KONAN™
in control

SMC-100/200
CP/M 2.X Support Manual

R-SMC-007-A

AUGUST 13, 1982

KONAN CORPORATION
1448 North 27th Avenue, Phoenix, Arizona 85009
(602) 269-2649, TWX/TELEX 9109511552

I Introduction

The purpose of this material is to help CP/M 2.X users interface to Konan's SMC-100 and SMC-200. It is assumed that the user is operational with CP/M 2.X on 8 inch CP/M compatible floppies. The components of this support package are as follows:

I Introduction

II System Overview

III Disk Diagnostics

IV Mapping/Formatting

V BIOS Driver

VI Boot Facilities

VII Fast Copy Program (MAPCOPY)

VIII Integration Guide

The CMD drive was selected as an example. This is due to both it's popularity and it's unique requirement to follow volume changes with a seek. Code written for the CMD is easily converted to non-CMD but the opposite is not true.

II System Overview

KONAN's BIOS is shipped configured for a 96 or 32 megabyte CMD drive. Due to CP/M 2.X's limitation of 8 megabytes per logical unit the CMD has been divided into 2 logical units per surface. The removable platter is drives A & B,(A=tracks Ø-4Ø1, B=4Ø2-8Ø3). CP/M will access the fixed surface as units C & D, C=tracks Ø-4Ø1, D=4Ø2-8Ø3) for 32 megabyte drives and as C through L for 96 megabyte drives.

The driver supplied will also handle an 8 inch floppy on a Tarbell controller. KONAN does not support the floppy drivers and makes no claims concerning their reliability or efficiency.

KONAN's hard disk driver has the following features:

* 512 byte sectors: Greatly increases storage over 128 byte formats.

* Integral driver: After boot the driver ROM is phantomed (SMC-1ØØ), allowing the use of all 64K.

* Mapping: Allows efficient foolproof mapping of bad media locations.

Programs currently supplied in KONAN's CP/M* 2.X Support Package include:

1. MAPBIOS3
   A bios for the SML-100/200. Included are drivers for Tarbell floppy controller.

2. CP/MNDMA
   The boot ROM source. (Applicable only to the SMC-100).

3. KONGEN
   A program to generate a bootable system on the hard disk. (Parallels CP/M* Sysgen).

4. DIAGNOSTICS
   IORAMS - Controller RAM test
   DMARAM - DMA test (SMC-100 only)
   LACEDIAG - Media test

5. MAPCOPY3
   Fast copy, 16 megabytes in 3 minutes, for binary backup. (For SMC-100 only).

6. MAPPER3
   Fast format program. Allows user to view, alter and create bad track map.

7. MEMMOVE
   Program used as part of the boot.

NOTE: All programs provided in source code.

III Diagnostics:

The following diagnostics should be run prior to attempting to integrate CP/M* onto your disk.

1. IORAMS
2. DMARAM (if applicable)
3. LACEDIAG

The CP/M Support Package does not normally use DMA. Failure of the DMARAM test will not prevent using this support package. If the user has a SMC-100 and DMARAM runs, then DMA can be equated TRUE in LACEDIAG and MAPBIOS3. MAPCOPY3 requires DMA in order to function. LACEDIAG IS A DESTRUCTIVE TEST. After running LACEDIAG the unit will require reformatting.

To test a system with data on it, first write protect your drive and then proceed as follows.

1. Run IORAMS
2. Run DMARAM (if applicable)
3. Run MAPPER3 (selection "3" then "5")

This will verify the S-100 and SMD interfaces, test the media, and verify all controller functions except writing. If disk has a removable media a "scratch" pack should be inserted and LACEDIAG run.

IORAMS

This test checks the sector buffer using I.O. commands, (non-DMA). It verifies the following:

1. Board selection address
2. On board RAMs
3. RAM counters

DMARAM

This test is similar to IORAMs except it uses DMA to transfer data from the SMC-100 to the host.

After starting the test a brief description of the test is displayed. Press any key to get the first of two questions. The questions are:

```
STARTING MEMORY PAGE: __00
ENDING MEMORY PAGE:   __00
```

The lowest possible starting memory page is 05. This causes the memory beginning at hex address 0500 to be tested. The ending page must be selected to not overlap the BIOS. When first running the test a low value, typically 10, is often used. These entries would test memory from 0500 thru 1000 hex.

LACEDIAG

This test serves two purposes:

1. It test the disk subsystem independently of the BIOS or Boot ROM
2. It aids in the location of Media flaws.

IMPORTANT! THIS TEST DESTROYS THE DATA AND MEDIA MAP ON THE DRIVES(S) BEING TESTED.

After running the Diagnostic/Media test the drive must be reformatted (and mapped if desired).

Disk Diagnostic/Media Test Abstract

Upon entering the Konan Media/Diagnostic test the following message is printed:

KONAN MEDIA TEST REVISION N DD-DD-DD where N is the revision and D is the last date updated.
DMA DISABLED BASE OF I.O. PORTS IS 90 SECTOR SIZE = 512
The operator is then prompted as follows:

UNIT? Enter a one (1) digit number corresponding to the unit to test. (Normally 0)

DRIVE TYPE? (C=CMD M=MMD)? Enter C for CMD (ie. Phoenix), M for all others.

CRCTEST? (N=NO Y=YES)? If yes, CRC is force to be bad. If following read fails to detect bad CRC an error is printed. Normally enter N.

BYTE FOR BYTE COMPARE?? Verifies each byte of data in addition to the CRC and header tests. Greatly slows the test. Normally enter N.

TEST READ DATA AT ALL OFFSETS? Causes the strobe early/late and offset forward/backward to be used on each read pass. Aids in finding marginal data. Greatly slows the test. Normally enter N.

FIRST TRACK? Decimal address of the first track to be tested.

LAST TRACK? Decimal address of the last track to be tested.

FIRST HEAD? Address of first head to test. With CMD, zero is for removable, 1 for first fixed, 2 for next, etc.. For MMD, zero = zero, 1 = 1, etc..

LAST HEAD? Address of last head to test.

INTERLACE PATTERN? Interlace pattern. Allows speed of diagnostic to be matched to system/disk.

LAST SECTOR? Last sector to test. Enter Ø to test only one sector per track, 35 to test all 36 sectors.

The operator may interrupt the test at any time by pressing a Control C, this will print the following message:

BREAK TYPE D FOR DOS, R TO RESTART AND CR TO PRINT STATUS AND CONTINUE

Typing D will cause a return to CP/M via a JMP Ø. R will restart the test, and carriage return will print the status as follows:

FULL PASSES=Ø9 R,W PASSES-ØØ37 ERRORS-ØØØØØØ
SECTORS READ=ØØØØE8Ø8 ØØØ2 Ø ØØ

Where FULL PASSES equal the number of times the drive has written and read all selected sectors with the following patterns: All zeroes, All ones, Ripple up, Ripple down, floating zero and floating one. One each read/write pass the base for the ripple and floating patterns changes.

R,W PASSES is a hex number equal to the number of read write passes through the selected portion of the disk.

ERRORS is a total of all errors occurring since the test was started, in hex.

SECTORS READ is the hex number of sectors read.

The last line is the decimal address where the break occured. The first four digits are track, then sector and finally head.

