

# **ACORN SYSTEM 5 HANDBOOK**



400,056/1/1st March 1983

© Copyright Acorn Computers Limited 1983

Neither the whole or any part of the information contained in, or the product described in, this manual may be adapted or reproduced in any material form except with the prior written approval of Acorn Computers Limited (Acorn Computers).

The product described in this manual and products for use with it, are subject to continuous development and improvement. All information of a technical nature and particulars of the product and its use (including the information and particulars in this manual) are given by Acorn Computers in good faith. However, it is acknowledged that there may be errors or omissions in this manual. A list of details of any amendments or revisions to this manual can be obtained upon request from Acorn Computers Technical Enquiries. Acorn Computers welcome comments and suggestions relating to the product and this manual.

All correspondence should be addressed to:

Technical Enquiries  
Acorn Computers Limited  
Fulbourn Road  
Cherry Hinton  
Cambridge  
CB1 4JN

All maintenance and service on the product must be carried out by Acorn Computers authorised dealers. Acorn Computers can accept no liability whatsoever for any loss or damage caused by service or maintenance by unauthorised personnel. This manual is intended only to assist the reader in the use of the product, and therefore Acorn Computers shall not be liable for any loss or damage whatsoever arising from the use of any information or particulars in, or any error or omission in, this manual, or any incorrect use of the product.

**WARNING: THE COMPUTER MUST BE EARTHED**

IMPORTANT: The wires in the mains lead for the computer are coloured in accordance with the following code:

GREEN AND YELLOW	EARTH
BLUE	NEUTRAL
BROWN	LIVE

As the colours of the wires may not correspond with the coloured markings identifying the terminals in your plug, proceed as follows:

The wire which is coloured green and yellow must be connected to the terminal in the plug which is marked by the letter E, or by the safety earth symbol  $\text{\AA}$  or coloured green, or green and yellow.

The wire which is coloured blue must be connected to the terminal which is marked with the letter N, or coloured black.

The wire which is coloured brown must be connected to the terminal which is marked with the letter L, or coloured red.

## **HOW TO USE YOUR SYSTEM 5 HANDBOOK**

### **INTRODUCTION**

This handbook describes the Acorn System 5 computer and is divided into two parts: Software and Hardware. These parts are sub-divided into sections on the Disc Operating System, and hardware details of the individual boards.

### **NEW USERS**

The handbook is technically complete but may initially be baffling to new users. New users wishing to

get their System 5 up and running should go straight to Disc Operating System, Part 2, Section 1. This section shows how to switch on the machine and use the Disc Operating System facilities,

### **NOTE**

There are references in the Disc Operating System section to System 5 Basic. Please note that Basic is not supplied as standard with System 5 machines configured as Econet File Servers.

# **SYSTEM 5 HANDBOOK**

## **CONTENTS**

### **PART 1            SOFTWARE**

Section 1        Disc Operating System

### **PART 2            HARDWARE**

Section 1        Rack, PSU and Backplane

Section 2        6502A CPU Board

Section 3        32K DRAM Board

Section 4        80 x 25 VDU Board

Section 5        Floppy Disc Controller Board

Section 6        Acorn Keyboard

Section 7        Econet Board

©1982 Acorn Computers Limited, 12 Fulbourn Road, Cherry Hinton, Cambridge.  
Written and Produced for Acorn Computers Limited by Arch Technical Publications Limited,  
St. Lukes House, 5, Walsworth Road, Hitchin, Herts.

## DISC OPERATING SYSTEM

### DESCRIPTION

### CONTENTS

Paragraph		Page
1.	INTRODUCTION	1
1.1	Disc Handling and Safe Keeping	1
1.2	What is Supplied in the System	1
1.3	The Disc Operating System	2
2.	GETTING STARTED	2
2.1	Switching On	2
2.2	The Keyboard	2
2.2.1	Shift Key	2
2.2.2	Shift Lock Key	2
2.2.3	Caps Lock (^v) Key	3
2.3	Keyboard Entries	3
2.3.1	Character Deletion (DELETE Key)	3
2.3.2	Character Correction	3
2.3.3	Line Deletion (CRTL—X)	3
2.3.4	Entering a Correct Line (RETURN Key)	3
2.4	Loading a Disc	3
2.5	Reading a Catalogue	3
2.6	Drive and Surface Numbering	4
2.6.1	Using Drives	4
2.7	Formatting a New Disc	4
2.8	Loading, Saving and Deleting Files	5
2.8.1	Introduction	5
2.8.2	Loading and File	6
2.8.3	Saving Files	6
2.8.4	Deleting Files	6
2.9	Using BASIC	6
2.9.1	Introduction	6
2.9.2	Loading and Running BASIC	6
2.9.3	Saving a BASIC Program	6
2.9.4	Loading a BASIC Program	6
2.9.5	Returning to DOS and Back	6
2.9.6	BASIC Commands Associated with DOS	7
2.9.7	Control Codes and BASIC	8
2.10	Disc Security	8
2.10.1	Write Protection Notch	9
2.10.2	Software Lock	9
2.10.3	Back-Up Discs	9

©1982 Acorn Computers Limited, 12 Fulbourn Road, Cherry Hinton, Cambridge.  
Written and Produced for Acorn Computers Limited by Arch Technical Publications Limited,  
St. Lukes House, 5, Walsworth Road, Hitchin, Herts.

## CONTENTS (Cont'd)

<b>Paragraph</b>		<b>Page</b>
	2.10.4 Transaction File Organization	9
	2.10.5 Ancestral File System	9
3.	DOS FEATURES	9
3.1	Tracks, Sectors and Bytes	9
3.2	Formats	9
3.2.1	Disc Format	9
3.2.2	Catalogue Format	10
3.3	The Command Line Interpreter	10
3.4	Keyboard Control Functions	11
3.4.1	Start Printer, STX [CTRL-B, 02 (Hex)]	11
3.4.2	End Printer, ETX [CTRL-C, 03 (Hex)]	11
3.4.3	Start Screen, ACK [CTRL-F, 06 (Hex)]	11
3.4.4	Backspace, BS [CTRL-H, 08 (Hex)]	11
3.4.5	Horizontal Tab, HT [CTRL-I, 09 (Hex)]	11
3.4.6	Line Feed, LF [CTRL-J, 0A (Hex)]	11
3.4.7	Vertical Tab, VT [CTRL-K, 0B (Hex)]	11
3.4.8	Form Feed, FF [CTRL-L, 00 (Hex)]	11
3.4.9	Return, CR [CTRL-M, 0D (Hex)]	11
3.4.10	End Screen, NAK [CTRL-U, 15 (Hex)]	11
3.4.11	Home Cursor, RS [CTRL-", 1E (Hex)]	11
3.4.12	Copy, DC1 [CTRL-Q, 11 (Hex)]	11
3.5	Abbreviated Entries	11
3.6	Printer, Screen and Keyboard Streams	11
3.6.1	SPOOL Command	11
3.6.2	EXEC Command	11
3.7	Auto-Start (OPTION and BOOT)	11
3.7.1	OPTION Command	11
3.7.2	BOOT File	12
3.8	EXEC Command	13
3.9	Qualifiers	13
3.10	Spooling	13
3.11	DOS Pointer	13
3.12	Disc Titles	<b>13</b>
3.13	Information on Files	14
3.13.1	INFO Command	14
3.13.2	MON and NOMON Commands	14
3.14	Machine Code Program Execution	14
4.	DOS COMMAND AND ERROR MESSAGES	14
4.1	DOS Commands	14
4.2	Error Messages	17
4.3	File Specifications	17

## CONTENTS (Cont'd)

<b>Paragraph</b>	<b>Page</b>
4.3.1     File Name	17
4.3.2     Qualifier	17
4.3.3     Load Address	18
4.3.4     Execution Address	18
4.3.5     File Length	18
4.3.6     Physical Location (Position on Disc)	18
5.       UTI LTI ES	18
5.1     INFALL	19
5.2     COPY	19
5.3     COMPACT	19
5.4     COPYF	19
5.5     DUTY	19
5.6     FORM40, FORM80	19
5.6.1    Soft Sectoring	19
6.       SEQUENTIAL FILE HANDLING	20
6.1     Pointer Operation	20
7.       ASSEMBLY CODE ACCESS TO DOS	21
7.1     Introduction	21
7.2     OS Subroutines	22
7.2.1    OSFIND	22
7.2.2    OSSH UT	22
7.2.3    OSCLI	22
7.2.4    OSWRCH	22
7.2.5    OSCRLF	22
7.2.6    OSASC I	22
7.2.7    OSECHO	22
7.2.8    OSRDCH	22
7.2.9    OSLOAD	22
7.2.10 OSSAVE	23
7.2.11 OSSTAR	23
7.2.12 OSBGET	23
7.2.13 OSBPUT	23
7.2.14 OSRDAR	23
7.3     Random File Operation	23
7.3.1    Introduction	23
7.3.2    OSRDAR and OSSTAR	24
7.3.3    OSBPUT	24
7.3.4    OSBGET	24
7.3.5    OSFIND	24
7.4     File Buffers and Control Blocks	24

## **TABLES**

<b>Table No.</b>	<b>Page</b>
1. Operating System Control Codes	8
2. Summary of Routines and Calls	21

## **FIGURES**

<b>Figure No.</b>	<b>Page</b>
1. The Mini-Floppy Disc	1
2. Driver Numbering	4
3. Command Line Interpreter	10
4. Full File Specification	17
5. Sector Format	20

## 1. INTRODUCTION

Your System 5 is supplied with a Disc Operating System (DOS) built in to its main memory (ROM). The DOS provides you with:

- Control of your screen and printer.
- Screen editing facilities.
- The means to save and load programs, data and text to and from disc.
- File management facilities.

All of the above facilities are available to you directly from the keyboard, or you can "embed" them within your programs.

### 1.1 DISC HANDLING AND SAFE KEEPING

Programs data and text that you have stored on disc may represent hours, weeks or months of effort. By careful and methodical handling and the use of simple security procedures your programs, data and text stored on disc will not be lost. The following security procedures should be observed:

- DO NOT REMOVE the circular magnetic disc from it's square black protective jacket.
- DO NOT TOUCH the exposed magnetic surfaces.
- AVOID DUST; keep the discs in their protective jackets and store in a storage box when not within the drive.
- DO NOT BEND, drop them, or rest heavy objects on them.
- KEEP THEM AWAY from strong magnetic fields such as those generated by televisions, monitors, tape recorders, transformers, telephones and calculators.
- AVOID excessive heat, moisture and direct sunlight.
- USE ONLY FELT TIPPED PENS to write on the labels and don't press too hard.
- INSERT DISC CAREFULLY; if a disc rotates noisily, open drive door and adjust the disc position.

You will want to protect some discs against accidental overwriting. Each disc has a Write Protection Notch, see Figure 1. When the Write Protection Notch is covered with a self adhesive tab (supplied with each box of discs) you cannot overwrite any programs, data or text stored on that disc.

You may have individual programs, data or text stored on disc that you want to write protect, while

writing to other areas on the same disc. Your DOS can do this for you, but we will describe how this is done later.

Finally, it is recommended that you keep copies of discs containing important programs, data or text.

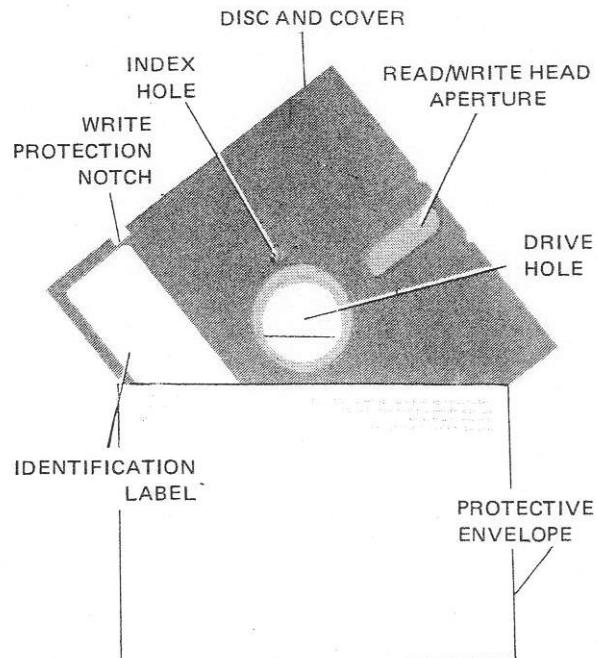


Figure 1. The Mini-Floppy Disc

### 1.2 WHAT IS SUPPLIED IN THE SYSTEM 5

Your System 5 will have been supplied according to your requirements, but will include a minimum of the following items:

#### HARDWARE

- Rack, PSU and Backplane
- 6502A Processor Board
- 32K Dynamic RAM Board
- 80 x 25 VDU Interface Board or Teletext VDU Interface Board
- Floppy Disc Controller Board
- One Floppy Disc Drive

#### SOFTWARE

- Resident DOS
- A DOS utilities disc (see Section 5)

A list of available hardware and software options is given in part 1 of System 5 System Description.

Note that a different version of DOS is required for different VDUs and different manufacturer's disc drives.

### 1.3 THE DISC OPERATING SYSTEM

Although the DOS is primarily concerned with disc access, handling and management, it also contains the System 5 Operating System (OS). This controls your screen and printer and provides you with screen editing facilities. Handling of the DOS from the keyboard is described in Sections 2, 3 and 4. Handling of the DOS from programs is described in Section 7.

The DOS recognises your screen and printer control, and screen editing instructions by control key entries from your keyboard (CTRL-A, CTRL-B, etc.).

The DOS recognises your disc access, handling and management instructions by DOS command entries from your keyboard or from within programs.

The DOS also recognises machine code program names. When the file name is entered from the keyboard the DOS loads the named machine code program. The DOS has an auto-start facility so that programs will be loaded and immediately run. The DOS will also execute a series of commands via an Exec file using the Exec command.

## 2. GETTING STARTED

Several terms will be unfamiliar to you — but don't worry. At first, for the sake of simplicity, we will gloss over some terms that we use but they will be explained later.

Programs, data or character strings (text) are handled by the DOS as Files. When you store a File on disc you must give the program a File Name which, when you subsequently call the program, data or text, the DOS will recognise.

The DOS will also recognise the following commands:

CAT	Catalogue disc
DELETE	Delete file
DIR	Direct Catalogue to memory
DRIVE	Select drive
EXEC	Display or print as text and execute
GO	Load and run machine code program
INFO	Display or print file specification
LOAD	Load file to memory
LOCK	Write Protect a file
MON	Start display or print of file specification messages
NOMON	Stop display or print of file specification messages
OPTION	Enable/disable auto-start facilities
RUN	Run program, file
SAVE	Save file to disc from memory

SET	Set file qualifier
SHUT	Shuts all open files
SPOOL	Opens file for copy
TITLE	Give title to disc
UNLOCK	Remove a file Write Protection
USE	Qualify which group of files that may be accessed

File names can be used directly as Load and Run commands.

You should refer to Section 4 to determine the precise keystrokes for each DOS command, and always follow a command with a RETURN.

### 2.1 SWITCHING ON

Switch your System on at the wall socket and at the red rocker switch at the lower left hand end of the computer front panel. If you have an Acorn keyboard, the right hand lamp at the front edge of the keyboard will be lit indicating that System power is On.

To enter DOS, press the DELETE key, followed by the BREAK key. If you have a non-Acorn keyboard, press and hold the DELETE key and press the BREAK key. You should now be given the screen message:

**Acorn Dos**

\*

—

The "Acorn Dos" indicates that the DOS is active. The \* indicates that the System is waiting for a keyboard entry, and the flashing cursor indicates the print position of the next character to be input.

### 2.2 THE KEYBOARD

Your keyboard is similar to a typewriter keyboard, though some keys have special functions.

#### 2.2.1 Shift Key

Each character key will enter either an upper case or lower case character. Pressing the Shift key while pressing a character key enters the upper case character.

#### 2.2.2 Shift Lock Key

Pressing the Shift Lock key locks the keyboard into upper case entry. Pressing and releasing the Shift key releases the keyboard to lower case entry. If you have an Acorn keyboard, the centre lamp on the right hand side of the keyboard will light when in shift lock.

### **2.2.3 Caps Lock (^v) Key**

BASIC programs and DOS commands are recognised only when entered as upper case letters and (lower case) numbers. This involves frequent use of the Shift and Shift Lock keys. The Caps Lock mode on the Acorn keyboard allows you to enter DOS commands and BASIC programs with minimum use of the Shift or Shift Lock keys.

If you have an Acorn keyboard, it has a push-on push-off key marked ^v. This key sets the keyboard into the Teletype (CAPS LOCK) mode. The CAPS LOCK mode allows the user to enter capital letters and numbers without having to use the shift key.

Press the ^v, key. The left hand lamp at the front edge of the keyboard should now be lit, confirming that the keyboard is in the CAPS LOCK mode. If the lamp is out press the ^v- key again.

To return to normal operation from CAPS LOCK, press the ^v button.

## **2.3 KEYBOARD ENTRIES**

DOS commands (and BASIC programs) are entered using upper case letters and numbers. Until the RETURN key is pressed corrections can easily be made.

### **2.3.1 Character Deletion (DELETE Key)**

Incorrect characters can be removed by pressing the DELETE key, in which case the character to the left of the cursor is erased and the cursor moved back one space. The cursor may be positioned by pressing and holding the CRTL key while the A key (cursor left) or S key (cursor right) is pressed.

### **2.3.2 Character Correction**

Characters can be corrected by positioning the cursor (see 2.3.1 above) and pressing the character key to be overwritten.

### **2.3.3 Line Deletion (CRTL-X)**

A complete line can be deleted by pressing and holding the CTRL key while the X key is pressed.

### **2.3.4 Entering A Correct Line (RETURN Key)**

When a command or line is correct it is finally entered by pressing the RETURN key.

## **2.4 LOADING A DISC**

Until we deal with formatting in para 2.7 you should use the utilities disc (supplied with your System 5).

Carefully remove your disc from its white protective envelope.

Do not try to remove the disc from its black cover. Open the disc retaining door on the front of the disc drive. If you have two disc drives use the left hand one (Drive 0). Carefully insert the disc, complete with black protective cover, label outermost, Write Protect Notch uppermost, into the slot in the disc drive. When the disc is fully inserted, close the disc retaining door. The disc is now ready to be read.

## **2.5 READING A CATALOGUE**

In para 2.4 the utilities disc was loaded into Drive 0.

You can find out what files are on a disc by reading its catalogue.

To read a Catalogue in Drive 0 key in:

**C | A | T | R | E | T | U | R | N**

The disc drive lamp which comes on for a short time while the System is accessing the disc confirms that the disc drive is working.

You should now have the Catalogue of the Drive 0 disc on the screen. The Catalogue will depend on the utilities you have been supplied with, but will be something like:

Utilities Disc drive 0 qual option 0  
: #COMPACT #COPY  
#COPYF #DUTY  
#FORM80 #INFALL  
\*—

File Names are listed alphabetically in two columns.

The screen tells us that:

- The disc TITLE is "Utilities Disc"
- The disc is located in Drive 0
- Qualifier space is currently being used (we will deal with this later).
- The disc is set to OPTION 0 (we will deal with this later).
- No characters to the left of the ":" de-limiters indicate that qualifier space has been allocated to all of the files (again, we will deal with this later).
- Six Files exist with the file names, INFALL, COPY, COPYF, COMPACT, DUTY and FORM80.

- All six Files are LOCKed (Write Protected by software — more of this later) indicated by the # symbol.

At the end of the CAT listing the DOS\* and flashing cursor indicates that the DOS is waiting for a new entry.

—

The DOS recognises the disc surfaces available to you as drive numbers. The drive numbers available depend on the hardware environment, i.e. the number and type (single or double sided) of disc drives that are fitted. The DOS allocates a drive number to each available surface. So for a system configuration with two double sided disc drives, the left hand disc position has two surfaces number 0 and 2, while the right hand disc position has two surfaces numbered 1 and 3. Figure 2(a) shows the numbering for this configuration. If we have two single sided disc drives fitted, the left hand one is numbered 0 and the right hand one is numbered 1. Figure 2(b) shows the numbering for this configuration. If only one double sided disc drive is fitted, the two surfaces are numbered 0 and 2, see Figure 2(c). One single sided disc drive provides one surface numbered 0, see Figure 2(d).

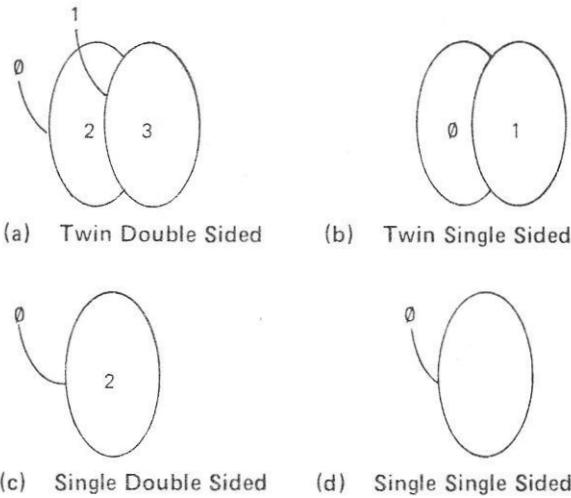


Figure 2. Drive Numbering

Mistakes in numbering can be made when specifying drive numbers in DOS commands. For example, in a twin drive double sided system, COPYing from Drive 0 to 2 would result in transcribing a file from one surface to the other on the same disc instead of, as intended, from disc to disc. It is a good idea to label the drives for the particular configuration so that mistakes of this sort do not happen.

In para 2.5 when we read the catalogue, we did not specify from which drive we were to read the Catalogue. Whenever the drive is not specified the current drive (set to 0 after Break) Catalogue is read. We can specify from which drive DOS is to read the Catalogue by the entries CAT0, CAT1, CAT2 or CAT3. When the DOS assumes a parameter through omission it is said that the DOS "defaults" to an assumed parameter. Some other DOS commands default through omission and will be described when they occur.

If the drive number is specified in the CAT command, it sets the current drive to that specified (the DRIVE command can also be used to set the current drive).

### 2.6.1 Using Drives

When we first entered DOS the current drive was set to Drive 0. This allows file access to Drive 0. To access a drive other than 0 the DRIVE command is used with the format:

D	R	I	V	E	1	RETURN
---	---	---	---	---	---	--------

where "1" is the new current drive.

An alternative way of changing the current drive is with the CAT command. In para 2.5 we read the catalogue of the current drive by not specifying the drive. If we, however, use the CAT command specifying a drive other than the current drive, the current drive is changed to that specified. For example, if the current drive is 0 and we enter:

C	A	T	RETURN
---	---	---	--------

the catalogue of Drive 1 is displayed and the current drive is changed to Drive 1.

Where a file is loaded and run using the file name alone, that file can be accessed from Drive 0 only.

## 2.7 FORMATTING A NEW DISC

Before you can use a new disc it must first be formatted. It is recommended that whenever a new box of discs is opened the discs are all immediately formatted.

When you read the catalogue of the utilities disc (in para 2.5), the listing included either FORM40 or FORM80. The version will depend on whether 40 or 80 track disc drives are fitted to your system. The procedure is the same for both FORM40 and FORM80.

To format a new disc, insert the utilities disc into drive 0, check using the CAT command that the file FORM40 or FORM80 is present. Type in the file name given in the Catalogue:

(either FORM40 or FORM 80 )

followed by RETURN .

DOS now loads the formatting program into memory at the location given in the File Specification and runs the program.

The message:

Do you really want to format drive 0 ?  
will appear on the screen.

The utilities disc in drive 0 must now be removed and replaced by the new disc to be formatted.

Once the disc to be formatted is in position, the YES response is keyed in.

The formatting starts immediately and the message:

Formatting drive 0

is displayed. As each track is formatted and verified, the track number is displayed as follows:

```
00 01 02 03 04 05 06 07 08 09  
0A 0B 0C 0D 0E 0F 10 11 12 13  
14 15 16 17 18 19 IA 1B IC ID  
IE IF 20 21 22 23 24 25 26 27  
disc formatted
```

If formatting is unsuccessful because the disc is Write Protected the message:

Crunch  
will be displayed,

If formatting is unsuccessful for other reasons the message:

YY Failure at XX

will be displayed, which means "disc error YY at sector XX on the disc".

By replacing the new formatted disc with an unformatted disc, and typing GO RETURN a series of discs can be formatted.

To format side 2 type DRIVE2 then GO.

With twin drive installations (either single or double sided), the utilities disc can be left in drive 0 and the disc to be formatted may be mounted in drive 1 (single sided) or drive 1/3 (double sided). In this case, before typing F 0 R M 4 0 or F 0 R M 8 0 , type

in DRIVE (*drive number*), selecting the drive to be formatted. Whenever a drive other than 0 is selected, any subsequent file execute call will search for that file in drive 0 instead of the drive selected, unless RUN is used.

