Franklin
Computer

# ACE 2000 SERIES COMPUTER TECHNICAL REFERENCE MANUAL




Franklin Computer Corporation
Route 73 and Haddonfield Road
Pennsauken, NJ 08110



This manual is Franklin Part Number FP-700001

©Copyright 1985, 1986 by Franklin Computer Corporation
Route 73 and Haddonfield Road
Pennsauken, NJ 08110
(609) 488-0666



Technical Support 609-488-5454
Sales 609-488-0600
Corporate Offices 609-488-0666

# GENERAL INFORMATION

The Franklin Technical Reference Manual for the ACE 2000 series of computers is composed of several sections. Every attempt has been made to include information helpful in the understanding of the function and maintenance of the ACE 2000 series of Apple compatible computers. It is not the function of this manual to be a tutorial or an introduction to the design of computers.

Several excellent books have been written on the subject of Apple related hardware and software. Although a check at your local bookstore or library is recommended to select books appropriate for your needs, the ones listed below have been shown to be particularly helpful.

APPLE II USER'S GUIDE by Lon Poole et al
Third Edition Copyright 1985
Osborne-McGraw Hill
2600 Tenth Street
Berkeley, California
94710
ISBN 0-07-881176-7

UNDERSTANDING THE APPLE IIe by Jim Sather
Copyright 1985
Quality Software
21601 Marilla Street
Chatsworth, CA
91311
ISBN 0-8359-8019-7

THE APPLE II BASIC HANDBOOK by Douglas Hergert
Copyright 1983
SYBEX
2344 Sixth Street
Berkeley, CA
94710
ISBN 0-89588-115-2

APPLE MACHINE LANGUAGE by Don Inman and Kurt Inman
Copyright 1981
Reston Publishing Company
Reston, Virginia
22090
ISBN 0-8359-0230-7

The text Understanding the Apple IIe by Jim Sather is particularly useful if you intend to get involved with detailed timing and design issues. The previous edition covering the II plus is also an excellent reference. Technical differences between the Apple IIe and the ACE 2000 Series Computer is discussed in this manual.

# Table of contents

# Franklin ACE 2XXX

# Technical Notes

**Version 1.0 dated 1/6/86**

## INTRODUCTION

This document is a technical description of all of the features found in the Franklin ACE 2XXX series of personal computers. It describes both the hardware, firmware and software aspects of the ACE 2XXX. The descriptions are presented at one of two levels, overview or detailed, depending on the topic being described. Those topics that discuss features that are identical to features found on an Apple II+, IIe or IIc are presented in an overview fashion on the assumption that the detailed aspects of such features are more than adequately described in many readily available publications. On the other hand, those features that are unique to the ACE 2XXX are discussed in detail since there are no other publications available yet that describe these features.

So, if you want to understand all of the nitty-gritty of Apple IIe video or some other standard Apple feature, this document isn't going to help you much. But if you want to know how the ACE 2XXX implements function keys, what's different and what's the same between the ACE and the Apple, or you need details on any other specific ACE feature, then read on...

One more thing: this document is presented from a programmer's point of view. Its main purpose is to provide information which enables existing or new Apple compatible application programs to also run correctly on the ACE 2XXX. Its secondary purpose is to allow third-party software developers to take advantage of some of the unique ACE features in their applications.

## HARDWARE

The ACE hardware is designed to be as compatible to the Apple IIe as possible. In actual fact, compatibility is almost 100%. This means that a programmer can expect to have access to all of the hardware features of an Apple IIe on an ACE. This includes all RAM, video displays, softswitches, I/O control registers, etc. If it's implemented on a IIe, then it's also on an ACE.

From a programmer's perspective, there is only one significant difference between the two machines - the ACE CPU runs at a slightly different speed than the Apple CPU. The Apple CPU runs at an average speed of 1.020484 MHz. The ACE CPU runs at one of two speeds, depending on the version. The original ACE's ran at an average of 1.0092109 MHz. Subsequent versions (which began production in early 1986), run at an exact 1.0227271 MHz. (The "average" is specified in these speeds because there is a "stretch" factor in the nominal clock speed which must be taken into account. The Apple and the original ACE "stretched" the clock at selected points in the horizontal retrace. The newer ACE's do not stretch the clock, so the 1.0227271 MHz is exact.)

In practical use, this difference is hardly noticeable. For most purposes, timing loops written for the Apple speed will work perfectly well on an ACE. There is one place, however, where this speed difference may present a problem: I/O timing loops, particularly in regard to disk I/O. If any of your programs rely on a very tight timing loop for disk I/O, then you should recheck your timing factors using the ACE speed. If the ACE speed pushes your loop out of its constraints, then you'll have to implement a special ACE-specific routine in your applications.

## MODEL NUMBERS

There are three models of the ACE 2XXXX, depending on how many integrated disk drives are in the unit. The 2000 contains no disk drive. The 2100 contains one disk drive, and the 2200 contains two disk drives. Except for the number of disk drives, all three models are identical in all other respects.

## CPU

The ACE uses a GTE 65SC02 (or compatible) microprocessor for its CPU. This is the CPU that implements the new instructions in the "enhanced" Apple IIe. All of the ACE firmware has been designed to utilize the expanded instruction set of the 65SC02. (Although some of the software, particularly FDOS, does not for compatibility reasons.)

## MEMORY

There is a memory map at the end of this document which describes the contents of the memory space in terms of RAM and ROM in detail.

## RAM

The ACE contains 128K of RAM. It is partitioned into two 64K banks in exactly the same way as in the Apple IIe. All of the memory control softswitches of the IIe work the same way on the ACE. (The memory map at the end of this document only shows one of the 64K banks of RAM. The second bank isn't shown to make the drawing a little easier to read, but it would appear on the map exactly the same as the first bank.)

## ROM

The ACE contains three 2764 EPROMS which provide for 19712 bytes of accessible read-only storage. Of this 19712 bytes, only about 18944 bytes are actually used, which are broken down as follows (all numbers are in decimal):

| | |
|---|---|
| Monitor | 2048 |
| BASIC | 10240 |
| 80 column | 2304 |
| Disk boot | 2304 |
| Printer | 1536 |
| Diagnostics | 256 |
| Scroll routine | 256 |

Just as in the IIe, the ACE has an "external" and an "internal" ROM space which is controlled by the same softswitch as in the IIe. The only things stored in the ACE internal ROM are the diagnostics and the scroll routine. See the memory map for details of where all of the ROM appears in the memory space.

## I/O SPACE

The I/O space is the range of addresses from C000h to C0FFh which is dedicated to the control of I/O devices. The I/O space in the ACE operates the same as the I/O space in the IIe. In particular, all of the softswitches are the same. (There are some additional softswitches in the ACE though, which we'll describe later.)

## DISPLAY

The ACE provides all of the display modes of the IIe (40 and 80 column text, lores graphics, hires graphics, and double hires graphics.) All of the display mode softswitches and the video RAM partitioning are the same on the ACE as they are on the IIe.

## DISK CONTROLLER

The ACE contains an integral disk controller chip hard-wired to slot six which is compatible with a standard Apple floppy disk controller card. (The firmware driving this chip is **very** different from the standard Apple Boot16 ROM however, so be careful if you're interfacing into the boot ROM.)

## DISK DRIVES

Depending on the model number, the ACE will have from none to two integral floppy disk drives in slot six. These are standard Apple-compatible drives with the additional capability of having forty tracks as opposed to the normal thirty-five tracks.

## PRINTER PORT

The ACE contains an integral parallel printer port hard-wired into slot one. This port is hardware (data and strobe registers) and firmware compatible with the standard Apple parallel printer card.

## FIRMWARE

The ACE contains firmware which implements the functions of the standard IIe and also some functions that are normally provided by peripheral cards. While all of the firmware is designed to be compatible, it is not the same as the IIe firmware. From a functional standpoint (top-level, user commands), you can do most of the things with an ACE that you can do with a IIe. However, from an assembly routines level (entry points), the ACE does not implement every IIe routine in exactly the same way or in the same location. Refer to the **ENTRY POINTS** section of this document for further details.

## MONITOR

The Monitor is a collection of general purpose routines that provide for the low-level management of the hardware features of the machine. Most of the Monitor deals with I/O, particularly screen and keyboard I/O. The Monitor is designed to be compatible with Apple's Autostart Monitor.

The design philosophy of the Monitor placed emphasis on application compatibility as opposed to user command compatibility. The reason is that most users do not require a full featured debugging monitor, while all users need that collection of subroutines in the Monitor which is most frequently called by application programs. Because of this, not all of the IIe Monitor user commands are implemented in the ACE Monitor. In particular, neither the disassembly or instant assembly commands are implemented. Here's a list of those user commands which are supported by the ACE Monitor:

| | |
|---|---|
| XXXX | Displays contents of memory location hex XXXX. |
| XXXX.YYYY | Displays contents of memory location from hex XXXX to YYYY. |
| .XXXX | Displays contents of memory from last used memory location to location hex XXXX. |
| RETURN | Displays contents of next 8 memory locations. |
| XXXX:YY ZZ... | Changes contents of memory locations beginning at hex XXXX to the hex values YY, ZZ, ... |

| | |
|---|---|
| :YY ZZ... | Changes contents of memory locations beginning at last used memory location to hex values YY, ZZ, ... |
| XXXX<YYYY.ZZZZ | Moves the block of memory beginning at hex YYYY and ending at hex ZZZZ to hex XXXX. |
| XXXXG | Calls an assembly language program at location hex XXXX. |
| N | Sets display output to normal video. |
| CTRL B | Coldstarts BASIC. |
| CTRL C | Warmstarts BASIC. |
| CTRL Y | Jumps to location 3F8h. |
| X CTRL P | Turns on output to slot X. |

## BASIC

The ACE contains a BASIC interpreter which is Applesoft compatible. FBASIC is syntatically and token compatible with Applesoft so that programs which are composed entirely of BASIC statements which run on an Apple should run on an ACE.

There are a few minor differents between the two BASICs:

* FBASIC handles lowercase differently.
* FBASIC does not implement any tape commands.
* FBASIC has an additional statement, **FKEY**.

The lowercase differences occur in three ways. First, you can enter any BASIC statement in lowercase and FBASIC will recognize it. Second, all listed output will show the BASIC keywords in lowercase. And third, lowercase variables are distinct from their uppercase counterparts. (In other words, the variables **aa** and **AA** represent two different variables.)

The tape commands (LOAD, SAVE, etc.) are not implemented because the ACE does not have a tape port.

The FKEY statement is used to define function key definitions as follows:

**FKEY** ***numeric_expression, string_expression***

***numeric_expression*** can be any valid BASIC number in the range 1-12 and ***string_expression*** can be any valid BASIC string. For examples:

```
FKEY 2,"Hello"
A$="Make ": B$="my ": C$="day!"
X = 3*2
FKEY X,A$+B$+C$
```

These example BASIC statements will give function key 2 a definition of "Hello" and function key 6 a definition of "Make my day!".

To redefine a function key that already has a definition, simply enter another **FKEY** statement for that function key - BASIC will delete the old definition and insert the new one.

You can have up to 233 characters in the defining strings of all of the function keys. These 233 characters can be allocated to the twelve function keys in any combination. BASIC does not check to make sure that the number of characters in all definitions doesn't exceed 233! If you define more than 233 characters in all of your definitions, you'll destroy the contents and/or the ordering of all the definitions.

## 80 COLUMNS

The ACE contains 80-column firmware that is designed to be compatible with IIe 80-column firmware. The 80-column firmware resides in slot three at C300-C3FFh and uses expansion peripheral card ROM space from C800-CFFFh. It supports the following standard IIe escape and control character commands:

| | |
|---|---|
| ESC @ | clear the screen |
| ESC A | cursor right |
| ESC B | cursor left |
| ESC C | cursor down |
| ESC D | cursor up |
| ESC E | clear to end of line |
| ESC F | clear to end of screen |
| ESC I | cursor up, stay in escape mode |
| ESC J | cursor left, stay in escape mode |
| ESC K | cursor right, stay in escape mode |
| ESC M | cursor down, stay in escape mode |
| ESC 4 | switch to 40 column mode |
| ESC 8 | switch to 80 column mode |
| ESC CURSOR RIGHT | cursor right, stay in escape mode |
| ESC CURSOR LEFT | cursor left, stay in escape mode |
| ESC CURSOR UP | cursor up, stay in escape mode |
| ESC CURSOR DOWN | cursor down, stay in escape mode |
| ESC CTRL Q | switch to 40 column mode & deactivate 80 cols |
| CTRL G | bell |
| CTRL H | cursor left |
| CTRL J | cursor down |
| CTRL K | clear to end of screen |
| CTRL L | clear the screen |
| CTRL M | carriage return |
| CTRL N | turn off reverse video |
| CTRL O | turn on reverse video |
| CTRL Q | switch to 40 column mode |
| CTRL R | switch to 80 column mode |
| CTRL U (kybd) | cursor right |
| CTRL U (prgm) | switch to 40 column mode & deactivate EVKS |
| CTRL V | scroll down |
| CTRL W | scroll up |
| CTRL Y | home the cursor |

| | |
|---|---|
| CTRL Z | clear line |
| CTRL \ | cursor right |
| CTRL ] | clear to end of line |
| CTRL ^ | position the cursor |

In addition to these standard IIe commands, the ACE 80-column firmware supports a number of special commands which are activated by the special keys in the numeric pad area of the keyboard:

**CLRS** - CLEAR THE SCREEN - clears the entire screen and positions the cursor to the first column in the top line.

**HOME** - HOME THE CURSOR - puts the cursor in the "home" position - the first column in the top line.

**CLRL** - CLEAR LINE - clears the entire screen line that the cursor is in. The cursor position is not affected.

**INSC** - INSERT A CHARACTER - "pushes" characters to the right of the current cursor column by one column position. Only the characters at and beyond the current cursor position will be shifted. A space character will be displayed in the current column to replace the character that was shifted to the right. Any character that was in the last column of the line will disappear.

**DELC** - DELETE A CHARACTER - shifts all characters that are to the right of the current cursor column left by one column position. The character that was under the cursor is lost (replaced by the character that was adjacent to it to the right). A space character will be displayed in the last column of the line.

**CPES** - COPY ENTIRE SCREEN LINE - sends all of the characters on the current screen line to the current running program for execution. This is equivalent to typing the **RIGHT ARROW** key until you get to the end of the line and then hitting **RETURN**. **CPES** will work regardless of the current cursor position in the screen line.

**CPES** is designed for editing lines in a BASIC program. To use it, first **LIST** the line that you want to edit. Then use the escape cursor movement keys to get to the section of the line that needs edited, edit the line using **INSC** and/or **DELC**, and then hit **CPES** when you're done editing. The edited line will be sent to BASIC with your changes incorporated.

If you use **CPES** on a full line (a line with 40 characters in 40 column mode or 80 characters in 80 column mode), then the last character on the line will be cleared to a space after **CPES** executes. Don't worry - the character is only cleared on the display screen. The correct last character will have been sent to the running program.

## BOOT16

The BOOT16 firmware resides in slot six at C600-C6FFh and uses expansion peripheral card ROM space from C800-CFFFh. Its only purpose is to boot a disk. Booting is accomplished by passing control to the entry point at C600h.

## PRINTER

The ACE contains firmware to drive the integral parallel printer port. The printer firmware resides in slot one at C100-C1FFh and uses expansion ROM at B000-B4FFh.

The printer firmware reads option switch number two on the backpanel to determine the **Auto Linefeed** setting. Set the switch to **ON** to cause a linefeed to be sent to the printer after every carriage return.

All commands to the printer firmware must be prefixed with **CTRL I**. Here's a complete description of all of the printer firmware commands: (Printer commands are not recognized until the printer firmware has been activated with a **PR#1** or equivalent.)

CTRL I A - EPSON SCREEN DUMP

Dumps a screen of data to an Epson printer. If you're dumping a hires graphics screen, the Epson must have the Graftrax option installed for the command to work correctly. The parameters of the dump (text/graphics, normal/inverse, primary page/secondary page and rotate/no rotate) should be setup with other commands before the dump command is executed. Once started, the dump can be aborted by typing the **ESC** key.

CTRL I B - OKIDATA SCREEN DUMP

Dumps a screen of data to an Okidata printer. If you're dumping a hires graphics screen, the Okidata must have the Oki-Plot option installed for the command to work correctly. The parameters of the dump (text/graphics, normal/inverse, primary page/secondary page and rotate/no rotate) should be setup with other commands before the dump command is executed. Once started, the dump can be aborted by typing the **ESC** key.

CTRL I C - C. ITOH SCREEN DUMP

Dumps a screen of data to a C. Itoh printer. The parameters of the dump (text/graphics, normal/inverse, primary page/secondary page and rotate/no rotate) should be setup with other commands before the dump command is executed. Once started, the dump can be aborted by typing the **ESC** key.

CTRL I D - GEMINI SCREEN DUMP

Dumps a screen of data to a Star Micronics Gemini printer. The parameters of the dump (text/graphics, normal/inverse, primary page/secondary page and rotate/no rotate) should be setup with other commands before the dump command is executed. Once started, the dump can be aborted by typing the **ESC** key.

CTRL I E - SCREEN ECHO ON

Turns on screen echo so that characters that are printed are also displayed on the screen.

CTRL I F - SCREEN ECHO OFF

Turns off screen echo so that characters that are printed are not displayed on the screen.

CTRL I H - SELECT HIRES GRAPHICS SCREEN DUMP

Informs the firmware that all subsequent screen dumps should be hires graphics dumps (as opposed to text dumps.)

CTRL I I - SELECT INVERSE GRAPHICS DUMP

Informs the firmware that all subsequent screen dumps should be in inverse mode.

CTRL I J - SELECT NON-INVERSE GRAPHICS DUMP

Informs the firmware that all subsequent screen dumps should be in normal (non-inverse) mode.

CTRL I L - TURN ON AUTO LINEFEED

Causes the firmware to send the printer a linefeed character automatically after every carriage return character.

CTRL I M - TURN OFF AUTO LINEFEED

Causes the firmware to not send the printer a linefeed character automatically after every carriage return character.

CTRL I P - SELECT PRIMARY SCREEN PAGE GRAPHICS DUMP

Informs the firmware that the primary page (page 1) is the page that should be dumped on all subsequent screen dumps.

CTRL I Q - SELECT NON-ROTATED SCREEN DUMP

Informs the firmware that all subsequent screen dumps should not be rotated.

CTRL I R - SELECT ROTATED SCREEN DUMP

Informs the firmware that all subsequent screen dumps should be rotated.

CTRL I S - SELECT SECONDARY SCREEN PAGE GRAPHICS DUMP

Informs the firmware that the secondary page (page 2) is the page that should be dumped on all subsequent screen dumps.

CTRL I T - SELECT TEXT SCREEN DUMP

Informs the firmware that all subsequent screen dumps should be text dumps (as opposed to hires graphics dumps.)

CTRL I U - ALLOW FULL CHARACTER SET 0-255

Normally, the firmware only allows the character codes from 0 to 127 to be sent to the printer. If it receives a code in the range of 128 to 255 it subtracts 128 from it to force it into the range 0 to 127. This command instructs the firmware not to subtract 128 from character codes.

CTRL I V - RESTRICT CHARACTER SET TO 0-127

Causes the firmware to restrict the codes sent to the printer to be in the range of 0-127. The firmware will subtract 128 from any code in the range 128-255 to force codes into the 0-127 range.

CTRL I X - ENTER TRANSPARENT MODE

Puts the firmware into a "transparent" mode where it will no longer recognize any commands.

CTRL I Z - CHANGE THE COMMAND DESIGNATOR CHARACTER

Changes the command designator character to be the next character typed (or **PRINT**ed) after the Z character. For example, **CTRL I Z CTRL A** will change the command designator character to **CTRL A**.

## DEFAULT SETTINGS

Many of the printer commands select between various firmware options. These are the defaults to those options - what the options are set to when you initially activate the printer firmware:

| Option | Default setting |
|---|---|
| Screen echo | on |
| Screen dump type | hires graphics, primary page, normal video (non-inverse), not rotated |
| Auto linefeed | set to the setting of switch #2 |
| Code set | 0-127 only |

## PRINTER COMMAND SUMMARY LIST

(These are the second characters of the commands - prefix them all with the command designator character which is normally a **CTRL I**.)

| Command | Function |
|---|---|
| A | Epson screen dump |
| B | Okidata screen dump |
| C | C. Itoh screen dump |
| D | Gemini screen dump |
| E | turn screen echo on |
| F | turn screen echo off |
| H | select hires graphics dump |
| I | select inverse image dump |
| J | select normal image dump |
| L | turn on auto linefeed |
| M | turn off auto linefeed |
| P | select primary page dump |
| Q | select non-rotated dump |
| R | select rotated dump |
| S | select secondary page dump |
| T | select text dump |
| U | allow character codes 0-255 |
| V | restrict character codes to 0-127 |
| X | enter transparent mode |
| Z | change command designator character so that the entire range of codes from 0 to 255 can be printed. |

## OPERATING SYSTEMS

In general, the ACE hardware and firmware support any operating system that will operate on the Apple IIe.

## DOS 3.3/FDOS

FDOS is the operating system which is supplied with the ACE. It is fully compatible with Apple's DOS 3.3 at a command level and a media formatting level. It is also compatible with all of the Apple-defined entry hooks to DOS 3.3. However, it is entirely different code than DOS 3.3 and loads into the language card RAM area, so it is not compatible with DOS enhancer programs which "patch" DOS to acheive enhanced functionality.

FDOS provides several features and enhancements over DOS 3.3. These include additional commands, abbreviated commands, an expanded catalog display, command line argument passing, easier program and hard disk interfacing, and a significant speed increase (FDOS is up to ten times faster than DOS 3.3 and about twice as fast as PRODOS.) For complete details on all of the features of FDOS, refer to the FDOS Technical Manual. In the meantime, here's an overview:

Here's a description of each of the new commands:

COPY filename, <drive> [,<volume>] [,<slot>]

Copies the specified file from the current drive to the specified drive. If the file already exists on the destination drive, it will be deleted before the COPY is performed. The destination drive must be different than the current drive. COPY cannot be executed by a BASIC program. COPY uses almost all of free memory to perform the copy. Because of this, any programs that had been stored in memory before the copy will be overwritten during the copy. (BASIC is coldstarted after the copy.)

INIT filename [,T <ntracks>] [,<volume>] [,<drive>] [,<slot>]

INIT is not a new command, but the ability to specify the number of tracks is. <ntracks> may either equal 35, 40 or 80. (80 is only available on double-sided drives.) If <ntracks> is omitted, then the default value is whatever type of diskette FDOS was booted from.

OPEN filename [,L <length>] [,T <type>] [,<volume>] [,<drive>] [,<slot>]

Again, the OPEN command is not new, but the <type> parameter is. <type> specifies the file's type byte. If you're opening an existing file, then <type> must match the file's type byte. If you're creating a new file, then <type> will be used for the file's type byte.

RDCAT R <cattype> [,<volume>] [,<drive>] [,<slot>]

RDCAT provides the ability to read catalog entries. Refer to the FDOS Technical Manual for details.

TYPE filename [,<volume>] [,<drive>] [,<slot>]

Outputs the contents of the specified file to the current output slot. If the file is a text file, then the file is output as ASCII characters, otherwise the file is hexadecimal bytes.

Here's a rundown on the abbreviated commands:

| Abbreviation | Full command name |
|---|---|
| DEL | DELETE |
| REN | RENAME |
| / | CATALOG |
| - | RUN, BRUN, or EXEC (action depends on file's type) |

One important difference between FDOS and DOS 3.3 is that FDOS runs out of the language card RAM area. This has two consequences: 1.) BASIC programs can be larger, and 2.) You can't use FDOS to load or save data stored in the language card area. A complete discussion of FDOS memory usage can be found in the FDOS Technical Manual.

**PRODOS**

The ACE supports PRODOS in that PRODOS works with no problems on the ACE. However, Franklin does not at this time supply a PRODOS compatible operating system.

## PASCAL

The ACE supports PASCAL in that PASCAL works with no problems on the ACE. However, Franklin does not at this time supply a PASCAL compatible operating system.

## DIFFERENCES

This section discusses the major differences between an ACE and an Apple IIe.

## HARDWARE

### SPEED

As mentioned earlier, the ACE CPU runs at a slightly different speed than the Apple IIe CPU. The Apple CPU runs at an average speed of 1.020484 MHz. The ACE CPU runs at one of two speeds, depending on the version. The original ACE's ran at an average of 1.0092109 MHz. Subsequent versions (which began production in early 1986), run at an exact 1.0227271 MHz. (The "average" is specified in these speeds because there is a "stretch" factor in the nominal clock speed which must be taken into account. The Apple and the original ACE "stretched" the clock at selected points in the horizontal retrace. The newer ACE's do not stretch the clock, so the 1.0227271 MHz is exact.)

### A-B ROM

As can be seen in the memory map, the ACE has additional expansion ROM located at addresses A000-BFFFh. The IIe does not have this. When the AB ROM is enabled, it disables the corresponding RAM in the locations A000-BFFFh. The AB ROM is enabled with an access to softswitch C079h. It is disabled with an access to softswitch C078h. The status of the AB ROM can be read at softswitch C022h (negative means enabled.) The AB ROM is normally enabled only when the printer firmware is active.

### CASETTE PORT

The ACE has no tape cassette port.

## KEYBOARD

The ACE keyboard is larger and has more keys (including a numeric pad) than a IIe keyboard. The ACE also has OPEN and CLOSED F keys as opposed to OPEN and CLOSED APPLE keys.

## ADDITIONAL SOFTSWITCHES

The ACE has some additional softswitches to control some of the ACE's enhanced features. These are described in the appropriate sections of this document.

## FIRMWARE

All of the firmware in the ACE is significantly different than the firmware in the IIe. However, even though the code is different, it is designed to function exactly the same as the IIe firmware as much as possible.

## SIGNATURE BYTE

The signature byte at FBB3h in the Monitor ROM has a value of 06h, just as in the IIe.

## INTERRUPT HANDLING

The ACE handles interrupts differently than the IIe does (although similarly to the IIc.) In the ACE firmware, interrupts are always enabled. This means that an interrupt might occur while the auxiliary 64K bank of memory or the internal ROM is enabled. To account for this, the interrupt routine saves the current state of the 64K memory banks and the internal/external ROM at the entrance to the interrupt routine. The main 64K bank and external ROM are then enabled before passing control to the RAM based interrupt handler. Control is passed to the RAM based interrupt handler in such a way that an RTI instruction causes control to return to the firmware interrupt handler. The firmware interrupt handler then restores the state of the 64K RAM banks and the internal/external ROM to their conditions at the time of the interrupt and exits from interrupt processing with an RTI.

## COMPATIBILITY

It is not practically possible to quantify the percentage of compatibility of the ACE with the IIe. This is due to the larger number of applications available for the IIe and also to their changing nature. (If version 1 of an application works correctly, there is no guarantee that version 2 will also.)

In general, the only way to determine if an application will run on the ACE is to try it out completely. Franklin tries to do this with the major applications, but obviously they cannot comprehensively test every available Apple compatible program. Franklin does make available a list of those programs known to run without problems on the ACE. Programs that are on this list have been 100% tested by Franklin.

There are four major areas of compatibility issues on the ACE: BASIC based software, PASCAL based software, assembly language based software, and the hardware.

## BASIC PROGRAMS

In most cases, programs that are written entirely in BASIC will work correctly on the ACE. Franklin's BASIC is 100% compatible with Applesoft. Franklin has only found one class of problems in programs of this type: if the BASIC program uses almost all of the available memory on a IIe (little or no free memory left), then there is a chance that such a program will abort on the ACE with an "out of memory" error. This has to do with the fact that the internal operations of the BASIC's are slightly different and in some cases the ACE BASIC will use an extra few bytes in expression evaluation or storage allocation.

## PASCAL PROGRAMS

Programs which are written entirely in PASCAL should run correctly on the ACE with no problems. Franklin has not found any PASCAL program which will not run on the ACE.

## ASSEMBLY LANGUAGE PROGRAMS

This is the biggest compatibility problem area. In general, an assembly language program will only work correctly on the ACE if it restricts its ROM calls and ROM accessing to those entry points which are defined in the **ENTRY POINTS** section of this document. (In actual fact, limiting ROM access to these entry points does not guarantee compatibility because of secondary issues such as subroutine side effects, but the converse is certainly true - if an entry point is used which is not on the list, the application will not work.)

Pure assembly language programs are not the only problem area. Franklin has also found many programs which are written mostly in BASIC but which make assembly language calls which also encounter some problems. Again, these problems stem from using entry points that the ACE doesn't support.

## HARDWARE

For the most part, there are no significant hardware related compatibilty problems on the ACE. Franklin has only found one class of programs that fail because of a hardware incompatibility and one class of hardware failures. These are disk formatting programs and the Unidisk 3.5 drive. Both of these fail because of the slight difference in system timing in the ACE. (Although the Unidisk 3.5 hardware will work on the newer ACE's with the changed no-stretch timing.)

## ENTRY POINTS

Here is a list of all firmware entry points in the ACE which are compatible with their corresponding locations in the Apple II, II+, IIe or IIc computers. The list shows entry points for both subroutines and data. A subroutine entry point is a memory location which is called by other programs. A data entry point is a memory location which is referenced (read) by other programs.

The purpose of this list is not to describe the function of each entry point but rather to be used to determine if a particular entry point is available in the ACE. Therefore the list does not describe the function of each entry point - the function is equivalent to the corresponding Apple function at the designated location. Use the list by checking the entry points you use in your programs against it. If you use an entry point that doesn't appear in the list then there's an excellent chance that your program won't work on the ACE.

Nothing should be assumed about the contents of the subroutines at the entry points. For example, if XXXX is an entry point on the ACE and XXXX+2 is an entry point on the Apple, you cannot assume that XXXX+2 is also an entry point on the ACE. In some cases it may be, but don't rely on things like that.

There are other entry points in the ACE that provide functions which are enhancements to the Apple machines. These enhanced entry points are not included in this list and can be found in another document.

For ease of use, the list is broken up into the major firmware categories - Monitor, BASIC, 80-column, printer and disk boot. All entry points in the list are hexadecimal.

MONITOR ENTRY POINTS

| | | | |
|---|---|---|---|
| F800 | FB60 | FC9E | FDF6 |
| F819 | FB6F | FCA0 | FDF9 |
| F828 | FBB3 | FCA8 | FE12 |
| F832 | FBC0 | FCBA | FE2C |
| F836 | FBC1 | FCE2 | FE80 |
| F847 | FBDD | FCF9 | FE84 |
| F864 | FBE2 | FD0C | FE86 |
| F871 | FBE4 | FD10 | FE89 |
| F87B | FBF0 | FD18 | FE8B |
| F88E | FBF4 | FD1B | FE93 |
| F940 | FBFC | FD35 | FE95 |
| F941 | FBFD | FD62 | FEBC |
| F944 | FC10 | FD67 | FF2D |
| F948 | FC14 | FD6A | FF3A |
| F94A | FC1A | FD6F | FF3F |
| FA40 | FC20 | FD75 | FF4A |
| FA59 | FC22 | FD78 | FF58 |
| FA62 | FC24 | FD8B | FF59 |
| FB1E | FC42 | FD8E | FF65 |
| FB2F | FC58 | FD96 | FF69 |
| FB36 | FC5A | FDB3 | FF70 |
| FB39 | FC62 | FDCE | FFA7 |
| FB3C | FC64 | FDDA | FFCB |
| FB40 | FC66 | FDE3 | |
| FB4B | FC70 | FDED | |
| FB5B | FC9C | FDF0 | |

## BASIC ENTRY POINTS

| | | | |
|---|---|---|---|
| D393 | DB87 | E793 | F070 |
| D39A | DD67 | E7A0 | F09E |
| D3D6 | DD6A | E7A7 | F10B |
| D410 | DD6C | E7AA | F114 |
| D412 | DD6D | E7BE | F128 |
| D419 | DD7B | E7C1 | F1D8 |
| D43C | DE59 | E913 | F1E1 |
| D43F | DE60 | E92D | F1E8 |
| D4F2 | DE6C | E941 | F1EC |
| D52C | DE81 | E97F | F1FD |
| D52E | DEB8 | E982 | F220 |
| D539 | DEBE | E9E3 | F228 |
| D553 | DEC9 | EA39 | F235 |
| D566 | DED8 | EA50 | F244 |
| D61A | DEE9 | EA55 | F252 |
| D64B | DFE3 | EA5E | F259 |
| D665 | E000 | EA66 | F265 |
| D66C | E003 | EA69 | F273 |
| D674 | E053 | EAF9 | F277 |
| D683 | E07D | EB1E | F280 |
| D697 | E108 | EB21 | F318 |
| D7D2 | E10C | EB23 | F328 |
| D810 | E169 | EB2B | F390 |
| D820 | E199 | EB53 | F399 |
| D823 | E2F2 | EB63 | F3D8 |
| D849 | E301 | EB66 | F3E2 |
| D850 | E3CF | EB72 | F3E4 |
| D858 | E3D5 | EB82 | F3EA |
| D863 | E3DD | EB90 | F3F2 |
| D865 | E3E7 | EB93 | F3F4 |
| D898 | E3E9 | EBAC | F3F6 |
| D93E | E3ED | EBAF | F411 |
| D941 | E423 | EBB2 | F417 |
| D995 | E42A | EBF2 | F457 |
| D998 | E452 | EC23 | F45A |
| D9A3 | E484 | EC4A | F465 |
| D9A6 | E5D4 | ED19 | F53A |
| DA0C | E5E2 | ED24 | F5CB |
| DA46 | E5FD | ED2E | F601 |
| DA52 | E600 | ED34 | F605 |
| DA65 | E604 | ED43 | F65D |

| | | | |
|---|---|---|---|
| DA7B | E635 | EE5F | F661 |
| DA9A | E680 | EE8D | F6B9 |
| DAA1 | E6F5 | EE97 | F6EC |
| DAFB | E6F8 | EED0 | F6F0 |
| DB16 | E6FB | EF09 | F6F6 |
| DB3A | E6FE | EFAE | F730 |
| DB3D | E746 | EFB4 | F7D9 |
| DB57 | E74C | EFEA | F7EA |
| DB5A | E752 | EFF1 | |
| DB5C | E77E | F03A | |

80-COLUMN ENTRY POINTS

| | | |
|---|---|---|
| C300 | C30B | C800 |
| C305 | C311 | C84D |
| C307 | C314 | C9AA |

PRINTER ENTRY POINTS

| | | |
|---|---|---|
| C100 | C105 | C107 |

DISK BOOT ENTRY POINTS

| | |
|---|---|
| C600 | C65C |

## I/O SLOT ARRANGEMENT

Like the IIe, the ACE has eight logical slots (0-7.) However, it only has three physical slots (slots where you can plug in a peripheral card.) The remaining five slots are either dedicated to internal functions or are unavailable. Here's a breakdown of the slot arrangement:

| Slot | Type | Usage |
|---|---|---|
| 0 | logical | internal 16K "language card" RAM |
| 1 | logical | internal parallel printer, jumper provided to disable |
| 2 | physical | inside unit, towards back |
| 3 | logical | internal 80 column card |
| 4/7 | physical | inside unit, towards front, jumper selectable for slot 4 or 7 |
| 5 | physical | outside unit, on right side |
| 6 | logical | internal disk controller, jumper provided to disable |

## ENHANCEMENTS

The following sections describe features of the ACE which are enhancements over the IIe.