DATA ERRORS PRINT AS FOLLOWS:

```
Decimal            Hex
Track              Track,Head,Sector
   Head            in the form Good/Bad
      Sector
0005  1 33         0005/FFFF 01/FF 21/FF L,BD
0005  1 33         0005/FFFF 01/FF 21/FF L,BD
0005  1 02         0005/FFFF 01/00 02/05 L,GD
0005  1 02         0005/0005 01/01 02/ID L,GD,R
PASS  0002         1 14 0002/FFFF 01/FF 0E/FF N,BD
0002  1 14         0002/FFFF 01/FF 0E/FF N,BD
0002  1 18         0002/FFFF 01/FF 12/FF N,BD
TRCK  H SC           TRACK  HEAD  SCTR  TYPE  WORD,GOOD/BAD
0002  1 18         0002/FFFF 01/FF 12/FF N,BD
0002  1 22         0002/FFFF 01/FF 16/FF N,BD
0002  1 22         0002/FFFF 01/FF 16/FF N,BD
0002  1 26         0002/FFFF 01/FF 1A/FF N,BD
0002  1 26         0002/0002 01/01 1A/0E N,GD,R <RETRY, if retry is
                                                  always bad,error is
                                                  usually a write error
PASS  0005         1 00 FFFA/FFFA FE/FE FF/FF E,BD
0005  1 00         FFFA/FFFA FE/FE FF/FF E,BD
0005  1 04         FFFA/FFFA FE/FE FB/FB E,BD
0005  1 04         FFFA/FFFA FE/FE FB/FB E,BD
0005  1 08         FFFA/FFFA FE/FE F7/F7 E,BD
TRCK  H SC           TRACK  HEAD  SCTR  TYPE  WORD,GOOD/BAD
0005  1 08         FFFA/FFFA FE/FE F7/F7 E,BD  013,  00/08 <word 13 is bad
0005  1 12         FFFA/FFFA FE/FE F3/F3 E,BD  013,  00/08  it should have
                                                             been 00,was 08
CRC                                            If bad data, printed in
BD=BAD                                         this area.  If this is
GD=GOOD                                        blank then data was OK.
```

NON-DATA ERRORS

Non-data errors are also displayed. After printing them the status also prints. These errors are:

SELECT ERROR: (could not select unit)
TIMEOUT ERROR: (done loop timed out)
WRITE FAULT ERROR
SEEK ERROR
HEAD SELECT ERROR

IV MAPPING/FORMATTING

Prior to using CP/M on the SMC-100, the drive should be formatted. The Format program formats one surface at a time. Please note that each surface is two logical units! The special mapping format program also performs a mapping function. This allows the user to reassign bad tracks to alternate tracks at the end of the disk. Once the reassignments have been made the substitutions are transparent to the customer. The mapping information is kept in the last sector of track zero on each surface. A system map is constructed each time the disk is warm or cold booted. When a track is reassigned the bad track is marked as "bad" by the Format/Mapping Program and the replacement map is formatted with the address of the bad track. If a bad track is found the driver will generate a new map and try again. This prevents errors if a pack is exchanged without performing a warm boot, ie control C.

The following points should be noted:

1. Track substitution (as opposed to sector substitution) reduces memory requirements and speeds operaton.

2. Interchanging packs will not cause data to be misplaced. (CP/M is write protected, format causes driver to rebuild the system map and retry on read).

3. Format time per surface will be under two minutes by use of a special interlace. This does not alter the format, ie, physical track N is always formatted as N.

MAPPER3 is shipped configured for a CDC Phoenix drive, (CMD EQU TRUE). For drives with a "normal" head select, change CMD EQUATE to FALSE and reassemble.

After selecting unit and head MAPPER3 gives the operator the following choices:

ENTER 1 TO CREATE NEW MAP, (NORMAL FORMAT)

2 TO APPEND OLD MAP, (REFORMAT DISK)

3 TO VIEW OLD MAP

4 TO REPLACE BLOWN MAP, (CAUTION!)

5 TO READ VERIFY DISK, (AFTER 1, 2, 3, OR 4 ONLY)

6 TO RESTART

7 TO ABORT

SELECTION 1 is the normal format. This is the only function in this program that is required. The others are for diagnostic purposes.

SELECTION 2 allows an additional bad track to be added to the map. The entire surface is then reformatted.

SELECTION 3 allows the current map to be displayed.

SELECTION 4 replaces a map. It does not reformat the disk. It's only use is as a diagnostic and, in the unlikely event that the map sector is glitched, it can build a new map sector without formatting.

SELECTION 5 is used to read the entire disk, using mapping. This is a useful diagnostic, as it will display the location of any data errors.

SELECTION 6 allows the head and unit to be changed.

SELECTION 7 returns to CP/M

Surface Map

Each surface appears as follows:

0	0	0	0	0	0	T_0	T_1	Track 0
T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	
0	0	0	0	0	0	T_0	T_1	Track 1
T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	
0	0	0	0	0	0	T_0	T_1	Track N
T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	End of Map
1	1	1	0	0	1	0	1	

The table may be of any length, up to the limit of the sector. (The total number of tracks mapped in any system, however, must not exceed MAPSIZE/2-1, where MAPSIZE is the storage reserved in the driver).

Good tracks are not listed. Each listed track is assigned a replacement track dependent upon its position in the table. The first track is assigned to 804, the next to 805 and so forth. If a spare is bad it may also be mapped, ie, track 213 could map to 805 which maps to 807. An alternate way to handle a bad spare map is to map it to itself, ie, make the fourth entry in the table an 807.