## 2.8 LOADING, SAVING AND DELETING FILES

### 2.8.1 Introduction

The LOAD and SAVE DOS commands are similar in function to the BASIC's LOAD and SAVE command but they are NOT the same. The DOS LOAD and SAVE commands are sometimes known as "Star Load" and "Star Save" to avoid confusion. The DOS DELETE command is unrelated to the Delete key function.

Before you start loading and saving files you need to know what the terms File Name, Start Address, End Address and Execution Address mean.

#### FILE NAME

The File Name:

- Must be from 0 to 7 characters long.
- Must not include spaces unless File Name is enclosed in quotes (" ") .
- Must not contain an odd number of quotes ("").
- Trailing spaces are ignored.

You must allocate a File Name to anything you save. You must also give the File Name when you load it.

#### START ADDRESS

The Start Address is the (start) address in memory to which the file is loaded, or, the (start) address in memory from which the file is saved. We will see in para 2.8.2 that we can either load a file to the same start address from which it was originally saved, or we can load the file to a new start address.

Programs, text or data entered from the keyboard normally have a start address of #2800. (# means hexadecimal).

#### END ADDRESS

When a file is saved the DOS needs to know the last memory address to be saved. This last memory address saved is known as the End Address.

#### EXECUTION ADDRESS

The Execution Address is the address in memory at which the program is entered when it is Run. If you fail to give an Execution Address, the program will begin to Run at the Start Address.

### **2.8.2 Loading a File**

In para 2.7 we used a file called FORM40 (or FORM80) simply by entering its name, followed by a RETURN. This is the method generally used to Load and Run a machine code program from Drive 0.

If you have BASIC and you want to Load and Run it, you should mount the BASIC disc in Drive 0 (the left hand drive, if you have a twin drive System 5), and key in BASIC followed by RETURN from your keyboard. Different versions of BASIC may have different names, so it's a good idea to check the name of your version by first using the CAT command (as described in para 2.5).

To Load and Run a machine code program from a drive other than Drive 0 or a BASIC Program listing, separate Load and Run DOS commands must be used after setting the current drive (see para 2.6.1).

LOAD (File Name) followed by RETURN, loads the named file from disc to the memory. The file is loaded to the same memory location that it was originally saved from.

If you wish to load a file to a memory address other than that from which it was originally saved, you must specify the start address, i.e. LOAD (File Name) (Start Address).

### **2.8.3 Saving Files**

Files are saved using the SAVE (File Name) (Start Address) (End Address + 1) DOS command.

Some program files may require to be executed from some point within the program. In this case the command SAVE (File Name) (Start Address) (End Address + 1) (Execution Address) is used.

### **2.8.4 Deleting Files**

Files may be deleted using the DOS command DELETE (File Name).

When files are deleted, gaps occur, often leading to inefficient disc usage. It is recommended that periodic housekeeping is carried out using the COMPACT utility (described later in Section 5).

## **2.9 USING BASIC**

### **2.9.1 Introduction**

We have already mentioned that files can consist of programs, text or data (or a combination of all three). When we speak of programs we mean machine code programs. Your BASIC programs are handled by the

DOS as text files. This may seem strange at first, but in Section 3 we will see that the DOS recognises only DOS commands, and not BASIC commands. BASIC, however, is a machine code program and can be loaded and run from Drive 0 by simply entering the file name BASIC (in the same way that we loaded and ran the FORM40 utility in para 2.7).

### **2.9.2 Loading and Running BASIC**

To load and run BASIC you must first insert your BASIC disc in Drive 0 (as described in para 2.4). Check the BASIC File Name using the CAT0 command (as described in para 2.5). You must now load and run BASIC by keying:

**BASIC RETURN**

Your screen will now show the > BASIC prompt (with the flashing cursor) indicating that you can now key in your BASIC Program. Your BASIC Manual tells you how to program in BASIC.

### **2.9.3 Saving a BASIC Program**

To save your BASIC program you use the BASIC SAVE command. This is slightly different than the DOS SAVE command in that the file name must be enclosed in quotes and no space is required after SAVE.

Your BASIC disc should be Write Protected, so remove it and replace it with a formatted disc in Drive 0. If you are going to call your BASIC program "FRED", key in:

**SAVE "FRED" RETURN**

If you have a twin drive system, you can save on to Drive 1 by keying \*DRIVE 1 before saving as above.

### **2.9.4 Loading a BASIC Program**

A BASIC program can only be loaded after BASIC itself has been loaded and run (see para 2.9.2).

To load a BASIC program insert your program disc in Drive 0 and key in:

**LOAD "FRED" RETURN**

You can check that it has been loaded by keying in:

**LIST RETURN**

and you can run your BASIC program by keying in:

**RUN RETURN**

### **2.9.5 Returning to DOS and Back**

When you are in BASIC you can return to DOS by pressing the BREAK key.

You can subsequently return to BASIC by keying in:

GO	C2B2	RETURN
----	------	--------

Unfortunately, when you re-enter BASIC in this way, you lose the BASIC program that you are using.

BASIC however, has a method of using all the DOS commands rather like subroutines. This method simply uses the DOS command preceded by a "\*" in a BASIC program line, e.g.

40 \*CAT

will when running BASIC (at line 40), print the catalogue of the current drive. if you wish to use DOS while entering a BASIC program you omit the BASIC line number.

### 2.9.6 BASIC Commands Associated with DOS

BASIC has a number of commands which allow you to create and use files for data, text and programs. We will look at these commands and how they are used later.

#### FOUT

This command creates and opens a file for output and gives you a File Handle which you subsequently use for output. The command is used in the form:

30 D=FOUT"Fred"

FRED is the File Name, "D" is the File Handle. You will subsequently use "D" in your BASIC program to identify that file in output operations.

If the file "FRED" already exists, DOS will use that file. If the file "FRED" does not exist, DOS will create that file.

#### FIN

This function opens a file for input and update. The function is used in the form:

70 D=FIN"Fred"

FRED is the File Name. "D" is the File Handle. You will subsequently use "D" to identify that file in input operations.

#### PUT

This statement sends a four byte word to a file. The statement is used in the form:

40 PUT D,563

where "D" is the File Handle (see FOUT above) and 563 is a number to be output to the file.

#### BPUT

This statement sends a single byte to a file. The statement is used in the form:

30 BPUT D,23

"D" is the File Handle (previously obtained using the FOUT function). 23 in this case is the value of the byte to be output (any number between 0 and 255 can be represented in a single byte).

#### SPUT

This statement sends a string to a file. The statement is used in the form:

50 SPUT D,A

"D" is the File Handle (previously obtained using the FOUT function). "A" is the pointer to the start of the string.

The use of "A" rather than "\$A" requires a few words of explanation at this point. Acorn BASIC requires the allocation of a space for a string anywhere safe (remember we said that BASIC programs are really text). This is done using a BASIC system variable called TOP which points to the next free location above your BASIC program text. If we wish to input a string \$A with a length of up to (say) twelve characters we could allocate space by the BASIC program line:

10 A=TOP; B=T0P+ 13

This allows twelve characters plus an "end of string marker".

The BASIC line 50 shown above will send out characters to the file (pointed to by "D"), starting at A and finishing at the "end of string marker". This means that within the limits set by BASIC line 10, whatever length strings we send out to disc, no file space is wasted by gaps when strings are different lengths.

#### GET

This function reads a four byte word from a file and reads its value. The function takes the form:

80 J=GET D

OR.

80 PRINT"THE FIRST NO.FROM FRED IS"GET L

where "D" is the File Handle. J in this program line points to the value read from the file. Before the GET function is used in a program the file must be opened for input and "D" determined by the FIN function in the form:

70 D=FIN"Fred"

The GET function is the usual way to input numbers from a data file.

### BGET

This function returns a single byte from a file. The function takes the form:

```
80 K=BGET D  
OR  
80 PRINT"THE FIRST BYTE FROM FRED IS"BGET D
```

where "D" is the File Handle previously determined by the FIN function (see BASIC line 70 above).

### SGET

This statement reads a string from a file. The statement is used in the form:

```
90 SGET D,A
```

where "D" is the File Handle (obtained by FIN). "A" is the pointer to the beginning of SA (initialized as described in SPUT).

### EXT

This function returns the extent (length) of a file in bytes. The function is used in the form:

```
60 R=EXT D
```

where "D" is the File Handle (obtained by FIN or FOUT) and R is the pointer to the number of bytes in that file.

### PTR

As we output to or input from a file DOS keeps a pointer to our position within that file. We can read this position using the PTR function for any file that is open for output, or open for input (FOUT and FIN respectively). The function is used in the form:

```
75 PRINT PTR D
```

where "D" is the File Handle (given by FIN). PTR gives our position in bytes.

In random file operation this DOS pointer can be modified through BASIC using the form:

```
40 PTR D=PTR D+23
```

where "D" is the File Handle obtained by the preceding FIN or FOUT statement. This BASIC program line will move the DOS pointer on by 23 bytes. The next file access will then be 23 bytes further on.

The pointer returns to the start of the file when the file is shut and subsequently re-opened (by FIN or FOUT)

### SHUT

This statement closes input or output files. We can shut individual files by using the form:

```
100 SHUT D
```

where "D" is the pointer to the file we wish to close.

We can shut all files by using the form:

```
100 SHUT 0
```

### 2.9.7 Control Codes and BASIC

The Operating System (OS) within DOS has a number of printer and screen control codes as shown in Table 1. All of the OS control codes except COPY and the Cursor controls can be used within BASIC Programs.

OSCLI Name	Keyboard Entry	(Hex) Value	Function
STX	CTRL-B	02	Start Printer
ETX	CTRL-C	03	End Printer
ACK	CTRL-F	06	Start Screen
NAK	CTRL-U	15	End Screen
BS	CTRL-H	08	Backspace
HT	CTRL-I	09	Horizontal Tab
LF	CTRL-J	0A	Line Feed
VT	CTRL-K	0B	Vertical Tab
FF	CTRL-L	0C	Form Feed
CR	CTRL-M	0D	Return
RS	CTRL-^	1E	Home Cursor
DEL	DELETE	7F	Delete and Backspace
ETB	CTRL-W	17	Cursor Up
SOH	CTRL-A	01	Cursor Left
DC3	CTRL-S	13	Cursor Right
SUB	CTRL-Z	1A	Cursor Down
DC1	CTRL-Q	11	Copy

**Table 1. Operating System Control Codes**

All of the Control Codes can be entered from the keyboard by pressing and holding the CTRL key while the appropriate character key is pressed.

To include a Control Code within a BASIC program the PRINT \$ command is used. The usual form is:

```
40 PRINT #$C  
OR  
40 PRINT $12
```

where #C is the hexadecimal number corresponding to Form Feed (Clear Screen and Home Cursor) and 12 is its decimal equivalent. These BASIC lines perform the Formfeed (clear the screen, moving the cursor to the top left hand position).

### 2.10 DISC SECURITY

Programs, text and data stored on disc may represent weeks, or even months of work, and care must be taken not to lose them. DOS has special facilities which prevent accidental overwriting or deletion.

### **2.10.1 Write Protection Notch**

Each disc has a square Write Protection Notch on one edge, see Figure 1. This may be covered to prevent accidental erasure of the disc. Every box of discs is supplied with a number of self-adhesive tabs which, when wrapped around the notch will prevent the disc being overwritten. All utilities should be write protected.

### **2.10.2 Software Lock**

DOS provides you with a command which will prevent accidental erasure of individual files.

The command:

```
LOCK FRED RETURN
```

will write protect the file "FRED". All utilities and programs should be locked.

You can unlock an individual file by the unlock command:

```
UNLOCK FRED RETURN
```

which will remove the write protection of the file "FRED".

### **2.10.3 Back-Up Discs**

It is important to keep back-up discs of all utilities, important programs and data files.

### **2.10.4 Transaction File Organization**

Data Processing Centres using large main frame computers usually use a disc or tape management system called Transaction File Organization. This can sometimes be used in System 5 applications to add a measure of security to large data files which are subject to frequent change. A typical application is in a stock control system where items are issued or received continually, but where re-ordering occurs only periodically (say once a day).

The principle is that the large database file (stock, re-order codes, suppliers, minimum stock levels etc) is kept locked during normal transaction periods. All issues and receipts of stock are handled by a simple (and fast) program which creates and adds to a file (on a separate disc) containing only changes or transactions.

When issues and receipts of stock cease (say lunch-time), another program is entered which uses the transaction file to update the large database file and, subsequently, to provide re-ordering printouts etc.

The large stock database file (which may take days or weeks of physical stocktaking to re-create) is thus protected for much of the time.

A secondary advantage of Transaction File Organization is that the lengthy program searches and sorts associated with re-ordering are avoided allowing the user immediate access for input of issue and receipt data.

### **2.10.5 Ancestral File System**

A disadvantage of the Transaction File Organization outlined in para 2.10.4 is that if a system failure occurs while the stock database is being updated, the database could be corrupted or lost. The Ancestral File System is a method of minimising the effect of such a catastrophe.

In principle, the Ancestral File System uses three database and three transaction files, all on separate discs, labelled Grandfather, Father and Son. At each update period, the Son database and transaction files are used to create a new database file on the (old) Grandfather disc, Thus the (old) Grandfather becomes the (new) Son, the (old) Son becomes the (new) Father and the (old) Father becomes the (new) Grandfather. The transaction and database files (now Fathers) used to create the Son database file are kept together in a safe place and, should a catastrophe occur, can be used to re-create the Son database file.

## **3. DOS FEATURES**

### **3.1 TRACKS, SECTORS AND BYTES**

Information is written onto the disc in concentric circles, called tracks. Your System 5 will be equipped with either 40 or 80 track Floppy Disc Drives. Each track is divided into 10 sectors and each sector is divided into 256 bytes. Each byte corresponds to one character. Floppy Disc Drives may be single or double sided types.

### **3.2 FORMATS**

#### **3.2.1 Disc Format**

Because the user allocates file names as character strings, and the computer recognises tracks and sectors by numeric representation of the file's physical location, an area on the disc is allocated for an index or Catalogue seen by typing CAT. This area is always located in Sectors 0 and 1 on disc. It supplies the DOS with a catalogue to locate a file quickly. The user may allocate a TITLE to a particular disc surface and the Title is stored in the Catalogue.

Each File Name in the Catalogue has its position on disc stored. Other file parameters, some allocated by the user, and some determined by the DOS, are

entered against each file in the Catalogue. Among the user defined parameters stored against each file in the Catalogue are:

- LOCK, (#), which is a software write protect facility.
- Qualifier, which is associated with the SET (*qualifier*) DOS command. This allows, in association with the USE (*qualifier*) DOS command, division of a TITLE (disc surface) into a number of groups of files.
- Start Address. This allows the DOS command LOAD (*file name*) to specify loading to memory location specified by the file. This parameter is originated by the SAVE (*file name*) (*start address*) (*end address*) DOS command.
- Execution Address. This parameter originated by the SAVE (*file name*) (*start address*) (*execution address*) DOS command is mainly applicable to program files used in association with the OPTION 2 facility and for command files, e.g. BASIC.

The length parameter is worked out by the DOS itself when the DOS SAVE command is executed. This parameter gives the number of bytes contained within the file.

### 3.2.2 Catalogue Format

When the disc is started the Catalogue is copied into memory, and this Catalogue is assumed valid while the disc is rotating. Care must be taken with some types of drive (which spin when a disc is inserted) that the disc has stopped before using it. Reading successive files therefore requires as little head movement as possible. Changes made to the Catalogue cause it to be written out to the disc. The Catalogue and file buffers are stored in Random Access Memory at locations 2000 to 27FF (hex) and this RAM must be present in the system.

### 3.3 THE COMMAND LINE INTERPRETER

It is useful, though not essential, to understand how the OS and DOS work. The two main inputs to the OS are from the keyboard and the disc. The input strings (of characters and codes) from the keyboard are put, first of all, into a buffer store (part of the computer's memory). Whenever a string ends with a Carriage Return code a special OS program called the Command Line interpreter is automatically run. The Command Line interpreter or OSCLI for short, does exactly what its name says — it looks for and

interprets a command line. But how does the OSCLI know that a string in the buffer store is a command?

The OSCLI, when it looks at the string in the buffer store, compares it with a Command List (or table) in the computer memory (DOS ROM). When comparison is found, an entry against that command code vectors the program execution to the subroutine in OS (or DOS) that executes the command. If the string is not found in the DOS ROM, the DOS looks it up on disc (and will load and run the file so named). If the string is still not found an error message occurs. Figure 3 illustrates the basic principles of operation of a Command Line Interpreter.

When the system is powered up or reset (using the BREAK key), the vectors are loaded from ROM into RAM at start address 0200 (hex).

All keyboard entries to the OS are buffered at RAM address 0100 (hex). BASIC, however, may put them elsewhere. The Operating System subroutine OSCLI, on recognising and identifying a valid OS command in the buffer, vectors the program execution to the appropriate subroutines to implement the command keyed in. The commands must be terminated by a RETURN (CR) code, 0D (hex).

The System 5 Disc Operating System (DOS) extends the Command List to provide the disc and file handling facilities that are available. Many of the utility programs available use the Command Line Interpreter in their operation.

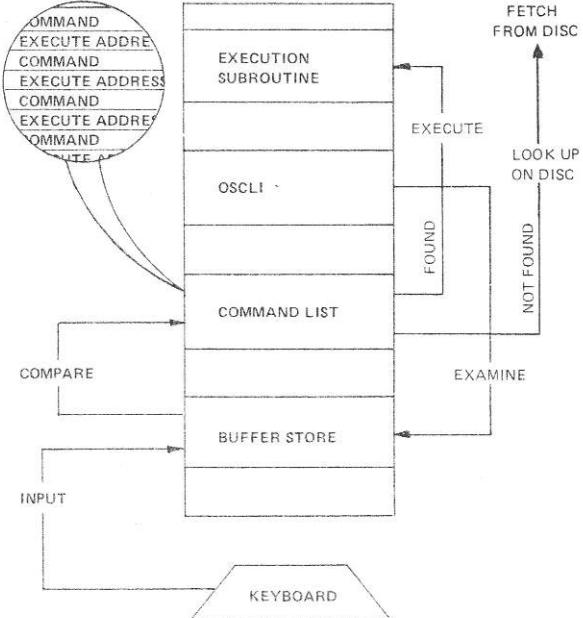


Figure 3. Command Line Interpreter

### 3.4 KEYBOARD CONTROL FUNCTIONS

The following facilities are available from the keyboard or from BASIC programs (see Table 1).

#### 3.4.1 Start Printer, STX [CTRL-B, 02 (hex)]

This code, which is not sent to the printer, starts the printer output stream. All further output is sent to the printer as well as the screen until receipt of an ETX (End Printer) code.

#### 3.4.2 End Printer, ETX [CTRL-C, 03 (hex)]

This code ends the printer output stream.

#### 3.4.3 Start Screen, ACK [CTRL-F, 06 (hex)]

This code starts the output stream to the screen.

#### 3.4.4 Backspace, BS [CTRL-H, 08 (hex)]

This code moves the cursor back one position.

#### 3.4.5 Horizontal Tab, HT [CTRL-I, 09 (hex)]

This code moves the cursor forward by one position.

#### 3.4.6 Line Feed, LF [CTRL-J, 0A (hex)]

This code moves the cursor down one line.

#### 3.4.7 Vertical Tab, VT [CTRL-K, 08 (hex)]

This code moves the cursor up one position.

#### 3.4.8 Form Feed, FE [CTRL-L, 0C (hex)]

This code clears the screen and moves the cursor to the top left hand corner of the screen.

#### 3.4.9 Return, CR [CTRL-M, 0D (hex)]

This code moves the cursor to the start of the current line.

#### 3.4.10 End Screen, NAK [CTRL-U, 15 (hex)]

This code ends the output stream, to the screen. The only code recognised in this condition is ACK.

#### 3.4.11 Home Cursor, RS [CTRL-^, 1E(hex)]

This code moves the cursor to the top left hand corner of the screen.

#### 3.4.12 Copy, DC1 [CTRL-Q, 11 (hex)]

This code copies the characters above the cursor to a new line.

### 3.5 ABBREVIATED ENTRIES

DOS recognises abbreviated entries for many DOS commands. Section 4 lists, against each command, the abbreviated entry, if any.

### 3.6 PRINTER, SCREEN AND KEYBOARD STREAMS

DOS sees the keyboard as an input stream and the screen and printer as output streams.

#### 3.6.1 SPOOL Command

The SPOOL command opens a file for output and copies all output stream onto that file. The file is closed by the SHUT command. The SPOOL command format is:

**SPOOL FRED RETURN**

where "FRED" is the user allocated file name to the SPOOL file.

#### 3.6.2 EXEC Command

The EXEC command reads the named file and displays the file as characters. If the stream contains valid DOS commands they are executed. If BASIC is entered via a DOS command in the EXEC file, subsequent BASIC commands are executed. This allows a series of BASIC programs, and/or data files to be handled consecutively.

The format of the EXEC command is:

**EXEC FRED RETURN**

where "FRED" is the name of the file to be executed.

### 3.7 AUTO-START (OPTION AND BOOT)

In para 2.5 when we read the catalogue of a disc, one of the parameters associated with the disc was Option. The DOS Option feature allows a series of files to be loaded and/or run sequentially by simply pressing and holding the SPACE bar while pressing the BREAK key. This feature is useful where BASIC programs are being used, especially where the BASIC programs require access to data or text files. The Boot file (para 3.7.2) must exist on Drive 0 to be used on auto-start,

#### 3.7.1 OPTION Command

The Option of a disc in the current drive is set using the OPTION command format:

**OPTION 3 RETURN**

where "3" is the Option to which the disc in the current drive is set.

The options available are as follows:

- Option 0 : Do not do anything
- Option 1 : Load the file named BOOT
- Option 2 : Run the file named BOOT
- Option 3 : Execute the file named BOOT

### 3.7.2 BOOT File

The Boot file provides the user with a facility which may be used in many applications. Essentially it allows you, in association with Option, to execute a series of DOS commands by simply pressing and holding the SPACE bar while pressing the BREAK key. If BASIC is present on disc, it can be loaded and run via Boot, and a series of BASIC Programs loaded and run via BASIC commands within the Boot file.

#### CREATING A BOOT FILE

To create a Boot file you will need (after loading and running BASIC) to enter and run a simple BASIC program as follows:

```
19 A=FOUT "BOOT"
20 INPUT $TOP
30 IF $TOP = "END" GOTO 60
40 SPUT A $TOP
50 GOTO 20
60 SHUT A
70 END
SAVE "INBOOT"
RUN
```

Line 10 opens a file named "BOOT" for output and allocates its pointer "A". Line 20 allows you to input a command which is stored immediately above your BASIC program "INBOOT". Line 30 allows you to terminate Boot file creation by typing "END". Line 40 puts the command (from \$TOP) into the Boot file (pointed to by "A"). Line 60 closes the Boot file. The "SAVE" line saves the Boot creation program with the file name "INBOOT" so you can use it again.

You will now have the "?" prompt for command input on screen. A series of commands can now be entered from the keyboard, each terminated by a RETURN. When your Boot file is complete typing "END" will close the Boot file.

The Boot file:

```
? LOAD BASIC
? RUN BASIC
? LOAD FRED
? RUN FRED
```

will not work but the following Boot file will:

```
? LOAD BASIC
? RUN BASIC
? LOAD "FRED"
? RUN
```

Once we run BASIC only commands in BASIC syntax will be executed.

As an example of the use of BOOT, we will write two simple programs called FRED and HARRY and create a Boot file to run them successively via the BREAK key. First of all press and hold the DELETE key while the BREAK key is pressed to enter DOS and then enter:

```
BASIC
10 PRINT ' "THIS SHOWS THAT FRED HAS RUN" '
20 END
SAVE "FRED"
10 PRINT " "THIS SHOWS THAT HARRY HAS RUN" "
20 END
SAVE "HARRY"
```

You have now loaded and run BASIC and created the files FRED and HARRY. If you have not already done so enter the INBOOT program and load and run it to create the Boot file by entering:

```
LOAD BASIC
RUN BASIC
LOAD "FRED"
RUN
LOAD "HARRY"
RUN
PRINT " "BOOT FILE HAS NOW BEEN EXECUTED" "
END
```

By pressing and holding the DELETE key while the BREAK key is pressed you will re-enter DOS, and we must now set the Option to 3 by keying:

OPTION 3

Now, by pressing and holding the SPACE bar while the BREAK key is pressed at any time, the Boot file will be executed, and with our example, the screen will display:

```
* LOAD BASIC
BASIC C000 C2B2 01000 003
* RUN BASIC
BASIC C000 C2B2 01000 003
>LOAD "FRED"
FRED 2800 C2B2 00032 002
>RUN
THIS SHOWS THAT FRED AS RUN
>LOAD "HARRY"
HARRY 2800 C2B2 00033 013
>RUN
THIS SHOWS THAT HARRY HAS RUN
>PRINT " "BOOT FILE HAS NOW BEEN EXECUTED" "
BOOT FILE HAS NOW BEEN EXECUTED
```

Should you wish to Break without executing Boot, you must press and hold the DELETE key while the BREAK key is pressed.

