exist when using System Status References* (called “Convenience References” in previous editions of 90-70 Reference manual):

- The reference FST_EXE is not available to standalone C programs.
- The reference FST_SCN does not refer to %S0001 within standalone C programs. Instead, a macro is provided by the C Toolkit to provide identical functionality.

*For information on System Status References, refer to Table 2-9 as well as section 2 of this chapter.

Section 4: PLC Sweep Modes and Program Scheduling Modes

Normal Sweep Mode

In Normal Sweep mode, each PLC sweep can consume a variable amount of time. The Logic Window is executed in its entirety each sweep. The Communications and Background Windows can be set to execute in a Limited or Run-to-Completion mode. Normal Sweep is the most common sweep mode used for PLC applications.

The following figure illustrates three successive PLC sweeps in Normal Sweep mode. Note that the total sweep times may vary due to sweep-to-sweep variations in the Logic Window, Communications Windows, and Background Window.

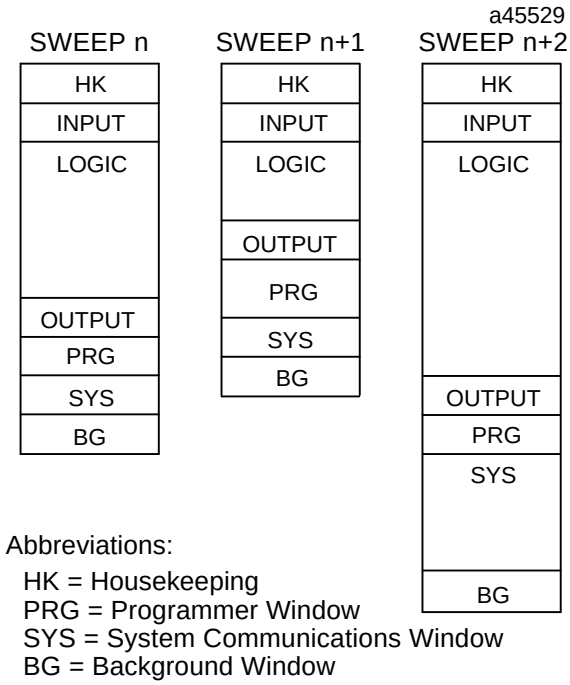


Figure 2-3. Typical Sweeps in Normal Sweep Mode

Constant Sweep Mode

In Constant Sweep mode, each PLC sweep begins at a specified Constant Sweep time after the previous PLC sweep began. The Logic Window is executed in its entirety each sweep. If there is sufficient time at the end of the sweep, the PLC will alternate among the Programmer Communications, System Communications, and Background Windows, allowing them to execute in Run-to-Completion mode until it is time for the next sweep to begin. Some or all of the Communications and Background Windows may not be executed. The Communications and Background Windows will terminate when the overall PLC sweep time has reached the value specified as the Constant Sweep time.

One reason for using Constant Sweep mode is to ensure that I/O are updated at constant intervals.

The value of the Constant Sweep timer can be configured to be any value from 3 to 255 milliseconds. The Constant Sweep timer value may also be set and Constant Sweep mode may be enabled or disabled by your programming software or by the user program using Service Request function #1. The Constant Sweep timer has no default value; a timer value must be set prior to or at the same time Constant Sweep mode is enabled.

If the PLC sweep exceeds the Constant Sweep time in a given sweep, the PLC places an oversweep alarm in the PLC fault table and sets the OV_SWP (%SA0002) status reference at the beginning of the next sweep. The OV_SWP status reference is reset when the time of the last sweep does not exceed the Constant Sweep timer or the PLC is not in Constant Sweep mode. Additional sweep time due to an oversweep condition in a given sweep does not affect the time given to the next sweep.

The following figure illustrates four successive PLC sweeps in Constant Sweep mode with a Constant Sweep time of 100 milliseconds. Note that the total sweep time is constant, but an oversweep may occur due to the Logic Window taking longer than normal.

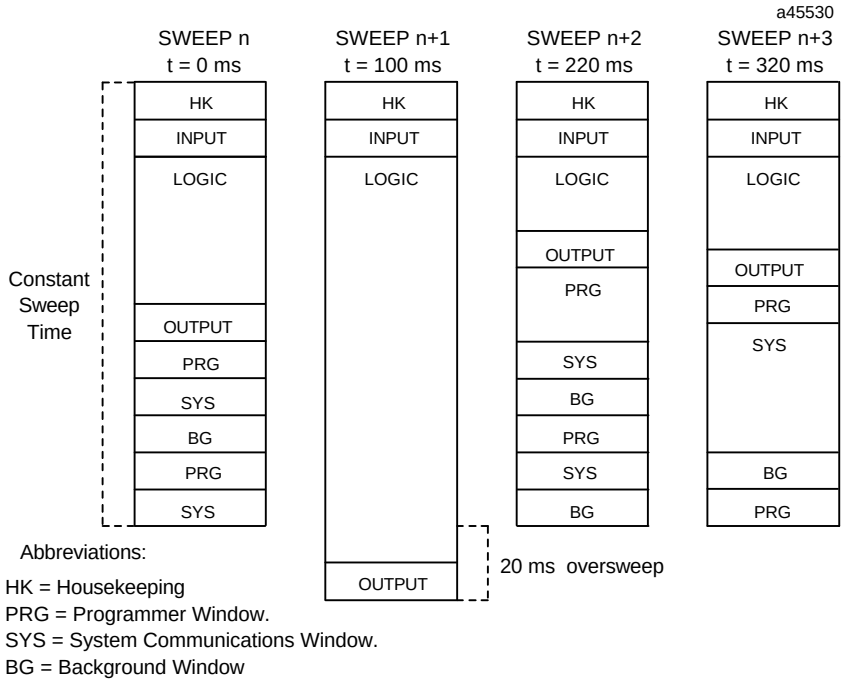


Figure 2-4. Typical Sweeps in Constant Sweep Mode

Constant Window Mode

In Constant Window mode, each PLC sweep can consume a variable amount of time. The Logic Window is executed in its entirety each sweep. In this mode, the PLC will alternate among the three windows, allowing them to run in a Run-to-Completion mode for a time equal to the value set for the Constant Window timer. The overall PLC sweep time is equal to the time required to execute the Housekeeping, Input Scan, Logic Window, and Output Scan phases of the sweep plus the value of the Constant Window timer. This time may vary due to sweep-to-sweep variances in the execution time of the Logic Window.

An application that requires a certain amount of time between the Output Scan and the Input Scan, permitting inputs to settle after receiving output data from the program, would be ideal for Constant Window mode.

The value of the Constant Window timer can be configured to be any value from 5 to 255 milliseconds. The Constant Window timer value may also be set by your programming software or by the user program using Service Request functions #3, #4, and #5.

The following figure illustrates three successive PLC sweeps in Constant Window mode. Note that the total sweep times may vary due to sweep-to-sweep variations in the Logic Window, but the

time given to the Communications and Background Windows is constant. Some of the Communications or Background Windows may be skipped, suspended, or run multiple times based on the Constant Window time.

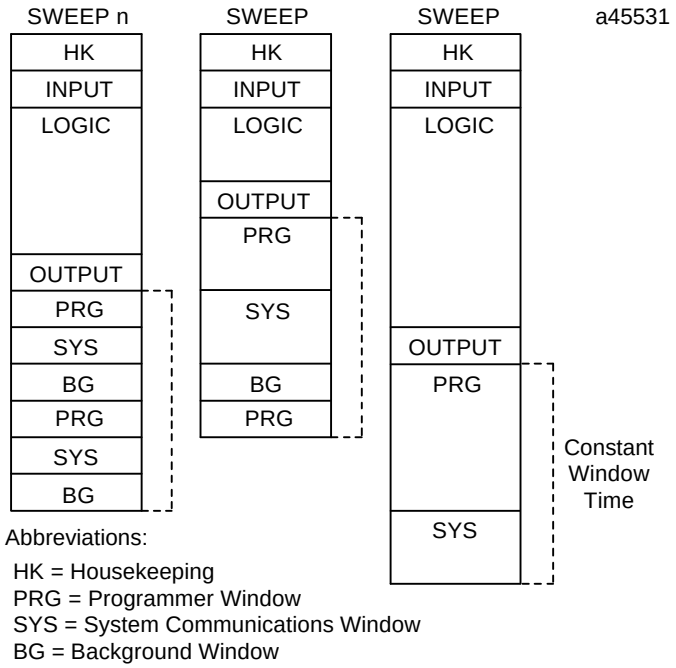


Figure 2-5. Typical Sweeps in Constant Window Mode

Microcycle Sweep Mode

In Microcycle Sweep mode, each PLC sweep begins at an absolute time— which is a multiple of the base cycle time—relative to the Stop-to-Run transition of the PLC. The base cycle time specifies how long each sweep should take (similar to the Constant Sweep time in Constant Sweep mode). The user programs are scheduled for execution each sweep based on their period and may execute in a time-sliced fashion over multiple PLC sweeps. The PLC will alternate between the Communications and Background Windows, allowing them to run in a Run-to-Completion mode until it is time for the next sweep to begin.

Microcycle Sweep mode can be used to allow some programs to execute more often than others. This allows more processing time to be applied to the more important or more time-critical tasks. Microcycle Sweep mode also allows programs to execute more in line with the time when their inputs are available.

Although Microcycle Sweep mode has a fixed sweep time, it is significantly different from Constant Sweep mode. First, user programs do not necessarily execute in their entirety each sweep. In order to maintain the base cycle time and the Communications and Background Window times, user programs may be suspended during execution and resumed the following PLC sweep. Also, additional sweep time due to an oversweep condition in a given sweep causes

the next sweep to be shortened by the oversweep time. In this way each PLC sweep (with the exception of sweeps which follow an oversweep condition) begins at an absolute time relative to the Stop-to-Run transition of the PLC. Finally, the Communications and Background Windows are guaranteed to run for at least the specified Window time each sweep. The Logic Window will be suspended, if necessary, to guarantee that the Communications and Background Windows get to run for the specified Window time.

The base cycle time and the window timer value can be configured with a base cycle time of between 5 and 2550 milliseconds. The Constant Window timer can be any value from 5 to 255 milliseconds. The base cycle time and Constant Window timer may also be set while the PLC is in Stop mode. The base cycle time and window timer cannot be changed while the PLC is in Run mode.

In Microcycle Sweep mode, Periodic programs execute on a priority basis. Periodic programs have priority inverse to their period (smallest period has highest priority). Refer to the "User Program Execution" discussion later in this section for more information on Periodic programs and their execution.

If the PLC sweep exceeds the base cycle time in a given sweep, the PLC places an oversweep alarm in the PLC fault table and sets the OV_SWP (%SA0002) status reference at the beginning of the next sweep. The OV_SWP status reference is reset when the last sweep time does not exceed the base cycle time. Sweep time due to an oversweep condition in a given sweep causes the next sweep to be shortened by the oversweep time.

The following figure illustrates three successive PLC sweeps in Microcycle Sweep mode with a base cycle time of 100 milliseconds. Note that the sweep time is constant and the Communications and Background Windows are guaranteed to run for the configured window timer. In sweep n and sweep n+1, the Logic finishes early; an I/O-Triggered program can execute during that time. In sweep n+2, the Logic Window is not complete and is suspended so that the Communications and Background Windows can run for the specified window time. In each case the logic window stays open for the entire time allowed for logic.

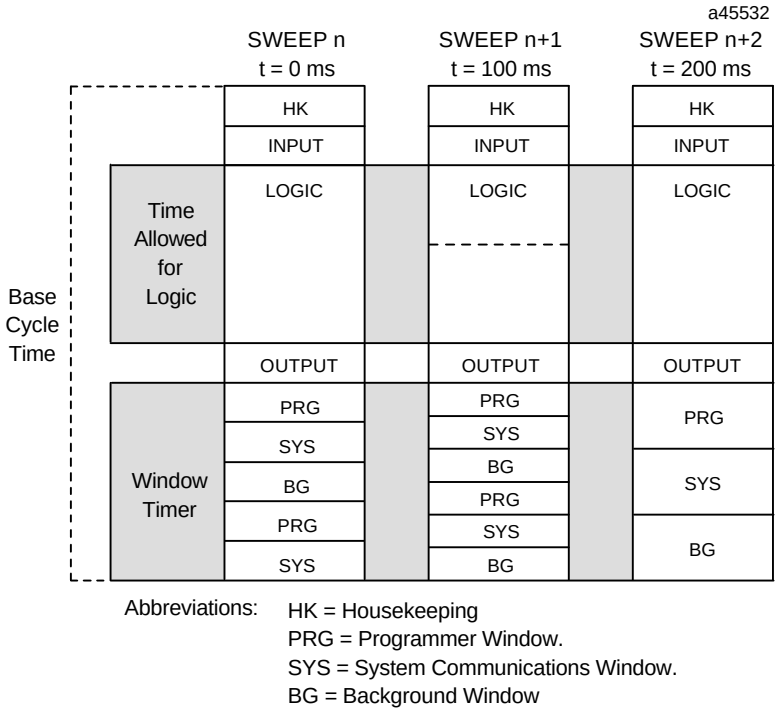


Figure 2-6. Typical Sweeps in Microcycle Sweep Mode

Note

Run Mode Store of logic is not supported in Microcycle Sweep mode. Also, the Single Sweep Debug feature is not supported in Microcycle Sweep mode.

Program Scheduling Modes

Each user program in the 90-70 PLC can execute, subject to sweep mode restrictions, in one of four program scheduling modes. This section will briefly describe the following four available program scheduling modes:

- Ordered
- Timed
- I/O-Triggered
- Periodic

Ordered Ordered programs are executed in the Logic Window with all other Ordered programs. Ordered programs are executed once per sweep in the sequence in which they are declared in the programming software. Ordered programs are not supported in Microcycle Sweep Mode.

Timed Timed programs are scheduled to execute on a specified time interval with an initial delay (if specified) applied on Stop-to-Run transition of the PLC. Timed programs are scheduled to execute on a priority basis during any phase of the PLC sweep. Timed programs are not supported in Microcycle Sweep mode.

I/O-Triggered I/O-Triggered programs are scheduled to execute on the receipt of a configured I/O Interrupt. I/O-Triggered programs are scheduled to execute on a priority basis during any phase of the PLC sweep when the PLC is in Normal Sweep, Constant Sweep, or Constant Window Sweep mode.

In Microcycle Sweep mode, I/O-Triggered programs are scheduled to execute on a priority basis in the Logic Window. In this case, the execution of I/O-Triggered programs may be time-sliced over multiple sweeps.

Periodic Periodic programs are scheduled for execution based on the user-configured period for the program and are scheduled to execute in the Logic Window with all other Periodic programs. Periodic programs execute on a priority basis relative to all other programs and may be time-sliced over multiple sweeps. Periodic programs are only supported in Microcycle Sweep mode.

Choosing PLC Sweep and Program Scheduling Modes

The table shown below indicates the availability of each program scheduling mode in each of the available PLC sweep modes.

Table 2-16. Available Program Scheduling Modes in Each PLC Sweep Mode

Sweep Mode	Program Scheduling Mode				Interrupt Blocks
	Ordered	Timed	I/O Triggered	Periodic	
Normal	Yes	Yes	Yes	No	Yes
Constant	Yes	Yes	Yes	No	Yes
Constant Window	Yes	Yes	Yes	No	Yes
Microcycle	No	No	Yes *	Yes	Yes

* Executes in Logic Window only.

User Program Execution

User Program Priorities

The priority of a user program specifies its priority relative to other programs. Higher priority programs execute before lower priority programs. If two or more programs with the same priority are scheduled at the same time, the order of execution is undefined. Programs can be suspended in the middle of execution by higher priority programs and Interrupt blocks.

Ordered programs all have the same priority and are executed in the order that you specify in the programming software. Ordered programs have lower priority than Timed programs, I/O-Triggered programs, and Interrupt blocks.

Periodic programs have priority inverse to their period (smallest period has highest priority). The order of execution of Periodic programs with the same period is undefined. Periodic programs have lower priority than Timed programs, I/O-Triggered programs, and Interrupt Blocks.

Timed and I/O-Triggered programs have higher priority than Ordered and Periodic programs. The priority of a Timed or I/O-Triggered program specifies its priority relative to other Timed and I/O-Triggered programs. The priority range 10–99 (10 being the highest priority) is reserved for Timed and I/O-Triggered programs which can run during any phase of the PLC sweep (that is, not restricted to running in the Logic Window). Timed and I/O-Triggered programs operate this way when the PLC is running in Normal Sweep, Constant Sweep, or Constant Window mode. In Microcycle Sweep mode, I/O-Triggered programs are executed in the Logic Window, and priorities of 100–109 (100 being the highest priority) are reserved for this mode.

Timed and I/O Interrupt blocks have the highest priority of any user logic.

Table 2-17. Priority Values for Timed and I/O-Triggered Programs

Scheduling Mode	Sweep Mode	
	Normal Sweep Constant Sweep Constant Window	**Microcycle Sweep**
Ordered *	Executed in order declared in the programming software.	Not supported.
Periodic *	Not supported.	Smallest period has highest priority.
I/O-Triggered	10–99	100–109
Timed	10–99	Not supported.

* Ordered and Periodic scheduling modes have lower priority than Timed and I/O-Triggered scheduling modes.

User Program Execution in Normal Sweep, Constant Sweep, and Constant Window Modes

In Normal Sweep, Constant Sweep, and Constant Window modes, the 90-70 PLC can execute Ordered, Timed, and I/O-Triggered programs as well as Timed and I/O Interrupt blocks.

Ordered programs execute in their entirety once per sweep in the Logic Window. The programs execute in the order in which they are declared in the programming software. The input specification is copied prior to execution of the program, and the output specification is copied upon completion of the program. In this way, the output of one program can be used as input for the next, if desired.

The following figure depicts two Ordered programs (A and B) executing in a typical PLC sweep in Normal Sweep mode.

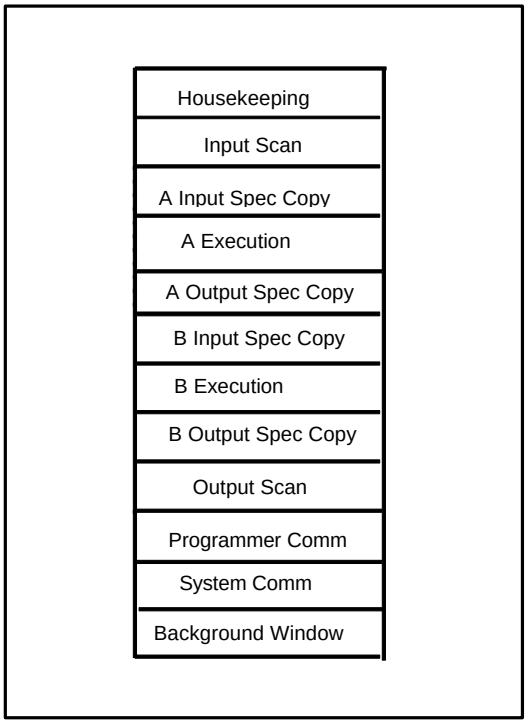


Figure 2-7. Ordered Program Execution Sequence

In Normal Sweep, Constant Sweep, and Constant Window mode, Timed and I/O-Triggered programs execute during any phase of the PLC sweep. These programs will preempt the execution of Ordered programs and lower priority Timed and I/O-Triggered programs. The input specification is copied at the time the program is scheduled to execute (that is, when the time interval expires or the I/O Interrupt occurs). The output specification is copied upon completion of the program.

Timed and I/O Interrupt blocks execute during any phase of the PLC sweep. These blocks will preempt the execution of all programs and have the highest priority of any user logic in the PLC. Timed and I/O Interrupt blocks do not have an input or output specification copy.

The following figure depicts 2 Ordered programs (A and B) , an I/O-Triggered program (C) with priority 10, a Timed program (D) with priority 20, and an I/O Interrupt block all executing in a typical PLC sweep in Normal Sweep mode.

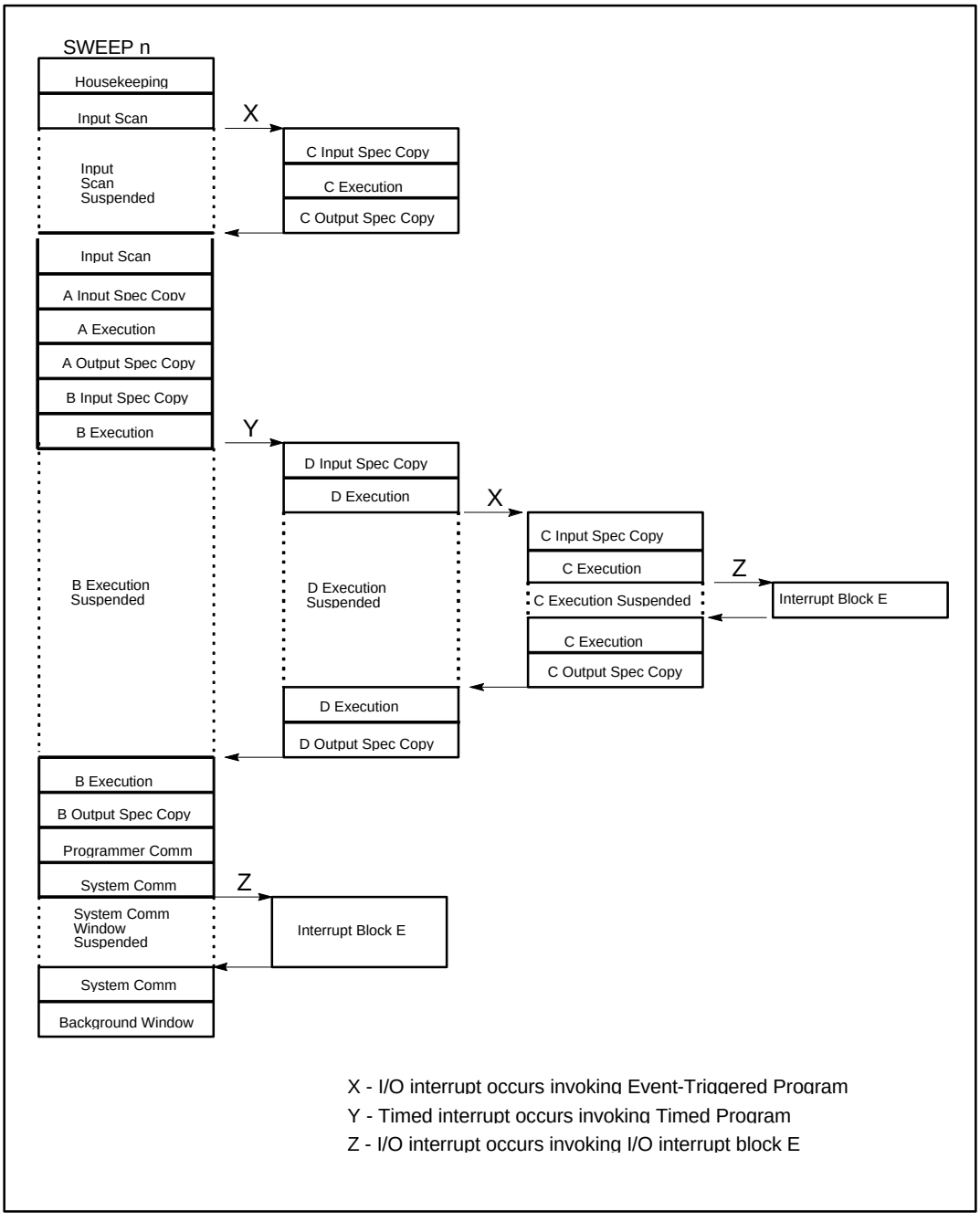


Figure 2-8. Ordered, Timed, I/O-Triggered and Interrupt Block Execution Sequence

User Program Execution in Microcycle Sweep Mode

In Microcycle Sweep mode, the 90-70 PLC can execute Periodic and I/O-Triggered programs as well as Timed and I/O Interrupt blocks.

Periodic programs execute in the Logic Window. These programs are scheduled to execute based on the program's period. For example, a program with a period of 1 will be scheduled to execute every PLC sweep and a program with a period of 2 will be scheduled to execute every other PLC sweep. Periodic programs have priority inverse to their period (smallest period has highest priority). These programs are subject to time-sliced execution over multiple sweeps based on the time available to the Logic Window. Unlike Ordered programs, the input specification is copied at the beginning of the Logic Window for all Periodic programs that are scheduled to begin execution in a given sweep. In other words, all input specification copies will occur for Periodic programs before any of the Periodic programs begin or continue executing. The output specification is copied upon completion of the program.

The following figure depicts two Periodic programs (A and B) executing in a typical PLC sweep in Microcycle Sweep mode.

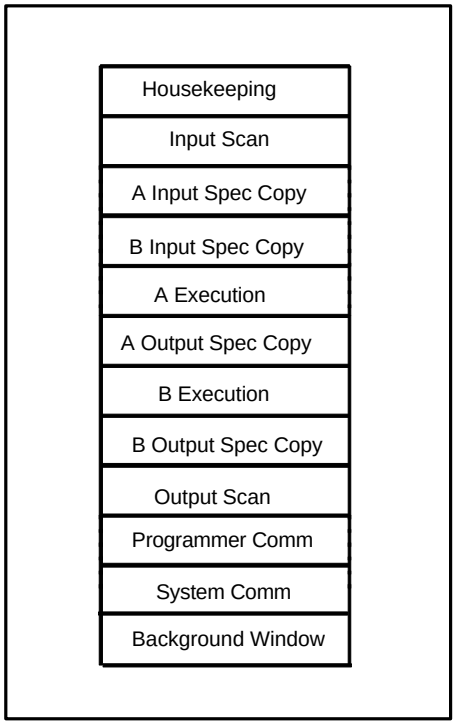


Figure 2-9. Periodic Program Execution Sequence

Unlike other sweep modes, I/O-Triggered programs execute in the Logic Window only when the PLC is in Microcycle Sweep mode. If the I/O Interrupt occurs during or prior to the end of the Logic Window, the I/O-Triggered program will be scheduled to execute in the Logic Window of the current PLC sweep. Otherwise, it will be scheduled to execute in the Logic Window of the next PLC sweep. I/O-Triggered programs will preempt the execution or resumption of Periodic programs and lower priority I/O-Triggered programs. These programs are subject to the same time-sliced execution over multiple sweeps as Periodic programs, based on the time available to the Logic Window. The input specification for an I/O-Triggered program is copied at the time the

program is scheduled to execute (that is, when the I/O interrupt occurs) not at the beginning of the Logic Window as with Periodic programs. The output specification is copied upon completion of the program.

Timed and I/O Interrupt blocks execute during any phase of the PLC sweep when the PLC is in Microcycle Sweep mode. These blocks will preempt the execution of all programs and have the highest priority of any user logic in the PLC. Timed and I/O Interrupt blocks do not have an input or output specification copy.

The following figure depicts two Periodic programs (A and B) and one I/O-Triggered program (C) executing in two successive Microcycle Sweeps. Periodic programs A and B both have a period of 1.

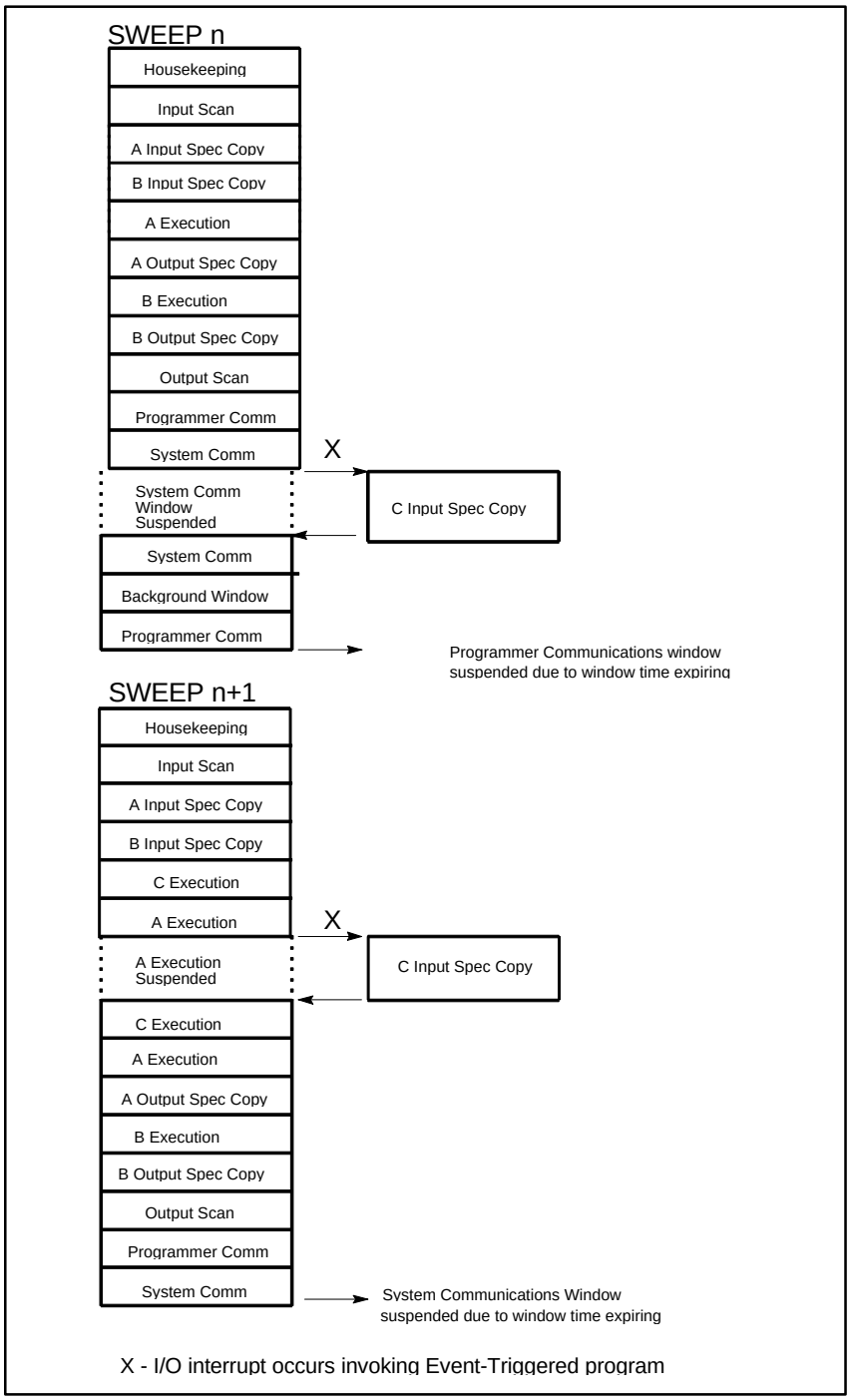


Figure 2-10. Periodic and I/O-Triggered Execution Sequence

Global Data in Microcycle Sweep Mode

Incoherent data can result if a program uses global data (%R, %I, %Q, etc.) that is suspended across multiple sweeps. The data referenced will be from two successive sweeps. Although data cannot be incoherent within a byte or word, global data should only be accessed using its basic type (byte, word, etc.); otherwise, incoherency can apply to individual elements as well. If possible, the input and output specifications should be used to access and update global data areas.

Interrupt Handling

There are two types of interrupts available for user program handling in the 90-70 PLC.

I/O Interrupts These interrupts are generated by 90-70 I/O modules to indicate discrete input state changes (rising/falling edge), analog range limits (low/high alarms), and high speed signal counting events.

Timed Interrupts These interrupts are generated by the 90-70 PLC CPU based on a user-specified time interval with an initial delay (if specified) applied on Stop-to-Run transition of the PLC.

Both of these types of interrupts may invoke a user program or block.

Caution

Interrupt blocks and programs can interrupt the execution of non-interrupt logic as well as other Timed and I/O-Triggered programs. Therefore, unexpected results may occur if the interrupting logic and interrupted logic access the same data. If necessary, Service Request #17 or Service Request # 32 can be used to temporarily mask I/O and Timed Interrupt blocks and programs from executing when shared data is being accessed.

Interrupt Handling and Scheduling with Blocks

An Interrupt block has the highest priority of any user logic in the system and may be programmed to execute upon the receipt of a Timed or I/O Interrupt block. The execution of a block triggered from a timed or I/O interrupt preempts the execution of the normal PLC sweep activities. Execution of the normal PLC sweep activities is resumed after the Interrupt block completes. There can be a maximum of 64 I/O Interrupt blocks and 16 Timed Interrupts blocks.

Note

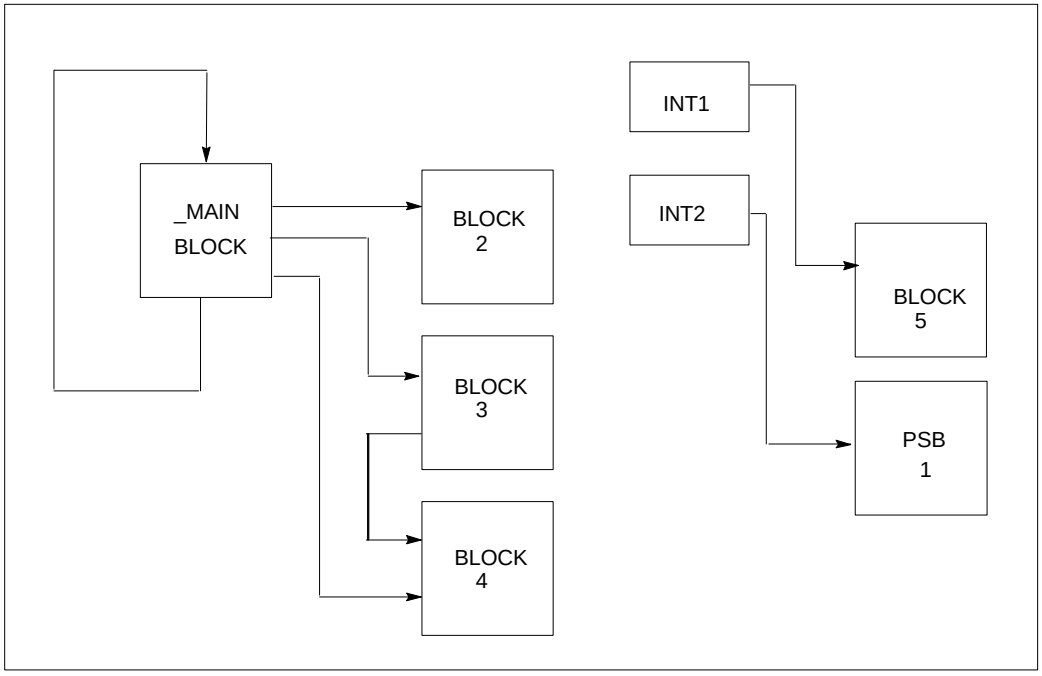
Timer function blocks do not accumulate time if used in a block that is executed as a result of a Timed or I/O Interrupt block.

Beginning with Release 6 of the PLC CPU, LD Interrupt blocks may make calls to other blocks. The application stack used during the execution of Interrupt blocks is different from the stack used by the LD program. Therefore, the nested call limit is different from the limit described for calls from the _MAIN block. The PLC will log an ”Application Stack Overflow” fault and the PLC will transition to Stop/Fault mode if a call results in insufficient stack space to complete the call.

Note

Blocks which may execute as a result of a timed or I/O interrupt should not be called from the _MAIN block or other Non-Interrupt blocks because portions of the code executed by blocks are not re-entrant. In the example below INT1,

INT2, BLOCK5, and PSB1 should not be called from _MAIN, BLOCK2, BLOCK3, or BLOCK4.



I/O Interrupt Blocks

A block may be triggered by an interrupt input from certain hardware modules. For example, on the 32-Circuit 24 VDC Input Module (IC697MDL650), the first input can be configured to generate an interrupt on either the rising or falling edge of the input signal. If the module is configured in this manner, that input can serve as a trigger to cause the execution of an LD or External block.

To program an I/O Interrupt block, the block must first be declared in the programming software. It must then be associated with the interrupt through the use of an interrupt declaration.

The figure below shows two I/O interrupt declarations. The trigger ST_BUT calls LD block INT1 if the input from a stop button wired to input 1 transitions in the configured direction. The module can be configured to generate the interrupt on a rising edge or a falling edge of the input. The LD block INT2 is triggered by %AI00009.

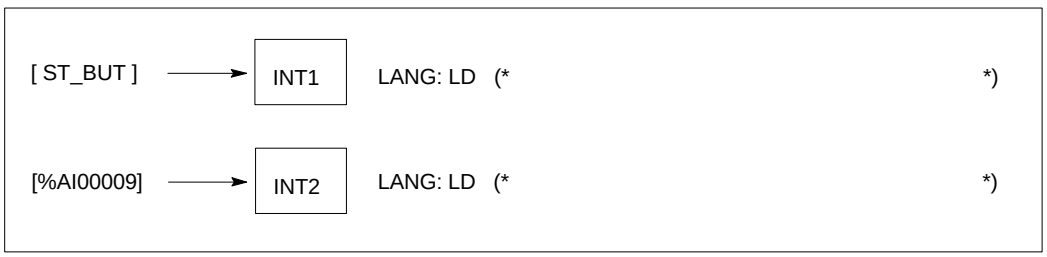


Figure 2-11. I/O Interrupt Block Declarations

Note

Parameterized subroutine blocks (PSBs) with zero parameters and External blocks (C blocks and C FBKs) with zero parameters may also be triggered by an interrupt input. (Zero parameter subroutine blocks are called SUBRs in Control.) For these types of blocks, the local data (%L) is inherited from the _MAIN local data (%P), for example, %L0005 = %P0005.

Timed Interrupt Blocks

A block may be executed on a user-specified time interval with an initial delay (if specified) applied on a Stop-to-Run transition of the PLC. In Control, the time base is 0.001 second (abbreviated *msec* within the Task Definitions dialog box).

To program a Timed Interrupt block, the block must first be declared. It must then be associated with a timed interrupt and given an interval and initial delay through the use of an interrupt declaration.

The first execution of a Timed Interrupt block will occur at ((DELAY * time base) + (INTVL * time base)) after the PLC is placed in Run mode. The figure below shows two timed interrupt declarations. The LD block BLK1 will be executed at times of 3 seconds, 5 seconds, 7 seconds, etc., after the PLC is placed in Run mode. The LD block BLK2 will be executed at two-second intervals, beginning two seconds after the PLC is placed in Run mode. The absence of a DELAY value for BLK2 indicates that there will not be an initial delay in the first execution of the block.