## SPECIAL KEYS

Special keys are those keys which are on the ACE keyboard but which are not found on a IIe. (All keys on the ACE keyboard which are also on the IIe keyboard produce the same codes as the IIe and are read by software the same way in both machines.) The special keys include the function keys in the top row of the keyboard and the command keys in the numeric pad area of the keyboard.

## KEYCODES FOR SPECIAL KEYS

The following chart shows the hex keycodes produced by the special keys:

| Key | Normal | CTRL | SHIFT | CTRL-SHIFT | ALT |
|---|---|---|---|---|---|
| F1 | 20 | 2C | 38 | 44 | 50 |
| F2 | 21 | 2D | 39 | 45 | 51 |
| F3 | 22 | 2E | 3A | 46 | 52 |
| F4 | 23 | 2F | 3B | 47 | 53 |
| F5 | 24 | 30 | 3C | 48 | 54 |
| F6 | 25 | 31 | 3D | 49 | 55 |
| F7 | 26 | 32 | 3E | 4A | 56 |
| F8 | 27 | 33 | 3F | 4B | 57 |
| F9 | 28 | 34 | 40 | 4C | 58 |
| F10 | 29 | 35 | 41 | 4D | 59 |
| F11 | 2A | 36 | 42 | 4E | 5A |
| F12 | 2B | 37 | 43 | 4F | 5B |
| INSC | 00 | 00 | 00 | 00 | 00 |
| CLRL | 01 | 01 | 01 | 01 | 01 |
| CPES | 02 | 02 | 02 | 02 | 02 |
| CLRS | 03 | 03 | 03 | 03 | 03 |
| HOME | 04 | 04 | 04 | 04 | 04 |
| DELC | 05 | 05 | 05 | 05 | 05 |
| RUN | 06 | 06 | 06 | 06 | 06 |
| LIST | 1F | 1F | 1F | 1F | 1F |

## READING THE SPECIAL KEYS IN SOFTWARE

As you can see by the above chart, the special keys produce keycodes in the range of 00-5Bh. The standard Apple keyboard convention only provides for keycodes in the range of 80-FFh. To handle the special keys, there is a softswitch you can read to determine if the key is a normal key or a special key. This softswitch is at C027h and returns a positive value if the key is a special key. The following code fragment is an example of how to read the keyboard to handle the special keys correctly:

```
READKEY   LDA   $C000     see if there's a key waiting
          BPL   NOKEY     branch if there's no key
          BIT   $C010     clear the keyboard strobe
          BIT   $C027     is it a special key ?
          BPL   SPECIAL   yes - is a special

NORMAL    ...   ...       else do your normal key handling here

SPECIAL   AND   #$7F      is special - clear bit 7 (strobe)
                          the special keycode 00-5B is now in
                            the accumulator
```

## FUNCTION KEYS

The ACE contains twelve user-definable function keys. The user may assign any definitions he likes to these twelve keys using the BASIC FKEY statement (described earlier.) The total number of user definable characters is 233. These 233 characters may be allocated to the twelve function keys in any combination.

The function key definitions are volatile. This doesn't only mean that they will be lost when the power is turned off. It also means that certain application programs may destroy the definitions, depending on the uses the application makes of system memory. The firmware that handles the function keys continuously checks the validity of the definitions every time a function key is pressed. If an application program has wiped out the definitions, then the firmware treats the function keys as all undefined.

Because the system is not always under firmware control, function keys are not always available. This means that function keys won't work in application programs that read the keyboard hardware directly. When such applications are running, the function keys will produce "garbage" keycodes instead of their intended definitions. The function keys will always work correctly when FDOS, BASIC, or the MONITOR are executing.

The function keys will also produce "garbage" keycodes if they're pressed in combination with the CTRL or ALT keys. To work correctly, the function keys should be pressed only by themselves - not in combination with any other key.

The function key definitions are stored in the stack page in the auxiliary memory bank. The chances of an application program using this area are remote, but it's not impossible. To guard against this possibility, an additive checksum is kept of the definition area (from 100-1FFh). Each time a function key is pressed, the system first checks the validity of the checksum. If it's still valid, then the definition is used. If it's not valid (someone has wiped out the definitions), then all function keys will return there normal single keycodes until they are redefined.

All system function key handling (as well as "hard" key handling) is performed by four routines in the 80 column expansion code. The routines are SETFKCS, FKINIT, KEYTST and KEYGET, and each one has an entry point in the 80 column slot code area (SETFKCS = C3E8, FKINIT = C3EE, KEYTST = C3F4, and KEYGET = C3FA.) The routines are used by other firmware in the system as follows: (These routines are described in detail in the next section.)

SETFKCS is used by BASIC to recalculate the checksum of the definition area every time a definition is changed. The checksum is kept in a slot one permanent "screenhole" at 4F9h.

FKINIT is used by the Monitor to initialize all of the definitions to empty - one space each. This is done on every coldstart (or forced disk boot with "Open F Reset").

KEYTST is used by the Monitor and the 80 column firmware to determine if any keycode is waiting. This could be either a hard key press or the next character of a function key definition. KEYTST returns carry set if a keycode is waiting.

KEYGET is used by the Monitor and the 80 column firmware to read the next keycode. KEYGET assumes that a key is waiting (KEYTST must return with the carry set before calling KEYGET) and returns that next key (either hard or the next character of a function key definition) in reg A. (Hard keypresses are ignored until the end of any currently in process function key definition is reached.)

## FIRMWARE KEYBOARD ROUTINES

There are several routines in the ACE firmware that will automatically handle special keys, function keys, and normal keys correctly for you. If the user has defined any function keys using the FKEY statement in BASIC, then these routines are designed to correctly return either normal keycodes or the keycodes in the defined function key strings if a function key is pressed. The routines are as follows:

| Routine | Address | Function |
| --- | --- | --- |
| KEYTST | C3F4h | Determines if a key is waiting. If a key is waiting the carry flag is set on exit. A "key" may be a hard key press or a keycode from a function key definition string. |
| KEYGET | C3FAh | Returns the keycode of a waiting key in the accumulator. Assumes a key is waiting (KEYTST has returned with carry set.) |
| FKINIT | C3EEh | Initializes all of the function key definitions to "empty." |
| SETFKCS | C3E8h | Calculates the function key string storage area checksum. |

These routines automatically handle function key presses for you. If the function key has been defined, then using KEYTST and KEYGET will result in your reading all of the keycodes of the function key definition string. If a function key is undefined, KEYGET will return a space code (A0h) for that key.

KEYGET will also perform the following translations on the special keys in the numeric pad area before returning the keycode:

| Key | Translated to keycode |
|---|---|
| CLRL | 9A - CTRL Z |
| CLRS | 8C - CTRL L |
| HOME | 99 - CTRL Y |
| RUN | 2C - pseudo function key 13 |
| LIST | 2D - pseudo function key 14 |

The routines FKINIT and SETFKCS are for advanced users who would like to define their own function key definitions directly in assembly language without using the BASIC FKEY statement. Function key strings are kept in aux memory from 100-1FFh. For each function key 1-12 and for the special keys RUN and LIST there is a definition string in the function key storage area. Each definition string is ended with a 00 byte.

To insure that the function key storage area contains valid function key definitions, the firmware keeps a checksum of that area which must be recalculated every time a definition is changed. If a function key is pressed the checksum is again recalculated. If the checksum doesn't match its stored value, the firmware assumes the function key storage area has been wiped out and all of the function keys are treated as undefined.

So, to define your own function keys from assembly language, first place a definition string for each key (F1-F12, RUN, and LIST) in the function key storage area. End each string with a 00 byte. Then call SETFKCS to recalculate the checksum. That's it. KEYTST and KEYGET will now operate correctly using your definitions. Also, don't forget to call SETFKCS again every time you change a definition so that the checksum matches the new definitions.

## FRONT PANEL LIGHTS

The ACE has four LED's on its front panel which are labeled **DIAG**, **DISK**, **HIRES**, and **CPU**. These are controlled by a combination of hardware and software as follows:

| | | |
|---|---|---|
| **DIAG** | S/W | C058h = ON<br>C059h = OFF |
| **DISK** | S/W | C05Ah = ON<br>C05Bh = OFF |
| **HIRES** | S/W | C05Ch = ON<br>C05Dh = OFF |
| **CPU** | H/W | toggles ON/OFF with accesses to the I/O space at C000-C0FFh |

## EXPANSION RAM CARD

Franklin offers an expansion RAM card as optional equipment for the ACE. The card can contain from 64K to 512K of expansion RAM, depending on how it's configured. The card plugs into the auxiliary connector on the motherboard, replacing the standard 64K 80-column card.

There are two rows of RAM chip sockets on the card. These are designated as **Row 0** and **Row 1**. Each of these rows contains sockets for eight RAM chips. Each row may be populated with either 64K or 256K RAM chips. All of the chips in a row must be the same size. Row 0 must always be populated with either 64K or 256K chips while row 1 may be left empty if desired. If any socket in a row is populated then all of the sockets in that row must be populated - you can have either zero or eight chips in a row. Here's a chart showing all of the legal configurations along with the total size of expansion memory provided by each one:

| Row 0 | Row 1 | Total expansion memory size |
|---|---|---|
| 64K | empty | 64K |
| 64K | 64K | 128K |
| 64K | 256K | 320K |
| 256K | empty | 256K |
| 256K | 64K | 320K |
| 256K | 256K | 512K |

Note that **row 0 must always be populated** or the machine will not function correctly as a IIe compatible!

BANKS

The expansion RAM card is organized into **banks** of memory. This is because the expansion RAM card is not a contiguous block of RAM but is broken up into 64K sections. Each 64K section is called a bank. At any particular time, only one 64K bank is available for use. Thus, software using the expansion memory must continually manage all of the available banks correctly so that the desired 64k bank is selected at the right time.

Since the card can contain from 64K to 512K of RAM, it can contain from one to eight banks, because each bank contains 64K of RAM.

SOFTWARE CONTROL OF THE CARD

The expansion RAM card is configurable by jumpers into three different modes of operation: CPU Banking Only, CPU/Video Combined Banking, and CPU/Video Separate Banking. The three modes operate as follows:

CPU Banking Only - bank selection only affects CPU accessing, video accessing is always to bank 0.

CPU/Video Combined Banking - bank selection affects both CPU and video accessing (both CPU and video access the selected bank.)

CPU/Video Separate Banking - CPU and video banks are separately selectable using the higher half of the selection register for video bank selection and the lower half for CPU bank selection.

This discussion only describes the CPU Banking Only mode of operation.

The expansion RAM card is partitioned into a collection of 64K banks. By selecting a particular bank, you cause that bank to be enabled as the auxiliary memory bank in the ACE. Which banks are actually present depends on the configuration of the card as follows:

| Bank reg. value = | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Configuration: | | | | | | | | | | | | | | | | |
| row 0 = empty<br>row 1 = 64K | - | - | - | 1 | - | - | - | 1 | - | - | - | 1 | - | - | - | 1 |
| row 0 = empty<br>row 1 = 256K | - | - | - | 1 | - | - | - | 2 | - | - | - | 3 | - | - | - | 4 |
| row 0 = 64K<br>row 1 = empty | 0 | - | - | - | 0 | - | - | - | 0 | - | - | - | 0 | - | - | - |
| row 0 = 64K<br>row 1 = 64K | 0 | - | - | 1 | 0 | - | - | 1 | 0 | - | - | 1 | 0 | - | - | 1 |
| row 0 = 64K<br>row 1 = 256K | 0 | - | - | 1 | 0 | - | - | 2 | 0 | - | - | 3 | 0 | - | - | 4 |
| row 0 = 256K<br>row 1 = empty | 0 | - | - | - | 1 | - | - | - | 2 | - | - | - | 3 | - | - | - |
| row 0 = 256K<br>row 1 = 64K | 0 | - | - | 1 | 2 | - | - | 1 | 3 | - | - | 1 | 4 | - | - | 1 |
| row 0 = 256K<br>row 1 = 256K | 0 | - | - | 1 | 2 | - | - | 3 | 4 | - | - | 5 | 6 | - | - | 7 |

Use the above table as follows: Find the memory configuration of the board in the list of configurations down the left-hand side. The number under a particular bank register value is a reference number for the actual 64K bank that will be selected if that value is placed into the bank register. If 64K chips are installed, the same actual bank is selected for any of four bank register values - any of those four values select the same actual bank. The reference numbers used to identify actual banks were selected for convenience and have no special significance.

The bank register is located at C071h, C073h, C075h, and C077h. Writing a value to any of these locations will select the bank specified by the value written.

Note that when 64K chips are installed in either of the rows, four different bank register values will all select the same physical 64K bank. This means that any software which attempts to detect which physical banks are actually present must take this quadruple replication of 64K banks into account during the detection process.

THE EXPANSION MEMORY FIRMWARE SUPPORT ROUTINE

The ACE has a firmware routine that facilitates use of the memory on the expansion RAM card. This routine is called **EXPMOVE** and is located at address C3DCh. EXPMOVE is an enhancement to the standard AUXMOVE routine located at C311h. They both allow memory moves between main memory and auxiliary memory, but EXPMOVE has capabilities that the standard AUXMOVE routine doesn't have. In particular, AUXMOVE doesn't allow for moves to or from the language card memory area at D000-FFFFh. EXPMOVE provides this capability.

EXPMOVE has two modes of operation: block move or page move. Block move allows for a move of any number of bytes. Page move only allows moving a number of bytes which is a multiple of a page size (256 bytes.) The block move mode is thus more general purpose while the page move mode is much faster in operation. The mode is selected by the contents of the X register on entry to the routine as shown in the following parameter description:

Block move input parameters:

| | | |
|---|---|---|
| 3C,3D | = | source start address |
| 3E,3F | = | source end address |
| 42,43 | = | destination start address |
| reg X | = | zero |
| carry clear | = | move from aux to main |
| carry set | = | move from main to aux |

Page move input parameters:

| | | |
|---|---|---|
| 3C,3D | = | source start address |
| 42,43 | = | destination start address |
| reg X | = | number of pages to move |
| carry clear | = | move from aux to main |
| carry set | = | move from main to aux |

Both modes of operation make the following assumptions:

* The language card is properly enabled for read or write as required.

* Main memory stack, zero page, and language card areas are enabled on entry (softswitch C008h.)

EXPMOVE will automatically turn off 80STORE (softswitch C000h) before the move is performed. On exit from EXPMOVE, 80STORE will be restored to the state it was in on entry to EXPMOVE. EXPMOVE can be called from either main or aux memory.

## IDENTIFYING AN ACE

If your program has to distinguish between an ACE and an Apple, you can use the eight bytes at location FB09h to do so. These eight bytes are guaranteed to be equal to hexadecimal CC, C1, CE, C7, C1, D2, D4, BB in an ACE 2XXX. The corresponding bytes in an Apple spell out the machine type in the sign-on message.

## COMPATIBILITY GUIDELINES

If you're writing a new program and you'd like to be sure it runs on an ACE as well as on an Apple IIe, here are some guidelines you should follow:

1.) If your program is in assembly language, don't use any ROM calls. This will guarantee compatibility.

2.) If your program is in assembly language and you have to use ROM calls, then restrict your usage of the ROM to the ACE-supported entry points defined earlier. This almost guarantees compatibility, but you'll still have to test to make sure there are no unexpected side effects or register return value quirks in your usage of the entry points.

Also, if you use ROM entry points, don't make any assumptions about the usage of page zero or screenhole variables without first confirming your assumptions by looking at the routines that you use. Many of the ACE page zero and screenhole variables are compatible with those on the IIe, but some are not. In particular, some of the 80-column screenholes are different.

3.) If your program is written in BASIC, then try not to run at the memory limit of an Applesoft program. Back down a little if you're really tight. This will almost guarantee compatibility.

4.) If your program is written in BASIC and it calls assembly language routines, then points 1 and 2 above apply also to those assembly routines. One particular problem we've found with assembly routines that interface to the BASIC interpreter assembly routines is in the area of string temporaries. Franklin's BASIC does support a number of the string handling assembly routines provided by Applesoft, but the internal workings of those routines are very different from Applesoft. In particular, the temporary string stack is not stored or controlled the same way - the page zero variables controlling temporary strings are very different in the ACE. Try to stay away from accessing Applesoft page zero variables.

5.) Under no circumstances should you perform ROM checksums or other tests of the ROM contents as a measure of copy protection. You shouldn't even test the interrupt and I/O vectors for jumps to particular addresses because the vectors in the ACE may be different from those in a IIe and are not guaranteed to remain the same.

6.) Be careful with copy protection schemes that really push the disk drive hardware to its limits or beyond. The Franklin disk controller chip is not 100% identical to Apple's nor are the disk drives necessarily identical to Apple's. If you're really pushing it, you should test your scheme adequately on several ACE's to make sure it boots reliably.

If you follow these simple guidelines, there is no reason why your program won't run on an ACE as well as on a IIe.

## SYSTEM MEMORY AND I/O MAP

This section lists some of the important locations in the memory address space. Some of these are memory locations used for variables and buffers and others are I/O locations used as softswitches.

| Hex Address | Decimal Address | Name | Description |
|---|---|---|---|
| 0-2 | 0-2 | | jmp to BASIC restart |
| 3-5 | 3-5 | | jmp to BASIC STROUT routine |
| 6-9 | 6-9 | | not used |
| A-C | 10-12 | | jmp to BASIC user address |
| D-1F | 12-31 | | BASIC temporaries & working variables |
| 20 | 32 | LFTMAR | left column of screen window [0...79] |
| 21 | 33 | WIDTH | width of screen window [1...80-LFTMAR] |
| 22 | 34 | TOPLIN | top line of screen window [0...22] |
| 23 | 35 | BOTLIN | bottom screen window line [[TOPLIN+1...23] (last line plus one) |
| 24 | 36 | COL | current column no. relative to starting left column of window [0...WIDTH-1] |
| 25 | 37 | ROW | current line no. [0...23] |
| 26,27 | 38,39 | GP | ptr to start of current lores line |
| 28,29 | 40,41 | TP | ptr to start of current text line |
| 2A | 42 | | Monitor temporary |
| 2B | 43 | | Monitor temporary |
| 2C | 44 | HEND | ending column no. for lores horizontal line draw routine in Monitor |
| 2D | 45 | VEND | ending line no. for lores vertical line draw routine in Monitor |
| 2E | 46 | NBMASK | nibble mask used by Monitor lores graphic routines |
| 2F | 47 | INSLEN | decoded instruction length in Monitor |
| 30 | 48 | COLOR | color nibbles used by Monitor lores graphic routines |
| 31 | 49 | | not used |
| 32 | 50 | VIDMSK | video flag: FF = normal, 7F = flashing (letters), 3F = inverse |
| 33 | 51 | PRMTCH | current input line prompt character |
| 34,35 | 52,53 | | not used |
| 36,37 | 54,55 | OUTVEC | character output vector |
| 38,39 | 56,57 | INPVEC | character input vector |
| 3A,3B | 58,59 | BRKPC | PC where a BRK occurred |
| 3C-43 | 60-67 | | arguments to Monitor and expansion memory move routines |
| 44 | 68 | | not used |
| 45 | 69 | RA | saved A register |
| 46 | 70 | RX | saved X register |
| 47 | 71 | RY | saved Y register |
| 48 | 72 | RP | saved status register |
| 49 | 73 | RS | saved stack register |
| 4A-4D | 74-77 | | not used |
| 4E,4F | 78,79 | RANDOM | random number seed |

**Some important BASIC zero page variables:**

| | | | |
|---|---|---|---|
| 67,68 | 103,104 | BEGPRG | ptr to beginning of program |
| 69,6A | 105,106 | BEGVAR | ptr to beginning of simple variable table |
| 6B,6C | 107,108 | BEGARY | ptr to beginning of array variable table |
| 6D,6E | 109,110 | BEGFRE | ptr to beginning of free space |
| 6F,70 | 111,112 | BEGSTR | ptr to beginning of string storage |
| 73,74 | 115,116 | ENDSTR | ptr to end of string storage |
| 75,76 | 117,118 | CURLNO | current line no. or FFXX if in direct mode |
| 77,78 | 119,120 | OLDLINE | address of last line executed |
| B8,B9 | 184,185 | TP | ptr to current program statement |

**End of BASIC zero page variables**

| | | | |
|---|---|---|---|
| 200-2FF | 512-767 | INPBUF | keyboard input buffer |

**If FDOS is loaded:**

| | | |
|---|---|---|
| 3D0-3D2 | 976-978 | jmp to FDOS warmstart |
| 3D3-3D5 | 979-981 | jmp to FDOS coldstart |
| 3D6-3D8 | 982-984 | jmp to FDOS File Manager subroutine |
| 3D9-3DB | 985-987 | jmp to FDOS RWTS subroutine |
| 3DC-3E2 | 988-994 | subroutine returning pointer to File Manager parameter list |
| 3E3-3E9 | 995-1001 | subroutine returning pointer to RWTS IOB |
| 3EA-3EC | 1002-1004 | subroutine to reconnect standard FDOS I/O vectors |
| 3ED,3EE | 1005,1006 | two NOP instructions |
| 3EF | 1007 | JMP instruction (for following break vector) |

**End of FDOS stuff**

| | | |
|---|---|---|
| 3F0,3F1 | 1008,1009 | break vector |
| 3F2,3F3 | 1010,1011 | reset vector |
| 3F4 | 1012 | power-up flag (= contents of 3F3 XOR #A5) |
| 3F5-3F7 | 1013-1015 | jmp to & routine handler |
| 3F8-3FA | 1016-1018 | jmp to **CTRL Y** routine handler |
| 3FB-3FD | 1019-1021 | jmp to NMI handler |
| 3FE,3FF | 1022,1023 | IRQ vector |
| 400-7FF | 1024-2047 | text/lores screen memory, primary page (screenholes also used for scratchpad) |

**Printer screenholes:**

| | | | |
|---|---|---|---|
| 479 | 1145 | FKPTR | index to next function key character |
| 4F9 | 1273 | FKCSUM | checksum of function key definition RAM |
| 579 | 1401 | FKFLAG | zero if not in a function key string |
| 5F9 | 1529 | DMPFLG | screen dump control flags:<br>bit 7,6,5 = 000 = EPSON<br>= 010 = OKIDATA<br>= 100 = C. ITOH<br>= 110 = GEMINI<br>bit 3 = 1 = rotate 90 degrees<br>bit 2 = 1 = inverse image<br>bit 1 = 1 = secondary page<br>bit 0 = 1 = hires screen, = 0 = text |
| 679 | 1657 | | temporary |
| 6F9 | 1785 | | temporary |

| | | | |
|---|---|---|---|
| 779 | 1913 | FLAGS | misc. flags:<br>bit 7 = 1 = auto linefeed<br>bit 6 = 1 = in command mode<br>bit 5 = 1 = strip bit 7<br>bit 4 = 1 = screen echo<br>bit 3 = 1 = changing cmd designator char<br>bit 2 = 1 = in transparent mode |
| 7F9 | 2041 | LSTCOL | last column no. printed |

**End of printer screenholes.**

**80-column screenholes:**

| | | | |
|---|---|---|---|
| 47B | 1147 | ESCCHAR | saved char under escape cursor |
| 4FB | 1275 | FLAGS | misc. flags:<br>bit 7 = 1 = in escape mode<br>bits 5,4 = 11 = initialized<br>bits 1,0 = gotoxy counter |
| 57B | 1403 | CURCOL | current screen column |
| 67B | 1659 | PASCBYTE | PASCAL I/O byte |
| 77B | 1915 | OLDTPL | saved line ptr (low byte) |
| 7FB | 2043 | OLDTPH | saved line ptr (high byte) and temporary |

**End of 80-column screenholes**

| | | |
|---|---|---|
| 800-BFF | 2048-3071 | text/lores screen memory, secondary page |
| 2000-3FFF | 8192-16383 | hires screen memory, primary page |
| 4000-5FFF | 16384-24575 | hires screen memory, secondary page |

**If FDOS is loaded:**

| | | |
|---|---|---|
| ????-BCFF | ????-48383 | FDOS file buffers (253h per file) |
| BD00-BDFF | 48384-48639 | FDOS catalog buffer |
| BE00-BEFF | 48640-48895 | FDOS VTOC buffer |
| BF00-BFFF | 48896-49151 | FDOS system page - see FDOS Tech Manual |

**End of FDOS stuff**

**Softswitches:**

| | | | |
|---|---|---|---|
| C000 | 49152 | KBDDAT | (R) keyboard data register: negative if key waiting, bits 6-0 are keycode |
| | | OFF80ST | (W) causes MAINRD, MAINWR, AUXRD, and AUXWR to control the screen display pages (otherwise PAGE1 and PAGE2 control the screen pages) |
| C001 | 49153 | ON80ST | (W) causes PAGE1 and PAGE2 to control the screen display pages (otherwise MAINRD, MAINWR, AUXRD, and AUXWR control the screen pages) |
| C002 | 49154 | MAINRD | (W) selects the main RAM bank for reading; if ON80ST is on, then this switch has no effect on the screen display pages |
| C003 | 49155 | AUXRD | (W) selects the aux RAM bank for reading; if ON80ST is on, then this switch has no effect on the screen display pages |

| | | | |
|---|---|---|---|
| C004 | 49156 | MAINWR | (W) selects the main RAM bank for writing; if ON80ST is on, then this switch has no effect on the screen display pages |
| C005 | 49157 | AUXWR | (W) selects the aux RAM bank for writing; if ON80ST is on, then this switch has no effect on the screen display pages |
| C006 | 49158 | SLOTCX00 | (W) enables slot ROM at CX00 |
| C007 | 49159 | INTCX00 | (W) enables internal ROM at CX00 |
| C008 | 49160 | MAINSTZP | (W) selects zero page, stack, and language card areas in the main RAM bank (MAINRD, AUXRD, MAINWR, and AUXWR have no effect on these areas) |
| C009 | 49161 | AUXSTZP | (W) selects zero page, stack, and language card areas in the aux RAM bank (MAINRD, AUXRD, MAINWR, and AUXWR have no effect on these areas) |
| C00A | 49162 | INTC300 | (W) enables internal ROM at C3XX |
| C00B | 49163 | SLOTC300 | (W) enables slot ROM at C3XX |
| C00C | 49164 | OFF80C | (W) turns off 80 column video |
| C00D | 49165 | ON80C | (W) turns on 80 column video |
| C00E | 49166 | STDCHSET | (W) selects normal character set |
| C00F | 49167 | ALTCHSET | (W) selects alternate character set |
| C010 | 49168 | KBDCLR | (R) clear key waiting bit from KBDDAT; also returns any-key-down flag (negative if key down) |
| C011 | 49169 | STALCBK2 | (R) negative if LC bank 2 is enabled, else bank 1 is enabled |
| C012 | 49170 | STALCRAM | (R) negative if LC RAM area is enabled, else LC ROM area is enabled |
| C013 | 49171 | STARAMRD | (R) negative if AUXRD is on, else MAINRD is on |
| C014 | 49172 | STARAMWR | (R) negative if AUXWR is on, else MAINWR is on |
| C015 | 49173 | STASLTCX | (R) negative if INTCX00 is on, else SLOTCX00 is on |
| C016 | 49174 | STASTZP | (R) negative if AUXSTZP is on, else MAINSTZ is on |
| C017 | 49175 | STASLTC3 | (R) negative if SLOTC300 is on, else INTC300 is on |
| C018 | 49176 | STA80ST | (R) negative if ON80ST is on |
| C019 | 49177 | STAVBL | (R) negative if vertical blanking |
| C01A | 49178 | STATEXT | (R) negative if TEXT is on, else GRAFIX is on |
| C01B | 49179 | STAMIXED | (R) negative if MIXTXT is on, else ALLTXT is on |
| C01C | 49180 | STAPAGE2 | (R) negative if PAGE2 is on, else PAGE1 is on |
| C01D | 49181 | STAHIRES | (R) negative if HIRES is on, else LORES is on |
| C01E | 49182 | STACHSET | (R) negative if ALTCHSET is on, else STDCHSET is on |
| C01F | 49183 | STA80C | (R) negative if ON80C is on |
| C022 | 49186 | STAABROM | (R) negative if ABROMON is on |
| C027 | 49191 | SPCKEY | (R) negative if not a special key, positive if a special key |
| C030 | 49200 | SPEAKR | (R) speaker toggle |
| C040 | 49216 | | (R) strobe output |
| C050 | 49232 | GRAFIX | (R) selects graphics screen mode |
| C051 | 49233 | TEXT | (R) selects text screen mode |
| C052 | 49234 | ALLTXT | (R) set screen mode to all text or all graphics |
| C053 | 49235 | MIXTXT | (R) set screen mode to text mixed with graphics |

| | | | |
|---|---|---|---|
| C054 | 49236 | PAGE1 | (R) selects primary screen page |
| C055 | 49237 | PAGE2 | (R) selects secondary screen page |
| C056 | 49238 | LORES | (R) selects lores graphics mode |
| C057 | 49239 | HIRES | (R) selects hires graphics mode |
| C058 | 49240 | | (R) annunciator 0 off, DIAG light on |
| C059 | 49241 | | (R) annunciator 0 on, DIAG light off |
| C05A | 49242 | | (R) annunciator 1 off, DISK light on |
| C05B | 49243 | | (R) annunciator 1 on, DISK light off |
| C05C | 49244 | | (R) annunciator 2 off, HIRES light on |
| C05D | 49245 | | (R) annunciator 2 on, HIRES light off |
| C05E | 49246 | | (R) annunciator 3 off |
| C05F | 49247 | | (R) annunciator 3 on |
| C061 | 49249 | SW1 | (R) switch input 1 (negative if OPEN F pressed) — 255 on |
| C062 | 49250 | SW2 | (R) switch input 2 (negative if CLOSED F pressed) - 255 on |
| C063 | 49251 | SW3 | (R) switch input 3 |
| C064 | 49252 | POTS | (R) analog input 1 |
| C065 | 49253 | | (R) analog input 2 |
| C066 | 49254 | | (R) analog input 3 |
| C067 | 49255 | | (R) analog inpt 4 |
| C070 | 49264 | POTCLR | (R) analog input clear strobe |
| C071 | 49265 | | (W) expansion RAM bank select register |
| C073 | 49267 | | (W) expansion RAM bank select register |
| C075 | 49269 | | (W) expansion RAM bank select register |
| C077 | 49271 | | (W) expansion RAM bank select register |
| C078 | 49272 | ABROMOFF | (W) enable RAM at A000-BFFF |
| C079 | 49273 | ABROMON | (W) enable ROM at A000-BFFF |

**From C080-C08F controls the language card RAM/ROM area:**

| | | |
|---|---|---|
| C080 | 49280 | (R/W) select RAM bank 2 for reading |
| C081 | 49281 | (R/W) select ROM for reading; two successive accesses also selects RAM bank 2 for writing |
| C082 | 49282 | (R/W) select ROM for reading |
| C083 | 49283 | (R/W) two successive accesses selects RAM bank 2 for reading and writing |
| C084-C087 | 49284-49287 | repeat of C080-C083 |
| C088 | 49288 | (R/W) select RAM bank 1 for reading |
| C089 | 49289 | (R/W) select ROM for reading; two successive accesses also selects RAM bank 1 for writing |
| C08A | 49290 | (R/W) select ROM for reading |
| C08B | 49291 | (R/W) two successive accesses selects RAM bank 1 for reading and writing |
| C08C-C08F | 49292-49295 | repeat of C088-C08B |

**End of language card area controls**

| | | | |
|---|---|---|---|
| C090 | 49296 | PRTDATA | (W) printer port data output register |

**From COEO-COEF controls the floppy disk controller chip:**

| | | |
|---|---|---|
| COE0 | 49376 | stepper motor phase 0 off |
| COE1 | 49377 | stepper motor phase 0 on |
| COE2 | 49378 | stepper motor phase 1 off |
| COE3 | 49379 | stepper motor phase 1 on |
| COE4 | 49380 | stepper motor phase 2 off |
| COE5 | 49381 | stepper motor phase 2 on |
| COE6 | 49382 | stepper motor phase 3 off |
| COE7 | 49383 | stepper motor phase 3 on |
| COE8 | 49384 | turn drive motor off |
| COE9 | 49385 | turn drive motor on |
| COEA | 49386 | select drive one |
| COEB | 49387 | select drive two |
| COEC | 49388 | disk data byte read register; write byte strobe |
| COED | 49389 | select write protect sense mode; disk data byte write register |
| COEE | 49390 | enable disk read; MSB set if write protected |
| COEF | 49391 | enable disk write |