To disable the auto-start, re-enter DOS and enter OPTION 0. Alternatively, if you are in BASIC, enter \*OPTION 0.

### 3.8 EXEC COMMAND

Any file created in the same way as Boot (para 3.7.2) can be executed by the DOS command EXEC. If you still have your Boot file on disc set Option to 0 (see para 3.7.1) and enter EXEC BOOT (from DOS) or (from BASIC) enter \*EXEC BOOT. The Boot file will now run with the same screen display as in para 3.7.2. The EXEC format is given in para 3.6.2.

### 3.9 QUALIFIERS

In para 2.5 the CAT command display showed that qualifier "space" was currently being used, and that all files had a space to the left of the colon (:) indicating that all files were allocated the qualifier "space".

The qualifier facility is a useful aid to file handling. It is implemented by means of the SET and USE DOS commands.

The SET command sets the current qualifier to any character until either the BREAK key is pressed, or the SET command is again used. Pressing the BREAK key sets the current qualifier to "space". Files are saved with the current qualifier (except when a file is saved immediately after a USE command). Note that no space is allowed between SET and the character, e.g. SETA is acceptable while SET A is not.

The USE command allows the next file operation to use other than the current qualifier. After the next file operation the qualifier reverts to the current qualifier.

The most common use of qualifiers is to separate files according to their content, e.g. qualifier B for BASIC programs and qualifier D for data files.

There may be two (or more) separate files with the same file name providing they are in separate qualifiers. Hence with the Assembler you can have a source file called, say, UADE01 in qualifier "space" which you will assemble into an object file with the same name providing it has a different qualifier.

### 3.10 SPOOLING

The SPOOL command will open a file for copying. To demonstrate you should enter:

SPOOL SPFILE

or if you are in BASIC:

\*SPOOL SPFILE

For demonstration purposes, to put something in the file named SPFILE you could EXEC BOOT, see para 3.8 (or \*EXEC BOOT if you are in BASIC).

To disable the SPOOL facility (closing the file) you must enter SHUT (or \*SHUT SPFILE from BASIC).

You will need a small BASIC program to read your SPFILE, so re-enter BASIC and key in this program:

```
10 D=FIN "SPFILE"
20 C=EXTD
30a IF C=PTRD GOTO b
40 A=BGET D
50 IF A=10 PRINT $13
60 PRINT $A
70 GOTO a
80b SHUT D
90 END
```

Line 20 uses the EXT command to determine the length of the file. Line 30 checks to see if the end of file has been reached before reading in a character from the file. Line 50 tidies up the screen format, adding a Return to each Line Feed. Saving this file with the file name "SPREAD" will allow you to use it again. By changing the file name in Line 10 you can read any file.

### 3.11 DOS POINTER

When accessing files DOS maintains a pointer to the current access position within the file. The pointer can be read or changed by the BASIC PTR function. We used PTR in the SPREAD program (Line 30) in para 3.10 to exit from a loop when the end of file was reached. The pointer is set to "0" whenever a file is opened (FIN, FOUT' SPOOL, etc). We will look at the DOS Pointer again in para 6.1.

### 3.12 DISC TITLES

We can give the disc in the current drive a title using the command TITLE followed by a space and the disc name we are allocating (in BASIC the command is preceded by a \*). Up to 13 characters may be used for a title.

### 3.13 INFORMATION ON FILES

#### 3.13.1 INFO Command

We can read information on any file in the current drive using the INFO command (followed by a space and the file name). The information displayed is in the following form:

current qualifier	: lock	file name	load address	execution address	length in bytes	start sector
:	#BASIC	C000 C2B2	01000 002			
:	#LISP	2800 2800	02000 012			
s:	ZOMBY	3000 C2B2	00312 032			

### 3.13.2 MON and NOMON Commands

The MON command turns on a message system which displays a file's information at each file access.

The NOMON command turns off the message system which was originally turned on by the MON command.

### 3.14 MACHINE CODE PROGRAM EXECUTION

To execute a machine code program the GO command may be used (followed by a space and the execution address). We used this command in para 2.9.5 to reenter BASIC at C2B2 from DOS.

If the execution address is omitted the last known execution address will be used. Note that the current execution address is destroyed by CAT, and INFO does not set the execution address.

If a machine code program is on Drive 0 you can use the file name directly (as we called FORM40 and BASIC).

## 4. DOS CO#MMANDS AND ERROR MESSAGES

In the following commands, error and other messages, text, data and programs are said to be "displayed". They can, in fact be displayed on screen or printed or both, subject to the preceding Operating System control codes entered, see Table 1 in para 2.9.7.

File names and disc titles must be enclosed in quotes unless no spaces exist in file name. Only even numbers of quotes (or non at all) are allowed in file names or disc titles. Up to 13 characters are allowed in disc titles, and up to 7 characters are allowed in file names.

Legal drive numbers are dependent on the disc drive configuration, see para 2.6, but illegal drive numbers will simply cause the system to wait for the drive which is not there. Abbreviated command formats are shown following the full command formats.

## 4.1 DOS COMMANDS

**C A T [ ] (drive number)**

**. [ ] (drive number)**

This command causes the Catalogue of drive (*drive number*), (or the current drive if (*drive number*) is omitted) to appear on the screen.

A typical Catalogue will look like:

\*CAT0

Basic disc v1 drive 0 qual s opt 0

: #BASIC #LISP

s: ZOMBY

The title of the disc is Basic disc v1 ; we are currently using drive 0 and qualifier s. The disc option is 0 (no auto-start features). Two files have been saved in qualifier "space", both of which have been locked to prevent careless deletion. One file has been saved in qualifier "s" and this has been left unlocked. The Catalogue is sorted alphabetically by qualifier and file name when it is output. The (*drive number*) can be omitted, or it must be 0, 1, 2 or 3. If outside the range 0 to 3 then the message:

Drive ?

will be displayed.

If the (*drive number*) is specified, the default drive for subsequent commands is changed to that specified.

**D E L E T E [ ] (file name)**

**D E . [ ] (file name)**

This command deletes the (*file name*) in the current qualifier from the current disc's Catalogue. If the entire disc is Write Protected a

Disc prot

message is displayed. If the file is not found a

File ?

message is displayed. If the file is Locked a

File prot

message is displayed.

**D I R [ ] (drive number)**

**D . [ ] (drive number)**

This command causes the Catalogue of the drive (*drive number*) to be loaded into memory at hex start

address 2000. The command is often used to wait until completion of the previous operation. The drive number can be omitted, or it must be in the range 0 to 3. If outside the range then the message:

Drive ?

will be displayed.

**D R I V E**  (drive number)  
**D R .**  (drive number)

This command sets the current drive to (drive number) where (drive number) is either in the range 0 to 3, or is omitted (for compatibility with CAT or DIR). If (drive number) is outside the range the message:

Drive ?

will be displayed. If (drive number) is entered as a multi-character then the message:

Syntax ?

will be displayed. Drive 0 is set on BREAK.

**E X E C**  (file name)  
**E .**  (file name)

This command reads and displays as characters the bytes from (file name) in the current qualifier on the current disc. If (file name) is not found a

File ?

message is displayed. After all the bytes have been read, (file name) is automatically closed.

**G O**  (execution address)

This command causes the machine code subroutine at (execution address) to be entered. If (execution address) is not given, the last known execution address is used. Note that the current execution address is destroyed by CAT, and INFO does not set the execution address.

**I N F O**  (file name)  
**I .**  (file name)

This command produces information about the (file name) in the current qualifier on the current disc. If the file is not found a

File ?

message is displayed. The information displayed is in the following form:

current qualifier	file lock	file name	load address	execution address	length in bytes	start sector
:						
:	#BASIC	C000	C2B2	01000	002	
:	#LISP	2800	2800	02000	012	
s:	ZOMBY	3000	C2B2	00312	032	

**L O A D**  (file name)  (start address)  
**L .**  (file name)  (start address)

This command loads the file (file name) on the disc in the current drive from the current qualifier into the memory at (start address). The (start address) may be omitted when the file's own (start address) is used. If the file is not found a

File ?

message is displayed.

**L O C K**  (file name)  
**L O .**  (file name)

This command Write Protects an individual file.

The command locks the (file name) in the current qualifier on the current disc. If entire disc is protected a

Disc prot

message is displayed. If the file is not found a

File ?

message is displayed.

**M O N**   
**M .**

This command turns on a message system which displays a file's information at every file access.

**N O M O N**   
**N .**

This command turns off the message system which was originally turned on by the MON command.

**O P T I O N**  (option)  
**O .**  (option)

This command sets the option of the disc in the current drive to the number (option). If the entire disc is protected, a

Disc prot

message is produced. The option enables auto-start use of the file BOOT in qualifier "space" on drive 0 when the system is reset by the BREAK key. The auto-start may be totally defeated by pressing the DELETE key while the BREAK key is pressed, and is subsequently enabled by pressing "space" while the BREAK key is pressed. The possible modes are:

- Option 0 : do not do anything
- Option 1 : load the file BOOT
- Option 2 : run the file BOOT
- Option 3 : exec the file BOOT

In option 0, the system will not mind if BOOT is not present, in the other modes, a

File ?

message will be displayed if BOOT does not exist when the BREAK key is pressed.

**R|U|N** (file name 1)  (file name 2)  
**R|.**  (file name 1)  (file name 2)  
 (file name 1)      (file name 2)

The (file name 2) is optional.

This command loads the file (file name 1) on the disc in the current drive from the current qualifier into memory at the (start address) of the file. The optional (file name 2) is turned back into the original string form and stored in memory at hex start address 0140, terminated by a RETURN (CR character).

**S|A|V|E** (fn)  (sa)  (ea+1)  (xa)  
 where (fn) = (file name)  
 (sa) = (start address)  
 (ea+1) = (end address + 1)  
 (xa) = (execution address)

The (execution address) is optional. When omitted it defaults to the start address.

This command saves the block of memory between (start address) and (end address + 1) to the file (file name) in the (current qualifier) of the Catalogue. If the entire disc is Write Protected a

Disc prot

message is displayed. If the (file name) is Locked a

File prot

message is displayed. If (file name) exists and is not Locked, or the disc is not Write Protected, the (file name) is deleted. Starting at the extreme outside edge of the disc (track 0), a gap is searched for which

is large enough to contain the block to be saved. If a large enough block cannot be found a

Disc full

message is displayed. If there are already 31 files on the disc Catalogue a

Full

message is displayed.

**S|E|T** (current qualifier)

This command sets the current qualifier to (current qualifier), where (current qualifier) can be any character. All following file access will use only the (current qualifier) portion of the Catalogue. Qualifier "space" is set on keying BREAK.

**S|H|U|T**

This command closes all files for input or output.

**S|P|O|O|L** (file name)

This command opens file and copies all characters going out onto it. The Spool facility is stopped by the SHUT command.

**T|I|T|L|E** (disc name)  
**T|.** (disc name)

This command sets the title of the disc in the current drive to the first 13 characters in (disc name), filling with spaces if there are fewer than 13 characters. It is possible to include Form Feed (CTRL-L) at the start of a Title, so that Catalogues appear at the top of the screen. If the entire disc is Write Protected, a

Disc prot

message is displayed.

**U|N|L|O|C|K** (file name)  
**U|.** (file name)

This command unlocks the (file name) in the current qualifier on the current disc. If entire disc is protected a

Disc prot

message is displayed. If the file is not found a

File ?

message is displayed.

**U|S|E** (qualifier)

This command allows the next file operation to use the (qualifier) portion of the Catalogue. After the file

operation is complete, the previous (*current qualifier*) will be made current again. If an error occurs in the file operation, the qualifier does not immediately revert so that the operation can be repeated. To force reversion after an error, use a MON or NOMON command. Two successive USE commands will result in the loss of the original (*current qualifier*).

#### 4.2 ERROR MESSAGES

- |                      |   |   |
|----------------------|---|---|
| <b>Crunch</b>        | : | An attempt has been made to format a Write Protected disc.  |
| <b>Disc Full</b>     | : | A block large enough to hold the file cannot be found. Periodic housekeeping with the COMPACT utility will minimise the occurrence of this error. |
| <b>Disc Prot</b>     | : | A Write operation has been attempted to a Write Protected disc.   |
| <b>Drive ?</b>       | : | Drive number is incorrectly (or not) specified.   |
| <b>File ?</b>        | : | File name cannot be found.  |
| <b>File Prot</b>     | : | A Write operation has been attempted to a Write Protected file.   |
| <b>Full</b>          | : | 31 files already exist on the current disc.   |
| <b>Syntax ?</b>      | : | Command is recognized but Syntax error exists.  |
| <b>Disc error 08</b> | : | During a disc Read operation a clock bit was missing.   |
| <b>Disc error 0A</b> | : | During a disc transfer the processor did not respond fast enough (probably due to a faulty Floppy Disc Interface Board)                           |
| <b>Disc error 0C</b> | : | The CRC (cyclic redundancy check) derived from the ID data read back, differed from that which was originally loaded to the disc.                 |
| <b>Disc error 0E</b> | : | The CRC derived from the data read back on a disc read differed from that loaded to the disc.   |
| <b>Disc error 10</b> | : | During a transfer the disc stopped rotating (probably due to a badly inserted disc).  |

**Disc error 14** : The Floppy Disc Controller Interface Board failed to find Track 0. This error usually results in attempting to access an unformatted disc.

**Disc error 18** : The Floppy Disc Controller Interface Board failed to find the required sector. This error usually results from either using an unformatted disc, or the disc being corrupted by magnetic fields etc.

#### 4.3 FILE SPECIFICATIONS

The full File Specification is shown in Figure 4.

##### 4.3.1 File Name

The File Name:

- Must be from 0 to 7 characters long.
- Must not include spaces unless File Name is enclosed in quotes (" ") .
- Must not contain an odd number of quotes ("").
- Trailing spaces are ignored.

The File Name is specified by the user when a file is created. Files are created by:

- The DOS SAVE command.
- Calling the OSFIND routine via assembler (see para 7.1 and 7.3) to perform an open-for-output.
- Calling the OSFIND routine via the BASIC FOUT function (see para 6.1), e.g.

X = FOUT "FRED" in 4K BASIC

The File Name can only be changed by execution of the DOS command LOAD (*old file name*) followed by the SAVE (*new file name*) command.

##### 4.3.2 Qualifier

The Qualifier:

- Must be any single character (or space).

All files are saved with a particular qualifier. The qualifier serves to divide the disc Catalogue into sections. The most common use of the qualifier is to

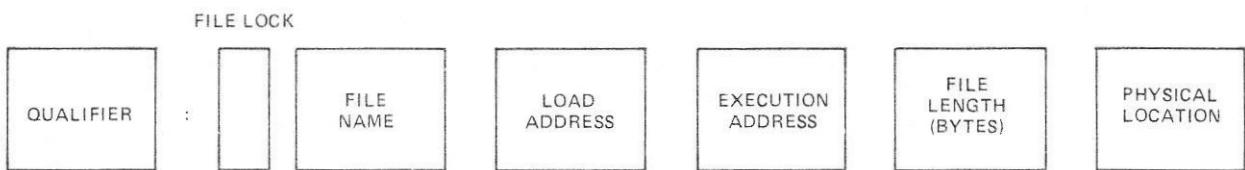


Figure 4. Full File Specification

separate the files according to their content, e.g. qualifier B for BASIC programs and qualifier D for data files.

There may be two (or more) separate files with the same File Name providing they are in separate qualifiers. Hence with the assembler the user may have a source file called UADE01 in qualifier "space" which will be assembled to give an object file with the same name providing it has a different qualifier.

All file operations use the currently selected qualifier. After a reset, the qualifier is set to "space". The DOS provides two commands to change the current qualifier:

- SET(qualifier)* sets the qualifier to *(qualifier)* until another command is used to change it.
- USE(qualifier)* sets the qualifier to *(qualifier)* for the next file operation only, after which it reverts to its previous value.

Note that there must be no space between the command and the *(qualifier)*.

#### 4.3.3 Load Address

The Load Address specifies the address in memory to which the file is loaded when the DOS command LOAD *(file name)* is executed. The Load Address is a four digit hexadecimal number.

The Load Address is defined when the DOS command SAVE *(file name)* *(load address)* is executed.

When the DOS command LOAD *(file name)* *(load address)* is executed, the *(load address)* in the File Specification defaults to the *(load address)* specified in the LOAD command, but the *(load address)* in the File Specification is not changed. Note that files created by OSFIND have a load address of 0000.

#### 4.3.4 Execution Address

The Execution Address specifies the address at which execution of a program in a File will commence when the DOS command RUN *(file name)* is executed. The Execution Address is a four digit hexadecimal number.

The Execution Address is specified when the DOS commands SAVE *(file name)* *(start address)* *(end address + 1)* *(execution address)* is executed. If the *execution address* Field is omitted in the SAVE DOS command, the Execution Address in the File Specification defaults to the *(start address)*.

#### 4.3.5 File Length

The File Length specifies the number of bytes in the file and is a five digit hexadecimal number.

The File Length is calculated by the DOS when the command SAVE *(file name)* *(start address)* *(end address + 1)* is executed.

#### 4.3.6 Physical Location (Position on Disc)

The Physical Location is a three digit hexadecimal number specifying the physical location of the start of the file on disc, i.e. consecutive (through all tracks) sector number. The Physical Location is allocated by the DOS when the file is created.

The DOS allocates a position on disc when the SAVE DOS command is executed.

### 5 UTILITIES

A Utility Mini-Floppy Disc will be supplied with your System. The programs may change with different system variants, so your Utility Disc may not be compatible with other systems.

If a disc with the utilities on it is in drive 0 typing just the utility name will cause the program to be loaded and run, thus the utilities appear to operate as additional DOS commands. The utilities disc may have a BOOT file which will cause a description of the disc programs to appear on the screen at system start up.

In a single drive system the disc with the utilities on will need to be removed and the disc to be operated upon inserted, the programs contain a waiting state for this operation where appropriate. Dual drive, single sided disc systems will usually have the utilities on a disc in drive 0 and the user disc in drive 1. Dual drive double sided disc systems will usually have the utilities on a disc in drive 0 and the user disc in drive 1/3. Select drive 1/3 using the CAT1 (or 3) or DRIVE1 (or 3) as appropriate before running the utility. The original utility disc supplied with a system is best kept Write Protected so that it cannot be accidentally formatted or copied on to.

## **5.1 INFALL**

INFALL uses the INFO command in the DOS to give load address, run address and disc sector information on all the files on a disc.

## **5.2 COPY**

COPY is for use on dual drive systems, it is loaded on entering COPY (*source drive*) (*destination drive*) and the system then stops. Pressing the space bar will cause a complete copy to occur.

## **5.3 COMPACT**

After saving and then deleting files on a disc unused sectors will appear where a file was deleted and no files of the same length have since been saved. Copies one file after the other into RAM and then re-saves them with no gaps between them. Waits for the space bar to be pressed.

"SAVE"ed files use the first available space on disc which is large enough to accommodate the file. Files created use the space occupied by a previous file of the same name (if it existed). If the spare space on disc becomes split up into small parts, attempting to create a large file will cause a "no room" error. To collect all the free space together, the COMPACT utility should be used.

## **5.4 COPYF**

COPYF copies particular files from one drive to another. The source and destination drives are specified as in COPY.

## **5.5 DUTY**

DUTY copies and compacts a disc from one drive to another.

## **5.6 FORM40, FORM80**

FORM (formats) initializes new discs with the track and sector format. The disc requiring formatting should be inserted and checked with the CAT command. A description of Formatting procedure is given in para 2.7.

### **5.6.1 Soft Sectoring**

#### **TRACKS, SECTORS AND BYTES**

Information is written onto the disc in concentric circles, called tracks. Your System 5 will be equipped with either 40 or 80 track Floppy Disc Drives. Each

track is divided into 10 sectors and each sector is divided into 256 bytes. Each byte corresponds to one character. Floppy Disc Drives may be single or double sided types.

## **DISC FORMAT**

Because the user allocates file names as character strings, and the computer recognises tracks and sectors by numeric representation of the file's physical location, an area on the disc is allocated for an index or Catalogue seen by typing CAT. This area is always located in Sectors 0 and 1 on disc. It supplies the DOS with a catalogue to locate a file quickly. The user may allocate a TITLE to a particular disc surface and the Title is stored in the Catalogue.

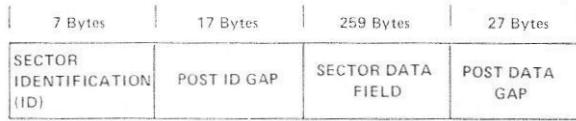
Each File Name in the Catalogue has its position on disc stored. Other file parameters, some allocated by the user, and some determined by the DOS, are entered against each file in the Catalogue. Among the user defined parameters stored against each file in the Catalogue are:

- LOCK (#), which is a software write protect facility.
- Qualifier, which is associated with the SET (*quainter*) DOS command. This allows, in association with the USER (*qualifier*) DOS command, division of a TITLE (disc surface) into a number of groups of files.
- Start Address. This allows the DOS command LOAD (*file name*) to specify loading to memory location specified by the file. This parameter is originated by the SAVE (*file name*) (*start address*) (*end address*) DOS command.
- Execution Address. This parameter, originated by the SAVE (*file name*) (*start address*) (*end address +1*) (*execution address*) DOS command is mainly applicable to program files used in association with the OPTION 2 facility, and for command files, e.g. BASIC.

The length parameter is worked out by the DOS itself when the DOS SAVE command is executed. This parameter gives the number of bytes contained within the file.

The DOS uses these above mentioned file parameters in its' various operations. Because the Catalogue has a fixed field format, the number of files on each disc surface is limited to 31. Similarly, File Names are a maximum of 7 characters and Titles are truncated to 13 characters. We will later see how these parameters can be displayed by the DOS commands INFO and MON.

To allow the computer to access the various tracks and sectors on the disc, after determining the relationship between the File Name and the position on disc, each track must be divided into sectors, identified, and the beginning and end of the sector marked. Figure 5 shows the format of each sector.



**Figure 5. Sector format.**

Before a disc is used for the first time it must be formatted. This is done with a utility called FORMat. If your System 5 is provided with 40 track disc drives you will have been supplied with the FORM40 version, but if you have 80 track disc drives you will have the FORM80 version. Use of these utilities is described in para 2.7.

The FORM80 and FORM40 utilities structure the Catalogue portion of the disc (Sectors 0 and 1), and records the sector IDs in all sectors. In subsequent operations, the DOS provides random File access by looking up the File Name in the Catalogue to determine the physical location and it can then immediately access that particular file without having to serially search until the file is found.

## 6. SEQUENTIAL FILE HANDLING

In para 3.11 we described the DOS file pointer. The pointer is used extensively within user programs for handling sequential files.

### 6.1 POINTER OPERATION

Whenever a file is opened for input or output DOS creates a pointer to the current access position (in bytes) to the file, and initializes this pointer to 0. The BASIC line

20 D= OUT "NED"

will open the file "NED" and initialize the pointer for "NED" to 0 which could be read by the BASIC program line

90 PRINT PTRD

Add the program line

100 END and RUN

and the pointer will be displayed on your screen. if

we add a few program lines to output 4 bytes at a time we can show the pointer at each file access.

```
30 INPUT "ENTER" B
40 PUT D,B
RUN
```

In response to the "?" key in any number (or character) and we will have the pointer value 4 displayed. This is because the BASIC PUT statement outputs four bytes at a time. If we add one more line we can output more numbers and see the pointer moving through the file.

95 GOTO 30 and RUN

Each time you enter a number (and RETURN) you will see the pointer increase by 4. Try keying instead of a single number or character, a number between 0 and 33, 554, 419 (we need four bytes to represent this magnitude number). The PUT statement is the most economic way of storing numbers on disc.

If we change line 40 to

40 BPUT B and RUN

we will see the pointer increment by 1 for each entry. Using BPUT is the most economic way of storing single characters.

If we now modify the program we can see how the file pointer behaves with strings, so we will change lines 30 and 40, and add lines 10, 15, 35 and 99:

```
10 A= TOP
15 B=A+ 51
30 INPUT "ENTER" $A
35 IF $A=%" GOTO 99
40 SPUTD, A
99 SHUTD
```

We have now tidied up our program, providing an exit at Line 35 and shutting the "NED" file at Line 99. Note that Lines 10 and 15 define the maximum length of any string to 50 characters.

