Core Architecture Group

Autotyping ATA Disk Drives

Version 0.9

July 26, 1995

Technical Editor:
Curtis E. Stevens
Phoenix Technologies
2575 M^{c}Cabe Way
Irvine, Ca. 92714
Phone: (714) 440-8000
Fax: (714) 440-8300
Curtis_Stevens@PTLTD.COM

Phoenix Technologies Ltd.

Phoenix Technologies LTD
2757 M^{c}Cabe Way
Ivine, Ca. 92714
Phone: (714) 440-8000
Fax: (714) 440-8300

Revision History				
Rev	**Date**	**File Name**	**Author**	**Description**
0.5	June 29, 1995	AUTOTYPE.DOC	Curtis E. Stevens Mike Coady	Initial Release
0.9	July 26, 1995	AUTO.DOC	Curtis E. Stevens	Dubayification

PRELIMINARY

Phoenix Technologies LTD
2757 M^{c}Cabe Way
Ivine, Ca. 92714

Phone: (714) 440-8000
Fax: (714) 440-8300

1. Overview

The move by PC manufacturers to minimize the use of the Setup interface has caused Phoenix to change the way PhoenixBIOS 4.0 configures Hard Drives. In the past, a user would enter Setup, select a menu for a specific drive, and then autotype the selected drive. Now systems are autotyping drives as a part of POST. This eliminates the need for a user to enter Setup, and it also creates a new environment for autotyping drives. This paper describes a method for reliably autotyping drives as a part of the Power On Self Test (POST).

2. Scope

The reader must be familiar with basic ATA device operation and protocols. Familiarity with ATA-2 (X3T10 948D) is not mandatory but will assist the reader in understanding the concepts presented here. This paper also assumes the reader is familiar with basic PC/ATA architectures.

3. Introduction

The PC industry is moving away from making systems dependent on the BIOS Setup utility. This has caused Phoenix to make several changes to its POST activities. The Hard Drive Subsystem is one of the components affected by these changes. In the past, a user would enter Setup and autotype his hard drive. This method of operation makes the following assumptions:

- The user knows which drive he is attempting to autotype.
- The drive is present.
- The drive is in operating condition.

The Setup interface is very forgiving. If the drive failed to autotype the first time, the user often would assume operator error or some other error and simply press the autotype key again. No harm done. If the user attempted to autotype a drive that was not present, then any resulting failures can be blamed on the user.

Autotyping during POST, however, cannot be forgiving and must resolve all failure conditions. Failure to autotype the first time is a catastrophic failure. It is not acceptable for the system to lose a hard drive that is currently connected.

A second problem comes from the overall timing requirements of POST. Many customers require POST to take less than 20 or in some cases 15 seconds. Normal ATA protocols allow drives 30 seconds for internal initialization. This means that POST code can not be sitting around on empty channels waiting to see if a drive is going to appear.

The remainder of this document is broken into two sections:

- A basic description of autotyping, which addresses the problems described above
- A set of flow charts which describe a possible implementation

4. Autotyping

Autotyping is comprised of 3 basic components:

- Empty-channel detection
- Detecting number of drives present
- Fetching the configuration information.

Empty-channel detection reports when no devices are attached to a channel. This function allows the BIOS to quickly move on to the next channel and scan for drives.

Detecting number of drives present reports how many drives are connected to a specific channel.

Fetching the configuration information reports the geometry and other useful information about how the drive operates.

4.1 Empty Channel Detection

The ATA-2 specification states that, when the BUSY bit is asserted, all other bits and all other ATA registers are undefined and should not be accessed. Further, ATA-2 states that drives can wait up to 31 seconds on a hard or a soft reset before they clear the BUSY bit. This means that any software which complies with the spec must wait 31 seconds if the BUSY bit is set before it can determine that no device is attached to the system. This 31 second delay is unacceptable during normal POST operation.

PhoenixBIOS 4.0 performs the following tests for detecting an empty channel. All of these tests involve looking at the ATA status port

Phoenix Technologies LTD
2757 M^{c}Cabe Way
Ivine, Ca. 92714

Phone: (714) 440-8000
Fax: (714) 440-8300

which is at 1F7h on the primary channel. **This empty channel procedure is only invoked when BUSY=1**.