**End of floppy disk controller chip controls**

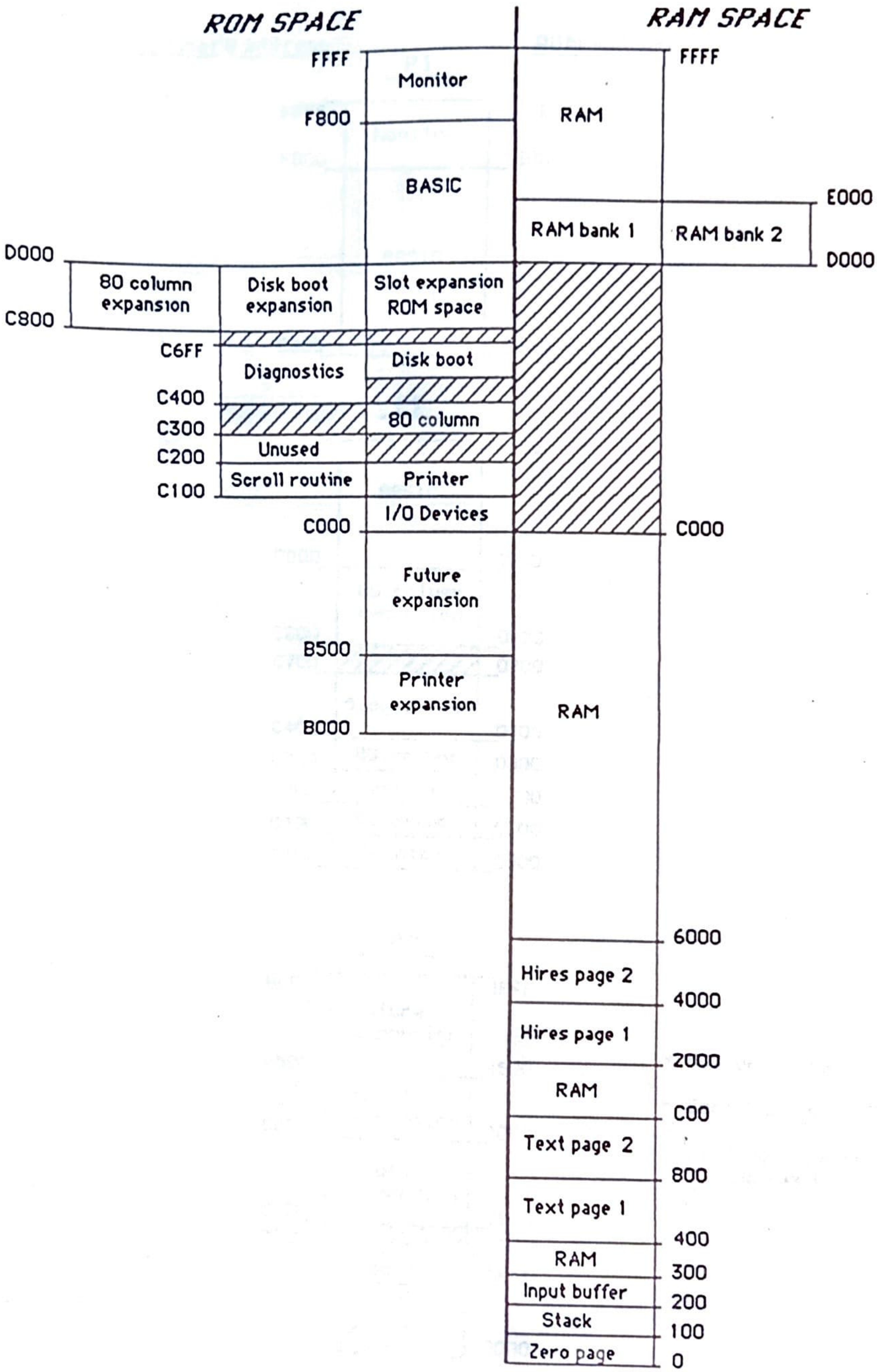
**The printer-busy register is in the printer ROM space:**

| | | | |
|---|---|---|---|
| C1C1 | 49601 | PRTBUSY | (R) negative if printer busy |

**End of softswitches**

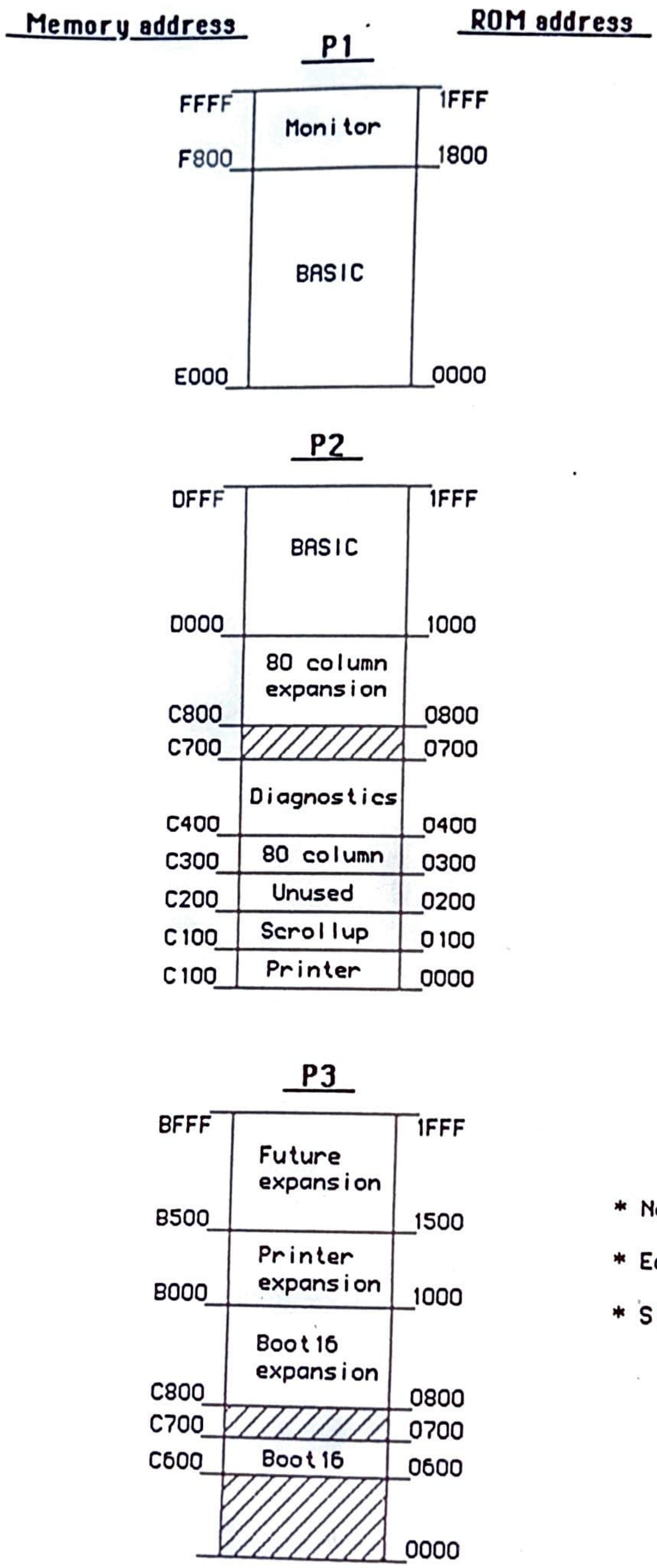
# FCC ACE 2X00 MEMORY MAP

Version 1.2



# FCC ACE 2X00 ROM MAP

Version 1.0




* Not drawn to scale
* Each ROM is a 2764 EPROM
* Slanted line areas can't be used

## DISK DRIVE REFERENCE

## 1.2 SPECIFICATIONS

Specifications of Diskette Drive FD-100 series are listed in Table 1-1.

Table 1-1 Specifications

| | |
|---|---|
| **PERFORMANCE SPECIFICATIONS:** | |
| Recording method | GCR |
| Capacity | |
| unformatted- | |
| per disk: | 250K bytes |
| per track: | 6.25K bytes |
| formatted (soft sectoring)- | |
| per disk: | 143.4K bytes |
| per track: | 4096 bytes |
| per sector: | 256 bytes |
| sector per track: | 16 |
| Transfer Rate | 250K bits/sec |
| Latency (average): | 100ms |
| Access Time | |
| track to track: | 6ms |
| average: | 93ms |
| settling time: | 15ms |
| Disk Motor Start Time: | 0.5sec |
| **FUNCTIONAL SPECIFICATIONS:** | |
| Rotational Speed: | 300rpm |
| Recording Density(inner track): | 5162bpi |
| Flux Density(inner track): | 5162fci |
| Track Density: | 48tpi |
| Tracks: | 35 (MAX 40) |
| Media: | Industry Compatible 5¼ inch diskette |

Table 1-1 Specifications (continuted)

| | |
|---|---|
| **PHYSICAL SPECIFICATIONS :** | |
| Environmental Limits | |
| Ambient temperature (exclusive of media) | |
| operating : | $5^\circ C$ to $45^\circ C$ ($41^\circ F$ to $113^\circ F$) |
| non-operating : | $-22^\circ C$ to $47^\circ C$ ($-8^\circ C$ to $125^\circ C$) |
| Relative humidity(exclusive of media) | |
| operating : | 20% to 80% (non-condensing) |
| non-operating : | 5% to 95% (non-condensing) |
| DC Voltage Requirements | |
| +12VDC ± 5% | 0.6 Amp (typical) |
| + 5VDC ± 5% | 0.2 Amp (typical) |
| -12VDC ± 5% | 0.01AMP (typical) |
| Mechanical Dimensions | |
| width : | 148mm (5.83") |
| height : | 48mm (1.89") |
| depth : | 208mm (8.19") |
| weight : | 1.95Kg (4.3lb) |
| **RELIABILITY SPECFICATIONS :** | |
| MTBF(Mean Time Betmeen Failure): | 10,000power-on hrs (typical usage) |
| PM(Preventive Maintenance): | not required |
| Error Rates | |
| soft errors : | $1/10^9$ bits read |
| hard errors : | $1/10^{12}$ bits read |
| seek errors : | $1/10^6$ seeks |
| Component Design Life : | 5 years |
| Media Life : | $3.0 \times 10^4$ passes per track |

# 2. PRINCIPLE OF OPERATION

### *2.1.1 GENERAL OPERATIONS*

The FD-100 series disk drive consists of read/write and control electronics, drive mechanism, motor control, read/write head, track positioning mechanism, and the removable Diskette. These components perform the following functions :

* Interpret and generate control signals.
* Move read/write head to the desired track.
* Read and write data.
* Maintain correct diskette speed.

The relationship of interface signals for the internal functions of the FD-100 series are shown in Figure 2-1.

The Head Positioning Actuator positions the read/write head to the desired track on the Diskette. The Head Load Actuator loads the Diskette against the read/write head and data may then be recorded or read from the Diskette.

The drive has two (2) PCBs, one is for the drive motor control and the other is the drive PCB. The electronics packaged on the drive PCB contains :

1. Read/Write Amplifier and Transition Detector
2. Step Control Logic

3. Write Protect

4. Track Ø detector and auto stop.

The drive motor control PCB contains the following electronics :

1. Motor on & off circuitry

2. Motor speed control

### *2.1.2 HEAD POSITIONING*

An electrical stepping motor drives the Head Position Actuator Cam which positions the read/write head. The stepping motor rotates the motor pulley clockwise or counter-clockwise will cause the steel band and head to move forward and backward to the desired track.

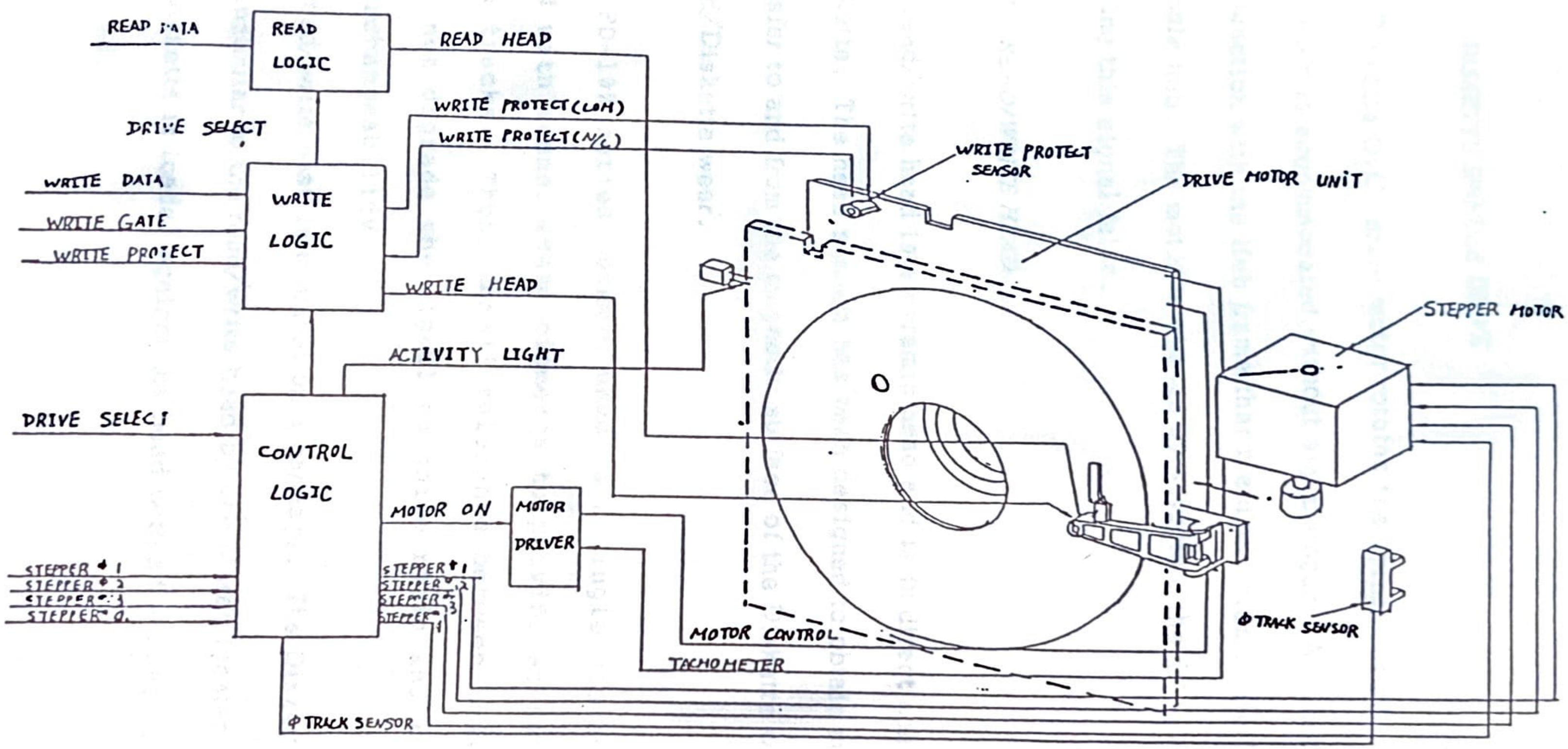



FIG. 2-1 FUNCTIONAL DIAGRAM.

### 2.1.3 DISKETTE SPINDLE DRIVE

The Diskette D.C. drive motor rotates the spindle at 300 rpm. 50 or 60 Hz operation is accommodated without any changes. A Clamping Hub moves in conjunction with the Hub frame that precisely clamps the Diskette to the spindle hub. The motor is started by selecting the drive and is stopped by making this signal false.

### 2.1.4 READ/WRITE HEAD

The read/write head is a ceramic head and is in direct contact with the Diskette. The head surface has been designed to obtain maximum signal transfer to and from the magnetic surface of the Diskette with minimum Head/Diskette wear.

The FD-100 series ceramic head is a single element read/write head with tunnel erase elements to provide erased areas between data tracks. Thus, normal tolerance between media and drives does not degrade the signal to noise ratio and insures disk interchangeability.

The read/write head is mounted on a carriage. The Diskette is held in a plane perpendicular to the read/write head by one platen located on the base casting. The Diskette is loaded against the head with a flat oad pad when the drive door is closed.

## 2.2 DRIVE MOTOR CONTROL

* Start/Stop
* Speed Control
* Over Current Protection
* Speed Adjust

The motor used in the FD-100 series is a DC drive motor and has a separate motor on and off interface line. After activating the motor on line, a 0.5 second delay must be introduced to allow proper motor speed before reading or writing.

When drive is selected, the motor will be started by causing current to flow thru the motor windings. Figure 2-2 shows the circuidt diagram of the motor speed control. The motor speed control utilizes an integral brushless tachometer. The output voltage signal from this tachometer is compared to a voltage/frequency reference level. The output from the voltage/frequency comparator will control the necessary current to maintain a constant motor speed of 300 rpm. Motor sped adjus tment changes the V ref thru a potentiometer.

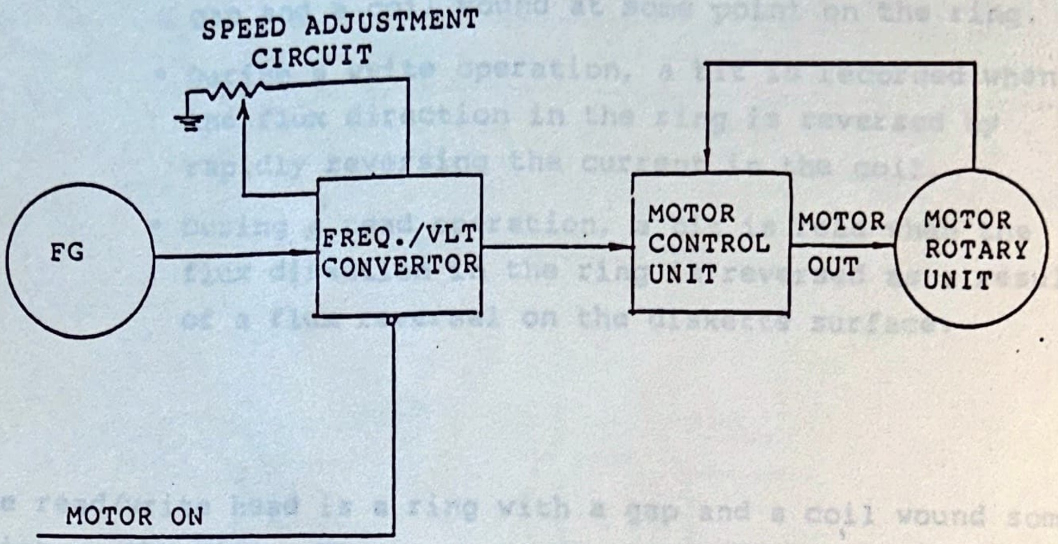



Fig. 2-2 Motor Control Functional Diagram

## 2.3 READ-WRITE OPERATIONS

* The read/write head, in general, is a ring with a gap and a coil wound at some point on the ring.
* During a write operation, a bit is recorded when the flux direction in the ring is reversed by rapidly reversing the current in the coil.
* During a read operation, a bit is read when the flux direction in the ring is reversed as a result of a flux reversal on the diskette surface.

The read/write head is a ring with a gap and a coil wound some point on the ring. When current flows through the coil, the flux induced in the ring fringes at the gap. As the diskette recording surface passes by the gap, the fringe flux magnetizes the surface in a longitudinal direction. See Figure 2-3.

During a write operation, a bit is recorded when the flux direction in the ring is reversed by rapidly reversing the current in the coil. The fringe flux is reversed in the gap and hence the portion of the flux flowing through the oxide recording surface is reversed. If the flux reversal is instantaneous in comparison to the motion of the diskette, it can be seen that the portion of the diskette surface that just passed under the gap is magnetized in one direction while the portion under the gap is magnetized in the opposite direction. This flux reversal represents a bit. See Figure 2-4.

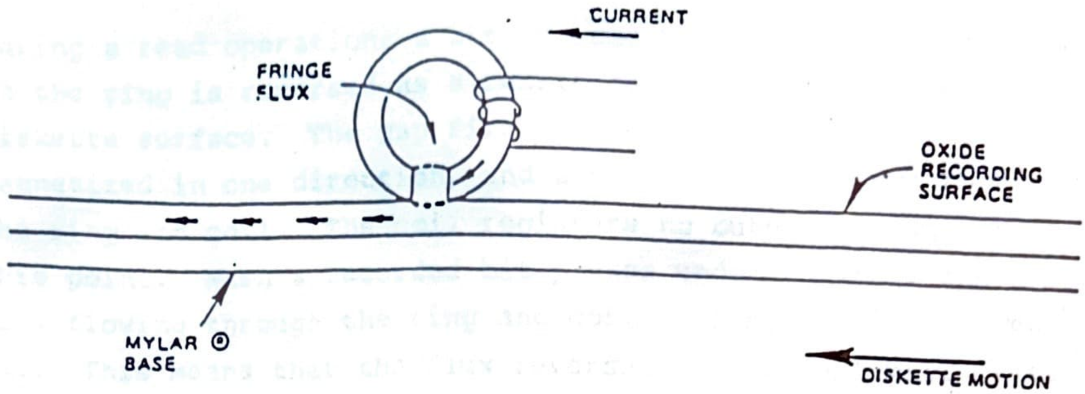



FIG. 2-3 BASIC READ/WRITE HEAD

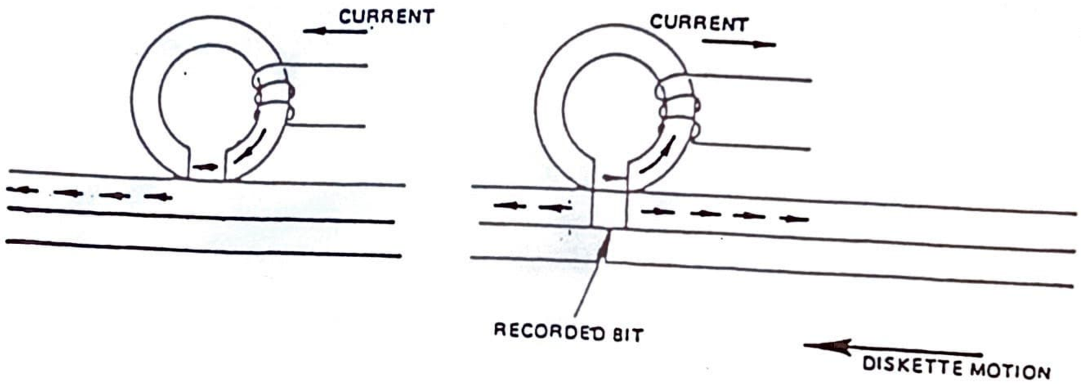



FIG. 2-4 RECORDED BIT

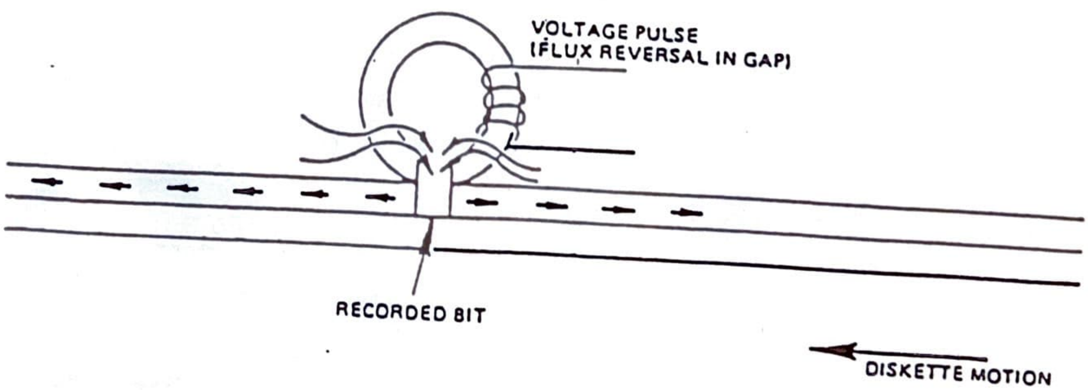



FIG. 2-5 READING A BIT

During a read operation, a bit is read when the flux direction in the ring is reversed as a result of a flux reversal on the diskette surface. The gap first passes over an area that is magnetized in one direction, and a constant flux flows thru the ring and coil. The coil registers no output voltage at this point. When a recorded bit passes under the gap, the flux flowing through the ring and coil will make a 180 reversal. This means that the flux reversal in the coil will cause a voltage output pulse. See Figure 2-5.

## 2.4 READ/WRITE HEAD DESIGN

* The read/write head contains three coils.
* When writing, the head erases the outer edges of the track to insure data recorded will not exceed the .012 track width.
* The head is ceramic.

The read/write head contains three coils. Two read-write coils are wound on a single core, center tapped and one erase coil is wound on a yoke that spans the track being written. The read-write and erase coils are connected as shown on Figure 2-6.

On a write operation, the erase coil is energized. This causes the outer edges of the track to be trim erased so as the track being recorded will not exceed the .012" track width. The straddle erasing allows for minor deviations in read/write head current so as one track is recorded, it will not "splash over" to adjacent tracks.

Each bit written will be directed to alternate read/write coils, thus causing a change in the direction of current flow through the read/write head. This will cause a change in the flux pattern for each bit. The current through either of the read/write coils will cause the old data to be erased as new data is recorded.

On a read operation, as the direction of flux changes on the diskette surface as it passes under the gap, current will be induced into one of the windings of the read/write head. This

Will result in a voltage output pulse. When the next data bit passes under the gap, another flux change in the recording surface takes place. This will cause current to be induced in the other coil causing another voltage output pulse of the opposite polarity.

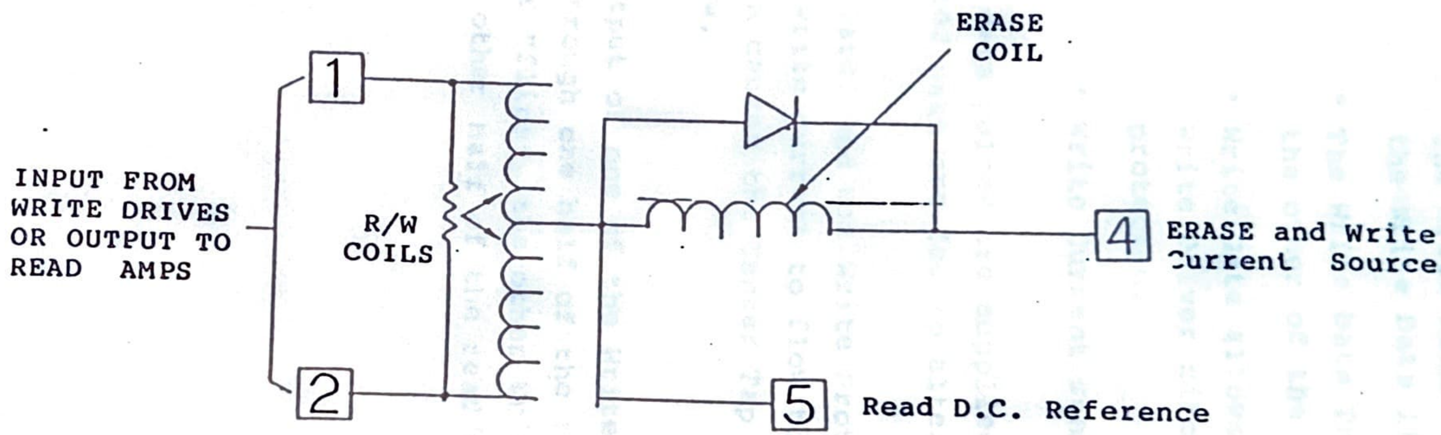



FIG. 2-6 READ/WRITE HEAD

## 2.5 *WRITE CIRCUIT OPERATION* (FIGURE 2-7)

* The Write Data Trigger flips with each pulse on the Write Data line.
* The Write Data Trigger alternately drives one or the other of the Write Drivers.
* Write Gate allows write current to flow to the Write Driver circuits if diskette is not write protected.
* Write Current sensed allows Erase Coil current.

Write data pulses are supplied by the using system. The data and inverted data are fed to alternate Write Drivers.

Write Gate, and Not Write Protect, are anded together and will cause write current to flow to the Write Driver Circuits, which in turn causes the Center Tap Switch to close and erase current to flow.

The output of one of the Write Drivers allows write current to flow through one half of the read/write coil. When the Write Trigger "flips", the other Write Driver provides write current to the other half of the read/write coil.

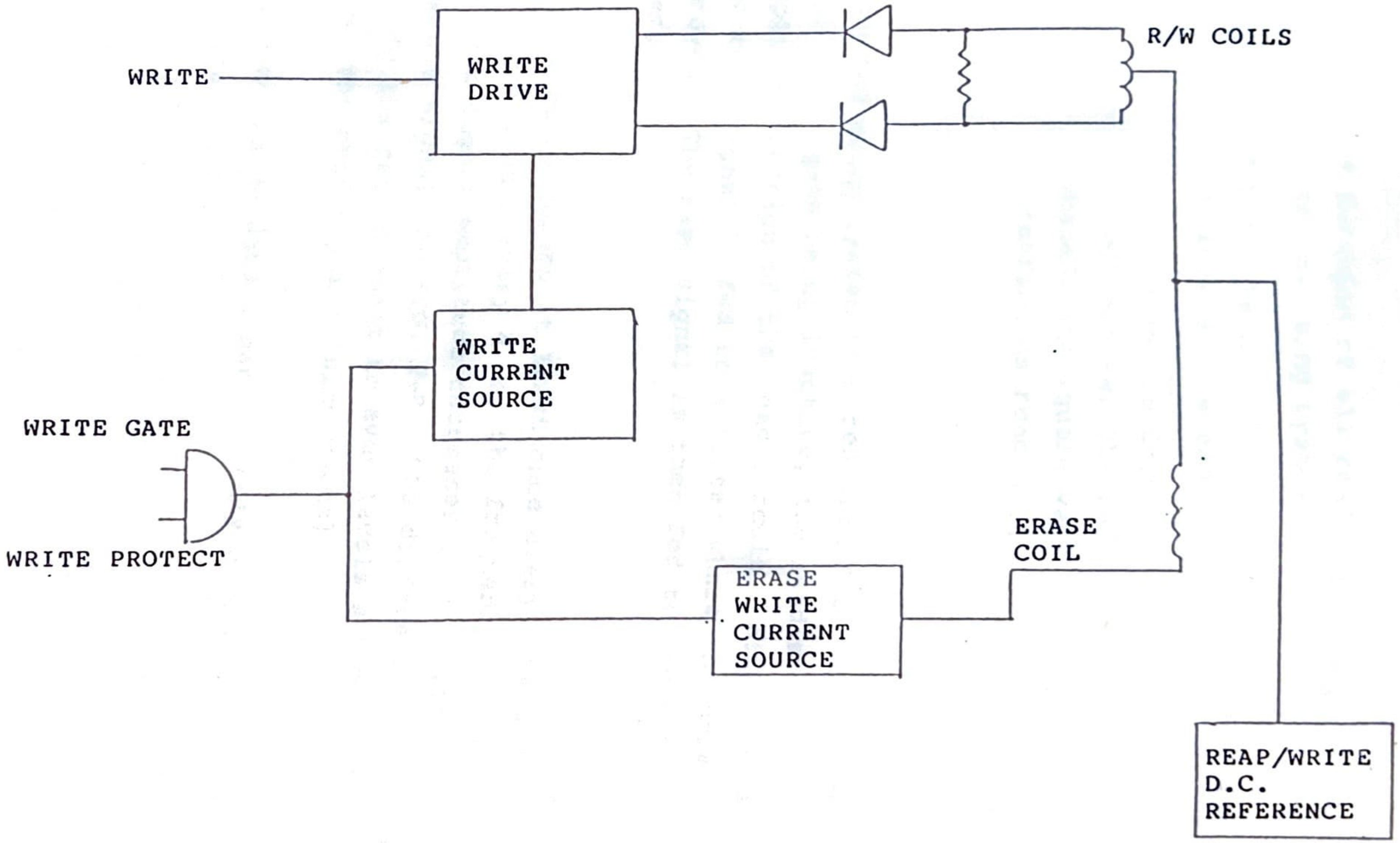



FIG.2-7 WRITE CIRCUIT FUNCTIONAL DIAGRAM.

## 2.6 READ CIRCUIT OPERATION (FIGURE 2-8)

* Duration of all read operations is under control of the using system.
* When the head is loaded, the read signal amplitude becomes active and is fed to the amplifier.
* As long as the drive is selected and write gate is not active, the read signal is amplified and shaped, the square wave signals are sent to the interface as read data.

When the using system requires data from the diskette drive, with write gate being inactive, the read signal is fed to the amplifier section of the read circuit. After the amplification, the read signal is fed to a filter where the noise spikes are removed. The read signal is then fed to the differential amplifier.

Since a pulse occurs at least once every 8 us and when data bits are present once every 4 us, the frequency of the read data varies. The read signal amplitude decreases as the frequency increases. Note the signals on FIG. 2-8. The differential amplifier will amplify the read signals to even levels and make square waves out of the read signals (sine waves).

The drive has no data separator only a pulse standardizer for the read data signal.