If we now run our program and enter a four character string we will see that the pointer moves from 0 to 5. This is because DOS inserts an "end of string marker". This allows DOS, when retrieving strings from the file using SGET to establish how many characters to read for each string. So the pointer moves on by the number of characters +1 for each string access to file. Enter a few more strings into your "NED" file (so we can read it later) before exiting using "%". Save this program -- you may want to use it later,

We will now enter a program for reading strings from "NED" sequentially, printing the File pointer after each string:

```

10 A=TOP
20 B=A+51
30 C=FIN"NED"
40 D=EXTC
50 IF D<=PTRC GOTO a
60 SGETC, A
70 PRINT '$A'
80 PRINT ' "THE POINTER IS" PTRC
90 INPUT $B
100 GOTO 50
110a PRINT ' "END OF FILE" '
120 SHUTC
130 END

```

When this program is run the first string is printed together with the pointer. Pressing RETURN will sequence through the file, printing the string and pointer at each access. When the end of file is reached a message is displayed and we return to BASIC.

By now you should be able to write a short program to read files using the GET (for 4 bytes at a time) and the BGET (one byte at a time) commands.

## 7. ASSEMBLY CODE ACCESS TO DOS

### 7.1 INTRODUCTION

The DOS file handling routines can be used with assembly language. These routines are entered via Jump Indirect instructions stored between FF00 and FFFF (hex). These Jump Indirect instructions, resident in ROM, refer to the vectors stored in RAM between 0200 and 023F. For example the routine to open or shut a file, OSFIND, is entered at FFCE which indirectly calls via FNDVEC at 0218. 0218 and 0219 contain the address of OSFIND (F97A).

The file handling routines, addresses, calls and call addresses and entry points are summarized in Table 2.

On NMI, any operating system interrupts are serviced, otherwise:

```

PHA
IMP (NMIVEC)
is executed.

```

On IRQ or BREAK (BRK):

```

STA 0OFF
PLA

```

```

PHA
AND #10
BNE BRK
LDA 00FF
PHA
JMP (IRQVEC)
LDA 00FF
PLP
PHP
JMP (BRKVEC)

```

is executed. On power up or restart, the OS initializes the vectors from 0200 to 021A to its own handling routines.

Acorn Operating systems use memory in Zero Page starting from location \$00AC up to location \$00AD for their own purposes. Location \$00FE contains the code of an ASCII character which is filtered from the output to the printer, this is initialized to \$0A a line feed. Location \$00FF is used for IRQ/BRK service as above.

ROUTINE		CALLED BY		DESCRIPTION
Address	Name	Address	Call	
		0200	NMIVEC	NMI Routine Entry
		0202	BRKVEC	BREAK Routine Entry
		0204	IRQVEC	IRQ Routine Entry
FFF7	OSCLI	0206	COMVEC	Command Line Interpreter
FFF4	OSWRCH	0208	WRCVEC	Write Character Subroutine
FFED	OSCRLF	—	—	Output CR, LF Subroutine
FFE9	OSASCI	—	—	Read character and check for CR Subroutine
FFE6	OSECHO	—	—	Echo character and check for CR Subroutine
FFE3	OSRDCH	020A	RDCVEC	Read Character Subroutine
FFE0	OSLOAD	020C	LODVEC	Load Program Subroutine
FFDD	OSSAVE	020E	SAVVEC	Save Program Subroutine
FFDA	OSRDAR	0210	RDRVEC	Read Arguments Subroutine
FFD7	OSSTAR	0212	STRVEC	Set Arguments Subroutine
FFD4	OSBGET	0214	BGTVEC	Read Byte From Random File
FFD1	OSBPUT	0216	BPTVEC	Write Byte to Random File
FFCE	OSFIND	0218	FNDVEC	Find Random File
FFCB	OSSHUT	021A	SHTVEC	Shut Random File

Table 2. Summary of Routines and Calls

## 7.2 OS SUBROUTINES

The OS routines are described below:

### 7.2.1 OSFIND

This routine returns, in the A Register, a file handle for the file whose name is pointed to by:



points to a  
string of characters  
terminated by 0D

The handle is zero if the file does not exist otherwise it is a byte of value 1 to 255. If the carry flag is set on input the file must already exist and is opened for reading and updating. If the carry flag is clear on input the file need not exist, and will be used for output. The sequential pointer is set to zero.

### 7.2.2 OSSHUT

This routine closes the file whose file handle is in the Y Register. This involves writing out any buffers that contain data that has been changed, and updating the main disc Catalogue to show the length of the file. A zero file handle in the Y Register will cause all sequential files to be closed. After a file has been closed, the same file handle may be eventually used for a different file.

### 7.2.3 OSCLI

This routine interprets a string of characters held at 0100, terminated by a Carriage Return (0D), as an operating system command. All processor registers are used and detected errors are met with a BRK.

### 7.2.4 OSWRCH

This routine sends the byte in A Register down the output channel. This channel output is usually treated as ASCII data and special action may be taken on ASCII control characters. No processor registers are destroyed.

### 7.2.5 OSCRLF

This routine generates a Line Feed and then a Carriage Return using the OSWRCH routine. A Register will contain 0D on exit.

### 7.2.6 OSACI

This routine uses the OSWRCH routine to output an ASCII character except that the Carriage Return character will be output as Line Feed and Carriage Return, using the OSCRLF routine.

### 7.2.7 OSECO

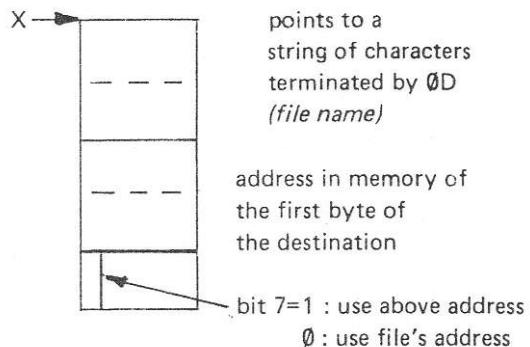
This routine fetches a byte using the OSRDCH routine and then writes it out using the OSACI routine.

### 7.2.8 OSRDCH

This subroutine fetches a byte from the input channel into the A Register. The state of N, Z and C flags is unknown, X and Y Registers are preserved.

### 7.2.9 OSLOAD

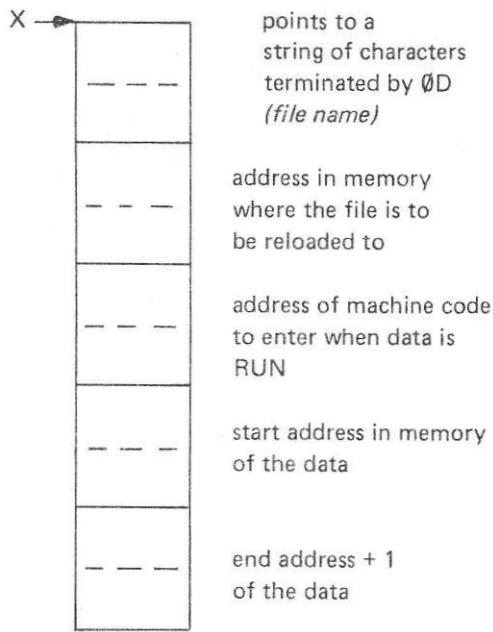
This routine loads all of a file into a specified area of memory. On entry the X Register must point to the following data in Zero Page:



This data is copied by the operating system and is not lost. All processor registers are used. If the processor's carry flag was set on input, a wait until completion is performed by interrupt or DMA driven systems. A BRK will occur if there is an error.

### 7.2.10 OSSAVE

This routine saves all of an area of memory to a specified file. On entry, the X Register must point to the following data in Zero Page:



This data is copied by the operating system and is not lost. All processor registers are used. If the carry flag was set on input, a wait until completion is performed by interrupt or DMA driven systems. A BRK will occur on an error.

### 7.2.11 OSSTAR

This routine sets the value of a file's pointer. On entry, the X Register points to Zero Page locations containing the value and the Y Register contains the file handle. The X and Y Register contents are not lost.

### 7.2.12 OSBGET

This routine returns the next byte from a random file. On entry the Y Register contains the file handle. The X and Y Register contents are not lost and the byte is returned in the A Register. The file's sequential pointer is incremented after the byte is read. Errors are met with a BRK.

### 7.2.13 OSBPUT

This routine adds the byte in the A Register to a random file. On entry, the Y Register contains the file handle. The A, X and Y Register contents are retained and the file's sequential pointer is incremented after the byte is added. Errors are met with a BRK.

### 7.2.14 OSRDAR

This subroutine returns the value of a random file's arguments. On entry, the X Register points to locations in Zero Page where the result is to be stored, and the Y Register contains the file handle, and the A Register specifies the argument, where:

- A=0 : the file's sequential pointer in bytes
- A=1 : the extent (length) of the file
- A=2 : the region of the file

The data, typically 3 bytes is placed at X, X+1, X+2. The X and Y Register contents are retained.

## 7.3 RANDOM FILE OPERATION

### 7.3.1 Introduction

The DOS entry points involved with random files are:

OSFIND	prepare file for random access
OSSHUT	close file, release buffer, tidy up
OSRDAR	read parameters of some open file
OSSTAR	update parameters of some open file
OSBGET	read byte from file
OSBPUT	write byte to file

At any one time there may be up to five random files active. These active files will each have a one byte internal name referred to as the "file handle". File handles are allocated by OSFIND, cancelled by OSSHUT and passed as arguments to all other routines. Valid file handles are all non-zero. The use of zero as a file handle cause some of the routines to perform special functions.

An open file has various status information associated with it, including:

The sequential pointer P (called PTR by BASIC)	
The file extent E (called EXT by BASIC)	
The file region R	

The file is viewed as a row of bytes labelled 0, 1, 2, 3 etc. The sequential pointer holds the number of the next byte to be read or updated. As OSBGET and OSBPUT access successive bytes of the file, they increment P, which is a three byte value. The file extent E is another three byte value which holds the number of characters stored in the file. E=0 indicates an empty file and when E=P an attempt to go further onwards will return an end of file marker and subsequently cause a 'BRK'. The region R is used when output is sent to a file.

When a new file is created a region of disc is set aside for it. The new file will have an extent of 0, and R will show the size of the disc block allocated. As bytes are written to the file, E is incremented and when E=R the file is full and no further bytes may be added. R is always a multiple of the disc sector size (256 bytes) and cannot subsequently be changed (files cannot be extended). When a file is SHUT any unused sectors are released. It will always be true that:

$$0 \leq P \leq E \leq R$$

and R is a multiple of 256.

### 7.3.2 OSRDAR and OSSTAR

OSRDAR and OSSTAR provide a means of interrogating P, E and R, and updating P. The ability to change P gives the user random access and the update capability for sequential files. If P is set beyond the extent of a file using OSSTAR, the space in the file from its old length to its new will be filled with bytes of value FF.

### 7.3.3 OSBPUT

OSBPUT writes bytes to a file. On entry, the A Register holds the byte to be written and the Y Register holds the file handle. If P=R, the file is full, and OSBPUT closes the file and executes a 'BRK'. Otherwise byte P of the file is updated and then, if P=E, both E and P are incremented (or P only is incremented). In the normal case when bytes are being added to the end of the file P=E.

### 7.3.4 OSBGET

OSBGET reads bytes from a file. On entry, the Y Register will hold the file handle. If P=E there are no more bytes in the file, so OSBGET sets the carry flag and returns the value FF. A second attempt to read at end of file causes a 'BRK'. Otherwise, OSBGET puts byte P of the file into the A Register, increments P and returns with the carry flag clear.

### 7.3.5 OSFIND

OSFIND opens files for input or output. Before calling OSFIND, it is necessary to provide a block of store containing the file name terminated by 0D, (Carriage Return). Two bytes in Zero Page point to this block, and the X Register contains the address of the pointer. If the carry flag is set the named file must already exist and E and R will be set to its actual size. If the file is not present on disc then OSFIND will return 0, this gives the user a way to detect whether a file exists or not.

If the carry flag is clear and the named file already exists, the old file region will be used, but with E set to zero initially. The result of this will be that the data in the old file cannot be accessed, but the region of the new file will be the same as the old, i.e. the old file is effectively deleted. If the old file was protected, OSFIND will fail. If no old file existed a new file is created with E set to zero and R given the default value of 4096 bytes. If there is not enough room on disc, then a 'BRK' is taken. If the user needs to control the size allowed for files (for instance requiring more than the default size), then the files should be pre-allocated by using SAVE so that OSFIND does not create them. Note that file names in OSFIND are effectively modified by the current drive and qualifier.

## 7.4 FILE BUFFERS AND CONTROL BLOCKS

The region of store from 2200 to 27FF is used for file buffers and control blocks. To reduce chance of disc corruption, the software maintains checksums on this memory, causing a 'BRK' if a check fails. In this event the safest thing to do is start from hardware reset, but in most cases it should be safe to shut files first.

## RACK, PSU AND BACKPLANE

### UNIT DESCRIPTION

### CONTENTS

Paragraph		Page
1.	INTRODUCTION	1
1.1	General	1
1.2	Leading Particulars	1
2.	RACK ASSEMBLY	1
3	POWER SUPPLY UNIT	2
4	BACKPLANE	3
5	PARTS LIST	4
5.1	Mechanical Parts	4
5.2	Electrical Parts	4

### TABLES

Table No.		Page
1	Power Supply Unit Connections	2
2	Typical Loads	2
3	Acorn Bus Connections	3

### FIGURES

Figure No.		Page
1	Front View	1
2	Rear View	2

©1982 Acorn Computers Limited, 12 Fulbourn Road, Cherry Hinton, Cambridge.  
Written and Produced for Acorn Computers Limited by Arch Technical Publications Limited,  
St. Lukes House, 5, Walsworth Road, Hitchin, Herts.

## 1. INTRODUCTION

### 1.1 GENERAL

The main Acorn System 5 components, printed circuit boards, disc drives, and power supply unit are mounted within a Rack Assembly. This rack is suitable for 19 in. rack mounting or as a free standing unit when housed in an optional instrument case (Figure 1).

The Acorn Bus is carried on two printed circuit backplanes with additional hard wired connections where necessary. With the exception of the mains lead (round cable) and backplane power connections, all other internal inter-unit wiring uses ribbon cable.

### 1.2 LEADING PARTICULARS

Size : Rack	: 480mm (W) x 177mm (H) x 240mm (D)
Cabinet	: 490mm (W) x 214mm (H) x 350mm (D)
Weight	: 16kg (approx.) fitted with one drive
Power Requirements	: 230V, 50/60Hz at 0.6A, OR 115V, 50/60Hz at 1.4

## 2. RACK ASSEMBLY

Figures 1 and 2 show the basic System 5 configuration with two floppy disc drives fitted.

The Rack Assembly consists of a front panel and two sideplates between which are fitted the upper and lower disc drive mounting plates and the card frame. A third sideplate provides a common endplate for the disc drive housing and card frame.

A small panel, mounted below the card frame carries the mains ON/OFF switch,

In its basic form, the rack assembly is suitable for 19 in. rack mounting. Alternatively, it may be housed in an instrument case as a free standing unit. The case includes feet and ventilation louvres and two carrying handles attached to the front panel.

The card frame will accommodate up to ten Eurocard printed circuit boards (100mm x 160mm). Figure 1 shows the card positions for the basic units. The remaining card slots will accommodate up to six further Acorn Bus compatible boards.

The disc drive housing accommodates one or two disc drives and appropriate front panels are provided. It should be noted that if a single disc drive system is updated to a dual drive system, the appropriate front panel should be ordered at the same time as the disc drive.

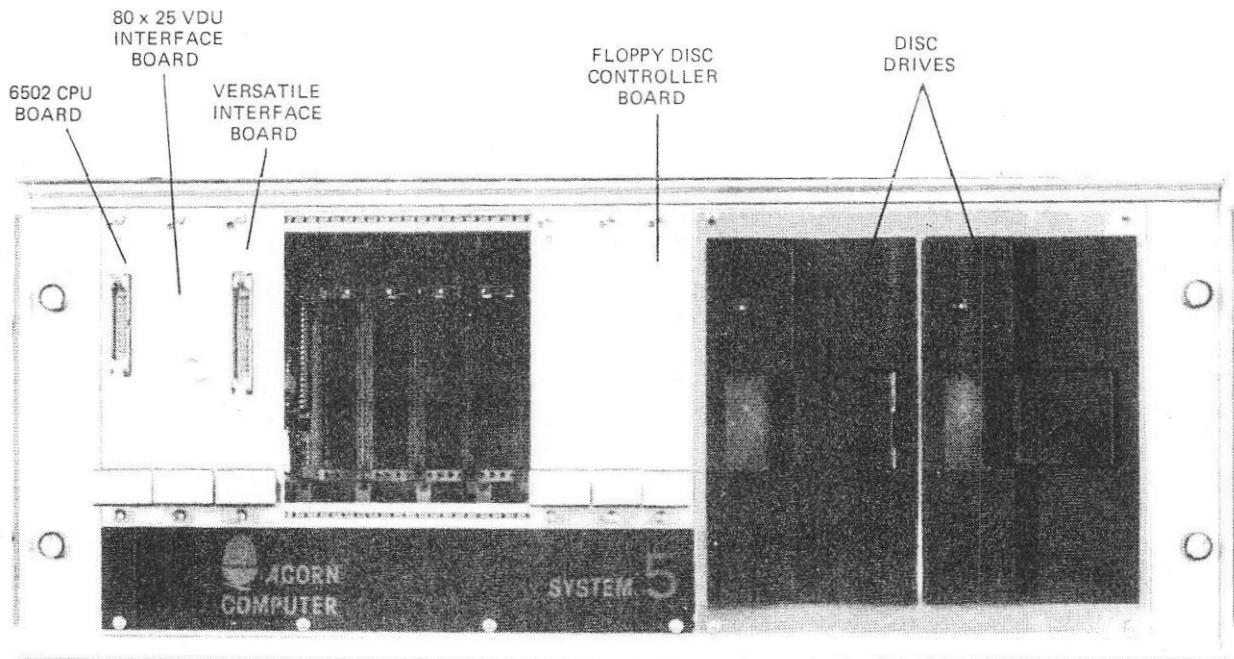
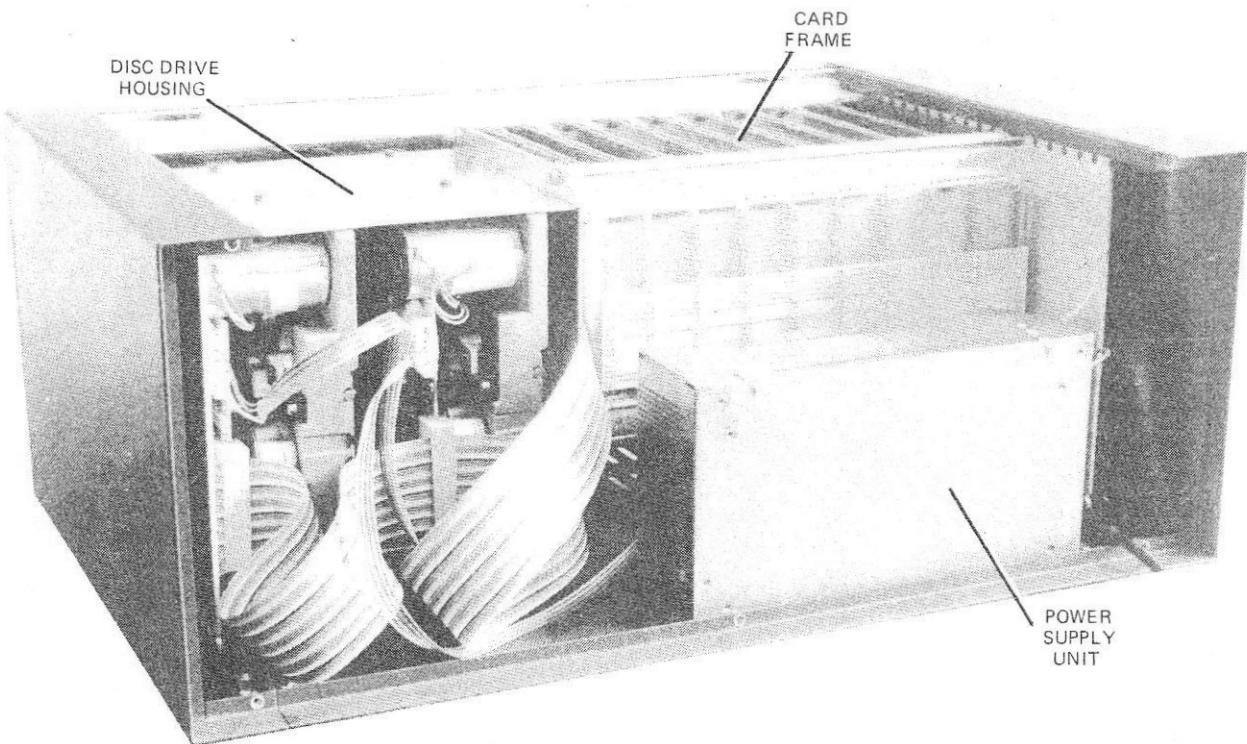


Figure 1. Front View



**Figure 2. Rear View**

### 3. POWER SUPPLY UNIT

Power for System 5 is provided by a type AC9231 Switch Mode Power Supply Unit. This unit is mounted behind the card frame and provides the following d.c. supplies:

- +5V at 6A
- -5V at 0.5A
- +12V at 2.5A
- -12V at 0.5A

The power supply unit will operate from mains supplies in the range 90V to 135V a.c. (115V a.c.

nominal) or 180V to 270V a.c. (230V a.c. nominal) with a supply frequency in the range 47Hz to 400Hz (50/60Hz nominal). It has the following features:

- Hold-up time of 24ms typical in the event of a power failure.
- 0.1% typical line regulation.
- 0.2% typical load regulation.
- Open or short circuit loads can be tolerated for an indefinite period.

Table 1 gives the input/output connections for the power supply unit and Table 2 lists typical loads drawn.

PIN	SERVICE
A.C. 1	LINE
2	NEUTRAL
D.C. 1	—
2	KEY
3/4	+12V
5/6	+5V
7/8/9	COMMON
10	+ve } 5V
11	-ve } 5V
12	+ve } 12V
13	-ve } 12V

**Table 1. Power Supply Unit Connections**

UNIT	TYPICAL LOAD CURRENTS	
	+5V	+12V
6502A CPU	520mA	
32K DRAM	280mA	
FDC	150mA	
80 x 25 VDU	450mA	
KEYBOARD	100mA	
DISC DRIVES:		
OLIVETTI	750mA	1.15A
SHUGART	500mA	900mA
TANDON	600mA	900mA

**Table 2. Typical Loads**

#### 4. BACKPLANE

The two backplane printed circuit boards provide the Acorn Bus connections to the ten card positions of the card frame.

The smaller backplane PCB carries the tracks for the B-side clock and interrupt signals.

The remaining B-side connections are hard wired between the ten edge connectors when necessary.

The large backplane PCB carries the printed circuit edge connectors and the tracks for the A-side connections.

The Acorn Bus connections are listed in Table 3.

PIN No.	SIDE A SIGNALS TO ALL BOARD POSITIONS	SIDE B – PINS MARKED ARE CONNECTED TO ALL BOARD POSITIONS									
		6502A CPU BOARD	32K DRAM BOARD	FDC BOARD	VERSA- TILE I/FACE BOARD	80 x 25 VDU I/FACE BOARD	TELE- TEXT VDU I/FACE BOARD	ICE BOARD	ADC BOARD	PROM PROG'R BOARD	ECONET I/FACE BOARD
1	+5V									+12V	
2	A15										
3	A14	PB7			PB7		PO				
4	NWDS	PB6			PB6		DE				
5	NRDS	PB5			PB5		NTLC				
6	NRST	PB4			PB4		NBCS				
7	A8	PB3			PB3		NDATA				
8	A7	PB2			PB2		DLIM				
9	A6	PB1			PB1						
10	A5	PBØ			PBØ						
11	A4	CB2			CB2						
12	A3	CB1			CB1						
13	A2	12MHz				12MHz					
14	A1	8MHz	8MHz								
15	AØ	6MHz									
16	D7	4MHz	4MHz	4MHz							
17	D6	3MHz									
18	D5	2MHz	2MHz			2MHz					
19	D4								+26V		
20	D3										
21	D2	1MHz	1MHz								
22	D1	NBGt									
23	DØ										
24	A13		NVMA		NVMA	NVMA		NVMA	NVMA	NVMA	NVMA
25	A12	NBRq									
26	A11	MR									
27	A1Ø										
28	A9	NIRQ			NIRQ			NIRQ	NIRQ		NIRQ
29	Ø2	NNMI		NNMI	NNMI			NNMI			
30	R/NW	SYNC			NIRQ			SYNC			
31	NBLKØ										
32	0V	0V	0V								

Notes: All A-side connections are carried on the main backplane PCB.

B-side connections B13 to B22, B28 and B29 are carried on the smaller backplane PCB.

