Modicon
984 Hot Standby System
Programming Manual

GM-S911-002 Rev. D

January, 1993

MODICON, Inc., Industrial Automation Systems
One High Street
North Andover, Massachusetts 01845

Preface

The data and illustrations found in this book are not binding. We reserve the right to modify our products in line with our policy of continuous product development. The information in this document is subject to change without notice and should not be construed as a commitment by Modicon, Inc., Industrial Automation Systems.

Modicon, Inc. assumes no responsibility for any errors that may appear in this document. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify by using the form on the last page of this publication.

Caution All pertinent state, regional, and local safety regulations must be observed when installing and using this product.
For reasons of safety and to ensure compliance with documented system data, repairs to components should be performed only by the manufacturer.

The following are trademarks of Modicon, Inc.:

Modicon	984	S911
Modbus	Modbus Plus	

Table of Contents

Figures

Tables

Chapter 1
Introduction

- What a Hot Standby System Does
- Configuring Hot Standby Ladder Logic
- Hot Standby System Software Capabilities

What a Hot Standby System Does

The Hot Standby concept provides fault-tolerant, high-availability programmable control. The system consists of two identical 984 Programmable Controllers configured so that a failure in one transfers system control duties to the other in less than one scan.

Primary and Standby Control

One controller—the *primary*—controls all aspects of the remote 800 Series I/O drops. It constantly scans and solves all the configured segments of ladder logic. The primary controller updates the *standby* controller with the most recent values of the inputs and internal states. Only state memory is transmitted.

Primary and standby states are switchable; either controller in a Hot Standby system can be put in the primary state, but the other must be in the standby state. Remote I/O is always controlled by the primary controller.

The Hot Standby Function Block

The key software component of a Hot Standby System is the loadable hot standby function block—HSBY. The HSBY function block in the primary controller automatically captures a *snapshot* of the 984 state table. This snapshot is transmitted from the primary to the standby through S911 Hot Standby option modules in each controller. This sharing of process information enables the standby to assume control with up-to-date system data in the event of a failure in the primary system.

The HSBY function block lets you reserve several registers for Hot Standby activities. Included in these reseved register areas are a *command* register and a *status* register, which provide a software interface to the Hot Standby System. The command register lets you assume control of the system and override the keyswitch control mechanism. The status register provides you with current system status.

S911 Hot Standby Modules

Each mainframe contains an S911 Hot Standby Module that monitors its own mainframe and communicates with the S911 in the other mainframe.

The timing diagram in Figure 1 shows how the HSBY function blocks in the two CPUs interact with the two S911s to transfer the state table from the primary to the standby controller.