1. Check for FFh. Several ATA chip manufacturers place pull-up resistors on the ATA data-lines. This technique (Don't use "this" by itself as a resumptive modifyer. The word following "this" has to sum up what you have just described. -bd) has the effect of flagging the port as BUSY and would normally cause a 31-second delay in determining when a channel is empty. However, it is a common practice in the BIOS industry to assume that the channel is empty if a status of FFh is returned on a specific channel. Today's drive manufacturing standards avoid returning a status of FFh because of this common practice by BIOS manufacturers.
2. If the status is not FFh then things get a little more tricky. The next level of testing involves debouncing some of the other status bits. In particular, if over a period of time, such as 20ms, all the status reads logically or points to an FFh, treat the port as if it returned FFh. If a value of FFh is not achieved, then we will assume that a device is not present.
3. If one of the above listed tests determines a drive is not present, then one more test must be performed. This second test is an added insurance step which will give us the ability to return a present drive to active service if, in fact, it returns a status of FFh. The ATA cylinder registers are read/write capable. ATA defines that, when a drive is busy, the host does not have access to the registers. Many drives allow writing of these registers even though the ATA-2 specification expressly forbids this operation. In the unlikely event that the BIOS chooses to flag a present drive as not being present, reading and writing several values to the cylinder registers can expose a present drive which has returned a status of FFh.

4.2 Detecting Number of Drives Present

The ATA-2 specification allows the host to issue an **Execute Drive Diagnostics** any time after the BUSY bit is cleared, even if the drive has not set the READY status. PhoenixBIOS 4.0 issues this command immediately when the BUSY bit is cleared. The result of this command is a status which indicates if 1 or 2 drives are present on the cable. This command can take up to 5 seconds to execute, but the information returned justifies the time.

If the host has hardware for pulling down data bit 7 (the BUSY bit), and no drives are attached, the system can detect that no drives are present within 400ns. The BIOS issues an Execute Drive Diagnostics and monitors the BUSY bit. If the bit remains clear (is set to 0) for more than 400ns, then no devices are present.

4.3 Fetching the Configuration Information

After the system has determined the locations of all the ATA devices, it will interrogate each drive to determine its capabilities. The following information is extracted from the ID Drive data returned by the drive:

- Geometry (CHS configuration)
- LBA capability
- Maximum HDD Multiple (Block PIO) setting.
- Fast PIO capabilities
- Fast DMA capabilities

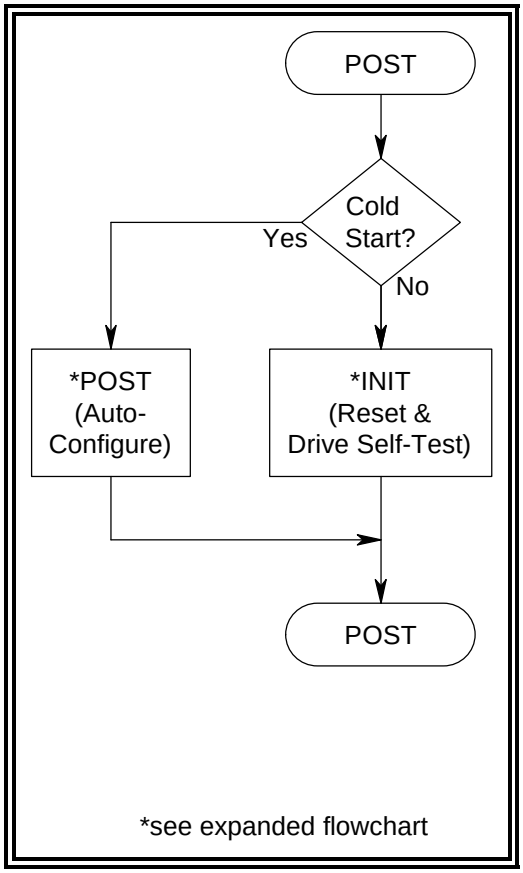
Next, the BIOS compares the system capability with the drive capability and selects the ***fastest*** mode of operation that is supported by both the drive and the motherboard.