B-side connections B1 to B12, B23 to B27 and B30 to B32 are hard wired if required by user,

Table 3. Acorn Bus Connections

## 5. PARTS LIST

### 5.1 MECHANICAL PARTS

ITEM	DESCRIPTION	VALUE	QTY	PART NO.
Instrument case		1	48-8346E *	
KM6 Universal end plates	4U x 240mm	2	173-24151C*	
End plate angles		2	173-24149E *	
Extrusion, top, front	84E	1	173-12552K *	
Extrusion, bottom, front	84E	2	173-12551B *	
Extrusion, rear, top	84E	1	173-13018E *	
Extrusion, rear, bottom	84E	1	173-12701A*	
Location mouldings, front		2	173-12557F *	
Location mouldings, rear		1	173-12558C *	
Tapped strip	84E	3	173-12704C *	
Card guides	160mm	20	173-12553G *	
Front panels	5E x 3U	8	174-12903K *	
Front panels	4E x 3U	2	174-12902B *	
Card mounting brackets		10	173-12525B *	
Screws	M4 x 12	10	173-12529A*	
Screws	M2.5 x 6	20	173-12530B *	
Screws, front panel		7	107-12712 *	
Bushes		7	172-12707 *	
Divider plate		1		
Drive fixing bracket (A)		1		
Drive fixing bracket (B)		1		
Drive front panel (single drive)		As Reqd		
Drive front panel (dual drive)		As Reqd		
Switch panel		1		
Strain relief bush		1		HEXCO 1140
Screws, Posidrive	M3 x 6	12		
Washers, plain	M3	12		
Washer, shakeproof	M3	1		
Screws, Posidrive	M4 x 12	7		
Washers, plain	M4	7		
Nut	M4	7		

\*Vero Part Nos.

### 5.2 ELECTRICAL PARTS

ITEM	DESCRIPTION	VALUE	QTY	PART NO.
Backplane Assembly comprising:				100,028
Backplane (1)		1		
Backplane (2)		1		
Sockets, 64-way		10		Straight DIN 41612
110 Fast on tabs		2		
Capacitor, ceramic disc	47n	1		
Power supply unit		1		Aztec AC9231
Mains warning label		1		RS556-159
Mains cable		1		201,057
PSU a.c. cable		1		201,056
PSU d.c. cable		1		201,055
Mains switch		1		Arrow 2600R21E
Earth Lead		1		201,072

## 6502A CPU BOARD

### UNIT DESCRIPTION

### CONTENTS

Paragraph	Page
1. INTRODUCTION	1
1.1 General	1
1.2 Leading Particulars	1
1.2.1 Mechanical	1
1.2.2 Power Supplies	1
1.2.3 Connections	1
2. CIRCUIT DESCRIPTION	3
2.1 Microprocessor	3
2.1.1 Microprocessor Registers	3
2.1.2 Microprocessor Inputs and Outputs	3
2.2 Address Select	5
2.2.1 RAM and VIA Selection	5
2.2.2 ROM Selection	7
2.3 System Clock	7
2.4 Wait Circuit	8
2.5 Ready Circuit	8
2.6 Read/Write Control	8
2.7 Read Only Memory	9
2.8 Random Access Memory	9
2.9 Data Buffer	9
2.10 Versatile Interface Adaptor	10
2.10.1 Peripheral Control Register	12
2.10.2 Auxiliary Control Register	13
2.10.3 Shift Register	13
2.10.4 Interrupt Flag Register	14
2.10.5 Interrupt Enable Register	14
2.10.6 Timer 1	14
2.10.7 Timer 2	14
2.11 Power-on Reset	14
3. SOFTWARE PARAMETERS	15
3.1 6502A Addressing Modes	15
3.2 6502A instruction Set	16
3.3 Interrupts and Reset	16
3.3.1 Interrupt Request	16
3.3.2 Non-Maskable interrupt	16
3.3.3 Reset	16
3.4 CPU Memory Addressing	18

©1982 Acorn Computers Limited, 12 Fulbourn Road, Cherry Hinton, Cambridge.  
Written and Produced for Acorn Computers Limited by Arch Technical Publications Limited,  
St. Lukes House, 5, Walsworth Road, Hitchin, Herts.

## CONTENTS (Cont'd)

<b>Paragraph</b>		<b>Page</b>
3.5	VIA Software Instructions	18
3.5.1	Data Input/Output	18
3.5.2	Timer 1	19
3.5.3	Timer 2	19
3.5.4	Shift Register Operation	19
3.5.5	Interrupt Operation	20
4.	6502A CPU CONNECTIONS	20
4.1	6502A CPU Board to Acorn Bus (PL2)	20
4.2	6502A CPU Board to I/O Unit (PL1)	21
5.	PARTS LIST	21
5.1	Mechanical Parts	21
5.2	Electrical Parts	21

## TABLES

Table No.		Page
1	Data Buffer Enable Conditions	10
2	VIA Register Selection	12
3	PCR Control Functions	12
4	SR Control Functions	14
5	Timer 1 Mode Selection	14
6	Instruction Set Coding	16
7	VIA Address Allocation	18

## FIGURES

<b>Figure No.</b>		<b>Page</b>
1	CPU Board Block Diagram	1
2	Microprocessor Internal Architecture	2
3	Microprocessor Registers	3
4	Microprocessor Signal Timing	4
5	DMA Timing (CPU Board Signals)	4
6	RAM and VIA Select Circuit	5
7	ROM Select Circuit	6
8	LK9 and LK4 Configurations	6
9	System Clock Timing Diagram	8
10	Read/Write Control Circuit	9
11	NRDS and NWDS Timing	9
12	Data Buffer Circuit	9
13	VIA Block Diagram	10
14	Typical VIA Operation Timing	11
15	PCR Bit Significance	12
16	ACR Bit Significance	13
17	Typical Serial Data Transfer Operation	13
18	IFR Bit Significance	14
19	IER Bit Significance	14
20	6502A CPU Memory Map	18
21	6502A CPU Board Circuit Diagram	23

## 1. INTRODUCTION

### 1.1 GENERAL

The 6502A CPU Board provides the Central Processor Unit (CPU) for a modular computer system. The CPU is constructed on a 100 x 160mm Eurocard which plugs into a standard Eurocard rack. A 6502A Microprocessor ( $\mu$ P) is run at 1, 2 or 3MHz from a 24MHz crystal controlled clock to provide the following CPU functions, refer also to the block diagram, Figure 1:

- **Read Only Memory (ROM).** A 28-pin DIL socket is provided for the installation of a ROM IC for the resident software. Links are wired on a header inserted in a 16-pin DIL socket to cater for a large range of 28-pin or 24-pin ROM ICs.
- **Random Access Memory (RAM).** A 24-pin DIL socket is provided for a 6116 2K x 8 bit static RAM IC. The RAM provides the stack and user memory areas. Other optional ICs such as 4118 may be used.
- **Input/Output (I/O).** A 6522A Versatile Interface Adaptor (VIA) IC fitted to the CPU Board provides two I/O Ports. The VIA Port (A) is

connected to PL1 to provide an interface with an external unit such as a Keyboard. The VIA Port (B) is connected to PL2 to provide a User Port.

The CPU address, data and control lines are buffered for connection to the standard Acorn Bus via PL2. Additional interrupt and Direct Memory Access (DMA) lines are also connected to PL2, as are the System Clock outputs.

### 1.2 LEADING PARTICULARS

#### 1.2.1 Mechanical

Construction : Single Eurocard printed circuit board.

Size : 100mm x 160mm

#### 1.2.2 Power Supplies

+5V ± 5% at 520mA typically.

#### 1.2.3 Connections

Pin connections are given in Section 4.

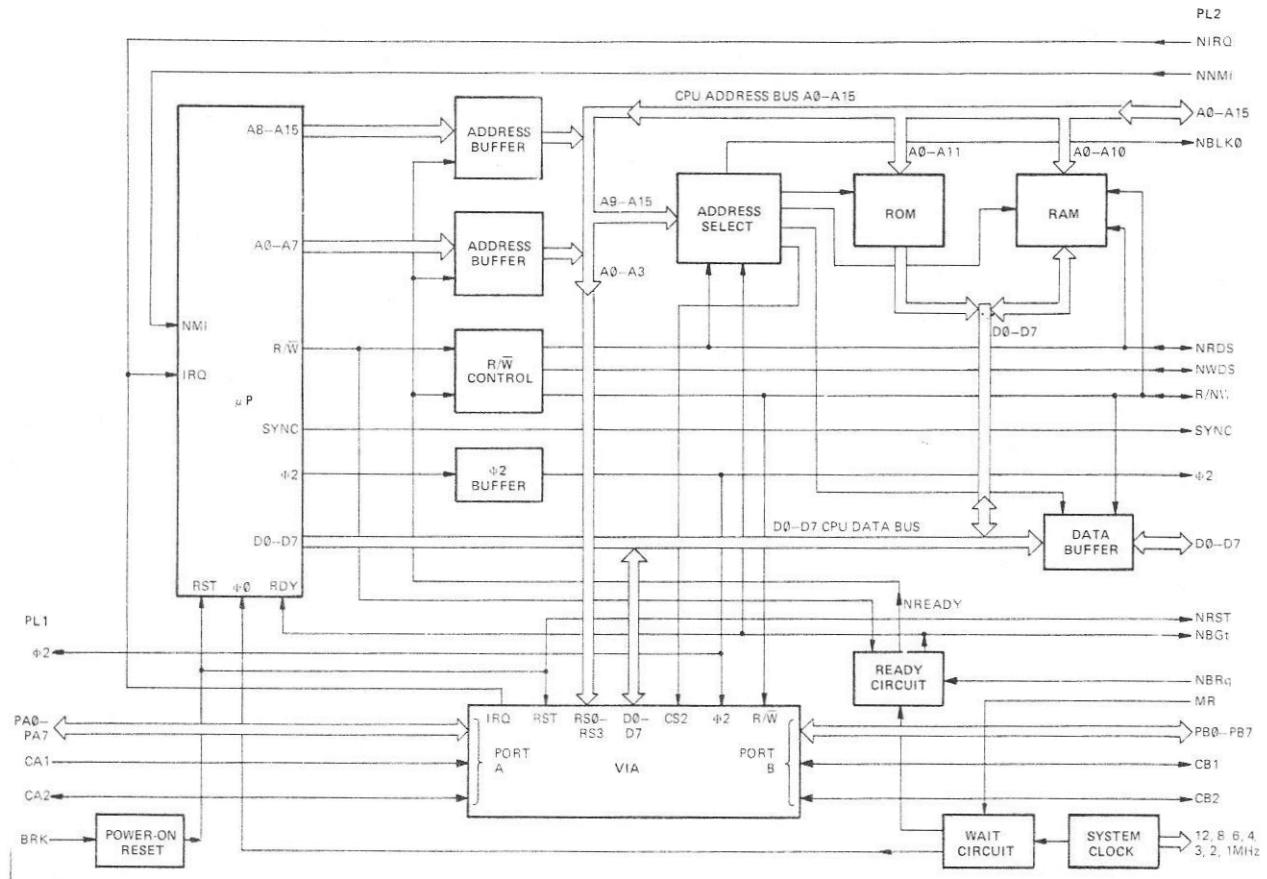


Figure 1. 6502A CPU Board Block Diagram

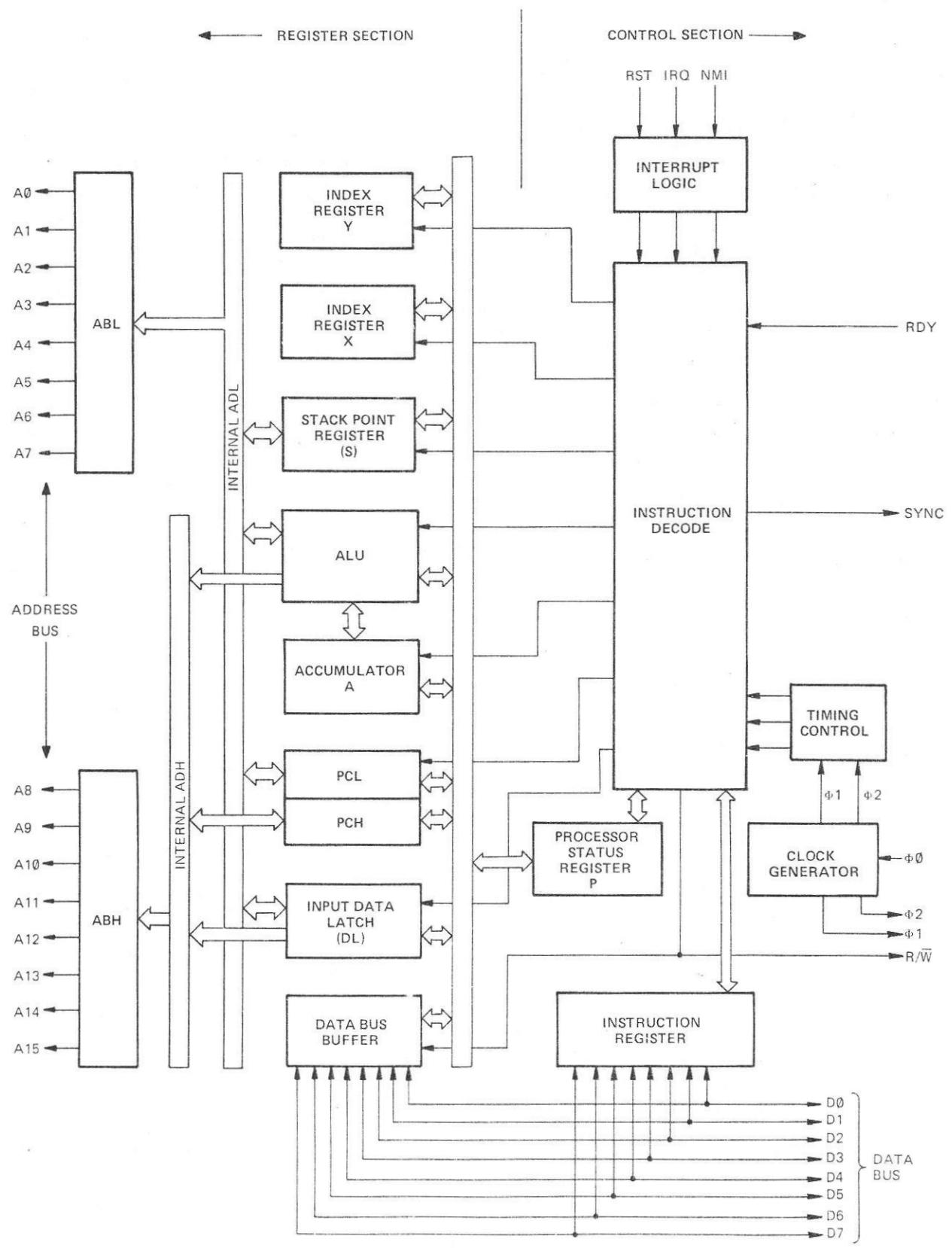


Figure 2. 6502A Microprocessor Internal Architecture

Connectors	: Double sided edge connector to Acorn Bus, TTL signal levels are used, DV to 0.4V = logic 0, >=+2.4V logic 1.
	20-way connector for external unit, i.e. Keyboard. TTL signal levels are used, 0V to 0.4V = logic 0, => +2.4V = logic 1.

## 2. CIRCUIT DESCRIPTION

The circuit diagram for the CPU Board is given on Figure 21 (filed at the end of this section) and should be referred to in conjunction with the description.

### 2.1 MICROPROCESSOR

The 6502A Microprocessor ( $\mu$ P) IC8 operates at 1MHz, 2MHz or 3MHz, depending upon the setting of links LK1, LK2 and LK3. The microprocessor has a 16 bit address bus and an 8 bit bi-directional data bus. CPU control is provided by the Read/Write (R/W) and clock ( $\$2$ ) output signals, refer to block diagram, Figure 2. Two interrupt inputs are provided, Interrupt Request (NIRQ) and Non-Maskable Interrupt (NNMI). Power-on reset, ready and synchronization connections are also provided.

The microprocessor address lines are buffered by IC9 and IC10. IC15 buffers the  $\$2$  and R/W lines.

#### 2.1.1 Microprocessor Registers

The 6502A Microprocessor internal registers are as shown on Figure 3. The functions of the registers are as follows:

- **Accumulator (A).** An 8 bit general purpose register used with the Arithmetic Logic Unit (ALU) for instruction execution. The A Register contents are not saved when an interrupt occurs.
- **Index (X and Y).** Two 8 bit index registers are provided. These registers are used for indexed addressing, a 16 bit offset can be used. The X and Y Register contents are not saved when an interrupt occurs.
- **Program Counter (PC).** A 16 bit register for the address of the instruction to be executed.
- **Stack Pointer (S).** A 9 bit register for the current stack address. On reset the S Register contains the address 01FF Hexadecimal (hex).
- **Status (P),** An 8 bit register that contains the microprocessor flags, refer to Figure 3. A '1' is loaded into the appropriate flag bit when a condition is true.

#### 2.1.2 Microprocessor Inputs and Outputs

The functions of the microprocessor inputs and outputs are as follows:

##### ADDRESS LINES : A0—A15

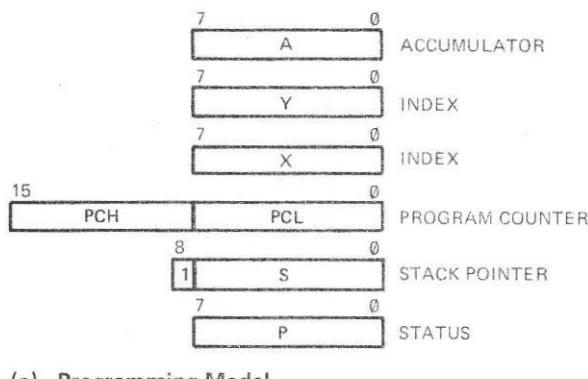
These 16 lines are used to carry the address output from the microprocessor. The address lines are buffered onto the CPU Address Bus and the Acorn Bus by IC9 and IC10. The NREADY signal low from the Ready bistable IC5/6, enables IC9 and IC10, refer to para 2.5. When the NREADY signal is high, the outputs of IC9 and IC10 are disabled (high impedance) to allow external DMA control of the CPU Board Address Bus via the Acorn Bus.

##### DATA LINES : D0—D7

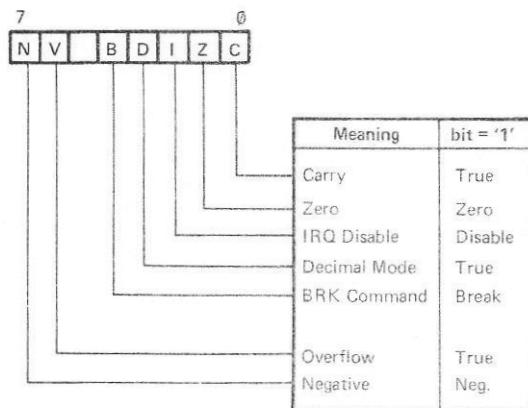
These 8 lines are used to carry the 8 bit data bytes to or from the CPU Data Bus. The CPU Data Bus is connected directly to the ROM IC11, the RAM IC13, the VIA IC7, and through the Data Buffer IC12, to the Acorn Bus.

##### READ/WRITE : R/W

The R/W signal is used to identify the direction of data transfer on the data lines. The R/W signal is high for data transfer to the microprocessor and low for data output from the microprocessor. The timing of the R/W signal is given on Figure 4. The R/W signal is buffered to the Acorn Bus by IC15/3, refer to Read/ Write Control, para 2.6.



(a) Programming Model



(b) P Register Bit Significance

Figure 3. Microprocessor Registers

#### CLOCK INPUT : \$0

The \$0 clock input drives the microprocessor clock circuits at the frequency selected, refer to para 2.3.

#### CLOCK OUTPUT : \$2

The \$2 clock output is derived internally by the microprocessor from the \$0 input. \$2 is in phase with \$0, refer to the timing diagram, Figure 4 (\$1 is

not used on the CPU Board). The \$2 signal is buffered by IC15/11 to the CPU Board and the Acorn Bus.

#### SYNC

The SYNC signal is used to identify the microprocessor, instruction fetch cycles. The timing of the SYNC signal is shown on Figure 4. The SYNC signal may be used in conjunction with the RDY signal to induce a microprocessor halt state, see RDY below.

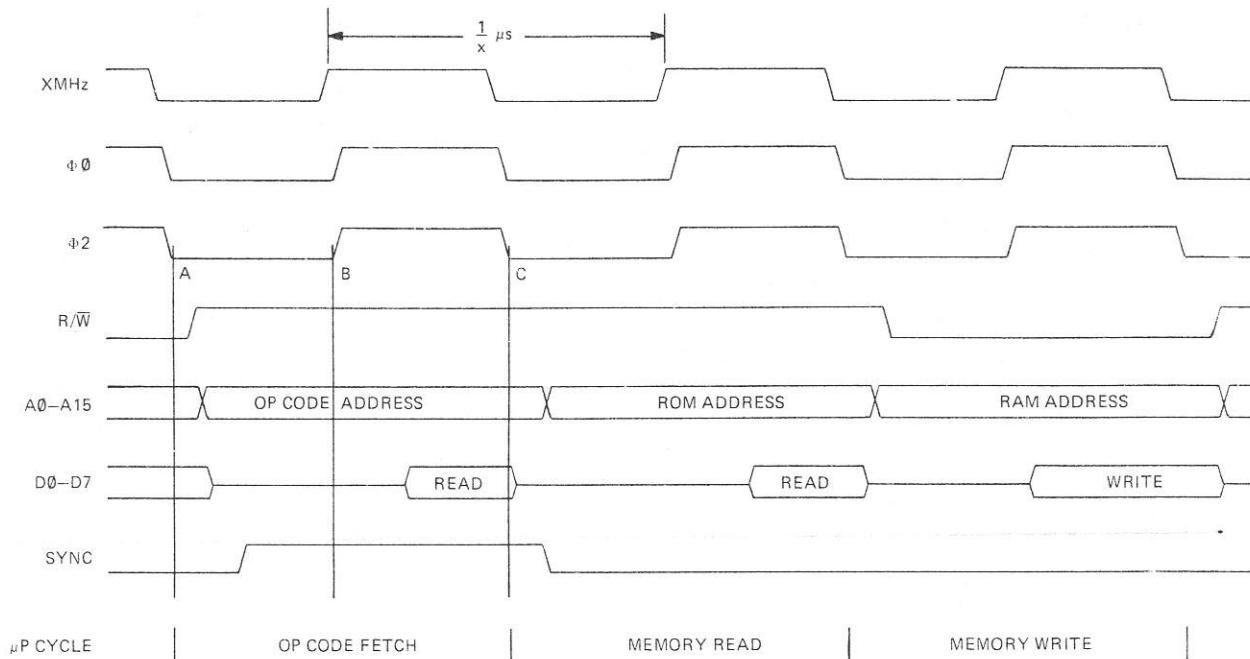


Figure 4. Microprocessor Signal Timing

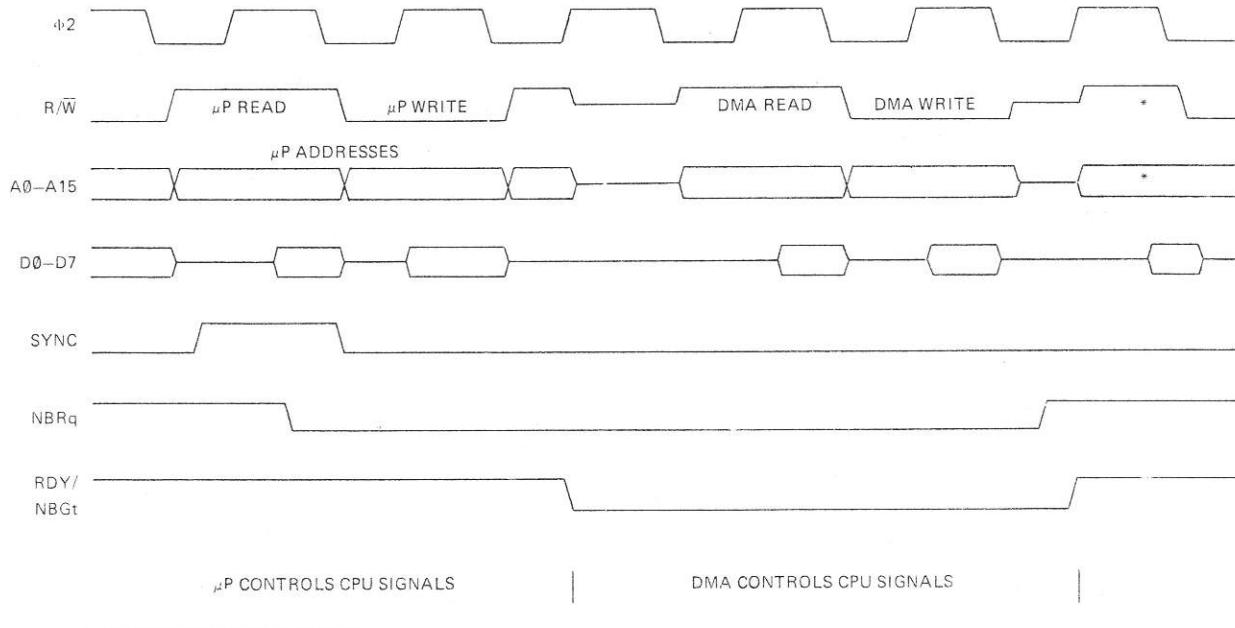


Figure 5. DMA Timing (CPU Board Signals)

#### READY SIGNAL : RDY

The RDY input signal to the microprocessor is normally high. When the microprocessor is used for DMA operations, a low input on the RDY line halts the microprocessor with the current address on the address lines, refer to para 2.5. If the microprocessor is executing a write cycle, the RDY signal is not set until the end of the cycle. Figure 5 shows a typical DMA operation with a halt state.