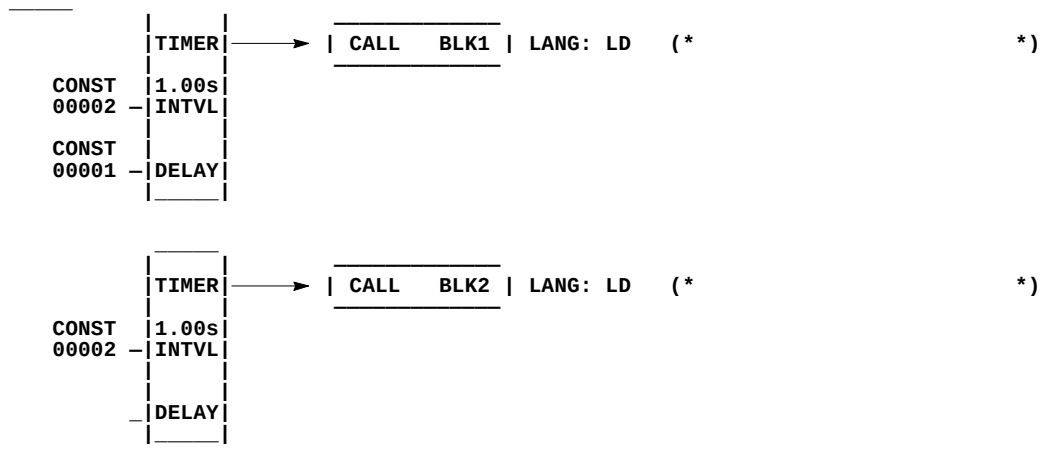


Figure 2-12. Timed Interrupt Block Declarations

Parameters:

Parameter	Description
INTVL	INTVL is a constant value that will be multiplied by the time base of the interrupt to establish the frequency of execution of the associated block.
DELAY	DELAY is an optional field for the timed interrupt. It is a constant value that will be multiplied by the interrupt time base to establish an additional delay for the first execution of the associated block.

Valid Memory Types:

Parameter	flow	%I	%Q	%M	%T	%S	%G	%U	%R	%P	%L	%AI	%AQ	%UR	const	none
INTVL															•	
DELAY															•	•

• = Valid data type, or place where power may flow through the function.

Note

Parameterized Subroutine Blocks (PSBs) with zero parameters and External blocks (C blocks and C FBKs) with zero parameters may also be triggered by a timed interrupt. (Zero parameter subroutine blocks are called SUBRs in Control.) For these types of blocks, the local data (%L) is inherited from the _MAIN local data (%P), for example, %L0005 = %P0005.

Interrupt Handling and Scheduling with User Programs

I/O-Triggered Programs

Beginning with Release 6 of the PLC CPU, one of the scheduling modes available for user programs is the activation of programs from an I/O interrupt. The I/O-Triggered scheduling mode allows a user program to be invoked, along with its corresponding input and output specification copy, when a configured I/O interrupt occurs. I/O-Triggered programs execute during any phase of the PLC sweep or only during the Logic Window, based on sweep mode. Refer to “User Program Execution” earlier in this section for more information on the scheduling and execution of I/O-Triggered programs.

To program an I/O-Triggered program, the program must first be declared in the programming software. The scheduling mode must then be set to I/O Interrupt Triggered.

Timed Programs

Beginning with Release 6 of the PLC CPU, one of the scheduling modes available for user programs is the activation of a program from a timed interrupt. The Timed scheduling mode allows a user program to be executed, along with its corresponding input and output specification copy, on a user specified time interval with an initial delay (if specified) applied on a Stop-to-Run transition of the PLC. Timed programs execute during any phase of the PLC sweep. (Refer to the "User Program Execution" earlier in this section for more information on the execution of Timed programs.)

To program a Timed program, the program must first be declared. The scheduling mode must then be set to Timed Interrupt.

The time base options for Timed programs are specified in milliseconds. The first execution of a Timed program will occur at [(Initial Delay) + (Time Interval * time base)] milliseconds after the PLC is placed in Run mode.

Note

Standalone programs execute on a priority basis. The user controls the setting of the priorities in I/O-Triggered and Timed scheduling modes via the chapter programming software.

Interrupt Blocks vs. Interrupt Programs

There are important differences to be aware of when choosing a program instead of a block to handle an interrupt. When a block is selected to handle a Timed or I/O Interrupt, the block will execute immediately upon receipt of the interrupt and run until it completes. Interrupt blocks can execute during any phase of the PLC sweep, regardless of the current PLC sweep mode. Pending Timed Interrupt blocks will execute before pending I/O Interrupt blocks, but once an Interrupt block (Timed or I/O) begins executing, it will run until it completes. If an interrupt occurs which attempts to execute a Timed or I/O Interrupt block which has not fully completed execution due to a previous interrupt, the interrupt will be queued and the block will be executed again after the Interrupt block completes execution. If an Interrupt block has already been queued in this manner once, any additional interrupts that occur for this block will be ignored.

Upon receipt of the Interrupt, Timed or I/O-Triggered programs are immediately scheduled to begin execution (including the copying of the input specification). However, the actual execution of the program occurs on a priority basis. Unlike Interrupt blocks, the execution of Timed or I/O-Triggered programs can be delayed or preempted by other Timed or I/O-Triggered programs of a higher priority as well as other Interrupt blocks. Additionally, if an interrupt occurs which attempts to schedule a Timed or I/O-Triggered program which has not fully completed execution due to a previous interrupt, a “Program not Readied” application fault will be logged in the PLC fault table and the interrupt will be ignored.

When the PLC is in Normal Sweep, Constant Sweep, or Constant Window mode, Interrupt programs can execute during *any* phase of the PLC sweep. When the PLC is in Microcycle Sweep mode, Interrupt programs are scheduled to execute in the Logic Window.

In summary, the primary differences between Interrupt blocks and Interrupt programs are as follows:

Interrupt Block

- Executed immediately upon receipt of interrupt
- Cannot be preempted by other logic once Interrupt block begins execution
- One additional interrupt is “queued” if the block is still executing due to a previous interrupt
- Executes during any phase of the PLC sweep

Interrupt Program

- Scheduled to execute on a priority basis
- Can be preempted by higher priority Interrupt program or Interrupt block
- Additional interrupts are ignored and a fault is logged if the program is still executing due to a previous interrupt
- Executes during any phase of the PLC sweep when PLC is in Normal Sweep, Constant Sweep, or Constant Window mode
- Executes in Logic Window when PLC is in Microcycle Sweep mode

Section 5: Run/Stop Operations

Modes of Operation

Four run/stop modes of operation are supported by the 90-70 PLC. You can change these modes in the following ways: the toggle switch, programming software, LD function blocks, and system calls from C applications. Switching to and from various modes can be restricted based on privilege levels, position of the PLC toggle switch, passwords, etc.

Run/Outputs Enabled — In this mode, the PLC runs user programs and continually scans inputs and updates physical outputs, including Genius and Field Control outputs. The Programmer and System Communications Windows are run in either Limited, Run-to-Completion, or Constant mode.

Run/Outputs Disabled — In this mode, the PLC runs user programs and continually scans inputs, but updates to physical outputs, including Genius and Field Control, are not performed. Physical outputs are held in their configured default state in this mode. The Programmer and System Communications Windows are run in either Limited, Run-to-Completion, or Constant mode.

Stop/IO Scan — In this mode the PLC does not run user programs, but the inputs and outputs are scanned. The Programmer and System Communications Windows are run in Run-to-Completion mode. The Background Window is limited to 10 ms.

Stop/No IO Scan — In this mode the PLC does not run user programs, and the inputs and outputs are not scanned. The Programmer and System Communications Windows are run in a Run-to-Completion mode. The Background Window is limited to 10 ms.

Note

You cannot store changes to %P and %L references in Run Mode unless the %P and %L references are the first of their type in the block being stored or the block being stored is a totally new block.

Stop/IO Scan mode is **not** supported in Microcycle Sweep mode.

Mode Transitions

Stop-to-Run Transition

Several operations are performed by the CPU on Stop-to-Run transition. These operations include the following:

- Validation of sweep mode and program scheduling mode selections
- Validation of references used by programs with the actual configured sizes
- Reinitialization of data areas for external blocks and standalone C programs
- Clearing of non-retentive memory

Run-to-Stop Transition

Wind-Down Period for Microcycle Sweep Mode

When the PLC is running in Microcycle Sweep mode (refer to section 4 for information about Microcycle Sweep mode), a wind-down or logic solution period may occur after the PLC is commanded to Stop mode. This wind-down period is equal to the amount of time that the currently executing program(s) take to complete their execution unless that amount of time exceeds 2.5 seconds. If the currently executing programs exceed 2.5 seconds in their attempt to complete their executions, a fault will be logged in the PLC fault table, and the CPU will complete its transition to Stop mode. During the wind-down period, no additional programs (including Interrupt programs and blocks) will be scheduled for execution. Input Scans, Output Scans, Communications Windows (Programmer and System), and the Background Window continue during the wind-down period.

Note

By definition, exceeding the CPU wind-down period means that not all programs completed execution prior to the PLC going to Stop mode. Furthermore, when the PLC is next commanded to Run mode, all programs will begin execution at their normal beginning point. Program(s) are **not** resumed at their “wind-down exceeded” execution point.

Section 6: Power-Up and Power-Down Sequences

Power-Up

System power-up consists of the following parts:

- Power-up self-test
- PLC memory validation
- System configuration
- Option module self-test completion
- Option module dual port interface tests
- I/O system initialization

Power-Up Self-Test

On system power-up, many modules in the system perform a power-up diagnostic self-test. Series 90-70 PLC modules execute hardware checks and software validity checks. Intelligent option modules perform setup and verification of onboard microprocessors, software checksum verification, local hardware verification, and notification to the CPU of self-check completion. Any failed tests are queued for reporting to the CPU during the system configuration portion of the cycle.

In the CPU, power-up will be either a quick power-up (a warm start) or a full power-up (a cold start), depending on whether the CPU is able to go to Run mode after powering up. If all the conditions are met for the CPU to go to Run mode—a valid program or configuration is present, the switch is in Run mode, and no fatal fault exists—then the CPU will perform a quick power-up. If any of the conditions are not met, a full power-up is performed.

A quick power-up will only perform the CPU processor and BCP (Boolean Coprocessor) tests, along with a minimal RAM test. The goal of a quick power-up is to get the CPU up and running as quickly as possible. The remaining tests, ROM CRC, exhaustive memory tests, and peripheral tests are only performed on a full power-up.

If a low battery indication is present, then a low battery fault is logged into the PLC fault table.

PLC Memory Validation

The next phase of system power-up is the validation of the PLC memory within the CPU. First, the system verifies that the battery is not low and that battery-backed RAM areas are still valid. A known area of battery-backed application RAM is checked to determine if data was preserved. Next, if a ladder diagram program exists, then a checksum is calculated across the _MAIN ladder block. If no ladder diagram program exists, then a checksum is calculated across the smallest standalone C program.

When the system is sure that the application RAM is preserved, then a known area of the BCP (Boolean Coprocessor) bit cache area is checked to determine if the BCP bit cache data was preserved. If this test passes, then the Bit Cache memory is left containing its power-up values. (Non-retentive outputs are cleared on a transition from Stop to Run mode.) If this checksum does not compare or the retentive test on the application RAM fails, the Bit Cache memory is assumed to be in error and all areas are cleared. The PLC is now in a cleared state, the same as if a new CPU module were installed. All logic and configuration files must be stored from the programmer to the PLC.

System Configuration

After completing its own self-test, the CPU performs the system configuration. It first clears all of the system diagnostic bits in the BCP (Boolean Coprocessor) Bit Cache memory. This prevents faults that were present before power-down, but are no longer present, from accidentally remaining as faulted. Then it polls each module in the system, checking for completion of the module's self-test.

The CPU reads information from each module, comparing it with user provided rack/slot configuration information. Any differences between actual configuration and user specified configuration are logged in the fault tables.

Option Module Self-Test Completion

Option modules may take a longer time to complete their self-tests than the CPU due to the time required to test communications media or other interface devices. As an option module completes its initial self-tests, it tells the CPU the time required to complete the remainder of these self-tests. During this time, the CPU provides whatever additional information the module needs to complete its self-configuration, and the module continues self-tests and configuration. If the module does not report back in the time it specified, the CPU marks the module as faulted and makes an entry in one of the fault tables. When all self-tests are complete, the CPU obtains reports generated during the module's power-up self-test and places fault information (if any) in the fault tables.

Option Module Dual Port Interface Tests

After completion of the option module self-test and results reporting, integrity tests are jointly performed on the dual port interface used by the CPU and option module for communications. These tests validate that the two modules are able to pass information back and forth, as well as verify the interrupt and semaphore capabilities needed by the communications protocol. After dual port interface tests are complete, the communications messaging system is initialized.

I/O System Initialization

If the module is a Model 70 input module, no further configuration is required. If the module is a Model 70 output module, the module is commanded to go to its default state. A Model 70 output module defaults to all inputs off at power-up and in failure mode, unless told otherwise. When the module is a Bus Transmitter Module (BTM), it is interrogated about what remote racks are present in the system. Based upon the BTM's response, the CPU adds those racks and their associated slots into the list of slots to be configured.

Finally, the I/O Scanner performs its initialization. The I/O Scanner initializes all the I/O controllers in the system by establishing the I/O connections to each I/O bus on the I/O controller and obtaining all I/O configuration data from that I/O controller. This configuration data is compared with the user specified I/O configuration and any differences reported in the I/O fault table. The I/O Scanner then sends each I/O controller a list of the I/O modules to be configured on the I/O bus. After the I/O controllers have been initialized, the I/O Scanner replaces the factory default settings in all I/O modules with any application specified settings.

For Model 70 input modules, the board may be set to interrupt when the signal(s) change state, and whether the interrupt will occur when the signal(s) transitions from high to low or low to high. For Model 70 output modules, their default state may be changed from Off to Hold Last State.

Power-Down Sequence

System power-down occurs when the power supply detects that incoming AC power has dropped for more than one power cycle. A signal line on the backplane is driven low to indicate the condition, which causes an interrupt to the CPU. From the time this signal occurs, a minimum of 5 milliseconds remain to complete power-down processing.

Retention of Data Memory Across Power Failure

Because application RAM and BCP memory are battery-backed, the following types of data are preserved across a power cycle:

- Application program
- Fault tables and other diagnostic data
- Checksums on programs and blocks
- Override data
- Data in register (%R), local register (%L), and program register (%P) memory
- Data in analog memory (%AI and %AQ)
- State of discrete inputs (%I)
- State of retentive discrete outputs (%Q)
- State of retentive discrete internals (%M)
- State of discrete system internals (system bits, fault bits, reserved bits)

The following types of data are not preserved across a power cycle:

- State of discrete temporary memory (%T)
- %M and %Q memories used on non-retentive -()- coils

Section 7: Clocks and Timers

Clocks and timers provided by the Series 90-70 PLC include an elapsed time clock, a time-of-day clock, and software and hardware watchdog timers. Three types of timer function blocks include an on-delay timer, an off-delay timer, and a start-reset timer. Timed contacts cycle on and off (in square wave form) every 0.01 second, 0.1 second, 1.0 second, and 1 minute.

Elapsed Time Clock

The elapsed time clock uses 100 microsecond "ticks" to track the time elapsed since the CPU powered on. The clock is not retentive across a power failure; it restarts on each power-up. Once per second the hardware interrupts the CPU to enable a seconds count to be updated. This seconds count rolls over (seconds count returns to zero) approximately 100 years after the clock begins timing.

Because the elapsed time clock provides the base for system software operations and timer function blocks, it may not be reset from the user program or the programmer. However, the application program can read the current value of the elapsed time clock by using Service Request function #16.

Time-of-Day Clock

The time of day in the Series 90-70 PLC is maintained by a hardware time-of-day clock. The time-of-day clock maintains the following seven time functions:

- Year (two digits)
- Month
- Day of month
- Hour
- Minute
- Second
- Day of week

The time-of-day clock is battery-backed and maintains its present state across a power failure. However, unless the user initializes the clock, the values it contains are meaningless. The application program can read and set the time-of-day clock using Service Request function #7. The time-of-day clock can also be read and set from your programming software.

The time-of-day clock is designed to handle month-to-month and year-to-year transitions. It automatically compensates for leap years into the second half of the twenty-first century.

Watchdog Timer

Software Watchdog Timer

A software watchdog timer in the Series 90-70 PLC is designed to detect “failure to complete sweep” conditions. The timer value for the software watchdog timer is set by using your programming software. The allowable range for this timer is 10 to 2550 milliseconds; the default value is 200 milliseconds. The software watchdog timer always starts from zero at the beginning of each sweep.

The software watchdog timer is useful in detecting abnormal operation of the application program which prevents the PLC sweep from completing within the user specified time. Examples of such abnormal application program conditions are as follows:

- Excessive recursive calling of a block
- Excessive looping (large loop count or large amounts of execution time for each iteration)
- Infinite execution loop

When selecting a software watchdog value, always set the value higher than the longest expected sweep time to prevent accidental expiration. For Constant Sweep and Microcycle Sweep modes, allowance for oversweep conditions should be considered when selecting the software watchdog timer value.

If the software watchdog timeout value is exceeded, the OK LED blinks, and the CPU goes to Stop/Halt mode. Certain functions, however, are still possible. A fault is placed in the PLC fault table, and outputs go to their default state. If you are using serial communications (not Ethernet), the CPU will only communicate with the programmer; no other communications or operations are possible. To recover, power must be cycled on the rack containing the CPU.

To extend the current sweep beyond the software watchdog timer value, the application program may restart the software watchdog timer using Service Request function #8. However, the software watchdog timer value may only be changed from the configuration software.

Hardware Watchdog Timer

A backup circuit provides additional protection for the PLC. If this backup circuit activates, the PLC is immediately placed in Reset mode. Outputs go to their default state; no communications of any form are possible, and the CPU will halt. To recover, power must be cycled.

Section 8: System Security

The Series 90-70 PLC supports the following three types of system security:

1. Passwords/privilege levels
2. OEM protection
3. Write protect keyswitch

Passwords and Privilege Levels

Passwords are a configurable feature of the Series 90-70 PLC. Their use is optional and may be set up using your programming software. The purpose of passwords is to provide different levels of access privilege for the PLC when the programmer is in Online or Monitor mode. Passwords are not used if the programmer is in Offline mode. The use of passwords may restrict the following:

- Changing I/O and PLC configuration data
- Changing programs
- Reading PLC data
- Reading programs
- Locking blocks

The default state is no password protection. There is one password for each privilege level in the PLC. Each password may be unique; however, the same password can be used for more than one level. Passwords are one to seven ASCII characters in length. Only the programmer may change passwords.

PLC password protection can be used to restrict access to selected PLC functions. After passwords have been set up, access to the PLC via any communications path is restricted unless the proper password has been entered. Once a password has successfully been accepted, access to the privilege level requested and below will be granted (for example, provide password for level 3 will allow access to functions at levels 0, 1, 2, and 3). If the PLC communications are suspended, protection level will automatically return to the lowest privilege level of the highest unprotected level or privilege level 2.

Table 2-18. Privilege Levels

Priv Level	Password	Access Description
4	Yes	Write to all configuration or logic. Configuration may only be written in Stop mode; logic may be written in Stop or Run mode. Set or delete passwords for any level.
3	Yes	Write to all configuration or logic when the CPU is in Stop mode, including word-for-word changes, the addition/deletion of program logic, and the overriding of discrete I/O.
2	Yes	Write to any data memory. This includes the toggle/force of reference values but does **not** include overriding discrete I/O. The PLC can be started or stopped. PLC and I/O fault tables can be cleared. **NOTE:** This is the default if no passwords are defined.
1	Yes	Read any PLC data, except for passwords. This includes reading fault tables, performing datagrams, verifying logic/config, and load program and configuration, etc. from the PLC. None of this data may be changed.
0	No	Read the current status of the PLC (including features supported by the PLC), read the name of the Resource (CP name prior to release 6.0), change privilege level, and log in as programmer.

Note

The CPU mode switch on the CPU overrides the password protection. Even though the programmer may not be able to switch between RUN and STOP mode, the physical switch can.

Protection Level Request from Programmer

Upon connection to the CPU, the Programmer requests the protection status of each privilege level from the CPU. The Programmer then requests the CPU to move to the highest non-protected level, thereby giving the programmer access to the highest non-protected level without having to specifically request any particular level.

A programmer requests a privilege level change by supplying the new privilege level and the password for that level. If the password sent by the programmer does not agree with the password stored in the PLCs password access table for the requested level, the privilege level change is denied and a fault is logged in the PLC fault table. The current privilege level is maintained, and no change will occur. A request to change to a privilege level that is not password protected is made by supplying the new level and a null (Hex 0) password. A privilege change may be to a lower level as well as to a higher level.

Disabling Passwords

The use of password protection is optional. If the user desires to prevent the use of password protection, passwords can be disabled using the programming software.

Note

To re-enable passwords after passwords have been disabled, the PLC must be power cycled with the battery removed.

Password protection also prevents firmware upgrades to the FLASH memories used on the CPM 914, 915, 924, 925, and the CPX772, 782, 928, and 935. Prior to attempting a firmware upgrade in any of these modules, disable password protection, then re-enable it after the upgrade.

OEM Protection

OEM protection is similar to the passwords and privilege levels; however, OEM protection provides a higher level of security. The OEM protection feature is enabled/disabled using a 1 to 7 character password. When OEM protection is enabled, all read and write access to the PLC program and configuration is prohibited.

Protection for OEMs' investment in software is provided in the form of a special password known as the *OEM key*. When the OEM key has been given a non-NULL value, the CPU may be placed in a mode in which reads and writes of the logic as well as writes to the configuration are prohibited. This allows a Third party OEM to create Control Programs for the PLC CPU and then set the OEM-locked mode that prevents the end user from reading or modifying the program.

Note

OEM protection also prevents firmware upgrades to the FLASH memories used on the CPM 914, 915, 924, 925, and CPX772, 782, 928, and 935. Prior to attempting a firmware upgrade in any of these modules, disable OEM protection, then enable it again after the upgrade.

Write Protect Keyswitch

The 90-70 CPU models CPU 780, 781, 782, 788, 789, 914, 924, 915, 925; CGR 772 and 935; and CPX 772, 782, 928 and 935 all contain a memory write protect keyswitch. This keyswitch is located on the top of the faceplate, above the upper faceplate-to-rack clip. When in the protected position, the PLC program and configuration cannot be modified or deleted.

Note

The write protect keyswitch, when in the "write protected" position, also prevents firmware upgrades to the Flash memories used on the CPM 914, 915, 924, 925, and CPX 772, 782, 928 and 935. Prior to attempting a firmware upgrade in any of these modules, place the write protect keyswitch into the "write enabled" position.

Section 9: Series 90-70 PLC I/O System

The Series 90-70 PLC I/O system provides the interface between the Series 90-70 PLC and user supplied devices and equipment. The I/O system supports the rack-type Model 70 I/O, the Genius I/O system, and the FIP I/O system. A Genius I/O Bus Controller (GBC) module provides the interface between the Series 90-70 PLC CPU and a Genius I/O bus. A FIP I/O Bus Controller (FBC) module provides the interface between the Series 90-70 PLC CPU and a FIP I/O bus. In addition to supporting these three I/O subsystems, the I/O system will also support Ethernet Interfaces and PCMs.

The I/O structure for the Series 90-70 PLC is shown in the following figure:

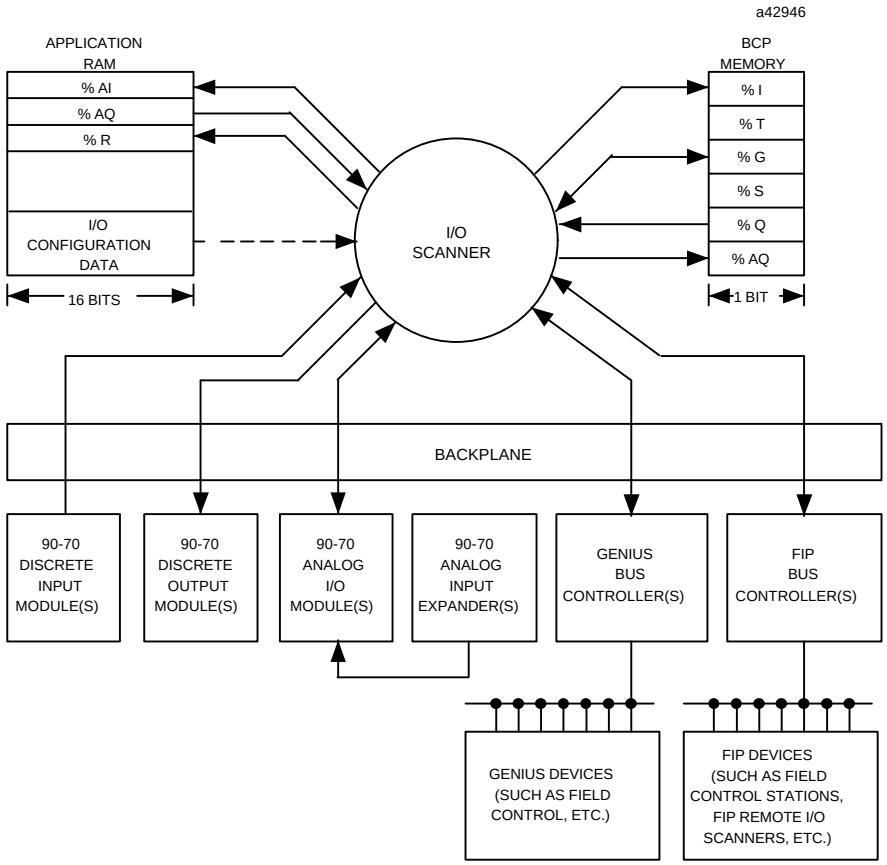


Figure 2-13. Series 90-70 PLC I/O Structure

I/O Data Mapping

Discrete inputs and outputs are stored as bits in the CPU BCP Bit Cache memory. Analog I/O is stored in the application RAM allocated for that purpose. Analog data is always stored in the demultiplexed state, with each channel requiring one word (16 bits).

Default Conditions

The programming software provides the ability to specify that the first input may be an interrupt input and for the filter speed to be fast or slow; but upon power-up, Model 70 discrete input modules always default to the first input on the module not interrupting and the input filter being slow speed. If changed by the user, new defaults are applied when the board is configured by the CPU during the power-up process or whenever else the module may go through configuration.

Model 70 discrete output modules default to all outputs off. The configuration utility provides the ability to specify whether the CPU transitions from Run/Enabled to Run/Disabled or Stop mode. It also applies this default information when the system halts.

Genius I/O

Information relative to using Genius I/O in a Series 90-70 PLC system is presented in the following paragraphs. For specific information on Genius I/O block types, configuration, and setup, refer to the *Genius I/O System User's Manuals*, GEK-90486-1 and -2.

Genius I/O Bus Configuration

The Bus Controller used in the Series 90-70 PLC controls a single Genius I/O bus. Any type of Genius I/O block may be attached to the bus.

In the I/O fault table, the rack, slot, bus, module, and I/O point number are given for a fault. Bus number one refers to the bus on the single-channel Genius Bus Controller.

Genius I/O Data Mapping

Genius I/O discrete inputs and outputs are stored as bits in the CPU Bit Cache memory. Genius I/O analog data is stored in the application RAM allocated for that purpose (%AI and %AQ). Analog data is always stored one channel per one word (16 bit).

An analog grouped module consumes (in the input and output data memories) only the amount of data space required for the actual inputs and outputs. For example, the Genius I/O 115 VAC Grouped Analog Block, IC660CBA100, has four inputs and two outputs; it consumes four words of Analog Input memory (%AI) and two words of Analog Output memory.

A discrete grouped module, each point of which is configurable with the Hand-Held Monitor (HHM) to be input, output, or output with feedback, consumes an amount in both discrete input memory (%I) and discrete output memory (%Q) equal to its physical size. Therefore, the 8 I/O 115 VAC Discrete Grouped Block (IC660CBD100) requires 8 bits in the %I memory and 8 bits in the %Q memory, regardless of how the block is configured.

The following four Genius I/O blocks are assigned to the analog memories:

- 6-Channel Analog Grouped Block
- 6-Channel Thermocouple Block
- 6-Channel RTD Block
- 4-Channel Strain Gauge/mV Analog Input Block

The Thermocouple and RTD blocks are also referred to as Low-Level Analog Input blocks.

Analog Grouped Block

The Analog Grouped block contains four analog input channels and two analog output channels. When a block gets its turn on the Genius I/O Bus, it broadcasts the data for all four input channels in one broadcast control message. Then, when the Bus Controller gets its turn, it sends the data for both output channels to the block in a directed control message.

Low-Level Analog Blocks

Unlike the Analog Grouped block, the low-level analog blocks are input-only blocks. All have six channels.

Default Conditions

Genius I/O blocks have a number of default conditions that may be set using the Genius I/O Hand-Held Monitor. These defaults include the following:

- Report faults
- Range select
- Analog input and output scaling
- Input filter time
- Alarm input mode
- Output hold last state
- Output default

These defaults are stored in EEPROM in the block itself. The Series 90-70 PLC configuration utility supports the changing of only a small subset of these defaults. For more information, refer to the *Genius I/O System User's Manuals*, GEK-90486-1 and -2.

Through the COMMREQ function block, the application program can request the Bus Controller to change any default condition on a specific block. However, this change will only be accepted by the block if it is not in Config Protect mode. If Config Protect mode is set, only the Hand-Held Monitor can be used to change the defaults. The format of the COMMREQ function block for Genius I/O is described in the *Genius Bus Controller User's Manual*, GFK-0398.

Genius Global Data Communications

The Series 90-70 PLC supports the sharing of data among multiple PLC systems that share a common Genius I/O bus. This mechanism provides a means for the automatic and repeated transfer of %G, %I, %Q, %AI, %AQ, and %R data. No special application programming is required to use global data since it is integrated into the I/O scan. All GE Fanuc PLCs that have Genius I/O capability can send global data to a Series 90-70 PLC and can receive data from a Series 90-70 PLC. Your programming software is used to configure the receiving and transmitting of global data on a Genius I/O bus.

Note

Genius global data communications do not continue to operate when the 90-70 PLC is in STOP/NOIO mode. However, if the 90-70 PLC is in STOP/IOSCAN mode, then Genius global data communications will continue to operate.

FIP I/O

Information relative to using FIP I/O in a Series 90-70 PLC system is presented in the following paragraphs. For specific information on FIP I/O types, configuration, and setup, refer to the *Series 90-70 FIP Bus Controller User's Manual*, GFK-1038.

FIP I/O Bus Configuration

The FIP Bus Controller used in the Series 90-70 PLC controls a single FIP I/O bus. Currently supported are the 90-30 FIP Remote I/O Scanner, FIP Bus Interface Unit (for Field Control), and generic FIP I/O module configurations. All of the FIP I/O interface modules (for example, the FIP Remote I/O Scanner) must provide input data to the 90-70 FIP Bus Controller so that the FIP Bus Controller has this same input data to provide to the 90-70 CPU during the next normal input scan. Similarly, when the 90-70 CPU performs the next output scan, the FIP Bus Controller accepts this output data and passes it on to the appropriate FIP I/O interface module to then update the local I/O.

90-30 FIP Remote I/O Scanner

The FIP Remote I/O Scanner provides the ability to use 90-30 I/O as a remote I/O node on a FIP I/O network. The FIP Remote I/O Scanner module provides the communications interface to the FIP I/O network (communications with the 90-70 FIP Bus Controller) and also provides the I/O scanning function for the local 90-30 I/O modules. (For more information on the 90-30 FIP Remote I/O Scanner, please refer to the *90-30 FIP Remote I/O Scanner User Manual*, GFK-1037)

FIP Bus Interface Unit (Field Control)

The FIP Bus Interface Unit provides the ability to use Field Control I/O as a remote I/O node on a FIP I/O network. The FIP Bus Interface Unit module provides the communication interface to the FIP I/O network (communications with the 90-70 FIP Bus Controller) and also provides the I/O

scanning function for the local Field Control modules. (For more information on FIP Field Control, please refer to the *FIP Bus Interface Unit User's Manual*, GFK-1175).

Generic FIP I/O

Generic FIP I/O allows for configuring FIP I/O other than the FIP Remote I/O Scanner and FIP Bus Interface Unit. This permits the 90-70 CPU and the 90-70 FIP Bus Controller to assign I/O reference addresses to the generic FIP I/O device. The configuration selection also permits the 90-70 FIP Bus Controller to recognize the generic FIP I/O module on the FIP I/O network.

FIP I/O Fault Data

In the I/O fault table, the rack, slot, FIP drop ID, remote rack, and remote slot number are given for faults occurring in an FIP Remote I/O Scanner or FIP Bus Interface Unit controlled remote I/O node. No fault information can be obtained from generic FIP I/O.

FIP I/O Data Mapping

FIP I/O discrete inputs and outputs are stored as bits in the CPU Bit Cache memory. FIP I/O analog data is stored in the application RAM allocated for that purpose (%AI and %AQ). Analog data is always stored one channel per one word (16 bit).

An analog grouped module consumes (in the input and output data memories) only the amount of data space required for the actual inputs and outputs. For example, an analog module with four inputs and two outputs consumes four words of Analog Input memory (%AI) and two words of Analog Output memory.

Default Conditions

FIP I/O devices have a number of default conditions that may be set using your programming software. The default conditions include the following:

- Range select
- Analog input scaling
- Analog output scaling
- Discrete output default Off (fixed)
- Analog output default Hold Last State (fixed)

Diagnostic Data Collection

Diagnostic data in a Series 90-70 PLC I/O system is obtained in one of the following two ways:

1. If an I/O module has an associated Bus Controller (Genius Bus Controller or FIP Bus Controller), then the Bus Controller provides the module's diagnostic data for the CPU.
2. If an I/O module is a Model 70 I/O module, then the CPU's I/O Scanner subsystem generates the diagnostic bits based on the data provided by the I/O module.

The diagnostic bits are derived from the diagnostic data sent from the I/O modules to their I/O controllers (Genius Bus Controller, FIP Bus Controller, or 90-70 CPU). Diagnostic bits always indicate the current fault status of the associated module. Bits are set when faults occur and are cleared when faults are cleared.

In general, diagnostic data is not maintained by the Series 90-70 PLC for foreign I/O (not GE Fanuc) modules. Any diagnostic information provided by those boards must be specifically accessed by the application program using the VME Read and VME Write function blocks.

Beginning with 90-70 CPU release 5.50, the 90-70 system has supported foreign I/O modules when developed under license agreement with GE Fanuc. These boards are then configured as "Foreign VME" and the interface mode is "I/O Scan." Boards developed to conform to the I/O Scan interface can provide discrete and analog diagnostic information to the 90-70 CPU.

Discrete I/O Diagnostic Information

Diagnostic information is maintained by the Series 90-70 PLC for each discrete I/O point. Two memory blocks are allocated in application RAM for discrete diagnostic data. One is associated with %I memory and the other with %Q memory. One bit of diagnostic memory is associated with each I/O point. This bit indicates the validity of the associated I/O data. Each discrete point has a fault reference available that may be interrogated using two special contacts: a fault contact (-[FAULT]-) and a no-fault contact (-[NOFLT]-). The PLC only collects this fault data if enabled to do so through your programming software. The following table shows the state of the fault and no-fault contacts.

Condition	[FAULT]	[NOFLT]
Fault Present	ON	OFF
Fault Absent	OFF	ON

Analog I/O Diagnostic Data

Diagnostic information is made available by the PLC CPU for each analog channel associated with Series 90-70 analog input modules, Series 90-70 analog output modules, Genius analog blocks, etc. Two memory blocks are allocated for analog diagnostic data. One is associated with %AI analog input memory and the other with %AQ analog output memory. One byte of diagnostic memory is allocated for each analog I/O channel. Since each analog I/O channel uses two bytes of %AI and %AQ memory, the diagnostic memory is half the size of the data memory.

The analog diagnostic data contains both diagnostics and process data with the process data being the High Alarm and Low Alarm bits. The diagnostic data is referenced with the -[FAULT]- and -[NOFLT]- contacts. The process bits are referenced with the -[HIALR]- and -[LOALR]- contacts. The memory allocation for analog diagnostic data is one byte per word of analog input and analog output allocated by the user. When an analog fault contact is referenced in the application program, the PLC does an Inclusive OR on all the bits in the diagnostic byte except the process bits. The alarm contact is closed if any diagnostic bit is ON and OFF, only if all bits are OFF.

Chapter 3 Fault Explanation and Correction

This chapter is an aid to troubleshooting a Series 90-70 PLC system. It explains the fault descriptions, which appear in the PLC fault table, and the fault categories, which appear in the I/O fault table.

Each fault explanation in this chapter lists the fault description for the PLC fault table or the fault category for the I/O fault table. Find the fault description or fault category corresponding to the entry on the applicable fault table displayed on your programmer screen. Beneath it is a description of the cause of the fault along with instructions to correct the fault.

Chapter 3 contains the following sections:

Section 1: System Faults

Certain discrete system references are assigned to correspond to specific fault types. These system references will go to a Logic 1 state when their corresponding fault occurs. They will remain at a Logic 0 level when no corresponding fault is detected. These fault references are summarized in the table below.

Although some of these seem to be duplicates, such as SY_FLT and SY_PRES for example, there is a difference in how these bits can be cleared. In this example, SY_FLT can be cleared either by cycling PLC power or by clearing the PLC fault table. SY_PRES, however, can only be cleared by clearing the PLC fault table.

The PLC and I/O fault tables are maintained in battery-backed RAM memory. As long as the battery is maintaining memory contents, cycling power will not clear these two fault tables.