5. Design Specification

The remainder of this document presents flowcharts that describe a possible implementation for autotyping during POST. The *autotyping* portions of these flowcharts apply to both Setupand POST time operation.

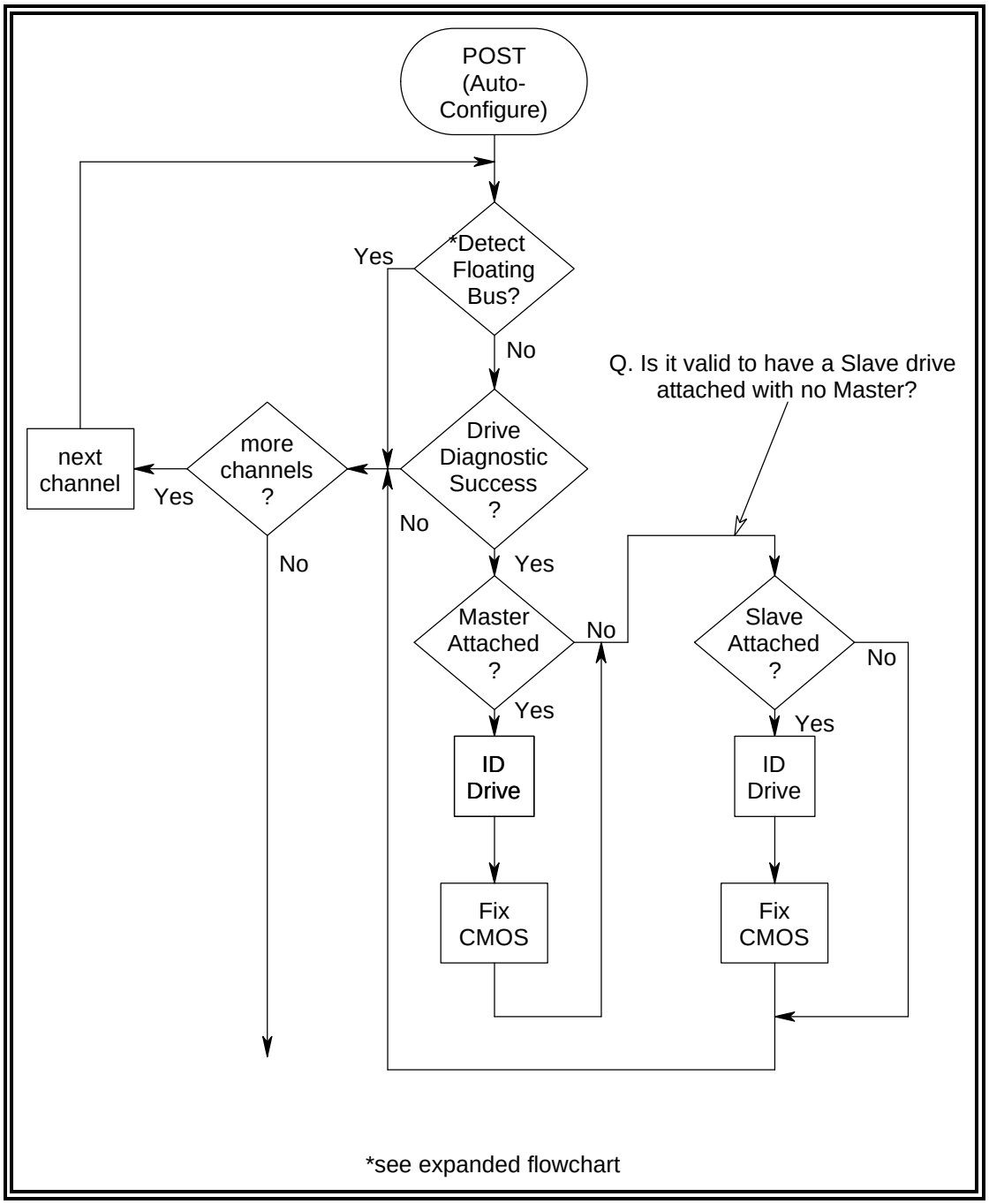
Phoenix Technologies LTD
2757 M^cCabe Way
Ivine, Ca. 92714