System Map

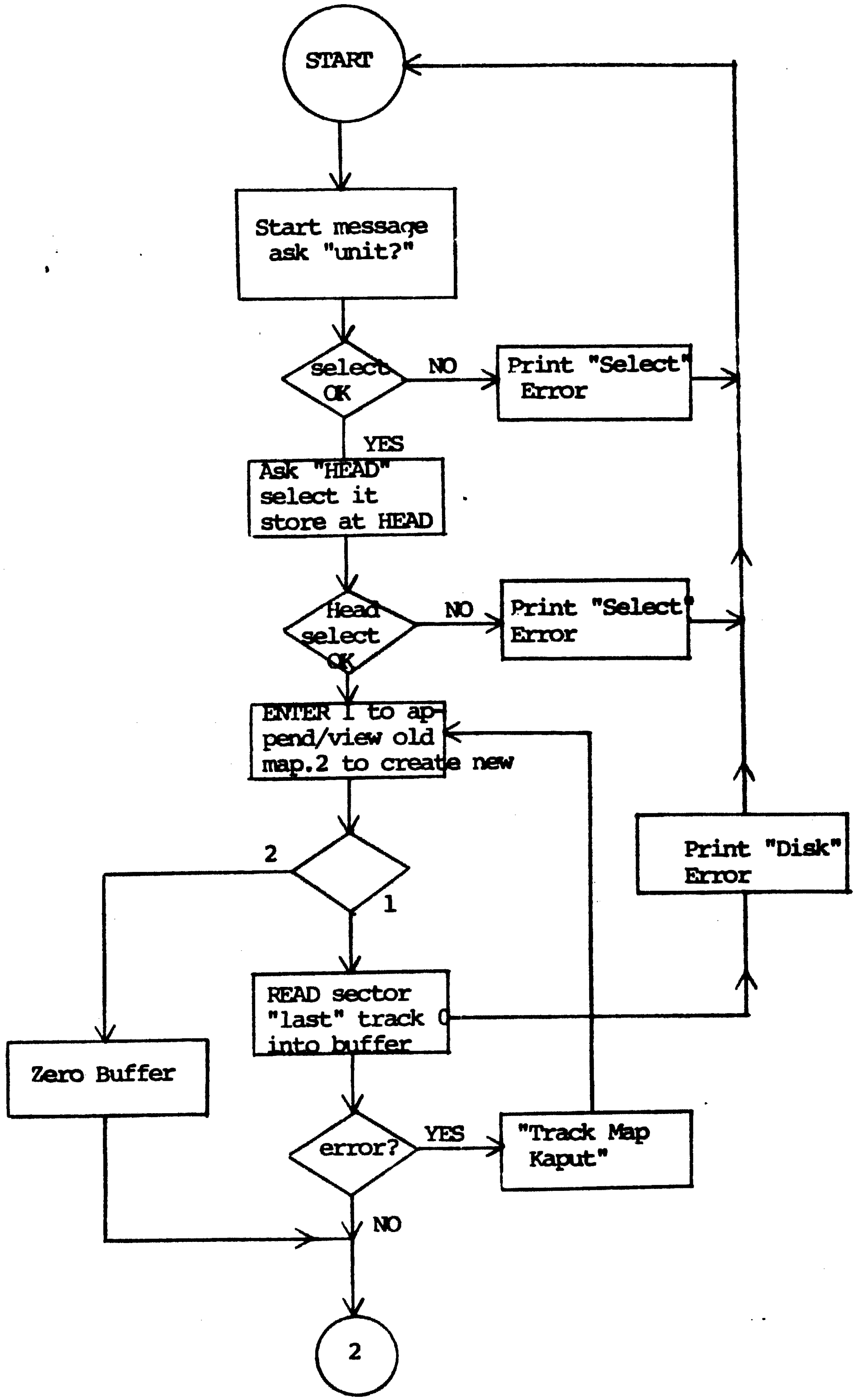
The system map constructed by MAPBIOS contains the maps of each surface. 'The MAPBIOS routine which constructs the SYSMAP is called SMAPPER. It builds a map from all drives and surfaces that are read. Not ready units are skipped by and no error is generated. The map looks as follows:

Ø	U_0	U_1	H_0	H_1	H_2	T_0	T_1	Entry 1
T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	Entry 2
Ø	U	H_0	H_1	H_2	H_3	T_0	T_1	
T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	
Ø	U_0	U_1	H_0	H_1	H_2	T_0	T_1	Entry N
T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	
1	1	1	Ø	Ø	1	Ø	1	End of Map

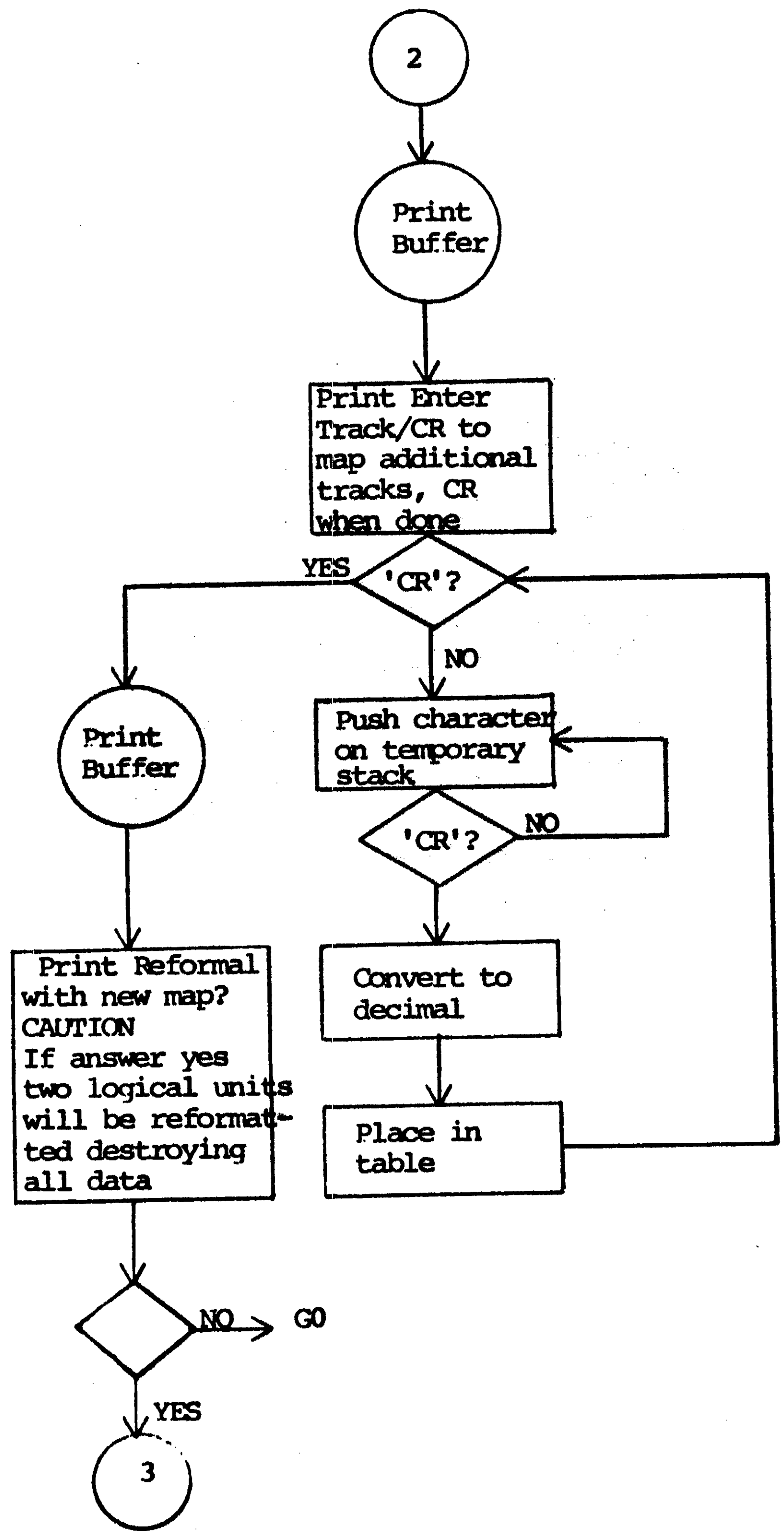
Where U is the unit #
H is the head
T is the track

The following pages flow chart the MAPPER3 program. Note that the routines KCOMMON and SMAPPER are the same ones used by MAPCOPY, MAPBIOS3, and MAPPER3.