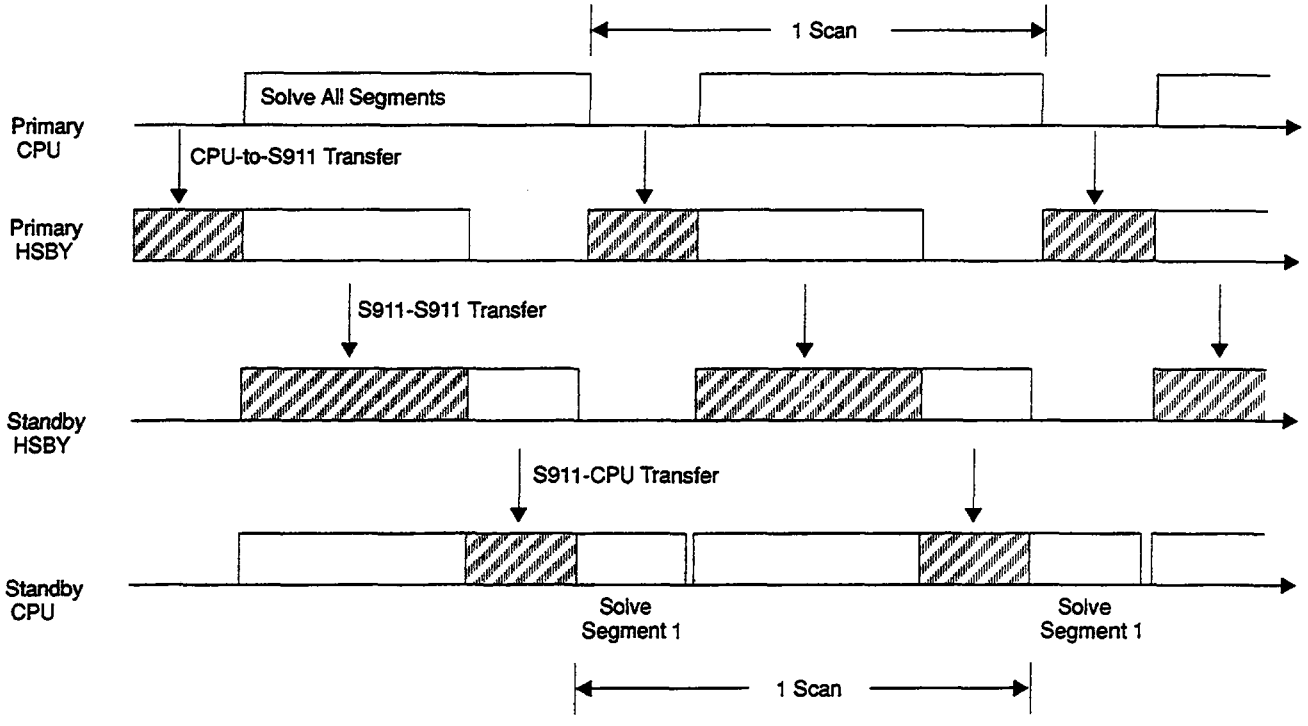

Figure 1 Hot Standby Scan Time Diagram

Hot Standby System Software Capabilities

For the Hot Standby System to function properly, some basic programming tasks must be performed: identical ladder logic must be installed in the primary and standby controllers, all ladder logic must be correct, and an HSBY function block must be loaded into both controllers. Both the primary and the standby units should be configured with a minimum of 1024 words of 0*x* ... 4*x* references, and the primary unit should contain enough user logic in segments 2 and above to guarantee that the standby controller can complete its tasks by the time the primary returns to begin another scan. An insufficient amount of configured references and user logic can cause system switchover to occur.

Application Control

Application control is through an HSBY function block that lets you influence operations by

- Designating a series of registers in the primary controller—the *nontransfer area*—that do not get transferred to the standby controller
- Specifying whether mismatched logic will fail the standby controller or simply report the mismatch
- Specifying whether the Modbus port addresses are offset between the primary and standby controllers—this mechanism provides an easy scheme for integrating a Hot Standby System into a Modbus network
- Forcing a switchover because of a user-defined fault condition in the primary controller

Manual Control

You can control a 984 Hot Standby System manually with the RUN/OFFLINE keyswitches on the S911 modules in your primary and standby controllers. If you switch the primary controller to OFFLINE, the standby controller changes to primary mode and functions as a standalone controller. If you switch the standby controller OFFLINE, the primary controller functions as a standalone.

You can use these keyswitches to force manual control for maintenance or troubleshooting.

The Effects of Switchover

Most operations in progress during a switchover will be completed or aborted without the unit's being aware that the switchover has occurred. The system that switches to primary mode maintains control over all the operations currently in progress and retains control unless or until another switchover occurs.

Modbus and Modbus Plus transactions or J892 ASCII operations in progress during a switchover typically return an error condition, then complete the transaction on the subsequent retry.

Keep in mind that the default drop holdup time in the Traffic Cop is 300 ms. User logic programs with long scan times and/or configurations with *reset Watchdog Timer* as part of the Segment Scheduler must insure that drop holdup is sufficient, particularly in the event of switchover.

Caution **It is imperative that functions controlled by your logic monitor the error outputs and perform retries when an error occurs.**

Caution **Up to two Modbus Plus messages may be affected during switchover due to protocol restraints. A timeout error or routing failure may be detected by the originator. User logic should be programmed to perform retries on critical applications.**

Note A Modbus Plus *No Response* status error cannot be transferred through a BM85 from Modbus Plus to Modbus. A Modbus *No Response* condition could be detected by the Modbus originator.

Configuring Hot Standby Ladder Logic

All ladder logic for *Hot Standby* functions should be in segment 1. Network 1 of segment 1 is reserved exclusively for the HSBY function block and ladder logic directly associated with it.

Segment 1

When your Hot Standby System is running, the primary controller scans all segments while the standby controller scans only segment 1 of the configured ladder logic program. This has three very important implications with respect to the way you configure the system logic:

- You must program all ladder logic specific to Hot Standby functions in segment 1
- You must not program I/O control logic in segment 1
- You must not schedule any I/O drops in segment 1

The standby controller in a Hot Standby System must never execute I/O control logic.

Caution To help protect against damage to application I/O devices through unexpected system actions, do not reschedule segment 1 via the Segment Scheduler.

Segment 1 may contain the ladder logic for diagnostics and optional Hot Standby functions—e.g., time-of-day clock updates or optional system control implementation.

As a minimum, segment 1 in a Hot Standby System must contain the HSBY function block and its associated ladder logic.

Network 1 of Segment 1

Load the HSBY function block to the logic programs in both the primary and standby controllers—in network 1 of segment 1. Then program input and output lines to and from the function block. The two HSBY blocks and all their associated ladder logic inputs must be *identical in both programmable controllers.*