#### INTERRUPTS : NIRQ, NNMI

The microprocessor has two interrupt inputs, Interrupt Request (NIRQ) and Non-Maskable Interrupt (NNMI).

The NIRQ signal low is sampled at \$2 time and if not masked by the interrupt mask flag, begin the interrupt sequence at the next microprocessor machine cycle, refer to para 3.3.1. The vectored addresses are loaded from memory locations FFFE and FFFF (hex).

The NNMI signal going low initiates an unconditional interrupt, which is otherwise similar to the NIRQ interrupt, refer to para 3.3.2. The vectored addresses in this case are loaded from memory locations FFFA and FFFB (hex).

#### RESET : RST

The RST input is used to reset the microprocessor, or start the microprocessor from a power down condition. The NRST signal is held low on power-up for approximately 10ms, a positive edge then initiates a reset sequence, refer to para 3.3.3. An external reset can be initiated by applying 0V to PL1 pin 15, refer to para 2.11.

## 2.2 ADDRESS SELECT

The Address Select circuits provide address line decoding in normal microprocessor and DMA operation, for the RAM and ROM fitted to the CPU Board, and also for the VIA. Links are used to change the signals applied to the ROM pins to cater for different ROM ICs, IC13 position may be used for an additional ROM IC, which is then selected by the RAM Select circuit.

### 2.2.1 RAM and VIA Selection

The RAM and VIA Select circuit is shown on Figure 6. IC20 is a dual two line decoder. IC20a is enabled by the Block 0 address, address lines A12—A15 all '0's (0 hex). The output from IC6/C is also connected to PL2 pin 31b to generate the NBLK0 signal low on the Acorn Bus.

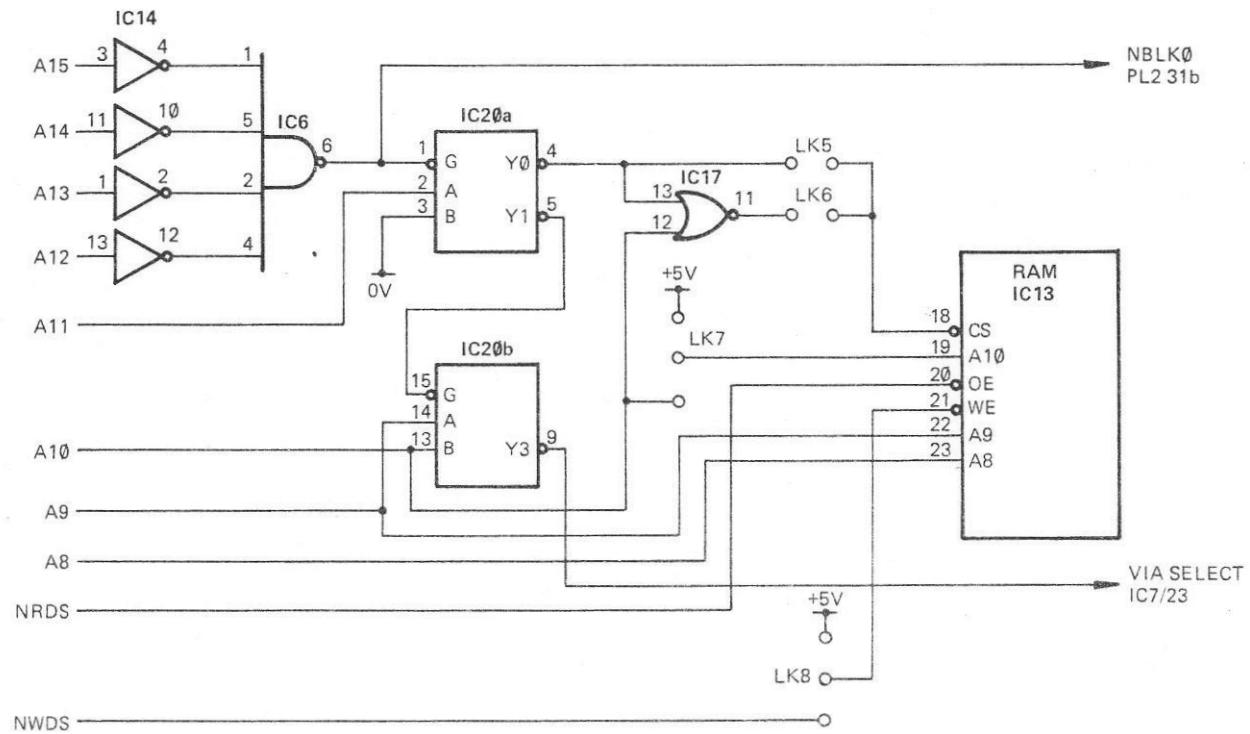
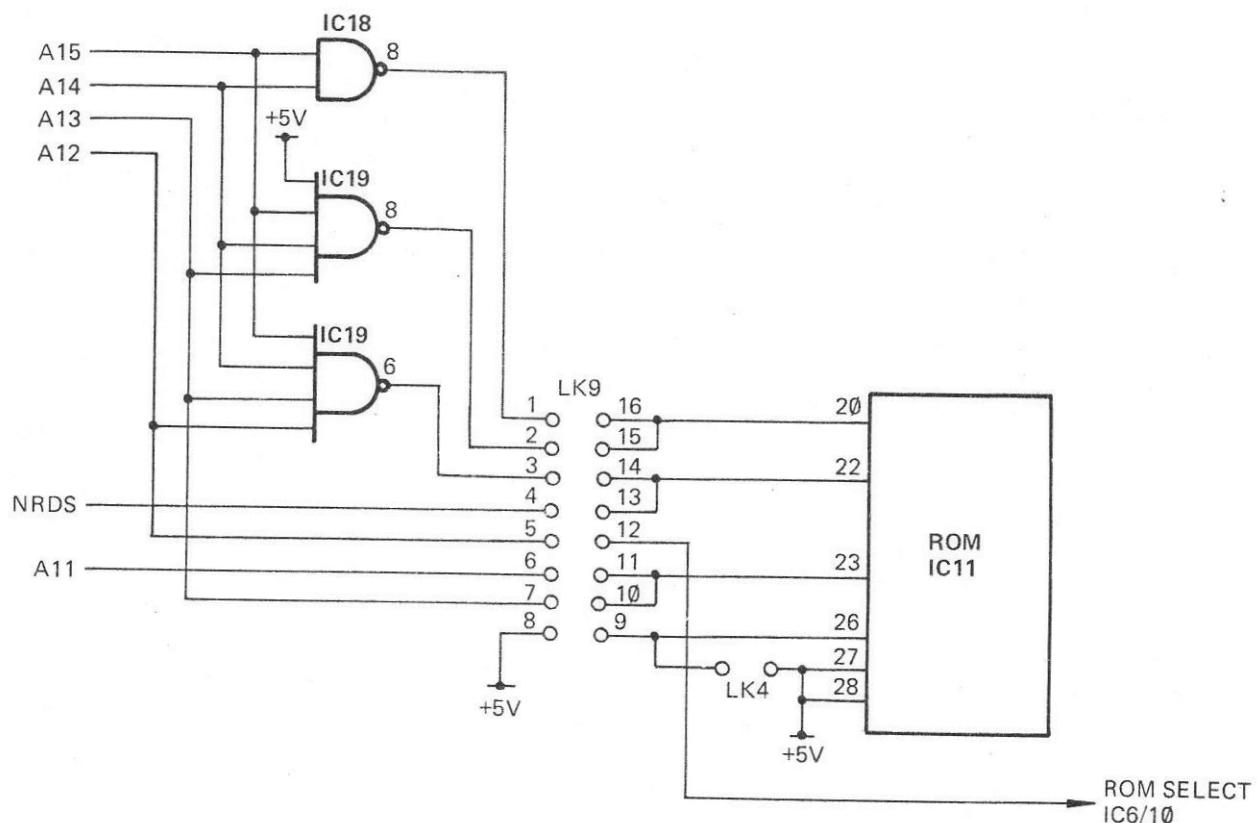
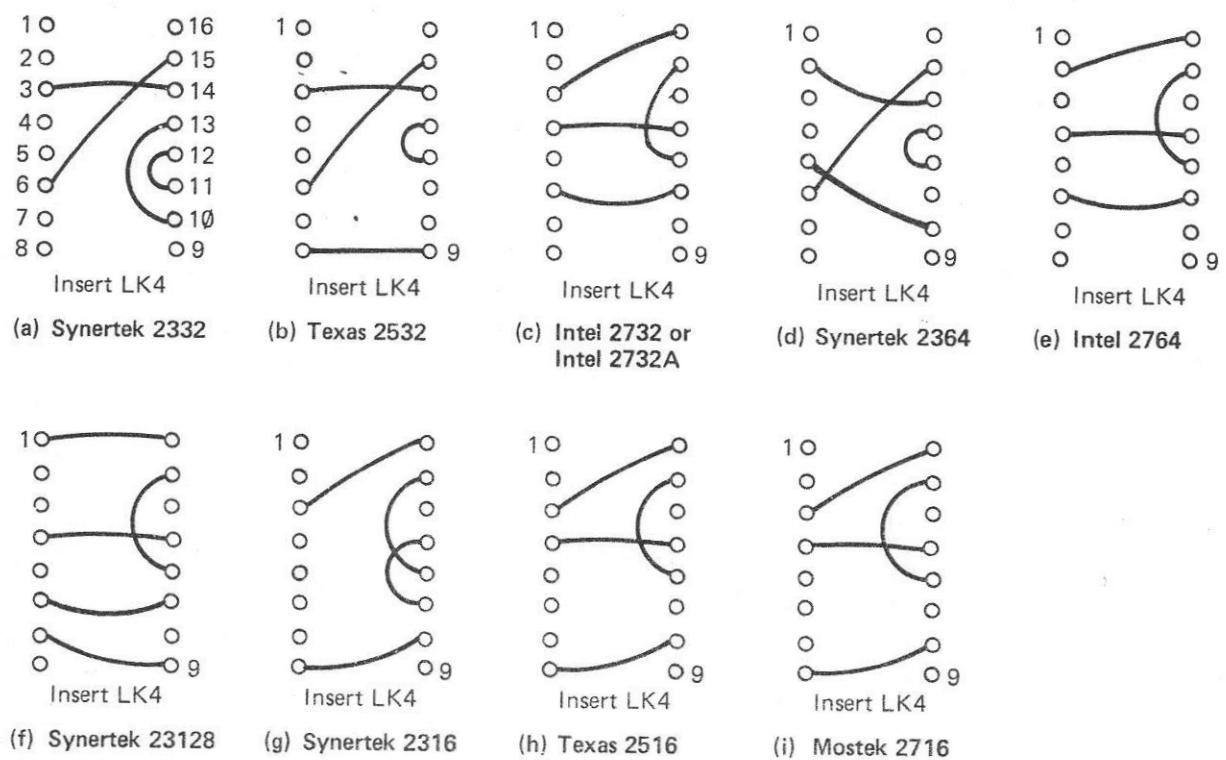


Figure 6. RAM and VIA Select Circuit



**Figure 7. ROM Select Circuit**



**Figure 8. LK9 and LK4 Configurations**

Address line A11 low generates a low on the Y0 output IC20/4, to select the RAM IC13, either via LK5, or via OR gate IC17/11 and LK6. LK5 is inserted when 2K of RAM is required in Block 0, addresses 0000 to 07FF (hex). When only 1K of RAM is required, LK6 is inserted and the RAM is selected for addresses 0000 to 03FF (hex) by address line A10 low at the OR gate. A10 high inhibits the RAM selection for addresses 0400 to 07FF (hex). The RAM select signal from LK5 or LK6 also inhibits the I/O Data Buffer for microprocessor RAM operations, refer to para 2.9.

The Y1 output of IC20a enables IC20b. IC20b decodes A10 and A9, both lines '1', to produce a low on the Y3 output IC20/9 and select the VIA for addresses 0E00 to 0FFF (hex). IC20/9 low also inhibits the I/O Data Buffer when the VIA is selected for microprocessor operations, refer to para 2.9.

Two links LK7 and LK8 are provided on the CPU Board to permit the use of other RAM or ROM ICs in the IC13 socket. The track on the Board has to be cut and links soldered to use this facility. LK8 connects pin 21 to +5V instead of NWDS and LK9 connects pin 19 to +5V instead of A10.

When a 4118 RAM IC is used in IC13, LK7 is connected to +5V and LK6 link inserted to give addresses 0000 to 03FF (hex) for the RAM.

The IC13 position can be used for ROM ICs. In this case LK8 is connected to +5V and LK5 link inserted to give addresses 0000 to 07FF (hex) for the ROM. Suitable ICs are 2516 and 2716.

### 2.2.2 ROM Selection

The ROM Select circuit is shown on Figure 7. The LK9 links are wired on a 16-pin DIL header that is inserted in the DIL socket. The link patterns for the various types of ROM IC which may be fitted in the IC11 socket, are shown on Figure 8. A link is also inserted in or removed from LK4 as indicated on Figure 8. The connection to LK9 pin 12 is used to inhibit the I/O Data Buffer during microprocessor memory read operations, refer to para 2.9.

The output of IC18/8 is low at LK9 pin 1 for all addresses in the range C000 to FFFF (hex) on the address lines.

The output of IC19/8 is low at LK9 pin 2 for all addresses in the range E000 to FFFF (hex) on the address lines.

The output of IC19/6 is low at LK9 pin 3 for all addresses in the range F000 to FFFF (hex) on the address lines.

The address lines A11 to A13 are connected to the appropriate pins on the ROM via links from LK9 pin 6, 5 and 7 respectively.

Signal NRDS is connected to the appropriate output enable (OE) pin on the ROM via a link from LK9 pin 4.

### 2.3 SYSTEM CLOCK

The System Clock is driven by the 24MHz crystal oscillator circuit X1, IC1, C3, R9, R10 and R13. Synchronous binary counters IC2, IC3 and IC4 divide the 24MHz clock signal output of the oscillator to produce: 12, 8, 6, 4, 3, 2 and 1MHz clock signals. The clock signals are synchronized to the falling edge of the 1MHz signal, refer to Figure 9. The microprocessor clock frequency is selected by a link as follows:

Link	Frequency
LK2	1MHz
LK3	2MHz
LK1	3MHz

The counter IC2 is configured to divide the 24MHz clock signal by three. The QB output is inverted by IC1 to clear IC2 every third clock pulse, refer to timing diagram, Figure 9. The IC2 QA output (8MHz) is connected to the enable P and T inputs of IC3, as well as to PL2 pin 14b via R15 (8MHz Acorn Bus clock).

The IC2 QA output is connected to the enable P and T inputs of counter IC3, so that IC3 counts every third 24MHz clock pulse. The QA, QB and QC outputs produce the 4, 2 and 1MHz clock signals. The 2 and 1MHz signals are connected to links LK2 and LK3 respectively, to provide the required microprocessor \$0 clock input via the Wait Circuit, refer to para 2.4. The 4, 2 and 1MHz clock signals are output via R17, R19 and R20 onto the Acorn Bus via PL2 pins 16b, 18b and 19b respectively. The IC3 carry output is used to synchronize IC4.

The counter IC4 divides the 24MHz clock signal to produce 12, 6 and 3MHz outputs at QA, QB and QC. The 3MHz output is connected to link LK1 to provide the microprocessor \$0 clock input if required. The 12, 6 and 3MHz clock signals are output via R14, R16 and R18 onto the Acorn Bus via PL2

pins 13b, 15b and 17b respectively. The carry output of IC3 is inverted by IC1 to clear IC4 every 2 $\mu$ s, thus ensuring that the IC4 outputs are synchronized to the 1MHz clock output from IC3, refer to timing diagram, Figure 9. Note that the IC3 carry output remains low, until the 8MHz clock signal at the enable T input goes high.

#### 2.4 WAIT CIRCUIT

The Wait Circuit provides the means for stopping the microprocessor for slow memory or peripheral operations. While the Memory Ready (MR) signal is high at PL2 pin 26b, the high on IC5/12 gives a low on the Q output pin IC5/8, so that the selected clock from link LK1, 2 or 3 is output from OR gate IC17/8. When signal MR goes low, the next positive going clock signal clears the Q output IC5/8 high to IC17/9. This applies a high on the microprocessor \$0 input to stop the microprocessor. When MR goes high again IC5/8 is set low and the clock output to the microprocessor is enabled.

#### 2.5 READY CIRCUIT

The Ready Circuit is used to introduce a halt state in the microprocessor for DMA operations.

While the Bus Request signal NBRq on the Acorn Bus is high at PL2 pin 25b, or the pin is open circuit (R7 pulls-up IC1/9), IC18/5 low applies a high to IC5/2. The Q output IC5/5 sets the microprocessor RDY input high to allow normal operations. This signal is also output on the Acorn Bus as the Bus Grant signal NBGt at PL2 pin 22b. -

When NBRq goes low and the microprocessor R/W signal output is high (read), IC18/6 goes low, so that the next positive edge of the \$0 clock from IC17/8 will reset IC5 Q output low. The low on the microprocessor RDY input induces a halt state, refer to Figure 5.

The Q output from IC5/6 inhibits the R/NW, NRDS and NWDS signals from buffers IC15 and the address line buffers IC9 and IC10, during the halt state, the buffer outputs go high impedance. When NBRq goes high the next positive edge of \$0 sets the Q output IC5/5 high, to generate the RDY signal high and end the microprocessor halt state.

#### 2.6 READ/WRITE CONTROL

The Read/Write Control Circuit buffers the microprocessor R/NW output from IC8/34 and generates the NRDS and NWDS signals for the Acorn Bus. The circuit is shown on Figure 10.

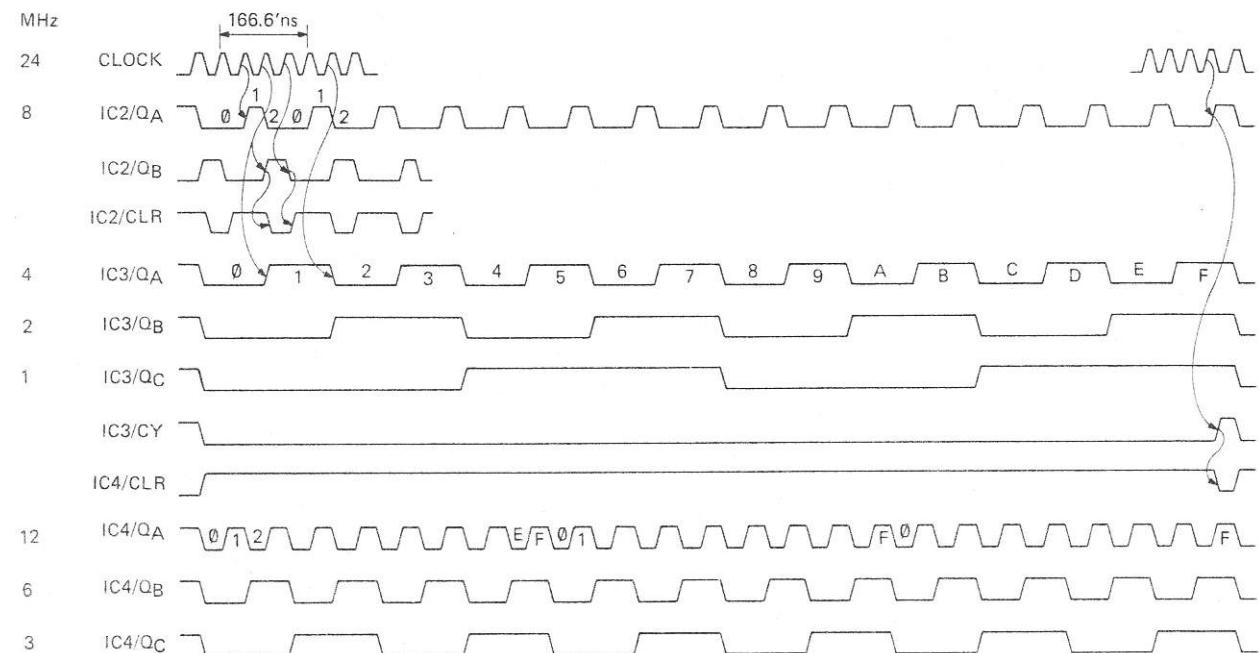
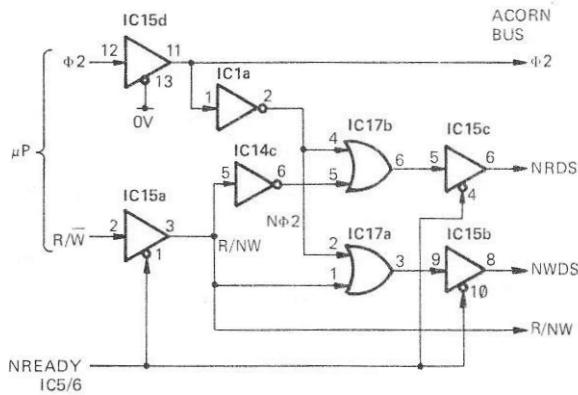
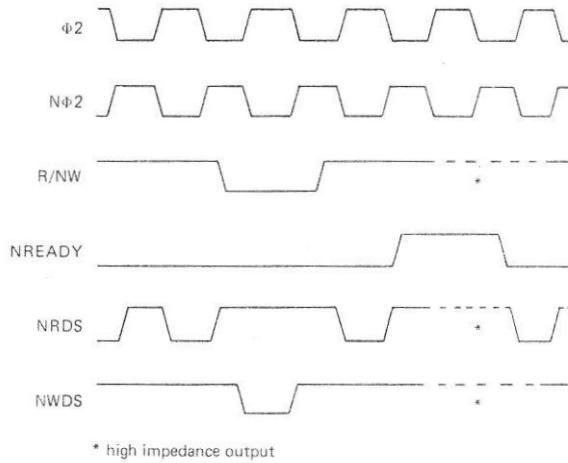


Figure 9. System Clock Timing Diagram



**Figure 10. Read/Write Control Circuit**

The  $\$2$  clock output from the microprocessor is buffered by IC15d and inverted by IC1a to enable IC17a and IC17b at each  $\$2$  clock time. The R/W signal output from the microprocessor is buffered by IC15a, then connected directly to IC17a and via inverter IC14c to IC17b. The Read and Write Control signals thus produced are buffered by IC15c and IC15b to generate the Read Strobe (NRDS) and Write Strobe (NWDS) signals for the Acorn Bus, refer to timing diagram, Figure 11. The IC5/6 Q output (N READY) high disables the buffers to the high impedance output state, to allow DMA control of the WNW, NR DS and NWDS Acorn Bus lines.



**Figure 11. NRDS and NWDS Timing**

## 2.7 READ ONLY MEMORY

The Read Only Memory (ROM) is selected by address lines A12—A15, refer to para 2.2.2. Various ROM ICs may be equipped in the IC11 position. IC13 may also be equipped with a ROM. When selected, address lines A0—A13 select the required location in the ROM, refer to para 2.2.2 and Figures 7 and 8 for

details of particular ROM address line connections. The data from the addressed location is output onto data lines D0—D7 by the NRDS signal low. When not selected the ROM data outputs are high impedance.