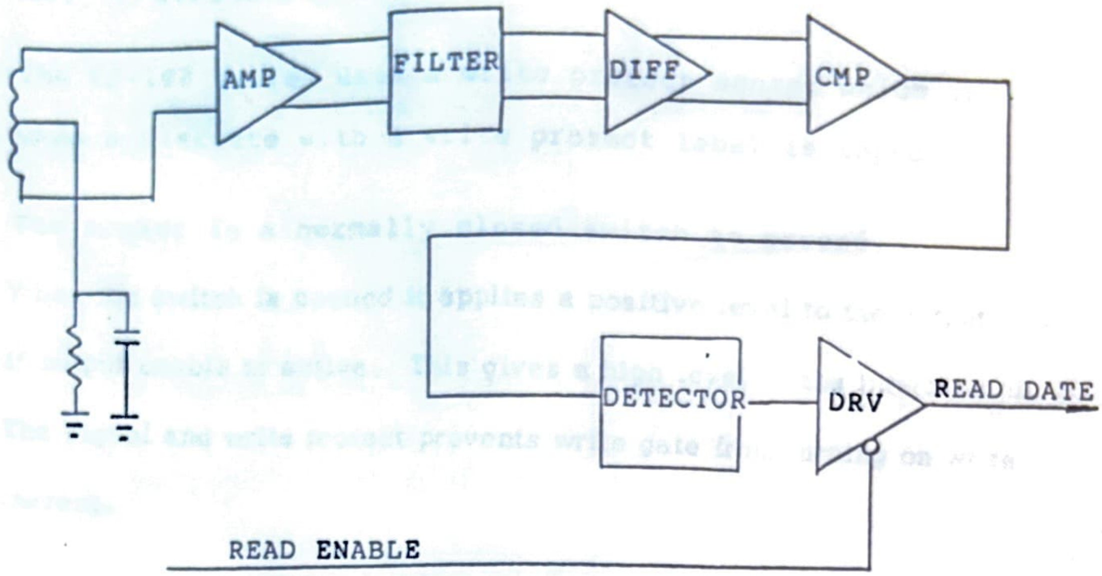



FIG.2-8 READ CIRCUIT FUNCTIONAL DIAGRAM

## 2.7 WRITE PROTECT

The FD-100 series uses a write protect sensor which is activated when a Diskette with a write protect label is taped.

The sensor is a normally closed switch to ground.

When the switch is opened it applies a positive level to the output driver if output enable is active. This gives a high level to the interface pin 20. The signal and write protect prevents write gate from turning on write current.

## 2.8 INTERFACE

The electrical interfa ce between the FD-100 series drive and the Computer is via connector J3. J3 provides the signal interface and DC power.

### 2.8.1 J1 CONNECTOR

Connection to J3 is a 20 pin right angle, double row, PC mounting header connector. Detail pin assignment refer to FIG. 2-9. DC power to the drive is via connector J3 Figure 2-10 outlines the voltage and current requirements.

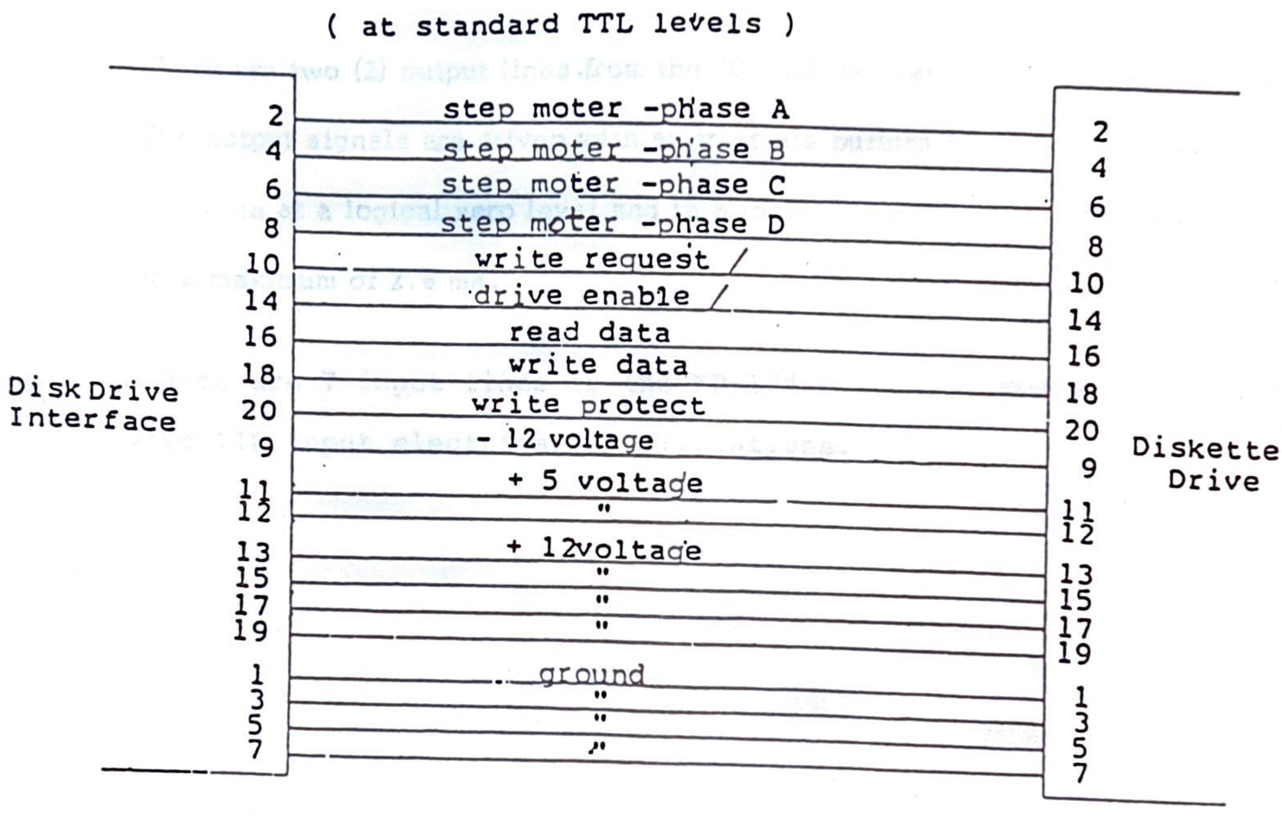



FIG. 2-9

| DC VOLTAGE | TOLERANCE | CURRENT | MAX RIPPLE (p-p) |
|---|---|---|---|
| +12VDC | ± 0.6VDC | 0.6AMP | 100mV |
| +5VDC | ± 0.25VDC | 0.2AMP | 50mV |
| -12VDC | ± 0.6VDC | 10MA | 100mV |

FIG. 2-10 POWER REQUIREMENT

### 2.8.2 INPUT/OUTPUT LINES

There are two (2) output lines from the FD-100 series to Computer. The output signals are driven with an tri-state buffers of sinking a maximum of 24 ma at a logical zero level and in a logical one, the collector current is a maximum of 2.6 ma.

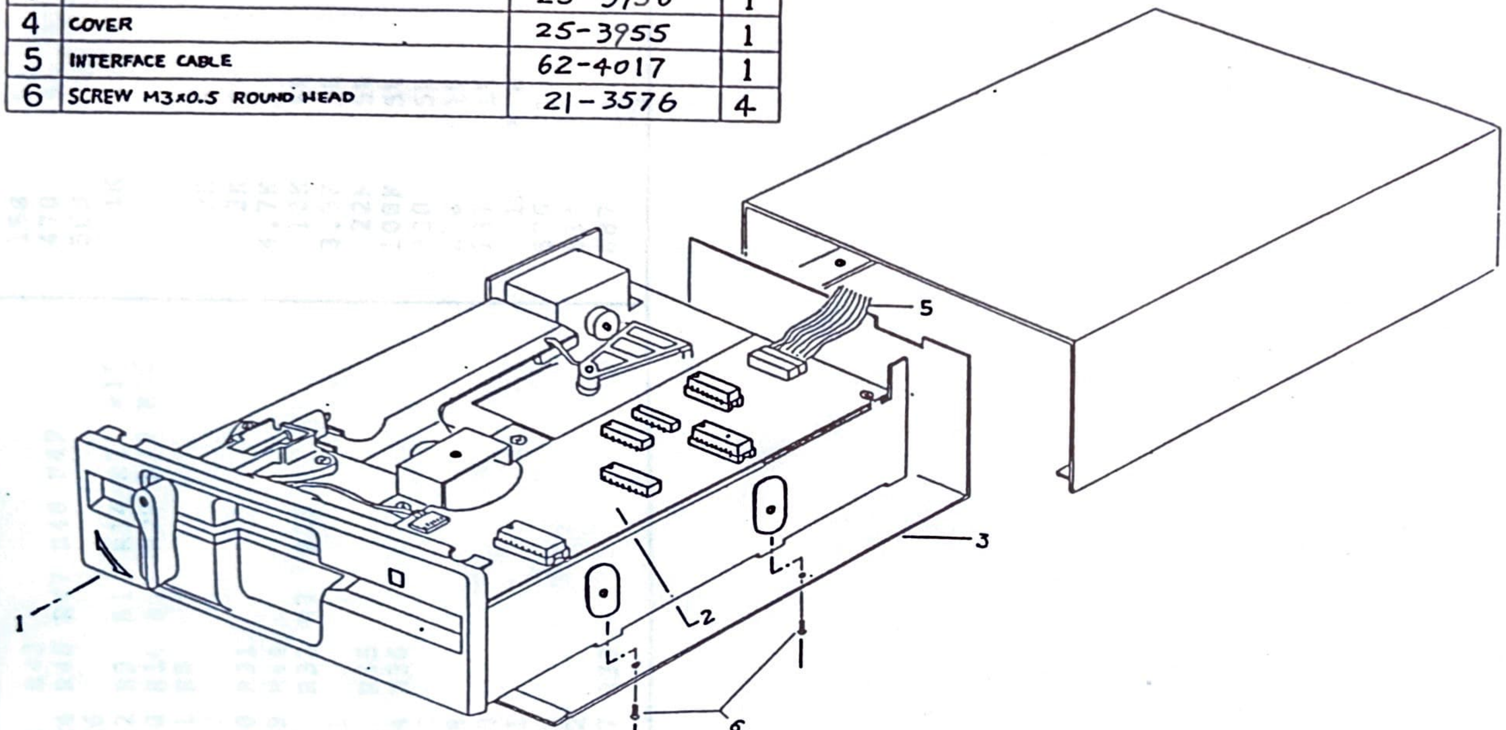
There are 7 input lines to the FD-100 series. These input lines are TTL input electrical specifications.

THIS PAGE INTENTIONALLY LEFT BLANK

## 3.3 USER-REPLACEABLE PARTS

Table 3-1 System FD-100 User-Replaceable Parts

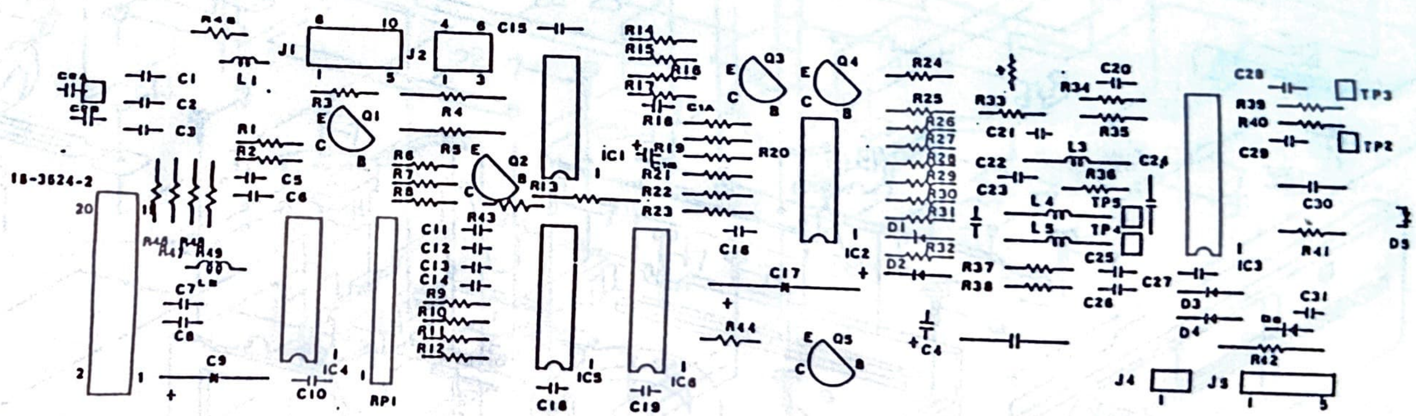
| Find No. | DESCRIPTION | Part number | QTY |
|---|---|---|---|
| 1 | DISK DRIVE LDD-103SSA (WITH SPEED CONTROL PCB) | 62-3879 | 1 |
| 2 | DRIVE PCB | 62-3677 | 1 |
| 3 | CHASSIS | 25-3956 | 1 |
| 4 | COVER | 25-3955 | 1 |
| 5 | INTERFACE CABLE | 62-4017 | 1 |
| 6 | SCREW M3x0.5 ROUND HEAD | 21-3576 | 4 |



## TABLE 3-2 DRIVE PCB USER-REPLACEABLE PARTS

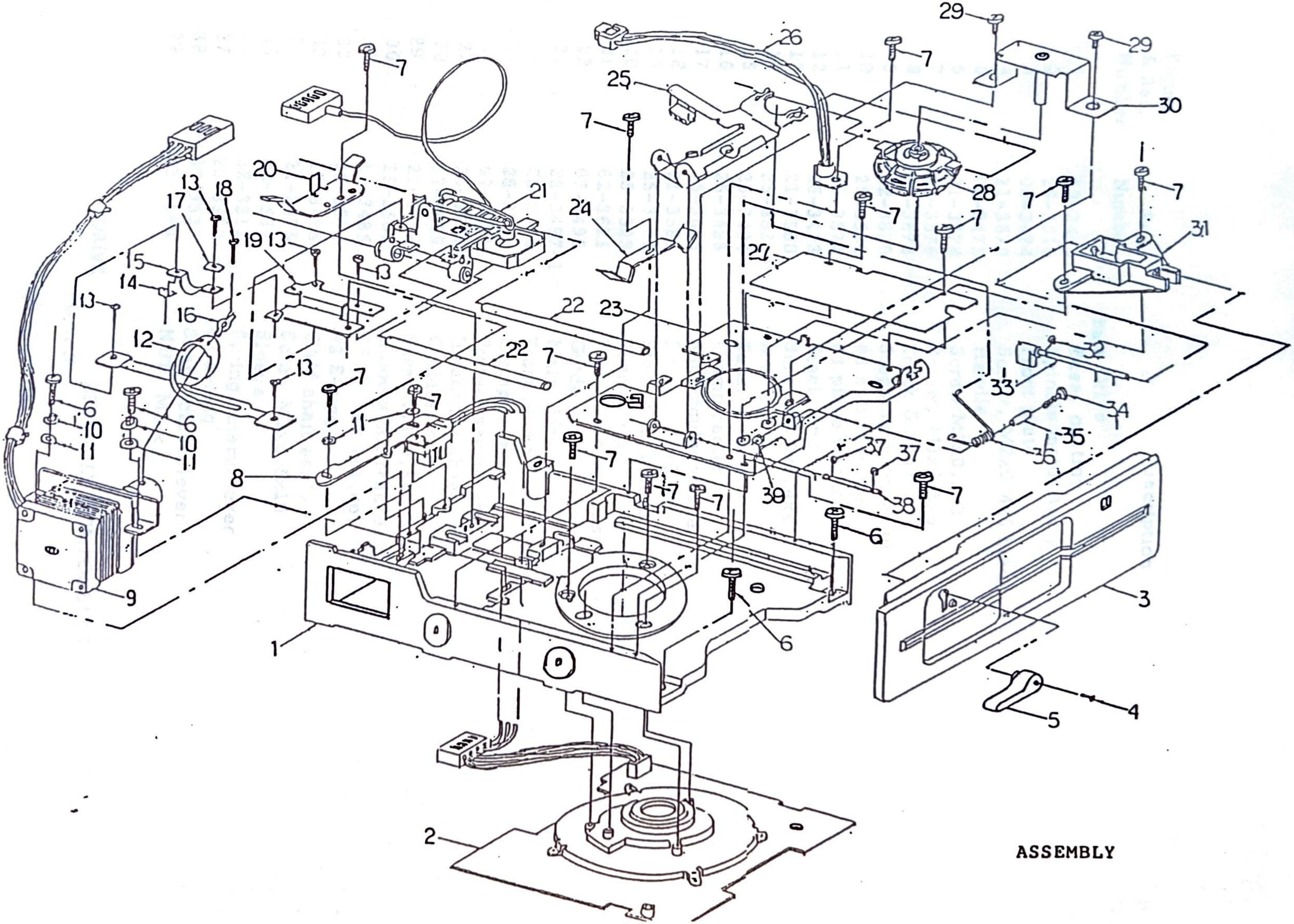
| INDEX NO. | DESCRIPTION | | | QTY |
|---|---|---|---|---|
| R3 R43 | 150 | 5% | 1/8 W | 2 |
| R24 R46 R47 R48 R49 | 470 | 5% | 1/8 W | 5 |
| R26 | 560 | 5% | 1/8 W | 1 |
| R12 R9 R1 R14 R11 R15<br>R10 R16 R17 R28 R19 R25<br>R21 R8 | 1K | 5% | 1/8 W | 14 |
| R27 | 2K | 5% | 1/8 W | 1 |
| R30 R31 | 3K | 5% | 1/8 W | 2 |
| R39 R44 | 4.7K | 5% | 1/8 W | 2 |
| R6 R33 R7 R29 | 10K | 5% | 1/8 W | 4 |
| R41 | 3.9K | 5% | 1/8 W | 1 |
| R2 R45 | 22K | 5% | 1/8 W | 2 |
| R34 R36 | 100K | 5% | 1/8 W | 2 |
| R13 | 330 | 5% | 1/2 W | 1 |
| R18 | 270 | 1% | 1/8 W | 1 |
| R23 | 137 | 1% | 1/8 W | 1 |
| Rp1 | 1K | * 8 | | 1 |
| R22 | 576 | 1% | 1/8 W | 1 |
| R32 | 634 | 1% | 1/8 W | 1 |
| R37 R38 | 887 | 1% | 1/8 W | 2 |

| | | |
|---|---|---|
| R20 | 104 1% 1/8 W | 1 |
| R40 | 9.1K 5% 1/8 W | 1 |
| R42 | 168 5% 1/2 W | 1 |
| VR | 10KB 10% 1/4 W | 1 |
| R35 | 91 5% 1/8 W | 1 |
| R4 R5 | 100 5% 1/2 W | 2 |
| L4 L5 | 470UH 10% | 2 |
| L3 | 68UH 10% | 1 |
| L1 L2 | 0.34UH | 2 |
| IC5 | 74LS75 | 1 |
| IC4 | 18SAO30 | 1 |
| IC6 | 2003AN | 1 |
| IC2 | LN3146N | 1 |
| IC3 | IC3470P | 1 |
| IC1 | IC74LS125 | 1 |
| Q1 Q2 Q5 | 2SC711E | 3 |
| Q3 Q4 | 2SA6950 | 2 |
| C11 C13 C12 C14 C15 | 220P 5% SL | 5 |
| C23 | 110PF/50V 5% M/C | 1 |
| C24 | 550PF/50V 5% M/C | 1 |
| C31 | 0.01UF/50V F.Z. | 1 |
| C1 C6 C10 C2 C7 C18<br>C15 C16 C3 C8 C19 C20<br>C22 C25 C26 C27 C1A | 0.1 UF/50V +80% -20% B/C | 17 |
| C4 C17 | 220UF/16V +80% -20% E/C | 2 |
| C30 | 0.22UF/100V 5% M/C | 1 |
| C9 | 10UF/16V +80% -20% E/C | 1 |
| C21 | 0.003UF/50V 5% M/C | 1 |
| C28 C29 | 330PF SL | 2 |
| C1B | 1UF E/C | 1 |
| C2A C2B | 100PF SL 5% | 2 |
| D1 D2 D3 D4 D5 D5B | IN4108 (LAY) | 6 |
| D5 | LED BL-R1130 5*5m/m | 1 |
| d6 | IN 4001 | 1 |




DRIVE PCB COMPONENT LOCATIONS

P/N 62-3677



1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39

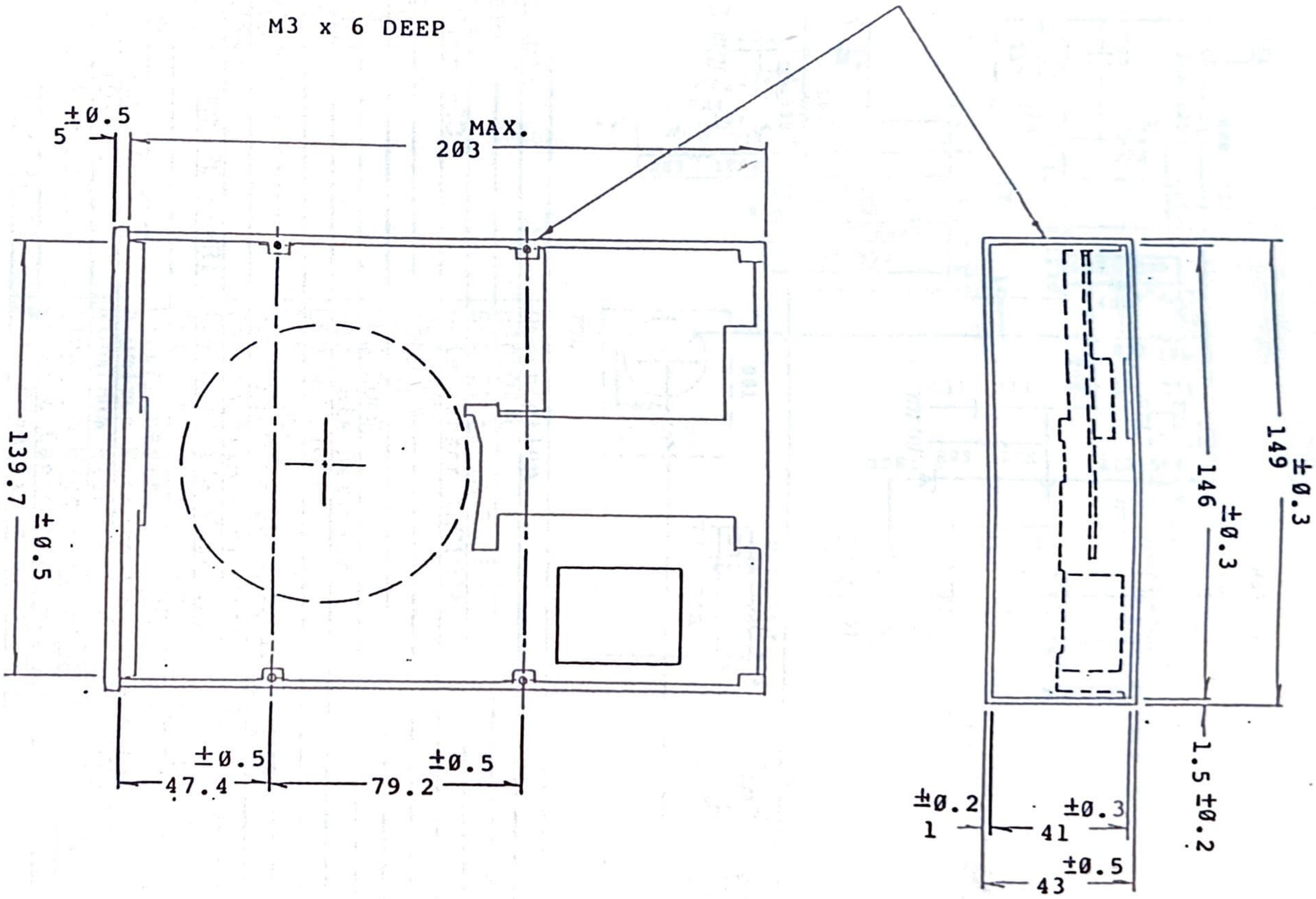

ASSEMBLY

| Figure & Ref. Number | Part Number | Description | Qty Per Un. |
|---|---|---|---|
| | | Bare Drive | |
| 1 | 59-3455 | Base, casting | 1 |
| 2 | 29-3515 | Spindle motor Ass'y | 1 |
| 3 | 06-3474 | Front plate | 1 |
| 4 | 21-3579 | Screw, M2x0.4x5 black | 1 |
| 5 | 06-3475 | Handle | 1 |
| 6 | 21-3578 | Screw M3x0.5x8 | 4 |
| 7 | 21-3576 | Screw, M3x0.5x6 | 14 |
| 8 | 62-3673 | Track & Assy | 1 |
| 9 | 62-3670 | Stepping motor assy | 1 |
| 10 | 25-3582 | #3 spring washer | 2 |
| 11 | 25-3571 | #3 plane washer | 4 |
| 12 | 25-3493 | Steel band | 1 |
| 13 | 21-3580 | Screw, M2x0.4x4 | 5 |
| 14 | 25-3533 | L type nut | 1 |
| 15 | 25-3494 | Tension spring | 1 |
| 16 | 25-3495 | Clamp center | 2 |
| 17 | 25-3623 | Rectangular washer | 1 |
| 18 | 21-3581 | Screw, M2x0.4x6 | 1 |
| 19 | 25-3501 | Steel band bracket | 1 |
| 20 | 25-3503 | Rear clamp | 1 |
| 21 | 62-3625 | Head, carriage assy | 1 |
| 22 | 07-3496 | Guide rod | 2 |
| 23 | 25-3497-1 | Main plate | 1 |
| 24 | 25-3502 | Front clamp | 1 |
| 25 | 25-3456 | Clamp lever | 1 |
| 26 | 62-3671 | Protect sensor ass'y | 1 |
| 27 | 25-3531 | Insulating sheet | 1 |
| 28 | 62-3669 | Collete ass'y | 1 |
| 29 | 21-3568 | Screw M3x0.5x4 | 1 |
| 30 | 25-3505 | Guide-shaft bracket | 2 |
| 31 | 17-3485 | CAM shaft bracket | 1 |
| 32 | 25-3574 | #3 E-ring | 1 |
| 33 | 17-3486 | CAM shaft ass'y | 1 |
| 34 | 21-3575 | Screw M3x0.5x10 | 1 |
| 35 | 02-3532 | Sleeve | 1 |
| 36 | 22-3511 | Spring, clamp lever | 1 |
| 37 | 25-3573 | #2 E-ring | 1 |
| 38 | 07-3530 | Shaft, clamp lever | 2 |
| 39 | 23-3569 | NUT, M3 x 0.5 | 1 |

DRIVE FD-100 ASSEMBLY COMPONENTS

MOUNTING HOLES:4 ON BOTTOM 2ØN EACH $IDE

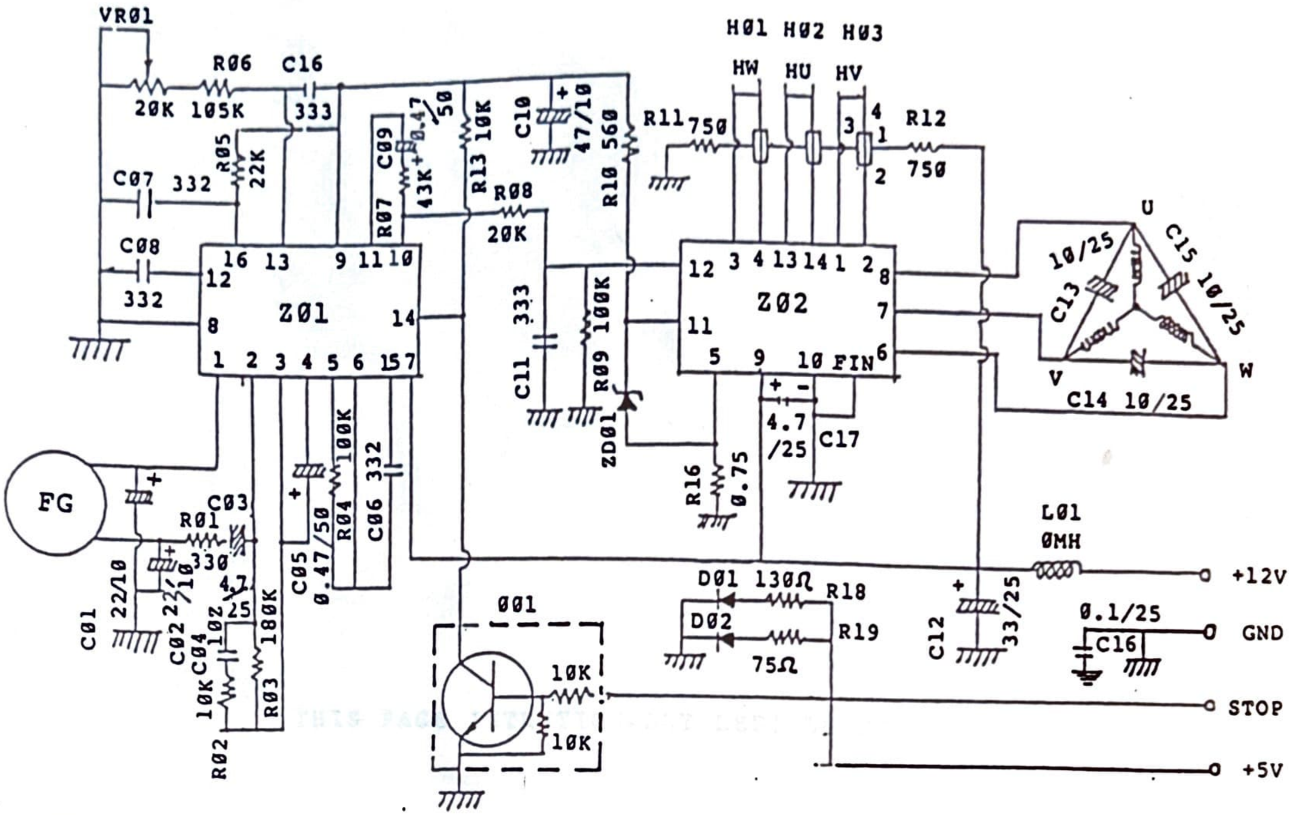
M3 x 6 DEEP




DIMENSION

# 3.4 SCHEMATIC DIAGRAMS

FD-100 DD MOTOR CIRCUIT




| Symbol | Description |
|---|---|
| Z01 | µPC1043C |
| Z02 | SA1001 or TA7245BP |
| H01---H03 | Hole sensor THS103A |
| Q01 | 2SC3402 |
| ZD01 | HZ 2-B |
| L01 | FL5R200B |
| VR01 | Variable Resistor |
| D01 | LED NJL1103E |
| D02 | LED LN66R |
| C16 | P.U. FILM Capacitors +-5% 100V |
| R18,19 | Carbon Film Resistors +-5% |
| R01---R05,R06---R16 | 1/4 W or 1/6 W or 1/2 W |
| R16 | Metal Film Resistor +-5% 1/2 W |
| R06 | Linear Resistor (+250 ppm) |
| C01---C03,C05 | Aluminum Electrolytic Capacitors +-20% |
| C09,C10,C12 | Aluminum Electrolytic Capacitors +-20% |
| C13---C15,C17 | Aluminum Electrolytic Capacitors +-20% |
| C18 | Ceramic Capacitors +80-20% |
| C04,c06--c08,c11 | Film Capacitors +-10% 50V |

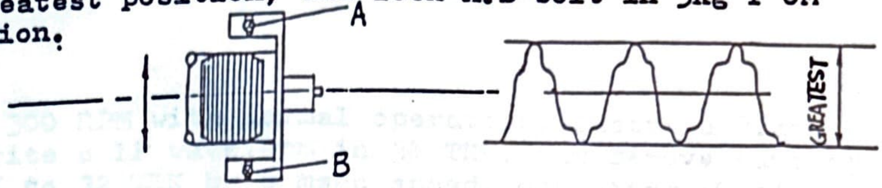
# 3.5 DISK ALIGNMENT

3.5.1 Alighment Diskette: use DYMEK 501-05

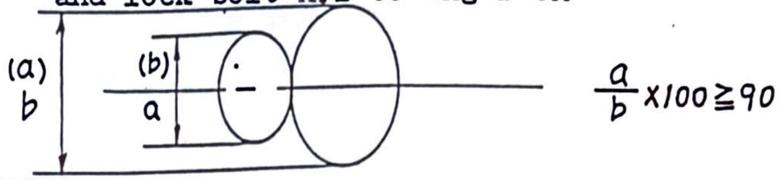
3.5.2 Method

a. Insert diskette 501-05 into drive

b. Set R/W head in track Ø, read IF (62.5KH) waveform on scope, moving step motor slowly check if wavefom at greatest position, and lock A.B bolt in 3Kg f CM position.




c. Remove R/W carriage head to 16 track, check and adjust ratio of cats eye to a:b x 100 $\geqq$ 90 and lock bolt A.B to 6Kg f CM



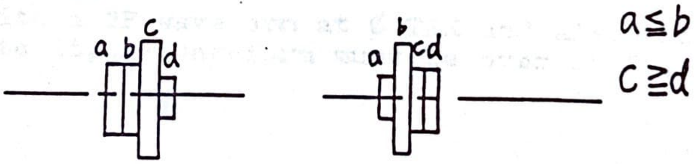

d. Hysterisis: Remove carriage head from Ø track to 16 track, and from 34 track to 16 track, the ratio of cats eye must be obtained $\frac{a}{b}$ x 100 $\geqq$ 80%

# 3.6 CHECK OF OTHER FUNCTION

Use DYMEK 501-02 Diskette

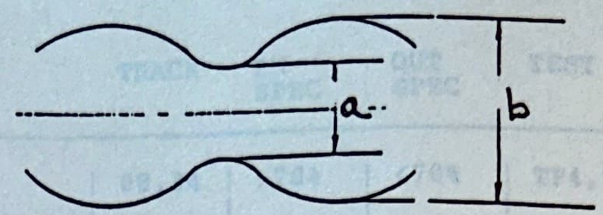
3.6.1 Guide ROD

Remove carriage head to 01 track and 34 track read the azimuth both of the two positions must be in the condition of a $\leqq$ b, c $\geqq$ d




3.6.2 Modulation

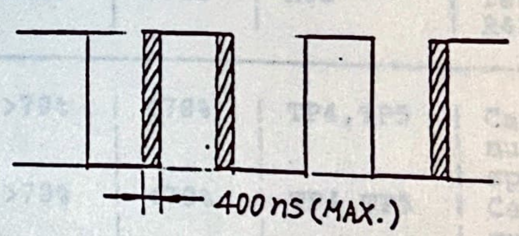
SPINDLE at 300 RPM with normal running, insert a blank diskette to 34 track then write a IF waveform into IT, operating in/out 5 times continually, the ration of waveform between maximum, and minimum must be over 75%




$\frac{a}{b} \times 100 \geqq 75$

3.6.3 Self Erase
SPINDLE at 300 RPM with normal operation, insert a blank diskette write a 1F waveform in 31 TRK, and remove carriage from 30 TRK to 32 TRK by 6 msec speed, operating 10 sec. continually, the wavefrom lossen must be not over 5%

3.6.4 Asymmetry
Set a standard PCB on a bare drive, and insert a blank diskette, push down the clamp, write a 2F waveform in 34 TRK, the pulse of timing shown on the scope must be under 400 ns. Then write a 1F waveform in 33 TRK and make same checking procedure.
Must get same result.