Nickname	Address	Description
ANY_FLT	%SC0009	Set when any fault occurs that causes an entry in the PLC or I/O fault tables. Cleared when both fault tables are cleared or when the PLC is power cycled.
SY_FLT	%SC0010	Set when any fault occurs that causes an entry in the PLC fault table. Cleared when the PLC fault table is cleared or when the PLC is power cycled.
IO_FLT	%SC0011	Set when any fault occurs that causes an entry in the I/O fault table. Cleared when the I/O fault table is cleared or when the PLC is power cycled.
SY_PRES	%SC0012	Indicates the presence of at least one entry in the PLC fault table. Cleared when the PLC fault table is cleared.
IO_PRES	%SC0013	Indicates the presence of at least one entry in the I/O fault table. Cleared when the I/O fault table is cleared.
HRD_FLT	%SC0014	Set when a hardware fault occurs. Cleared when both fault tables are cleared or when PLC is power cycled.
SFT_FLT	%SC0015	Set when a software fault occurs. Cleared when both fault tables are cleared or when PLC is power cycled.

These bits are set or reset during the input scan of the sweep following the one in which the change occurs.

System Fault References

When a system fault reference is set, additional fault references are also set. The following table lists these other types of faults. References marked with asterisks below are configurable system fault references.

Table 3-1. System Fault References

Fault Type	PLC System Fault (SY_FLT)	I/O Fault (IO_FLT)
Hardware Fault (HRD_FLT)	SBUS_ER System bus error HRD_CPU PLC CPU hardware fault. HRD_SIO Module hardware fault SBUS_FL System bus failure *	
Software Fault (SFT_FLT)	SFT_SIO Option module software fault SFT_CPU PLC software fault * MAX_IOC Too many Bus Controllers * STOR_ER Programmer download failed *	SFT_IOC Bus Controller software fault
Other Faults	PB_SUM block checksum fault LOW_BAT Low battery signal OV_SWP Over constant sweep time SY_FULL PLC fault table full IO_FULL I/O fault table full APL_FLT Application program fault NO_PROG No application program at power-up * BAD_RAM Corrupted program memory * WIND_ER Incomplete window service * BAD_PWD Password access failure * NUL_CFG No configuration present * LOS_SIO Loss of option module. * ADD_RCK Addition of expansion rack * ADD_SIO Addition of option module * CFG_MM Configuration mismatch * LOS_RCK Loss of rack *	LOS_IOC Loss of Bus Controller LOS_IOM Loss of I/O module ADD_IOC Addition of bus controller ADD_IOM Addition of I/O module IOC_FLT Bus or Bus Controller fault IOM_FLT I/O module fault

* Configurable system fault references.

Configurable Fault Actions

For some faults, the PLC must stop execution. For other faults, the appropriate response to a fault may depend on the nature of the application. All faults are initially assigned to one of these three actions:

Fault Action	Description
Fatal	These faults halt the system, set diagnostic variables, and are logged in a fault table.
Diagnostic	These faults do not halt the system. They do, however, set diagnostic variables and are logged in a fault table.
Informational	These faults are logged in a fault table, but cause no other action.

For some faults, called "non-configurable" faults, the fault action cannot be changed. Other faults, called "configurable" faults, can have their fault type changed to another fault action if such a change is suitable for the application.

The following table lists configurable faults. If the fault also causes additional faults, only the initiating fault is logged.

Table 3-2. Configurable Fault References

Fault (Default Action)	Description	May Also Be Set
SBUS_ER (diagnostic)	System bus error. (The BSERR* signal was generated on the VME system bus.)	HRD_FLT SY_PRES, SY_FLT Other references may also be set depending on the type of access when the BSERR* occurred.
HRD_CPU (fatal)	PLC CPU hardware fault, such as failed memory device or failed serial port).	SY_FLT, SY_PRES HRD_FLT
HRD_SIO (diagnostic)	Non-fatal hardware fault on any module in the system, such as failure of a serial port on a PCM.	SY_FLT, SY_PRES HRD_FLT
SFT_IOC (diagnostic)	Non-recoverable software error in a Genius Bus Controller.	IO_FLT, IO_PRES SFT_FLT
SFT_SIO (diagnostic)	Non-recoverable software error in a PCM or LAN interface module.	SY_FLT, SY_PRES SFT_FLT
PB_SUM (fatal)	Program or block checksum failure during power-up or in Run mode.	SY_FLT, SY_PRES
LOW_BAT (diagnostic)	Low battery signal from CPU or another module in system.	SY_FLT, SY_PRES
OV_SWP (diagnostic)	Constant sweep time exceeded.	SY_FLT, SY_PRES
SY_FULL IO_FULL (diagnostic)	PLC fault table full (16 entries). I/O fault table full (32 entries).	SY_FLT, SY_PRES IO_FLT, IO_PRES

Table 3-2. - Continued Configurable Fault References

Fault (Default Action)	Description	May Also Be Set
APL_FLT (diagnostic)	Application fault.	SY_FLT, SY_PRES
LOS_RCK (diagnostic)	Loss of rack (BRM failure, loss of power), or missing a configured rack.	SY_FLT, SY_PRES IO_FLT, IO_PRES
LOS_IOC* (diagnostic)	Loss of Bus Controller channel, or missing a configured Bus Controller.	IO_FLT, IO_PRES
LOS_IOM (diagnostic)	Loss of I/O module (does not respond), or missing a configured I/O module.	IO_FLT, IO_PRES
LOS_SIO (diagnostic)	Loss of option module (does not respond), or missing a configured module.	SY_FLT, SY_PRES
ADD_RCK (diagnostic)	New rack added, or previously faulted rack has returned.	SY_FLT, SY_PRES
ADD_IOC (diagnostic)	Previously faulted Bus Controller is no longer faulted.	IO_FLT, IO_PRES
ADD_IOM (diagnostic)	Previously faulted I/O module is no longer faulted.	IO_FLT, IO_PRES
ADD_SIO (diagnostic)	New option module is added, or previously faulted module no longer faulted.	SY_FLT, SY_PRES
IOC_FLT (diagnostic)	Non-fatal bus or Bus Controller error, more than 10 bus errors in 10 seconds (error rate is configurable).	IO_FLT, IO_PRES
IOM_FLT (diagnostic)	Point or channel on an I/O module; a partial failure of the module.	IO_FLT, IO_PRES
CFG_MM (fatal)	Wrong module type detected during power-up or Run mode. The PLC does not check the configuration parameters set up for individual modules such as Genius I/O blocks.	SY_FLT, SY_PRES

*The IOC software fault will have the same action as what you set for LOS_IOC.

Note

If the fault action for a fault logged to the fault table is informational, the configured action is not used. For example, if the logged fault action for an SBUS_ERR is informational, but you configure it as fatal, the action is still informational.

Non-Configurable Faults

For non-configurable faults, the fault action cannot be changed. These faults must always be either fatal or informational faults.

The following table lists non-configurable faults. If the fault also causes additional faults, only the initiating fault is logged.

Table 3-3. Non-Configurable Faults

Fault	Description	Result
SBUS_FL (fatal)	System bus failure. The PLC CPU was not able to access the VME bus. BUSGRT*NMI error.	Sets SY_FLT, HRD_FLT, and SY_PRES.
NO_PROG (information)	No application program is present at power-up. Should only occur the first time the PLC is powered up or if the battery-backed RAM containing the program fails.	Does not set any references. PLC will not go to Run mode; it continues executing Stop mode sweep until a valid program is loaded. This can be a "null" program that does nothing. Sets SY_FLT and SY_PRES.
BAD_RAM (fatal)	Corrupted program memory at power-up. Program could not be read and/or did not pass checksum tests.	Sets SY_FLT and SY_PRES.
WIND_ER (information)	Window completion error. Servicing of Programmer or Logic Window was skipped. Occurs in Constant Sweep or Microcycle Sweep Mode.	Sets SY_FLT and SY_PRES.
BAD_PWD (information)	Change of privilege level request to a protection level was denied; bad password.	Sets SY_FLT and SY_PRES.
NUL_CFG (fatal)	No configuration present upon transition to Run mode. Running without a configuration is equivalent to suspending the I/O scans.	Sets SY_FLT and SY_PRES.
SFT_CPU (fatal)	CPU software fault. A non-recoverable error has been detected in the CPU. May be caused by Watchdog Timer expiring.	PLC immediately transitions to Error Sweep mode. The only activity permitted is communication with the programmer. To be cleared, PLC power must be cycled. Sets SY_FLT, SY_PRES, and SFT_FLT.
MAX_IOC (fatal)	The maximum number of bus controllers has been exceeded. The Series 90 PLC supports 32 bus controllers.	Sets SY_FLT, SY_PRES, and SFT_FLT.
STOR_ER (fatal)	Download of data to PLC from the programmer failed; some data in PLC may be corrupted.	PLC will not transition to Run mode. This fault is not cleared at power-up, intervention is required to correct it. Sets SY_FLT and SY_PRES.

Fault Contacts

Fault (-[FAULT]-) and no-fault (-[NOFLT]-) contacts can be used to detect the presence of various faults in the system. These contacts can not be overridden. The following table shows the state of fault and no-fault contacts.

Condition	[FAULT]	[NOFLT]
Fault Present	ON	OFF
Fault Absent	OFF	ON

Fault Locating References (Rack, Slot, Bus, Module)

The Series 90-70 PLC supports reserved fault names for each rack, slot, bus, and module if the hardware is configured. By programming these names on the FAULT and NOFLT contact instructions, logic can be executed to locate faults associated with configured racks and modules.

Format of Fault References

Table 3-4. Fault Reference Names

Fault Reference Type	Reserved Name	Comment
Rack	RACK_0r	Where r is rack number 0 to 7.
Slot	SLOT_rs	Where r is rack number 0 to 7 and s is slot number 0 to 9.
Bus	BUS_rsb (Genius only)	Where r is rack number 0 to 7, s is slot number 0 to 9, and b is the Genius bus number 1 or 2.
Module	M_rsbmm (Genius only)	Where r is rack number 0 to 7, s is slot number 0 to 9, b is the Genius bus number 1 or 2, and mm is the Serial Bus Address (SBA) number 00 to 31.
FIP Module	F_rsmmm (FIP only)	Where r is rack number 0 to 7, s is slot number 0 to 9, and mmm is the FIP station address 000 to 255.

These fault names can only be programmed on the FAULT and NOFLT contacts. The reserved fault names are always available. It is not necessary to enable a special option, such as point faults.

These fault names do not correspond to %SA, %SB, %SC, or to any other reference type. Only the name is displayed. A reference table screen is not provided for the fault references.

The format of a rack fault name is RACK_0r, where r is the rack number 0 to 7. For example, RACK_01 shown in the example below represents rack 1.

The format of a slot fault name is SLOT_rs, where r is the rack number 0 to 7 and s is the slot number 0 to 9. For example, SLOT_15 shown in the example below represents rack 1, slot 5.

```
|
| RACK_01 SLOT_15                                                    %Q00002
|-[FAULT]-[NOFLT]-------------------------------------------------------( )-
|
```

The format of a bus fault name is a BUS_rsb, where r is the rack number 0 to 7, s is the slot number 0 to 9, and b is the bus number 1 or 2. For example, BUS_241 represents rack 2, slot 4, bus 1.

The format of a module fault name is M_rsbmm, where r is the rack number 0 to 7, s is the slot number 0 to 9, b is the bus number 1 or 2, and mm is the module number 00 to 31. For example, M_26128 represents rack 2, slot 6, bus 1, module 28.

The format of a FIP module fault is F_rsmmm, where *r* is the rack number 0 to 7, *s* is the slot number 0 to 9, and *mmm* is the FIP station address.

Behavior of Fault References

At power-up, all fault-locating references are cleared in the PLC. When a fault is logged, the PLC transitions the state of the affected reference(s). The state of the fault reference remains in the fault state until one of the following actions occurs:

- Both the PLC and the I/O fault tables are cleared through your programming software either by clearing each table individually or clearing the entire PLC memory.
- The associated device (rack, I/O module, or Genius device) is added back into the system. Whenever an "Addition of. . . " fault is logged, the PLC initializes all fault references associated with the device to the NoFlt state. These references remain in the NoFlt state until another fault associated with the device is reported. (This could take several seconds for distributed I/O faults, especially if the bus controller has been reset.)
- For FIP faults, refer to the *Series 90-70 FIP Bus Controller User's Manual* (GFK-1038).

Note

These fault references are set for informational purposes only. They should not be used to qualify I/O data. The I/O point fault references (described on page 3-9) may be used to qualify I/O data. The PLC does not halt execution as a result of setting a fault locating reference to the Fault state.

The fault references have a cascading effect. If there is a problem in the module located at rack 5, slot 6, bus 1, module 29, the following faults references are set: RACK_05, SLOT_56, BUS_561, and M_56129. There will only be one entry in the fault table to describe the problem with the module. The fault table does not show entries pertaining to the rack, slot, and bus in this case. A fault in FIP device at rack 5, slot 6, module 29 results in the following fault references being set: RACK_05, SLOT_56, F_56029.

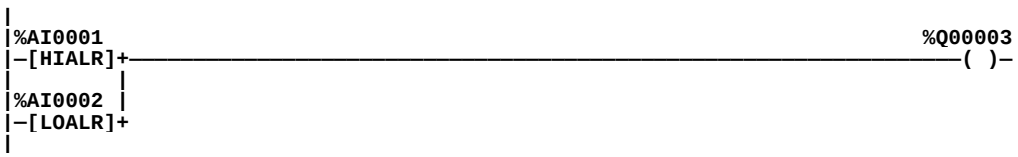
A module fault for a FIP Remote I/O Scanner device results in only one module fault type fault reference being set. For example, if FIP module with station ID 43 in 90-70 rack 3, slot 9 is an FIP Remote I/O Scanner, and within the FIP Remote I/O Scanner there is a faulted module in the FIP Remote I/O Scanner's rack 1, slot 5, then only fault reference F_39043 will be set.

Alarm Contacts

High (-[HIALR]-) and low (-[LOALR]-) alarm contacts are used to represent the state of the analog input module comparator function. However, the use of point faults must first be enabled in Hardware Configuration. For details about enabling Point Fault References, refer to the

Logicmaster User's Manual (GFK-0263) or the online help in Hardware Configuration within Control.

The following example logic uses both high and low alarm contacts.



Note

HIALR and LOALR contacts will not create an entry in a fault table.

Point Faults

Point faults pertain to external I/O faults, although they will also be set due to the failure of associated higher-level internal hardware (for example, IOC failure or loss of a rack). In order to use point faults, they must be enabled in Hardware Configuration. For details about enabling Point Fault References, refer to the Logicmaster User's Manual (GFK-0263) or the online help in Hardware Configuration within Control.

When enabled, a boolean reference for each discrete I/O point fault and a byte reference for each analog I/O channel level fault are allocated in PLC memory. The PLC memory used for point faults is included in the total reference table memory size. The FAULT and NOFLT contacts described above provide access to the point fault.

Section 2: Fault Handling

Note

This information on fault handling applies to systems programmed using Control.

Faults occur in the Series 90-70 PLC system when certain failures or conditions happen which affect the operation and performance of the system. These conditions, such as the loss of an I/O module or rack, may affect the ability of the PLC to control a machine or process. These conditions may also have beneficial effects, such as when a new module comes online and is now available for use. Or these conditions may only act as an alert, such as a low battery signal which indicates that the battery protecting the memory needs to be changed.

For information on system status/fault references, refer to Chapter 2, section 3, "Program Organization."

Alarm Processor

The condition or failure itself is called a fault. When a fault is received and processed by the CPU, it is called an alarm. The software in the CPU which handles these conditions is the Alarm Processor. The interface to the user for the Alarm Processor is through the programming software. Any detected fault is recorded in a fault table and displayed on either the PLC fault table screen or the I/O fault table screen, as applicable.

Classes of Faults

The Series 90-70 PLC detects several classes of faults. These include internal failures, external failures, and operational failures.

Table 3-5. Classes of Faults

Fault Class	Examples
Internal Failures	Non-responding modules. Low battery condition. Memory checksum errors.
External I/O Failures	Loss of rack or module. Addition of rack or module. Loss of Genius I/O block.
Operational Failures	Communication failures. Configuration failures. Password access failures.

System Reaction to Faults

Typically, hardware failures require that either the system be shut down or the failure be tolerated. I/O failures may be tolerated by the PLC system, but they may be intolerable by the application or the process being controlled. Operational failures are normally tolerated. Series 90-70 PLC faults have three attributes:

Table 3-6. Fault Attributes

Attribute	Description
Fault Table Affected	I/O fault table PLC fault table
Fault Action	Fatal Diagnostic Informational
Fault Response	Configurable Non-configurable

Fault Tables

The two fault tables—the PLC fault table and the I/O fault table—are provided in this chapter to make faults easier to find.

Fault Action

Fatal faults cause the fault to be recorded in the appropriate table, any diagnostic variables to be set, and the system to be stopped. Diagnostic faults are recorded in the appropriate table, and any diagnostic variables are set. Informational faults are only recorded in the appropriate table.

Table 3-7. Fault Actions

Fault Action	Response by CPU
Fatal	Log fault in fault table. Set fault references. Go to Stop mode.
Diagnostic	Log fault in fault table. Set fault references.
Informational	Log fault in fault table.

Your programming software provides the capability to change the fault action of certain faults. There are two possible classifications in the utility: fatal and non-fatal. These correspond to fatal and diagnostic fault action in the PLC. Only fatal faults cause the system to halt. Additionally, the informational fault action only logs faults in the fault table.

When a fault is detected by the CPU, it uses a default fault action for that fault. For those faults which may have their action changed by the programming software, the CPU uses the fault action specified by the software; this may be the default action or the action chosen by the user.

Fault Response

Fault response refers to the ability of a fault to have its fault action changed. Those faults that can have their fault action changed are called configurable faults. Those which cannot are called non-configurable faults. Non-configurable faults are either fatal or informational. Also, non-configurable faults do not cause application available references to be set and cannot have alarm blocks associated with the detection of the fault. Some non-configurable faults also have other effects associated with them. Generally, these effects control the changing of the CPU's execution mode (Stop, Run/Disabled, Run/Enabled). An example of such an effect is the disabling of I/O when a null system configuration is detected in the system.

PLC Fault Table

The PLC fault table displays PLC faults such as password violations, PLC/configuration mismatches, parity errors, and communications errors.

The programmer may be in any operating mode. However, if the programmer is in Offline mode, no faults are displayed. In Online or Monitor mode, PLC fault data is displayed. In Online mode, faults can be cleared (this may be password protected).

Field	Description
Top Fault Displayed	The index of the PLC fault currently at the top of the fault display is shown on the first line of this screen.
Total Faults	The total number of faults since the table was last cleared.
Table Last Cleared	The date and time faults were last cleared from the fault table. This information is maintained by the PLC.
Entries Overflowed	The number of entries lost because the fault table has overflowed since it was cleared. The PLC fault table can contain up to 40 faults (16 prior to Release 6—configurable in Control).
PLC Time/Date	The current date and time. This is also maintained by the PLC.

Note

The size of the PLC fault table is configurable (with a default of 16 and a maximum of 40—configurable in Control). Additional faults (over the configured limit) cause the table to overflow, and faults are lost. The system reference SY_FULL (%S0009) is set to indicate that the fault table is full.

I/O Fault Table

The I/O fault table displays I/O faults such as circuit faults, address conflicts, forced circuits, and I/O bus faults.

The programmer may be in any operating mode. However, if the programmer is in Offline mode, no faults are displayed. In Online or Monitor mode, PLC fault data is displayed. In Online mode, faults can be cleared (this feature may be password protected).

Field	Description
Top Fault Displayed	The index of the I/O fault currently at the top of the fault display is shown on the first line of this screen.
Total Faults	The total number of faults since the table was last cleared.
Fault Description	A more specific indication of the type of fault that is currently highlighted in the I/O fault table.
Table Last Cleared	The date and time faults were last cleared from the fault table. This information is maintained by the PLC.
Entries Overflowed	The number of entries lost because the fault table has overflowed since it was cleared. The I/O fault table can contain up to 32 faults.
PLC Time/Date	The current date and time. This is also maintained by the PLC.

Note

The size of the I/O fault table is configurable (with a default of 32 and a maximum of 40—configurable in Control). Additional faults (over the configured limit) cause the table to overflow, and faults are lost. The system reference IO_FULL (%S0010) is set to indicate that the fault table is full.

User-Defined Faults

User-defined faults can be logged in the PLC or I/O fault table. When a user-defined fault occurs, it is logged in the appropriate fault table as “Application Msg (error_code):” and may be followed by a descriptive message up to 24 characters. All characters in the descriptive message can be defined by the user. Although the message must end with the null character, e.g., zero (0), the null character does not count as one of the 24 characters. If the message contains more than 24 characters, only the first 24 characters are displayed.

Accessing Additional Fault Information

The fault tables displayed by the programming software contain basic information regarding the fault. For additional information a pertaining to each fault, double-click the fault as it appears in the programming software to access the Details window.

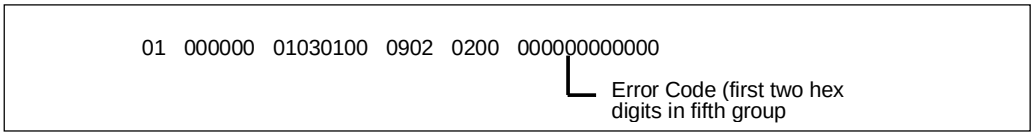
The last entry, ***Correction***, for each fault explanation in this chapter lists the action(s) to be taken to correct the fault. Note that the corrective action for some of the faults includes the statement:

Display the PLC fault table on the programmer. Contact GE Fanuc PLC Field Service, giving them all the information contained in the fault entry.

This second statement means that you must tell Field Service both the information readable directly from the fault table **and** the hexadecimal information. Field Service personnel will then give you further instructions for the appropriate action to be taken.

Section 3: PLC Fault Table Explanations

Each fault explanation contains a fault description and instructions to correct the fault. Many fault descriptions have multiple causes. In these cases, the error code and additional fault information are used to distinguish among different fault conditions sharing the same fault description. The error code is the first two hexadecimal digits in the fifth group of numbers, as shown in the following example.



Some faults can occur because random access memory on either the PLC CPU board or the expansion memory board has failed. These same faults may also occur because the system has been powered off and the battery voltage is (or was) too low to maintain memory. To avoid excessive duplication of instructions when corrupted memory may be a cause of the error, the correction simply states:

Perform the corrections for Corrupted Memory.

This means:

1. If the system has been powered off, replace the battery. Battery voltage may be insufficient to maintain memory contents.
2. Replace the expansion memory board. Integrated circuits on the memory board may be failing.
3. Replace the PLC CPU board. The integrated circuits on the PLC CPU board may be failing.

Note

For information about values for fault groups, refer to Appendix B.

Configurable Faults

Configurable faults can have their fault action (fatal or diagnostic) changed. The CPU uses the fault action specified by the software; this may be the default action or a fault action chosen by the user. In this section, the default fault action is listed for configurable faults.

Loss of or Missing Rack

The fault group Loss of or Missing Rack occurs when the system cannot communicate with an expansion rack because the BTM in the main rack failed, the BRM in the expansion rack failed, power failed in the expansion rack, or the expansion rack was configured in the configuration file but did not respond during power-up. The default fault action for this group is Diagnostic.

Error Code:	1
Name:	Rack Lost
Description:	The PLC generates this error when the main rack can no longer communicate with an expansion rack. The error is generated for each expansion rack that exists in the system.
Correction:	(1) Power off the system. Verify that both the BTM and the BRM are seated properly in their respective racks and that all cables are properly connected and seated. (2) Replace the cables. (3) Replace the BRM. (4) Replace the BTM.
Error Code:	2
Name:	Rack Not Responding
Description:	The PLC generates this error when the configuration file stored from the programmer indicates that a particular expansion rack should be in the system but none responds for that rack number.
Correction:	(1) Check rack number jumper behind power supply—first on missing rack and then on all other racks—for duplicated rack numbers. (2) Update the configuration file if a rack should not be present. (3) Add the rack to the hardware configuration if a rack should be present and one is not. (4) Power off the system. Verify that both the BTM and the BRM are seated properly in their respective racks and that all cables are properly connected and seated. (5) Replace the cables. (6) Replace the BRM. (7) Replace the BTM.

Loss of or Missing Option Module

The fault group Loss of or Missing Option Module occurs when a GEnet, PCM, BTM, or BRM fails to respond. The failure may occur at power-up if the module is missing or during operation if the module fails to respond. The default fault action for this group is Diagnostic.

Error Code:	3
Name:	Bus Transmitter Module Found in Expansion Rack
Description:	The PLC generates this error when a Bus Transmitter Module is found in an expansion rack.
Correction:	Power off the system and remove the BTM from the expansion rack.
Error Code:	16
Name:	Analog Expander Located to the Left of the Base Converter module.
Description:	An Analog Expander module has been placed in a rack to the left of its Base Converter module.
Correction:	Power off the system. Move the Analog Expander module to the right of the Base Converter module.
Error Code:	19
Name:	Lost Analog Expander module
Description:	Base Converter module has lost communications with the Analog Expander module.
Correction:	(1) Verify wiring linking Base Converter module with the Analog Expander module. (2) Replace the Analog Expander module. (3) If communication with all Analog Expanders is lost, replace the Base Converter module.
Error Code:	2C, 2D
Name:	Option Module Soft Reset Failed
Description:	PLC CPU unable to re-establish communications with option module after soft reset.
Correction:	(1) Try soft reset a second time. (2) Replace the option module. (3) Power off the system. Verify that both the BTM and the BRM are seated properly in their respective racks and that all cables are properly connected and seated. (4) Replace the cables. (5) Replace the BRM (Bus Receiver Module). (6) Replace the BTM (Bus Transmitter Module). (7) Report failure to GE Fanuc PLC Field Service.
Error Code:	3B
Name:	Loss of, or Missing Communications Driver
Description:	The PLC generates this error when VME communications fail between the PLC CPU and a third party VME module using the FULL MAIL configuration mode.
Correction:	(1) Update the configuration file with the correct communications parameters. (2) Replace the communications driver on the module. (3) Remove the module from the configuration file. (4) Replace the module.

Error Code:	3C
Name:	Module in Firmware Update Mode
Description:	The PLC generates this error when it finds a module in Firmware Update mode. Modules in this mode will not communicate with the PLC CPU.
Correction:	(1) Run the firmware update utility for the module. (2) Reset the module with the push-button. (3) Power-cycle the entire system. (4) Power-cycle the rack containing the module.
Error Code:	41
Name:	Unable to Establish VME Communications
Description:	The PLC generates this error when it finds a module in Standalone mode. A module in Standalone mode will appear to be operating correctly, but it will not communicate with the PLC CPU.
Correction:	(1) Reset the module with the push-button. (2) Power-cycle the entire system. (3) Power-cycle the rack containing the module.
Error Code:	FF
Name:	Option Module Communications Failed
Description:	PLC CPU generates this error when communication to the option module has failed.
Correction:	(1) Check the bus for abnormal activity. (2) Replace the intelligent option module to which the request was directed. (3) Check the parallel programmer cable for proper attachment.
Error Code:	All Others
Name:	Module Failure During Configuration
Description:	The PLC generates this error when a module fails during power-up or configuration store.
Correction:	(1) Power off the system. Replace the module located in that rack and slot. (2) If the board is located in an expansion rack, verify BTM/BRM cable connections are tight and the modules are seated properly; verify the addressing of the expansion rack. (3) Replace the BTM. (4) Replace the BRM. (5) Replace the rack.

Addition of or Extra Rack

The fault group Addition of Extra Rack occurs when a configured expansion rack with which the PLC CPU could not communicate comes online or is powered on, or an unconfigured rack is found. The default fault action for this group is Diagnostic.

Error Code:	1
Name:	Extra Rack
Correction:	(1) Check rack jumper behind power supply for correct setting. (2) Update the configuration file to include the expansion rack. (3) Remove the expansion rack from the hardware configuration. Note: No correction necessary if rack was just powered on.

Reset of, Addition of, or Extra Option Module

The fault group Reset of, Addition of, or Extra Option Module occurs when an option module (PCM, BTM, etc.) comes online, is reset, or a module is found in the rack but none is specified in the configuration. The default fault action for this group is Diagnostic.

Error Code:	1
Name:	Extra Option Module
Correction:	(1) Update the configuration file to include the module. (2) Remove the module from the system.
Error Code:	2
Name:	Module Restart Complete
Description:	Restart of module is complete.
Correction:	None
Error Code:	3
Name:	LAN Interface Restart Complete, Running Utility
Description:	The LAN Interface module has restarted and is running a utility program.
Correction:	Refer to the LAN Interface manual, GFK-0868 or GFK-0869 (previously GFK-0533).

System Configuration Mismatch

The fault group Configuration Mismatch occurs when the module occupying a slot is different from that specified in the configuration file. The default fault action is Fatal. When the I/O Scanner generates the mismatch because of a Genius block, the second byte in the Fault Extra Data field contains the bus address of the mismatched block.

Error Code:	2
Name:	Genius I/O Block Number Mismatch
Description:	The PLC generates this fault when the configured and physical Genius I/O blocks have different model numbers.
Correction:	(1) Replace the Genius I/O block with one corresponding to the configured module. (2) Update the configuration file.
Error Code:	4
Name:	I/O Type Mismatch
Description:	The PLC generates this fault when the physical and configured I/O types of Genius grouped blocks are different.
Correction:	(1) Remove the indicated Genius module and install the module indicated in the configuration file. (2) Update the Genius module descriptions in the configuration file to agree with what is physically installed.
Error Code:	7
Name:	Daughter Board Mismatch
Description:	The PLC generates this error when the configuration file indicates one size memory daughter (expansion) board should be on the PLC CPU and a different size is actually present.
Correction:	(1) Replace the module. (2) Replace the daughter board with the size indicated in the configuration file. (3) Update the configuration file to agree with the size of the daughter board actually installed on the PLC CPU.
Error Code:	8
Name:	Analog Expander Mismatch
Description:	The PLC generates this error when the configured and physical Analog Expander modules have different model numbers.
Correction:	(1) Replace the Analog Expander module with one corresponding to configured module. (2) Update the configuration file.
Error Code:	9
Name:	Genius I/O Block Size Mismatch
Description:	The PLC generates this error when block configuration size does not match the configured size.
Correction:	Reconfigure the block.

Error Code:	A
Name:	Unsupported Feature
Description:	Configured feature not supported by this revision of the module.
Correction:	(1) Update the module to a revision that supports the feature. (2) Change the module configuration.
Error Code:	B
Name:	Revision A of BTM not in Right-most Slot
Description:	The BTM (Revision A version) is not the right-most module in the rack.
Correction:	(1) Move the BTM to the right of all other modules in the rack. (2) Upgrade the BTM to a newer version (Revision B or higher).
Error Code:	E
Name:	LAN Duplicate MAC Address
Description:	This LAN Interface module has the same MAC address as another device on the LAN. The module is off the network.
Correction:	(1) Change the module's MAC address. (2) Change the other device's MAC address.
Error Code:	F
Name:	LAN Duplicate MAC Address Resolved
Description:	Previous duplicate MAC address has been resolved. The module is back on the network. This is an informational message.
Correction:	None required.
Error Code:	10
Name:	LAN MAC Address Mismatch
Description:	MAC address programmed by softswitch utility does not match configuration stored from software.
Correction:	Change MAC address on softswitch utility or in software.
Error Code:	11
Name:	LAN Softswitch/Modem mismatch
Description:	Configuration of LAN module does not match modem type or configuration programmed by softswitch utility.
Correction:	(1) Correct configuration of modem type. (2) Consult LAN Interface manual for configuration setup.
Error Code:	17
Name:	Invalid Memory Reference
Description:	Memory references in the logic program exceed what is available.
Correction:	Update the configuration file and store it to the PLC.
Error Code:	1E
Name:	Reference Length Mismatch
Description:	The PLC generates this error when the I/O reference lengths specified in the configuration for this module do not match the actual data sizes reported by the board.
Correction:	Update the configuration file with the correct reference lengths.

Error Code:	1F
Name:	Invalid Configuration Parameters
Description:	The PLC generates this error when it determines that critical values in the module's configuration are unacceptable.
Correction:	Update the configuration file with the correct values.
Error Code:	20
Name:	New Configuration Requires Reset
Description:	The PLC generates this error when it determines that a store of configuration attempted to change critical configuration values for the specified module. The new configuration will not take effect until the module is reset.
Correction:	(1) Power-cycle the entire system. (2) Power-cycle the rack containing the module.
Error Code:	27
Name:	Unresolved or Disabled Interrupt Reference
Description:	The PLC generates this error when an interrupt trigger reference is either out of range or disabled in the I/O module's configuration.
Correction:	(1) Remove or correct the interrupt trigger reference. (2) Update the configuration file to enable this particular interrupt.
Error Code:	1D
Name:	Incompatible Scheduling Mode
Description:	A program with a scheduling mode that is incompatible with the sweep mode has been stored. Logged on a Stop-to-Run transition.
Correction:	(1) Change the sweep mode and try again. (2) Change the scheduling mode or delete the offending program(s) from the program declaration screen.
Error Code:	24
Name:	I/O Specification Mismatch
Description:	The I/O specification of a program does not match the specification given in the program.
Correction:	Correct the mismatch between the I/O specification by changing the I/O specification declaration or the corresponding macro declaration in the C program source file.
Error Code:	25
Name:	Controller Reference Out of Range
Description:	A reference on either the trigger, disable, or I/O specification is out of the configured limits.
Correction:	Modify the incorrect reference to be within range, or increase the configured size of the reference data.
Error Code:	26
Name:	Bad Program Specification
Description:	The I/O specification of a program is corrupted.
Correction:	Contact GE Fanuc Field Service.
Error Code:	All Others
Name:	Module and Configuration Do not Match
Description:	The PLC generates this fault when the module occupying a slot is not of the same type that the configuration file indicates.
Correction:	(1) Replace the module in the slot with the type indicated in the configuration file. (2) Update the configuration file.

System Bus Error

The fault group System Bus Error occurs when the PLC CPU receives a non-configurable interrupt bus error from the bus system. The default fault action is Diagnostic.

Error Code:	4
Name:	Unrecognized VME Interrupt Source
Description:	The PLC generates this error when a module generates an interrupt not expected by the CPU (unconfigured or unrecognized).
Correction:	(1) Ensure that all modules configured for interrupts have corresponding interrupt declarations in the program logic. (2) Ensure that no third-party VME module is generating interrupts on the IRQ6 and IRQ7 lines.
Error Code:	All Others
Name:	System Bus Error
Description:	The PLC generates this fault when it has detected an error signal on the VME backplane, such as a parity error.
Correction:	(1) Ensure that all expansion rack cables are properly connected and seated. (2) Take action to minimize system noise.

PLC CPU Hardware Failure

The fault group PLC CPU Hardware occurs when the PLC CPU detects a hardware failure, such as a RAM failure or a communications port failure. When the failure is a RAM failure, the address of the failure is stored in the first four bytes of the Fault Extra Data field.

When a PLC CPU Hardware failure occurs, the PLC OK LED will flash on and off to indicate that the failure was not serious enough to prevent programmer communications to retrieve the fault information. The default fault action for this group is Fatal.

Error Code:	6Eh
Name:	Time-of-Day Clock not Battery-Backed
Description:	The battery-backed value of the time-of-day clock has been lost.
Correction:	(1) Replace the battery. Do not remove power from the main rack until replacement is complete. Reset the time-of-day clock using your programming software. (2) Replace the module.
Error Code:	All Others
Correction:	Replace the module.

Module Hardware Failure

The fault group Module Hardware Failure occurs when the PLC CPU detects a non-fatal hardware failure on any module in the system, for example, a serial port failure on a PCM. The default fault action for this group is Diagnostic.

Error Code:	1A0
Name:	Missing 12 Volt Power Supply
Description:	A power supply that supplies 12 volts is required to operate the LAN Interface module.
Correction:	(1) Install/replace a GE Fanuc 100 watt power supply. (2) Connect an external VME power supply that supplies 12 volts.
Error Code:	1C2 -1C6
Name:	LAN Interface Hardware Failure
Description:	Refer to the LAN Interface manual, GFK-0868 or GFK-0869 (previously GFK-0533), for a description of these errors.
Error Code:	All Others
Name:	Module Hardware Failure
Description:	A module hardware failure has been detected.
Correction:	Replace the affected module.

Option Module Software Failure

The fault group Option Module Software Failure occurs when a non-recoverable software failure occurs on a PCM. It is also generated when the identification data read from a module indicates that the module is a GE Fanuc module but the module type is not a supported GE Fanuc type. The default fault action for this group is Fatal.

Error Code:	1
Name:	Unsupported Board Type
Description:	The PLC generates this fault when the identification data read from a board indicates that the board is a GE Fanuc board but the type of board is not one of the GE Fanuc board types.
Correction:	(1) Upload the configuration file and verify that the software recognizes the board type in the file. If there is an error, correct it, download the corrected configuration file, and retry. (2) Display the PLC fault table on the programmer. Contact GE Fanuc PLC Field Service, giving them all the information contained in the fault entry.
Error Code:	2, 3
Name:	COMMREQ Frequency Too High
Description:	COMMREQs are being sent to a module faster than it can process them.
Correction:	Change the PLC program to send COMMREQs to the affected module at a slower rate or monitor the completion status of each COMMREQ before sending the next.
Error Code:	4
Name:	More Than One BTM in a Rack
Description:	There is more than one BTM present in the rack.
Correction:	Remove one of the BTMs from the rack; there can only be one in a CPU rack.
Error Code:	191, 195
Name:	LAN Interface Software Failure
Description:	Refer to the LAN Interface manual, GFK-0868 or GFK-0869 (previously GFK-0533), for a description of these errors.
Error Code:	All Others
Name:	Option Module Software Failure
Description:	Software failure detected on an option module.
Correction:	(1) Reload software into the indicated module. (2) Replace the module.

Program or Block Checksum Failure

The fault group Program or Block Checksum Failure occurs when the PLC CPU detects error conditions in program or blocks received by the PLC. It also occurs during Run mode background checking. In all cases, the Fault Extra Data field of the PLC fault table record contains the name of the program or block in which the error occurred. The default fault action for this group is Fatal.

Error Code:	All
Name:	Program or Block Checksum Failure
Description:	The PLC generates this error when a program or block is corrupted.
Correction:	(1) Clear PLC memory and retry the store. (2) Examine C application for errors. (3) Display the PLC fault table on the programmer. Contact GE Fanuc PLC Field Service, giving them all the information contained in the fault entry.

Low Battery Signal

The fault group Low Battery Signal occurs when the PLC CPU detects a low battery on the PLC CPU board, the PLC CPU memory daughter board, or a module such as the PCM reports a low battery condition. The default fault action for this group is Diagnostic.