Phone: (714) 440-8000
Fax: (714) 440-8300



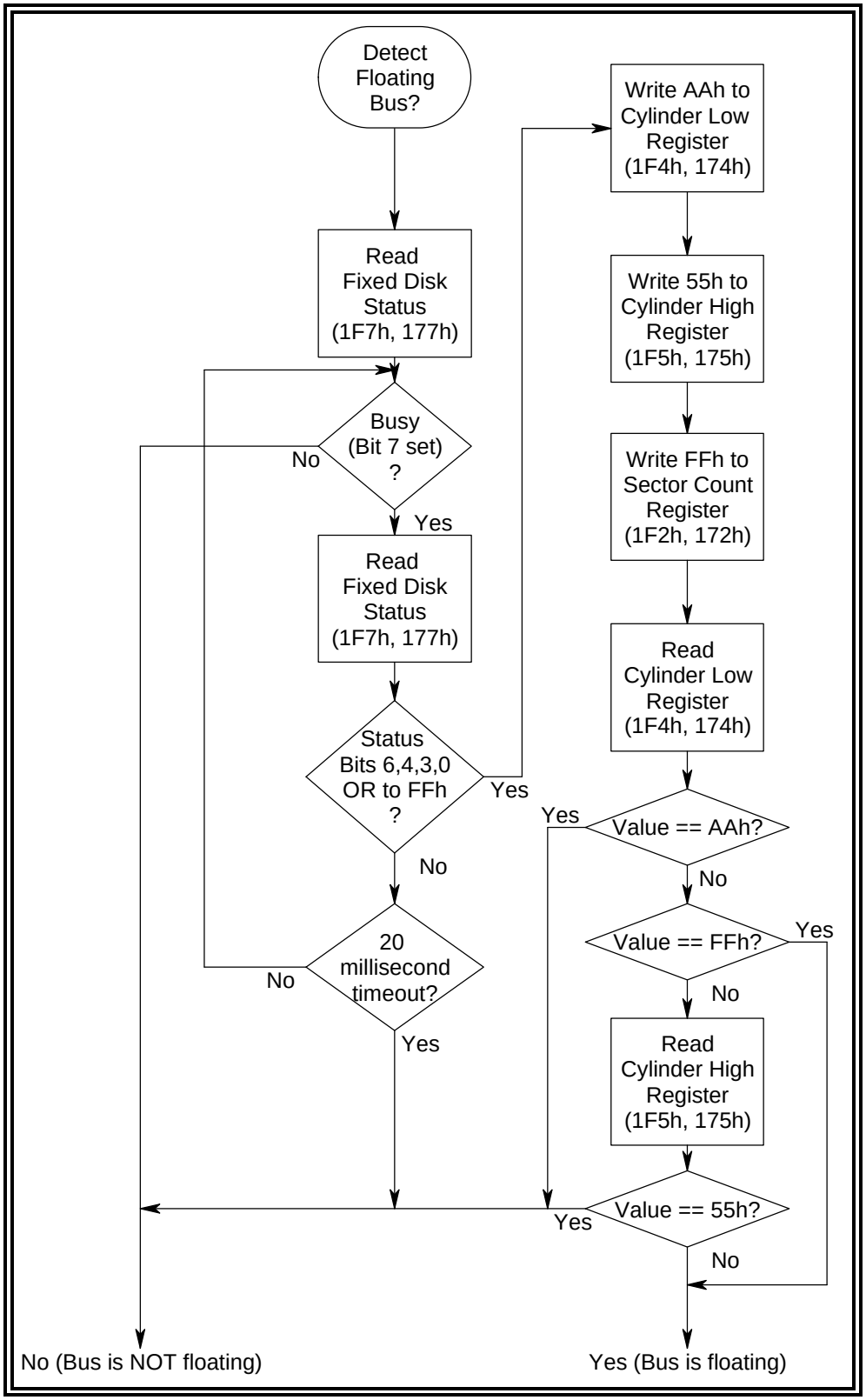
Phoenix Technologies LTD
2757 M^{c}Cabe Way
Ivine, Ca. 92714