3.6.5 Overwrite
Write a 2F waveform at Ø TRK and also write a STD 1F waveform into it, 1F waveform must be over 26 DB.

3.6.6 Resulution

$$\frac{E\ out}{E\ out} \quad \frac{TRK.\ 392F}{TRK.\ 321F} \geqq 0.55$$

$$\frac{E\ out}{E\ out} \quad \frac{TRK.00,2F}{TRK.00,1F} \leqq 0.95$$

| TEST | DISK | TRACK | IN SPEC | OUT SPEC | TEST PTS. | SPECS. |
|---|---|---|---|---|---|---|
| Track 00-34 | Dymek Align. DK501 3E | 00,34 | >70% | <70% | TP4,TP5 | See Cats Eyes at tracks 00 and 34 Must be within specs. |
| Speed | Blank Disk | None | 198us to 200us | <198us | None | Drive speed must conform to specs. (300 RPM) |
| Run Out | Blank Disk | 00<br>34 | <10mV<br>>2.5mV | >10mV<br><2.5mV | TP4,TP5<br>TP4,TP5 | At track 00 write 1F See Amplifier level<br>At track 34 write 2F See Amplifier level |
| Resolu-tion | Blank Disk | 00<br>34 | >90%<br>>70% | <90%<br><70% | TP4,TP5<br>TP4,TP5 | At track 00 write 2F/1F x 100%<br>At track 34 write 2/1 x 100% |
| Erase | None | None | None | None | R42 | Test both sides of R42. Verify voltage |
| Track Alignment<br>Hyst-ersis | Dymek Align. Disk DK501-2 | 0->16<br>34->16 | >70%<br>>70% | <70%<br><70% | TP4,TP5<br>TP4,TP5 | Cats Eye pattern must be within specs.<br>Cats Eye pattern must be within specs. |
| Azimuth | Dymek Align. Disk DK501-2 | 34 | <0'+18 | >0'+18 | TP4,TP5 | Observe signal at track 34 must be within specs. |
| Asymme-try | Blank Disk | 34 | <400ns | >400ns | Pin 10 of 3470 | At track 34 write 2F. |

# SECTION 3

## Diagrams

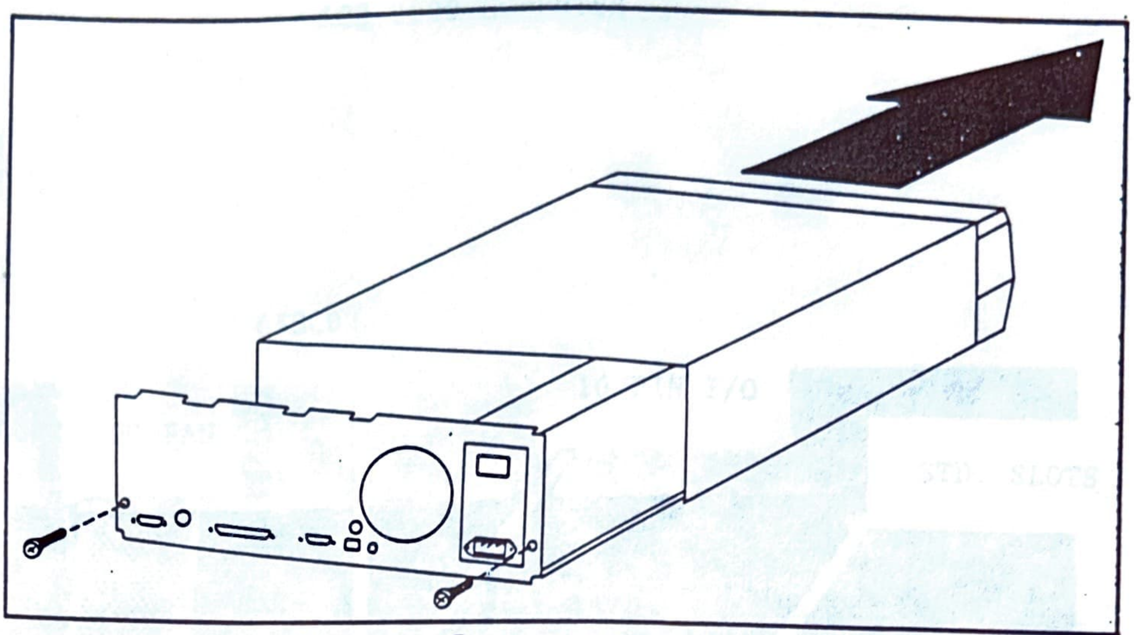
1. System Block Diagram
2. Case Removal Pictorial
3. Motherboard Pictorial
4. Power Supply Pictorial

## Schematics

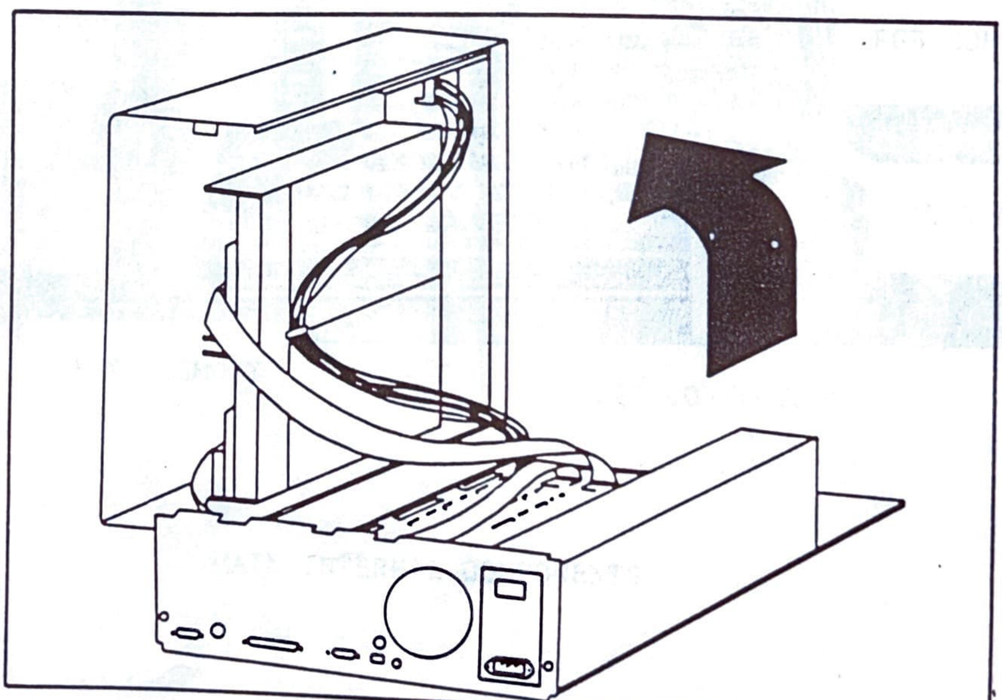
1. CPU
2. Memory Decode
3. Main Memory
4. AUX Memory
5. Main Timing
6. Timing
7. Video and RF
8. I/O Decode
9. Game I/O and Softswitches
10. Keyboard Interface
11. Sound and Printer I/O
12. FDD Controller
13. Connector Pinouts and Power
14. Power Supply
15. Keyboard Version One
16. Keyboard Version Two

Note: Disk drive schematics are in Section 2.

# COMPUTER CASE REMOVAL

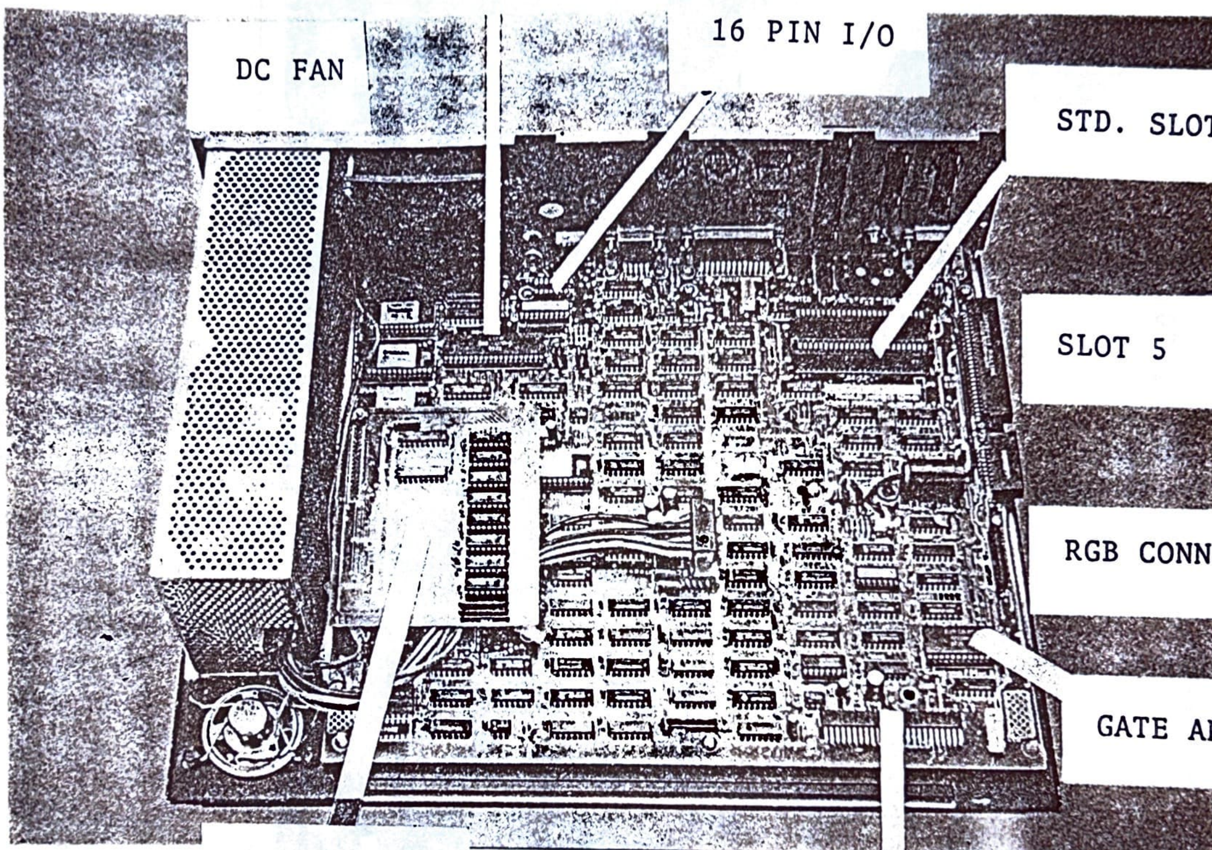


Removing Computer Case—Step 1



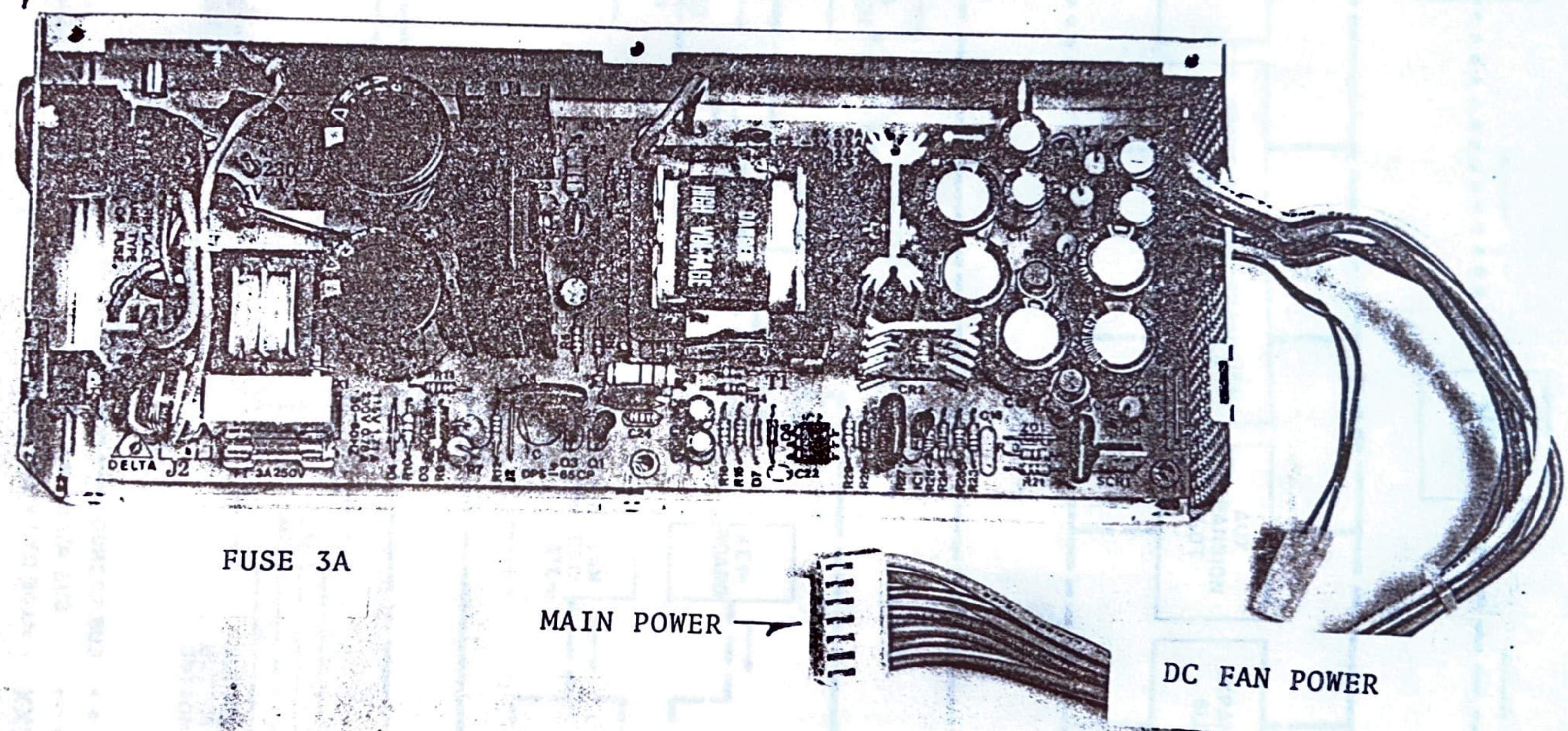
Removing Computer Case—Step 2

ACE 2000 COMPUTER




MAIN INTERNAL COMPONENTS

AC INPUT



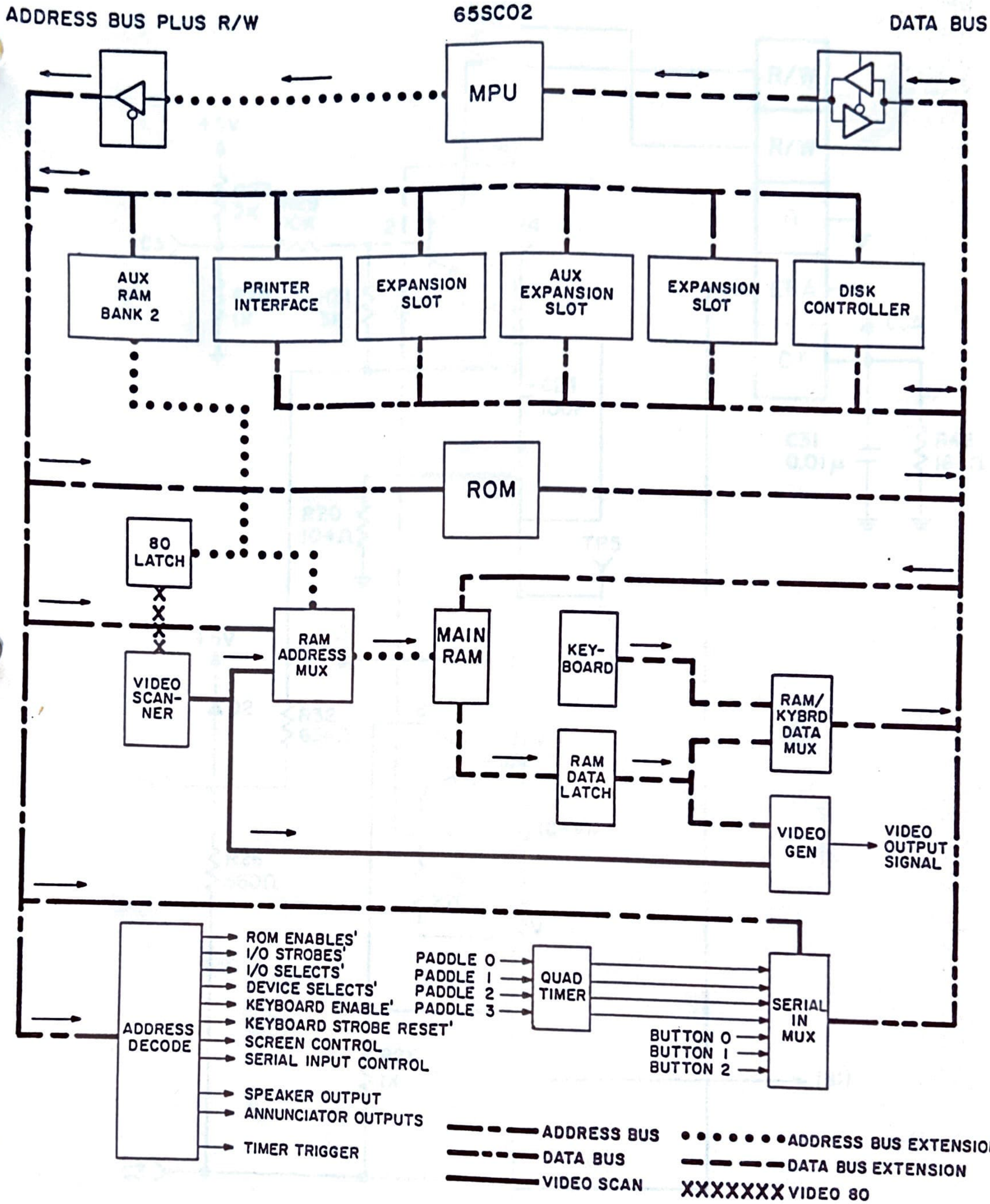
FUSE 3A

MAIN POWER

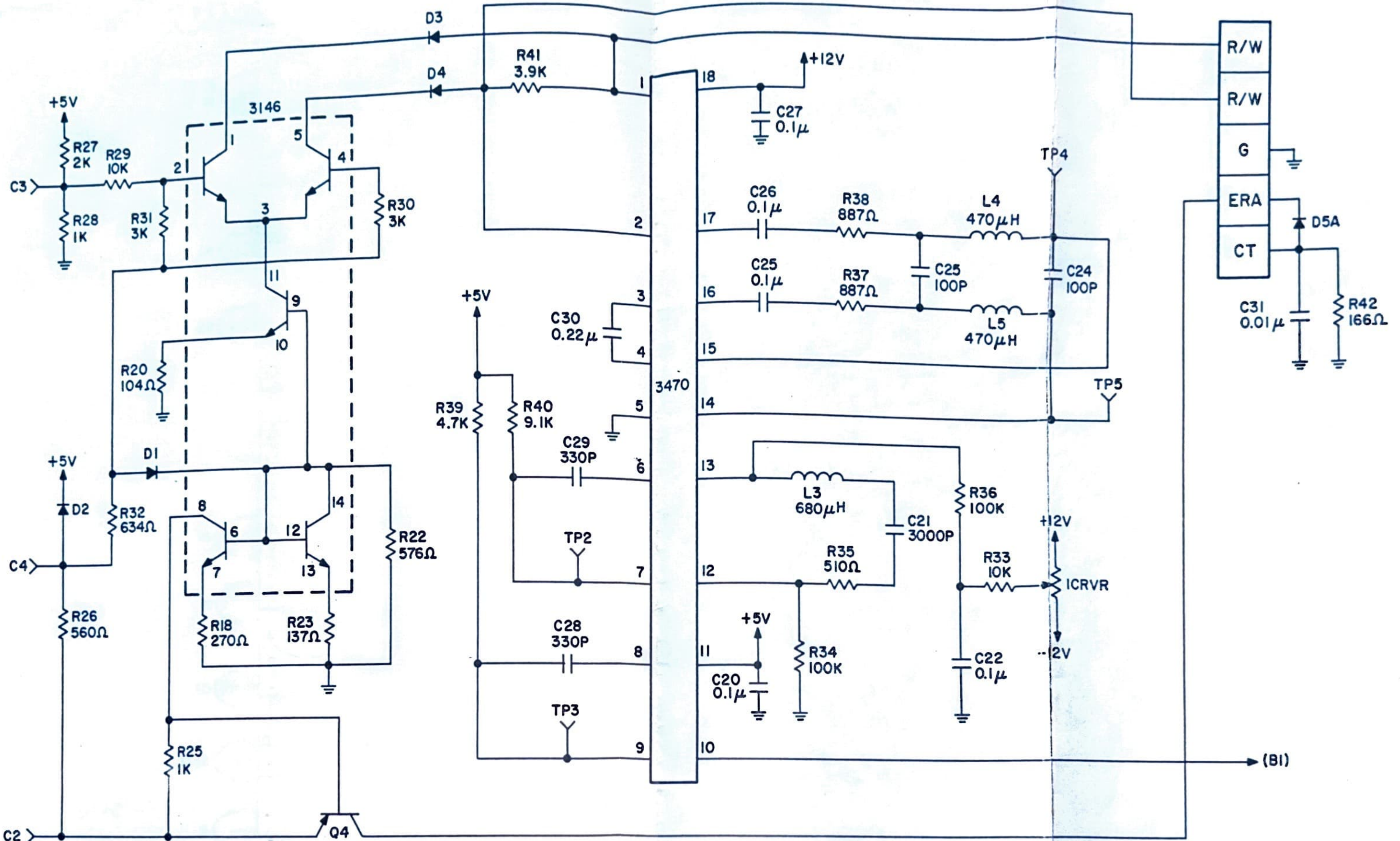
DC FAN POWER

ACE 2000

POWER SUPPLY

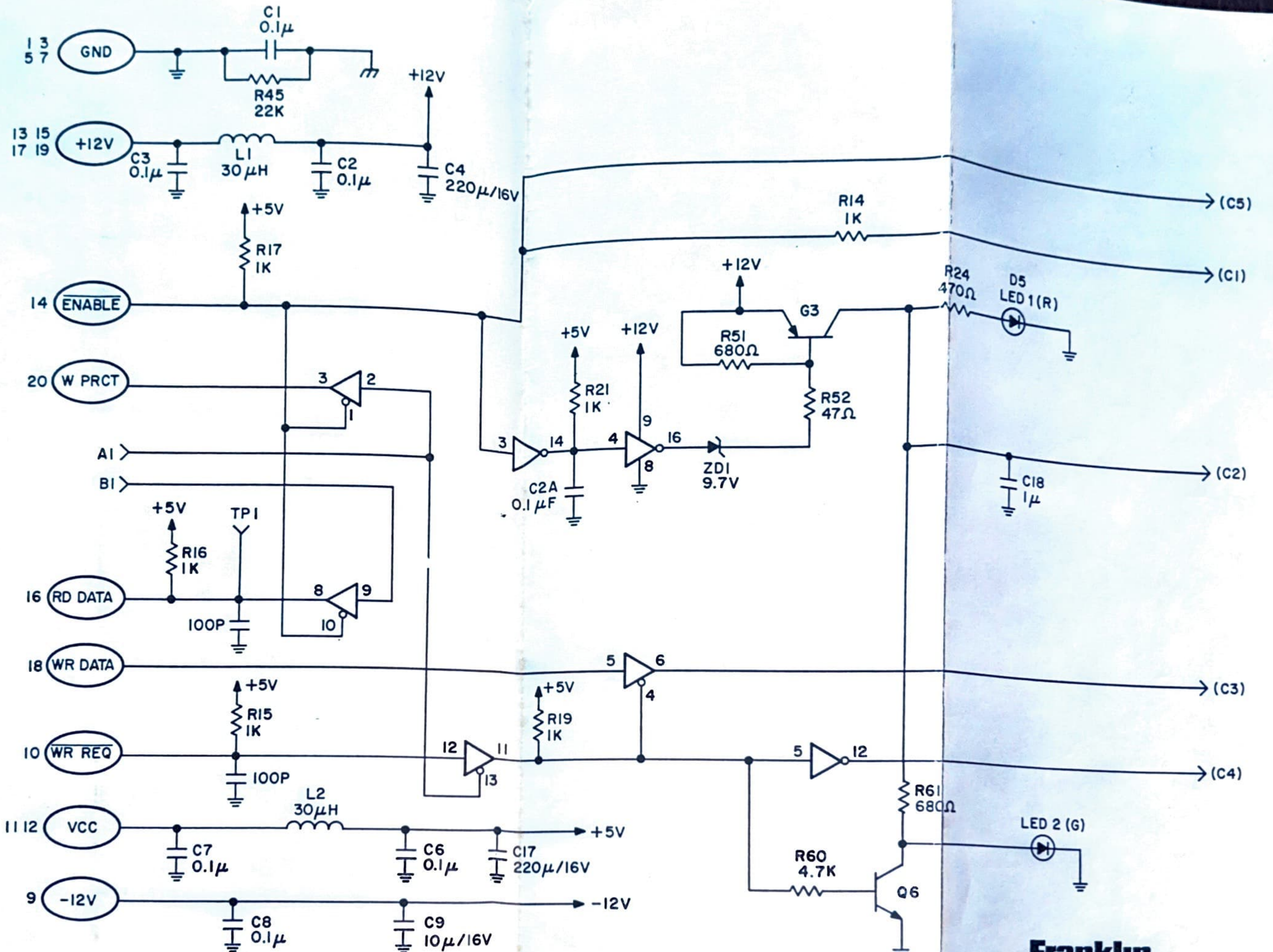



ACE 2000 BLOCK DIAGRAM




**Franklin** Computer
RT. 73 & HADDONFIELD RD.
PENNSAUKEN, N.J. 08110

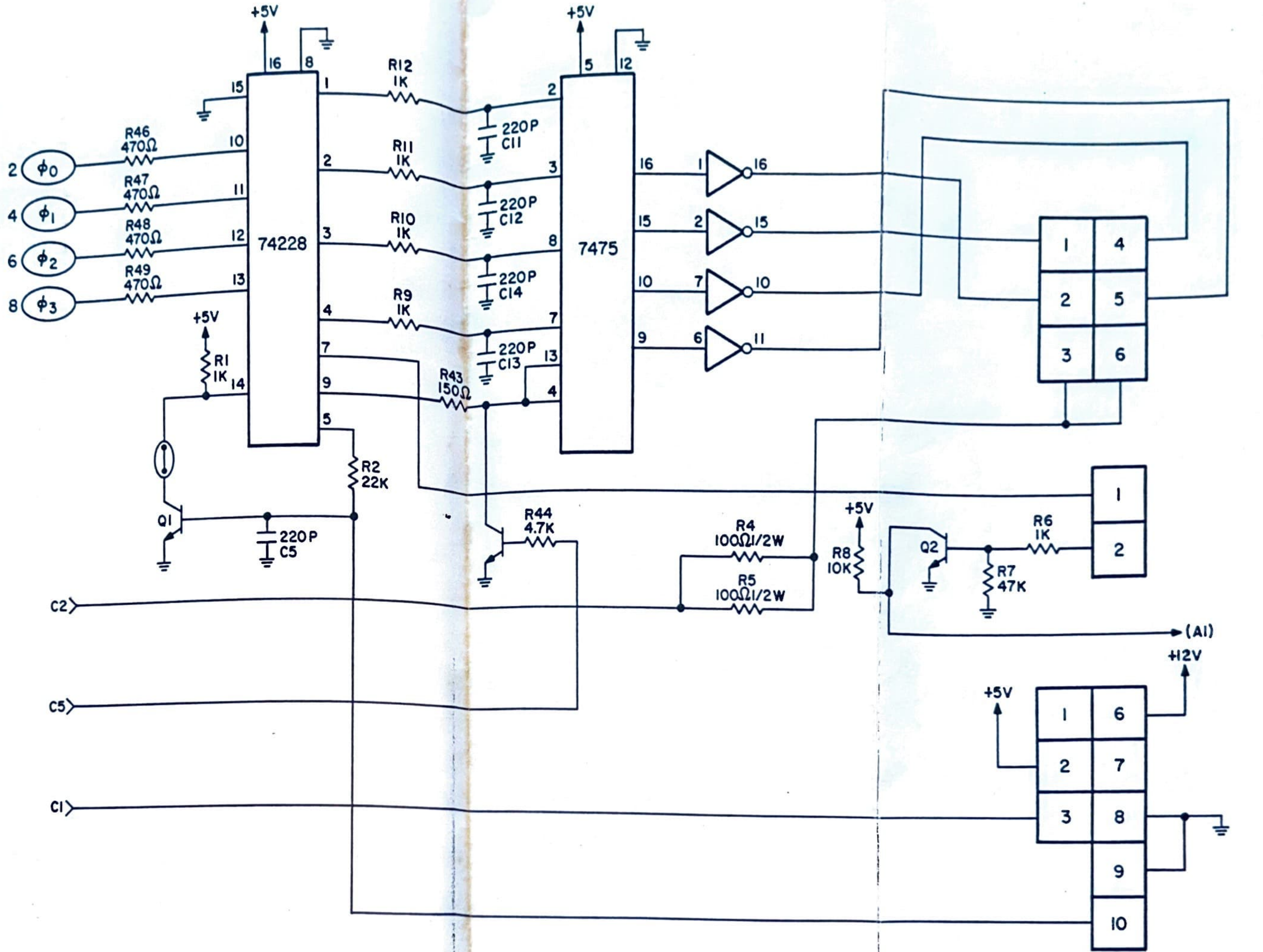
DISK DRIVE




Franklin Computer
RT. 73 & HADDONFIELD RD.
PENNSAUKEN, N.J. 08110

DISK DRIVE

2 of 3




Franklin Computer
RT. 73 & HADDONFIELD RD.
PENNSAUKEN, N.J. 08110

DISK DRIVE

## RGB JUMPER SELECTS

The Franklin RGB Manual refers to various positions for the selection of the jumpers on the 8 position switch on the printed circuit board. The placement of the jumpers vary widely based upon the manufacturer of the RGB monitor in use. In addition to the directions in the manual, the following selection of jumpers have been shown to work on some models of monitors:

1. TATUNG: select (jumper block ON) 2,3,4,6,7,8

2. SEARS combination RGB/TV/COLOR types: 2,3,5,6,7,8

3. IBM type RGB: 2,3,5,6,7,8

Please note that positions 6 and 7 can be any selection since they are only text color selection jumpers (ie: blue, green, amber, white) when in full text mode.

Selection of jumpers for proper RGB function will require several attempts to find the proper combination. There are no true standards which manufacturers follow and, therefore, this is the reason for the 8 position options.

pz
2/26/86

Franklin Computer

# ACE 2000 Series Computer
# RGB Interface Manual Supplement

Franklin Computer Corporation
Route 73 & Haddonfield Road
Pennsauken, NJ 08110
(609) 488-0666

## RGB INSTALLATION BOOKLET WITH PICTORIAL

STEP #1: Remove Cover (Refer to Fig. 1)

1) [ ] Power down unit, remove power cord, remove video cable, remove any other cables that are attached to the back of the unit.

2) [ ] From the back of the unit remove the screws from the lower right and left corners (See Fig. 1 (AA) for location).

3) [ ] Slide the cover forward off of the base plate to expose the motherboard.

*NOTE: It is NOT necessary to remove any of the cables connecting the cover to the motherboard, however, if any were removed, be sure to re-install them exactly as they connected originally.

4) [ ] Remove slot cover #6 from the back of the unit (See Fig. 1 (BB) for location). Place slot cover aside - it will not be needed.

STEP #2: RGB Board Installation (Refer to Fig. 2)

*NOTE: Configure the jumpers on the RGB board for your type of monitor - See RGB Manual for configurations.

1) [ ] Remove screw located near the outer corner of connector J-11 (Connector (C) on Fig. 2) (See Fig. 2 (F) for screw location): Discard screw.

2) [ ] Install the metal spacer in the hole provided by removing the screw in the above step (See Fig. 2 (A), (F) ).

3) [ ] Press (Install in direction shown: Fig. 2 (B) ) The plastic spacer in the hole on the motherboard located between the drive #2 connector and the IC located at U109 (Refer to Fig. 2 (B) ).

4) [ ] Align the corner and tab holes on the RGB board with the metal and plastic spacers installed on the motherboard (Refer to Fig. 2 (A) , (B) ).

5) [ ] Press the RGB board down, check to see that the RGB connector (C) is properly mated to the mother board connector (C) (Refer to Fig. 2).

6) [ ] Install and tighten the hex nut to the top of the metal standoff (See Fig. 2 (A) ).

STEP #3: RGB Cable Installation (Refer to Figs. 2 & 3)

1) [ ] Arrange the RGB cable so that the female connector is placed in slot cover #6, the ferrite coil will be on the side of the connectors pointing towards the power supply (See Fig. 2 Ⓔ).

2) [ ] Attach the RGB cable to the back of the base plate with the pair of screws, lock washers and nuts provided (Refer to Fig. 2 ⑤, ⑥, Fig. 3 ⑥), tighten.

3) [ ] Locate on the motherboard the area marked "spare" (Location U82), peel one side off of the "sticky pad" and attach the pad to "spare".