Error Code:	0
Name:	Failed Battery Signal
Description:	The CPU module (or other module having a battery) battery is dead.
Correction:	Replace the battery. Do not remove power from the rack until replacement is complete.
Error Code:	1
Name:	Low Battery Signal
Description:	A battery on the CPU or other module has a low signal.
Correction:	Replace the battery. Do not remove power from the rack until replacement is complete.

Constant Sweep or Microcycle Time Exceeded

The fault group Constant Sweep or Microcycle Time Exceeded occurs when the PLC CPU operates in Constant Sweep or Microcycle mode and detects that the sweep has exceeded the constant sweep timer. The fault extra data contains the name of the folder in eight bytes. The default fault action for this group is Diagnostic.

Error Code:	0 Constant Sweep 1 Microcycle
Correction:	If Constant Sweep: (1) Increase constant sweep time. (2) Remove logic from application program. If Microcycle: (1) Increase the time. (2) Modify execution intervals to give programs more time to execute.

PLC System Fault Table Full

The fault group PLC System Fault Table Full occurs when the PLC Fault Table reaches its maximum configured limit (see page 3-13). The default fault action for this group is Diagnostic.

Error Code:	0
Correction:	Clear the PLC fault table.

I/O Fault Table Full

The fault group I/O Fault Table Full occurs when the I/O Fault Table reaches its maximum configured limit (see page 3-14). To avoid loss of additional faults, clear the earliest entry from the table. The default fault action for this group is Diagnostic.

Error Code:	0
Correction:	Clear the I/O fault table.

Application Fault

The fault group Application Fault occurs when the PLC CPU detects a fault in the user program. The default fault action for this group is Diagnostic.

Error Code:	1
Name:	Indirect Address Out of Range
Description:	The PLC generates this error when one of the parameters to a function block is an indirect reference (that is, the parameter is an address within that memory type which contains the parameter value) and the contents of the indirect reference are out of range for the memory type. For example, consider a system with 500 %R registers defined. This fault would be generated if the parameter address were %R00100, and the contents of %R00100 were greater than 500 or zero. The Fault Extra Data field contains in the first two bytes the offset address of where the call was made, the segment selector and offset (reference) in the next four bytes, and the name of the program or block in which the function call resides in the next eight bytes.
Correction:	(1) Correct the indirect reference. (2) Increase the number of registers available, if possible.
Error Code:	2
Name:	Software Watchdog Timer Expired
Description:	The PLC generates this error when the watchdog timer expires. The PLC CPU stops executing the user program and enters Stop mode. The only recovery is to cycle power to the PLC CPU. Examples causing timer expiration: Looping, via jump, very long program, etc.
Correction:	(1) Determine what caused the expiration (logic execution, external event, etc.) and correct. (2) Use the system service function block to restart the watchdog timer.
Error Code:	5
Name:	COMMREQ WAIT Mode Not Supported
Description:	The module receiving the COMMREQ does not support WAIT mode COMMREQs.
Correction:	Use NOWAIT mode COMMREQs.
Error Code:	6
Name:	COMMREQ Bad Task ID
Description:	The task selected by the COMMREQ does not exist on the option module.
Correction:	Correct the task ID.
Error Code:	7
Name:	Application Stack Overflow
Description:	Block call depth has exceeded the PLC capability.
Correction:	Increase the program's stack size or adjust application program to reduce nesting.
Error Code:	8 through D
Name:	LAN Interface Application Faults
Description:	Refer to the LAN Interface manual, GFK-0868 or GFK-0869 (previously GFK-0533), for a description of these errors.

Error Code:	1C
Name:	Program Exceeded Wind Down
Description:	A program failed to complete execution within the wind-down period (currently 2.5 seconds) after the PLC was commanded to stop.
Correction:	A program has gone into an infinite loop or is taking too long to execute. Correct the coding error or modify the program.
Error Code:	1D
Name:	Program Not Readied
Description:	A program scheduled to be readied has not completed its previous execution. The base cycle time is too small (Periodic programs), or the interrupt rate is too high (I/O-Triggered or Timed programs).
Correction:	(1) Increase the base cycle time or decrease the interrupt rate. (2) Increase the execution interval time to allow the program to finish execution.
Error Code:	0E
Name:	External Block Run-Time Error
Description:	A run-time error occurred during execution of an external block.
Correction:	Based on the fault information, correct the specific problem in the external block.
Error Code:	0F
Name:	SORT Interrupt Error
Description:	A SORT function executed in a timed or I/O interrupt at the same time a SORT function was executing in another block.
Correction:	Do not use the SORT function in both Interrupt and Non-Interrupt blocks.
Error Code:	11
Name:	Standalone Run-Time Error
Description:	A run-time error occurred during execution of a Standalone program.
Correction:	Based on the fault information, correct the specific problem in the external block.

Non-Configurable Faults

The fault action of Non-Configurable Faults cannot be changed. Fatal faults cause the PLC to enter a form of Stop mode at the end of the sweep in which the error occurred. Diagnostic faults are logged and corresponding fault contacts are set. Informational faults are simply logged in the PLC fault table.

System Bus Failure

The fault group System Bus Failure occurs when the PLC CPU software receives the non-configurable interrupt bus failure from the bus system. The default fault action for this group is Fatal.

Error Code:	1
Name:	Bus Grant Failure
Description:	The PLC operating software generates this error when the PLC CPU is unable to obtain control of the VME bus when required.
Correction:	(1) Ensure that any non-GE Fanuc boards which can become bus masters are relinquishing control of the VME bus when requested to do so by the PLC CPU. (2) Replace the PLC CPU module.

No User Program on Power-Up

The fault group No User Program on Power-Up occurs when the PLC CPU powers up with its memory preserved but no user program exists in the PLC. The PLC CPU detects the absence of a user program on power-up; the controller stays in Stop mode, performing the Stop mode sweep until a valid program is downloaded. The default fault action for this group is Informational.

Correction:	Download an application program before attempting to go to Run mode.

Corrupted User Program on Power-Up

The fault group Corrupted User Program on Power-Up occurs when the PLC CPU detects corrupted user RAM. The PLC CPU will remain in Stop mode until a valid user program and configuration file are downloaded. The default fault action for this group is Fatal.

Error Code:	1
Name:	Corrupted User RAM on Power-Up
Description:	The PLC generates this error when it detects corrupted user RAM on power-up.
Correction:	(0) Cycle power without battery. (2) Examine any C applications for errors. (3) Replace the battery on the PLC CPU. (4) Replace the expansion memory board on the PLC CPU. (5) Replace the PLC CPU.
Error Code:	2
Name:	Illegal Boolean OpCode Detected
Description:	The PLC generates this error when it detects a bad instruction in the user program.
Correction:	(1) Restore the user program and references, if any. (2) Examine any C applications for errors. (3) Replace the expansion memory board on the PLC CPU. (4) Replace the PLC CPU.
Error Code:	6
Name:	Corrupted Remote I/O Scanner EEPROM
Description:	The configuration in the Remote I/O Scanner EEPROM was found to be corrupted at power-up.
Correction:	Restore the Remote I/O Scanner configuration.

Window Completion Failure

The fault group Window Completion Failure is generated by the pre-logic and end-of-sweep processing software in the PLC. The fault extra data contains the name of the task that was executing when the error occurred. The default fault action for this group is Informational.

Error Code:	0
Name:	Window Completion Failure
Description:	The PLC generates this error when the PLC is operating in Constant Sweep mode and the constant sweep time was exceeded before the programmer window had a chance to begin executing.
Correction:	Increase the constant sweep timer value.
Error Code:	1
Name:	Logic Window Skipped
Description:	The logic window was skipped due to lack of time to execute.
Correction:	(1) Increase base cycle time. (2) Reduce Communications Window time.

Password Access Failure

The fault group Password Actual Failure occurs when the PLC CPU receives a request to change to a new privilege level and the password included with the request is not valid for that level. The default fault action for this group is Informational.

Error Code:	0
Correction:	Retry the request with the correct password.

Null System Configuration for Run Mode

The fault group Null System Configuration for Run Mode occurs when the PLC transitions from Stop to one of the Run modes and a configuration file is not present. The transition to Run is permitted, but no I/O scans occur. The effect of this fault is to perform the function of a Suspend I/O. The default fault action for this group is Informational.

Error Code:	0
Correction:	Download a configuration file.

PLC CPU System Software Failure

Faults in the fault group PLC CPU System Software Failure are generated by the operating software of the Series 90-70 PLC CPU. They occur at many different points of system operation. When a fatal fault occurs, the PLC CPU immediately transitions into a special Error Sweep mode. The only activity permitted when the PLC is in this mode is communications with the programmer. The only method of clearing this condition is to cycle power on the PLC. The default fault action for this group is Fatal.

Error Code:	14, 27
Name:	Corrupted PLC Program Memory
Description:	The PLC generates these errors when certain PLC operating software problems occur. These should *not* occur in a production system.
Correction:	(1) Display the PLC fault table on the programmer. Contact GE Fanuc PLC Field Service, giving them all the information contained in the fault entry. (2) Perform the corrections for corrupted memory.
Error Code:	52
Name:	Backplane Communications Failed
Description:	The PLC generates this error when it attempts to comply with a request that requires backplane communications and receives a rejected mail response.
Correction:	(1) Check the bus for abnormal activity. (2) Replace the intelligent option module to which the request was directed. (3) Check parallel programmer cable for proper attachment.
Error Code:	5A
Name:	User Shut Down Requested
Description:	The PLC generates this informational alarm when SVCREQ #13 (User Shut Down) executes in the application program.
Correction:	None required. Information-only alarm.
Error Code:	7B
Name:	Remote I/O Scanner Communications Heartbeat Failure
Description:	Refer to the *Series 90-70 Remote I/O Scanner User's Manual*, GFK-0579, for a description of this error.
Correction:	None required. Information-only alarm.
Error Code:	94
Name:	Units Contain Mismatched Firmware, Update Recommended
Description:	This fault is logged each time the redundancy state changes and the two 780 CPUs contain incompatible firmware.
Correction:	Ensure that both 780 CPUs have compatible firmware.
Error Code:	All Others
Name:	PLC CPU Internal System Error
Description:	An internal system error has occurred that should **not** occur in a production system.
Correction:	Display the PLC fault table on the programmer. Contact GE Fanuc PLC Field Service, giving them all the information contained in the fault entry.

Too Many Bus Controllers

The fault group Too Many Bus Controllers occurs when the I/O Scanner portion of the PLC operating software detects that more than the maximum number (32) of bus controllers has been defined. The PLC CPU itself is a bus controller for the Model 70 I/O present in the system. The default fault action for this group is Fatal.

Note

Genius bus controllers which are configured for redundant and non-redundant blocks count as two bus controllers.

Correction:	(1)	Determine which modules are bus controllers and remove the extra ones.
	(2)	Delete a bus controller from the configuration file and store the file to the PLC CPU.
	(3)	If bus controllers have been moved from one slot in the rack to a different slot and this error did not occur before the move, cycle power on the rack. **No module should be inserted with power applied to rack.**
	(4)	Display the PLC fault table on the programmer. Contact GE Fanuc PLC Field Service, giving them all the information contained in the fault entry.

Communications Failure During Store

The fault group Communications Failure During Store occurs during the store of programs or blocks and other data to the PLC. The stream of commands and data for storing programs or blocks and data starts with a special start-of-sequence command and terminates with an end-of-sequence command. If communications with the programming device performing the store is interrupted or any other failure occurs which terminates the store, this fault is logged. As long as this fault is present in the system, the controller will not transition to Run mode.

This fault is *not* automatically cleared on power-up; the user must specifically clear the condition. The default fault action for this group is Fatal.

Error Code:	0
Correction:	Clear the fault and retry the download of the program or configuration file.

Run Mode Store Failure

Error Code:	1
Description:	Communications was lost, or power was lost during a Run Mode Store. The new program or block was not activated and was deleted.
Correction:	Perform the Run Mode Store again. This fault is diagnostic.
Error Code:	2
Description:	Communications was lost, or power was lost during the cleanup of old programs or blocks during a Run Mode Store. The new program or block is installed, and the remaining programs and blocks were cleaned up.
Correction:	None required. This fault is informational.
Error Code:	3
Description:	Power was lost in the middle of a Run Mode Store.
Correction:	Delete and restore the program. This error is fatal.

Section 4: I/O Fault Table Explanations

The I/O fault table reports data about faults in three classifications:

- Fault category
- Fault type
- Fault description

All faults have a fault category, but a fault type and fault group may not be listed for every fault. For additional information pertaining to each fault, double-click the fault to access its Details window.

The following table describes the information provided with each fault category.

Table 3-8. Fault Category Descriptions

Fault Category	Fault Type	Fault Description	Fault Specific Data
Circuit Fault	Discrete Fault	Loss of User Side Power	Circuit Configuration *
		Short Circuit in User Wiring	Circuit Configuration *
		Sustained Overcurrent	Circuit Configuration *
		Low or No Current Flow	Circuit Configuration *
		Switch Temperature Too High	Circuit Configuration *
		Switch Failure	Circuit Configuration *
		Point Fault	Circuit Configuration *
		Output Fuse Blown	Circuit Configuration *
	Analog Fault	Input Channel Low Alarm	Circuit Configuration *
		Input Channel High Alarm	Circuit Configuration *
		Input Channel Under Range	Circuit Configuration *
		Input Channel Over Range	Circuit Configuration *
		Input Channel Open Wire	Circuit Configuration *
		Output Channel Under Range	Circuit Configuration *
		Output Channel Over Range	Circuit Configuration *
		Invalid Data	Circuit Configuration *
		Expansion Channel Not Responding	Circuit Configuration *
	Low-Level Analog Fault	Input Channel Low Alarm	Circuit Configuration *
		Input Channel High Alarm	Circuit Configuration *
		Input Channel Under Range	Circuit Configuration *
		Input Channel Over Range	Circuit Configuration *
		Input Channel Open Wire	Circuit Configuration *
		Wiring Error	Circuit Configuration *
		Internal Fault	Circuit Configuration *
		Input Channel Shorted	Circuit Configuration *
		Invalid Data	Circuit Configuration *
	GENA Fault	GENA Circuit Fault	GENA Fault Byte 2
	Remote I/O Scanner Fault	Remote I/O Scanner Circuit Fault	Byte 1: Circuit Type Byte 2: I/O Type
Loss of IOC			Timeout Unexpected State Unexpected Mail Status VME Bus Error
Addition of IOC			
Loss of I/O Module			
Addition of I/O Module			
Extra I/O Module			
Loss of Block	Fault Not Specified	Communications Lost	
Addition of Block			
Extra Block			
I/O Bus Fault	Bus Fault Bus Outputs Disabled		
Global Memory Fault			Subnet Group Number Global Variable Name

* Refer to table on next page.

Table 3-8. - Continued Fault Category Descriptions

Fault Category	Fault Type	Fault Description	Fault Specific Data
Module Fault	Headend Fault	EPROM or NVRAM Failure Calibration Memory Failure Shared Ram Failure Configuration MisMatch Watchdog Timeout Output Fuse Blown	
IOC Software Fault			
IOC Hardware Failure			
Forced Circuit			Block Configuration * Discrete/Analog Indication*
Unforced Circuit			Block Configuration * Discrete/Analog Indication*

*Refer to table below.

Three types of fault specific data occur in more than one fault category; they are block configuration, circuit configuration, and analog/discrete indication. The codings are shown in the following table.

Value	Description
	Circuit Configuration
1	Circuit is an input.
2	Circuit is an output.
3	Circuit is an output with feedback.
	Block Configuration
1	Block is configured for inputs only.
2	Block is configured for outputs only.
3	Block is configured for inputs and outputs (grouped block).
	Discrete/Analog Indication
1	Block is a discrete block.
2	Block is an analog block.

Circuit Fault

Circuit Fault has four fault types. Three of the four fault types have fault descriptions. Fault specific data is available for all faults. Circuit faults apply specifically to Genius I/O modules. The default fault action is Diagnostic. The following table describes the circuit fault category.

Table 3-9. Circuit Fault Category Description

Fault Category	Fault Type	Fault Description	Fault Specific Data
Circuit Fault	Discrete Fault	Loss of User Side Power	Circuit Configuration
		Short Circuit in User Wiring	Circuit Configuration
		Sustained Overcurrent	Circuit Configuration
		Low or No Current Flow	Circuit Configuration
		Switch Temperature Too High	Circuit Configuration
		Switch Failure	Circuit Configuration
		Point Fault	Circuit Configuration
		Output Fuse Blown	Circuit Configuration
	Analog Fault	Input Channel Low Alarm	Circuit Configuration
		Input Channel High Alarm	Circuit Configuration
		Input Channel Under Range	Circuit Configuration
		Input Channel Over Range	Circuit Configuration
		Input Channel Open Wire	Circuit Configuration
		Output Channel Under Range	Circuit Configuration
		Output Channel Over Range	Circuit Configuration
		Invalid Data	Circuit Configuration
		Expansion Channel Not Responding	Circuit Configuration
	Low-Level Analog Fault	Input Channel Low Alarm	Circuit Configuration
		Input Channel High Alarm	Circuit Configuration
		Input Channel Under Range	Circuit Configuration
		Input Channel Over Range	Circuit Configuration
		Input Channel Open Wire	Circuit Configuration
		Wiring Error	Circuit Configuration
		Internal Fault	Circuit Configuration
		Input Channel Shorted	Circuit Configuration
		Invalid Data	Circuit Configuration
	Remote Fault	Remote I/O Scanner Fault	

Discrete Fault

Discrete Fault has eight fault descriptions. More than one condition may be present in a particular reporting of the fault.

Name:	Loss of User Side Power
Description:	The Genius Bus Controller generates this error when there is a power loss on the field wiring side of a Genius I/O block.
Correction:	Correct the power failure.
Name:	Short Circuit in User Wiring
Description:	The Genius Bus Controller generates this error when it detects a short circuit in the user wiring of a Genius block. A short circuit is defined as a current level greater than 20 amps.
Correction:	Fix the cause of the short circuit.
Name:	Sustained Overcurrent
Description:	The Genius Bus Controller generates this error when it detects a sustained current level greater than 2 amps in the user wiring.
Correction:	Fix the cause of the over current.
Name:	Low or No Current Flow
Description:	The Genius Bus Controller generates this error when there is very low or no current flow in the user circuit.
Correction:	Fix the cause of the condition.
Name:	Switch Temperature Too High
Description:	The Genius Bus Controller generates this error when the Genius block reports a high temperature in the Genius Smart Switch.
Correction:	(1) Ensure that the block is installed to provide adequate circulation. (2) Decrease the ambient temperature surrounding the block.
Name:	Switch Failure
Description:	The Genius Bus Controller generates this error when the Genius block reports a failure in the Genius Smart Switch.
Correction:	Replace the Genius I/O block.
Name:	Point Fault
Description:	The PLC generates this error when it detects a failure of a single I/O point on a Genius I/O module.
Correction:	Replace the Genius I/O block.
Name:	Output Fuse Blown
Description:	The PLC generates this error when it detects a blown fuse on a Genius I/O output block.
Correction:	(1) Determine and repair the cause of the fuse blowing, and replace the fuse. (2) Replace the block.

Analog Fault

Analog Fault has nine fault descriptions. More than one condition may be present in a particular reporting of the fault.

Name:	Input Channel Low Alarm
Description:	The Genius Bus Controller generates this error when the Genius Analog module reports a low alarm on an input channel.
Correction:	Correct the condition causing the low alarm.
Name:	Input Channel High Alarm
Description:	The Genius Bus Controller generates this error when the Genius Analog module reports a high alarm on an input channel.
Correction:	Correct the condition causing the high alarm.
Name:	Input Channel Under Range
Description:	The Genius Bus Controller generates this error when the Genuis Analog module reports an under-range condition on an input channel.
Correction:	Correct the problem causing the condition.
Name:	Input Channel Over Range
Description:	The Genius Bus Controller generates this error when the Genuis Analog module reports an over-range condition on an input channel.
Correction:	Correct the problem causing the condition.
Name:	Input Channel Open Wire
Description:	The Genius Bus Controller generates this error when the Genuis Analog module detects an open wire condition on an input channel.
Correction:	Correct the problem causing the condition.
Name:	Output Channel Under Range
Description:	The Genius Bus Controller generates this error when the Genuis Analog module reports an under-range condition on an output channel.
Correction:	Correct the problem causing the condition.
Name:	Output Channel Over Range
Description:	The Genius Bus Controller generates this error when the Genuis Analog module reports an over-range condition on an output channel.
Correction:	Correct the problem causing the condition.
Name:	Invalid Data
Description:	The Genius Bus Controller generates this error when it detects invalid data from a Genuis Analog input block.
Correction:	Correct the problem causing the condition.
Name:	Expansion Channel Not Responding
Description:	The PLC generates this error when data from an expansion channel on a multiplexed analog input board is not responding.
Correction:	(1) Check wiring to the module. (2) Replace the module.

Low-Level Analog Fault

Low-Level Analog Fault has nine fault descriptions. More than one condition may be present in a particular reporting of the fault.

Name:	Input Channel Low Alarm
Description:	The Genius Bus Controller generates this error when the Genuis Analog module reports a low alarm on an input channel.
Correction:	Correct the condition causing the low alarm.
Name:	Input Channel High Alarm
Description:	The Genius Bus Controller generates this error when the Genuis Analog module reports a high alarm on an input channel.
Correction:	Correct the condition causing the high alarm.
Name:	Input Channel Under Range
Description:	The Genius Bus Controller generates this error when the Genuis Analog module reports an under-range condition on an input channel.
Correction:	Correct the problem causing the condition.
Name:	Input Channel Over Range
Description:	The Genius Bus Controller generates this error when the Genuis Analog module reports an over-range condition on an input channel.
Correction:	Correct the problem causing the condition.
Name:	Input Channel Open Wire
Description:	The Genius Bus Controller generates this error when the Genuis Analog module detects an open wire condition on an input channel.
Correction:	Correct the problem causing the condition.
Name:	Wiring Error
Description:	The Genius Bus Controller generates this error when the Genuis Analog module detects an improper RTD connections or thermocouple reverse junction fault.
Correction:	Correct the problem causing the condition.
Name:	Internal Fault
Description:	The Genius Bus Controller generates this error when the Genuis Analog module reports a cold junction sensor fault on a thermocouple block or an internal error in an RTD block.
Correction:	Correct the problem causing the condition.
Name:	Input Channel Shorted
Description:	The Genius Bus Controller generates this error when it detects an input channel shorted on a Genius RTD or Strain Gauge Block.
Correction:	Correct the problem causing the condition.
Name:	Invalid Data
Description:	The Genius Bus Controller generates this error when it detects invalid data from a Genuis Analog input block.
Correction:	Correct the problem causing the condition.

GENA Fault

The GENA Fault has no fault descriptions associated with it. GENA Fault Byte 2 is the first byte of the fault specific data.

Description:	The Genius I/O operating software generates this error when it detects a failure in a GENA block attached to the Genius I/O bus.
Correction:	Replace the GENA block.

Loss of IOC (I/O Controller)

The fault category Loss of IOC has no fault types or fault descriptions associated with it. The default fault action is Fatal.

Name:	Loss of or Missing IOC
Description:	The PLC generates this error when it cannot communicate with an I/O Controller and an entry for the IOC exists in the configuration file.
Correction:	(1) Verify that the module in the slot/bus address is the correct module. (2) Review the configuration file and verify that it is correct. (3) Replace the module. (4) Display the PLC fault table on the programmer. Contact GE Fanuc PLC Field Service, giving them all the information contained in the fault entry.

Addition of IOC (I/O Controller)

The fault category Addition of I/O Module has no fault types or fault descriptions associated with it. The default fault action for this category is Diagnostic.

Name:	Addition of IOC
Description:	The PLC generates this error when an IOC which has been faulted returns to operation or when an IOC is found in the system and the configuration file indicates that no IOC is to be in that slot.
Correction:	(1) No action is necessary if the faulted module is in a remote rack and is returning due to a remote rack power cycle. (2) Update the configuration file or remove the module.

Loss of I/O Module

The fault category Loss of I/O Module applies to Model 70 I/O discrete and analog modules. There are no fault types or fault descriptions associated with this category. The default fault action is Diagnostic.

Name:	Loss of I/O Module
Description:	The PLC generates this error when it detects that a Model 70 I/O module is no longer responding to commands from the PLC CPU, or when the configuration file indicates an I/O module is to occupy a slot and no module exists in the slot.
Correction:	(1) Replace the module. (2) Correct the configuration file. (3) Display the PLC fault table on the programmer. Contact GE Fanuc PLC Field Service, giving them all the information contained in the fault entry.

Addition of I/O Module

The fault category Addition of I/O Module applies to Model 70 discrete and analog I/O modules. There are no fault types or fault descriptions associated with this category. The default fault action is Diagnostic.

Name:	Addition of I/O Module
Description:	The PLC generates this error when an I/O module which had been faulted returns to operation.
Correction:	(1) No action necessary if module was removed or replaced or if the remote rack was power cycled. (2) Update the configuration file or remove the module.

Extra I/O Module

The fault category Extra I/O Module applies only to Model 70 I/O modules. There are no fault types or fault descriptions associated with this category. The default fault action is Diagnostic.

Name:	Extra I/O Module
Description:	The PLC generates this error when it detects a Model 70 I/O module in a slot which the configuration file indicates should be empty.
Correction:	(1) Remove the module. (It may be in the wrong slot.) (2) Update and restore the configuration file to include the extra module.

Loss of Block

The fault category Loss of Block applies to Genius blocks. There are no fault types or fault descriptions associated with this category. The default fault action is Diagnostic.

Name:	Loss of Block
Description:	The PLC generates this error when it receives a Loss of Block fault from a Genius Bus Controller but the reason for the loss is unspecified.
Correction:	(1) Verify power and wiring to the block. (2) Replace the block.
Name:	Loss of Block -A/D Communications Fault
Description:	The Genius I/O operating software generates this error when it detects a loss of communications with a Genius I/O block.
Correction:	(1) Verify power and serial bus wiring to the block. (2) Replace the block.

Addition of Block

The fault category Addition of Block applies only to Genius blocks. There are no fault types or fault descriptions associated with this category. The default fault action is Diagnostic.

Name:	Addition of Block
Description:	The Genius operating software generates this error when it detects that a Genius block which stopped communicating with the controller starts communicating again.
Correction:	Informational only. None required.

Extra Block

The fault category Extra Block applies only to Genius I/O blocks. There are no fault types or fault descriptions associated with this category. The default fault action is Diagnostic.

Name:	Extra Block
Description:	The PLC operating software generates this error when it detects a Genius I/O block on the bus at a serial bus address which the configuration file should not have a block.
Correction:	(1) Remove or reconfigure the block. (It may be at the wrong serial bus address.) (2) Update and restore the configuration file to include the extra block.

I/O Bus Fault

The fault category I/O Bus Faults has two fault types associated with it. The default fault action is Diagnostic.

Name:	Bus Fault
Description:	The Genius Bus Controller operating software generates this error when it detects a failure with a Genius I/O bus.
Correction:	(1) Determine the reason for the bus failure and correct it. (2) Replace the Genius Bus Controller.
Name:	Bus Outputs Disabled
Description:	The Genius Bus Controller operating software generates this error when it times out waiting for the PLC CPU to perform an I/O scan.
Correction:	(1) Replace the PLC CPU. (2) Display the PLC fault table on the programmer. Contact GE Fanuc PLC Field Service, giving them all the information contained in the fault entry.
Name:	SBA Conflict
Description:	The Genius Bus Controller detected a conflict between its serial bus address and that of another device on the bus.
Correction:	Adjust one of the conflicting serial bus addresses.

Module Fault

The fault category Module Fault has one fault type, one headend fault, and eight fault descriptions. No fault specific data is present. The default fault action for this category is Diagnostic.

Name:	Configuration Memory Failure
Description:	The Genius Bus Controller generates this error when it detects a failure in a Genius block's EEPROM or NVRAM.
Correction:	Replace the Genius block's electronics module.
Name:	Calibration Memory Failure
Description:	The Genius Bus Controller generates this error when it detects a failure in a Genius block's calibration memory.
Correction:	Replace the Genius block's electronics module.
Name:	Shared RAM Fault
Description:	The Genius Bus Controller generates this error when it detects an error in a Genius block's shared RAM.
Correction:	Replace the Genius block's electronics module.
Name:	Watchdog Timeout
Description:	The PLC generates this error when it detects that a Model 70 input module watchdog timer has expired.
Correction:	Replace the Model 70 input module.
Name:	Output Fuse Blown
Description:	The PLC generates this error when it detects a blown fuse on a Model 70 output module.
Correction:	(1) Determine and repair the cause of the fuse blowing, and replace the fuse. (2) Replace the module.
Name:	Module Fault
Description:	An internal failure has been detected in a module.
Correction:	Replace the affected module.

IOC (I/O Controller) Software Fault

The fault category IOC Software Fault applies to any type of I/O Controller. There are no fault types or fault descriptions associated with it. The default fault action is Fatal.

Name:	Datagram Queue Full, Read/Write Queue Full
Description:	Too many datagrams or read/write requests have been sent to the Genius Bus Controller.
Correction:	Adjust the system to reduce the request rate to the Genius Bus Controller.
Name:	Response Lost
Description:	The Genius Bus Controller is unable to respond to a received datagram or read/write request.
Correction:	Adjust the system to reduce the request rate to the Genius Bus Controller.

IOC (I/O Controller) Hardware Failure

The fault category IOC Hardware Fault has no fault types or fault descriptions. The default fault action is Diagnostic.

Description:	The Genius operating software generates this error when it detects a hardware failure in the Bus Communication hardware or a baud rate mismatch.
Correction:	(1) Verify that the baud rate set in the configuration file for the Genius Bus Controller agrees with the baud rate programmed in every block on the bus. (2) Change the configuration file and restore it, if necessary. (3) Replace the Genius Bus Controller. (4) Selectively remove each block from the bus until the offending block is isolated then replace it.

Forced and Unforced Circuit

The fault categories Forced Circuit and Unforced Circuit report point conditions and therefore are not technically faults. They have no fault types or fault descriptions. These reports occur when a Genius I/O point was forced or unforced with the Hand-Held Monitor. The default fault action is Informational.

Fault Specific Data contains data as shown below.

Byte Number	Description
1	Block Configuration
2	Analog/Discrete Information

Block Switch

The Fault Category Block Switch has no fault types or fault descriptions. The default fault action is Diagnostic.

Name:	Block Switch
Description:	The PLC generates this error when a Genius block on redundant Genius buses switches from one bus to another.
Correction:	(1) No action is required to keep the block operating. (2) The bus that the block switched from needs to be repaired. (a) Verify the bus wiring. (b) Replace the I/O controller. (c) Replace the Bus Switching Module (BSM).

Appendix A

CPU Performance Data

This appendix contains instruction and overhead timing for each Series 90-70 CPU module. This timing information can be used to predict CPU sweep times.

Instruction Timing

The Series 90-70 PLC supports many different functions and function blocks. Table A-1, beginning on the next page, lists the execution time in microseconds and the memory size in bytes for each function supported by these CPUs:

1. Model CPM 924/925
2. Model CPM 914/915
3. Model CPU 781/782
4. Model CPU 788/89
5. Models CPU 731R and later, CPU 732, CPU 771P and later, and CPU 772
6. Models CPX 935 and CGR 935
7. Model CPX 928
8. Models CPX 772 and CGR 772, and CPX 782

Note

All timing information for all CPUs was not available at the time this manual was printed. Additional information will be added to a future edition.

Two execution times are shown for each function:

Execution Time	Description
Enabled	Time required to execute the function or function block when power flows into and out of the function. Typically, best-case times are when the data used by the block is contained in user RAM (word-oriented memory).
Disabled	Time required to execute the function when it is not enabled.

Note

Timers are updated each time they are encountered in the logic by the amount of time consumed by the last sweep.

Table A-1. Instruction Timing

Function Group	Function	Enabled				Disabled				Increment				Size
		924/ 925	914/ 915	781/782 788/789	731/732 771/772	924/ 925	914/ 915	781/782 788/789	731/732 771/772	924/ 925	914/ 915	781/782 788/789	731/732 771/772	
Timers	ONDTR	6.5	10.5	34.5	78	5.5	8	22.5	53	-	-	-	-	18
	OFDT	6	9.5	32.5	68	6	9.5	30	65.5	-	-	-	-	15
	TMR	7.5	10	37	79	8.5	10.5	32	62	-	-	-	-	15
Counters	UPCTR	8.5	11	28.5	67	8	11	29	63	-	-	-	-	18
	DNCTR	6	9.5	28.5	66	6	9.5	28	62.5	-	-	-	-	18
Math	ADD (INT)	5	6	22	34.5	4	5.5	14.5	18	-	-	-	-	15
	ADD (DINT)	7	9.5	34.5	63.5	6	6.5	19.5	30.5	-	-	-	-	15
	SUB (INT)	5	6.5	17	35	4	4.5	10.5	19	-	-	-	-	15
	SUB (DINT)	8	10	36	63.5	6	7.5	20	30	-	-	-	-	15
	MUL (INT)	5	6.5	21	38	4	4.5	11.5	18.5	-	-	-	-	15
	MUL (DINT)	8	10.5	35.5	76.5	5.5	6.5	19.5	31	-	-	-	-	15
	DIV (INT)	5.5	7	22.5	41	4.5	4.5	14.5	19	-	-	-	-	15
	DIV (DINT)	8.5	12	41	82	5.5	6.5	18	30.5	-	-	-	-	15
	MOD (INT)	5.5	7.5	23.5	41	4	5	12.5	18.5	-	-	-	-	15
	MOD (DINT)	10.5	14	46.5	85	5.5	7.5	21	31	-	-	-	-	15
	ABS (INT)	5.5	7.5	25.5	50	4.5	5.5	18	25.5	-	-	-	-	12
	ABS (DINT)	7	8.5	28.5	51.5	5	7	18.5	26.5	-	-	-	-	12
	SQRT (INT)	10	14	40.5	82.5	4	4	11	16.5	-	-	-	-	12
	SQRT (DINT)	15.5	27.5	80.5	166	5	6.5	16.5	26.5	-	-	-	-	12
Relational	EQ (INT)	6	8	21	37	3.5	4.5	12.5	17.5	-	-	-	-	15
	EQ (DINT)	6.5	9	30	58.5	5	7	18.5	30.5	-	-	-	-	15
	NE (INT)	6	8	20	36.5	3.5	4.5	14	18.5	-	-	-	-	15
	NE (DINT)	6.5	10.5	31	58	5	7	20	30.5	-	-	-	-	15
	GT (INT)	6.5	9	24	43.5	4	4.5	13.5	17.5	-	-	-	-	15
	GT (DINT)	7.5	10.5	28.5	61	5	7.5	18	30.5	-	-	-	-	15
	GE (INT)	6.5	9	25	41.5	3.5	4.5	14	17.5	-	-	-	-	15
	GE (DINT)	6.5	9.5	31.5	58.5	5	7	20	30.5	-	-	-	-	15
	LT (INT)	7.5	9.5	27.5	43.5	3.5	4.5	14.5	17.5	-	-	-	-	15
	LT (DINT)	7.5	10.5	35	61	5	7	20	30.5	-	-	-	-	15
	LE (INT)	6	8.5	20	41.5	3.5	4.5	14	17.5	-	-	-	-	15
	LE (DINT)	6.5	10.5	32	58.5	5	7	21.5	30.5	-	-	-	-	15
	CMP (INT)	10.5	12.5	34.5	62	9	14	33.5	55.5	-	-	-	-	21
	CMP (DINT)	12	14.5	41	79.5	9	12	32	56	-	-	-	-	21
	RANGE (INT)	10.5	16.5	46	82.5	5.5	8	26	33	-	-	-	-	
	RANGE (DINT)	9	14.5	53.5	96.5	8	9	21.5	34	-	-	-	-	
Bit Operation	AND (WORD)	9.5	13.5	53.5	116.5	5	7	28.5	35	0.7	1.5	8.2	18.1	18
	AND(DWORD)	9.5	15	58	123.5	6	7.5	25.5	36	1	2	10.1	24.4	18
	OR (WORD)	8.5	14	57	118	5.5	7	25	35	0.8	1.6	9	19.6	18
	OR (DWORD)	9.5	15	63	126.5	6.5	8.5	24.5	36	1.2	2.3	11.7	27.5	18
	XOR (WORD)	8.5	14	58.5	118	5	7	24.5	35	0.8	1.6	8.9	19.3	18
	XOR (DWORD)	9.5	15	65	124.5	5.5	7.5	26.5	34.5	1.1	2.3	11.6	26.6	18
	NOT (WORD)	7	10.5	42.5	76.5	5.5	6.5	22	32.5	0.3	0.7	3.5	8.4	15
	NOT (DWORD)	7.5	11	41	77.5	5.5	7	25.5	32	0.3	0.7	3.6	10.3	15
	MCMP (WORD)	14	24.5	97.5	213	7	10.5	36.5	56.5	0.6	1.3	3.8	8.4	30
	MCMP (DWORD)	15.5	27	100	221	7.5	11	35	56	1.3	2.7	7.7	16.9	30
	SHL (WORD)	11	17.5	61	137.5	5.5	8.5	25.5	41.5	0.6	1	3.9	10.2	24
	SHL (DWORD)	12	18.5	66.5	147	6	8.5	27	43	1.2	2.1	7.9	20.5	24
	SHR (WORD)	10.5	16.5	60.5	137.5	5	8	26.5	41	0.6	1	3.9	10.1	24
	SHR (DWORD)	12	18	68	147	6	8.5	25	43	1.2	2.1	7.9	20.3	24
	ROL (WORD)	12.5	18	55	117.5	5	7.5	24	33.5	0.6	1.1	4	9.4	18
	ROL (DWORD)	14	20	57.5	127	6	7.5	23.5	35.5	1.3	2.2	8.2	19	18
	ROR (WORD)	8.5	14	51	113	5	7	20.5	33.5	0.6	1.1	4.1	9.4	18
	ROR (DWORD)	10	16	55.5	122	6.5	7.5	21.5	35.5	1.3	2.3	8.2	19	18
	BTST (WORD)	7.5	11	46.5	92	5	7	22.5	35	-	-	-	-	18
	BTST (DWORD)	7	11	43	91	4	6	21.5	34	-	-	-	-	18
	BSET (WORD)	9	11	42.5	89.5	4	5.5	23	32.5	-	-	-	-	15
	BSET (DWORD)	7.5	10.5	40.5	88.5	4	5.5	18.5	31.5	-	-	-	-	15
	BCLR (WORD)	6	10.5	44	89.5	4	5.5	20.5	33.5	-	-	-	-	15
	BCLR (DWORD)	6	10	42	88.5	4	5.5	20	32.5	-	-	-	-	15
	MOVE (BIT)	14	19	58	117.5	3.5	5.5	21	31	0.8	1.2	3.2	6	15
	BPOS (WORD)	10	15	53	108	4	6	22.5	34	0.2	0.4	1.8	4.1	18
	BPOS (DWORD)	13	20.5	81	157.5	4	6	22	35	0.4	0.8	3.1	6.9	18
	SHFR (BIT)	11	20.5	70.5	161.5	4	5.5	12.5	27.5	0.3	0.7	3.1	6.1	24

Note:
1. Time (in microseconds) is based on Release 5.0 of Logicmaster 90-70 software. For information not available at print time for this manual (represented by a dash: –), refer to the IPI for that CPU.
2. For table functions, increment is in units of length specified. For bit operation functions, microseconds/bit. For data move functions, microseconds/the number of bits or words.
3. Enabled time is for single length units of type %R.
4. COMMREQ time has been measured between CPU and EX7 with NOWAIT option.
5. DOIO is the time to output values to discrete output module.