Phone: (714) 440-8000
Fax: (714) 440-8300



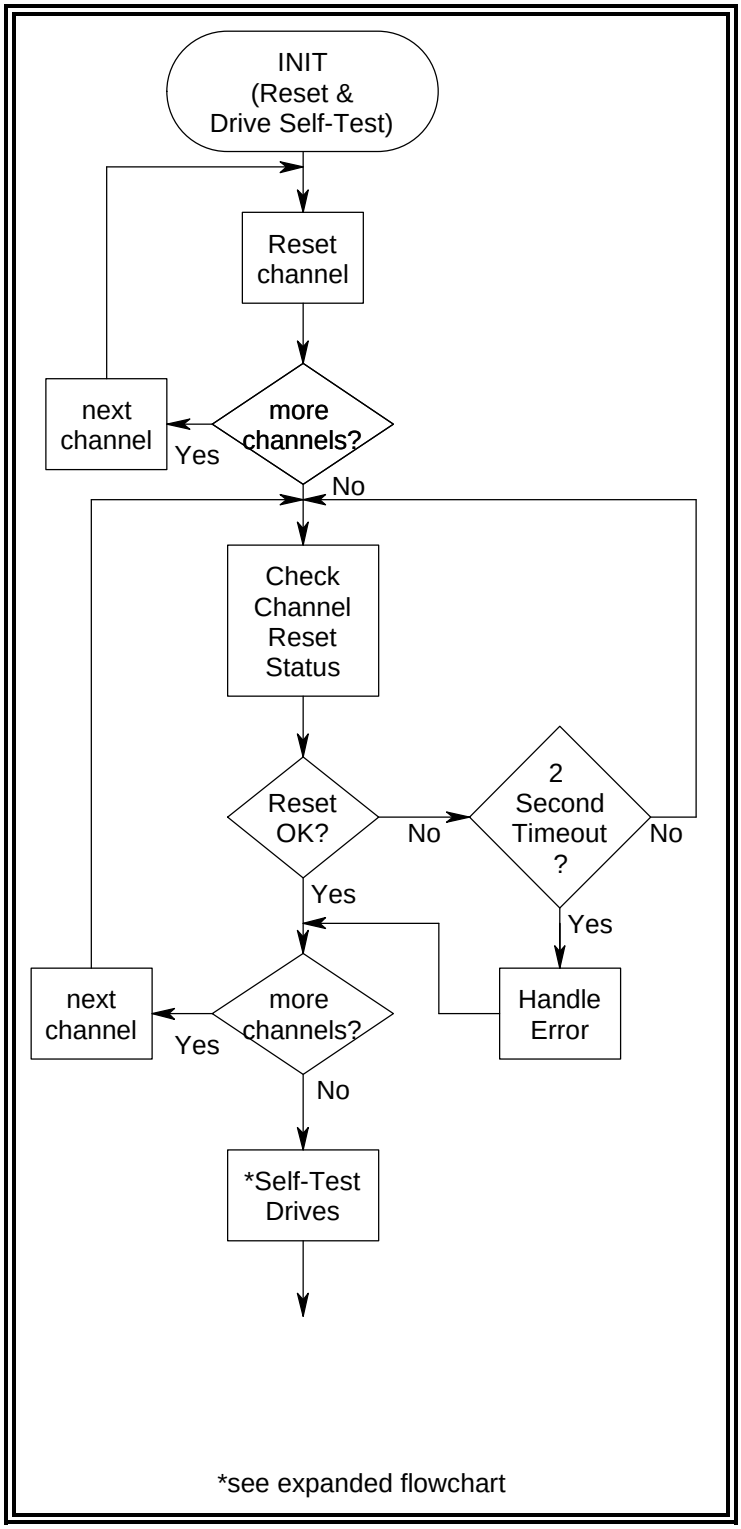
Phoenix Technologies LTD
2757 M^{c}Cabe Way
Ivine, Ca. 92714