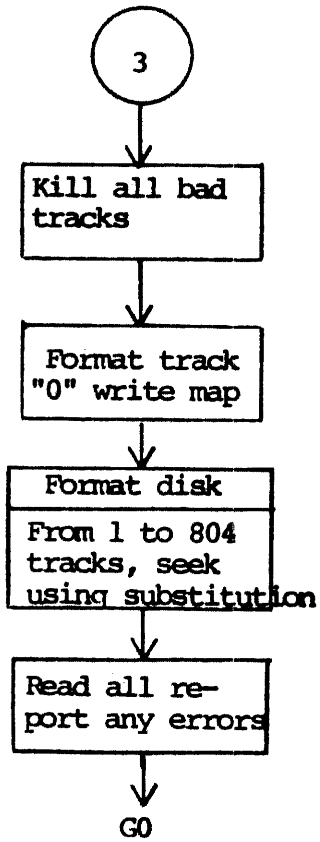
MAPPING


START
Start message ask "unit?"
select OK
NO
Print "Select" Error
YES
Ask "HEAD" select it store at HEAD
Head select OK
NO
Print "Select" Error
ENTER 1 to append/view old map.2 to create new
2
1
Print "Disk" Error
READ sector "last" track 0 into buffer
Zero Buffer
error?
YES
"Track Map Kaput"
NO
2