Table A-1. Instruction Timing - Continued

Function Group	Function	Enabled				Disabled				Increment				Size
		924/ 925	914/ 915	781/782 788/789	731/732 771/772	924/ 925	914/ 915	781/782 788/789	731/732 771/772	924/ 925	914/ 915	781/782 788/789	731/732 771/772	
Data Move	MOVE (WORD)	4.5	5.5	14.5	29.5	3.5	3.5	11.5	18.5	0.05	0.1	0.4	0.8	15
	MOVE (DWORD)	7.5	9.5	32.5	56.5	5.5	6.5	18	31	0.2	0.2	1	1.7	15
	BLKMOV (WORD)	5.5	8	21	36	3	3.5	7.5	16.5	-	-	-	-	30
	BLKMOV (DWORD)	7	10	28.5	55	6.5	9	28.5	50.5	-	-	-	-	44
	SWAP (WORD)	5.5	7.5	25	45.5	3	3.5	12	17.5	0.3	0.6	1.7	3.5	15
	SWAP (DWORD)	7	10.5	36.5	65	5	6.5	20.5	31	0.5	0.9	2.7	5.6	15
	BLKCLR	4.5	6	19	30.5	3	3.5	10.5	16.5	0.1	0.2	0.6	1.4	12
	BITSEQ	10	15	46.5	95	7	11.5	38.5	80.5	-	-	-	-	24
	SHFR (WORD)	11	15.5	54	113.5	4	5	11.5	28.5	0.05	0.05	0.3	0.6	24
	SHFR (DWORD)	11.5	16.5	54	117	5	6	14.5	30.5	0.1	0.1	0.8	1.3	24
	SORT	258	423	1403.5	2894	5	6	18.5	28	2.02	3.61	14.43	36.92	15
Data Table	TBLRD (INT)	5.5	8.5	27.5	53.5	3.5	4	11	17.5	-	-	-	-	21
	TBLRD (DINT)	7.5	11.5	28.5	58.5	4	4	12.5	18	-	-	-	-	21
	TBLWR (INT)	6	9	30	60.5	3.5	3.5	11.5	17.5	-	-	-	-	21
	TBLWR (DINT)	8.5	14	52	104.5	6	8.5	19.5	37.5	-	-	-	-	21
	FIFORD (INT)	5.5	9	29	58.5	2.5	3	9.5	17	-	-	-	-	21
	FIFORD (DINT)	7	11	31.5	61	3.5	4	12	18	-	-	-	-	21
	FIFOWRT (INT)	5.5	8	25.5	55	3	4	11.5	17.5	-	-	-	-	21
	FIFOWRT (DINT	7	13	45	85	5	7.5	21.5	37.5	-	-	-	-	21
	LIFORD (INT)	6	9	27.5	52	2.5	3.5	11.5	17	-	-	-	-	21
	LIFORD (DINT)	7.5	12	29.5	57	3.5	4	11.5	18	-	-	-	-	21
	LIFOWRT (INT)	5.5	8	30	55	2.5	3.5	11.5	17	-	-	-	-	21
	LIFOWRT (DINT)	8	14	44.5	85	6	9	23.5	37.5	-	-	-	-	21
	ARRAY_MOVE (BIT)	18.5	27.5	91	197.5	5.5	9	23	41.5	0.01	0.01	0.10	0.30	–
	ARRAY_MOVE (BYTE)	12	20.5	74	163	6	9.5	23	42.5	0.10	0.21	0.50	1.20	–
	ARRAY_MOVE (WORD)	12	20.5	76	164	5.5	9.5	23	41.5	0.03	0.04	0.10	0.30	–
	ARRAY_MOVE (DWORD)	12	20.5	76	165	6.5	9.5	25.5	43	0.09	0.09	0.30	0.60	–
	SRCH (BYTE)	9.5	15	56	119	5.5	8.5	25	40.5	0.03	0.07	0.30	0.60	–
	SRCH (WORD)	9.5	15	57	122.5	5.5	8.5	25.5	41	0.04	0.08	0.30	0.60	–
	SRCH (DWORD)	10	16	60	132	6	8.5	24	41	0.12	0.25	0.80	1.50	–
	ARRAY_RANGE (WORD)	12	16.5	60.5	124	5.5	8	22.5	38	0.24	0.46	2	4.80	–
	ARRAY_RANGE (DWORD)	11.5	16.5	57.5	127.5	7.5	8	20	38	0.25	0.50	2.10	4.80	–
Conversion	to INT (UINT)	4.5	6.5	21.5	32	3	3.5	10.5	16.5	-	-	-	-	12
	to INT (DINT)	5.5	7	23	40	4.5	5.5	16	26	-	-	-	-	12
	to INT (BCD-4)	5	7	20	38.5	3.5	3.5	9	17	-	-	-	-	12
	to DINT (INT)	6.5	7.5	21	33	4.5	5	11	17.5	-	-	-	-	12
	to DINT (UINT)	5.5	7	23.5	32.5	4	4.5	11	17.5	-	-	-	-	12
	to DINT (BCD-8)	11.5	13	44.5	76.5	5	6	19.5	26.5	-	-	-	-	12
	to UINT (INT)	4.5	6	16.5	32	4	5	7.5	17	-	-	-	-	12
	to UINT (DINT)	5.5	6.5	19.5	39.5	5	5.5	15	26	-	-	-	-	12
	to UINT (BCD-4)	5	7.5	20	38.5	3.5	3.5	7.5	17	-	-	-	-	12
	to BCD-4 (INT)	7	9.5	29	44	3.5	3.5	16	16.5	-	-	-	-	12
	to BCD-4 (UINT)	5.5	7.5	25	43	3	3.5	12	16.5	-	-	-	-	12
	to BCD-8 (DINT)	8.5	13	38.5	77	5	6	18.5	26.5	-	-	-	-	12

Note:
1. Time (in microseconds) is based on Release 5.0 of Logicmaster 90-70 software. For information not available when this manual was being printed (represented by a dash: –), refer to the IPI for each CPU.
2. For table functions, increment is in units of length specified. For bit operation functions, microseconds/bit. For data move functions, microseconds/the number of bits or words.
3. Enabled time is for single length units of type %R.
4. COMMREQ time has been measured between CPU and EX7 with NOWAIT option.
5. DOIO is the time to output values to discrete output module.

Table A-1. Instruction Timing - Continued

Function Group	Function	Enabled				Disabled				Increment				Size
		924/ 925	914/ 915	781/782 788/789	731/732 771/772	924/ 925	914/ 915	781/782 788/789	731/732 771/772	924/ 925	914/ 915	781/782 788/789	731/732 771/772	
Control	JUMP	2.5	2.5	6.5	13	2	2	2.5	5	-	-	-	-	–
	FOR/NEXT	13.5	19.5	57	124	5.5	8.5	22	52.5	-	-	-	-	–
	MCR/ENDMCR Combined	7	11	28	51	6	9.5	28	46.5	-	-	-	-	9
	DOIO	25.5	40	122	243.5	3.5	4.5	11	20.5	-	-	-	-	15
	DOIO with ALT	32.5	47	140.5	311.5	4	5	11.5	20.5	-	-	-	-	15
	SUSIO	3.5	4.5	12	22	3	4	7	15.5	-	-	-	-	6
	COMMREQ	113	137	350	790	9	11	22.5	38.5	-	-	-	-	18
	CALL/RETURN (LD)	23	31	90.5	229.5	2.5	2.5	5	6	-	-	-	-	15
	CALL/RETURN (SFC)	104.5	143	390	731.5	2.5	2.5	5	6	-	-	-	-	15
	CALL/RETURN (PSB)	16.5	23.5	73.5	145	3	3	3.5	7	-	-	-	-	15
	CALL/RETURN (External Block)	42.5	74	262.5	309.5	2.5	2.5	3	6	-	-	-	-	15
	PIDISA	6.5	11	40	75.5	6	9.5	33.5	65	-	-	-	-	27
	PIDIND	6	11	41.5	76.5	6.5	10	31	66.5	-	-	-	-	27
	VMERD (BYTE)	16	27.5	89.5	173.5	4.5	7.5	20.5	35	0.8	0.8	0.9	1.2	18
	VMERD (WORD)	16	27.5	89.5	171.5	4.5	7.5	19	34	0.8	0.8	0.9	1.2	18
	VMEWRT (BYTE)	19	30	102.5	192.5	5.5	8	20	36.5	0.8	0.8	0.8	0.9	18
	VMEWRT (WORD)	19	30	97.5	188	4.5	7.5	19.5	34	0.8	0.8	0.8	0.9	18
	VMERMW	21	33.5	99.5	191.5	4.5	7.5	20.5	35	-	-	-	-	18
	VMETST	21	28.5	86	155.5	6.5	9.5	25.5	39	-	-	-	-	15
	VME_CFG_RD	31.5	50.5	170.5	345.5	5.5	9.5	26.5	45	-	-	-	-	–
	VME_CFG_WRT	26.5	43	159.5	333.5	5.5	8.5	28	45	-	-	-	-	–
	SVCREQ:													12
	#1	12	23	79	97	4.5	6	14	26	-	-	-	-	12
	#2	12	21	77	86	4.5	6	14	26	-	-	-	-	12
	#3	12	22	76	85	4.5	6	14	26	-	-	-	-	12
	#4	11	21	74	85	4.5	6	14	26	-	-	-	-	12
	#6	13	22	82	93	4.5	6	14	26	-	-	-	-	12
	#7	24	41	152	288	4.5	6	14	26	-	-	-	-	12
	#8	20	37	143	279	4.5	6	14	26	-	-	-	-	12
	#9	23	36	130	202	4.5	6	14	26	-	-	-	-	12
	#10	15	25	90	127	4.5	6	14	26	-	-	-	-	12
	#11	12	22	82	117	4.5	6	14	26	-	-	-	-	12
	#12	11	20	77	82	4.5	6	14	26	-	-	-	-	12
	#13	-	-	-	-	-	-	-	-	-	-	-	-	12
	#14	150	203	643	963	4.5	6	14	26	-	-	-	-	12
	#15	15	25	91	156	4.5	6	14	26	-	-	-	-	12
	#16	19	31	115	148	4.5	6	14	26	-	-	-	-	12
	#17	23	40	94	346	4.5	6	14	26	-	-	-	-	12
	#18	2939	2998	3199	1203	4.5	6	14	26	-	-	-	-	12
	#19	34	52	77	92	4.5	6	14	26	-	-	-	-	12
	#20	26	42	128	248	4.5	6	14	26	-	-	-	-	12
	#21	138	178	586	1558	4.5	6	14	26	-	-	-	-	12
	#22	11	21	77	85	4.5	6	14	26	-	-	-	-	12

Note:
1. Time (in microseconds) is based on Release 5.0 of Logicmaster 90-70 software. For information not available when this manual was being printed (represented by a dash: –), refer to the IPI for each CPU.
2. For table functions, increment is in units of length specified. For bit operation functions, microseconds/bit. For data move functions, microseconds/the number of bits or words.
3. Enabled time is for single length units of type %R.
4. COMMREQ time has been measured between CPU and EX7 with NOWAIT option.
5. DOIO is the time to output values to discrete output module.

Table A-1. Instruction Timing - Continued

Function Group	Function	Enabled				Disabled				Increment				Size
		924/ 925	914 915	781/782 788/789	731/732 771/772	924/ 925	914/ 915	781/782 788/789	731/732 771/772	924/ 925	914/ 915	781/782 788/789	731/732 771/772	
Floating Point	**Math**:													
	ADD_REAL	8.5	13.5	52.5	108	5.5	6.5	19	31	-	-	-	-	–
	SUB_REAL	8.5	13	51.5	108	5.5	6.5	22.5	31	-	-	-	-	–
	MUL_REAL	8.5	13	51	108	5.5	6.5	21	31	-	-	-	-	–
	DIV_REAL	9.5	14	54	109	5.5	7	22	31	-	-	-	-	–
	ABS_REAL	7	11	42.5	85	5	6.5	17	26.5	-	-	-	-	–
	SQRT_REAL	8	13	43.5	90.5	6	7	14	26.5	-	-	-	-	–
	Trigonometric:													
	SIN	10.5	20.5	63.5	126.5	3.5	6	16	25.5	-	-	-	-	–
	COS	10	20	61.5	121.5	2.5	5.5	14	25.5	-	-	-	-	–
	TAN	9	18	58.5	114.5	3	5.5	14	25	-	-	-	-	–
	ASIN	6.5	13.5	59	132	2.5	5.5	12.5	25.5	-	-	-	-	–
	ACOS	8.5	17.5	69.5	165	2.5	5.5	14.5	25	-	-	-	-	–
	ATAN	10.5	19.5	54	110.5	3.5	5.5	16	25	-	-	-	-	–
	Logarithmic:													
	LOG	6	13	48	100.5	2.5	5.5	13.5	25.5	-	-	-	-	–
	LN	7	13.5	47.5	99.5	2.5	5.5	15.5	25	-	-	-	-	–
	EXPT	10.5	21.5	92	228	3	6	17.5	30	-	-	-	-	–
	EXP	11	22.5	90.5	194	2.5	5.5	14	25.5	-	-	-	-	–
	Comparison:													
	EQ_REAL	7.5	11.5	40	87	5	6	17.5	31	-	-	-	-	–
	NE_REAL	7	10.5	41	85	5	6	20	31	-	-	-	-	–
	GT_REAL	7.5	11.5	46	92	5	6	15	31	-	-	-	-	–
	GE_REAL	7.5	11	46	92	5	6	13.5	30	-	-	-	-	–
	LT_REAL	8	11	43.5	90	5.5	8	19.5	31	-	-	-	-	–
	LE_REAL	7.5	11	45	90	5	6	21.5	31	-	-	-	-	–
	CMP_REAL	17	20	51	114	13	16.5	35.5	64.5	-	-	-	-	–
	Data Move:													
	MOVE_REAL	6	9	29	55.5	3.5	6	14	30	-	-	-	-	–
	Conversion:													
	REAL_TO_INT	7	12.5	43	96.5	4	6	17	26	-	-	-	-	–
	REAL_TO_UINT	8.5	13	46.5	112.5	4	6	16	25.5	-	-	-	-	–
	REAL_TO_DINT	8	13.5	41.5	94	5	7	14.5	25.5	-	-	-	-	-
	INT_TO_REAL	6	10	36.5	68	4	6	14.5	25	-	-	-	-	–
	UINT_TO_REAL	7.5	10.5	30	70.5	4	6	11.5	26	-	-	-	-	-
	DINT_TO_REAL	5.5	9.5	28	65.5	5	5.5	14	25.5	-	-	-	-	–
	REAL_TRUN_INT	8.5	13.5	50	126.5	5	7	15.5	26	-	-	-	-	–
	REAL_TRUN_DINT	8	13.5	51.5	121.5	4	6.5	14.5	26	-	-	-	-	–
	DEG_TO_RAD	9	14.5	43.5	101	5.5	7	12	26	-	-	-	-	–
	RAD_TO_DEG	7.5	14	46	105	4	6	12	26	-	-	-	-	–
	BCD4_TO_REAL	7.5	13	36	92	4	5.5	15	25.5	-	-	-	-	–
	BCD8_TO_REAL	9.5	15	39.5	100	4	6	13.5	26	-	-	-	-	–

Note:
1. Time (in microseconds) is based on Release 5.0 of Logicmaster 90-70 software. For information not available when this manual was being printed (represented by a dash: –), refer to the IPI for each CPU.
2. For table functions, increment is in units of length specified. For bit operation functions, microseconds/bit. For data move functions, microseconds/the number of bits or words.
3. Enabled time is for single length units of type %R.
4. COMMREQ time has been measured between CPU and EX7 with NOWAIT option.
5. DOIO is the time to output values to discrete output module.

Table A-1. Instruction Timing - Continued

Function Group	Function	Enabled			Disabled			Increment			Size
		935	928	CPX782 CPX772	935	928	CPX782 CPX772	935	928	CPX782 CPX772	
Timers	ONDTR	5.5	6	23.5	5	5	16.5	-	-	-	18
	OFDT	5.5	5.5	21	5.5	5.5	20	-	-	-	15
	TMR	6	6.5	23.5	5.5	6	20.5	-	-	-	15
Counters	UPCTR	5.5	5.5	19	5	5	18	-	-	-	18
	DNCTR	5.5	5.5	19	5	5	18	-	-	-	18
Math	ADD (INT)	4	4	10.5	3.5	3.5	7.5	-	-	-	15
	ADD (DINT)	5.5	6	20	4.5	5	11.5	-	-	-	15
	SUB (INT)	4	4	10.5	3.5	3.5	7.5	-	-	-	15
	SUB (DINT)	5.5	6	20	4.5	4.5	11	-	-	-	15
	MUL (INT)	4	4	11	3.5	3.5	8	-	-	-	15
	MUL (DINT)	6	6	21	4.5	4.5	11.5	-	-	-	15
	DIV (INT)	4.5	4.5	11.5	3.5	3.5	8	-	-	-	15
	DIV (DINT)	6.5	6.5	21	5.5	5.5	11.5	-	-	-	15
	MOD (INT)	4.5	4.5	12	3.5	3.5	8	-	-	-	15
	MOD (DINT)	7	7	25	5	5	12	-	-	-	15
	ABS (INT)	4.5	5	17	3.5	4	9.5	-	-	-	12
	ABS (DINT)	5.5	5.5	17	4.5	4.5	10	-	-	-	12
	SQRT (INT)	6.5	6.5	20	3	3	6.5	-	-	-	12
	SQRT (DINT)	9	9	37.5	4.5	4.5	10	-	-	-	12
Relational	EQ (INT)	5.5	5.5	14	3	3.5	7	-	-	-	15
	EQ (DINT)	6	6	19.5	4.5	4.5	11	-	-	-	15
	NE (INT)	5.5	5.5	13	3	3.5	7	-	-	-	15
	NE (DINT)	6	6	19	4.5	4.5	11	-	-	-	15
	GT (INT)	6.5	6.5	16	3	3.5	7	-	-	-	15
	GT (DINT)	7	7	20.5	4.5	4.5	12	-	-	-	15
	GE (INT)	5.5	5.5	15.5	3	3.5	7	-	-	-	15
	GE (DINT)	6	6	19.5	4.5	4.5	11	-	-	-	15
	LT (INT)	6.5	6.5	17	3	3.5	7.5	-	-	-	15
	LT (DINT)	7	7	20.5	4.5	4.5	11.5	-	-	-	15
	LE (INT)	5.5	5.5	15.5	3	3.5	7	-	-	-	15
	LE (DINT)	6	6	20	4.5	4.5	11	-	-	-	15
	CMP (INT)	9.5	9.5	22.5	7.5	8	20.5	-	-	-	21
	CMP (DINT)	10	10.5	27.5	8	8	21	-	-	-	21
	RANGE (INT)	7	7.5	28	4.5	4.5	13	-	-	-	
	RANGE (DINT)	7.5	7.5	30.5	4.5	5	13.5	-	-	-	
Bit Operation	AND (WORD)	6.5	7	32	4.5	4.5	13	0.5	0.5	5.1	18
	AND(DWORD)	7.5	7.5	34	5	5	13.5	0.6	0.7	6	18
	OR (WORD)	6.5	7	32.5	4.5	4.5	13	0.5	0.5	5.4	18
	OR (DWORD)	7.5	8	35.5	5	5	14	0.7	0.8	7.3	18
	XOR (WORD)	6.5	7	32.5	4.5	4.5	13.5	0.5	0.5	5.4	18
	XOR (DWORD)	7.5	7.5	34.5	5	5	13.5	0.7	0.8	7	18
	NOT (WORD)	5.5	5.5	23.5	4	4	12	0.2	0.2	1.9	15
	NOT (DWORD)	6	6	23.5	5	5	12	0.2	0.2	1.6	15
	MCMP (WORD)	9.5	10	58.5	6.5	6.5	18	0.4	0.4	3.3	30
	MCMP (DWORD)	10.5	11	62	7	7	18	0.8	0.9	6.7	30
	SHL (WORD)	9.5	9.5	37.5	4.5	4.5	13.5	0.4	0.5	2.3	24
	SHL (DWORD)	10.5	10.5	40	5.5	5.5	14.5	0.9	1.1	4.8	24
	SHR (WORD)	8.5	8.5	39	4.5	4.5	15	0.4	0.5	2.3	24
	SHR (DWORD)	9.5	9.5	41	5.5	5.5	15.5	0.9	1.1	4.7	24
	ROL (WORD)	7.5	7.5	32	4.5	4.5	12.5	0.4	0.5	2.3	18
	ROL (DWORD)	8	8	34.5	5	5	13	1	1.1	4.8	18
	ROR (WORD)	6.5	6.5	31.5	4.5	4.5	12.5	0.5	0.5	2.5	18
	ROR (DWORD)	8	8	34	5	4.5	14	1	1.1	5.1	18
	BTST (WORD)	6.5	6.5	26	4.5	4.5	12.5	-	-	-	18
	BTST (DWORD)	6.5	6.5	25	4.5	4.5	11.5	-	-	-	18
	BSET (WORD)	5.5	5.5	25	4	4	11	-	-	-	15
	BSET (DWORD)	5.5	5.5	23.5	4.5	4.5	9.5	-	-	-	15
	BCLR (WORD)	5.5	5.5	24.5	4	4	11	-	-	-	15
	BCLR (DWORD)	5.5	5.5	24.5	4	4	11	-	-	-	15
	MOVE (BIT)	11	11	36	4	4	11.5	0.7	0.7	1.8	15
	BPOS (WORD)	6.5	6.5	30	4	4	11	0.1	0.1	1.3	18
	BPOS (DWORD)	8.5	8.5	40.5	4	4	11	0.2	0.2	2.3	18
	SHFR (BIT)	8	8	43	3.5	3.5	8.5	0.2	0.2	1.8	24

Note: 1. Time (in microseconds) is based on Release 5.0 of Logicmaster 90-70 software. For information not available at print time for this manual (represented by a dash; –), refer to the IPI for that CPU.
2. For table functions, increment is in units of length specified. For bit operation functions, microseconds/bit. For data move functions, microsec onds/the number of ts or rds.
3. Enabled time is for single length units of type %R.
4. COMMREQ time has been measured between CPU and EX7 with NOWAIT option.
5. DOIO is the time to output values to discrete output module.

Table A-1. Instruction Timing - Continued

Function Group	Function	Enabled			Disabled			Increment			Size
		935	928	CPX772 CPX782	935	928	CPX772 CPX782	935	928	CPX772 CPX782	
Data Move	MOVE (WORD)	4	4	9.5	3	3	7	0.1	0.1	0.3	15
	MOVE (DWORD)	5.5	5.5	18.5	4.5	4.5	11	0.2	0.2	0.8	15
	BLKMOV (WORD)	4.5	4.5	13	2.5	2.5	5	-	-	-	30
	BLKMOV (DWORD)	6	6	18	5.5	5.5	19	-	-	-	44
	SWAP (WORD)	4.5	4.5	14	2.5	2.5	6.5	0.2	0.2	1.3	15
	SWAP (DWORD)	6	6	20.5	4.5	4.5	11	0.3	0.3	1.5	15
	BLKCLR	4	4	10.5	2.5	2.5	5.5	0.1	0.2	0.3	12
	BITSEQ	9	9	28.5	6	6	24	-	-	-	24
	SHFR (WORD)	7	7	32.5	3.5	3.5	9	0.1	0.1	0.3	24
	SHFR (DWORD)	8	8	34	4.5	4.5	10	0.2	0.2	0.7	24
	SORT	215.5	260	788	4.5	4	13	1.38	1.53	8.8	15
Data Table	TBLRD (INT)	4	4	15.5	2	2	4.5	-	-	-	21
	TBLRD (DINT)	5	5	17	3	3	5.5	-	-	-	21
	TBLWR (INT)	4.5	4.5	17.5	2	2	5	-	-	-	21
	TBLWR (DINT)	6.5	6.5	31.5	4	4	12.5	-	-	-	21
	FIFORD (INT)	4	4	16.5	2	2	5.5	-	-	-	21
	FIFORD (DINT)	5	5	18	3	3	5.5	-	-	-	21
	FIFOWRT (INT)	4	4	15	2	2	5	-	-	-	21
	FIFOWRT (DINT	5.5	5.5	25.5	4	4	13	-	-	-	21
	LIFORD (INT)	4	4	14	2	2	5	-	-	-	21
	LIFORD (DINT)	5	5	17	3	3	6	-	-	-	21
	LIFOWRT (INT)	4	4	15	2	2	5	-	-	-	21
	LIFOWRT (DINT)	5.5	5.5	26	4.5	4.5	13.5	-	-	-	21
Array	ARRAY_MOVE (BIT)	9.5	10	52.5	5	5	14.5	0.01	0.01	0.07	–
	ARRAY_MOVE (BYTE)	8.5	9	44	5	5	15	0.07	0.07	0.40	–
	ARRAY_MOVE (WORD)	8.5	9	45.5	5	5	15.5	0.04	0.06	0.10	–
	ARRAY_MOVE (DWORD)	9.5	9.5	46	5.5	5.5	16	0.09	0.12	0.30	–
	SRCH (BYTE)	8	8	36	5	5	14.5	0.02	0.02	0.10	–
	SRCH (WORD)	8	8	35.5	5	5	14.5	0.02	0.02	0.10	–
	SRCH (DWORD)	8	8.5	38.5	5	5	14.5	0.08	0.08	0.40	–
	ARRAY_RANGE (WORD)	7.5	7.5	38	5	5	14	0.15	0.16	1.30	–
	ARRAY_RANGE (DWORD)	7.5	7.5	37.5	5	5	14	0.16	0.17	1.30	–
Conversion	to INT (UINT)	4	4	11.5	3	3	5.5	-	-	-	12
	to INT (DINT)	5	5	13.5	4	4	9.5	-	-	-	12
	to INT (BCD-4)	4.5	4.5	12	3	3	6	-	-	-	12
	to DINT (INT)	5	5	12	3.5	3.5	5.5	-	-	-	12
	to DINT (UINT)	5	5	11.5	3.5	3.5	6.5	-	-	-	12
	to DINT (BCD-8)	6.5	6.5	19.5	4.5	4	9.5	-	-	-	12
	to UINT (INT)	4	4	10.5	3	3	6.5	-	-	-	12
	to UINT (DINT)	5	5	13.5	4	4	9	-	-	-	12
	to UINT (BCD-4)	4.5	4.5	12.5	3	3	6.5	-	-	-	12
	to BCD-4 (INT)	4.5	4.5	12	3	3	6	-	-	-	12
	to BCD-4 (UINT)	4.5	4.5	13	3	3	6	-	-	-	12
	to BCD-8 (DINT)	7	7	18	4.5	4.5	9.5	-	-	-	12

Note: 1. Time (in microseconds) is based on Release 5.0 of Logicmaster 90-70 software. For information not available when this manual was being printed (represented by a dash: –), refer to the IPI for each CPU.
2. For table functions, increment is in units of length specified. For bit operation functions, microseconds/bit. For data move functions, microseconds/ /the number of bits or words.
3. Enabled time is for single length units of type %R.
4. COMMREQ time has been measured between CPU and EX7 with NOWAIT option.
5. DOIO is the time to output values to discrete output module

Table A-1. Instruction Timing - Continued

Function Group	Function	Enabled			Disabled			Increment			Size
		935	928	CPX772 CPX782	935	928	CPX772 CPX782	935	928	CPX772 CPX782	
Control	JUMP	2.5	2.5	3.5	2	2	3	-	-	-	–
	FOR/NEXT	10.5	11	38	5	5	16.5	-	-	-	–
	MCR/ENDMCR Combined	6	6.5	18.5	5	5.5	17.5	-	-	-	9
	DOIO	13.5	15	81.5	3	3	7.5	-	-	-	15
	DOIO with ALT	14	16	95.5	3	3	7.5	-	-	-	15
	SUSIO	3	3	7.5	2.5	2.5	6	-	-	-	6
	COMMREQ	84.5	90	240.5	8.5	8.5	16	-	-	-	18
	CALL/RETURN (LD)	23	24.5	57.5	2	2	2.5	-	-	-	15
	CALL/RETURN (SFC)	82	88	251	2	2	2.5	-	-	-	15
	CALL/RETURN (PSB)	15.5	17.5	47.5	3	3	3.5	-	-	-	15
	CALL/RETURN (External Block)	27.5	31.5	152.5	2	2	2.5	-	-	-	15
	PIDISA	6	6	22.5	6	6	20.5	-	-	-	27
	PIDIND	6	6	23	6	6	20.5	-	-	-	27
	VMERD (BYTE)	13.5	13.5	53	4	4	13	0.8	0.8	0.9	18
	VMERD (WORD)	13.5	13.5	50.5	4	4	12	0.8	0.8	0.9	18
	VMEWRT (BYTE)	14.5	14.5	55	4	4	13	0.8	0.8	0.8	18
	VMEWRT (WORD)	14.5	14.5	54.5	4	4	12.5	0.8	0.8	0.8	18
	VMERMW	15	15.5	56.5	4	4	13	-	-	-	18
	VMETST	17	17	49	6	6	14.5	-	-	-	15
	VME_CFG_RD	26	26.5	104	4.5	4.5	16.5	-	-	-	–
	VME_CFG_WRT	17.5	18	93	4.5	4.5	16	-	-	-	–
	SVCREQ:										12
	#1	9	9	49	3.5	3.5	10.5	-	-	-	12
	#2	9	9	49	3.5	3.5	10.5	-	-	-	12
	#3	9	9	48	3.5	3.5	10.5	-	-	-	12
	#4	8	9	48	3.5	3.5	10.5	-	-	-	12
	#6	11	11	50	3.5	3.5	10.5	-	-	-	12
	#7	19	21	96	3.5	3.5	10.5	-	-	-	12
	#8	22	25	120	3.5	3.5	10.5	-	-	-	12
	#9	19	19	71	3.5	3.5	10.5	-	-	-	12
	#10	11	11	54	3.5	3.5	10.5	-	-	-	12
	#11	10	10	50	3.5	3.5	10.5	-	-	-	12
	#12	9	9	48	3.5	3.5	10.5	-	-	-	12
	#13	-	-	-	-	-	-	-	-	-	12
	#14	128	159	370	3.5	3.5	10.5	-	-	-	12
	#15	10	10	55	3.5	3.5	10.5	-	-	-	12
	#16	16	17	64	3.5	3.5	10.5	-	-	-	12
	#17	14	15	91	3.5	3.5	10.5	-	-	-	12
	#18	2985	2986	546 (772) 3025 (782)	3.5	3.5	10.5	-	-	-	12
	#19	24	24	112	3.5	3.5	10.5	-	-	-	12
	#20	20	23	99	3.5	3.5	10.5	-	-	-	12
	#21	77	91	381	3.5	3.5	10.5	-	-	-	12
	#22	9	9	47	3.5	3.5	10.5	-	-	-	12

Note 1. Time (in microseconds) is based on Release 5.0 of Logicmaster 90-70 software. For information not available when this manual was being printed (represented by a dash: –), refer to the IPI for each CPU.

2. For table functions, increment is in units of length specified. For bit operation functions, microseconds/bit. For data move functions, microseconds/ /the number of bits or words.
3. Enabled time is for single length units of type %R.
4. COMMREQ time has been measured between CPU and EX7 with NOWAIT option.
5. DOIO is the time to output values to discrete output module.

Table A-1. Instruction Timing - Continued

Function Group	Function	Enabled			Disabled			Increment			Size
		935	928	CPX772 CPX782	935	928	CPX772 CPX782	935	928	CPX772 CPX782	
Floating Point	**Math**:										
	ADD_REAL	7	7	21.5	4.5	4.5	11	-	-	-	–
	SUB_REAL	7	7	21	4.5	4.5	11	-	-	-	–
	MUL_REAL	7	7	21.5	4.5	4.5	11	-	-	-	–
	DIV_REAL	7	7	21.5	4.5	4.5	12	-	-	-	–
	ABS_REAL	6	6	17.5	4.5	4.5	10	-	-	-	–
	SQRT_REAL	6.5	6.5	18	4.5	4.5	10.5	-	-	-	–
	Trigonometric:										
	SIN	8	8	19.5	3	3	9	-	-	-	–
	COS	8	8	20	3	3	9	-	-	-	–
	TAN	7.5	7.5	19.5	3	3	9.5	-	-	-	–
	ASIN	6	6	17.5	3	3	9.5	-	-	-	–
	ACOS	7	7	19	3	3	9.5	-	-	-	–
	ATAN	8	8	20	3	3	9.5	-	-	-	–
	Logarithmic:										
	LOG	5.5	5.5	19	3	3	9	-	-	-	–
	LN	5.5	5.5	18.5	3	3	9	-	-	-	–
	EXPT	8.5	8.5	27	3	3	10.5	-	-	-	–
	EXP	9	9	20	3	3	9	-	-	-	–
	Comparison:										
	EQ_REAL	7	7	21	5	5	12	-	-	-	–
	NE_REAL	7	7	19.5	5	5	11.5	-	-	-	–
	GT_REAL	7	7	20.5	5	5	11.5	-	-	-	–
	GE_REAL	7	7	20.5	5	5	11.5	-	-	-	–
	LT_REAL	7	7	19.5	5	5	11.5	-	-	-	–
	LE_REAL	7	7	19.5	5	5	11	-	-	-	–
	CMP_REAL	16.5	16.5	32	13.5	13.5	25.5	-	-	-	–
	Data Move:										
	MOVE_REAL	4.5	4.5	17.5	3.5	3.5	10	-	-	-	–
	Conversion:										
	REAL_TO_INT	6	6	18	3.5	3.5	9.5	-	-	-	–
	REAL_TO_UINT	6	6	20	3.5	3.5	9.5	-	-	-	–
	REAL_TO_DINT	5.5	5.5	18	3.5	3.5	9.5	-	-	-	-
	INT_TO_REAL	5	5	17	3.5	3.5	9.5	-	-	-	–
	UINT_TO_REAL	5	5	17.5	3.5	3.5	9.5	-	-	-	-
	DINT_TO_REAL	4.5	4.5	16	3.5	3.5	9.5	-	-	-	–
	REAL_TRUN_INT	6	6	20.5	3.5	3.5	10	-	-	-	–
	REAL_TRUN_DINT	6	6	20	3.5	3.5	9	-	-	-	–
	DEG_TO_RAD	5.5	5.5	17.5	3.5	3.5	9.5	-	-	-	–
	RAD_TO_DEG	6	6	18.5	3.5	3.5	9.5	-	-	-	–
	BCD4_TO_REAL	6	6	19.5	3.5	3.5	9.5	-	-	-	–
	BCD8_TO_REAL	6.5	6.5	21.5	3.5	3.5	9	-	-	-	–

Note: 1. Time (in microseconds) is based on Release 5.0 of Logicmaster 90-70 software. For information not available when this manual was being printed (represented by a dash: –), refer to the IPI for each CPU.
2. For table functions, increment is in units of length specified. For bit operation functions, microseconds/bit. For data move functions, microseconds/the number of bits or words.
3. Enabled time is for single length units of type %R.
4. COMMREQ time has been measured between CPU and EX7 with NOWAIT option.
5. DOIO is the time to output values to discrete output module.

Overhead Sweep Impact Time

This part of the appendix contains overhead timing information for the Series 90-70 PLC CPU. This information can be used in conjunction with the estimated logic execution time to predict sweep times for each of the Series 90-70 CPUs. The information in this section is made up of a base sweep time plus sweep impact times for each of the CPU models: 731, 732, 771, 772, 781, 782, 914, 915, 924, 925, 928 and 935. The predicted sweep time is computed by adding the sweep impact time(s), the base sweep, and the estimated logic execution time.

Two examples of predicting sweep times are provided at the end of this appendix.

Sweep impact times are composed of four basic sections:

1. Programmer communications sweep impact
2. I/O Scan and fault sweep impact
3. Ethernet Global Data sweep impact
4. Intelligent Option Module (PCMs and LAN modules) sweep impact
5. I/O interrupt performance and sweep impact

Each of these sections describes the functions and provides tables with the corresponding times for each CPU model.

The information in these tables may be used to predict sweep time based on a given configuration.

What the Tables Contain

The following tables contain sweep impact times for overhead functions for the Series 90-70 PLC. Base sweep time is the time for an empty _MAIN program block to execute, with no configuration stored and none of the windows active. The rest of the timing values are given as sweep impact times, that is, the time added to the sweep by the function in question. Sweep impact times are nominal.

Note

There are two categories of sweep impact numbers listed in the tables—those that impact the sweep every sweep and those that impact the sweep only when invoked. The functions that impact the sweep every sweep are listed in bold type in each table.

In some of the tables, functions are shown as asynchronously impacting the sweep. This means that there is not a set phase of the sweep in which the function takes place. For instance, the scanning of all I/O modules takes place during either the input or output scan phase of the PLC CPU's sweep. However, I/O interrupts are totally asynchronous to the sweep and will interrupt any function currently in progress.