4) [ ] Connect the unattached end of the RGB cable to the connector on the topside of the RGB board (Refer to Fig. 2 Ⓓ, Fig. 3).

*NOTE: These connectors are "polorized" and should only be connected with their "keys" lining up (See Fig. 2 ②).

5) [ ] Peel the topside off of the "sticky pad" installed in Step #3-3. Press the ferrite coil (The one closest to the RGB board) to the pad (Refer to Fig. 3 ③, ④).

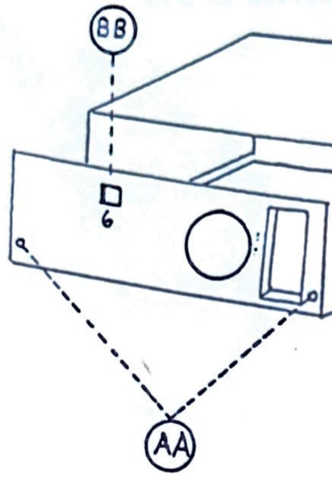
STEP #4: Replace Cover

1) [ ] Check to see that all cables leading from the cover to the motherboard have not become detached.

2) [ ] Slide cover onto the base plate and attach the two screws removed in Step #1-2 (Refer to Fig. 1 (AA))

3) [ ] Reconnect all cables removed during Step #1-1.

FIGURE 1



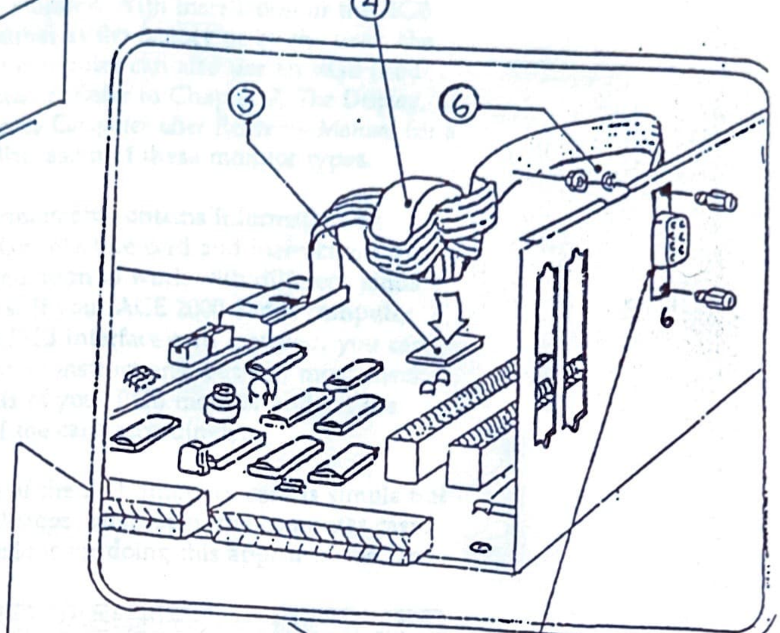




FIGURE 3

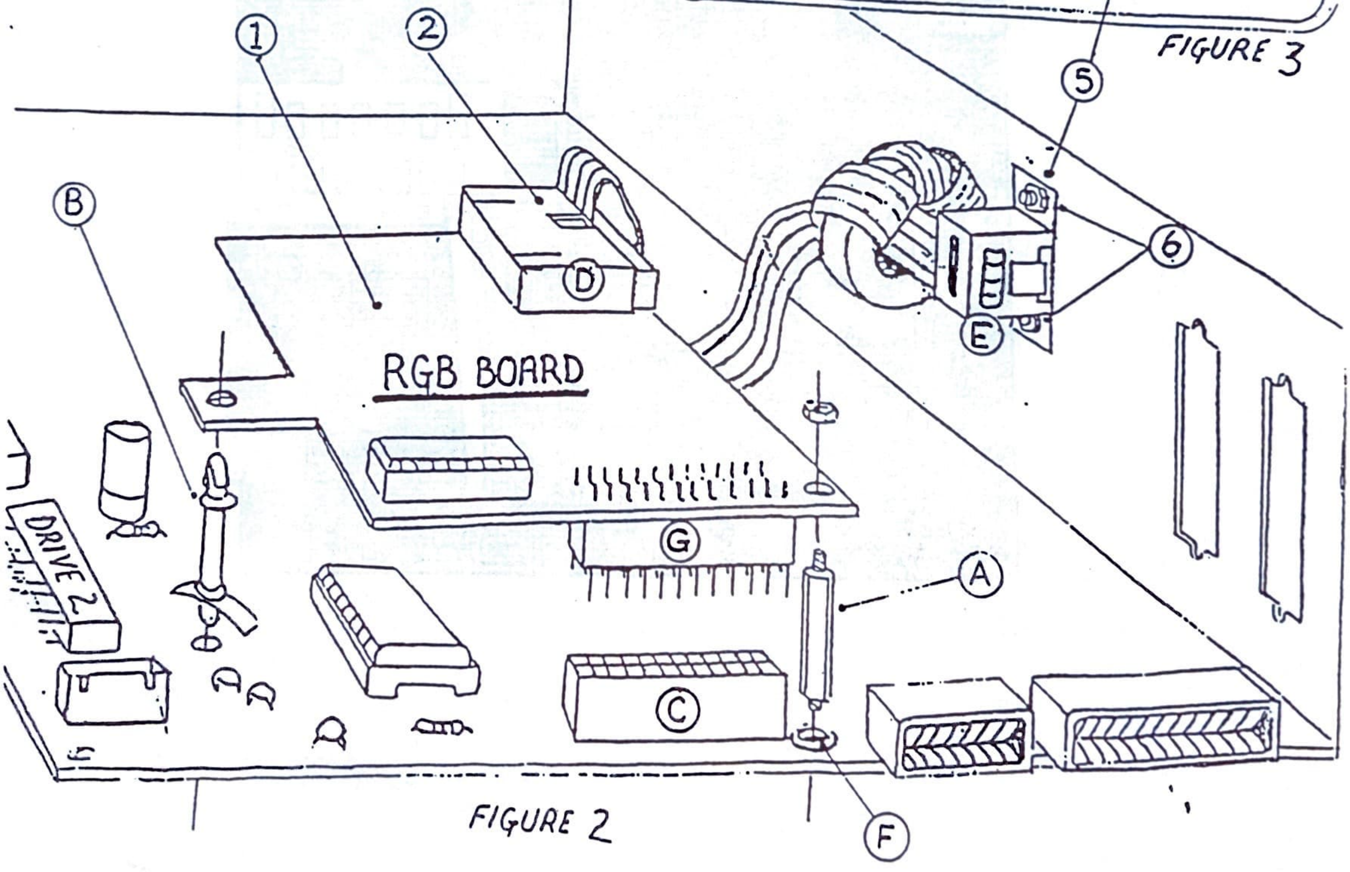



FIGURE 2

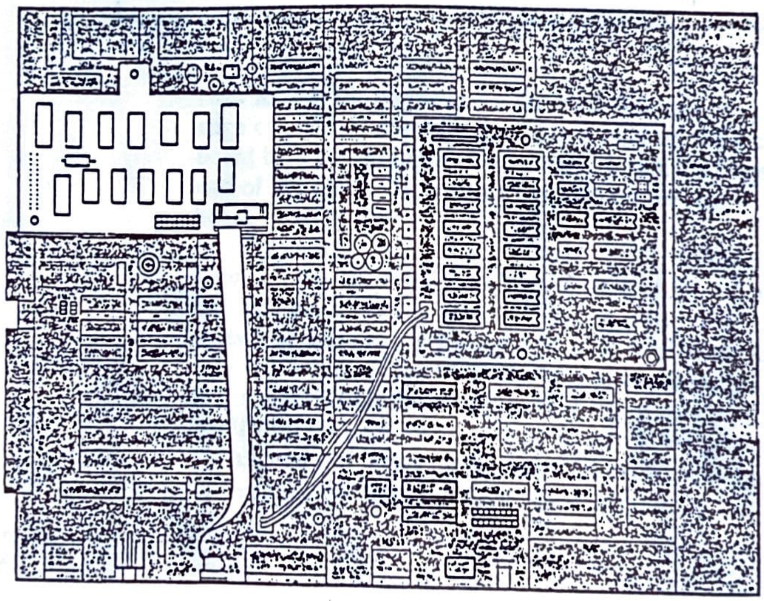
# RGB Interface Card

All Franklin ACE 2000 Series computers can use a composite color monitor. With installation of the RGB Interface card, either at the factory or by the user, the ACE 2000 Series computer can also use an RGB (Red-Green-Blue) monitor. Refer to Chapter 7, *The Display*, of the *ACE 2000 Series Computer User Reference Manual* for a more detailed discussion of these monitor types.

This manual supplement contains information on installing the RGB Interface card and instructions on setting its configuration to work with different kinds of RGB monitors. If your ACE 2000 Series computer already has the RGB Interface card installed, you can skip the installation instructions, but you must check the requirements of your RGB monitor and set the configuration of the card accordingly.

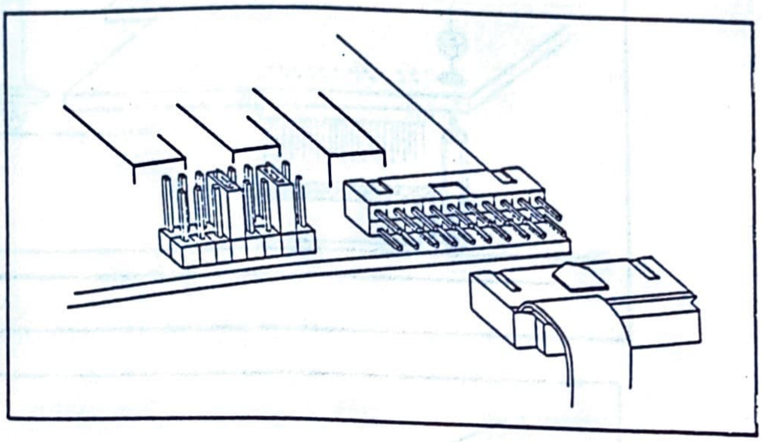
**Installing The RGB Interface Card**

The installation of the RGB Interface card is simple but involves several steps. First, open the computer case. Detailed instructions for doing this appear in the *User's*



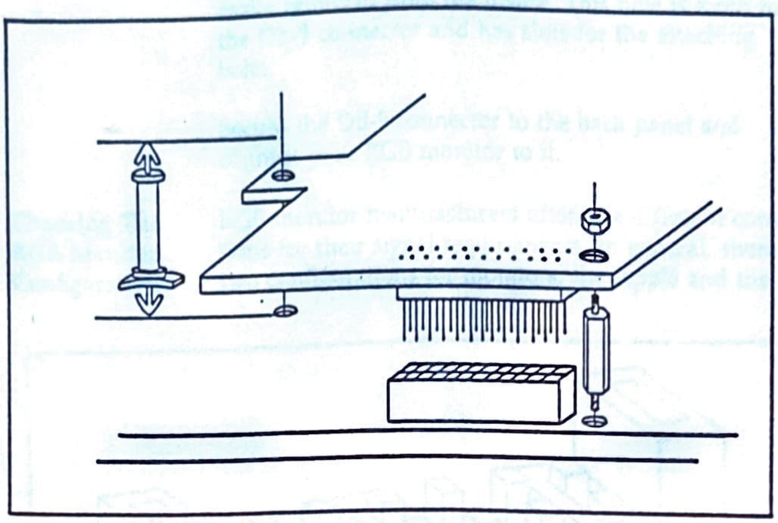
*Reference Manual,* pages 1-19 through 1-20. When the computer is open, locate the RGB connector on the right side of the main board.

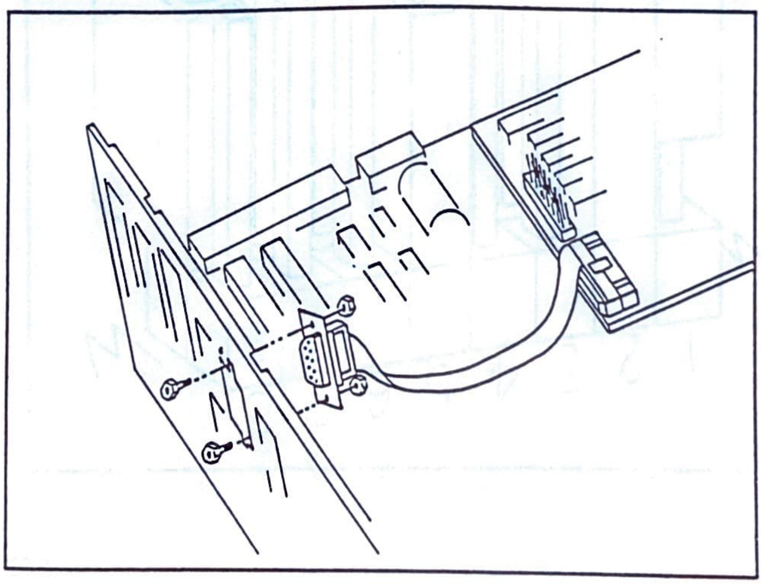
Before you insert the card's connectors into the RGB socket, connect the ribbon cable to the connector on the card. The 20-pin connector is keyed so the plug fits only one way.



Now look at the supports for the card. The tab on the edge of the card opposite the cable is held on the main board by a plastic support, as shown. Insert the large end of the plastic support into its hole on the main board first.

Then remove the screw holding the edge of the main board to the computer case and replace it with the metal standoff piece provided that acts as an extension screw. Now insert the RGB Interface card connectors into the socket on the main board, making sure you don't shift the pins one row right or left. If the pins are not aligned, the holes for the supports won't line up. Then attach the RGB Interface board with the nut provided to the top of the metal standoff and snap the plastic support into the hole on the card's tab.



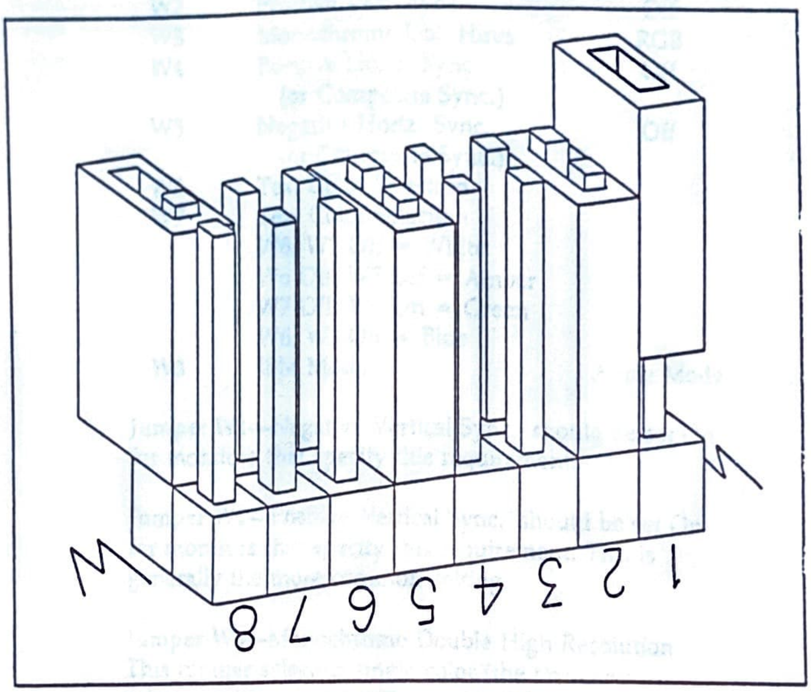


Lay the cable flat along the main board (it must lie flat to pass under any peripheral cards which may be resident in the expansion slots) and lead the cable connector, called a DB-9 connector, up to hole #6 in the back panel. The metal tabs covering the holes are easily removed from the inside. This hole is sized for the DB-9 connector and has slots for the attaching bolts.

Secure the DB-9 connector to the back panel and connect your RGB monitor to it.

### Choosing The RGB Monitor Configuration

RGB monitor manufacturers often use different conventions for their signal requirements. In general, there are two configurations for monitors, the Apple and the



IBM, which will be discussed in more detail below. Consult the monitor's manual to see which settings on the RGB Interface card are required, and whether the Apple or the IBM configuration meets the needs of your RGB monitor.

The RGB Interface card has a set of 8 jumpers with shorting pins which select the configuration used by the computer. Six shorting pins are included for choosing different configurations. The shorting pins can be stored by slipping them over ONE of the posts. The pins have the following assignments:

| Jumper | Jumper Clip On | Clip Off |
|---|---|---|
| W1 | Negative Vert. Sync. | Off |
| W2 | Positive Vert. Sync. | Off |
| W3 | Monochrome Dbl. Hires | RGB |
| W4 | Positive Horiz. Sync. (or Composite Sync.) | Off |
| W5 | Negative Horiz. Sync. (or Composite Sync.) | Off |
| W6 | Text Color Selection | |
| W7 | Text Color Selection<br>W6, W7 Off = White<br>W6 On, W7 Off = Amber<br>W7 Off, W7 On = Green<br>W6, W7 On = Blue | |
| W8 | IBM Mode | Apple Mode |

Jumper W1—Negative Vertical Sync. should be set On for monitors that specify this requirement.

Jumper W2—Positive Vertical Sync. should be set On for monitors that specify this requirement. This is generally the more common setting.

Jumper W3—Monochrome Double High Resolution This jumper selects a single color (the text color selected with W6 and W7) on a black background with double high resolution. This is generally only used with specific software applications, such as Computer Assisted Design, where the 4 color pixels normally used for RGB are traded for maximum screen resolution.

Jumper W4—Positive Horizontal Sync. should be selected On for monitors that specify this requirement. This is sometimes called Composite Sync, which means that the (positive) vertical and the horizontal sync signals are together.

Jumper W5—Negative Horizontal Sync. should be selected On for monitors that specify this requirement. This is sometimes called Composite Sync, which means that the (negative) vertical and the horizontal sync signals are together.

Jumpers W6 and W7—Text Color Selection These jumpers select the color of the text displayed on the black background according to the following arrangement:

W6, W7 Off = White
W6 On, W7 Off = Amber
W6 Off, W7 On = Green
W6, W7 On = Blue

The factory setting has white text selected.

Jumper W8—Apple/IBM Mode This jumper selects the configuration of the color matrix that the computer uses, either according to the Apple or the IBM formats.

If your RGB monitor says "Apple compatible" or "IBM compatible", or has some other indication that it is designed to work with one of those computers, try the configuration suggested for that computer.

**Suggested Configuration:**

| Apple | IBM |
| --- | --- |
| W1 – Off | W1 – Off |
| W2 – On | W2 – On |
| W4 – Off | W4 – On |
| W5 – On | W5 – Off |
| W8 – Off | W8 – On |

The other jumpers (W3, W6, and W7) are user preferences.

If these configurations don't work, consult the manual of the monitor to determine the settings needed. If you still have problems, consult your dealer or the monitor manufacturer.

Franklin
Computer

## EXPANDED FRANKLIN RAM ADDENDUM REV 1.0

The orginal Extended Memory Card Manual was written based on a utility disk containing the following files:

Hello
Framdisk
Pramdisk
Ramtest
Awstartup11
Awstartup12

Current products will be shipped with the files "Framdisk and Ramtest" added to the Franklin DOS disk. These DOS disks can easily be indentified by an orange dot attached to the label or doing a disk catalog.

The utility disk supplied with the Extended Memory Card contains the following files:

Hello
Pramdisk
Awstartup11
Awstartup12

NOTE: This new utility has its files saved in "PRO-DOS" format. As a result, you can NOT catalog or access files on this disk with FDOS.

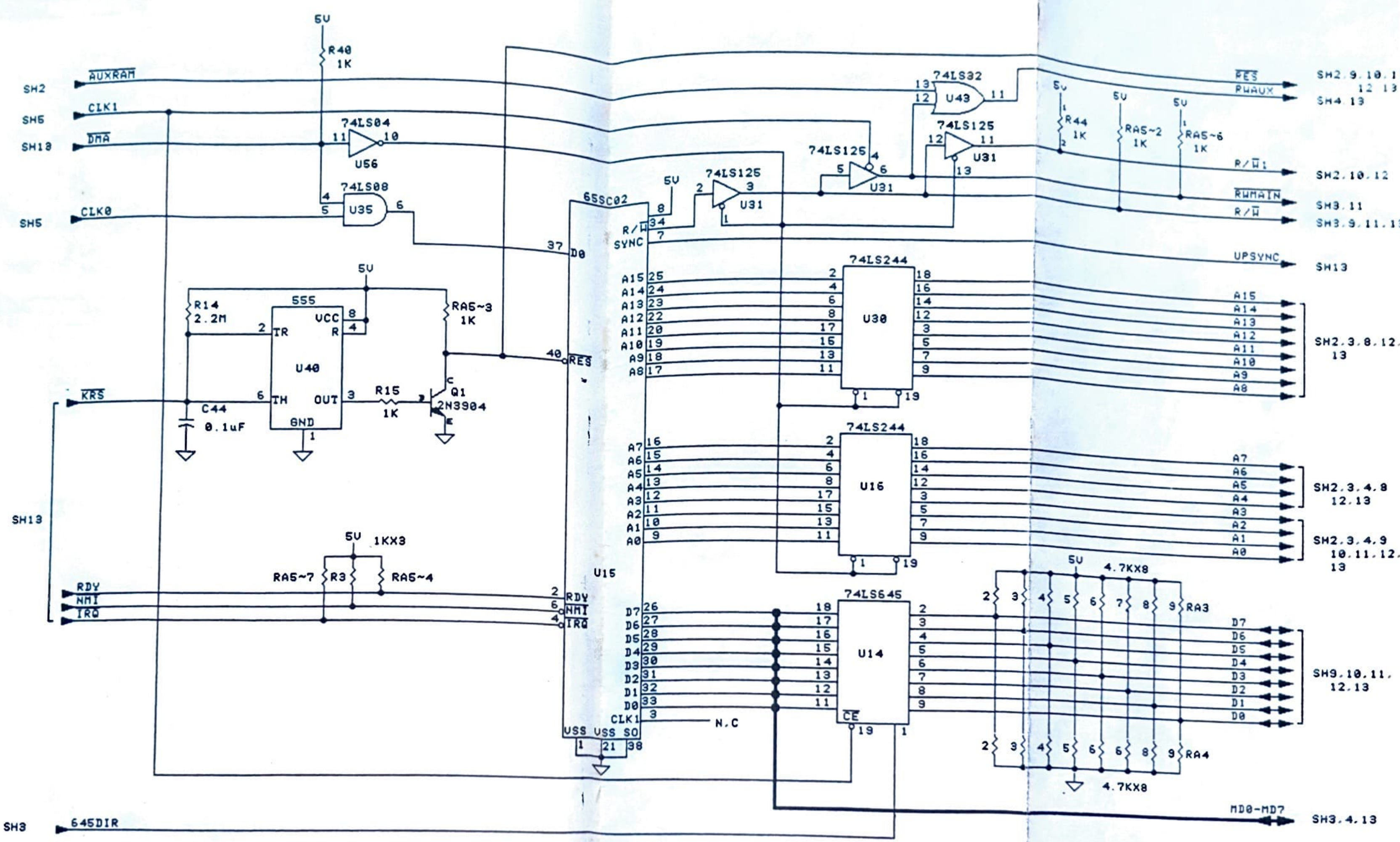
A PRO DOS utility disk is required to use the programs on this disk. The utility disk is in PRO DOS format to eliminate the need to convert these files from FDOS to PRO DOS.

Because of these changes it is no longer required for the user to convert these files as stated at the bottom of page 11-10 and continued to the top of page 11-11.

On page 11-12, 2nd paragraph, it specifies renaming "Appleworks. System" file to "AW". The actual name on the disk is "Aplworks.System".

Both "Awstartup11 and 12" files will work on single drive computers. These programs will NOT increase desktop space.