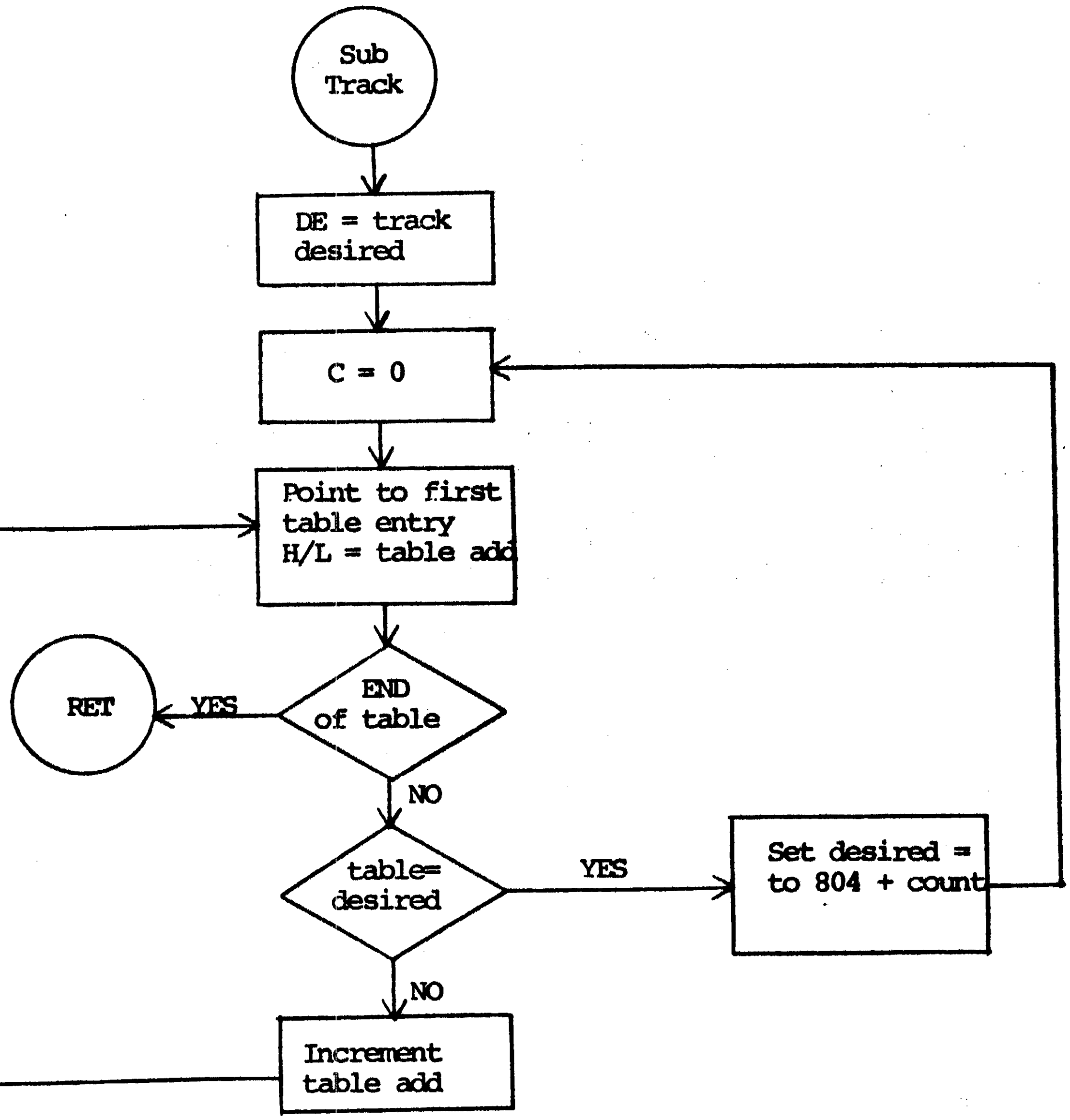
MAPPING cont'd



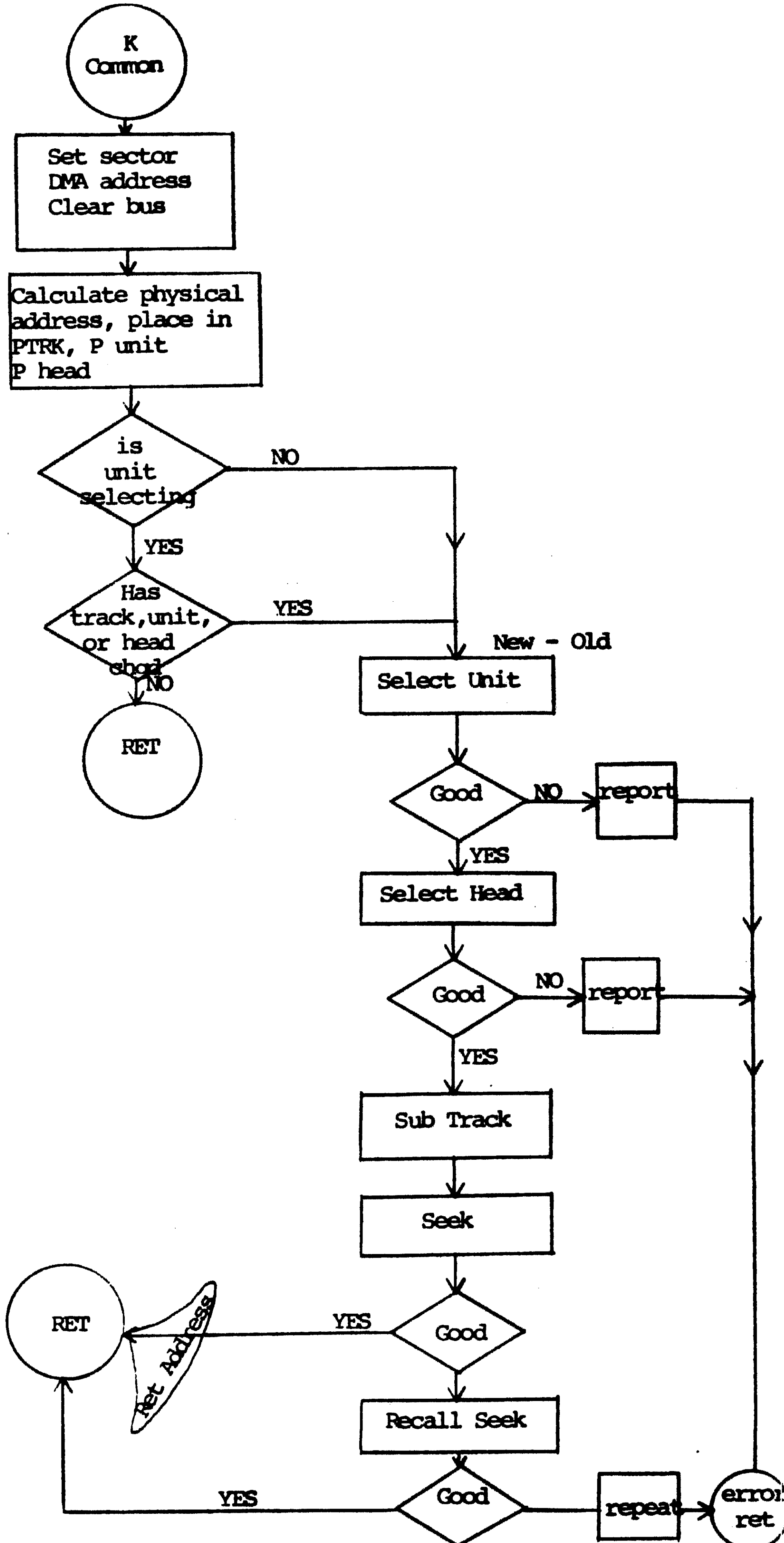
MAPPING cont'd



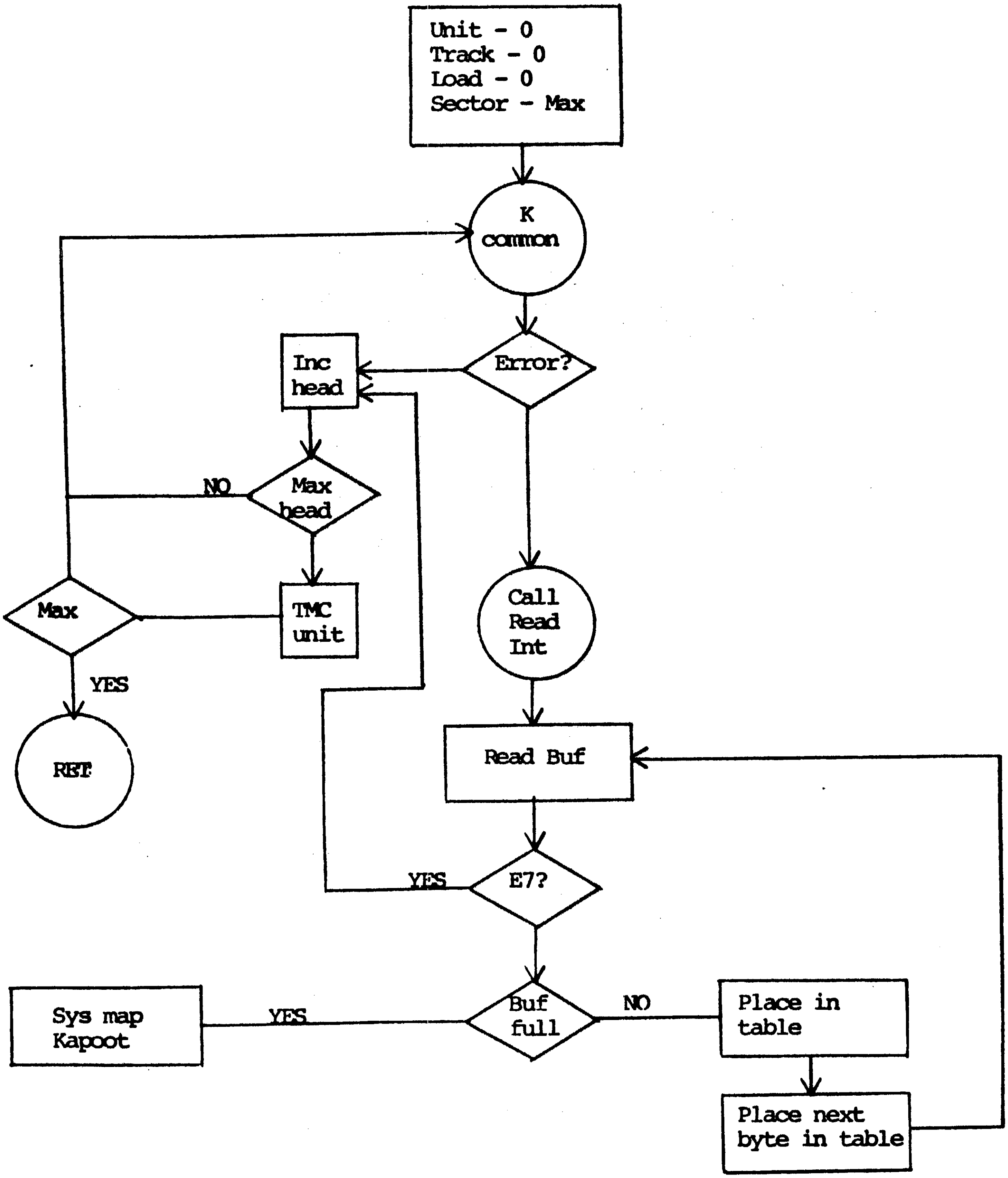
SUB TRACK - this routine uses the system map to map bad tracks to good tracks.



SUB TRACK cont'd



SMAPPER The purpose of this program is to create the system track map. The map is built by adding each surface to the map.



V BIOS Disk Driver - The BIOS's supplied by KONAN have the following features:

1. Blocking and Deblocking
   The disk uses 516 byte sectors, with 4 byte header and four 128 byte logical sectors per physical sector. This increases the storage in a CDC CMD-32 from 24 megabytes with 128 byte sectors to 29.5 megabytes. Performance is also improved.

2. No hardware interlace.
   Blocking and Deblocking make sectors available in groups of four. Typically, system overhead between sectors is 7 to 20 milliseconds or 28 to 80 per physical sector. The disk revolves every 17 milliseconds. This speed makes interlace unnecessary. Format and Duplicate programs use a logical interlace making system backup and formatting extremely fast.

3. Track Mapping
   Allows media flaws to be bypassed with minimal memory requirements, completely negligible system degradation and safegaurds against improper pack exchanges.

4. Integral Driver
   Removes 1K driver ROM after boot to allow full use of 64K.

5. 32 or 96 Megabytes
   Supports 32 megabyte disk as units A, B, C and D or a 96 megabyte disk as units A through L. Drive M is a Siemens or Persci floppy drive with a Tarbell controller.

6. Error Recovery
   The driver tries to read with all possible combinations of offsets and strobes. Each is tried twice, then the drive recalls, builds a new system map and tries again. In all, 39 retries preceed flagging a hard disk error.

7. Error Reporting

The hard disk reports the following errors.

- A write time out error
- B write fault
- C read time out 1
- D checkword error 1
- E format error
- F hard read error
- G unit select error
- H head select error
- I hard seek error
- J seek time out
- K recal time out
- L S-mapper error

Driver

The driver uses Digital Research's Blocking and Deblocking Algorithms. The driver has two entry points, writehst: and readhst:. Parameters are passed to it in the following memory locations:

hstsec: defines the host sector
hsttrk: host track #. Must be added to the units first track, ie add 4Ø2 to even unit numbers.
hstdsk: Ø for A, 1 for B, etc.

The equate headg: is set to the number of heads per unit. It is set to 2 for 32 megabyte units, 4 for 64 and 6 for 96.