The communication functions (with the exception of the high priority programmer requests) are all processed within one of the two windows in the sweep (the Programmer Communications Window and the System Communications Window). Sweep impact times for the various service requests are all minimum sweep impact times for the defined functions, where the window times have been adjusted so that no timeslicing (limiting) of the window occurs in a given PLC sweep. This means that, as much as possible, each function is completed in one occurrence of the window (between consecutive logic scans). The sweep impact of these functions can be spread out over multiple sweeps (limited) by adjusting the window times to a value lower than the documented sweep impact time. For the programmer, the default time is 10 milliseconds; therefore, some of the functions listed in that section will naturally timeslice over successive sweeps.

Base Sweep Times

The base sweep time for each CPU model is shown below. This time is for an empty _MAIN program block with no programmer attached, no configuration downloaded, and no other module present in the system other than the CPU. The following diagram shows the full sweep phases and the base sweep phases contrasted so that the optional parts of the sweep are illustrated.

Table A-2. Base Sweep vs. Full Sweep Phases

BASE SWEEP	FULL SWEEP
<START OF SWEEP>	<START OF SWEEP>
Sweep Housekeeping	Sweep Housekeeping
↓	↓
NULL Input Scan *	Input Scan *
↓	↓
Program Logic Execution	EGD Consumption Scan***
↓	↓
NULL Output Scan *	Program Logic Execution
↓	↓
↓	Output Scan *
↓	↓
↓	EGD Production Scans ***
↓	↓
↓	Poll for Missing I/O Modules **
↓	↓
↓	Programmer Communications Window
↓	↓
↓	System Communications Window
<END OF SWEEP>	<END OF SWEEP>

* If I/O is suspended, then the input and output scans are skipped.
** Polling for missing I/O modules only occurs if a “Loss of ...” fault has been logged for a Series 90-70 I/O module.
*** If no Ethernet Global Data (EGD) exchanges are configured, then the consumption and production scans are skipped.

For the base sweep, the lack of configuration means that the input and output scan phases of the sweep are NULL (i.e., check for configuration and then end). The presence of a configuration with no I/O modules or intelligent I/O modules (GBC, PSM, etc.) would have the same effect.

The logic execution time is not zero in the base sweep. The time to execute the empty _MAIN program is included so that you only need to add the estimated execution times of the functions actually programmed. The base sweep also assumes no missing I/O modules. The lack of programmer attachment means that the Programmer Communications Window is never opened. The lack of intelligent option modules means that the System Communications Window is never opened.

The following table gives the base sweep times in milliseconds for each CPU model.

Table A-3. Base Sweep Times

CPU Model	CPX 935	CPX 928	924/ 925	914/ 915	CPX 782	781/782 788/789	CPX 772	731/732 771/772
Base Sweep Time	.3731	.4001	.1525	.2074	1.0687	.7025	1.0687	2.00

Programmer Sweep Impact Times

The following table shows the programmer sweep impact times in milliseconds. The times are broken up into parallel and serial since these interfaces are significantly different. (Each item in the table is described in more detail at the end of the table.)

Table A-4. Programmer Sweep Impact Times*

Sweep Impact Item	CPX 935	CPX 928	914/ 925	924/ 925	CPX 782	781/782 788/789	CPX 772	731/731 771/772
Parallel Programmer								
Programmer window								0.75
Reference table monitor								4.90
Editor monitor								4.90
Word-for-word change								28.10
ALT-S store								14.80
High priority request								9.90
Serial programmer								0.50
Programmer Window								5.60
Reference table monitor								6.40
Editor monitor								

* Not all of the timing information needed for the above table was available at print time for this manual (the blank spaces).

Note

Functions in bold type in Table A-4 above impact the sweep continuously. All other functions impact the sweep only when invoked.

Each of the items included in the table is described below.

Sweep Impact Item	Description
Programmer Window	**Parallel:** The time to open the Programmer Window but not process any requests. The programmer is attached with a parallel connection; no reference values are being monitored. **Serial:** Same as above except attached serially. Unlike the parallel programmer, the sweep impact cannot be limited to a single sweep. Serial communications, in addition to the service processing time that is performed inside the Programmer Communications Window, has processing time associated with the receiving of data over the RS422/485 serial link. This serial communications processing time is asynchronous to the sweep in order to meet the timing requirements of the serial protocol.
Reference Table Monitor	The sweep impact to refresh the reference table screen. (The %R table was used as the example.) Mixed table display impacts are slightly larger. The sweep impact may not be continuous, depending on the sweep time of the PLC and the speed of the host of the programming software.
Editor Monitor	The sweep impact to refresh the editor screen when monitoring ladder logic. The times given in the table are for a logic screen containing one contact, two coils, and eleven registers. As with the reference table sweep impact, the impact may not be continuous.
Word-for-Word Change	The sweep impact to change a constant input on a MOVE_UINT function from 1 to 2. This is the smallest change that can be made. A change to a coil requires updates to the coil use and retentive maps contained in the PLC. If the %Q or %M reference address is changed on a function block, then the size of the change to the coil use and retentive maps can be quite large. A large word-for-word change will have very little sweep impact if done with the parallel programmer (worst-case of 35 ms on a 731 CPU). A large word-for-word change will have a big sweep impact if done with the serial programmer. The time is linear for the number of bytes in the request with the worst case being slightly larger than the worst case for parallel.
ALT-S Store	The sweep impact to store a 933 byte program block. The sweep impact is linear as the program block gets larger.
High Priority Request (Parallel only)	There are several programmer high priority requests: change PLC mode (includes ALT-R), read constant sweep state/time, read window times, change constant sweep state/time, and change window times. Unlike the other programmer service requests, which are serviced inside the programmer communications window, high priority requests interrupt the sweep and are serviced asynchronously at the time they are issued by the programmer. The worst case high priority request is monitoring/changing the window times. This time is shown below.

I/O Scan and I/O Fault Sweep Impact

The I/O scan sweep impact has two parts, Series 90-70 I/O and Genius I/O. The equation for computing I/O scan sweep impact is:

I/O Scan Sweep Impact = I/O Scan Overhead + Series 90*70 I/O Scan + Genius I/O Scan

The following table shows the I/O scan overhead in milliseconds for each CPU:

Table A-5. I/O Scan Overhead *

CPU Model	CPX 935	CPX 928	924/ 925	914/ 915	CPX 782	781/782 788/789	CPX 772	731/732 771/772
I/O scan overhead	.0293	.0321	.1348	.1382	.0638	.1756	.0638	0.38

* Times are in milliseconds. For information not available when this manual was being printed (represented by a dash: –), refer to the IPI for each CPU.

Note

I/O scan overhead impacts the sweep continuously.

Sweep Impact of Series 90-70 I/O Modules

The I/O scan of the Series 90-70 I/O modules is impacted as much by location and reference address of a module as it is by the number of modules. The I/O scan has several basic parts.

I/O Scan	Description
I/O Scan Overhead	Includes the setup for input and output scan and the selection of the main rack.
Rack Setup Time	Each expansion rack is selected separately because of the addressing of expansion racks on the VME bus. This results in a fixed overhead per expansion rack, regardless of the number of modules in that rack.
Per Module Setup Time	Each Series 90-70 I/O module has a fixed setup scan time.
Byte Transfer Time	The actual transfer of bytes is much faster for modules located in the main rack than for those in expansion racks. The byte transfer time differences will be accounted for by using different times for I/O modules in the main rack versus expansion racks.

In addition, analog input expander modules (the same as Genius blocks) have the ability to be grouped into a single transfer as long as consecutive reference addresses are used for modules that have consecutive slot addresses. Each sequence of consecutively addressed modules is called a scan segment. There is a time penalty for each additional scan segment.

The following form can be used for computing I/O module sweep impact. The calculation contains times for analog input expanders that are either grouped into the same scan segment as the preceding module or are grouped in a separate new scan segment. The sweep impact times can be found in table A-7.

Table A-6. Worksheet A: I/O Module Sweep Time

Number of expansion racks		_____	
Sweep impact per expansion rack	x	_____	= _____
Number of discrete I/O modules—main rack		_____	
Sweep impact per discrete I/O module—main rack	x	_____	= _____
Number of discrete I/O modules—expansion rack		_____	
Seep impact per discrete I/O module—expansion rack	x	_____	= _____
Number of analog input base and output modules—main rack		_____	
Sweep impact per analog input base and output module—main rack	x	_____	= _____
Number of analog input exander modules (same segment)—main rack		_____	
Sweep impact per analog input expander module (same segment)—main rack	x	_____	= _____
Number of analog input exander modules (new segment)—main rack		_____	
Sweep impact per analog input expander module (new segment)—main rack	x	_____	= _____
Number of analog input base and output modules—expansion rack		_____	
Sweep impact per analog input base and output module—expansion rack	x	_____	= _____
Number of analog input base and output modules (same segment)—exp. rack		_____	
Sweep impact per analog input base and output module (same seg.)—exp. rack	x	_____	= _____
Number of analog input base and output modules (new segment)—exp. rack		_____	
Sweep impact per analog input base and output module (new seg.)—exp. rack	x	_____	= _____
Predicted Series 90-70 I/O Module Sweep Impact			_____

Note

If point faults are enabled, substitute the corresponding times for point faults enabled, as shown in the following table.

An approximate per point or per channel average is shown in the following tables. These averages are based on 1024 points (512 in and 512 out) for discrete and 128 channels (96 in and 32 out) for analog. The 96 analog input channels consist of two base modules and five expanders. Actual values will vary from the approximate average, depending on the system I/O configuration.

Note

Not all of the sweep time information was available at the time this manual was printed (the blank spaces in the table below). Refer to the IPI for the specific CPU for this information.

Table A-7. Sweep Impact Time for Model 70 I/O Modules and Racks *

	CPX 935	CPX 928	924/ 925	914/ 915	CPX 782	781/782 788/789	CPX 772	731/732 771/772
Rack Setup								
per expansion rack	.0010	.0010	.0010	.0010	.0010	.0010	.0010	.0500
Discrete I/O Modules								
per I/O module in main rack	.0074	.0086	.0209	.0105	.0144	.0090	.0144	.0800
per I/O module in main rack w/point faults enabled	.0084	.0125	.0136	.0156	.0220	.0283	.0220	.1200
per I/O module in expansion rack	.0119	.0144	.0226	.0110	.0154	.0102	.0154	.0900
per I/O module in expansion rack w/point faults enabled	.0135	.0184	.0398	.0183	.0273	.0149	.0273	.1300
per fault message **	.1314	.1781	.4256	.1570	.3980	.1189	.3980	1.700
Rough Average per Point (no point faults)								3.1µs
Rough Average per Point (w/ point faults)								4.4µs
Analog I/O Modules								
per input/output module in main rack	.0203	.0230	.0279	.0160	.0501	.0149	.0501	0.08
per input/output module w/point faults enabled in main rack	.0259	.0281	.0308	.0223	.0579	.0207	.0579	0.12
per input or output module in expansion rack	.0313	.0359	.0321	.0318	.0472	.0480	.0472	0.10
per input/output module w/point faults enabled in exp. rack	.0451	.0499	.0464	.0523	.0688	.0761	.0688	0.15
per input expander module in same segment in main rack	.0166	.0183	.0152	.0169	.0187	.0188	.0187	0.02
per input expander module w/point faults enabled in same segment in main rack	.0249	.0271	.0242	.0259	.0279	.0275	.0279	0.03
per input expander module in new segment in main rack	.0189	.0199	.0203	.0227	.0267	.0366	.0267	0.03
per input expander module w/point faults enabled in new segment in main rack	.0272	.0308	.0279	.0324	.0395	.0514	.0395	0.05
per input expander module in same segment in expansion rack	.0537 .0796	.0534 .0822	.0558 .0824	.0567 .0836	.0277 .0558	.0570 .0856	.0277 .0558	0.06 0.10
per input expander module w/point faults enabled in same segment in expansion rack	.0557	.0566	.0598	.0653	.0356	.0705	.0356	0.10
per input expander module in new segment in expansion rack	.0851	.0861	.0883	.0934	.0675	.1047	.0675	0.15
per input expander module w/point faults enabled in new segment in expansion rack	.1613	.2100	.2597	.3296	.5362	.9362	.5362	2.10
per fault message **								10.8µs
Rough Average per Channel (no point faults)								14.0µs
Rough Average per Channel (w/ point faults)								

* Times are in milliseconds, except for those identified as microseconds.

** Faults for discrete Series 90-70 I/O modules are always polled for by the PLC CPU. When one occurs, it is always logged during one of the I/O scan phase of the sweep. These faults are only polled when point faults are enabled.

Note

Functions in bold type in Table A-7 above impact the sweep continuously. All other functions impact the sweep only when invoked. Also, not all of the timing information needed for the above table was available at print time for this manual (the blank spaces).

Sweep Impact of Genius I/O and GBCs

The sweep impact of Genius I/O and Genius Bus Controllers (GBC) is similar to that of Series 90-70 I/O. There is an overhead for the I/O scan that should be counted only once between the Series 90-70 I/O scan and the Genius I/O scan. There is also a per Genius Bus Controller sweep impact, a per scan segment sweep impact, and a transfer time (per word) sweep impact for all I/O data.

The sweep impact per Genius Bus Controller has three parts:

1. Sweep impact to open the System Communications Window. This is added only once when the first intelligent option module (of which the Genius Bus Controller is one) is placed in the system.
2. Sweep impact to poll each Genius Bus Controller for background messages (datagrams). This part is an impact for every Genius Bus Controller in the system.

Note

Both the first and second parts of the Genius Bus Controller's sweep impact may be eliminated by closing the System Communications Window (setting its time to 0). This should only be done to reduce scan time during critical phases of a process to ensure minimal scan time. Incoming messages will timeout, and COMMREQs will stop working while the window is closed. Communications with PCM and LAN modules will also stop.

3. Sweep impact to scan the Genius Bus Controller. This impact results from the PLC CPU notifying the Genius Bus Controller that its new output data has been transferred and commanding the Genius Bus Controller to ready its input data, as well as informing the Genius Bus Controller that the PLC has finished another sweep and is still in RUN mode.

A scan segment for Genius I/O consists of Genius blocks on the same bus with consecutive reference addresses and consecutive bus addresses. The time to process a single scan segment is higher for an input scan segment than it is for an output scan segment. The scan segment processing is the same for analog, discrete, and global data scan segments. Discrete data is transferred a byte at a time and takes longer to complete the transfer than analog data, which is transferred a word at a time. Global data should be counted as either discrete or analog, based on the memory references used in the source or destination.

Note

Not all of the sweep time information was available at the time this manual was printed (the blank spaces in the table below). Refer to the IPI for the specific CPU for this information.

Table A-8. Sweep Impact Time of Genius I/O and GBCs *

	CPX 935	CPX 928	924/ 925	914/ 915	CPX 782	781/782 788/789	CPX 772	731/732 771/772
Genius Bus Controller								
open system communications window	.0583	.0631	.0432	.0616	.1625	.1890	.1625	.6000
per Genius Bus Controller polling for background messages	.0072	.0083	.0081	.0086	.0160	.0200	.0160	.1000
per Genius Bus Controller I/O Scan	.5658	.5619	.4237	.6476	.5215	.8110	.5215	.7000
First Genius Bus Controller **								2.4000
Subsequent Genius Bus Controllers								1.6000
Genius I/O Blocks								
per I/O block scan segment	.0070	.0102	.0270	.0300	.0314	.0444	.0314	.0300
per I/O block scan segment w/point faults enabled	.0259	.0266	.0540	.0600	.0584	.0725	.0584	.0900
per byte discrete I/O data in the main rack	.0451	.0460	.0015	.0017	.0475	.0031	.0475	.0032
per byte discrete I/O data in expansion racks	.0468	.0471	.0020	.0025	.0442	.0045	.0442	.0045
per word analog I/O data in the main rack	.0634	.0643	.0011	.0011	.0604	.0042	.0604	.0029
per word analog I/O data in expansion racks	.0654	.0651	.0040	.0057	.0623	.0131	.0623	.0077
Asynchronous Events								
Fault Message								2.00

* Times are in milliseconds, except for those identified as microseconds.

** The extra time for the first GBC is the same time as shown in the next table for the first intelligent option module. This time should be counted only once.

Note

Functions in bold type in Table A-8 above impact the sweep continuously. All other functions impact the sweep only when invoked.

Note

Not all of the timing information needed for the above table was available at print time for this manual (the blank spaces).

Use the following form for predicting the sweep impact due to Genius I/O. The sweep impact times can be found in table A-8.

Table A-9. Worksheet B: Genius I/O Sweep Time

Open system communications window		_____	=	_____
GBC I/O scan		_____		
GBC poll for background messages	+	_____	=	_____
Number of GBCs	x	_____	=	_____
Input block scan segments—number of		_____		
Input block scan segments—sweep impact	x	_____	=	_____
Output block scan segments—number of		_____		
Output block scan segments—sweep impact	x	_____	=	_____
Bytes of discrete I/O data on GBCs—main rack		_____		
Sweep impact/bytes of discrete I/O data—main rack	x	_____	=	_____
Bytes of discrete I/O data on GBCs—expansion racks		_____		
Seep impact/bytes of discrete I/O data—expansion racks	x	_____	=	_____
Words of analog I/O data on GBCs—main rack		_____		
Sweep impact/word analog I/O data—main rack	x	_____	=	_____
Words of analog I/O data on GBCs—expansion racks		_____		
Sweep impact/word analog I/O data—expansion racks	x	_____	=	_____
		Predicted Genius I/O Scan Impact		_____

Sweep Impact of FIP I/O and FBCs

The sweep impact of FIP I/O and FIP Bus Controllers (FBC) is similar to that of Series 90-70 I/O. There is an overhead for the I/O scan that should be counted only once between the Series 90-70 I/O scan and the FBC I/O scan. There is also a per FIP Bus Controller sweep impact, a per scan segment sweep impact; and a transfer time (per word) sweep impact for all I/O data.

The sweep impact per FIP Bus Controller has three parts:

1. Sweep impact to open the System Communications Window. This is added only once when the first intelligent option module (of which the FIP Bus Controller is one) is placed in the system.
2. Sweep impact to poll each FIP Bus Controller for background messages (datagrams). This part is an impact for every FIP Bus Controller in the system.

Note

Both the first and second parts of the FIP Bus Controller's sweep impact may be eliminated by closing the System Communications Window (setting its time to 0). This should only be done to reduce scan time during critical phases of a process to ensure minimal scan time. Incoming messages will timeout, and COMMREQs will stop working while the window is closed. Communications with PCM and LAN modules will also stop.

3. Sweep impact to scan the FIP Bus Controller. This impact results from the PLC CPU notifying the FIP Bus Controller that its new output data has been transferred and commanding the FIP Bus Controller to ready its input data, as well as informing the FIP Bus Controller that the PLC has finished another sweep and is still in RUN mode.

A scan segment for FIP I/O consists of FIP Transfer Variables (TVAs) on the same bus with consecutive reference addresses containing up to 256 bytes. The scan segment processing is the same for analog and discrete scan segments.

Note

Some of the sweep time information was not available at the time this manual was printed (the blank spaces in the table below).

Table A-10. Sweep Impact Time of FIP I/O and FBCs *

	CPX 935 **	CPX 928 **	924/ 925**	914/ 915**	CPX 782**	781/ 782 **	CPX 772**
FIP Bus Controller							
open system communications window	.0583	.0631	.0432	.0616	.1625	.1890	.1625
per FIP Bus Controller polling for background messages	.0072	.0083	.0081	.0086	.0160	.0200	.0160
per FIP Bus Controller I/O Scan	.0839	.1147	___	___	.0931	___	.0931
First FIP Bus Controller ***							
Subsequent FIP Bus Controllers							
FIP I/O Blocks							
per I/O block scan segment	.0061	.0132	.0114	.0116	.0161	.0025	.0161
per I/O block scan segment w/point faults enabled	.0148	.0176	.0119	.0153	.0323	.0670	.0323
per byte discrete I/O data in the main rack	.0018	.0018	.0019	.0024	.0019	.0025	.0019
per byte discrete I/O data in expansion racks	.0025	.0021	.0024	.0026	.0024	.0044	.0024
per word analog I/O data in the main rack	.0012	.0012	.0008	.0011	.0011	.0071	.0011
per word analog I/O data in expansion racks	.0024	.0024	.0043	.0032	.0024	.0098	.0024
Asynchronous Events							
Fault Message							

* Times are in milliseconds, except for those identified as microseconds.

** These are typical scan impacts. It is possible to incur up to an additional 2.1 milliseconds per scan segment, but this is not typical. This additional scan impact can usually be avoided when using synchronous scan sets that include only 1 FBC.

*** The extra time for the first FBC is the same time as shown in the next table for the first intelligent option module. This time should be counted only once.

Note

Functions in bold type in Table A-8 above impact the sweep continuously. All other functions impact the sweep only when invoked.

Also please note that the blank lines represent information not available at this time.

Use the following form for predicting the sweep impact due to FIP I/O. The sweep impact times can be found in table A-10.

Table A-11. Worksheet B: FIP I/O Sweep Time

Open system communications window		_____	=	_____
FBC I/O scan		_____		
FBC poll for background messages	+	_____	=	_____
Number of FBCs	x	_____	=	_____
Input block scan segments—number of		_____		
Input block scan segments—sweep impact	x	_____	=	_____
Output block scan segments—number of		_____		
Output block scan segments—sweep impact	x	_____	=	_____
Bytes of discrete I/O data on FBCs—main rack		_____		
Sweep impact/bytes of discrete I/O data—main rack	x	_____	=	_____
Bytes of discrete I/O data on FBCs—expansion racks		_____		
Seep impact/bytes of discrete I/O data—expansion racks	x	_____	=	_____
Words of analog I/O data on FBCs—main rack		_____		
Sweep impact/word analog I/O data—main rack	x	_____	=	_____
Words of analog I/O data on FBCs—expansion racks		_____		
Sweep impact/word analog I/O data—expansion racks	x	_____	=	_____
		Predicted FIP I/O Scan Impact		_____

Ethernet Global Data Sweep Impact

Depending on the relationship between the CPU sweep time and an Ethernet Global Data (EGD) exchange's period, the exchange data may be transferred every sweep or periodically after some number of sweeps. Therefore, the sweep impact will vary based on the number of exchanges that are scheduled to be transferred during the sweep. However, at some point during the operation of the PLC, all of the exchanges will be scheduled to transfer data during the same sweep. Therefore, all of the exchanges must be taken into account when computing the worst case sweep impact.

Each Ethernet Global Data exchange configured for either consumption or production can add up to 1 ms to the sweep time. This sweep impact should be taken into consideration when configuring the PLC constant sweep mode and setting the watchdog timeout.

The Ethernet Global Data (EGD) sweep impact has two parts, Consumption Scan and Production Scan:

EGD Sweep Impact	=	Consumption Scan	+	Production Scan

Where the Consumption and Production Scans consist of two parts, exchange overhead and byte transfer time:

Scan Time	=	Exchange Overhead	+	Byte Transfer Time

Exchange Overhead

Exchange overhead includes the setup time for each exchange that will be transferred during the sweep. When computing the sweep impact, include overhead time for each exchange.

Exchange Overhead*

	CPU Model	
Exchange Type	**924/925**	**914/915**
Consumed	61	106
Produced	88	133

* Times are in microseconds.

Byte Transfer Time

This is the time required to transfer the data between the PLC CPU module and the Ethernet module. The times shown in the following table represent the time to transfer one data byte.

Byte Transfer Time*

	CPU Models	
Exchange Type	**924/925**	**914/915**
Consumed	2.4	2.4
Produced	1.8	1.8

* Times are in microseconds

Table A-12. Worksheet: Ethernet Global Data Sweep Time

Number of consumed exchanges		_____		
Sweep impact per exchange	x	_____	=	_____
Number of data bytes in all of the consumed exchanges		_____		
Sweep impact per consumed data byte	x	_____	=	_____
Number of produced exchanges		_____		
Sweep impact per exchange	x	_____	=	_____
Number of data bytes in all of the produced exchanges		_____		
Sweep impact per produced data byte	x	_____	=	_____
		Predicted Ethernet Global Data Sweep Impact		_____

Note

If the PLC is configured for Microcycle Sweep Mode, the EGD sweep impact time is allocated to the Logic Window. Therefore, the time available to execution of the user program is reduced by the time required by Ethernet Global Data exchanges.

Sweep Impact of Intelligent Option Modules

Intelligent option modules include Programmable Coprocessor Modules (PCM), Alphanumeric Display Coprocessor (ADC) Modules, Graphics Display Coprocessor (GDC) Modules, MAP LAN Modules, Ethernet LAN Modules, and Genius Bus Controllers being used for Genius LAN capabilities. The sweep impact for these intelligent option modules is highly variable. The opening of the System Communications Window and the polling of each module have relatively small impacts compared to the sweep impact of CPU memory read or write requests.

The following equations show how to calculate the fixed sweep of each module.

PCM = Polling Sweep Impact + Clock Refresh Impact once every 1/2 sec.

ADC = Polling Sweep Impact + Clock Refresh Impact once every 1/2 sec.

GDC = Polling Sweep Impact + Clock Refresh Impact once every 1/2 sec.

MAP LAN = Polling Sweep Impact + LAN I/O Scan Impact

Ethernet LAN = Polling Sweep Impact + LAN I/O Scan Impact

GBC = Polling Sweep Impact + GBC I/O Scan Impact

FBC = Polling Sweep Impact + FBC I/O Scan Impact

The table below shows the fixed sweep impact times in milliseconds for intelligent option modules. It also contains sweep impact times for reading and writing the PLC's system memories (includes all memories except %P and %L, which are slightly slower). The read and write service requests have two boundary conditions that change the times and are, therefore, broken up into three sets of times to reflect these boundary conditions.

Table A-13. Fixed Sweep Impact Times for Intelligent Option Modules *

Sweep Impact Item	CPX 935	CPX 928	924/ 925	914/ 915	CPX 782	781/782 788/789	CPX 772	731/732 771/772
Intelligent Option Modules								
First module (open comm window)								
Per module (polling)								0.60
LAN module I/O Scan								0.10
PCM, ADC, GDC clock refresh	.0751	.0833	.0471	.0586	.1960	.0665	.1960	0.15
	.1630	.1434	.1386	.1455	.3651	.3248	.3651	0.60
PLC Memory Access from IOMs								
Read/write 1 to 3 words. *								
Read/write 4 to 128 words.	.4940	.5586	.3990	.4450	1.252	1.206	1.252	3.00
Read/write each additional 128 words.	.6320	.6964	.4830	.5790	1.420	1.393	1.420	3.10
	.2274	.2495	.6530	.8030	.4140	1.835	.4140	1.00

* Times are in milliseconds. Not all of the timing information needed for the above table was available at print time for this manual (the blank spaces). Refer to the IPI for each CPU for this information.

** Reads can only fit 3 words into the basic message.
Writes can fit 4 words before a 256 byte text buffer is needed.

Note

Functions in bold type in the previous table impact the sweep continuously. All other functions impact the sweep only when invoked.

I/O Interrupt Performance and Sweep Impact

There are several important performance numbers for I/O interrupt blocks or I/O-Triggered programs. The sweep impact of an I/O interrupt invoking an empty program or block measures the overall time of fielding the interrupt, starting up the program or block, exiting the program or block, and restarting the interrupted task. The maximum I/O interrupt rate reflects the limit of I/O interrupts invoking a minimal program or block at a sustained rate over time. The time to execute the logic contained in the interrupt program or block will affect the limit by causing the PLC to spend more time servicing I/O interrupts and thus reduce the maximum I/O interrupt rate.

The minimum, typical, and maximum interrupt response times reflect the time from when a single I/O module sees the input pulse until the first line of ladder logic or C code is executed in the I/O interrupt program or block. Minimum response time reflects a 300 microsecond input card filter time + time from interrupt occurrence to first line of ladder logic in I/O interrupt program or block. The minimum response time can only be achieved when no intelligent option modules are present in the system and the programmer is not attached. Typical response time is the minimum response time plus a maximum interrupt latency of 2.0 milliseconds for the model 731 CPU. This interrupt latency time is valid, except when one of the following operations occurs:

- The programmer is attached.
- A store of logic, RUN mode store, or word-for-word change occurs.
- A fault condition (logging of a fault) occurs.
- Another I/O interrupt occurs.
- The CPU is transferring a large amount of input (or output) data from an I/O controller (such as a Genius Bus Controller or a FIP Bus Controller). Heavily loaded I/O controllers should be placed in the main rack whenever possible.

Any one of these events extends the interrupt latency (the time from when the interrupt card signals the interrupt to the CPU to when the CPU services the interrupt) beyond the typical value. However, the latency of an interrupt occurring during the processing of a preceding I/O interrupt is unbounded. I/O interrupts are processed sequentially so that the interrupt latency of a single I/O interrupt is affected by the duration of the execution time of all preceding interrupt blocks. (Worst case is that every I/O interrupt in the system occurs at the same time so that one of them has to wait for all others to complete before it starts.)

The maximum response time shown below does not include the two unbound events.

Table A-14. I/O Interrupt Block Performance and Sweep Impact Times *

Sweep Impact Item	CPX 935	CPX 928	924/ 925	914/ 915	CPX 782	781/782 788/789	CPX 772	731/732 771/772
I/O interrupt sweep impact								0.95
I/O interrupt response time								
Input card filter time								0.30
+ typical interrupt latency								1.40
+ interrupt to logic time								0.48
Minimum response time	.3851	.3851	.313		.5402	.675	.5402	0.78
Typical response time	.3873	.3873	.316		.5421	.679	.5421	2.18
Maximum response time	.5424	.5424	.627		1.0528	1.633	1.0528	3.60
I/O interrupt rate limit								450 ints/sec
I/O interrupt rate limit w/no IOMs								750 ints/sec

* Times are in milliseconds. Not all of the timing information needed for the above table was available at print time for this manual (the blank spaces).

Table A-15. I/O-Triggered Interrupt Performance and Sweep Impact Times *

Sweep Impact Item	CPX 935	CPX 928	924/ 925	914/ 915	CPX 782	781/782 788/789	CPX 772
I/O interrupt sweep impact							
I/O interrupt response time							
Input card filter time							
+ typical interrupt latency							
+ interrupt to logic time							
Minimum response time	.3046	.3259			.4796		.4796
Typical response time	.3072	.3273			.4797		.4797
Maximum response time	.4618	.4822			.9921		.9921
I/O interrupt rate limit							
I/O interrupt rate limit w/no IOMs							

* Times are in milliseconds. Not all of the timing information needed for the above table was available at print time for this manual (the blank spaces).

The following form is a worksheet for the sweep impact times of programmer sweep impact, intelligent option modules, and I/O Interrupts. (Refer to tables A-4, A-10, and A-11.)

Note

Not all of the timing information needed for the above table was available at printing time for this manual (the blank spaces). Refer to the IPI for each CPU for this information.

Table A-16. Worksheet C: Programmer, IOM, I/O Interrupt Sweep Time

Programmer sweep impact			= _____
IOM—first module (open comm window)		_____	
IOM—per module (polling)		+ _____	
LAN module I/O scan		+ _____	
	Total IOM Sweep Impact		= _____
PLC memory access from IOMs			= _____
I/O interrupt sweep impact		_____	
I/O interrupt response time		+ _____	= _____
	Predicted Sweep Time (Other)		_____

Timed Interrupt Performance

The sweep impact of a timed interrupt invoking an empty program block or timed program measures the overall time of fielding the interrupt, starting up the program or block, exiting the program or block, and restarting the interrupted task. The minimum, typical, and maximum interrupt response times reflect the time from when a single timed interrupt occurs until the first line of ladder logic or C code is executed in the timed interrupt program or block. The minimum response time can only be achieved when no intelligent option modules are present in the system and the programmer is not attached. Typical response time is the minimum response time plus the CPU's maximum latency time. This interrupt latency time is valid, except when one of the following operations occurs:

- The programmer is attached
- A store of logic, ALT-S store, RUN mode store, or word-for-word change occurs
- A fault condition (logging of a fault) occurs
- Another timed interrupt or I/O interrupt occurs

Any one of these events extends the interrupt period beyond the typical value. However, the latency of an interrupt occurring during the processing of a preceding timed or I/O interrupt is unbounded. For interrupts, the worst case is that every timed and I/O interrupt in the system occurs at the same time so that one of them has to wait for all others to complete before it starts.

The maximum response time shown below does not include the two unbound events.

Table A-17. Timed Interrupt Performance and Sweep Impact Times *

Sweep Impact Item	CPX 935	CPX 928	924/ 925	914/ 915	CPX 782	781/782 788/789	CPX 772	731/732 771/772
Timed interrupt sweep impact								1.15
Timed interrupt response time								
Typical interrupt latency								1.40
+ interrupt to logic time								0.68
Minimum response time			.108	.147		.227		0.68
Typical response time			.143	.185		.346		2.08
Maximum response time			.163	.219		.464		3.50

* Times are in milliseconds. Not all of the timing information needed for the above table was available at print time for this manual (the blank spaces).

Table A-18. I/O-Triggered Interrupt Performance and Sweep Impact Times *

Sweep Impact Item	CPX 935	CPX 928	924/ 925	914/ 915	CPX 782	781/782 788/789	CPX 772
Timed interrupt sweep impact							
Timed interrupt response time Typical interrupt latency + interrupt to logic time Minimum response time Typical response time Maximum response time							

* Times are in milliseconds. Not all of the timing information needed for the above table was available at print time for this manual (the blank spaces).

Examples of Calculating Predicted Sweep Times

The following two examples are provided to show how to calculate predicted sweep times. The first example is a small system and the second is a large system. Neither of these sweep time estimates include a time for logic execution. In both of these systems, the calculated sweep is for normal sweep time with point faults disabled, the PCM idle, and the programmer not attached. The times used in the calculation are extracted from tables A-3, A-5, A-7, A-8, and A-10. Sample forms for calculating predicted sweep times are provided after the examples.

Small System

PS	CPU 731	BTM	32PT Input	32PT Input	32PT Output	32PT Output	8CHN Analog Input	4CHN Analog Output	PCM
0	1	2	3	4	5	6	7	8	9

MAIN RACK

Sweep Calculations

Predicted Sweep = Base Sweep + I/O Scan + PCM Impact

Base Sweep Time			=	2.00
I/O Scan Impact = I/O Scan Overhead + Series 90-70 I/O Scan Impact				
Number of discrete I/O modules—main rack		4		
Sweep impact time per discrete I/O module	x	0.08	=	0.32
Number of analog base and output modules—main rack		2		
Sweep impact time per analog base and output module	x	0.08	=	0.16
I/O Scan Impact = 0.32 + 16		.48	=	0.48
I/O scan overhead		.38	=	0.38
PCM Impact				
First module (open comm window)		.60		
Per module (polling)	+	.10	=	0.70
Predicted Sweep Time			=	4.04

Note

Times are in milliseconds.

Large System

PS	CPU 781	BTM	GBC 20Blks	GBC 20Blks	GBC 16Blks	GBC 16Blks	PCM	PCM	ENET LAN
0	1	2	3	4	5	6	7	8	9

MAIN RACK

PS	BRM	32PT Input	32PT Input	32PT Output	32PT Output	8CHN Analog Input	16CHN Analog Expander	4CHN Analog Output	4CHN Analog Output
0	1	2	3	4	5	6	7	8	9

RACK1

PS	BRM	32PT Input	32PT Input	32PT Output	32PT Output	8CHN Analog Input	16CHN Analog Expander	4CHN Analog Output	4CHN Analog Output
0	1	2	3	4	5	6	7	8	9

RACK2

Note

The Genius block configuration used for this example is ala 16-point grouped (QI) blocks with all bus addresses having contiguous reference addresses.

Predicted Sweep Calculations

Predicted Sweep = Base Sweep + IOM Impact

Base Sweep Time		0.10	=	0.10
I/O Scan overhead		0.20		
90-70 I/O scan impact (see table ____, Worksheet A)	+	0.80		
Genius I/O scan impact (see table _____, Worksheet B)	+	5.88		
I/O scan overhead			=	6.88
PCM impact	x	.30		
Number of PCMs		2		
LAN impact		0.60		
IOM impact	+	.08	=	.68
Predicted Sweep Time			=	8.66

Note

Times are in milliseconds.

Table A-19. Worksheet A

Number of expansion racks		2		
Sweep impact per expansion rack	x	.03	=	.06
Number of discrete I/O modules—main rack		_____		
Sweep impact per discrete I/O module—main rack	x	_____	=	_____
Number of discrete I/O modules—expansion rack		8		
Seep impact per discrete I/O module—expansion rack	x	.05	=	.40
Number of analog input base and output modules—main rack		6		
Sweep impact per analog input base and output module—main rack	x	.04	=	.24
Number of analog input exander modules (same segment)—main rack		_____		
Sweep impact per analog input expander module (same segment)—main rack	x	_____	=	_____
Number of analog input exander modules (new segment)—main rack		_____		
Sweep impact per analog input expander module (new segment)—main rack	x	_____	=	_____
Number of analog input base and output modules—expansion rack		_____		
Sweep impact per analog input base and output module—expansion rack	x	_____	=	_____
Number of analog input base and output modules (same segment)—exp. rack		2		
Sweep impact per analog input base and output module (same seg.)—exp. rack	x	.05	=	.10
Number of analog input base and output modules (new segment)—exp. rack		_____		
Sweep impact per analog input base and output module (new seg.)—exp. rack	x	_____	=	_____
Predicted Series 90-70 I/O Module Sweep Impact				.80

Table A-20. Worksheet B

Open system communications window		0.30	=	0.30
GBC I/O scan		0.94		
GBC poll for background messages	x	0.04	=	0.98
Number of GBCs	x	4	=	3.92
Input block scan segments—number of		4		
Input block scan segments—sweep impact	x	0.02	=	0.08
Output block scan segments—number of		4		
Output block scan segments—sweep impact	x	0.02	=	0.08
Bytes of discrete I/O data on GBCs—main rack		288		
Sweep impact/bytes of discrete I/O data—main rack	x	.0018	=	.5184
Bytes of discrete I/O data on GBCs—expansion racks		_____		
Seep impact/bytes of discrete I/O data—expansion racks	x	_____	=	_____
Words of analog I/O data on GBCs—main rack		_____		
Sweep impact/word analog I/O data—main rack	x	_____	=	_____
Words of analog I/O data on GBCs—expansion racks		_____		
Sweep impact/word analog I/O data—expansion racks	x	_____	=	_____
Predicted Genius I/O Scan Impact				5.88

Table A-21. Sample Worksheet A

Number of expansion racks		_____	
Sweep impact per expansion rack	x	_____	= _____
Number of discrete I/O modules—main rack		_____	
Sweep impact per discrete I/O module—main rack	x	_____	= _____
Number of discrete I/O modules—expansion rack		_____	
Seep impact per discrete I/O module—expansion rack	x	_____	= _____
Number of analog input base and output modules—main rack		_____	
Sweep impact per analog input base and output module—main rack	x	_____	= _____
Number of analog input exander modules (same segment)—main rack		_____	
Sweep impact per analog input expander module (same segment)—main rack	x	_____	= _____
Number of analog input exander modules (new segment)—main rack		_____	
Sweep impact per analog input expander module (new segment)—main rack	x	_____	= _____
Number of analog input base and output modules—expansion rack		_____	
Sweep impact per analog input base and output module—expansion rack	x	_____	= _____
Number of analog input base and output modules (same segment)—exp. rack		_____	
Sweep impact per analog input base and output module (same seg.)—exp. rack	x	_____	= _____
Number of analog input base and output modules (new segment)—exp. rack		_____	
Sweep impact per analog input base and output module (new seg.)—exp. rack	x	_____	= _____
Predicted Series 90-70 I/O Module Sweep Impact			_____

Table A-22. Sample Worksheet B

Programmer sweep impact			= _____
IOM—first module (open comm window)		_____	
IOM—per module (polling)	+	_____	
LAN module I/O scan	+	_____	
Total IOM Sweep Impact			= _____
PLC memory access from IOMs			= _____
I/O interrupt sweep impact		_____	
I/O interrupt response time	+	_____	= _____
Predicted Sweep Time (Other)			_____

Relative CPU Performance Comparison

This section contains a relative CPU performance comparison based on lab test results on the Series 90-70 CPUs.