## 2.8 RANDOM ACCESS MEMORY

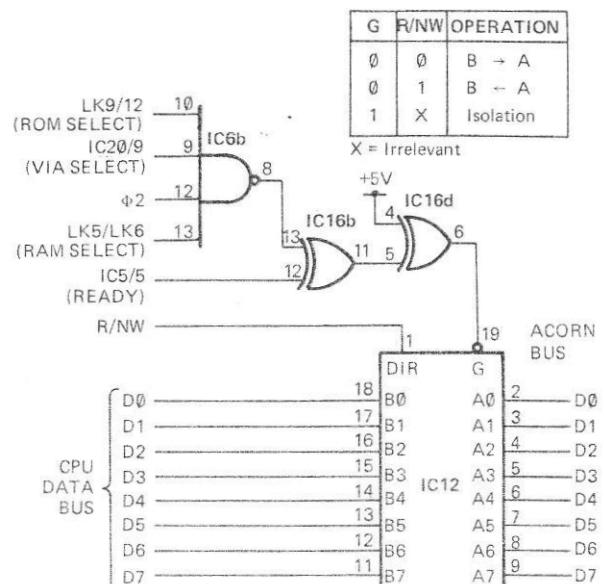
The Random Access Memory (RAM) is selected by address lines A11—A15, refer to para 2.2.1. The CS input low selects the RAM. Signal NRDS low outputs the data from the location addressed by address lines A0—A10 onto data lines D0—D7. Signal NWDS low writes the data present on data lines D0—D7 into the location addressed by address lines A0—A10. When not selected the RAM data lines are high impedance.

## 2.9 DATA BUFFER

The Data Buffer Circuit controls the transfer of data between the CPU Data Bus and the Acorn Bus. The circuit consists of the bi-directional octal Bus Buffer IC12, exclusive OR gates IC16/6 and IC16/11, and NAND gate IC6/8, refer to Figure 12. The IC12 I/O lines are high impedance when the IC is not enabled.

For normal CPU operation the Bus Buffer is enabled for data transfers during the Phase 2 ( $\$2$ ) clock time, unless the ROM, RAM or VIA on the CPU Board is addressed, refer to Table 1 for truth table.

For DMA operation, the Bus Buffer is disabled for data transfers during the  $\$2$  clock time, when neither a ROM, nor a RAM address is present on the Acorn Bus address lines. The Bus Buffer is enabled by a ROM, or RAM address.



**Figure 12. Data Buffer Circuit**

READ Y	SELECT ROM RAM VIA			§2	IC16/ 13 12 11	IC12 G
1	1	1	1	1	0 1 1	0
1	X	X	X	0	1 1 0	1
1	0	X	X	X	1 1 0	1
1	X	0	X	X	1 1 2	1
1	X	X	0	X	1 1 0	1
0	1	1	1	1	0 0 0	1
0	X	X	X	0	1 0 1	0
0	0	X	X	X	1 0 1	0
0	X	0	X	X	1 0 1	0

X = Irrelevant

Table 1. Data Buffer Circuit Enable Conditions

When IC12 is enabled, the direction of data transfer is determined by the R/NW signal. R/NW high selects data transfer from the Acorn Bus (A input), to the CPU Data Bus (B output). R/NW low selects output from the CPU Data Bus (B input) to the Acorn Bus (A output). The Acorn Bus data lines D0—D7 are tied down to 0V by the Resistor Pack RP1.

## 2.10 VERSATILE INTERFACE ADAPTOR

The Versatile Interface Adaptor (VIA) is used to provide two Input/Output (I/O) Ports for the CPU Board. Port A (PA0—PA7) is connected to PL1 for use with a Keyboard, Port B (PB0—PB7) is connected to PL2 to provide a general purpose I/O Port on the Acorn Bus. Handshaking capability is provided by the control lines CA1, CA2, CB1 and CB2. Refer to block diagram, Figure 13.

The VIA is selected by address lines A9 and A10 both high at IC20b, when the Y1 output from IC20a is low, this selection allocates addresses 0E00 to 0FFF (hex) to the VIA. The functions of the VIA are then selected by address lines A0—A3, which are connected to the register select inputs RS0—RS3. The VIA has the following internal registers:

- Two Output Registers (OR), one for each Port.
- Two Input Registers (IR), one for each Port.
- Two Data Direction Registers (DDR), one for each Port.
- A Peripheral Control Register (PCR).

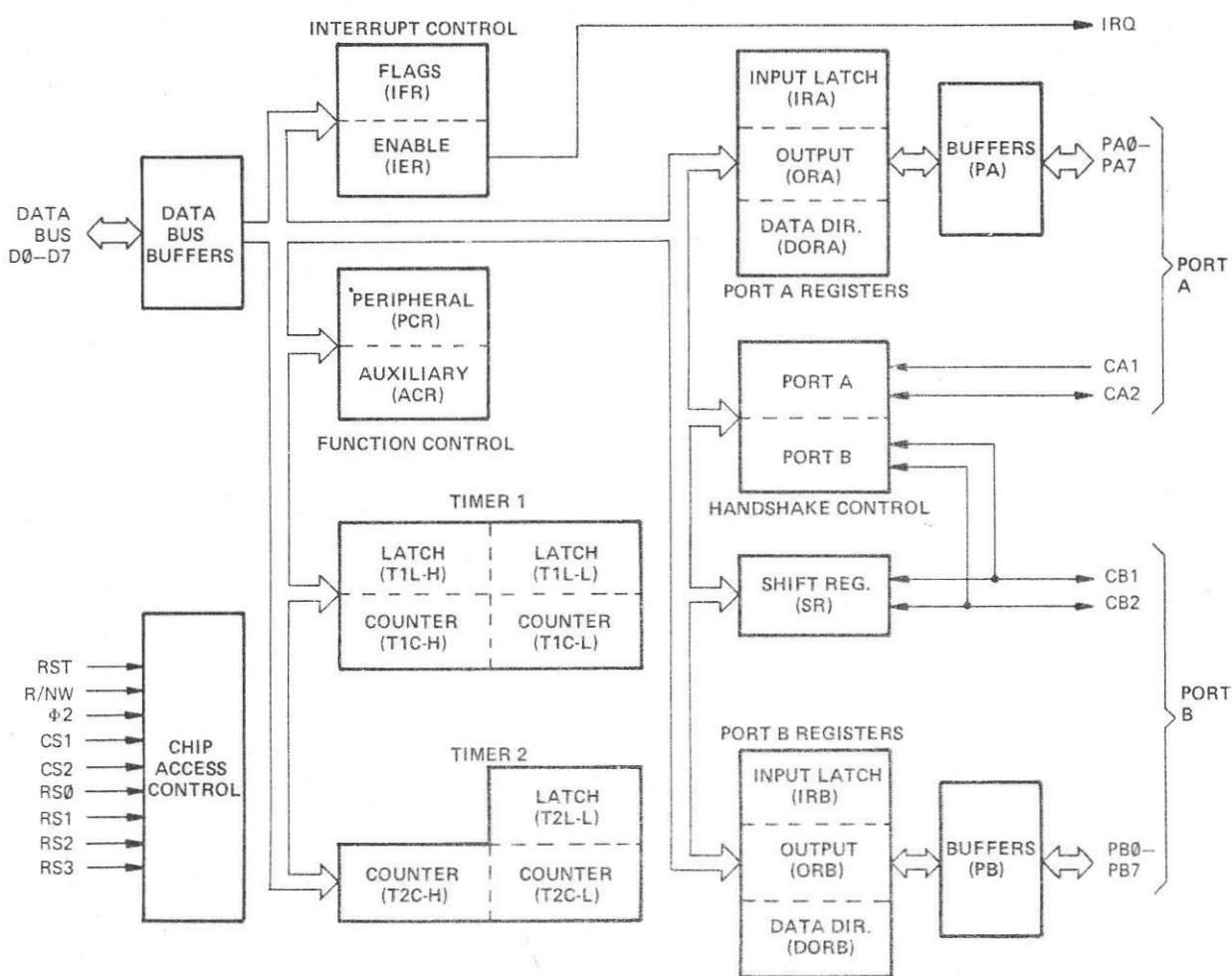
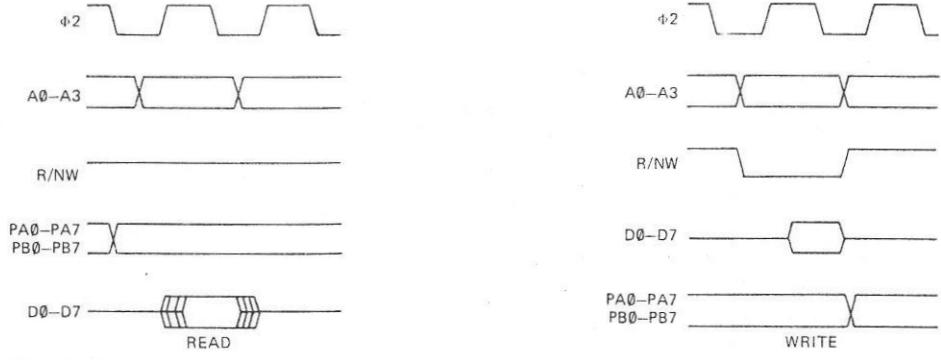
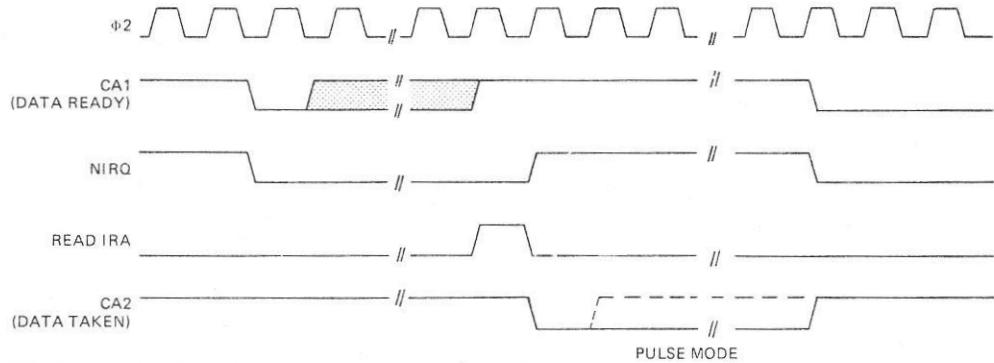


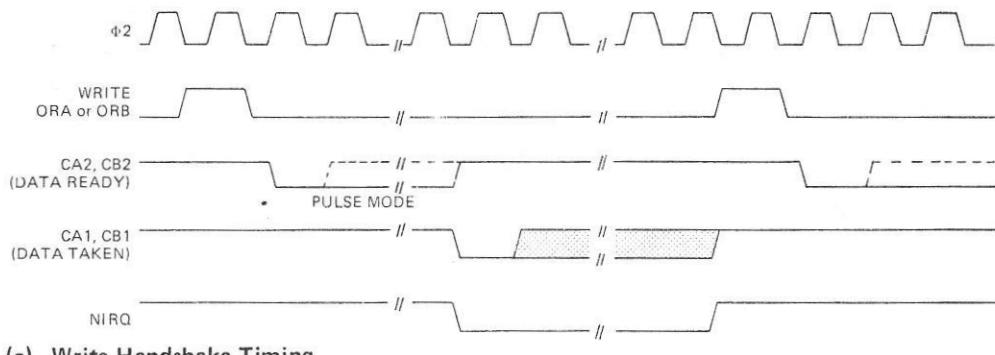
Figure 13. VIA Block Diagram



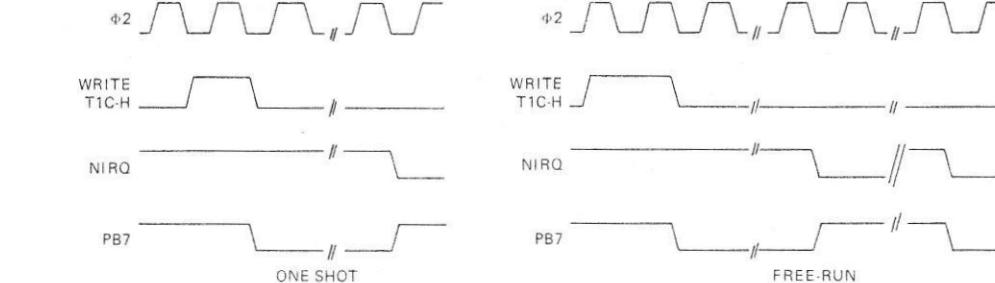
(a) CPU Timing



(b) Read Handshake Timing (Port A Only)



(c) Write Handshake Timing



(d) T1 Operation

Figure 14. Typical VIA Operation Timing

- An Auxiliary Control Register (ACR).
- An Interrupt Flag Register (IFR).
- An Interrupt Enable Register (IER).

The address selection of the registers is given in Table 2.

Hex	RS3	RS2	RS1	RS0	Register	Remarks
0	0	0	0	0	ORB/IRB	Controls hand-shake Port B
1	0	0	0	1	ORA/IRA	Controls hand-shake Port A
2	0	0	1	0	DDRB	Program Port B I/O pins
3	0	0	1	1	DDRA	Program Port A I/O pins
4	0	1	0	0	T1L-L	Low order latch (Write)
					T1C-L	Low order counter (Read)
5	0	1	0	1	T1C-H	Refer to para 2.10.6
6	0	1	1	0	T1L-L	
7	0	1	1	1	T1L-H	
8	1	0	0	0	T2L-L	Low order latch (Write)
					T2C-L	Low order counter (Read)
9	1	0	0	1	T2C-H	Refer to para 2.10.7
A	1	0	1	0	SR	
B	1	0	1	1	ACR	
C	1	1	0	0	PCR	
D	1	1	0	1	IFR	
E	1	1	1	0	IER	
F	1	1	1	1	ORA	No effect on handshake

**Table 2. VIA Register Selection**

The VIA is also provided with two Timers T1 and T2, and an 8 bit Shift Register (SR). The Timers can be used to control the Port I/O lines to generate programmable-frequency square waves and/or to count externally generated pulses. The SR can be used to translate the 8 bit parallel data on D0—D7, to serial data on CB2 and vice versa,

The Reset (RST) input of the VIA is connected to the Power-on Reset Circuit. Signal NRST low clears all the internal registers to the '0' state, except T2 and SR. All I/O lines are set to input.

A timing diagram of some typical VIA operations is given on Figure 14. All timed operations are controlled by the 6502A CPU Board Phase 2 (\$2) clock, except for serial data transfers when external clock mode is selected.

A link (LK10) is provided to connect the PA7 line from PL1 to the CA1 input, when Keyboard input operation under interrupt control is required.

### 2.10.1 Peripheral Control Register

The PCR is loaded by a microprocessor Write to the PCR address. The PCR is organized as shown on Figure 15. Table 3 gives the functions selected by PCR bits 1-3 for the Port A CA2 control line, the CA2 interrupt flag is IFR bit 0 (IFR0).

D	7	6	5	4	3	2	1	0
FUNCTION	CB2 Control		CB1 Control	CA2 Control	CA1 Control			

**Figure 15. PCR Bit Significance**

PCR BIT 3 2 1	FUNCTION
0 0 0	Input mode — Set CA2 interrupt flag (IFR0) on a negative transition of the input signal. Clear IFR0 on a Read or Write of the Port A output Register.
0 0 1	Independent Interrupt Input mode — Set IFR0 on a negative transition of the CA2 input signal. Reading or writing ORA does not clear the CA2 interrupt flag.
0 1 0	Input mode — Set CA2 interrupt flag on a positive transition of the CA2 input signal. Clear IFR0 with a Read or Write of the Port A Output Register.
0 1 1	Independent Interrupt Input mode — Set IFR0 on a positive transition of the CA2 input signal. Reading or writing ORA does not clear the CA2 Interrupt flag.
1 0 0	Handshake Output mode — Set CA2 output low on a Read or Write of the Port A Output Register. Reset CA2 high with an active transition on CA1.
1 0 1	Pulse Output mode — CA2 goes low for one cycle following a Read or Write of the Port A Output Register.
1 1 0	Manual Output mode — The CA2 output is held low in this mode.
1 1 1	Manual Output mode — The CA2 output is held high in this mode.

**Table 3 PCR Control Functions**

When 0V is applied to PL1 pin 15, C2 is rapidly discharged via R3. The NRST signal is set low until the 0V is removed and C2 has charged again. Similarly, C2 discharges via D1 when the +5V supply is switched off or interrupted.

### 3. SOFTWARE PARAMETERS

The following is a brief description of the software parameters affecting the CPU Board. Details are given of the 6502A Addressing Modes, Instruction Set and Interrupts, CPU memory addressing and the VIA software instructions.

#### 3.1 6502A ADDRESSING MODES

The 6502A Microprocessor is provided with the following 13 Addressing Modes:

##### ACCUMULATOR ADDRESSING

This form of addressing is represented with a one byte instruction, implying an operation on the accumulator.

##### IMMEDIATE ADDRESSING

In immediate addressing, the operand is contained in the second byte of the instruction, with no further memory addressing required.

##### ABSOLUTE ADDRESSING

In absolute addressing, the second byte of the instruction specifies the eight low order bits of the effective address while the third byte specifies the eight high order bits. Thus, the absolute addressing mode allows access to the entire 64K bytes of addressable memory.

##### ZERO PAGE ADDRESSING

The Zero Page instructions allow for shorter code and execution times by only fetching the second byte of the instruction and assuming a zero high address byte. Careful use of the Zero Page can result in significant increase in code efficiency.

##### INDEXED ZERO PAGE ADDRESSING — (X, Y INDEXING)

This form of addressing is used in conjunction with the Index Register and is referred to as "Zero Page, X" or "Zero Page, Y". The effective address is calculated by adding the second byte to the contents of the Index Register. Since this is a form of "Zero Page" addressing, the content of the second byte references a location in page zero. Additionally due to the "Zero Page" addressing nature of this mode, no carry is added to the high order of 8 bits of memory and crossing of page boundaries does not occur.

##### INDEX ABSOLUTE ADDRESSING --- (X, Y INDEXING)

This form of addressing is used in conjunction with X and Y Index Register and is referred to as "Absolute, X" and "Absolute, Y". The effective address is formed by adding the contents of X or Y to the address contained in the second and third bytes of the instruction. This mode allows the Index Register to contain the index or count value and the instruction to contain the base address. This type of indexing allows any location referencing and the index to modify multiple fields resulting in reduced coding and execution time.

##### IMPLIED ADDRESSING

In the implied addressing mode, the address containing the operand is implicitly stated in the operation code of the instruction.

##### RELATIVE ADDRESSING

Relative addressing is used only with branch instructions and establishes a destination for the conditional branch.

The second byte of the instruction becomes the operand which is an "Offset" added to the contents of the lower eight bits of the Program Counter when the counter is set at the next instruction. The range of the offset is -128 to +127 bytes from the next instruction.

##### INDEXED INDIRECT ADDRESSING

In indexed indirect addressing (referred to as [Indirect, X]), the second byte of the instruction is added to the contents of the X index Register, discarding the carry. The result of this addition points to a memory location on page zero whose contents is the lower order eight bits of the effective address. The next memory location in page zero contains the high order eight bits of the effective address. Both memory locations specifying the high and low order bytes of the effective address must be in page zero.

##### INDIRECT INDEXED ADDRESSING

In indirect indexed addressing (referred to as [Indirect, Y]), the second byte of the instruction points to a memory location in page zero. The contents of this memory location is added to the contents of the Y Index Register, the result being the low order eight bits of the effective address; The carry from this addition is added to the contents of the next page zero memory location, the result being the high order eight bits of the effective address.

##### ABSOLUTE INDIRECT

The second byte of the instruction contains the low order eight bits of a memory location. The high order eight bits of that memory location is contained in the

## INSTRUCTION SET

Table 6. Instruction Set Coding

third byte of the instruction. The contents of the fully specified memory location is the low order byte of the effective address. The next memory location contains the high order byte of the effective address which is loaded into the sixteen bits of the Program Counter.

### 3.2 6502A INSTRUCTION SET

The following Instruction Set is used by the 6502.A:

ADC	Add Memory to Accumulator with Carry
AND	"AND" Memory with Accumulator
ASL	Shift left One Bit (Memory or Accumulator)
BCC	Branch on Carry Clear
BCS	Branch on Carry Set
BEQ	Branch on Result Zero
BIT	Test Bits in Memory with Accumulator
BMI	Branch on Result Minus
BNE	Branch on Result Not Zero
BPL	Branch on Result Plus
BRK	Force Break
BVC	Branch on Overflow Clear
BVS	Branch on Overflow Set
CLC	Clear Carry Flag
CLD	Clear Decimal Mode
CLI	Clear Interrupt Disable Bit
CLV	Clear Overflow Flag
CMP	Compare Memory and Accumulator
CPX	Compare Memory and Index X
CPY	Compare Memory and Index Y
DEC	Decrement Memory by One
DEX	Decrement Index X by One
DEY	Decrement Index Y by One
EOR	"Exclusive or" Memory with Accumulator
INC	Increment Memory by One
INX	Increment Memory X by One
INY	Increment Memory Y by One
JMP	Jump to New Location
JSR	Jump to New Location Saving Return Address
LDA	Load Accumulator with Memory
LDX	Load Index X with Memory
LDY	Load Index Y with Memory
LSR	Shift One Bit Right (Memory or Accumulator)
NOP	No Operation
ORA	"OR" Memory with Accumulator
PHA	Push Accumulator on Stack
PHP	Push Processor Status on Stack
PLA	Pull Accumulator from Stack
PLP	Pull Processor Status from Stack
ROL	Rotate One Bit Left (Memory or Accumulator)
ROR	Rotate One Bit Right (Memory or Accumulator)

RTI	Return from Interrupt
RTS	Return from Subroutine
SBC	Subtract Memory from Accumulator with Borrow
SEC	Set Carry Flag
SED	Set Decimal Mode
SEI	Set Interrupt Disable Status
STA	Store Accumulator in Memory
STX	Store Index X in Memory
STY	Store index Y in Memory
TAX	Transfer Accumulator to Index X
TAY	Transfer Accumulator to Index Y
TSX	Transfer Stack Pointer to Index X
TXA	Transfer Index X to Accumulator
TXS	Transfer Index X to Stack Register
TYA	Transfer Index Y to Accumulator

The coding, number of cycle and number of bytes for each address mode, together with the status codes are given on Table 6.

### 3.3 6502A INTERRUPTS AND RESET

The 6502A microprocessor has two interrupt signal inputs NIRQ and NNMI and a reset input NRST. These inputs are used to load vectored addresses into the PC. The instructions located at the addresses are the first instructions of the relevant interrupt or start routine.

#### 3.3.1 Interrupt Request

A low level on the NIRQ input will initiate an interrupt sequence at the end of the current instruction, provided that the interrupt mask bit (I) in the Processor Status Register (P) is '0'. The contents of the Program Counter (PC) and the P Register are stored in the stack. The I bit is then set to '1' in the P Register. At the end of this cycle the PCL byte is loaded from address FFFE (hex) and the PCH byte is loaded from address FFFF (hex), transferring program control to the memory vector located at these addresses.

#### 3.3.2 Non-Maskable Interrupt

A negative going edge on the NNMI input will initiate an interrupt sequence at the end of the current instruction, the P Register I bit has no effect. The sequence is as described for NIRQ, except that PC is loaded from addresses FFFA (hex) and FFFB (hex).

#### 3.3.3 Reset

The NRST input low inhibits microprocessor operation. When a positive edge is detected the microprocessor starts the reset sequence. After the system initialization time of six clock cycles, the P Register I bit is set to a '1'. The PC is loaded from addresses

FFFF (hex) and FFFD (hex), transferring program control to the memory vector located at these addresses.

### 3.4 CPU MEMORY ADDRESSING

The CPU RAM and ROM are assigned to the addresses shown in the Acorn Memory Map, Figure 20. The different types of ROM are selected by means of the links fitted to the DIL header that is inserted in LK9, refer to para 2.2.2. The 1K or 2K RAM option 1 is selected by LK5 and LK6, refer to Para 2.2. IC13 may also be used for a ROM IC instead of a RAM IC.

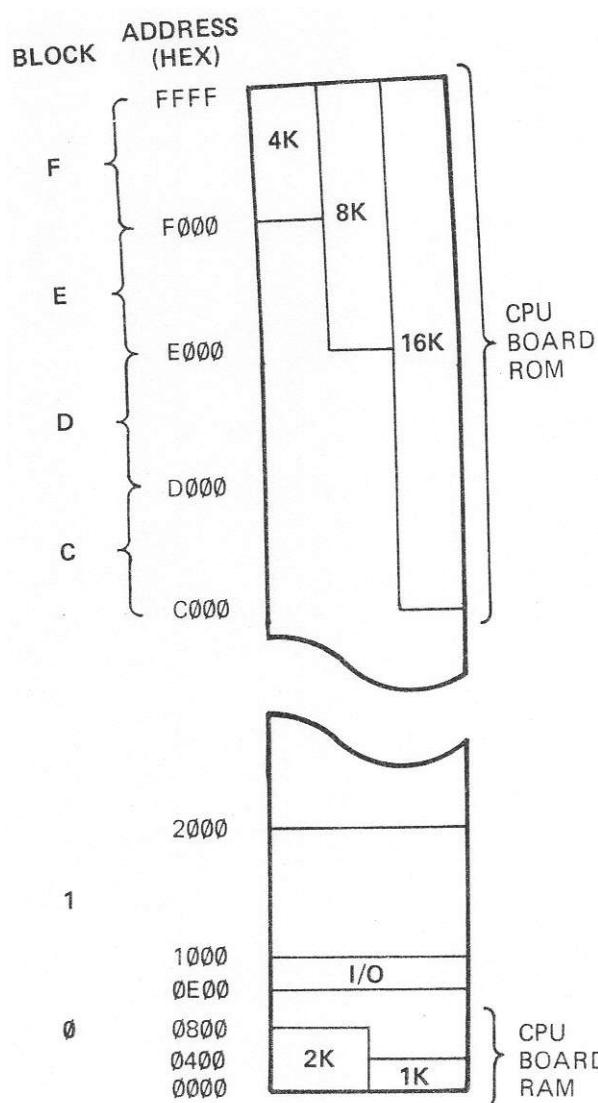


Figure 20. 5502A CPU Memory Map

### 3.5 VIA SOFTWARE INSTRUCTIONS

The operation of the VIA is controlled by program instructions from the microprocessor. The VIA is

assigned addresses 0E20 to 0E2F (hex), although because of incomplete decoding on the CPU Board, the address block 0E00 to 0FFF (hex) can only be used for the VIA. The VIA Register addressing is given on Table 7. The two Ports of the VIA are designated Port A and Port B.