HEADQ EQU 2

Logical Unit	Physical Unit	Head	Tracks
A	0	0	0-410
B	0	0	402-803
C	0	10	0-401
D	0	10	402-803
E	1	0	0-401
F	1	0	402-803
G	1	10	0-401
H	1	10	402-803
I	2	0	0-401
J	2	0	402-803
K	2	10	0-401
L	2	10	402-803

HEADQ EQU 4

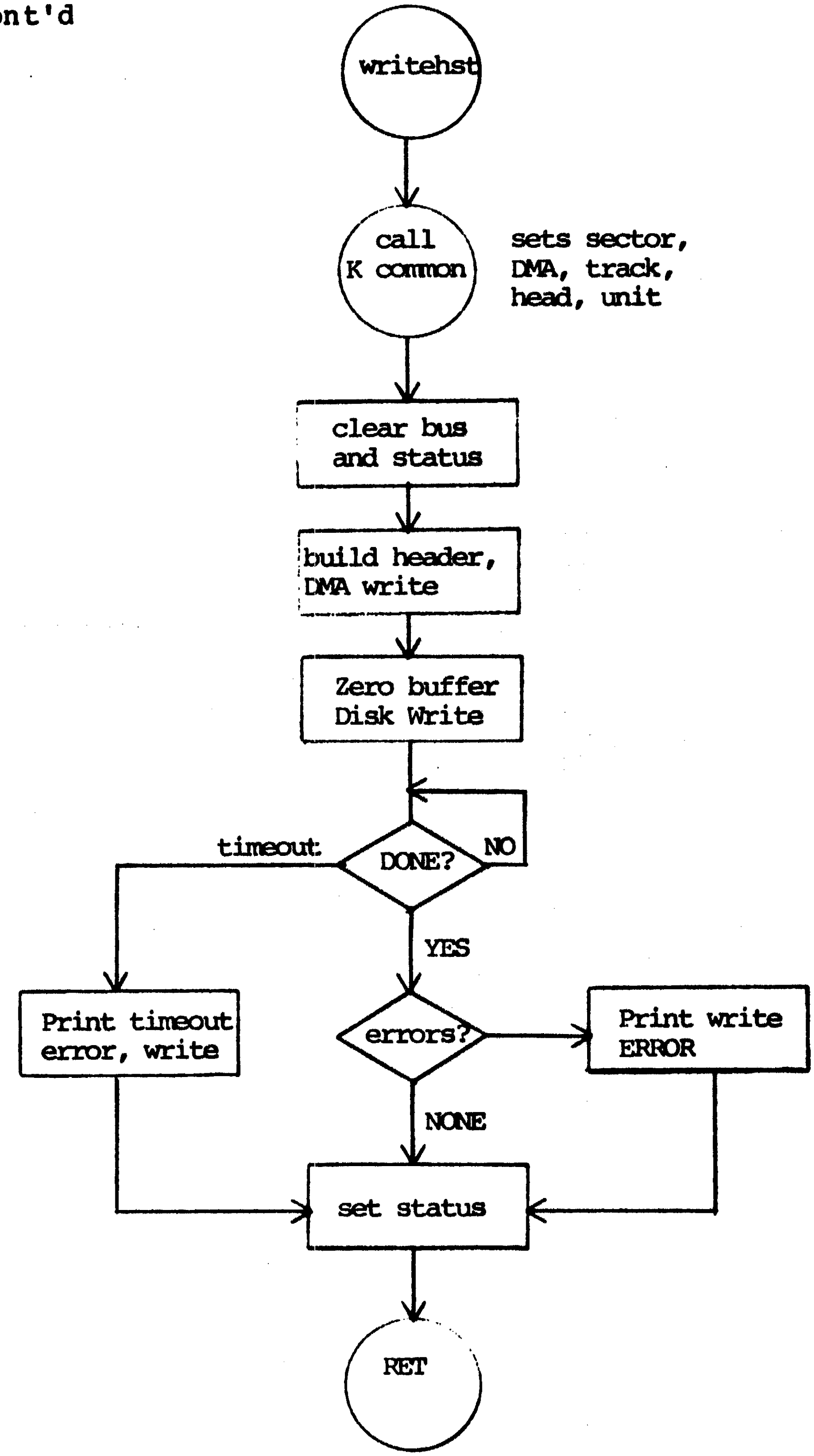
Logical Unit	Physical Unit	Head	Tracks
A	0	0	0-401
B	0	0	402-803
C	0	10	0-401
D	0	10	402=803
E	0	11	0-401
F	0	11	402-803
G	0	12	0-401
H	0	12	402-803
I	1	0	0-401
J	1	0	402-803
K	1	10	0-401 *
L	1	10	402-803