Test Program

The test program used for the performance tests consisted of a 190 Kilobyte folder run on a CPX935 CPU. Sweep time was approximately 115 ms. In this test program, the Main block calls the same subroutine (S1) 4 times. The S1 subroutine, in turn, calls 60 other subroutines (S2 – S61) which each contain the same logic. The logic consists of :

- 140 contacts
- 50-70 coils
- 5 ADDs
- 2 DNCTRs
- 5 UPCTRs
- 10 EQ_INT
- 3 FIFO RDs
- 12 MOVEs
- 1 MUL_INT
- 1 NE_INT
- 8 ONDTRS
- 1 CALL
- 2 Bit Shifts
- 4 SUB_INTs
- 4 TMRs
- 1 DIVIDE
- 1 PI-ISA

Interpreting the Chart

The CPX935 performance time for running the test program was assigned a value of 100%, and the other CPUs were compared to that standard. The chart shows that the CPX935 had the fastest time. The CPU782 took the longest, 446% longer than the CPX935.

Note

The figures in the following chart were measured for the test program described above. They will not be exactly the same for different programs, but should provide a basic guide to relative CPU performance.

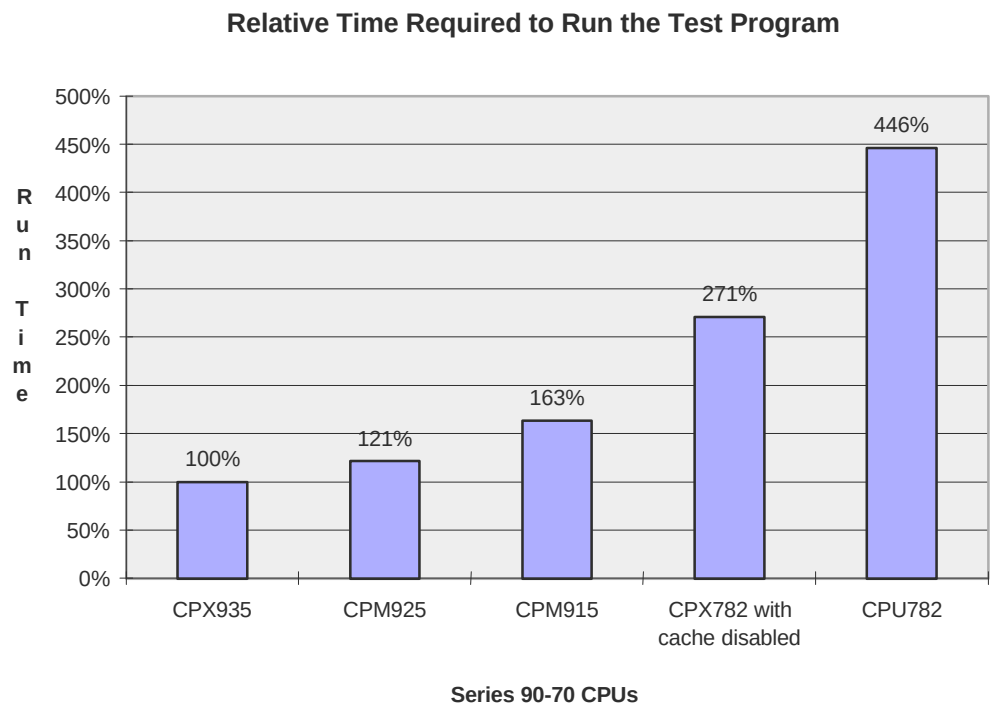


Figure A-1. Chart of Relative CPU Performance

Appendix B Interpreting Faults

The Series 90-70 PLC maintains two fault tables, the I/O fault table for faults generated by I/O devices (including I/O controllers) and the PLC fault table for internal PLC faults. The information in this appendix will enable you to interpret the message structure format when reading these fault tables.

This is a sample I/O Fault Table:

```
       CURRENT TABLE: IO                    COMMUNICATIONS STATUS: Connected
  TOP FAULT DISPLAYED: 0                       TABLE LAST CLEARED: 09-17-96 12:37:48
        TOTAL FAULTS: 1                      ENTRIES OVERFLOWED: 0
   FAULT DESCRIPTION:                              PLC DATE/TIME: 10-09-96 06:53:59

 FAULT   CIRC  REFERENCE        FAULT                  FAULT          DATE       TIME
LOCATION NO.     ADDR          CATEGORY               TYPE          MM-DD-YY HH:MM:SS.mm
0.3             %I   00081 Loss of I/O Module                      10-09-96 06:53:22.40
```

This is a sample PLC Fault Table:

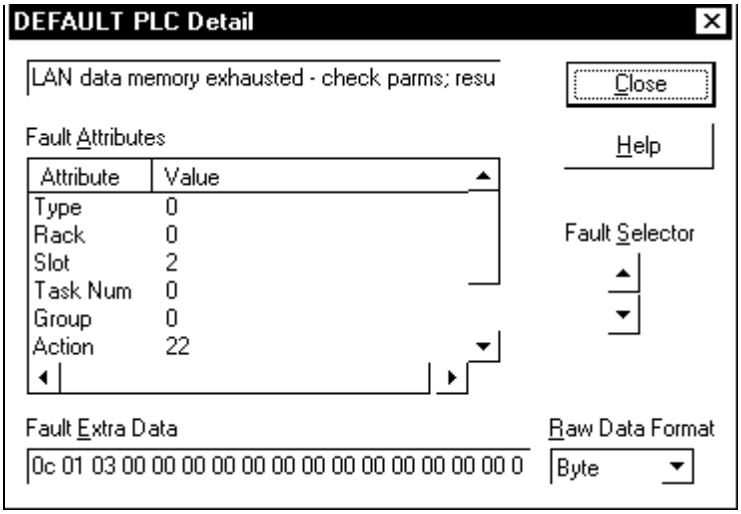
```
       CURRENT TABLE: PLC                   COMMUNICATIONS STATUS: Connected
  TOP FAULT DISPLAYED: 0                       TABLE LAST CLEARED: 10-02-96 16:06:03
        TOTAL FAULTS: 6                      ENTRIES OVERFLOWED: 0
                                                    PLC DATE/TIME: 10-09-96 10:42:52

 FAULT                              FAULT                          DATE       TIME
LOCATION                          DESCRIPTION                    MM-DD-YY HH:MM:SS:mm
0.1       Null system configuration for run mode                  10-03-96 09:09:30.00
0.1       Null system configuration for run mode                  10-03-96 06:48:00.00
0.2       LAN data memory exhausted - check parms; resuming       10-03-96 06:30:46.00
0.2       User application fault                                  10-02-96 16:24:12.00
0.1       Null system configuration for run mode                  10-02-96 16:17:49.00
0.1       Null system configuration for run mode                  10-02-96 16:07:07.00
```

Both tables contain similar information.

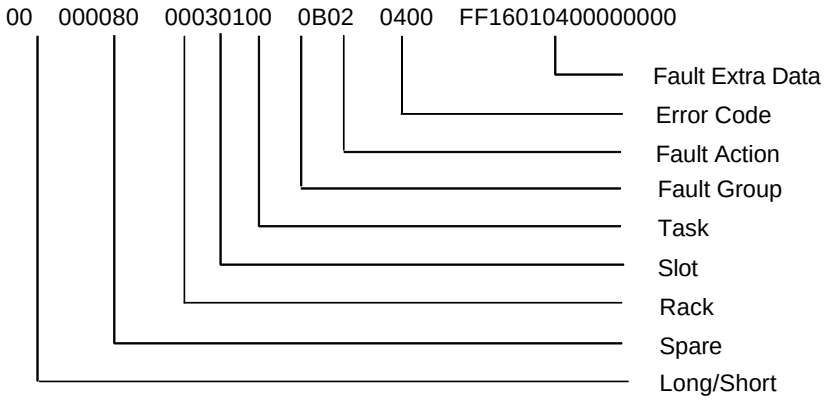
- The PLC fault table contains:
  - Fault location.
  - Fault description.
  - Date and time of fault.
- The I/O fault table contains:
  - Fault location.
  - Circuit number.
  - Reference address.
  - Fault category.
  - Fault type.
  - Date and time of fault.

The Series 90-70 PLC maintains additional information on each fault that is helpful when troubleshooting your system. This information is called CTRL-F data, or Fault Detail. You can access this information by highlighting the fault and pressing the **CTRL** and **F** keys together. You can also double-click the fault to see the Fault Detail dialog box.



PLC Fault Table

The following diagram identifies each field in the fault entry:



The following paragraphs describe each field in the fault entry. Included are tables describing the range of values each field may have.

Long/Short Indicator

This byte indicates whether the fault contains 8 bytes or 24 bytes of fault extra data.

Type	Code	Fault Extra Data
Short	00	8 bytes
Long	01	24 bytes

Spare

These 3 bytes are pad bytes, used to make the PLC fault table entry exactly the same length as the I/O fault table entry.

Rack

The rack number ranges from 0 to 7. Zero is the main rack, containing the PLC. Racks 1 through 7 are expansion racks, connected to the PLC through a Bus Transmitter Module in the main rack and Bus Receiver Modules in the expansion racks.

Slot

The slot number ranges from 0 to 9. The PLC CPU always occupies slot 1 in the main rack (rack 0).

Task

The task number ranges from 0 to +65,535. Sometimes the task number provides additional information to PLC engineers; typically, however, the task number can be ignored.

PLC Fault Group

Fault group is the highest classification of a fault. It identifies the general category of the fault. The fault description text displayed by your programming software is based on the fault group and the error codes.

Table B-1 lists the possible fault groups in the PLC fault table. Group numbers less than 80 (Hex) are maskable faults, while group numbers greater than or equal to 80 (Hex) are non-maskable faults.

The last non-maskable fault group, Additional PLC Fault Codes, is declared for the handling of new fault conditions in the system without the PLC having to specifically know the alarm codes. All unrecognized PLC-type alarm codes belong to this group.

Table B-1. PLC Fault Groups

Group Number		Group Name	Fault Action
Decimal	**Hexa-decimal**		
1	1	Loss of, or missing, rack	Fatal
4	4	Loss of, or missing, option module	Diagnostic
5	5	Addition of, or extra, rack	Diagnostic
8	8	Addition of, or extra, option module	Diagnostic
11	B	System configuration mismatch	Fatal
12	C	System bus error	Diagnostic
13	D	PLC CPU hardware failure *	Fatal
14	E	Non-fatal module hardware failure	Diagnostic
16	10	Option module software failure	Diagnostic
17	11	Program block checksum failure	Fatal
18	12	Low battery signal	Diagnostic
19	13	Constant sweep time exceeded	Diagnostic
20	14	PLC system fault table full	Diagnostic
21	15	I/O fault table full	Diagnostic
22	16	User Application fault	Diagnostic
–	–	Additional PLC fault codes	As specified
128	80	System bus failure	Fatal
129	81	No user's program on power-up	Informational
130	82	Corrupted user RAM detected	Fatal
131	83	Window completion failure in Constant Sweep mode (that is, all windows failed to receive their allotted time)	Informational
132	84	Password access failure	Informational
134	86	Null system configuration for RUN mode	Informational
135	87	PLC CPU software failure	Fatal
136	88	More than the allowable number of I/O bus controllers were found in the system	Fatal
137	89	PLC sequence-store failure	Fatal

* The PLC OK LED will flash on and off to indicate that the failure was not serious enough to prevent programmer communications from retrieving the fault table information.

PLC Fault Action

Each fault may have one of three actions associated with it.

Table B-2. PLC Fault Actions

Fault Action	Action Taken by CPU	Code
Informational	Log fault in fault table.	1
Diagnostic	Log fault in fault table. Set fault references.	2
Fatal	Log fault in fault table. Set fault references. Go to Stop mode.	3

Error Code

The error code further describes the fault. Each fault group has its own set of error codes.

The first table below shows error codes for the PLC Software Error Group (Group 87H).

Table B-3. Alarm Error Codes for PLC CPU Software Faults

Decimal	Hexa-decimal	Name
20	14	Corrupted PLC Program Memory.
39	27	Corrupted PLC Program Memory.
82	52	Backplane Communications Failed.
90	5A	User Shut Down Requested.
123	7B	Remote I/O Scanner Communications Failure.
149	95	Store from Flash on Power-Up Failed. **Note:** The first byte of the Fault Specific Data describes why the store from flash failed: **Error** / **Fault Specific Data Value** / **Description** DEVICE_NOT_AVAILABLE — CF — Specific device is not available in the system. BAD_DEVICE_DATA — CC — Data stored on device has been corrupted and is no longer reliable. Or, Flash Memory has not been initialized. DEVICE_RW_ERROR — CB — Error occurred during a read/write of the Flash Memory device. FLASH_INCOMPAT_ERROR — 8E — Data in Flash Memory is incompatible with the PLC CPU firmware release due to the CPU firmware revision numbers, the instruction groups supported, or the CPU model number. ITEM_NOT_FOUND_ERROR — 8D — One or more specified items were not found in Flash Memory.
All	All	PLC CPU Internal System Error

The next table shows the error codes for all the other fault groups.

Table B-4. Alarm Error Codes for PLC CPU Faults

Decimal	Hexa-decimal	Name
		PLC Error Codes for Loss of Option Module Group
3	3	Bus Transmitter Module Found in Expansion Rack
22	16	Analog Expander Located to Left of Base Converter Module
25	19	Lost Analog Expander Module
44	2C	Option Module Soft Reset Failed
45	2D	Option Module Soft Reset Failed
59	3B	Loss of, or missing communications driver
60	3C	Module in firmware update mode
65	41	Module is in Standalone mode; mail system not initialized
255	FF	Option Module Communication Failed
		Error Codes for Addition of, or Extra Rack Group
1	1	Addition of, or Extra Rack
		Error Codes for Reset of, Addition of, or Extra Option Module Group
2	2	Module Restart Complete
3	3	LAN Interface Restart Complete; Running a Utility
All others		Reset of, Addition of, or Extra Option Module
		Error Codes for Module Hardware Error Group
1	1	Non-fatal LAN Hardware error
416	1A0	Required 12V PS failed or missing
450	1C2	LAN Controller Underrun/Overrun; Resuming
451	1C3	LAN Interface Failure; Switched Off Network
452	1C4	LAN Network Problem; Performance Degraded
453	1C5	LAN Severe Network Problem; Attempting Recovery
454	1C6	LAN Transceiver Fault; Off Network Until Fixed
		Error Codes for Option Module Software Failure Group
1	1	Unsupported Board Type
2	2	COMREQ _ mailbox full on outgoing message that starts the COMREQ
3	3	COMREQ _ mailbox full on response
4	4	More Than One BTM in Rack
5	5	Backplane Communications with PLC; Lost Request
10	A	Error with LAN interaction
11	B	Resource (alloc, tbl ovrflw, etc.) error
12	C	VME backplane error
13	D	User program error
401	191	Module Software Corrupted; Requesting Reload
402	192	LAN System Software Fault; Resuming
403	193	LAN System Software Fault; Aborted Assoc and Resuming
404	194	LAN System Software Fault; Restarted LAN I/F
405	195	LAN System Software Fault; Reinitializing LLC

Table B-4. Alarm Error Codes for PLC CPU Faults (continued)

Decimal	Hexa-decimal	Name
Error Codes for System Configuration Mismatch Group		
2	2	Genius Block Model Number Mismatch
4	4	Genius Block I/O Type Mismatch
7	7	Daughter Board Mismatch
8	8	Analog Expansion Mismatch
9	9	Genius Block Broadcast Control Data (BCD) Length Mismatch
10	A	Unsupported Feature
11	B	Revision A BTM not in Right Slot
14	E	LAN Duplicate MAC Address
15	F	LAN Duplicate MAC Address Resolved
16	10	LAN MAC Address Mismatch
17	11	LAN Soft Switch/Modem Mismatch
19	13	Genius Block Direct Control Data (DCD) Length Mismatch
23	17	Program exceeds memory limits
29	1D	Incompatible scheduling mode
30	1E	Reference length mismatch
31	1F	Invalid configuration parameters
32	20	New configuration requires reset
36	24	I/O specification mismatch
37	25	Controller reference out of range
39	27	Bad interrupt trigger
Error Codes for System Bus Error Group		
4	4	Unrecognized VME Interrupt Error
All others		System Bus Error
Error Codes for Program Block Checksum Group		
0	0	Corruption of program block header in the Series 90-70 PLC
1	1	Corruption of stored OMF records stored in Series 90-70 PLC
2	2	Corruption of stored OMF records stored in Series 90-70 PLC
3	3	Program or program block checksum failure
Error Codes for Low Battery Signal		
0	0	Failed battery on PLC CPU or other module
1	1	Low battery on PLC CPU or other module
2	2	Failed battery from VME backplane
3	3	Low battery from VME backplane

Table B-4. Alarm Error Codes for PLC CPU Faults (continued)

Decimal	Hexa-decimal	Name
		Error Codes for User Application Fault Group
1	1	Indirect Reference Address Out of Range
2	2	PLC Watchdog Timer Timed Out
3	3	GBC COMREQ
4	4	GBC Bkgnd msg – Bad Genius Bus Request
5	5	COMREQ – WAIT mode not available for this command
6	6	COMREQ – Bad Task ID
7	7	Application Stack Overflow
8	8	LAN Data Memory Exhausted – Check Parms; Resuming
9	9	Bad Remote Application Request; Discarded Request
10	A	Bad Local Application Request; Discarded Request
11	B	LAN I/F Capacity Exceeded; Discarded Request
12	C	LAN PROM/Software Mismatch; Running Soft Sw Util
13	D	LAN I/F Can't Init – Check Parms; Running Soft Sw Util
14	E	Run-time error detected in an external block
15	F	SORT function in an interrupt did not execute
17	11	Standalone Run-Time Error
28	1C	Program Exceeded Wind Down
29	1D	Program Not Readied
		Error Codes for System Bus Failure Group
1	1	Operating system
		Error Codes for Corrupted User RAM on Powerup Group
1	1	Corrupted User RAM on Power-up
2	2	Illegal Boolean Opcode Detected
5	5	Partial Store failure on second pass of parsing OMF
6	6	Corrupted Remote I/O Scanner EEPROM; Config Lost
		Error Codes for PLC CPU Hardware Faults
2169	879	Remote I/O Scanner Comm Failure; Verify Bus
2172	87C	Remote I/O Scanner Serial Bus Address Conflict
2048	800	Remote I/O Scanner Hardware Fault
to	to	
4095	FFF	
All other codes		PLC CPU Hardware Failure

PLC Fault Extra Data

The Fault Extra Data field in the PLC fault table contains details of the fault entry. Some examples of what data may be present are:

System Configuration Mismatch

Four error codes in the System Configuration Mismatch group supply fault extra data:

Table B-5. PLC Fault Extra Data – System Configuration Mismatch

Fault Extra Data Byte Number	Model Number Mismatch
[0]	FF
[1]	Bus address
[2]	Installed module's model number
[3]	Configured model number
Fault Extra Data Byte Number	**I/O Type Mismatch**
[0]	FF
[1]	Bus address
[2]	Installed module's I/O type
[3]	Configured module's I/O type
Fault Extra Data Byte Number	**BCD Length Mismatch**
[0]	FF
[1]	Bus address
[2]	Module's broadcast data length
[3]	Configured module's broadcast data length
Fault Extra Data Byte Number	**DCD Length Mismatch**
[0]	FF
[1]	Bus address
[2]	Module's directed data length
[3]	Configured module's directed data length

The following table shows the Genius numbers used when a model number mismatch occurs.

Table B-6. Genius Block Model Numbers

Number		Description
Decimal	**Hexadecimal**	
4	4	Genius Network Interface (GENI)
5	5	Phase B Hand Held Monitor
6	6	Phase B Series Six Genius Bus Controller with Diagnostics
7	7	Phase B Series Six Genius Bus Controller without Diagnostics
8	8	PLCM/Series Six
9	9	PLCM/Series 90-40
10	A	Series 90-70 Single Channel Bus Controller
11	B	Series 90-70 Dual Channel Bus Controller
12	C	Series 90-10 Genius Communications Module
13	D	Series 90-30 Genius Communications Module
32	20	High Speed Counter
69	45	Phase B 115Vac 8-point (2 amp) Grouped Block
70	46	Phase B 115Vac/125Vdc 8-point Isolated Block
70	46	Phase B 115Vac/125Vdc 8-point Isolated Block without Failed Switch
71	47	Phase B 220Vac 8-point Grouped Block
72	48	Phase B 24-48Vdc 16-point Proximity Sink Block
72	48	Phase B 24Vdc 16-point Proximity Sink Block
73	49	Phase B 24-48Vdc 16-point Source Block
73	49	Phase B 24Vdc 16-point Proximity Source Block
74	4A	Phase B 12-24Vdc 32-point Sink Block
75	4B	Phase B 12-24Vdc 32-point Source Block
76	4C	Phase B 12-24Vdc 32-point 5V Logic Block
77	4D	Phase B 115Vac 16-point Quad State Input Block
78	4E	Phase B 12-24Vdc 16-point Quad State Input Block
79	4F	Phase B 115/230Vac 16-point Normally Open Relay Block
80	50	Phase B 115/230Vac 16-point Normally Closed Relay Block
81	51	Phase B 115Vac 16-point AC Input Block
82	52	Phase B 115Vac 8-point Low-Leakage Grouped Block
127	7F	Genius Network Adapter (GENA)
131	83	Phase B 115Vac 4-input, 2-output Analog Block
132	84	Phase B 24Vdc 4-input, 2-output Analog Block
133	85	Phase B 220Vac 4-input, 2-output Analog Block
134	86	Phase B 115Vac Thermocouple Input Block
135	87	Phase B 24Vdc Thermocouple Input Block
136	88	Phase B 115Vac RTD Input Block
137	89	Phase B 24/48Vdc RTD Input Block
138	8A	Phase B 115Vac Strain Gauge/mV Analog Input Block
139	8B	Phase B 24Vdc Strain Gauge/mV Analog Input Block
140	8C	Phase B 115Vac 4-input, 2-output Current Source Analog Block
141	8D	Phase B 24Vdc 4-input, 2-output Current Source Analog Block

If the model number is 7FH (Genius Network Adapter), the block may be one of the following. (The GENA Application ID is shown for reference.)

Table B-7. GENA Application ID Numbers

Number		Description
Decimal	Hexadecimal	
131	83	115Vac/230Vac/125Vdc Power Monitor Module
132	84	24/48Vdc Power Monitor Module
160	A0	Genius Remote 90-70 Rack Controller

When the system configuration mismatch is an I/O type mismatch, the installed module I/O type is one of the following:

Table B-8. Genius Installed Module I/O Types

Value	Description
01	Input only
02	Output only
03	Combination

When the system configuration mismatch is an I/O type mismatch, the configured module I/O type is one of the following. (All values are in hexadecimal.)

Table B-9. Genius Configured Module I/O Types

Value		Description
Decimal	Hexadecimal	
0	0	Discrete input
1	1	Discrete output
2	2	Analog input
3	3	Analog output
4	4	Discrete grouped
5	5	Analog grouped
20	14	Analog in, discrete in
21	15	Analog in, discrete out
24	18	Analog in, discrete grouped
30	1E	Analog out, discrete in
31	1F	Analog out, discrete out
34	22	Analog out, discrete grouped
50	32	Analog grouped, discrete in
51	33	Analog grouped, discrete out
54	36	Analog grouped, discrete grouped

Program Block Checksum Failure

The name of the offending program block is contained in the first eight bytes of the Fault Specific Data field.

PLC CPU Hardware Failure (RAM Failure)

For a RAM failure in the PLC CPU (one of the faults reported as a PLC CPU hardware failure), the address of the failure is stored in the first four bytes of the field.

Application Fault

Indirect Reference Overflow: The offset address of where the call was made is located in the first two bytes. The offending reference (segment selector and offset) is located in the next four bytes. The name of the program block in which the function call resides is contained in the next eight bytes.

Bad COMREQ Status Pointer: The first byte contains a hex FF. The next four bytes contain the segment selector and offset of the status pointer into which the PLC CPU could not write.

Bad Genius Bus Request: Four bytes are used, unless the request is a read or write device. In these two datagrams, eight bytes are used.

Table B-10. Fault Specific Data - Bad Genius Bus Request

Fault Specific Data	Bad Genius Bus Request
[0]	FF
[1]	Bus address of requestor
[2]	Function code
[3]	Subfunction code
[4]	Segment selector, if Read/Write device
[5]	LSB of offset, if Read/Write device
[6]	MSB of offset, if Read/Write device
[7]	Length, if Read/Write device

PLC Fault Time Stamp

The six-byte time stamp is the value of the system clock when the PLC CPU recorded the fault. (Values are coded in BCD format.)

Table B-11. PLC Fault Time Stamp

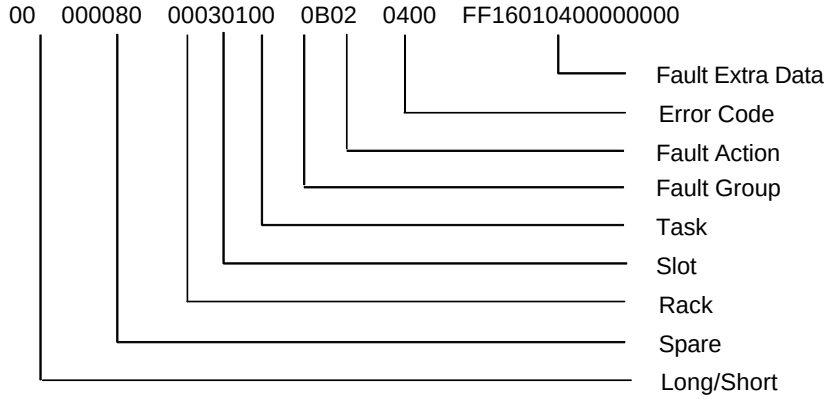
Byte Number	Description
1	Seconds
2	Minutes
3	Hours
4	Day of the month
5	Month
6	Year

Genius Block I/O Type Mismatch Example

The Genius Block I/O Type Mismatch fault entry is explained below. (All data is in hexadecimal.)

Field	Value	Description
Long/Short	00	This fault contains 8 bytes of fault extra data
Rack	00	Main rack (rack 0)
Slot	03	Slot 3. In this configuration, slot 3 contains a Genius Bus Controller
Task	01	Single channel bus controller has only one task
Fault Group	0B	System Configuration Mismatch fault
Fault Action	02	Diagnostic fault
Error Code	04	Error code 04 in the System Configuration Mismatch group is a Genius Block I/O Type Mismatch
Fault Extra Data		IO Type Mismatch error has four bytes of fault extra data
[0]	FF	Flag byte
[1]	16	Serial bus address: 16 hex is 22 decimal
[2]	01	Installed module type. 01 is an inputs only module
[3]	04	Configured module I/O type. 04 is a combination module

The configuration file stored from the programming software shows that the device at serial bus address 22 is a combination (mixed) module. However, the “Read ID Reply” message the Genius Bus Controller received from the device at serial bus address 22 shows that the device is configured for inputs only. The Genius Bus Controller logged this fault when it detected the mismatch.

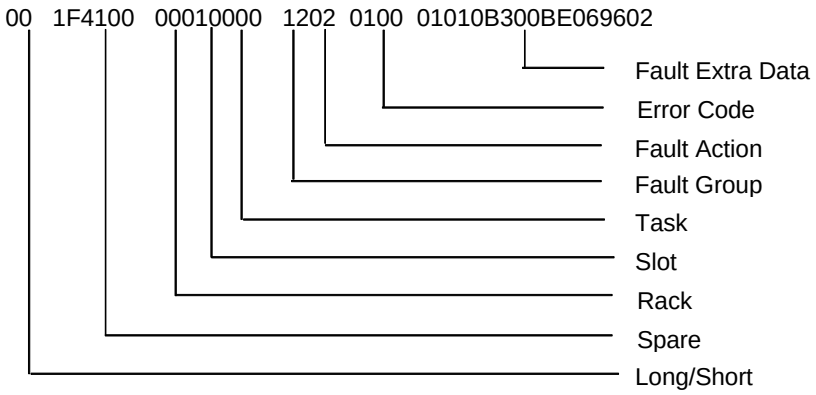


Low Battery Signal Example

The Low Battery fault is described below. (All data is in hexadecimal.)

Field	Value	Description
Long/Short	00	This fault contains 8 bytes of fault extra data.
Rack	00	Main rack (rack 0).
Slot	01	Slot 1. In all configurations, slot 1 in rack 0 contains the PLC CPU.
Task	00	PLC CPU.
Fault Group	12	Low battery signal.
Fault Action	02	Diagnostic fault.
Error Code	01	Error code 01 in the Low Battery group is one of four low battery conditions detected by the PLC CPU.
Fault Extra Data		No fault extra data for a low battery signal.

This fault occurred because the PLC CPU detected a low battery signal.

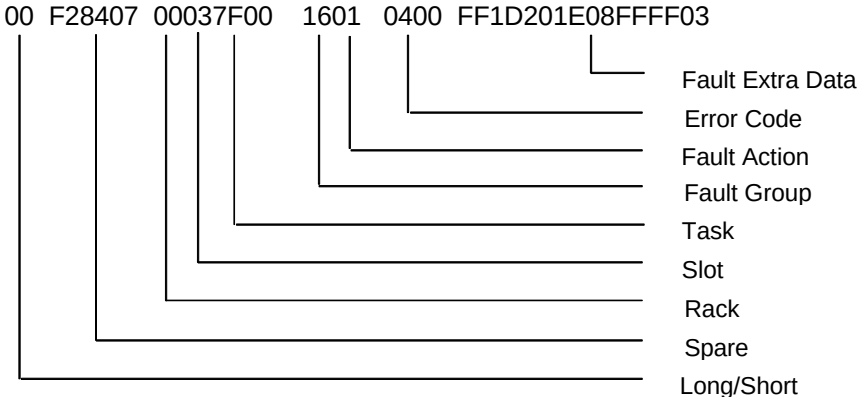


User Application Fault Example

The User Application Fault is described below. (All data is in hexadecimal.)

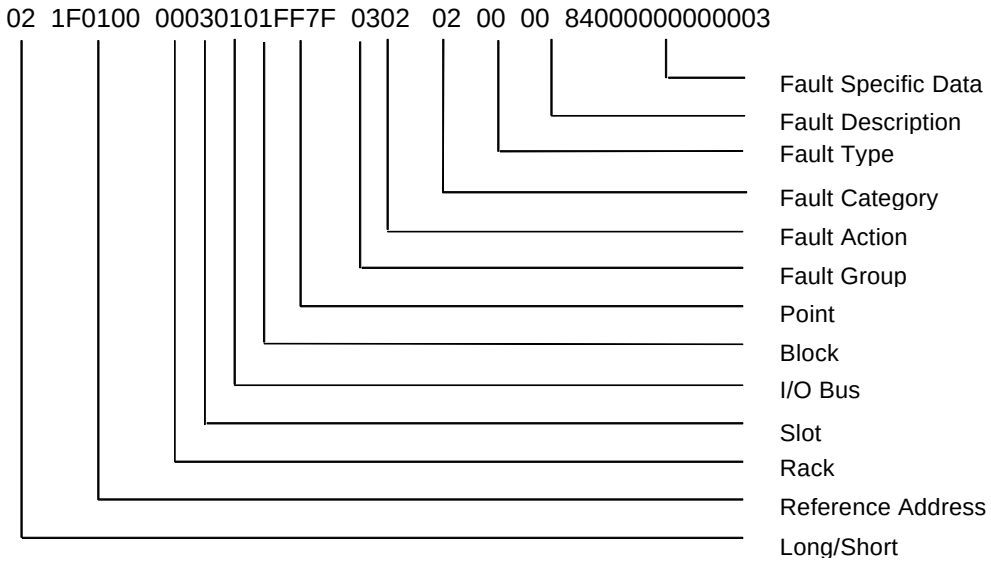
Field	Value	Description
Long/Short	00	This fault contains 8 bytes of fault extra data
Rack	00	Main rack (rack 0)
Slot	03	Slot 3. In this configuration, slot 3 contains a Genius Bus Controller
Task	7F	When the GBC registers a User Application fault in the PLC fault table, it places a 7F in the task byte of the fault.
Fault Group	16	User Application fault
Fault Action	01	Informational fault
Error Code	04	Error code 04 in the Application Fault group is a Bad Genius Bus Request. This fault occurs when the Genius Bus Controller receives a Read or Write Device datagram from another device on the Genius Bus that cannot be successfully completed.
Fault Extra Data		Bad Genius Bus Request error has eight bytes of fault extra data
[0]	FF	Flag byte
[1]	1D	Serial bus address: 1D hex is 29 decimal
[2]	20	Function code in the datagram: a GE Fanuc datagram
[3]	1E	Subfunction code in the datagram: a Read Device
[4]	08	Segment selector: 08 is %R memory
[5]	FF	Least significant byte of offset
[6]	FF	Most significant byte of offset
[7]	03	Length of data to read: 3 words

The Genius Bus Controller received a Read Device datagram from serial bus address 29, which requested three words of %R memory be read starting at %R65536. Since this is beyond the range of the largest value %R can have, the bus controller registered an informational user application fault in the PLC fault table.



I/O Fault Table

The following diagram identifies the hexadecimal information displayed in each field in the fault entry.



The following paragraphs describe each field in the I/O fault table. Included are tables describing the range of values each field may have.

Long/Short Indicator

This byte indicates whether the fault contains 5 bytes or 21 bytes of fault specific data.

Table B-12. I/O Fault Table Format Indicator Byte

Type	Code	Fault Specific Data
Short	02	5 bytes
Long	03	21 bytes

Reference Address

Reference address is a three-byte address containing the I/O memory type and location (or offset) in that memory which corresponds to the point experiencing the fault. Or, when a Genius block fault or integral analog module fault occurs, the reference address refers to the first point on the block where the fault occurred.

Table B-13. I/O Reference Address

Byte	Description	Range
0	Memory Type	0 - FF
1-2	Offset	0 - 12K (decimal)

The memory type byte is one of the following values.

Table B-14. I/O Reference Address Memory Type

Name	Value (Hexadecimal)
Analog input	0A
Analog output	0C
Analog grouped	0D
Discrete input	10 or 46
Discrete output	12 or 48
Discrete grouped	1F

I/O Fault Address

The I/O fault address is a six-byte address containing rack, slot, bus, block, and point address of the I/O point, which generated the fault. The point address is a word; all other addresses are one byte each. All five values may not be present in a fault.

When an I/O fault address does not contain all five addresses, a 7F hex appears in the address to indicate where the significance stops. For example, if 7F appears in the bus byte, then the fault is a module fault. Only rack and slot values are significant.

Rack

The rack number ranges from 0 to 7. Zero is the main rack, that is, the one containing the PLC. Racks 1 through 7 are expansion racks, connected to the PLC through a Bus Transmitter Module in the main rack and Bus Receiver Modules in the expansion racks.

Slot

The slot number ranges from 0 to 9. The PLC CPU always occupies slot 1 in the main rack (rack 0).

I/O Bus

The I/O bus number ranges from 0 to 15. When the module in the slot is a single-channel Genius Bus Controller, this number is always one. When the module is an integral analog module, this designates which expansion channel generated the fault.

Block

Block refers to the serial bus address of the Genius block that reported or has the fault.

Point

Point ranges from 1 to 1024 (decimal). It tells which point on the block has the fault when the fault is a point-type fault.

I/O Fault Group

Fault group is the highest classification of a fault. It identifies the general category of the fault. The fault description text displayed by the software is based on the fault group and the error codes.

Table B-15 lists the possible fault groups in the I/O fault table. Group numbers less than 80 (Hex) are maskable faults.

The last non-maskable fault group, Additional I/O Fault Codes, is declared for the handling of new fault conditions in the system without the PLC having to specifically know the alarm codes. All unrecognized I/O-type alarm codes belong to this group.

Table B-15. I/O Fault Groups

Group Number	Group Name	Fault Action
2	Loss of, or missing, IOC	Fatal
3	Loss of, or missing, I/O module	Diagnostic
6	Addition of, or extra, IOC	Diagnostic
7	Addition of, or extra, I/O module	Diagnostic
9	IOC or I/O bus fault	Diagnostic
A	I/O module fault	Diagnostic
F	IOC software failure	Fatal
–	Additional I/O fault codes	As specified

I/O Fault Action

The fault action specifies what action the PLC CPU should take when a fault occurs. The following table lists possible fault actions.

Table B-16. PLC Fault Actions

Fault Action	Action Taken by CPU	Code
Informational	Log fault in fault table	1
Diagnostic	Log fault in fault table Set fault references	2
Fatal	Log fault in fault table Set fault references Go to Stop mode	3

I/O Fault Category

The I/O fault category specifies a general classification of the fault. It is similar to the I/O fault group, discussed earlier. I/O fault categories are listed in the following table.