Phone: (714) 440-8000
Fax: (714) 440-8300



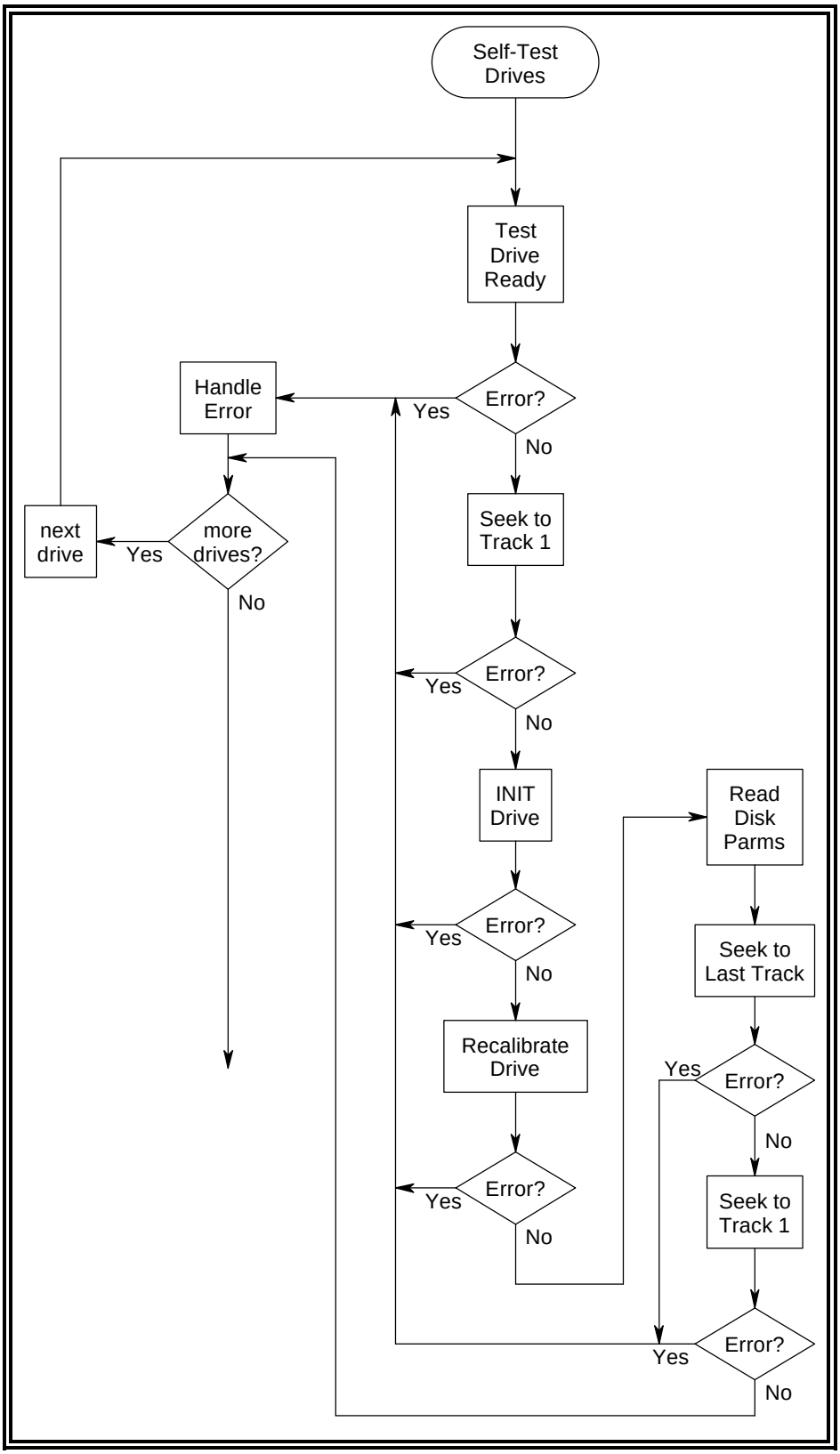
Phoenix Technologies LTD
2757 M^cCabe Way
Ivine, Ca. 92714

Phone: (714) 440-8000
Fax: (714) 440-8300



Phoenix Technologies LTD
2757 M^cCabe Way
Ivine, Ca. 92714

Phone: (714) 440-8000
Fax: (714) 440-8300



Phoenix Technologies LTD
2757 M^{c}Cabe Way
Ivine, Ca. 92714

Phone: (714) 440-8000
Fax: (714) 440-8300