*Must be 32 megabyte drive

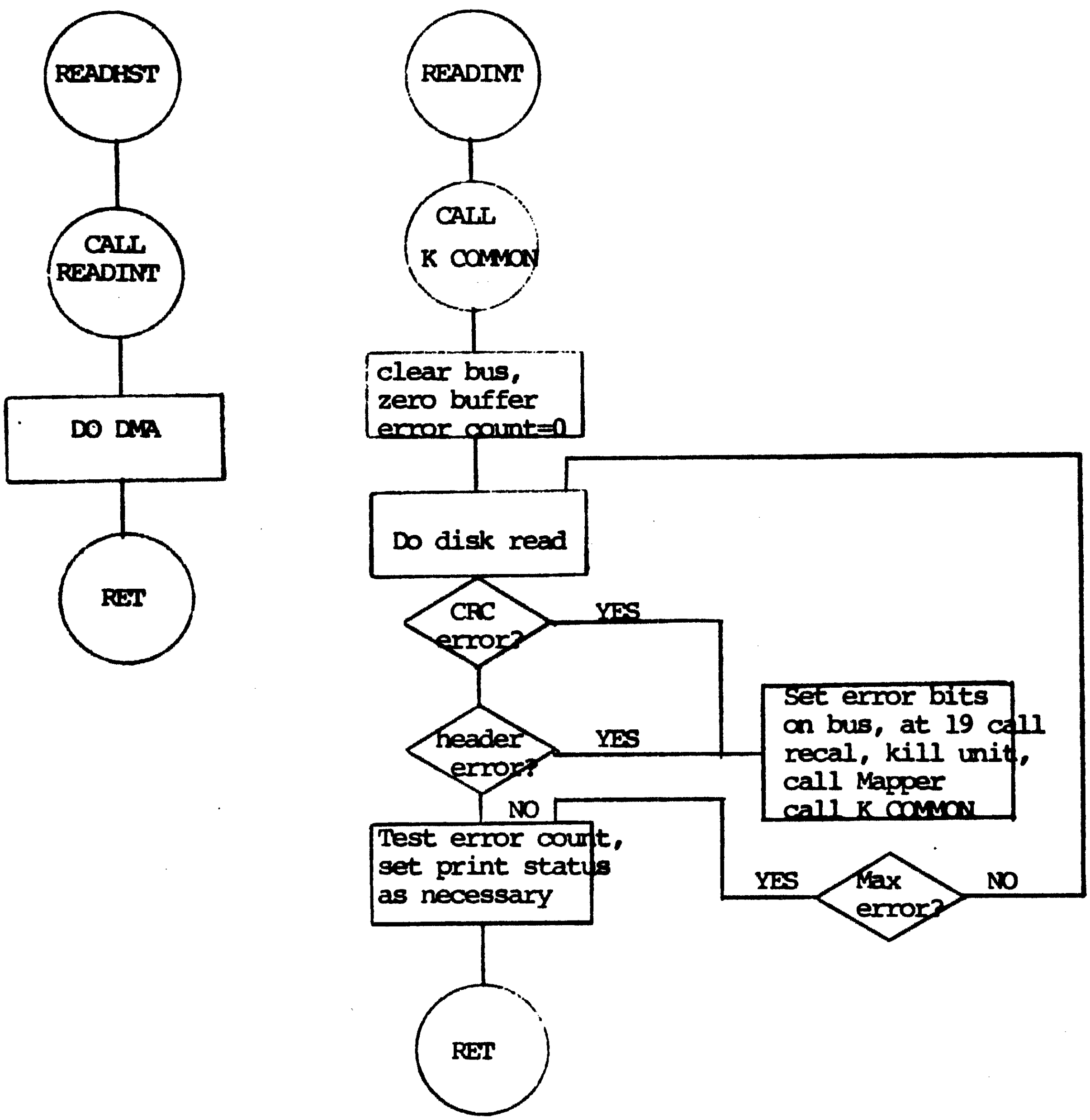
HEADQ EQU 6

Logical Unit	Physical Unit	Head	Tracks
A	0	0	0-401
B	0	0	402-803
C	0	10	0-401
D	0	10	402-803
E	0	11	0-401
F	0	11	402-803
G	0	12	0-401
H	0	12	402-803
I	0	13	0-401
J	0	13	402-803
K	0	14	0-401
L	0	14	402-803

DRIVER cont'd

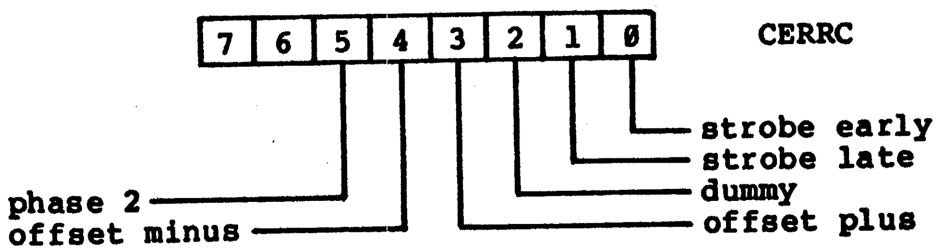


DRIVER cont'd



Read Recovery
RCOVRE:

The read recovery program sets the recovery bits on the bus. After trying all offsets and strobes multiple times it will recall, reselect unit, head, track and sector, and then retry all legal combinations of offsets and strobes again. This is accomplished by incrementing an error counter, CERRC, whose bits get mapped onto the disks bus. Definition of the bits in CERRC are as follows:



CERRC is incremented between each read attempt, and then the bus is updated to perform the desired recovery operation. Illegal combinations, (strobe early and strobe late or offset plus and offset minus) are skipped. Bit two is ignored, causing all strobes to be tried twice at each offset. When the hex count 2Ø is reached the drive is recalibrated, reselected, a new system map is built and the entire sequence is repeated.

VI BOOT

Cold Boot (This Subsection applies only to the SMC-100).

The Cold Boot program resides in ROM. It occupies 1K of memory space. The ROM may be phantomed after the boot. KONAN's SMC-100 can generate and/or respond to phantom via jumpers.

To help accomodate the many different memory/system configurations KONAN provides the user many boot options. (See attached configuration sheet). These options include:

1. Phantom: Allows boot ROM to disappear after boot and memory to replace it.

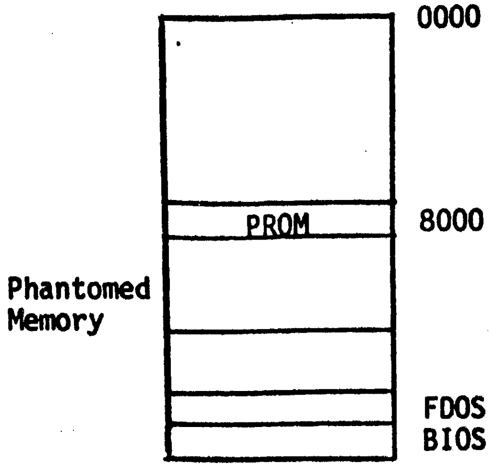
2. ROM address: ROM may be placed on any 1K boundary from 8000H to FC00H (Note - the SMC-100 may be jumpered to any of these addresses but the ROM is "blown" for one specific address and is not relocatable). Unless otherwise needed, ROM's are blown for F000.

3. Boot Memory Address and Jump Address: May be set to the base of BIOS.

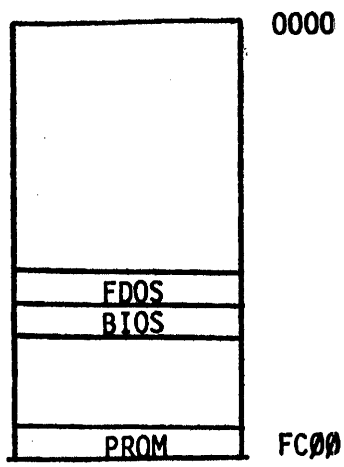
Note that the SMC-100 will not provide a power on Jump. This must be performed by the user, normally on his CPU.

Note that the Boot process is multi-staged. First, the PROM copies the necessary portion of itself into system RAM. Next, if this option is needed, the ROM phantoms away. Then the code in RAM is executed, reading the entire system image into RAM, beginning at location 0000H Track 0, Sector 0 contains a short program (MEMMOVE) to relocate the system image to the appropriate RAM location. This program is executed upon jumping to 0000H. Finally, a jump is executed to the BIOS, and the system is up and running.

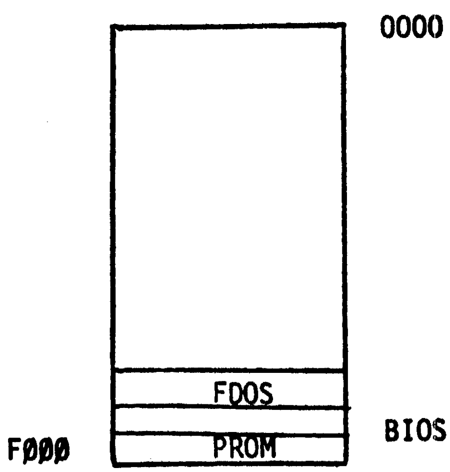
TYPICAL COLD BOOT CONFIGURATIONS



PROM is at location 8000H. Phantom is enabled. Memory from 8000H to 83FFH must phantom. Memory from 100H to the base of the BIOS may phantom. After boot all memory is available.



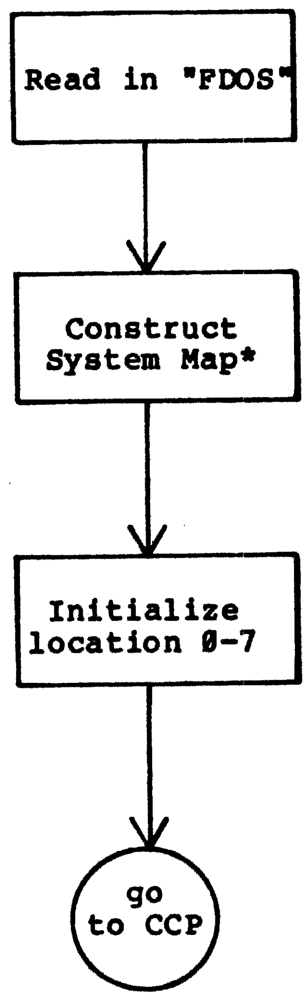
PROM resides above BIOS. Phantom maybe enabled or disabled.