Table B-17. I/O Fault Categories

Decimal Number	Hex Code	Description
1	1	Circuit fault: Short circuit, open wire, etc.
2	2	Loss of block: Block no longer responding
3	3	Addition of block: New block appeared
4	4	Unused category
5	5	Unused category
6	6	Genius bus fault
7	7	Global memory fault
8	8	EEPROM fault, watchdog timeout
9	9	Addition of IOC
10	A	Loss of, or missing, IOC
11	B	IOC software fault
12	C	Forced circuit: A Genius I/O point was forced with the HHM
13	D	Unforced circuit: HHM force was removed
14	E	Loss of Series 90 integral card
15	F	Addition of Series 90 integral card
16	10	Found extra Series 90 integral card
17	11	Found extra Genius block
18	12	An IOC detected a hardware failure or a baud rate mismatch
19	13	Genius bus controller has stopped reporting faults because too many faults have occurred
20	14	Configuration mismatch fault for I/O modules
21	15	GBC software exception
22	16	Redundant Genius block switched bus
23	17	Block not active on redundant bus

I/O Fault Type

The I/O fault type component creates subcategories under the circuit fault, module fault, I/O bus fault, loss of block, excessive faults, I/O configuration mismatch, and GBC software exception categories. It is undefined for other fault categories, but is always set to zero when the fault category is something other than these seven categories. Table B-18 lists the I/O fault types.

Table B-18. I/O Fault Types

Number	Description
I/O Fault Types for the Circuit Fault Category	
1	Circuit fault on a discrete I/O point
2	Circuit fault on an analog I/O channel
3	Fault on a GENA
4	Fault on a low-level analog input channel
5	Fault on Remote I/O Scanner
I/O Fault Types for the Module Fault Category	
0	Block Fault (EEPROM, watchdog)
1	Analog to digital communications fault or calibration error
5	User scaling error caused out of range values
I/O Fault Types for I/O Bus Fault Category	
0	Genius IOC disabled all outputs on the bus because communications timed out between the PLC CPU and the Genius IOC
1	Genius Bus fault (No interrupt to the GBC for its turn on the bus within the time-out period)
3	Genius IOC detected a conflict between its SBA and another device on the bus
I/O Fault Types for the Loss of Block Category	
0	No reason specified.
1	Loss of A/D communications.
I/O Fault Types for Excessive Faults Category	
1	Genius IOC detected a high error count on the Genius I/O Bus and dropped off the bus for at least 1.5 seconds
I/O Fault Types for Configuration Mismatch Category	
2	Model number mismatch detected by I/O scanner
3	Nonexistent I/O module detected by I/O scanner
4	I/O type mismatch detected by I/O scanner
8	Integral analog module detected; expansion analog module mismatch
9	Broadcast Data Length mismatch
A	A configured feature is not supported

Table B-18. I/O Fault Types (cont'd.)

I/O Fault Types for GBC Software Exception Category	
1	Incoming datagram queue is full
2	The queue for Read/Write requests in the GBC is full. The requests may be from the Genius Bus or from COMMREQs
3	The low priority mail queue from the GBC to the PLC is full. The response to the PLC was lost.
4	Genius background message requiring PLC action was received before PLC completed initialization. Message was ignored.
5	Genius Report Fault message was not processed because GBC software revision is too old.
6	Excessive usage of internal GBC memory. User should verify COMMREQ usage.

I/O Fault Description

The I/O fault description component provides a specific fault code when the I/O fault category is a circuit fault (discrete circuit fault, analog circuit fault, low-level analog fault) or module fault. It is undefined for other faults, but is always set to zero. The next two tables list the possible fault descriptions.

Table B-19. I/O Fault Descriptions

Number	Description
Fault Descriptions for Discrete Circuit Faults	
01	Loss of user side power.
02	Short in user wiring (for Genius, current level greater than 20 Amps).
04	Sustained overcurrent (for Genius, current level greater than 2 Amps).
08	Very low or no current flow.
10	Switch temperature too high.
20	Genius smart switch failure.
83	Series 90-70 I/O individual point fault (also indicated for I/O Module Fault category).
84	Series 90-70 output fuse blown (also indicated for I/O Module Fault category).
Fault Descriptions for Analog Circuit Faults	
01	Input channel low alarm.
02	Input channel high alarm.
04	Input channel under range.
08	Input channel over range.
10	Open wire detected on input channel.
20	Output channel under range.
40	Output channel over range.
80	Expansion channel not responding.
80	Feedback error for Genius Current Source Analog Block.

Table B-19. I/O Fault Descriptions (cont'd)

Number	Description
Fault Descriptions for Low-level Analog Circuit Faults *	
20	Improper RTD connection or thermocouple reverse junction fault.
40	Cold junction sensor fault on thermocouple block or internal error in RTD block.
80	Input channel shorted (Genius RTD and Strain Gauge Blocks only).
Fault Descriptions for Module Faults **	
08	Genius EEPROM or NVRAM failure.
20	Genius calibration memory failure.
40	Genius shared RAM fault.
80	Genius internal circuit fault.
81	Watchdog timeout (discrete I/O modules only).
82	Aux fault on discrete I/O modules.
83	Series 90-70 I/O individual point fault (also indicated for CIRCUIT_FAULT category).
84	Series 90-70 output fuse blown (also indicated for CIRCUIT_FAULT category).
Fault Descriptions for GENA Faults	
80	Fault on a GENA analog or discrete point.
87	Fault on Remote I/O Scanner

* The following analog faults also apply when the low-level analog fault type is indicated: AI_LOW_ALARM, AI_HI_ALARM, AI_UNDER_RANGE, AI_OVER_RANGE, and OPEN_WIRE.

** These faults are reported under the HEADEND_FAULT fault type.

I/O Fault Specific Data

An I/O fault table entry may contain up to 21 bytes of I/O fault specific data. In general, this area contains additional information related to the fault. Not all entries contain I/O fault specific data. This section describes those that do. All but one of these faults uses five bytes of I/O fault specific data; the global memory fault uses the 21-byte entry. If a fault is not listed, it does not have I/O fault specific data.

Faults originated by the Genius Bus Controller always have at least one byte of I/O fault specific data. This byte is in addition to whatever other data might be present.

Circuit Fault

Circuit fault entries use one or two bytes of the fault specific data area. When the Genius Bus Controller reports the fault, the Genius Bus Controller generates the first byte. If the fault type is a GENA fault, the second byte contains the data that was reported from the GENA module in fault byte 2 of its “Report Fault” message. If the fault type is not a GENA fault, the second byte contains the circuit configuration and is encoded, as shown in table B-20.

Loss/Addition of Block

In the case of a Loss of Block or Addition of Block fault, four bytes of fault specific data are used. The first byte is encoded, as shown in table B-20. The second byte contains the block configuration and is encoded as shown in table B-20. The third byte specifies the number of input circuits possibly used, and the fourth byte specifies the number of output circuits possibly used.

Global Memory Fault

The Global Memory fault uses 10 data bytes. The first byte contains the subnet group number. The other nine bytes contain the global variable name, formatted as a null terminated string.

Forced/Unforced Circuit

Three bytes of fault specific data are present when a circuit force is added or removed (Forced circuit or Unforced circuit). The first byte contains a code from table B-20. The second byte contains the block configuration, and byte 3 contains the discrete/analog indication.

Loss of or Missing IOC

When the PLC CPU registers a Loss of or Missing IOC fault, it includes in the I/O fault specific data one of the codes in table B-20.

Other I/O Faults

In addition to those faults listed above, when the Genius Bus Controller reports one of the following faults, it includes one of the bytes in table B-20 in the I/O fault specific data.

- I/O Bus fault
- Module fault
- IOC software fault
- Extra Genius block
- IOC hardware fault
- Excessive faults
- GBC software exception

Block Switch

When the fault category is Block Switch, five bytes of fault specific data are used. The first byte is encoded, as shown in table B-20. The second byte contains the block configuration and is encoded as shown in table B-20. The third byte specifies the number of input circuits possibly used, and the fourth byte specifies the number of output circuits possibly used. The last byte (byte 5) of fault specific data contains the rack/slot address of the Genius bus controller which controlled the bus that the block switched off. It is encoded with the slot number in the low four bits and the rack number in the high four bits.

Symbolic Fault Specific Data

The following table lists data that is required for four kinds of fault specific data:

- Block circuit configuration
- Block usage indication
- Discrete/analog indication
- Loss of IOC error code

Table B-20. I/O Fault Specific Data

Decimal Number	Hex Code	Description
		Circuit Configuration
	1	Circuit is an input
	2	Circuit is an output
	3	Circuit is an output with feedback
		Block Configuration
	1	Block configured for inputs only
	2	Block configured for outputs only
	3	Block has both inputs and outputs
		Discrete/Analog Indication
	1	Discrete block
	2	Analog block
		LOSS_OF_IOC Error Code
1	1	IOC failed to respond to a CPU request
2	2	CPU and IOC lost synchronization
3	3	CPU/IOC communications failed
4	4	VME bus error
5	5	VME bus error
6	6	CPU/IOC communications failed
7	7	CPU/IOC communications failed
8	8	IOC failed to respond to a CPU request
9	9	CPU/IOC communications failed
10	A	VME bus error
11	B	BME bus error
12	C	CPU/IOC communications failed
13	D	CPU/IOC communications failed
14	E	CPU/IOC communications filed.
15	F	IOC failed to respond to a CPU request
16	10	CPU/IOC communications failed
17	11	CPU/IOC communications failed
18	12	IOC failed to respond to a CPU request
19	13	CPU/IOC communications failed
20	14	CPU/IOC communications failed
21	15	Internal I/O scanner fault detected
22	16	IOC failed to respond to a CPU request
23	17	IOC failed to respond to a CPU request
24	18	IOC failed to respond to a CPU request
25	19	IOC failed to respond to a CPU request
26	1A	IOC failed to respond to a CPU request
27	1B	CPU/IOC communications failed
28	1C	VME bus error occurred while reading input data

Table B-20. I/O Fault Specific Data (cont'd)

Decimal Number	Hex Code	Description
29	1D	VME bus error occurred while reading input diagnostics
30	1E	CPU/IOC communications failed
31	1F	VME bus error occurred while writing output data to IOC
32	20	CPU/IOC communications failed
33	21	CPU/IOC communications failed
34	22	CPU/IOC communications failed
35	23	VME bus error
36	24	VME bus error
37	25	Unable to read data from IOC for redundant I/O blocks
38	26	Unable to write data to IOC for redundant I/O blocks
39	27	IOC does not support configured I/O redundancy
40	28	IOC failed to respond to a CPU request
41	29	I/O scanner detected too many IOCs in the system
42	2A	I/O scanner detected too many IOCs in the system
43	2B	VME bus error
44	2C	VME bus error

Fault Actions for Specific Faults

Forced/unforced circuit faults are reported as informational faults. All others are diagnostic or fatal.

The model number mismatch, I/O type mismatch and nonexistent I/O module faults are reported in the PLC fault table under the System Configuration Mismatch group. They are not reported in the I/O fault table.

I/O Fault Time Stamp

The six-byte time stamp is the value of the system clock when the PLC CPU recorded the fault. Values are coded in BCD format.

Table B-21. I/O Fault Time Stamp

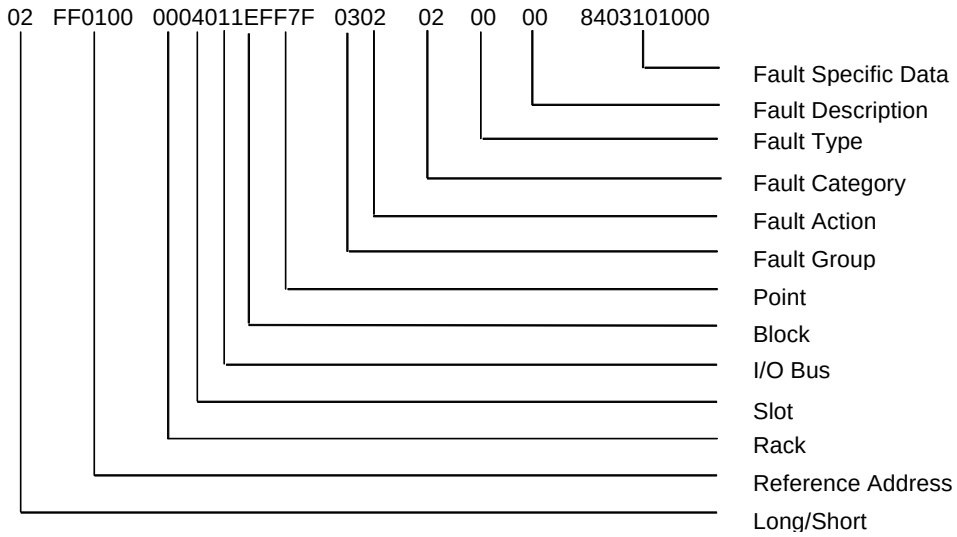
Byte Number	Description
1	Seconds.
2	Minutes.
3	Hours.
4	Day of the month.
5	Month.
6	Year.

Loss of I/O Block Example

The loss of I/O Block fault is described below. (All data is in hexadecimal.)

Field	Value	Description
Long/Short	02	This fault contains up to 5 bytes of fault specific data.
Reference Address	FF0100	FF indicates that the reference address is not given. The fault applies to the entire block.
Rack	00	Main rack (rack 0).
Slot	04	Slot 4. In this configuration, slot 4 contains a Genius Bus Controller.
Bus	01	Bus 1 on the bus controller. When this byte is significant, the single channel bus controller always shows this field as a 1.
Block	1E	1E is 30 decimal. This fault was logged for the Genius device at serial bus address 30.
Point	FF7F	FF in the first (low byte) indicates that the point is not meaningful for this fault entry.
Fault Group	03	Loss of, or missing, I/O module fault.
Fault Action	02	Diagnostic fault.
Fault Category	02	Loss of block.
Fault Type	00	00 fault type in the Loss of Block category indicates no specific fault was given.
Fault Description	00	No fault description information is given.
Fault Specific Data	84	GBC reported a lost device. Only 1 byte of fault specific data is significant.

The Genius Bus Controller in slot 4 determined that the device at serial bus address 30 on bus 1 failed to send data in three consecutive bus scans. The bus controller then marked the device as lost and logged a fault in the I/O fault table. The fault type and fault description are not meaningful on this fault. The fault specific data contains a byte from the GBC echoing the fault.

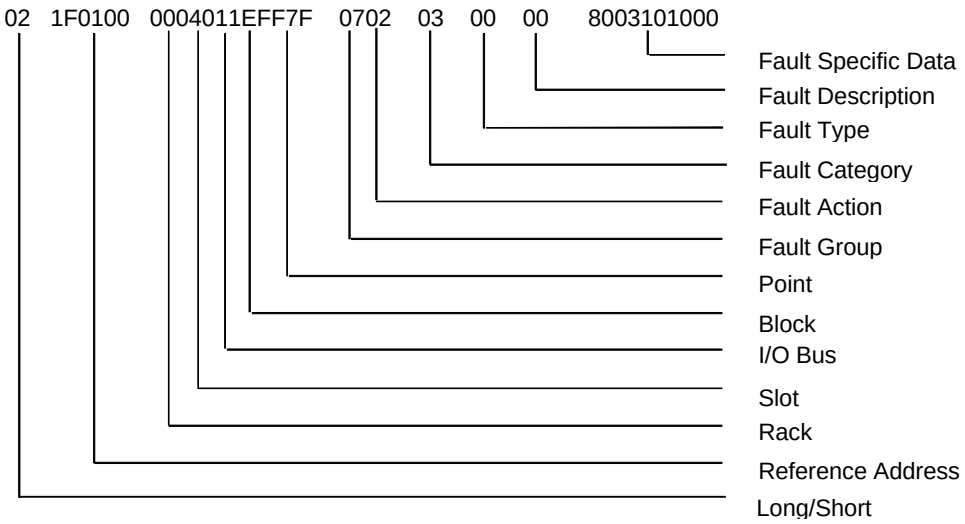


Addition of I/O Block Example

The Addition of I/O Block fault is described below. (All data is in hexadecimal.)

Field	Value	Description
Long/Short	02	This fault contains up to 5 bytes of fault specific data.
Reference Address	1F0100	1F (31 decimal) indicates that the block has a discrete grouped reference address. The software shows this by displaying a %QI in the reference address column. 0100 indicates this is address 0001 (hexadecimal data is displayed low byte first, then high byte) in the discrete grouped address space.
Rack	00	Main rack (rack 0).
Slot	04	Slot 4. In this configuration, slot 4 contains a Genius Bus Controller.
Bus	01	Bus 1 on the bus controller. When this byte is significant, the single channel bus controller always shows this field as a 1.
Block	1E	1E is 30 decimal. This fault was logged for the Genius device at serial bus address 30.
Point	FF7F	FF in the first (low byte) indicates that the point is not meaningful for this fault entry.
Fault Group	07	Addition of, or extra, I/O module fault.
Fault Action	02	Diagnostic fault.
Fault Category	03	Fault category 03 is an addition of block.
Fault Type	00	No fault type data occurs for an Addition of Block fault. The 00 is meaningless.
Fault Description	00	No fault description information is given.
Fault Specific Data	80	GBC reported an added device. Only 1 byte of fault specific data is significant.

The bus controller in slot 4 in the main rack received data from the device at serial bus address 30, indicating that the device was again on the bus. From the configuration table stored from your programming software to the PLC CPU, the CPU determined that the first point on the device was %QI0001. The fault type and fault description are not meaningful on this fault. The fault specific data contains a byte from the GBC echoing the fault.

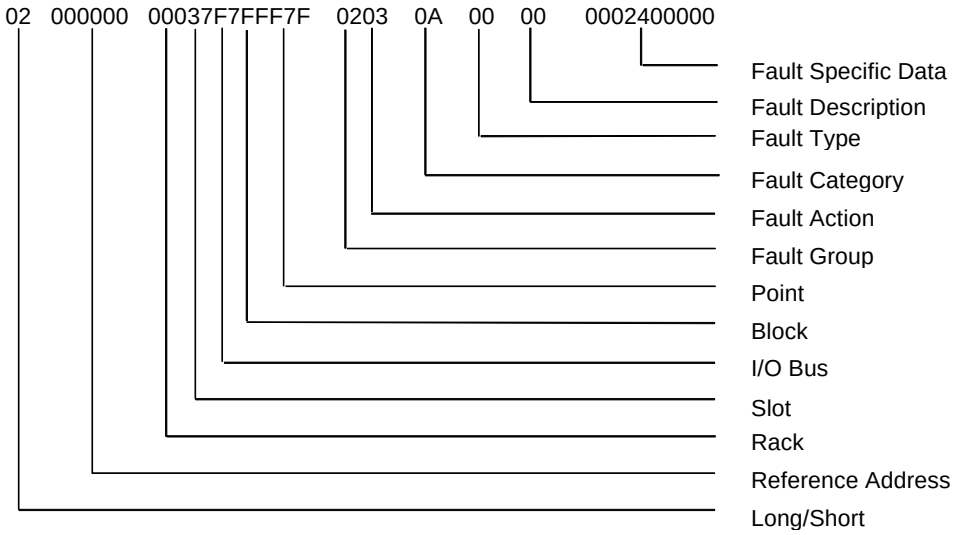

02 1F0100 0004011EFF7F 0702 03 00 00 8003101000
Fault Specific Data
Fault Description
Fault Type
Fault Category
Fault Action
Fault Group
Point
Block
I/O Bus
Slot
Rack
Reference Address
Long/Short

Loss of IOC (I/O Controller)

The Loss of IOC fault is explained below. (All values are in hexadecimal.)

Field	Value	Description
Long/Short	02	This fault contains up to 5 bytes of fault specific data.
Reference Address	000000	All zeros in this field indicates that the reference address is not meaningful for this fault.
Rack	00	Main rack (rack 0).
Slot	03	Slot 3. In this configuration, slot 3 contains a Genius Bus Controller.
Bus	7F	Since this is a single channel GBC, the bus number is not needed. 7F indicates that the bus number is not significant.
Block	7F	A block number of 7F hex indicates that the block is not meaningful for this fault.
Point	FF7F	FF in the first (low byte) indicates that the point is not meaningful for this fault entry.
Fault Group	02	Loss of, or missing, IOC fault.
Fault Action	03	Fatal fault.
Fault Category	0A	Fault Category 0A (10 decimal) is a Loss of, or Missing, IOC fault.
Fault Type	00	No fault type data occurs for a Loss of, or Missing, IOC fault.
Fault Description	00	No fault description information is given.
Fault Specific Data	00	There is no entry for a zero in fault specific data, so there is no additional information available on this instance of LOSS of IOC.

The PLC CPU detected a loss of, or missing, IOC and logged this fault. The fatal action indicates that the PLC CPU will not transition to RUN mode until the fault is cleared. Fault type and fault description are not meaningful for this fault; fault specific data may be meaningful.

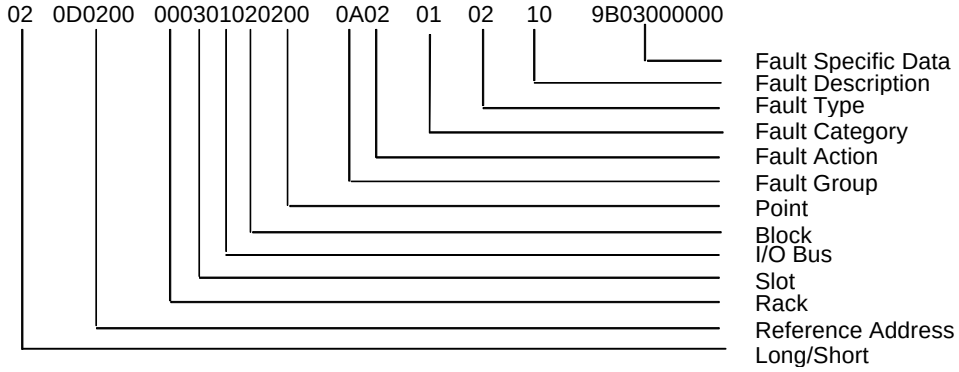


Circuit Fault

The Circuit Fault is explained below. (All values are in hexadecimal.)

Field	Value	Description
Long/Short	02	This fault contains up to 5 bytes of fault specific data.
Reference Address	0D0200	0D (13 decimal) indicates that the block has an analog grouped reference address. Your programming software shows this by displaying a %AQI in the reference address column. 0200 indicates this is address 0002 (hexadecimal data is displayed low byte first, then high byte) in the analog grouped address space.
Rack	00	Main rack (rack 0).
Slot	03	Slot 3. In this configuration, slot 3 contains a Genius Bus Controller.
Bus	01	Bus 1 on the bus controller. When this byte is significant, the single channel bus controller always shows this field as a 1.
Block	02	This fault was logged for the Genius device at serial bus address 02.
Point	0200	This fault was logged for point 2 on the device at serial bus address 02.
Fault Group	0A	0A hexadecimal is 10 decimal. I/O Module fault.
Fault Action	02	Diagnostic fault.
Fault Category	01	Fault Category 01 is a circuit fault.
Fault Type	02	Fault Type 02 under the Circuit Fault category is a circuit fault on an analog I/O channel.
Fault Description	10	A fault description of 10 hex under the analog circuit fault description is open wire.
Fault Specific Data	9B03	The GBC reported a circuit fault. Only the 9B byte is significant.

The Genius Bus Controller in slot 3 in the main rack received a circuit fault "Report Fault" message from the analog block located at serial bus address 2. The block reported an Open Wire fault on point 2. From the configuration tables stored from your programming software, the PLC CPU determined that the analog block at serial bus address 02 was mapped to both the %AI and %AQ user references and that point 2 corresponded to location 0002.



Appendix C Memory Allocation

Use this worksheet to determine the total number of bytes of memory used by %R, %AI, %AQ, %P, %L, and associated point faults, if used. Then, add all the totals together. To determine the amount of memory left for the user program, subtract the grand total from the number of bytes for the memory size being used.

Table C-1. Series 90-70 PLC Memory Allocation Worksheet

%R allocated = _____ # of %R used x 2 bytes (# of %R used is in 1K increments based on configuration)	_____
Note: When %P used is in the range 1 to 128, the %P allocated = 160 Otherwise, compute the number of %P allocated %P allocated = _____ {[(# of %P used) – 128] / 32 rounded up to whole integer} * 32 + 160; %P Block Memory Used = (%P allocated) * 2 + 79 bytes =	Totals _____
Note: When %L used is in the range 1 to 64, the %L allocated = 96 Otherwise, compute the number of %L allocated %L allocated = _____ {[(# of %L used) – 64] / 32 rounded up to whole integer} * 32 + 96; %L Block Memory Used = (%L allocated) * 2 + 79 bytes =	 _____

Note

%P or %L allocation for a program block cannot be changed when the programming software is online, unless no %P or %L is currently used in that block. To change the current %P or %L allocation, the programming software must be offline. The resulting program must then be stored to the PLC in Stop mode.

Table C-1. Series 90-70 PLC Memory Allocation Worksheet (cont'd)

%AI allocated (if point fault Disabled) = # %AI used x 2 (bytes)	+	
	=	_____
OR		
	+	OR
%AI allocated (if point fault Enabled) = # %AI used x 3 (bytes)	=	_____
%AQ allocated (if point fault Disabled) = # %AI used x 2 (bytes)	+	
	=	_____
OR		
	+	OR
%AQ allocated (if point fault Enabled) = # %AI used x 3 (bytes)	=	_____
If point fault is Enabled, add: _____ 128 bytes for 731/732 CPU OR add: _____ 512 bytes for 771/772 CPU OR add: _____ 3072 bytes for 781/782 CPU OR add: _____ 3072 bytes for 914–925 CPUs		_____
GRAND Total (bytes)	=	_____

CPU Memory Total Bytes =	_____	32K =	32768		
OR =	_____	64K =	65536		
OR =	_____	128K =	131072		
OR =	_____	256K =	262144		
OR =	_____	512K =	524288	=	_____
Subtract Grand Total (from above)				–	_____
User Program Memory Available (in bytes)				**=**	_____

Note

Although 915 and 925 CPUs have a total of one megabyte (1024K), both CPUs have the same 512K limit for a single user program as other CPUs. The additional 512K can be used for standalone C programs.

Appendix D Using Floating-Point Numbers

Floating-Point Numbers

Your programming software provides the ability to edit, display, store, and retrieve numbers with real values. Some functions operate on floating-point numbers. However, in order to use floating-point numbers, you must have a 732, 772, 782, 914 or higher CPU. Floating-point numbers are represented in decimal scientific notation, with a display of six significant digits.

Note

Use of floating-point numbers within a parameterized subroutine block (PSB) usually necessitates the use of NWORD parameters rather than WORD parameters.

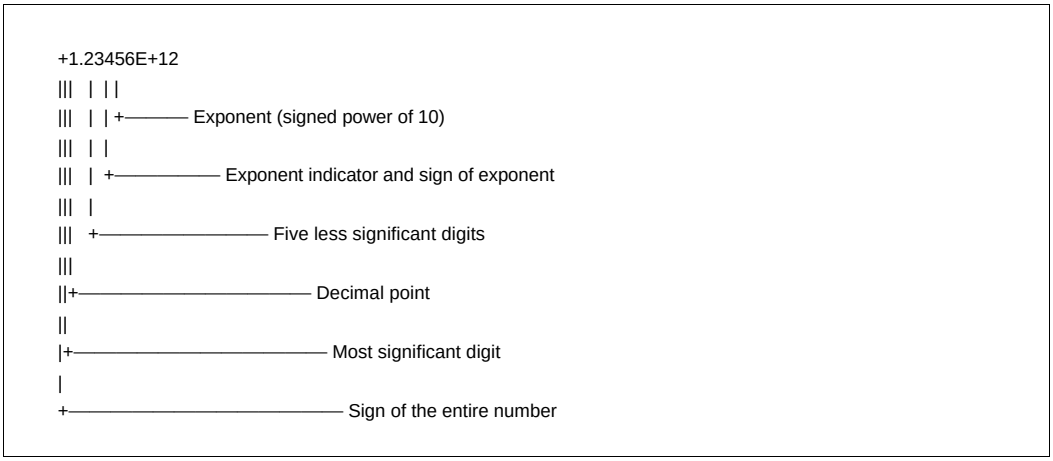
Note

In this manual, the terms floating-point and REAL are used interchangeably to describe the floating-point number display/entry feature of the software.

In the software, the following format is used. For numbers in the range 9999999999 to .0001, the display has no exponent and up to six or seven significant digits. For example:

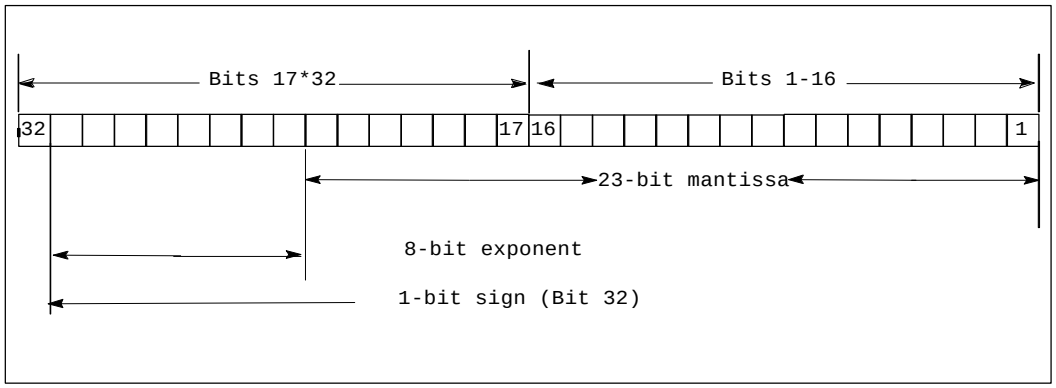
Entered	Displayed	Description
.000123456789	+.0001234567	Ten digits, six or seven significant.
–12.345e-2	–.1234500	Seven digits, six or seven significant.
1234	+1234.000	Seven digits, six or seven significant.

Outside the range listed above, only six significant digits are displayed and the display has the form:

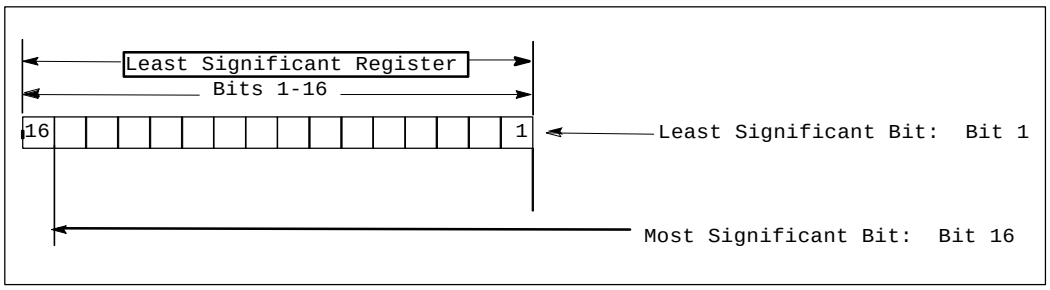


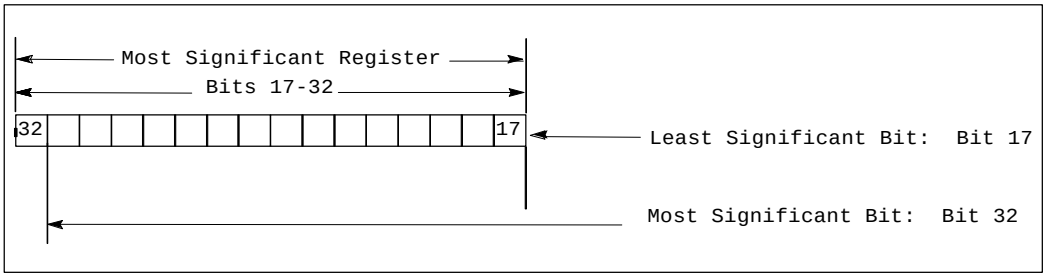
Internal Format of Floating-Point Numbers

Floating-point numbers are stored in single precision IEEE-standard format. This format requires 32 bits, which translates to two (adjacent) 16-bit PLC registers. The encoding of the bits is diagrammed below.



Register use by a single floating-point number is diagrammed below. In this diagram, if the floating-point number occupies registers R5 and R6, for example, then R5 is the least significant register and R6 is the most significant register.





Values of Floating-Point Numbers

Use the following table to calculate the value of a floating-point number from the binary number stored in two registers.

Exponent (e)	Mantissa (f)	Value of Floating Point Number
255	Non-zero	Not a valid number (NaN).
255	0	$-1^s * \infty$
0 < e < 255	Any value	$-1^s * 2^{e-127} * 1.f$
0	Non-zero	$-1^s * 2^{-126} * 0.f$
0	0	0

f = the mantissa. The mantissa is a binary fraction.
e = the exponent. The exponent is an integer E such that E+127 is the power of 2 by which the mantissa must be multiplied to yield the floating-point value.
s = the sign bit.
* = the multiplication operator.

For example, consider the floating-point number 12.5. The IEEE floating-point binary representation of the number is:

01000001 01001000 00000000 00000000

or 41480000 hex in hexadecimal form. The most significant bit (the sign bit) is zero (s=0). The next eight most significant bits are 10000010, or 130 decimal (e=130).

The mantissa is stored as a decimal binary number with the decimal point preceding the most significant of the 23 bits. Thus, the most significant bit in the mantissa is a multiple of 2^{-1}, the next most significant bit is a multiple of 2^{-2}, and so on to the least significant bit, which is a multiple of 2^{-23}. The final 23 bits (the mantissa) are:

1001000 00000000 00000000

The value of the mantissa, then, is .5625 (that is, $2^{-1} + 2^{-4}$).

Since e > 0 and e < 255, we use the third formula in the table above:

$$\begin{aligned} \text{number} &= -1^s * 2^{e-127} * 1.f \\ &= -1^0 * 2^{130-127} * 1.5625 \\ &= 1 * 2^3 * 1.5625 \\ &= 8 * 1.5625 \\ &= 12.5 \end{aligned}$$

Thus, you can see that the above binary representation is correct.

The range of numbers that can be stored in this format is from ± 1.401298E–45 to ± 3.402823E+38 and the number zero.

Entering and Displaying Floating-Point Numbers

In the mantissa, up to six or seven significant digits of precision may be entered and stored; however, the software will display only the first six of these digits. A positive or negative sign may precede the mantissa. If no sign is entered, the floating-point number is assumed to be positive.

If an exponent is entered, it must be preceded by the letter "E" or "e" and the mantissa must contain a decimal point to avoid mistaking it for a hexadecimal number. The exponent may be preceded by a sign, but if none is provided, it is assumed to be positive. If no exponent is entered, it is assumed to be zero. No spaces are allowed in a floating-point number.

To provide ease-of-use, several formats are accepted in both command line and field data entry. These formats include an integer, a decimal number, or a decimal number followed by an exponent. These numbers are converted to a standard form for display once the user has entered the data and pressed the **Enter** key.

Examples of valid floating-point number entries and their normalized display are shown below.

Entered	Displayed
250	+250,0000
+4	+4.000000
–2383019	–2383019.
34.	+34.00000
–.0036209	–.003620900
12.E+9	+1.20000E+10
–.0004E–11	–4.00000E–15
731.0388	+731.0388
99.20003e–29	+9.92000E–28

Examples of invalid floating-point number entries are shown below.

Invalid Entry	Explanation
–433E23	Missing decimal point.
10e-19	Missing decimal point.
1 0.e19	The mantissa cannot contain spaces between digits or characters. This is accepted as 10.e0, and an error message is displayed.
4.1e1 9	The exponent cannot contain spaces between digits or characters. This is accepted as 4.1e0, and an error message is displayed.

Errors in Floating-Point Numbers and Operations

Overflow occurs when a number greater than 3.402823E+38 or less than –3.402823E+38 is generated by a REAL function. When this occurs, the Enable Out output of the function is set Off; and the result is set to positive infinity (for a number greater than 3.402823E+38) or negative infinity (for a number less than –3.402823E+38). You can determine where this occurs by testing the sense of the Enable Out output.

POS_INF	= 7F800000h	– IEEE positive infinity representation in hex
NEG_INF	= FF800000h	– IEEE negative infinity representation in hex

If the infinities produced by overflow are used as operands to other REAL functions, they may cause an undefined result. This undefined result is referred to as an NaN (Not a Number). For example, the result of adding positive infinity to negative infinity is undefined. When the ADD_REAL function is invoked with positive infinity and negative infinity as its operands, it produces an NaN for its result.

Each REAL function capable of producing an NaN produces a specialized NaN that identifies the function.

NaN_ADD	= 7F81FFFFh	– Real addition error value in hex
NaN_SUB	= 7F81FFFFh	– Real subtraction error value in hex
NaN_MUL	= 7F82FFFFh	– Real multiplication error value in hex
NaN_DIV	= 7F83FFFFh	– Real division error value in hex
NaN_SQRT	= 7F84FFFFh	– Real square root error value in hex
NaN_LOG	= 7F85FFFFh	– Real logarithm error value in hex
NaN_POW0	= 7F86FFFFh	– Real exponent error value in hex
NaN_SIN	= 7F87FFFFh	– Real sine error value in hex
NaN_COS	= 7F88FFFFh	– Real cosine error value in hex
NaN_TAN	= 7F89FFFFh	– Real tangent error value in hex
NaN_ASIN	= 7F8AFFFFh	– Real inverse sine error value in hex
NaN_ACOS	= 7F8BFFFFh	– Real inverse cosine error value in hex
NaN_BCD	= 7F8CFFFFh	– BCD-4 to real error
REAL_INDEF	= FFC00000	– Real indefinite, divide 0 by 0 error

When an NaN result is fed into another function, it passes through to the result. For example, if an NaN_ADD is the first operand to the SUB_REAL function, the result of the SUB_REAL is NaN_ADD. If both operands to a function are NaNs, the first operand will pass through. Because of this feature of propagating NaNs through functions, you can identify the function where the NaN originated.

Note

For NaN, the Enable Out output is Off (not energized).

%

A

B

C

D

E

F

G

N

O

P

R

S

T

U

W