1/31/86

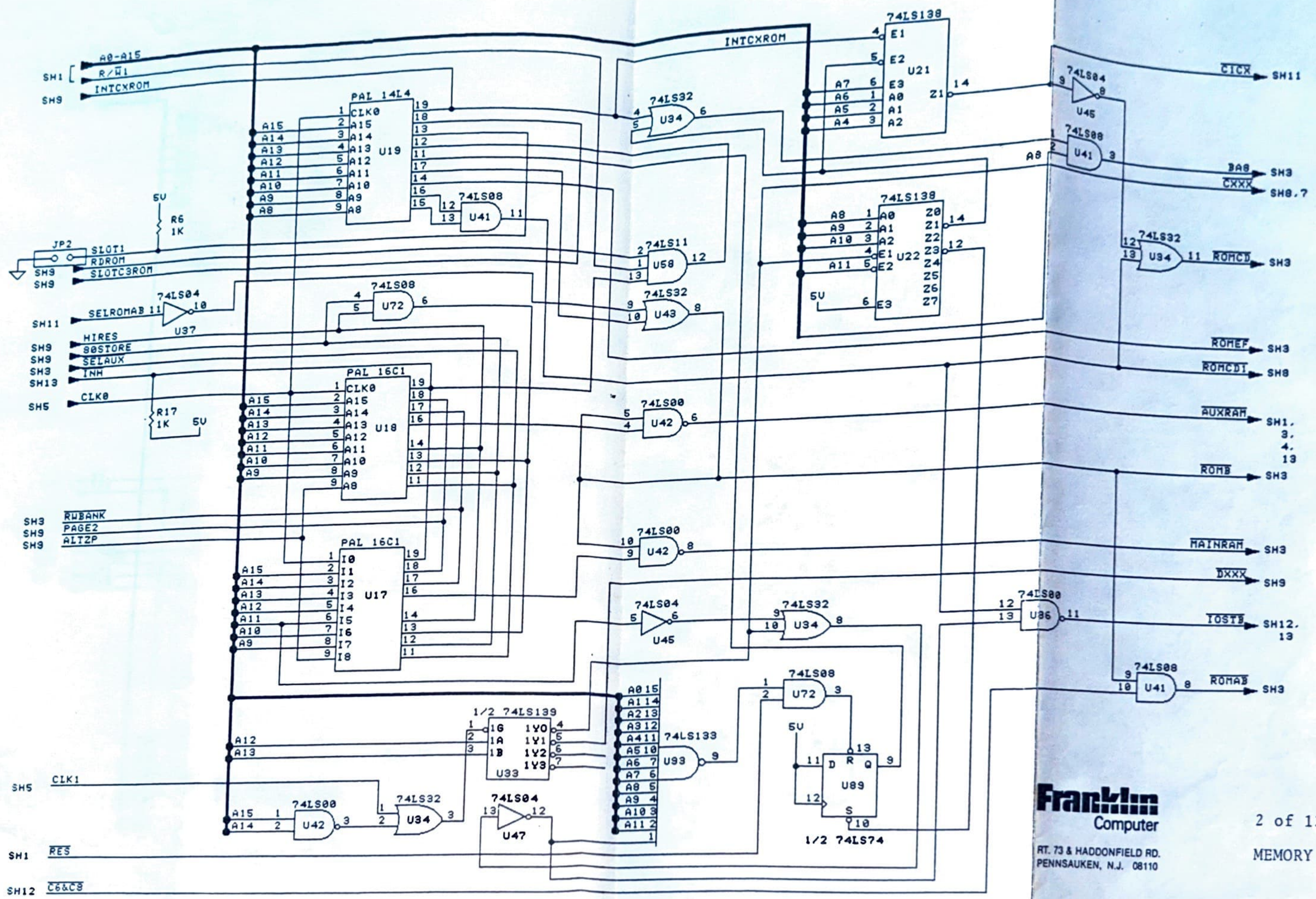



**Franklin Computer**

RT. 73 & HADDONFIELD RD.
PENNSAUKEN, N.J. 08110

1 of 13

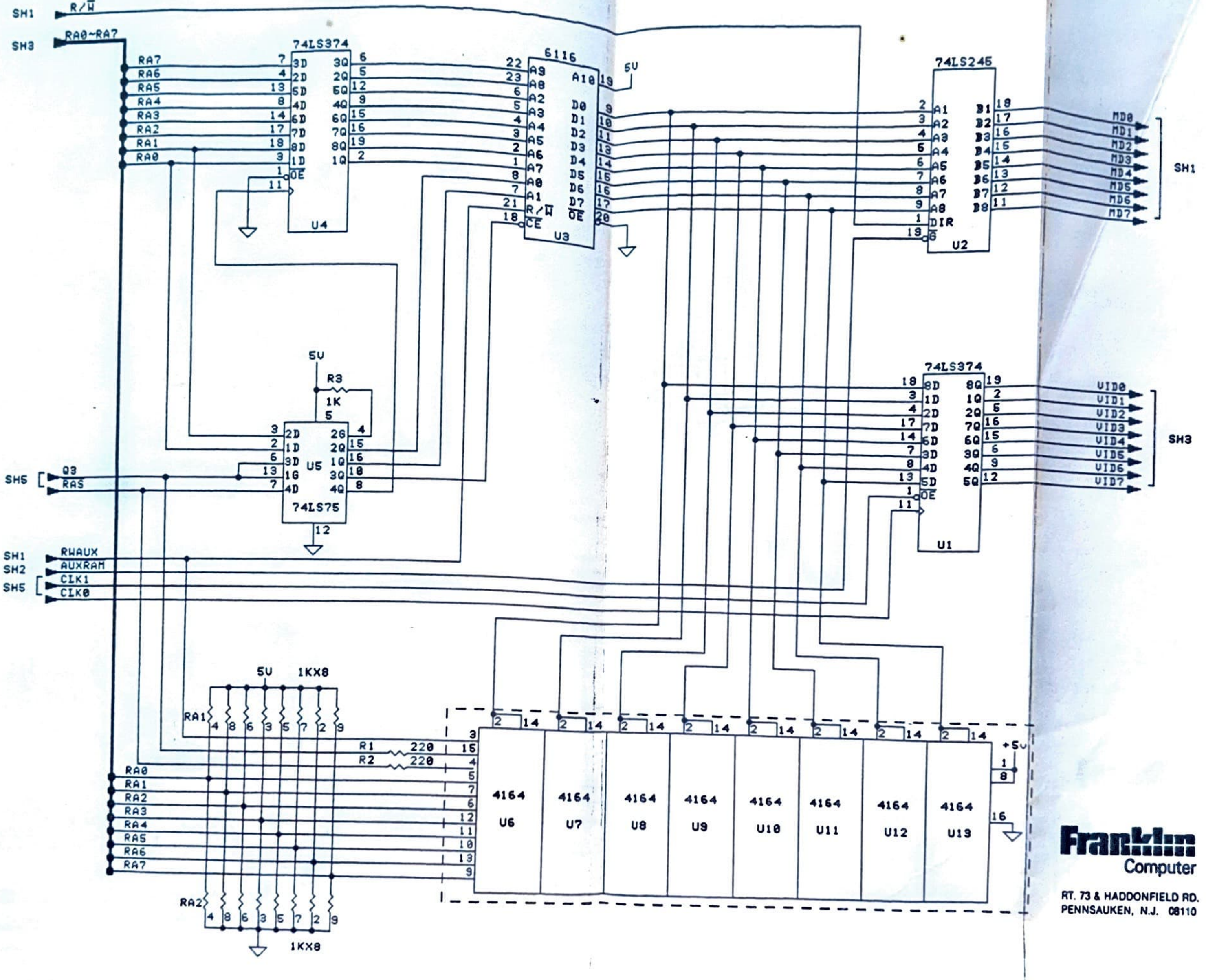
CPU



Franklin Computer
RT. 73 & HADDONFIELD RD.
PENNSAUKEN, N.J. 08110



MEMORY DECODE

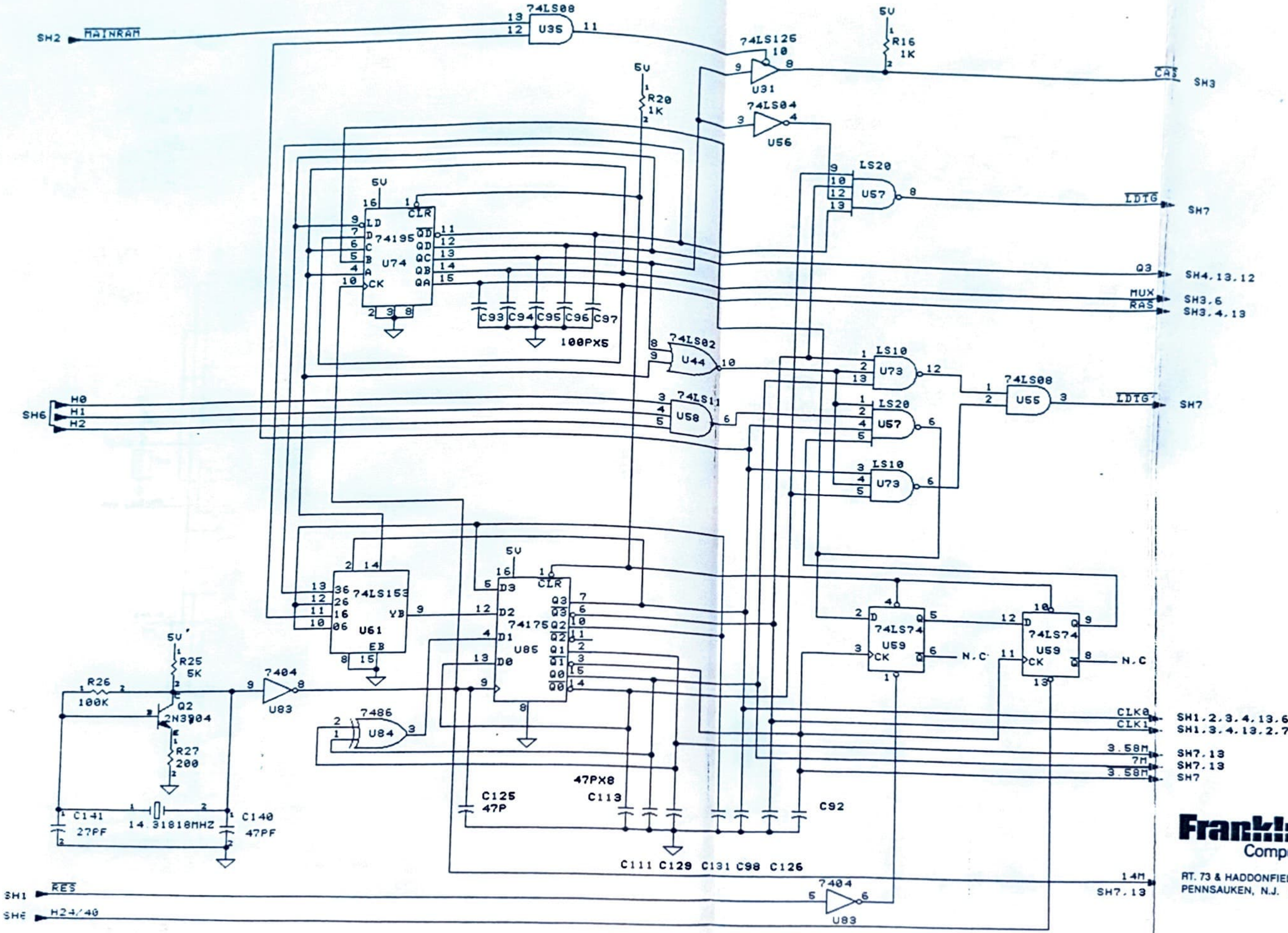



Franklin Computer

RT. 73 & HADDONFIELD RD.
PENNSAUKEN, N.J. 08110

4 of 13

AUX MEMORY

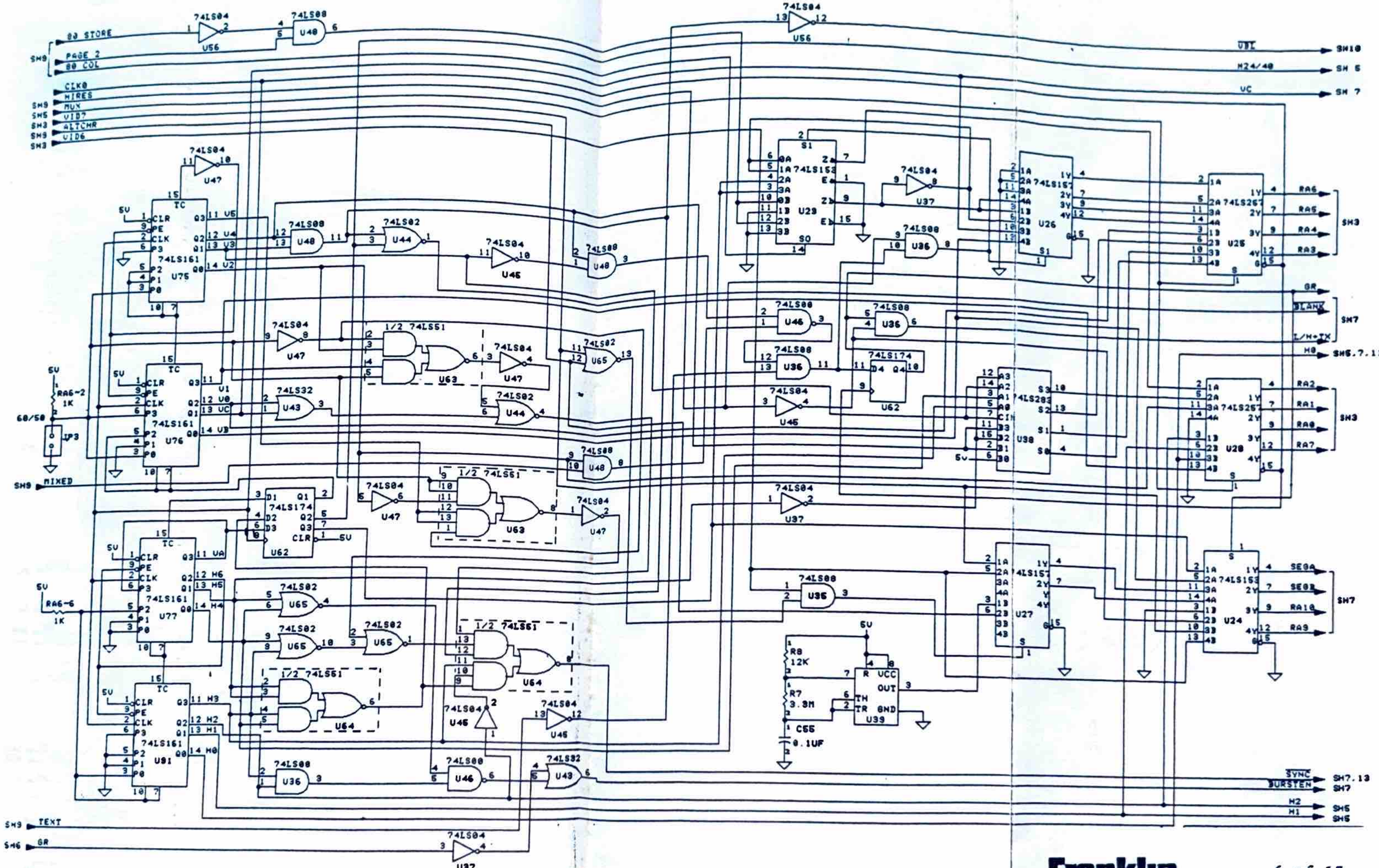



Franklin Computer

RT. 73 & HADDONFIELD RD.
PENNSAUKEN, N.J. 08110



MAIN TIMING

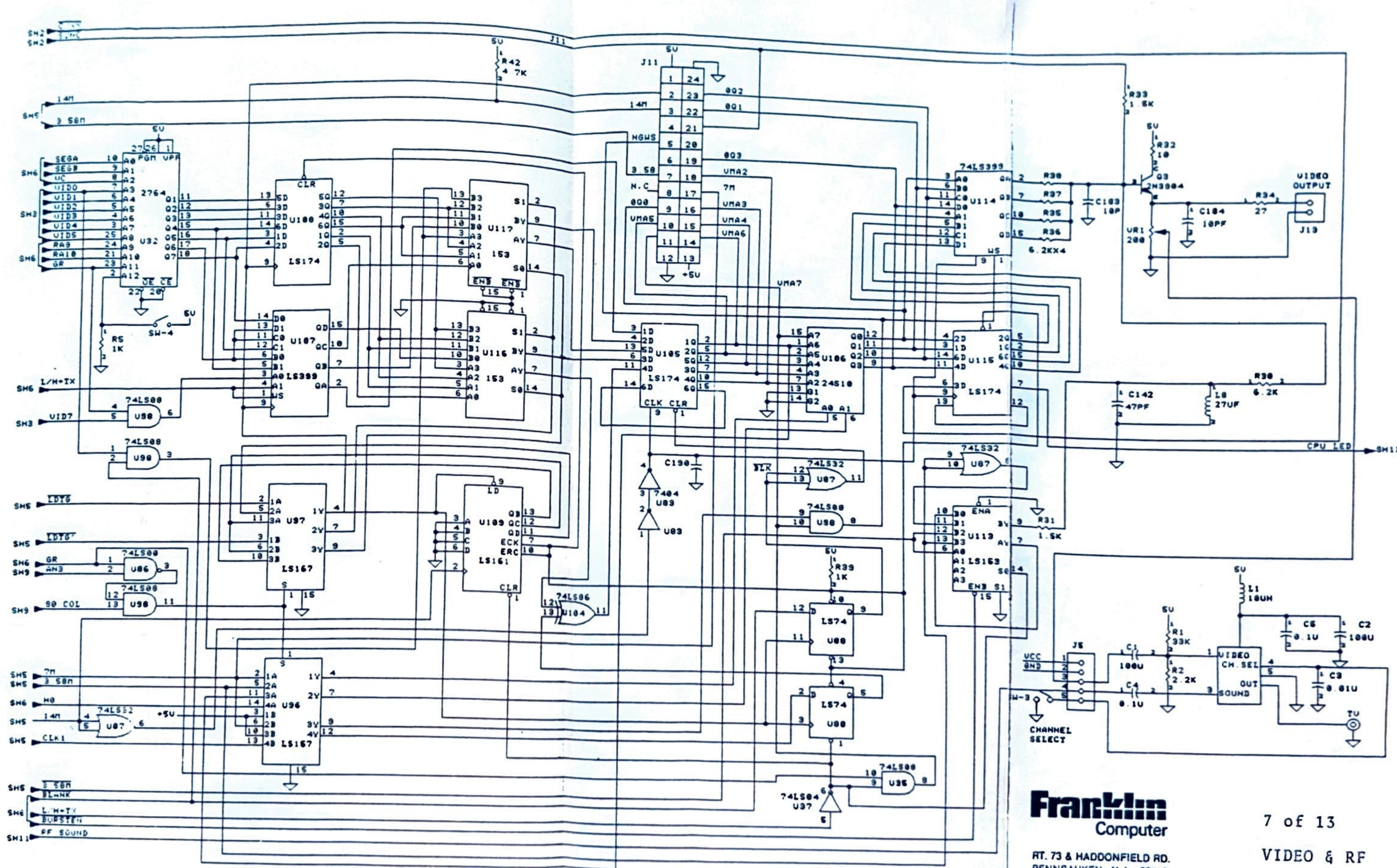


**Franklin Computer**

RT. 73 & HADDONFIELD RD.
PENNSAUKEN, N.J. 08110



TIMING

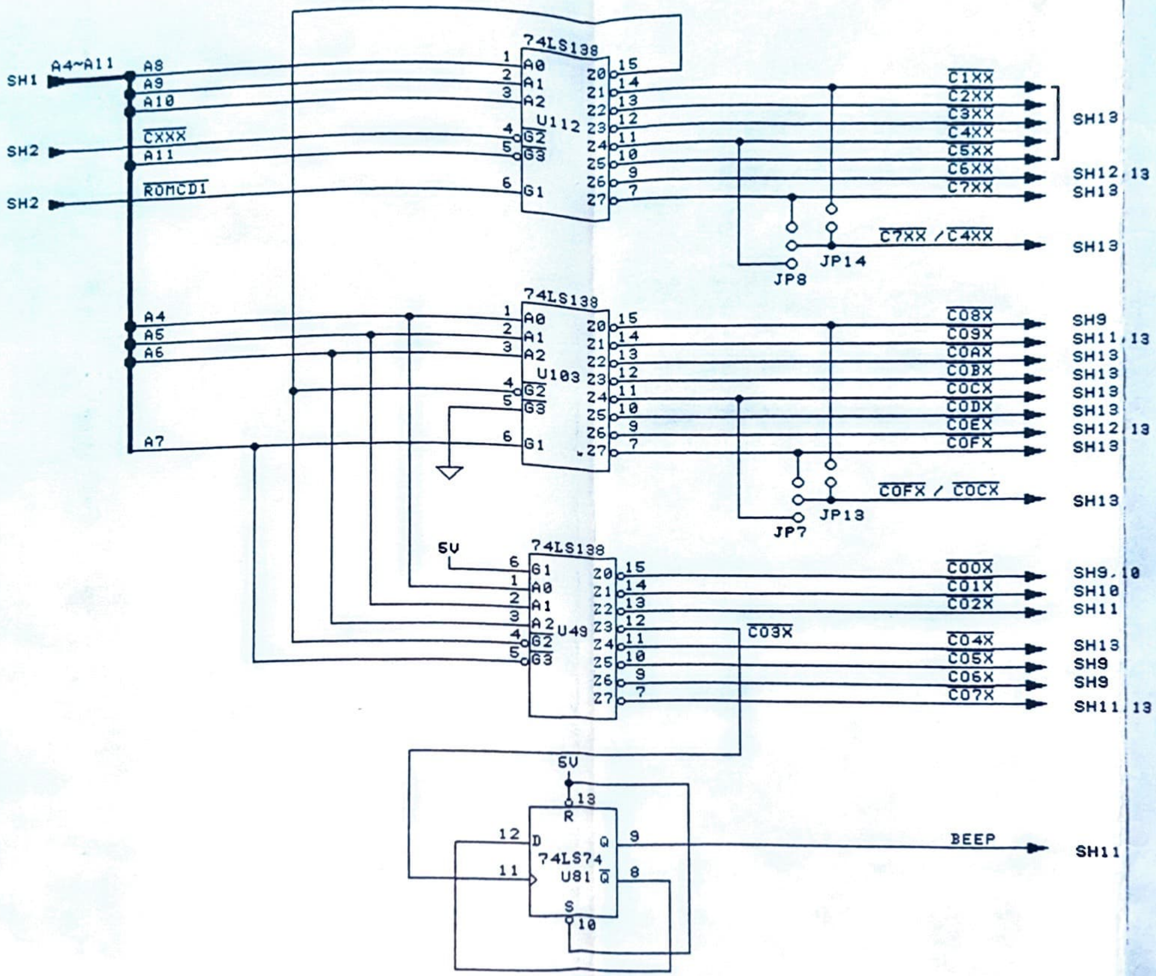



**Franklin** Computer

RT. 73 & HADDONFIELD RD.
PENNSAUKEN, N.J. 08110

7 of 13

VIDEO & RF

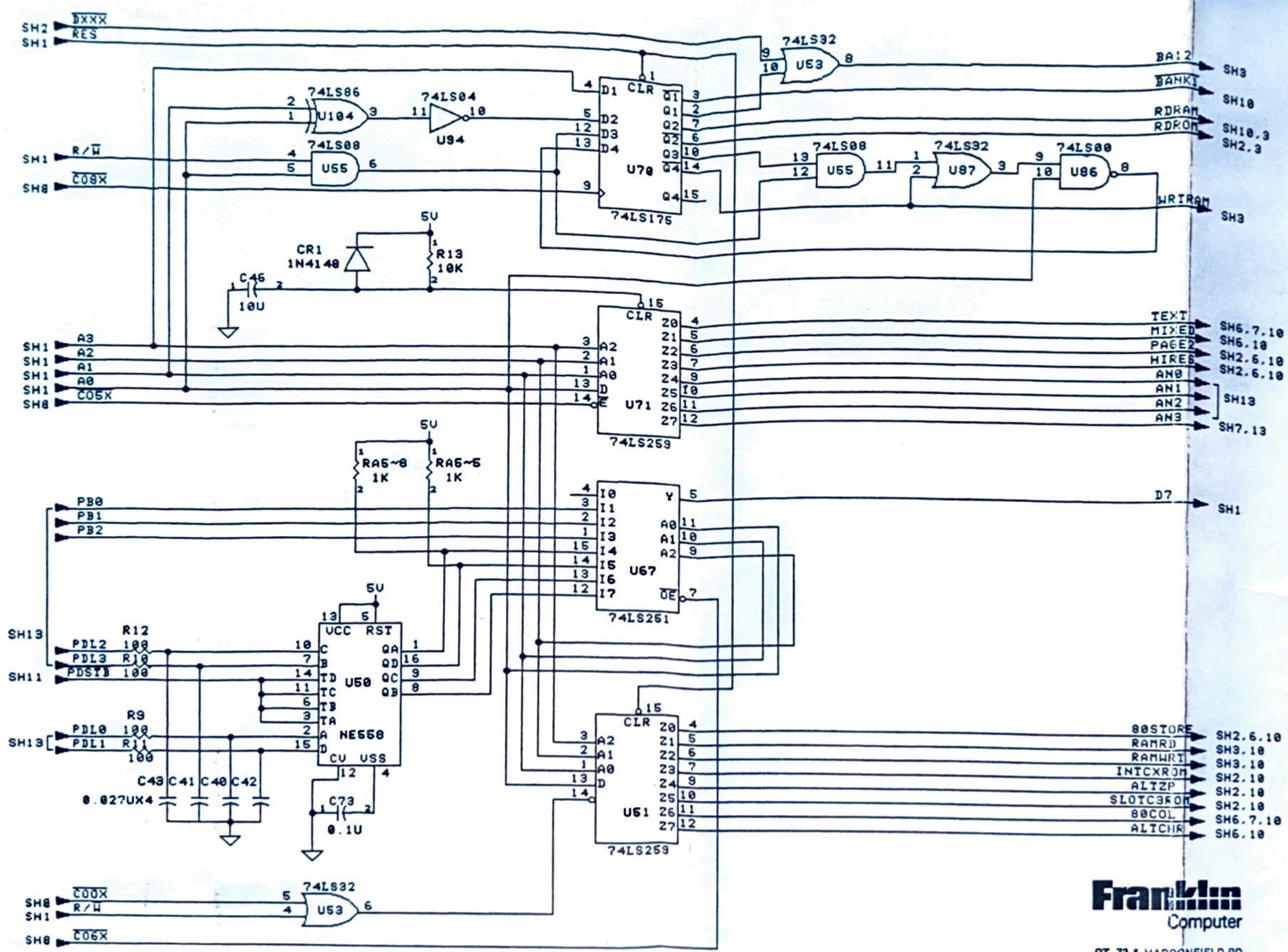



Franklin Computer

RT. 73 & HADDONFIELD RD.
PENNSAUKEN, N.J. 08110



I/O DECODE

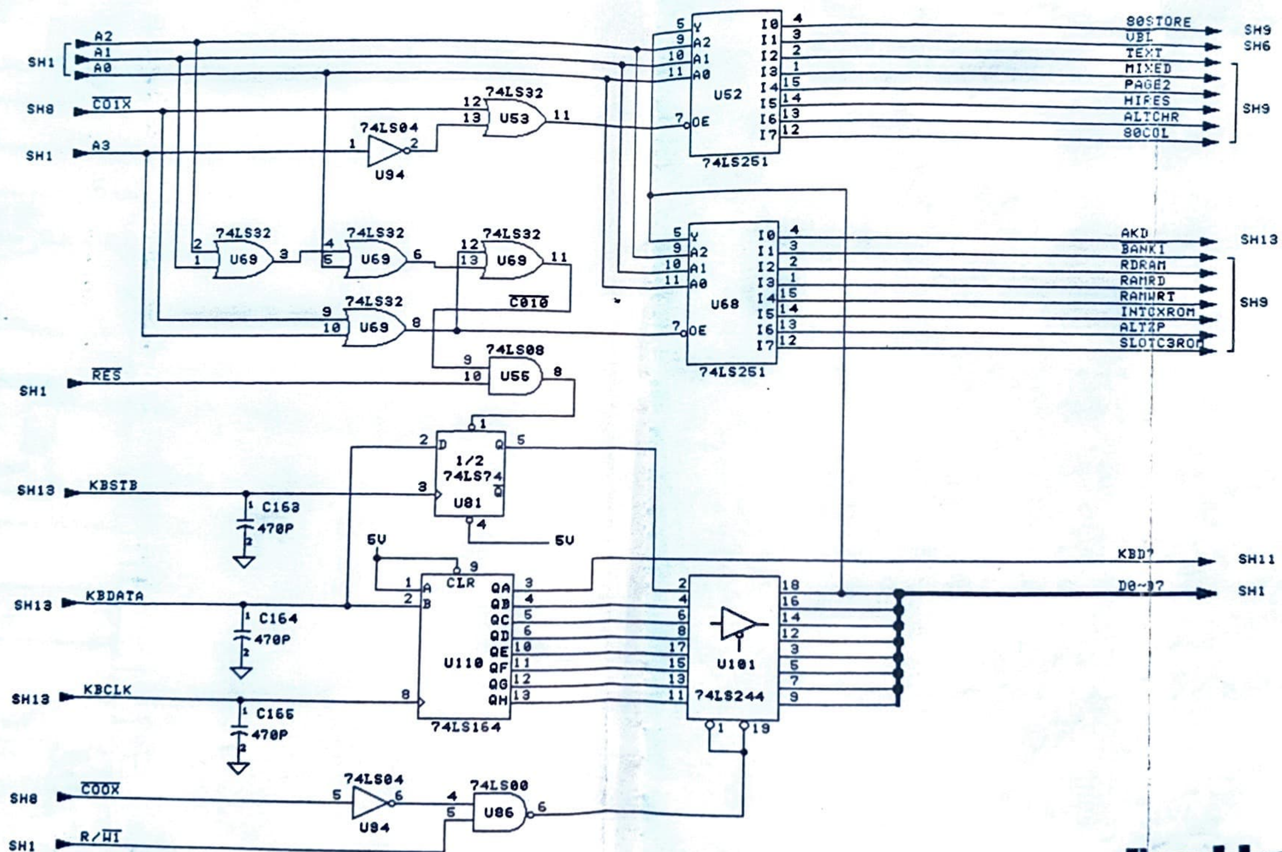



Franklin Computer

RT. 73 & HADDONFIELD RD.
PENNSAUKEN, N.J. 08110



GAME I/O & SOFTSWITCHES

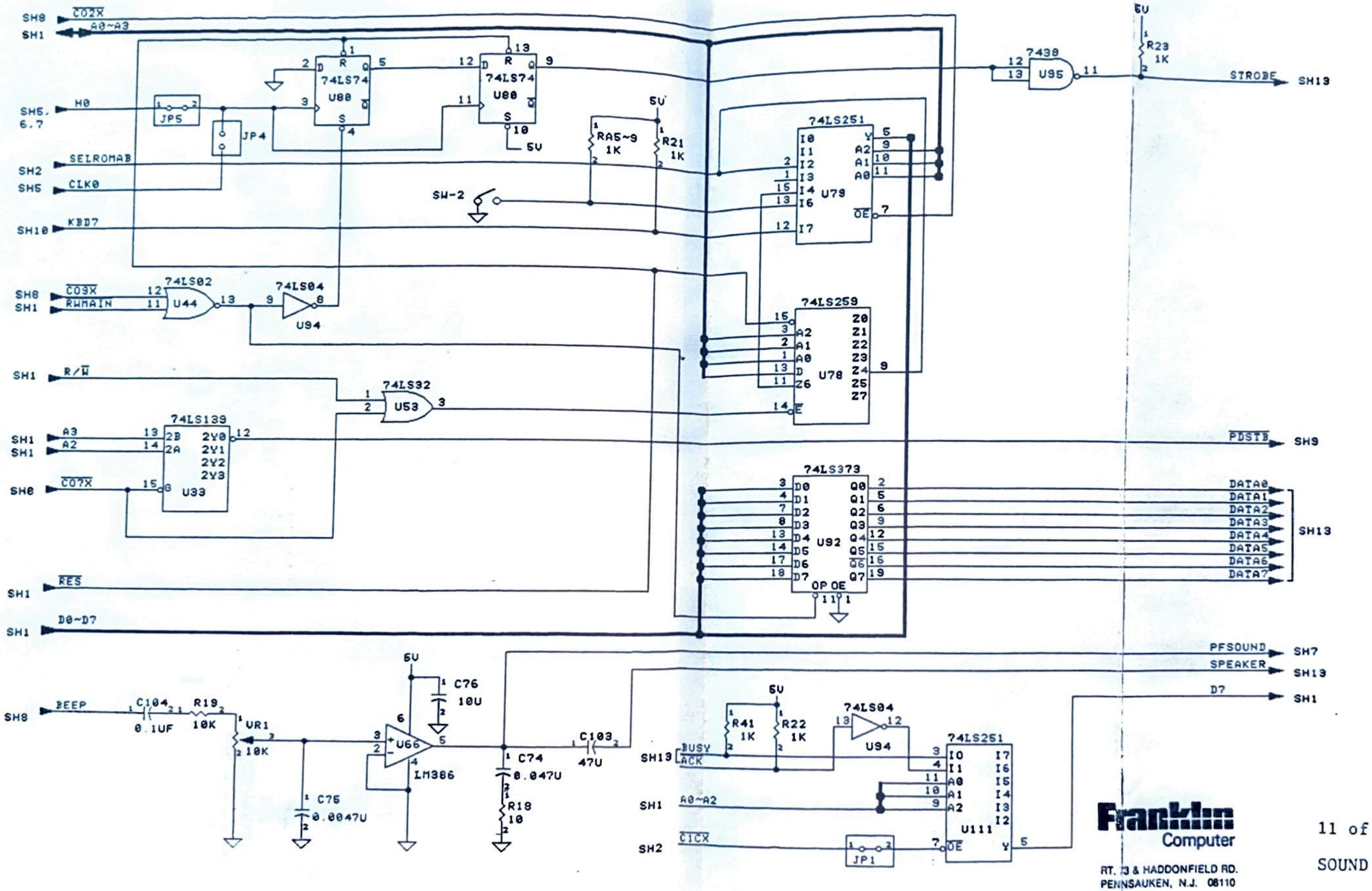



**Franklin Computer**

RT. 73 & HADDONFIELD RD.
PENNSAUKEN, N.J. 08110



KEYBOARD INTERFACE

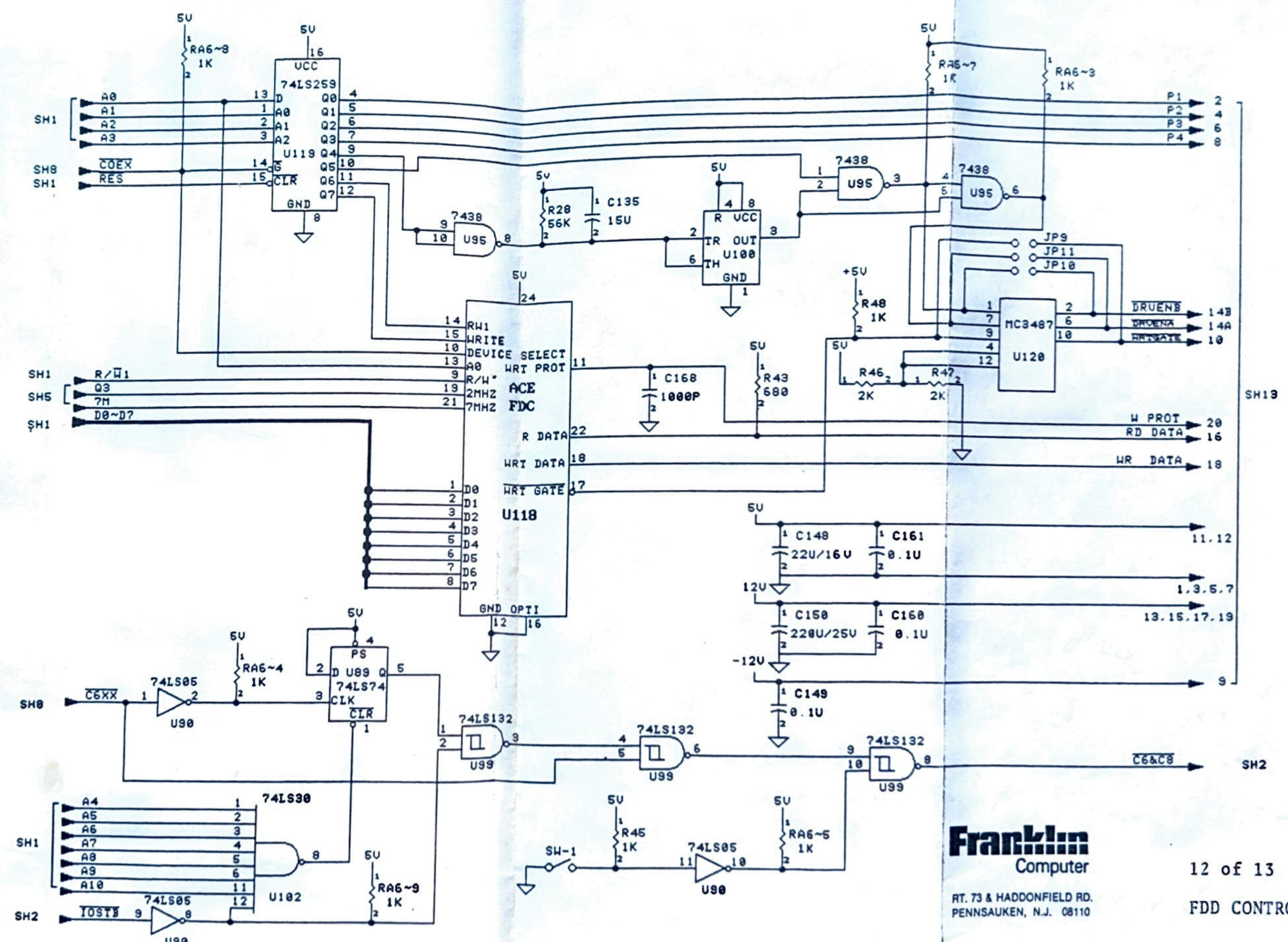



Franklin Computer

RT. 73 & HADDONFIELD RD.
PENNSAUKEN, N.J. 08110



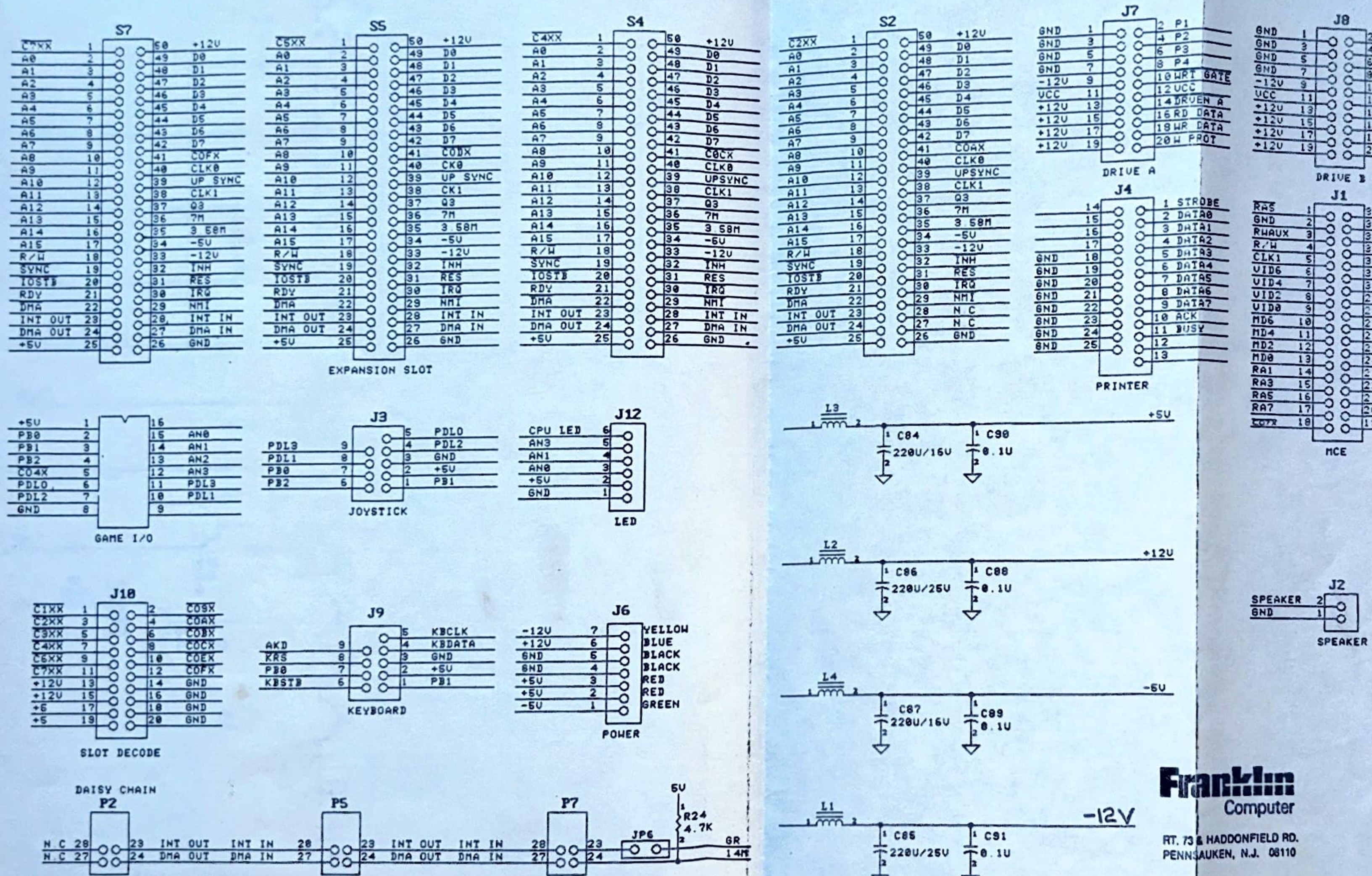
SOUND & PRINTER I/O




Franklin Computer
RT. 73 & HADDONFIELD RD.
PENNSAUKEN, N.J. 08110

12 of 13

FDD CONTROLLER



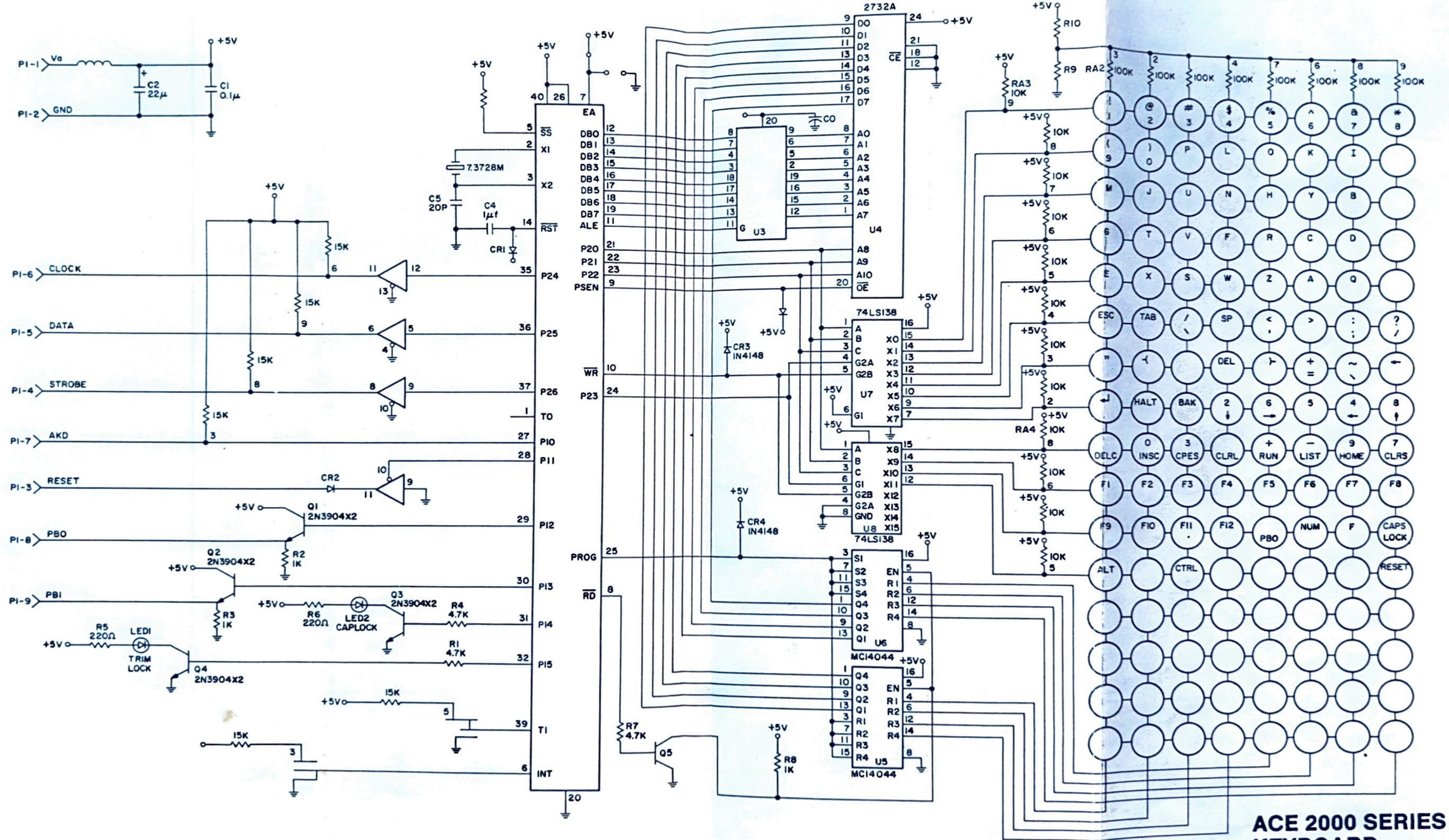

Franklin Computer

RT. 73 & HADDONFIELD RD.
PENNSAUKEN, N.J. 08110



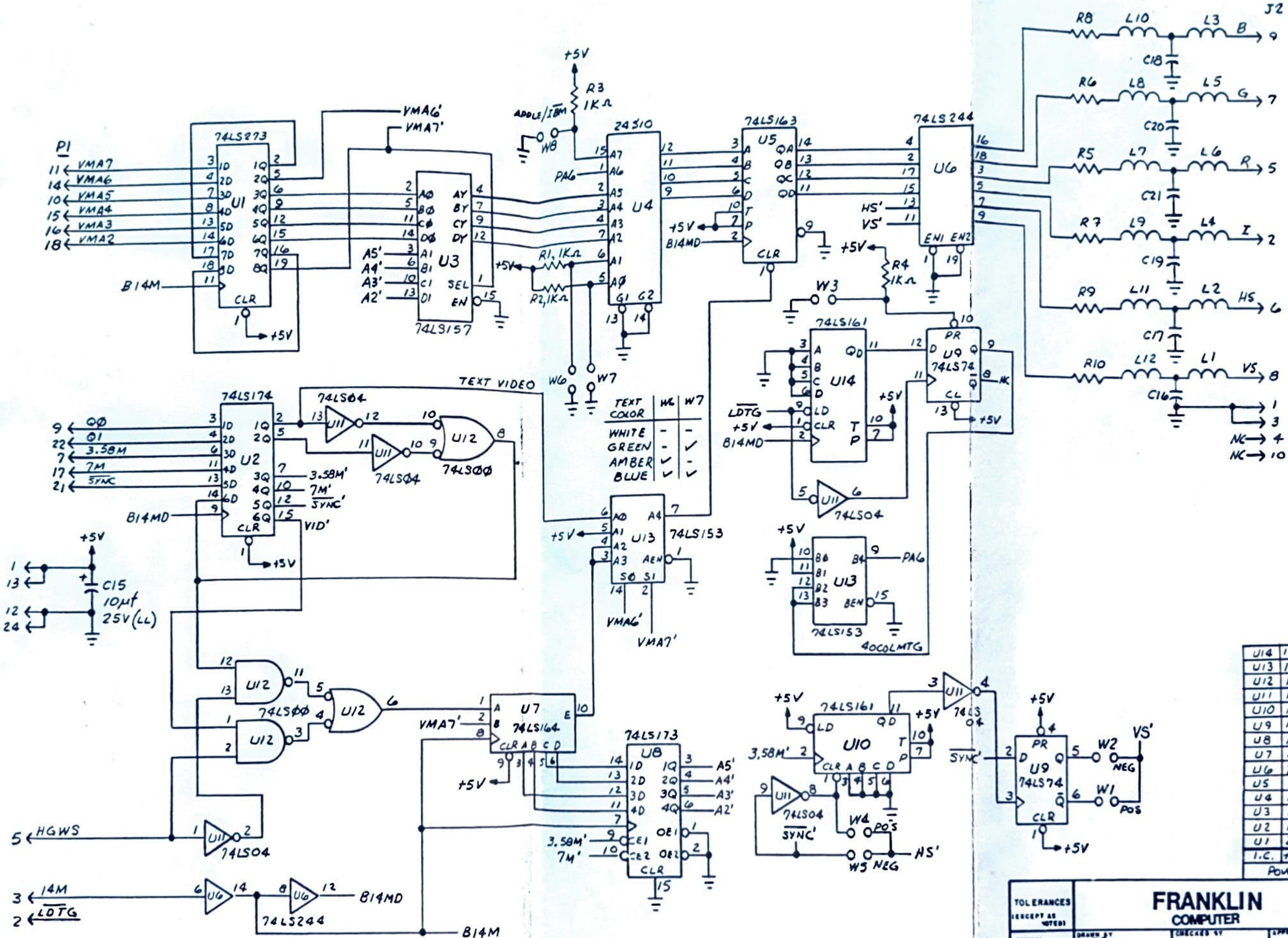
CONNECTOR PINOUTS

POWER DISTRIBUTION




**ACE 2000 SERIES KEYBOARD VERSION 2**

| DATE | SYM | REVISION RECORD | AUTH | DR | CK |
|---|---|---|---|---|---|




| U14 | 16 | 8 | C14 |
|---|---|---|---|
| U13 | 16 | 8 | C13 |
| U12 | 14 | 7 | C12 |
| U11 | 14 | 7 | C11 |
| U10 | 16 | 8 | C10 |
| U9 | 14 | 7 | C9 |
| U8 | 16 | 8 | C8 |
| U7 | 14 | 7 | C7 |
| U6 | 20 | 10 | C6 |
| U5 | 16 | 8 | C5 |
| U4 | 16 | 8 | C4 |
| U3 | 16 | 8 | C3 |
| U2 | 16 | 8 | C2 |
| U1 | 20 | 10 | C1 |
| I.C. | +5V | GND | CAP |

POWER PINS

NOTE
1. UNLESS OTHERWISE SPECIFIED
a) RES. ARE 1/4W, 5%
b) CAPS ARE 0.1µf, 50V

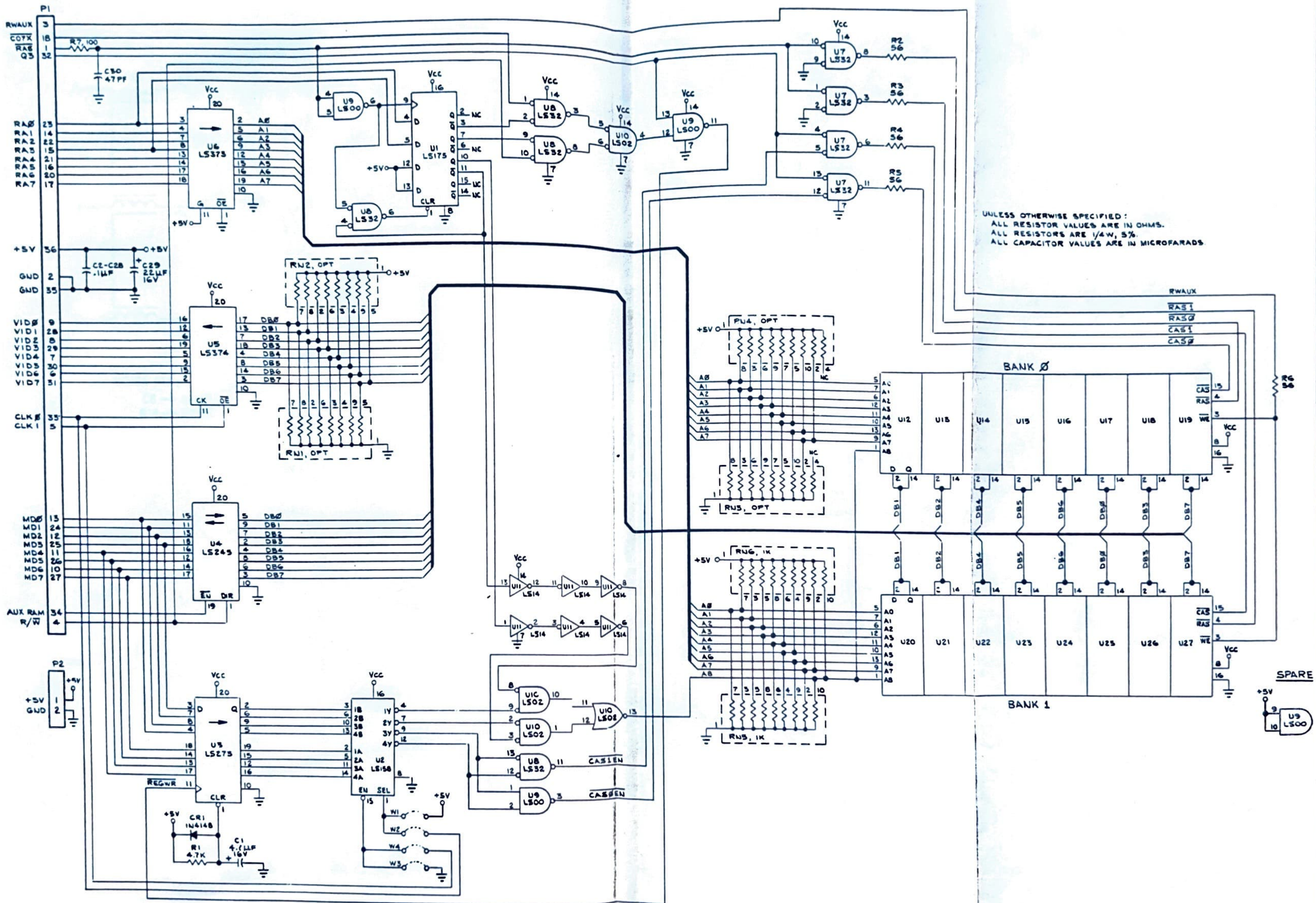
**FRANKLIN COMPUTER**

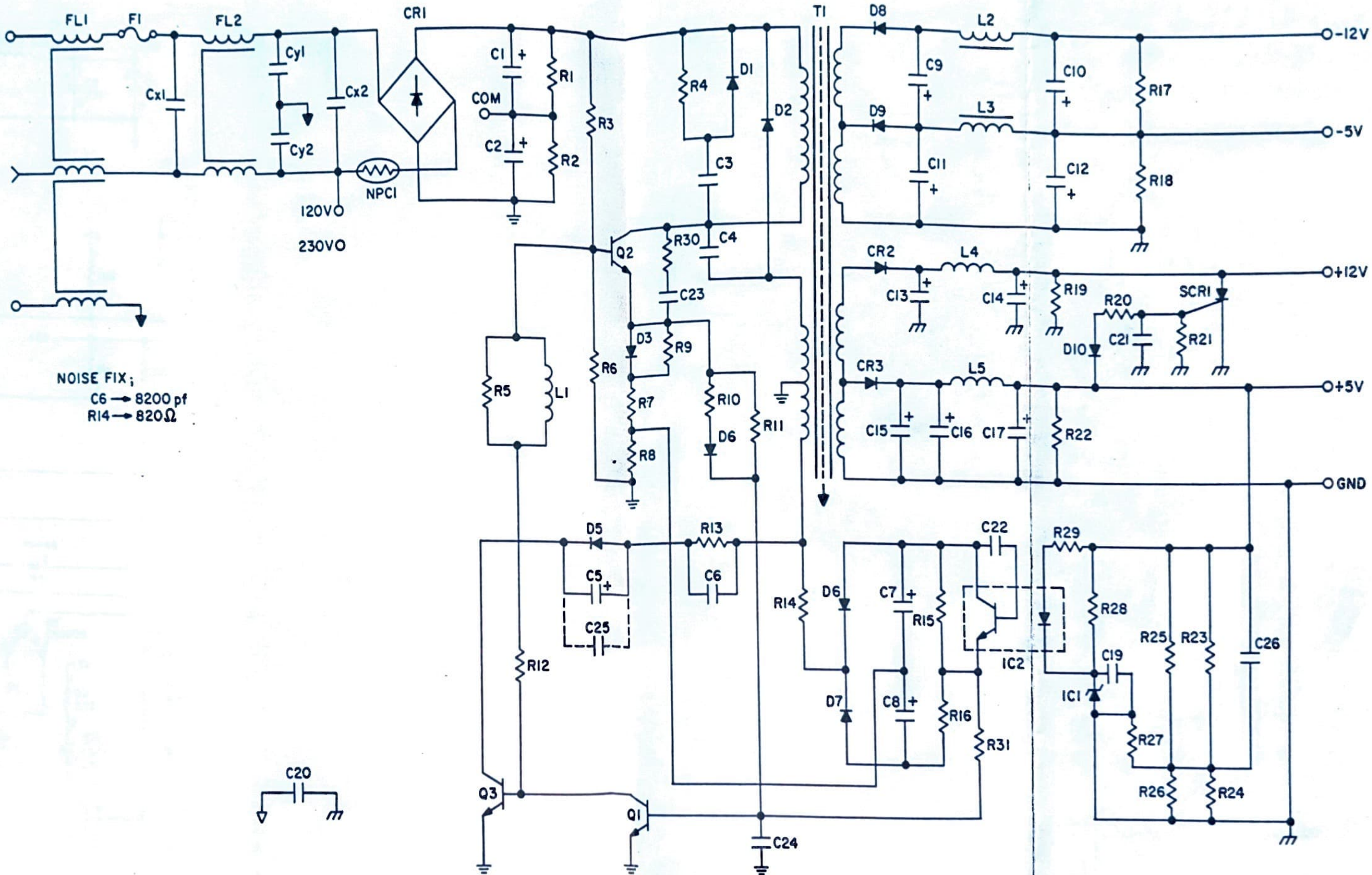
TOLERANCES (EXCEPT AS NOTED) DECIMAL ± — FRACTIONAL ± — ANGULAR ± —

DRAWN BY EDA | CHECKED BY | APPROVED BY
SCALE — | RELEASED TO PROD. | DATE

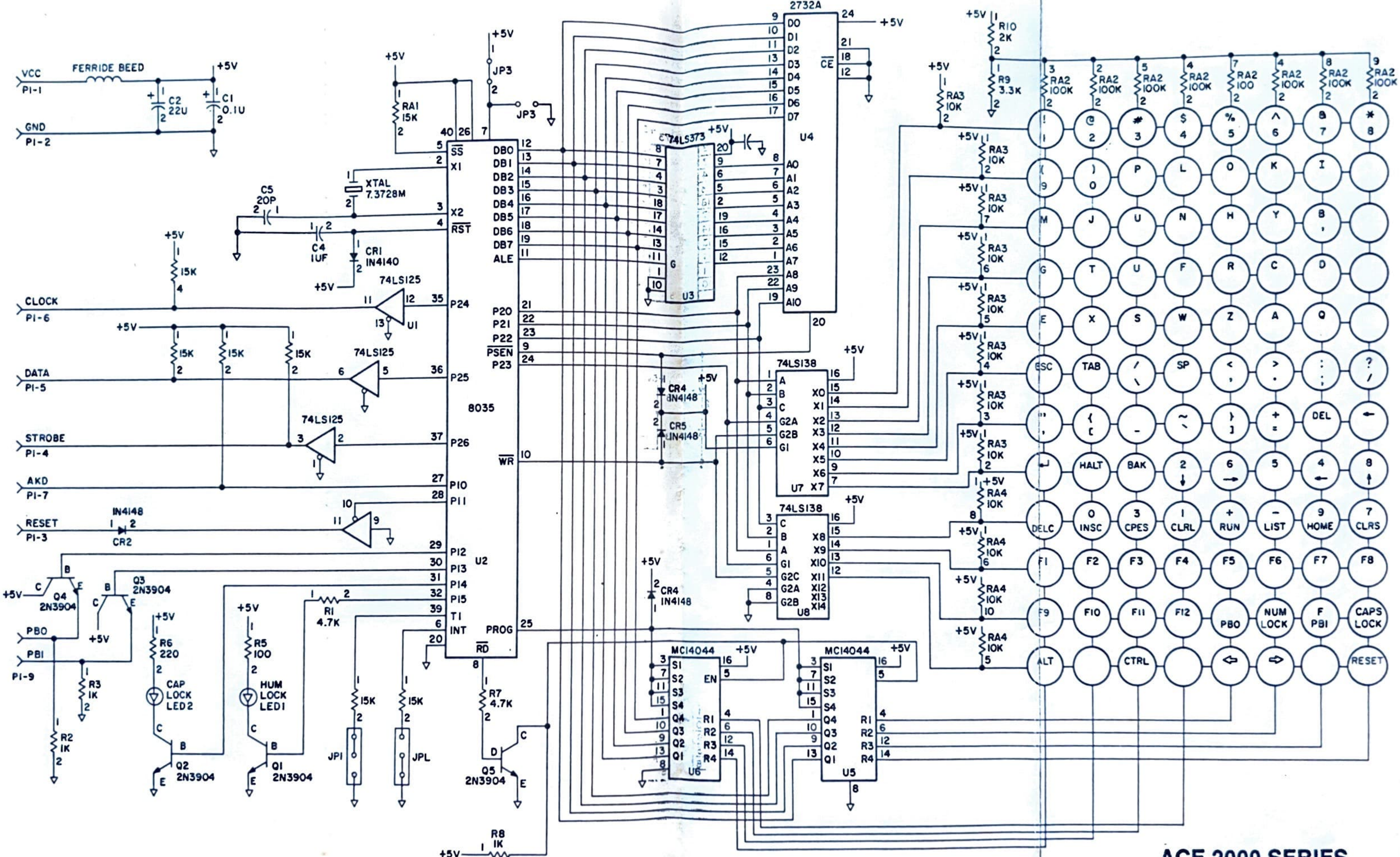
TITLE: SCHEMATIC RGB BOARD

DATE 12/30/85 | SIZE D | DRAWING NUMBER FC111585 | SHT 1 OF SHT 1


P1
RWAUX
CO7X
RAS
Q3
R7, 100
C30 47PF
RAØ
RA1
RA2
RA3
RA4
RA5
RA6
RA7
U6 LS373
U1 LS175
U9 LS00
U8 LS32
U10 LS02
U7 LS32
R2 56
R3 56
R4 56
R5 56
+5V
GND
C2-C28 .1µF
C29 22µF 16V
VIDØ
VID1
VID2
VID3
VID4
VID5
VID6
VID7
U5 LS374
RN2, OPT
RN1, OPT
CLKØ
CLK1
MDØ
MD1
MD2
MD3
MD4
MD5
MD6
MD7
U4 LS245
AUX RAM
R/W
P2
U3 LS273
U2 LS158
U11 LS14
REGWR
CR1 1N4148
R1 4.7K
C1 4.7µF 16V
W1
W2
W4
W3
CAS1EN
CASØEN
UNLESS OTHERWISE SPECIFIED:
ALL RESISTOR VALUES ARE IN OHMS.
ALL RESISTORS ARE 1/4W, 5%.
ALL CAPACITOR VALUES ARE IN MICROFARADS.
RWAUX
RAS1
RASØ
CAS1
CASØ
RN4, OPT
RN3, OPT
RN6, 1K
RN5, 1K
BANK Ø
U12
U13
U14
U15
U16
U17
U18
U19
R6 56
BANK 1
U20
U21
U22
U23
U24
U25
U26
U27
SPARE
U9 LS00




**ACE 2000 SERIES**
**67 WATT POWER SUPPLY**




# ACE 2000 SERIES KEYBOARD VERSION 1

**Franklin** Computer

# TECH NOTES

SUBJECT: EXTENDED MEMORY USAGE

NO. TN001

There has been some confusion concering the use of the extended ram card in regards to the following:

1. What comes with it?
2. What can it be used for?
3. What software and hardware is it compatible with?

Here are some answers:

1. What comes with it:
   a. A 320K or 512K ram board
   b. Manual insert
   c. Addendum sheet to manual
   d. Utility diskette version 1.0
      NOTE: this disk is in ProDOS format. It DOES NOT contain the ProDOS operating system. It cannot be used or cataloged with FDOS. A ProDOS utility disk is needed to use this disk.
   e. Two ROM chip upgrades
      NOTE: some kits went out without the diagram showing the ROM installation. If this happens, call tech support at 609-488-5454 and instructions will be given over the phone.

2. What can it be used for:
   The utility disk version 1.0 contains files to allow Appleworks version 1.1 and 1.2 to run out of the memory card to greatly increase the operating speed. It will NOT increase the desktop beyond 55K. It also contains a file called "PRAMDISK" which initializes the card as "/ACERAM" in slot 4. This can be used by some ProDOS programs as a disk storage device.

   There are two other FDOS files which use the extended memory card: "RAMTEST" and "FRAMDISK". These are found on the Franklin DOS2 diskettes which have an orange dot attached. If the DOS disk does not contain these two files, do the following:
   a. Make several working copies.
   b. Send the original to tech support with a note explaining the problem. Please list your return address and phone number. You will receive an updated version.

   The FRAMDISK program will initialize the extended ram card as a DOS disk which can be used by some basic programs and files as a storage device.

3. What hardware and software is it compatible with:
The Franklin extended memory card is compatible with Applied Engineering Ramworks card. Applied Engineering's ads state that their ramworks card is Franklin 2000 software compatible. Therefore, software which is written to run on the Franklin extended ram card will run on an Apple IIe if it is equiped with an Applied Engineering card; NOT an Apple extended memory card.