PROM is at F000. It must be phantomed and so must the top 1K of memory. This configuration can only be used if the memory can phantom the top 1K. Not the BIOS and PROM overlap. This is possible because the last 1K of BIOS is uninitialized buffer areas.

Warmboot

Warmboot performs the following functions:



* If Mapping is implemented

Alternate Boot procedures:
The SMC-2ØØ has no ROM. The nature of the SMC-1ØØ's Phantom line occasionally makes booting off the ROM very difficult. The following options may be helpful.

A. 1) Boot off floppy.

2) Move hard disk bios into the TBA using DDT.

3) Move BIOS into place with DDT's memmove. (This is also an excellent diagnostic step when integrating an SMC-1ØØ boot.

B. 1) Boot off floppy

2) Execute a COM program that performs steps 2 & 3 above wthout use of DDT.

C. 1) Write a boot ROM to reside on CPU board or elsewhere. It can imitate the COLD boot provided or just read in sector zero which would hold a boot program.

VII Fast Copy

The Fast Copy program copies any hard disk surface to any other. It is a high speed copy using an interlace of 3 or 4, depending upon host system speed. A system map is built to map the bad tracks to alternates to maintain compatibility with the operating system. The primary function of this program is to shorten the length of time required to perform a backup. The duplicate program differs from PIP** in the following ways:

1. Speed: MAPCOPY is many times faster that PIP.
2. Order: MAPCOPY does not reorganize files in any way. PIP** makes all files contiguous. PIP should be used occasionally to reorganize files.

MAPCOPY uses DMA and will, therefore, only run on an SMC-100 that can run DMARAM.

VIII INTERGRATION GUIDE

The task of integrating your particular hardware configuration into the CP/M* operating system can be quite confusing. This guide is meant to be used in conjunction with the alteration manual supplied with your CP/M system. It is not intended for use by the individual who is totally unfamiliar with CP/M integrations. For those who have never attempted this before, please refer to the manuals supplied with your operating system. The diagnostics in Section III should be run <u>PRIOR TO INTEGRATION!</u>

The assumption made at this point is that you are already 'up and running'on a CP/M system, based on either standard 8", single density, single-sided, IBM/CPM compatible floppies

The first step is to assemble, load, and run the MAPPER3 program. This will format any surface of the hard disk. Run menu selection, 1 for each surface of the hard disk. (See section IV for further details on the MAPPER3 program). The hard disk drivers are integral to the program, which calls CP/M only for the terminal I/O. This program takes about 90 seconds to run, and responds with the same prompt. To exit the program and warmboot, simply enter Control C.

The next step is to assemble and load the KONGEN program. This program is KONAN's analogue to the Sysgen program provided by Digital Research. The purpose of this program is to place an image of the operating system on the boot track of the hard disk from memory or to place an image of the boot track into memory. If you are simply making a copy of an already functional operating system (which is good practice, in case you "blow" it during a modification of your operating environment), you need only retrieve the image from the hard disk surface which contains it, swap platters, and place the image on the new platter. The KONGEN program will leave any existing files and the directory intact. This program also contains integral drivers for the hard disk and can, of course, be run under your floppy disk CP/M.

To procede with the integration process, you must somehow create the desired system image in memory, so, that you may place it on the hard disk (using KONGEN). There are three "blocks" which must be loaded into the appropriate memory locations. These blocks include the CP/M FDOS, (the CCP and BIOS), the "boot" routine, and the BIOS. Each of these blocks will be discussed next.

The first block, the FDOS, is created using the MOVCPM program supplied with your operating system. This program creates a version of the FDOS relocated to be used with your desired memory size. Due to the size of the disk allocation tables, which reside at the end of the BIOS, you should use MOVCPM to create an FDOS image for a system 3K smaller than your actual memory requirements. In the following, it will be assumed that the COM file created with MOVCPM will have been called CPM.COM.

The creation of the second block is generally the most formidable of the steps that must be taken in the intergration process. This is the BIOS. KONAN provides a BIOS assembly file which is to be used as a guide for the installation of your particular hardware requirements. The two simplest steps in the modification of this BIOS, are setting the memory size desired, and altering the terminal I/O drivers.

To set the memory size, set the 'MSIZE' equate at the beginning of the BIOS file to the desired value. If you want to generate a 48K system, set MSIZE to 48. (Recall that the CPM.COM file will have been generated for 3K less than this value). After setting this value, find the terminal routines included by KONAN and alter or replace them to meet your need. For the sake of sanity, it is generally a good idea to alter only these two things the first time you attempt your integration, and only after succeeding to generate a bootable hard disk, attempt to install any other changes (such as floppy disk routines), that you will need.

For your convenience, KONAN has included sample Tarbell floppy drives and the logic in the disk select routine to route through either the hard disk drivers or the Tarbell drives, depending on the logical drive number. These routines are meant only as a sample. Although we have been using them in-house for quite some time, they are not meant to be assumed bug-free or compatible with your equipment. For your information, the SMC-100 version of the BIOS is designed to be used with a 96 megabyte Control Data Phoenix unit (although it need not be altered to run with the 32 megabyte units), as logical devices 'A' though 'L' ('A'-'D' for 32 megabyte units) and the floppies as logical units 'M' and 'N'.

When you have completed making the desired changes to the BIOS, assemble it. It is not necessary to load this file. After assembling, type the PRN-file created by the assembler, and take note of the 'CCP' and 'BIOS' equate values. as you will need them later. In particular, the CCP equate value is the value to which you must set the 'DESTIN' equate in the MEMMOVE file. The BIOS equate will be used to calculate the read-offset in the following paragraphs.

The third block, the "BOOT" routine, is a short file called MEMMOVE. The boot process for the hard disk using the KONAN controller is actually done by the ROM on the controller. This ROM brings the entire system into memory, beginning with memory address 0000H. A jump is then performed to location 0000, which will contain the MEMMOVE routine. This routine has only a single equate which must be modified for each different memory size intergrated. This equate is called 'DESTIN' in the MEMMOVE.ASM file. It should be set to the address for the CCP (FDOS). You may determine this value by inspecting the BIOS print file created in the following paragraphs. Once determined, this value should be set and the MEMMOVE program assembled. It is not necessary to load this file.

At this point, you should have created the three blocks needed to combine into the total system image which must ultimately reside on the hard disk boot track. Use DDT (see your CP/M manuals) to bring these files into memory. The following are the steps you should take under DDT:

YOU TYPE	
DDT CPM.COM IBIOS.HEX	This is the file created by MOVCPM or whatever you named your modified BIOS>
R (OFFSET)	This offset value is easily calculated by subtracting the 'BIOS' equate from 11F80 HEX
IMEMMOVE.HEX	This always resides at 780H, regardless of your memory size.
R780	
G0	Warmboot
KONGEN	and run KONGEN

Since you have just formed the system image in memory, when KONGEN asks you for the image souce, simply hit return. When it asks you for the destination, answer 'A'. These responses will place the image in RAM onto the outer boot track of the 'top' platter. For the SMC-100 version with the Control Data drive, this is the removable cartridge.

We hope that this guide will clear up the traditionally muddy waters of system integration and modification. Using the various tools that we have provided, the generation process should proceed in a fairly straight forward manner. Good Luck!!!