Software which runs using the extended memory card besides Appleworks version 1.1 and 1.2 is being tested and documented. Updates ond procedures for use will be available as testing continues.

**Franklin** Computer

# TECH NOTES

SUBJECT: PRODOS PFS SERIES WITH EXTENDED MEMORY NO. TN002

Using Franklin's extended memory card with PFS ProDos series.

1. Boot ProDOS utility disk.

2. Go to Basic

3. Insert Franklin utility disk version 1.0

4. Type " -PRAMDISK" (return)
   NOTE: this will initialize the extended ram card as "/ACERAM" in slot 4.

5. Insert PFS WRITE, PLAN or FILE disk.

6. Type " -/PFS.WRITE/WRITE.SYSTEM" OR
   " -/PFS.PLAN/PLAN.SYSTEM" OR
   " -/PFS.FILE/FILE.SYSTEM" .

7. If using combination of WRITE, FILE, PLAN on the unidisk type " -/PFS/WRITE.SYSTEM" OR
   " -/PFS/PLAN.SYSTEM" OR
   " -/PFS/FILE.SYSTEM" .

8. When saving files to /ACERAM be sure to change prefix to ACERAM first.

IMPORTANT: Save any files in ACERAM to a floppy disk before exiting the program.

Franklin Computer

# TECH NOTES

SUBJECT: 220V 50HZ CONVERSION OF ACE 2000 NO. TN003

1. On the mainboard, solder a connection across butterfly jumper JP3. This performs the 60 to 50HZ conversion.

2. On the power supply PC board find the wire jumper that is connected to the 115V connection. Change over to 230V.

**Franklin** Computer

# TECH NOTES

SUBJECT: PHANTOM SLOT# 1 NO. TN004

The following information will allow you to disable the built in parallel printer interface card and to phantom slot# 1 to slot#2.

A) Solder a connection across the butterfly jumper JP2 at pin #28 of the Video EPROM (underneath extended RAM card). This disables the on-board printer port control.

B) Pull pins #13 & #14 of U103 and U112 (74LS138's) and reconnect pin #14 of both chips to previous position of pin #13 of both. This connects the DEV SEL and I/O SEL control lines of slot# 1 to slot# 2.

To easily select between using slot# 2 as slot #1 or slot# 2, a header and shorting clip can be used to enable/disable the printer port control firmware, and the two 74LS138's can be socketed.

SUBJECT: APPLE/AMDEK COLOR MONITORS NO. TN005

A compatibility problems exsists between the ACE 2000 and the Apple and Amdek composite color monitors. To generate color on these monitors the following changes can be done to the mainboard of the ACE 2000 CPU.

(A) Correction of VSYNC circuitry which consists of:

1. Cut U65 pin# 10 from the PC board.
2. Jump U65 pin# 10 to U72 pin# 10.
3. Jump U37 pin# 2 to U72 pin# 9.
4. Jump U72 pin# 8 to PC board at U65 pin# 10.

(B) Change capacitor C141 to 33 pF.

(C) Add 10pF capacitor in parallel to C140.

**Franklin** Computer

# TECH NOTES

SUBJECT: POWER SUPPLY WHINE NO. TN006

To eliminate the high pitched whine exhibited by some power supplies used in the ACE 2000 series the following can be done:

On the power supply PC board

1) Change R14 to 820 ohms 1/4W

2) Change capacitor C6 to 8200pF, 1kV

1. The ACE 2000's built in parallel port emulates either the Apple Parallel or the Epson APL interface cards. This means if the program your using asks for what type of interface card you are using then either one of these selections can be used depending on what is available.

2. The built in 80 column card emulates the Apple IIe extended 80 column card. As with above, this selection would be used if your program requests that information.

3. The Open-F and Closed-F are equal to the Open and Closed Apples on an Apple IIe.

4. The 12 function keys are predefined when FDOS 5.0 is loaded from disk and are only valid while using that program. If another program is loaded, the function key assignments are overridden. Information on how to program the function keys to your own specifications is located on pages 6-1 to 6-6 of the ACE 2000 User Reference Manual.

5. Editing in BASIC becomes easy when using the built in Enhanced Video Software. See pages 7-19 to 7-21 of you User Reference Manual.

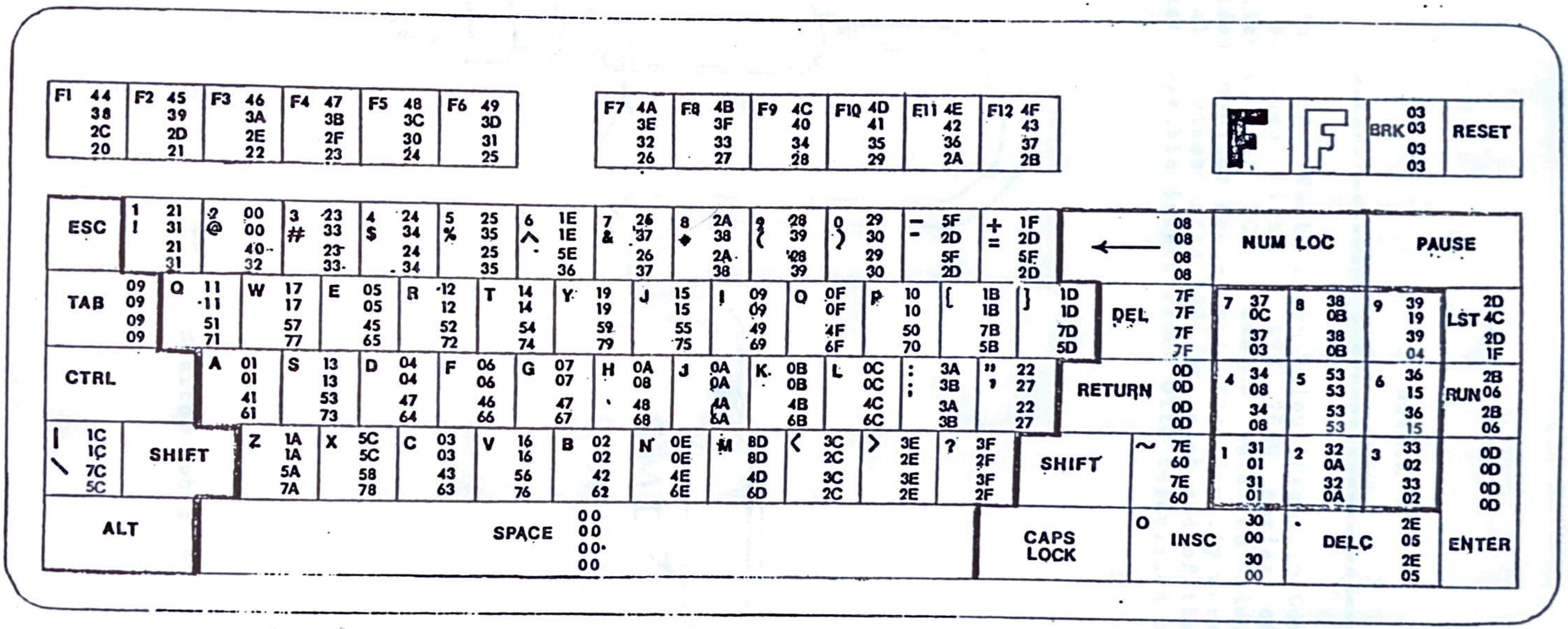
6. To reset the computer, you use the Control Reset sequence. To envoke a warm boot, press Control OPEN-F and then reset while still depressing the Control and Open-F.

7. Extended 80 column boards sold for the Apple IIe are not compatible with the ACE 2000. The cards will not physically fit into your ACE.

8. To envoke the alternate character set (mouse icons),option switch #4 must be turned on.

9. Franklin publishes a Software and Hardware Compatibility list which is updated monthly. If you have any questions concerning either of these two areas consult your dealer.

10. You can use any Apple II+ or IIe compatible disk drive. Consult the Hardware Compatibility list for more information.



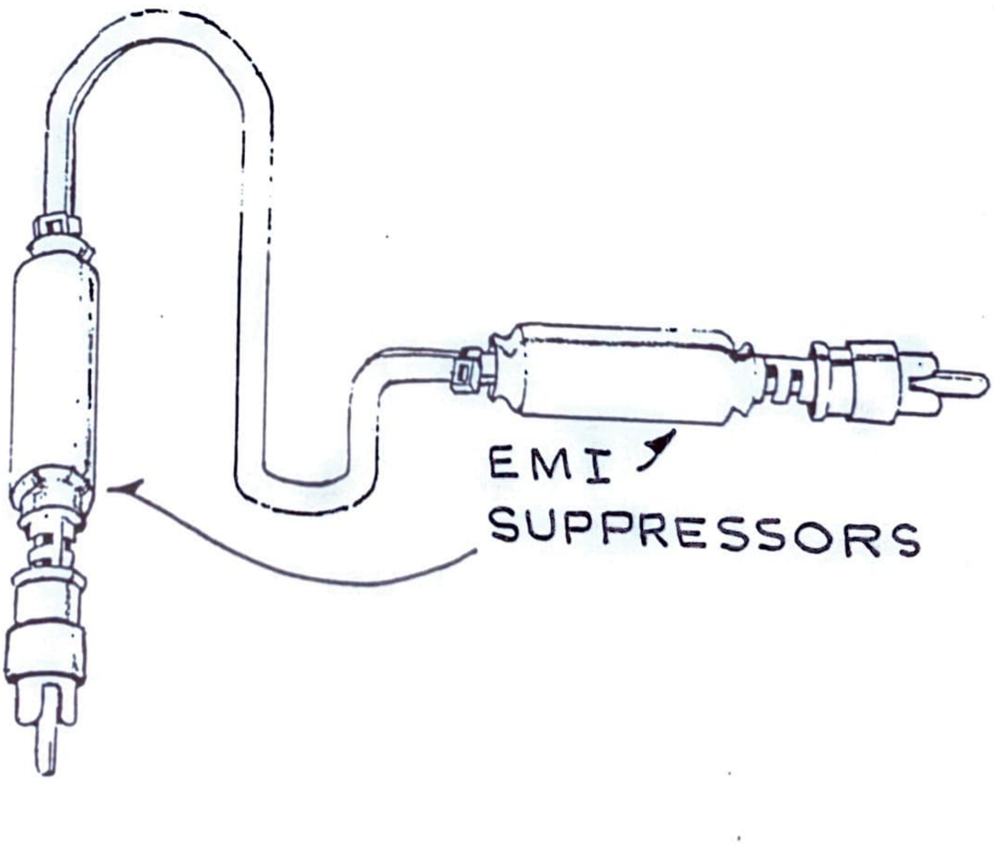

| H | HH | CONTROL-SHIFTED |
|---|---|---|
| | HH | CONTROL |
| | HH | SHIFTED |
| | HH | NORMAL |

* NUM LOCK ON = SHIFT

## NOTICE

---

The Video Cable shown below (Franklin Computer Part #VC200) must be used with an ACE 2000 series computer to assure compliance with FCC Class B emission limits. Alternative shielded monitor cables will functionally operate but lack the special devices attached at each end of the VC200 cable to prevent possible interference with television or radio reception.




VC 200 Monitor Cable

To install the new ROM set that is enclosed, first remove the ROM chips from locations U1, U2, and U3 on the mainboard of your computer.

*Then install the ROM chip marked P1 at location U3, the ROM chip marked P2 at location U2 and the ROM chip marked P3 at location U1.

Please return the unused ROMs to the Technical Services Dept. of Franklin Computer as soon as possible.

Thank you,

Franklin Computer Corporation
Technical Services Department

*Note: When installing the ROMs, make certain that the "notch" on the ROM(s) alines with the "notch" on the socket.