ADDRESS BITS	R/NW	REGISTER	FUNCTION
A3 A2 A1 A0			
0 0 0 0	1	IRB	Input Register B
0 0 0 0	0	ORB	Output Register B
0 0 0 1	1	IRA	Input Register A
0 0 0 1	0	ORA	Output Register A
0 0 1 0	-	DDRB	Data Direction Register B
0 0 1 1	-	DDRA	Data Direction Register A
0 1 0 0	1	T1C-L	T1 Low Order Counter
0 1 0 0	0	T1L-L	T1 Low Order Latch
0 1 0 1	-	T1C-H	T1 High Order Counter
0 1 1 0	-	T1L-L	T1 Low Order Latch
0 1 1 1	-	T1L-H	T1 High Order Latch
1 0 0 0	1	T2C-L	T2 Low Order Counter
1 0 0 0	0	T2L-L	T2 Low Order Latch
1 0 0 1	-	T2C-H	T2 High Order Counter
1 0 1 0	-	SR	Shift Register
1 0 1 1	-	ACR	Auxiliary Control Register
1 1 0 0	-	PCR	Peripheral Control Register
1 1 0 1	-	IFR	Interrupt Flag Register
1 1 1 0	-	IER	Interrupt Enable Register
1 1 1 1	1	IRA	Input Register A (no handshaking)
1 1 1 1	0	ORA	Output Register A (no handshaking)

Table 7. VIA Address Allocation

#### 3.5.1 Data Input/Output

The selection of the input/output pins of both Ports is controlled by the Data Direction Registers (DDRA, DDRB). A pin is selected for input or output by the corresponding bit in the DDR, input is selected by a '0' hit, output by a '1' bit.

The data bits to be output from a Port are loaded into the corresponding Output Register (ORA, ORB). The bits are set onto the Port output lines at the end of the Write instruction. Writing to a pin selected for input loads the Output Register bit, but has no effect on the output pin.

The Input Registers (IRA, IRB) are loaded by the corresponding CA1 or CB1 signal, when input latching is enabled by the control bit in the ACR. A microprocessor Read operation then reads the contents of the IR. When input latching is disabled, a Read operation reads the current state of the Port input pin.

"Handshaking" is available on Port A for both input and output, but only for output on Port B. Handshaking signals are controlled automatically by Reading or Writing to Port A or Port B Registers, RS0—RS3 = 0 or 1 (hex). When Writing or Reading to Port A Registers with RS0—RS3 = F (hex), handshaking is disabled.

The handshaking signal operation is defined by the appropriate bits of the PCR, refer to para 2.10.1.

### 3.5.2 Timer 1

Timer 1 is controlled by the ACR, refer to para 2.10.2. For a single timed interrupt operation, Timer 1 is set by loading the Low Order Latch T1L-L and then the High Order Counter T1C-H. The contents of T1L-L are loaded into Counter T1C-L by the action of loading T1C-H. T1 is counted down to zero by the  $\frac{1}{2}$  clock. When zero is reached the interrupt flag IFR6 is set and, if interrupt enable IER6 is a '1', signal NIRQ is set low. IFR6 is reset by writing to T1C-H, or reading T1C-L.

For continuous interrupt (free-run) operation the Latch T1L-H has to be loaded as well as the Counter T1C-H. When T1 counts down to zero in this mode, T1C-L and T1C-H are reloaded from T1L-L and T1L-H, to restart the count down cycle.

When the PB7 output pin is selected for a single pulse operation, T1 operates as described above. A negative pulse is generated on PB7 for the duration of the timed interrupt. The pulse length =  $(n + 1.5) \frac{1}{2}$ , where n is the number loaded in the Counter. The DDRB bit 7 must be 1 for PB7 to function as an output.

For a square wave output on PB7, T1 operates in the free-run mode as described above. PB7 is low for  $(n + 1.5) \frac{1}{2}$  and high for  $(n + 2) \frac{1}{2}$ .

The T1 Latches may be loaded with a different number, while the Counters are counting down. When in the free-run mode, the new value is loaded into the Counters, when the count down reaches zero. If PB7 output is selected the output can thus be a complex waveform.

### 3.5.3 Timer 2

Timer 2 is controlled by the ACR, refer to para 2.10.2.

The T2 single timed interrupt operation is similar to that described for T1. The T2 interrupt flag is IFR5 and the interrupt enable IER5. After the count down the Counters roll over to all '1's and continue counting. The Counters can be read to determine how long the interrupt has been set. Reading T2C-L resets the interrupt flag IFR5. Writing to T2C-H also clears IFR5. T2 does not have a High Order Latch.

The T2 Low Order Latch T2L-L is also used to control Shift Register operations in the following modes:

- Shift in under T2 control.
- Free-running output at rate determined by T2.
- Shift out under T2 control.

For further details, refer to Shift Register Operation para 3.5.4.

In the pulse counting mode T2 is counted down from a number (loaded into the Counters as described above) by negative going pulses on input PB6. The DDRB bit 6 must be set to '0' to select input. When T2 = '0' IFR5 is set and, if IER5 is a '1', signal NIRQ is set low. Writing to T2C-H clears IFR5.

### 3.5.4 Shift Register Operation

The SR has eight operating modes that are controlled by setting the ACR bits 2, 3 and 4, refer to para. 2.10.3. The SR interrupt flag IFR bit 2 (IFR2) is used to request a data byte from the microprocessor, or signal that a data byte can be read from the SR. When the interrupt enable IER2 is '1', setting IFR2 sets signal NIRQ low. IFR2 is cleared by Writing to or Reading the SR. When the SR is disabled, ACR bits 2-4 = '000', CB1 and CB2 are controlled by the PCR and IFR2 is held at '0'.

For serial data input or output on C82, the data bits are clocked in or out under the control of either an internal clock or an external clock input on CB1 as follows:

- *Input, T2 clock control.* The clock output on CB1 is a square wave of period 2  $(n + 2) \frac{1}{2}$ , where n is the number-stored in T2L-L. Transfer is initiated by a Write or Read to the SR. The data bit on CB2 is shifted into the SR by the  $\frac{1}{2}$  clock cycle following the positive going edge on CB1. After 8 CB1 pulses IFR2 is set.
- *Input,  $\frac{1}{2}$  clock control,* The clock output on CB1 is  $\frac{1}{2}$  divided by 2. Transfer is initiated

by a Read or Write to the SR. The clock on CB1 stops after 8 pulses and IFR2 is set.

- *Input, external clock control.* The SR is loaded with the data bit on CB2 by the first  $\$2$  clock cycle after the positive going clock edge on CB1. The SR sets IFR2 after 8 clock pulses.
- *Output, T2 control free-running.* The 8 bits stored in the SR are clocked onto CB2 repetitively. The clock period on CB1 is  $2(n + 2) \$2$ , where n is the number stored in T2L-L. Signal NIRQ is not set. Transmission is initiated by a Write to the SR.
- *Output, T2 clock control.* The 8 bits loaded in the SR are clocked onto CB2. The clock period on CB1 is  $2(n + 2) \$2$ , where n is the number stored in T2L-L. Transmission is initiated by a Write to the SR. When the 8 data bits have been sent IFR2 is set.
- *Output, \\$2 clock control.* The clock output on CB1 is  $\$2$  divided by 2. The 8 bits loaded in the SR are clocked onto CB2. Transmission is initiated by a Write to the SR. When the 8 data bits have been sent IFR2 is set.
- *Output, external clock control.* The 8 bits loaded in SR are clocked onto CB2 by the first  $\$2$  clock cycle after the negative going edge of the external clock input on CB1. Transmission is initiated by Writing to the SR. When 8 clock pulses have been counted IFR2 is set.

### 3.5.5 Interrupt Operation

The IFR flags are set by the various functions in the VIA. When a flag is set IFR bit 7 is set to indicate an interrupt condition. Setting a flag sets the NIRQ output low, provided that the corresponding IER bit is set. The IFR can be read by the microprocessor. An IFR flag can be reset by Writing a '1' to the appropriate IFR bit.

The IER flags are set to enable the corresponding IFR flags by Writing to the IER with a '1' in bit 7 and '1's corresponding to the flags. An IER flag can be reset by Writing to the IER with a '0' in bit 7 and a '1' corresponding to the flag. The setting of the IER flags can be read by the microprocessor, bit 7 will be read as a '1'.

## 4. 6502A CPU CONNECTIONS

### 4.1 6502A CPU BOARD TO ACORN BUS (PL2)

Pin	Mnemonic	Meaning	I/O
Side A			
1	+5V	+5V Board Supply	I
2	A15		
3	A14		
4	NWDS		
5	NRDS		
6	NRST		O
7	A8		
8	A7		
9	A6		
10	A5		
11	A4		
12	A3		
13	A2		
14	A1		
15	A0		
16	D7		
17	D6		
18	D5		
19	D4		
20	D3		
21	D2		
22	D1		
23	D0		
24	A13		
25	A12		
26	A11		
27	A10		
28	A9		
29	$\Phi_2$	Phase 2 Clock	O
30	R/NW	Read/Write Control Line	I/O
31	NBLK0	Block 0 Address Select	O
32	OV	OV Board Supply	I

Pin	Mnemonic	Meaning	I/O
Side B			
1			
2			
3	PB7		
4	PB6		
5	PB5		
6	PB4		
7	PB3		
8	PB2		
9	PB1		
10	PB0		
11	CB2		
12	CB1		

Pin	Mnemonic	Meaning	I/O
13	12MHz		
14	8MHz		
15	6MHz		
16	4MHz		O
17	3MHz		
18	2MHz		
19			
20			
21	1MHz	Clock Signal	O
22	NBGt	Bus Grant	O
23			
24			
25	NBRq	Bus Request	I
26	MR	Memory Ready	I
27			
28	NIRQ	Interrupt Request	I
29	NNMI	Non-Maskable Interrupt	I
30	SYNC	Microprocessor Instruction Fetch	O
31			
32	0V	0V Board Supply	I

#### 4.2 6502A CPU BOARD TO I/O UNIT (PL1)

Pin	Mnemonic	Meaning	I/O
1	CA1		
2	CA2		
3			
4			
5			
6			
7			
8	Φ2	Phase 2 Clock	O
9			
10	PA0	Port A Data Line	I/O
11	0V	0V Supply	O
12	PA1	Port A Data Line	I/O
13	+5V	+5V Supply	O
14	PA2	Port A Data Line	I/O
15	RESET	System Reset	I
16	PA3		
17	PA7		
18	PA4	Port A Data Lines	I/O
19	PA6		
20	PA5		

## 5. PARTS LIST

### 5.1 MECHANICAL PARTS

ITEM	DESCRIPTION	VALUE	QTY	PART NO.
	Printed Circuit Board 200,005		1	
	14 Pin IC Socket		9	
	16 Pin IC Socket		5	
	20 Pin IC Socket		3	
	24 Pin IC Socket		1	
	28 Pin IC Socket		1	
	40 Pin IC Socket		2	
	16 Pin DIL Header		1	
	2 Pin Links	Molex 22-03-2021	6	
	Molex Mini-Shunts	7859-2	6	

### 5.2 ELECTRICAL PARTS

ITEM	DESCRIPTION	VALUE	QTY	PART NO.
C1	Capacitor Electrolytic	22µF 16V	1	
C2	Capacitor Electrolytic, Tantalum Bead	10µF 16V	1	
C3	Capacitor Ceramic Disc	100pF	1	
C4...C9	Capacitor Ceramic	47nF	6	

ITEM	DESCRIPTION	VALUE	QTY	PART NO.
PL1	Connector 20 Way, Right Angle Plug Solder Tail	3M 3428-1302		
PL2	Connector 64 Way, Right Angle Plug Solder Tail	DIN 41612		
D1	Diode	IN4148	1	
IC1	Integrated Circuit	74S04	1	
IC2...IC4	Integrated Circuit	74S163	3	
IC5	Integrated Circuit	74LS74	1	
IC6	Integrated Circuit	74LS20	1	
IC7	Integrated Circuit	SY6522A	1	
IC8	Integrated Circuit	SY6502A	1	
IC9, IC10	Integrated Circuit	74 LS244	2	
IC11	Integrated Circuit	*	1	
IC12	integrated Circuit	74LS245	1	
IC13	Integrated Circuit	6116LP-4	1	
IC14	Integrated Circuit	74LS04	1	
IC15	Integrated Circuit	74LS125	1	
IC16	Integrated Circuit	74LS86	1	
IC17	Integrated Circuit	74LS32	1	
IC18	Integrated Circuit	74LS132	1	
IC19	Integrated Circuit	74LS20	1	
IC20	Integrated Circuit	74LS139	1	
* PROM supplied in accordance with System requirements.				
R1, R2	Resistor	4.7kOhm	2	
R3	Resistor	100Ohm	1	
R4...R6	Resistor	4.7kOhm	3	
R7, R8	Resistor	1kOhm	2	
R9, R10	Resistor	470Ohm	2	
R11, R12	Resistor	4.7kOhm	2	
R13	Resistor	10kOhm	1	
R14...R20	Resistor	68Ohm	7	
R21	Resistor	1kOhm	1	
RP1	Resistor Pack SI L 850-91-2K2 AB		1	
X1	Crystal 24MHz (Fun. amental)		1	

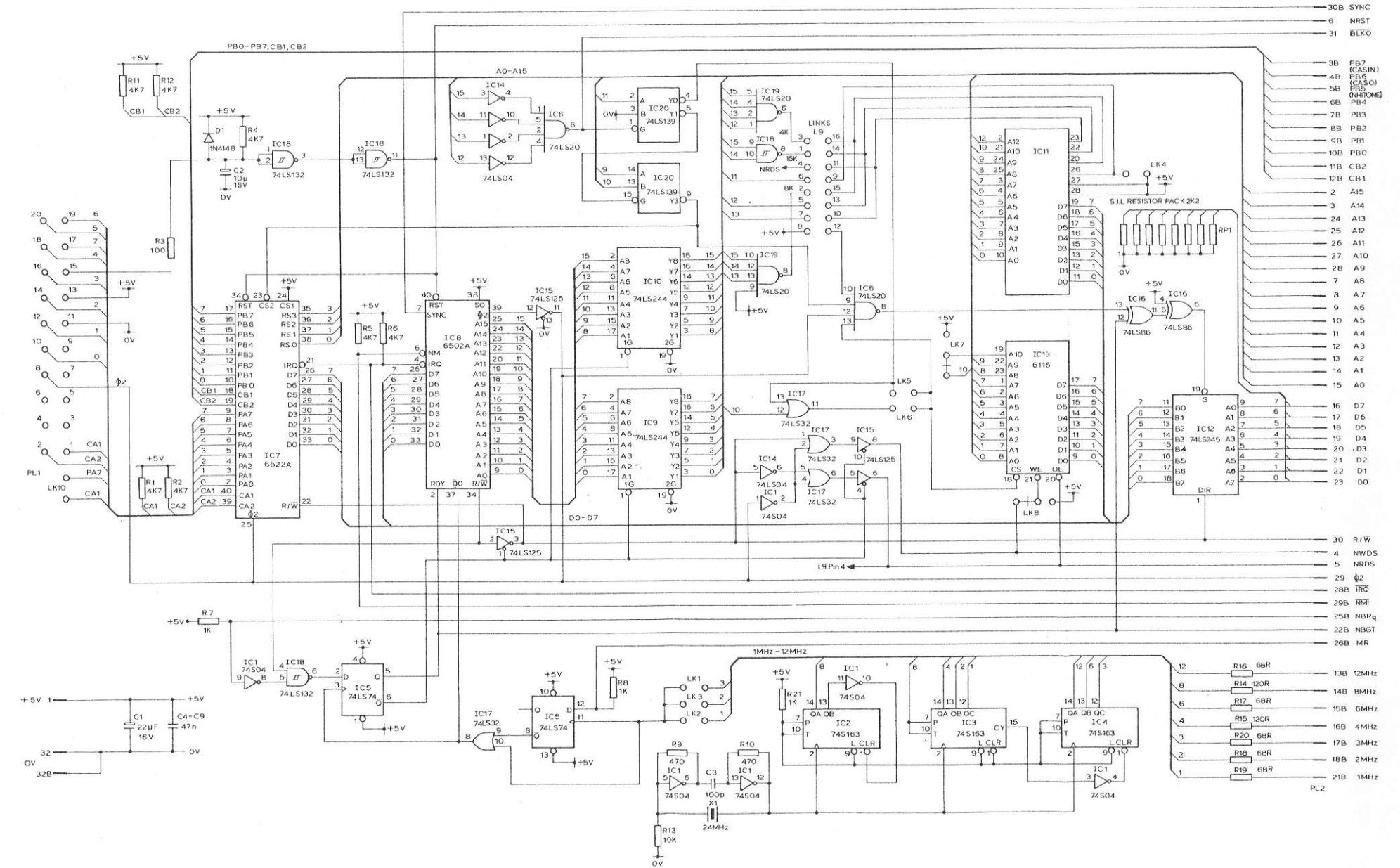


Figure 21 6502A CPU Board Circuit Diagram

# ACORN 32K DYNAMIC RAM BOARD

## UNIT DESCRIPTION

### CONTENTS

Paragraph		Page
1	INTRODUCTION	1
1.1	General	1
1.2	Leading Particulars	1
1.2.1	Mechanical	1
1.2.2	Power Supplies	1
1.2.3	Connections	1
2.	CIRCUIT DESCRIPTION	1
2.1	Dynamic RAM	2
2.2	Memory Address Selection	2
2.2.1	Address Decoding	2
2.2.2	Address Signal Timing	4
2.3	Clock Circuit	5
2.3.1	Acorn Bus Clock Operation	5
2.3.2	Local Oscillator Operation	3
2.3.3	Local Oscillator Setting-Up Procedure	3
2.4	Refresh Circuit	7
2.5	Data Buffer	7
3.	SOFTWARE PARAMETERS	7
3.1	Memory Addressing	7
3.2	Memory Paging	7
4.	32K DYNAMIC RAM BOARD CONNECTIONS	7
4.1	32K Dynamic RAM Board to Acorn Bus	7
5.	PARTS LIST	8
5.1	Mechanical Parts	8
5.2	Electrical Parts	8

### TABLES

Table No.		Page
1	32K DRAM Address Block Allocation	4
2	Clock Selection Links	6

### FIGURES

Figure No.		Page
1	32K Dynamic RAM Board Block Diagram	1
2	32K DRAM Bank and Link Location	2
3	32K DRAM Read, Write and Refresh Timing	3
4	Memory Address Selection	4
5	Acorn Memory Map and DRAM Address Selection	5
6	Address Signal Timing	5

©1982 Acorn Computers Limited, 12 Fulbourn Road, Cherry Hinton, Cambridge.  
Written and Produced for Acorn Computers Limited by Arch Technical Publications Limited,  
St. Lukes House, 5, Walsworth Road, Hitchin, Herts.

## **FIGURES (Cont'd)**

<b>Figure No.</b>		<b>Page</b>
7	Clock Circuit	6
8	32K DRAM Clock Timing	6
9	32K Dynamic RAM Board Circuit Diagram	9

## 1. INTRODUCTION

### 1.1 GENERAL

The 32K Dynamic RAM Board provides 32768 (32K) bytes of Dynamic Random Access Memory (DRAM) for a computer system. The Board is a 100 x 160mm Eurocard and is provided with a 64-way connector, for connection to the Acorn Bus.

The DRAM ICs are configured in two 16K by 8 bit Banks. Links on the Board provide individual address selection of four 8K sections of the DRAM, each section address begins at an even memory Block address. The Board can be partially equipped to provide 16K of DRAM. The Board is also provided with a paging facility.

The DRAM timing may be generated from a single clock input (e.g. 6502 §2) by an on-board Clock Circuit. Alternatively, three external synchronized clock inputs may be used. A Refresh Circuit refreshes the DRAM ICs between external Read or Write cycles, allowing access to the DRAM during any Central Processor Unit (CPU) cycle.

## 1.2 LEADING PARTICULARS

### 1.2.1 Mechanical

Construction : Single Eurocard PCB  
Size : 100mm x 160mm

### 1.2.2 Power Supplies

+5V at 280mA.

### 1.2.3 Connections

Pin connections are given in Section 4.

Connector : Double-sided edge connector to Acorn Bus. TTL signal levels are used, 0V to +0.4V = logic '0', +2.4V = logic '1'.

## 2. CIRCUIT DESCRIPTION

A block diagram of the 32K Dynamic RAM Board is given on Figure 1. Reference should also be made to the circuit diagram, Figure 9. The DRAM Bank and Link location is given on Figure 2.

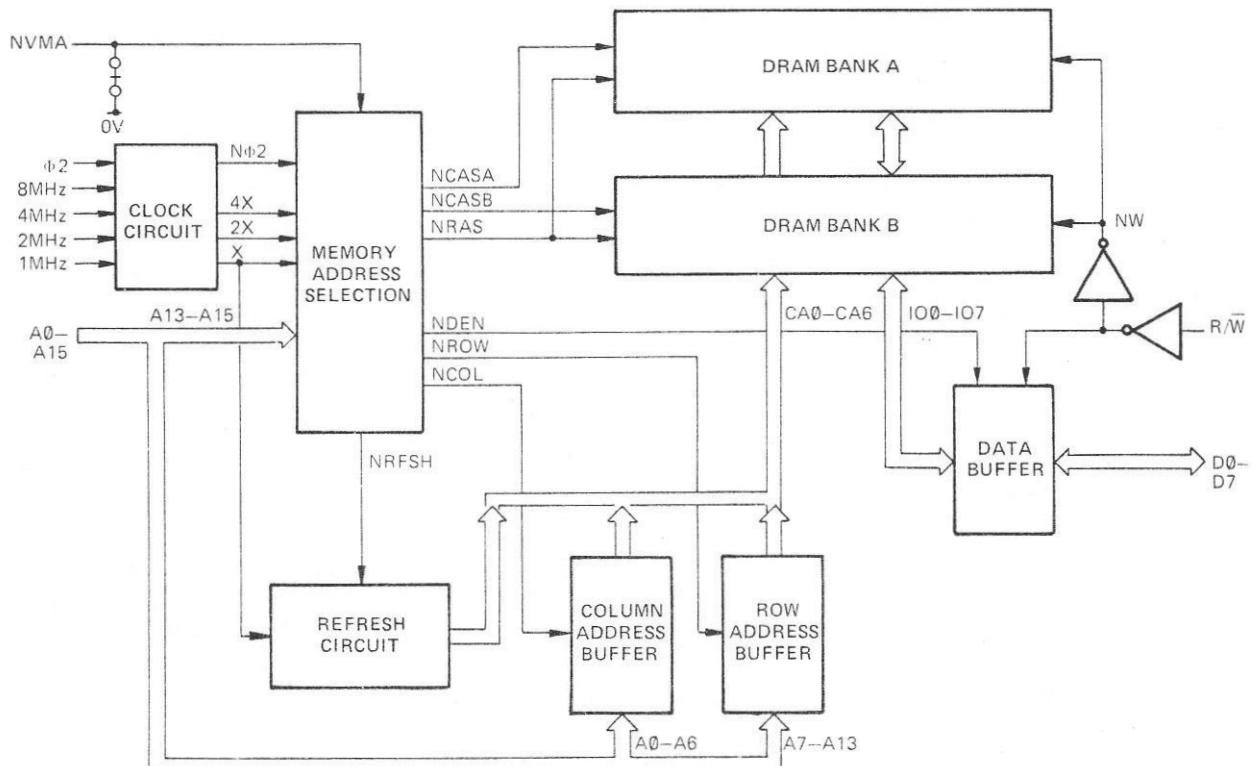
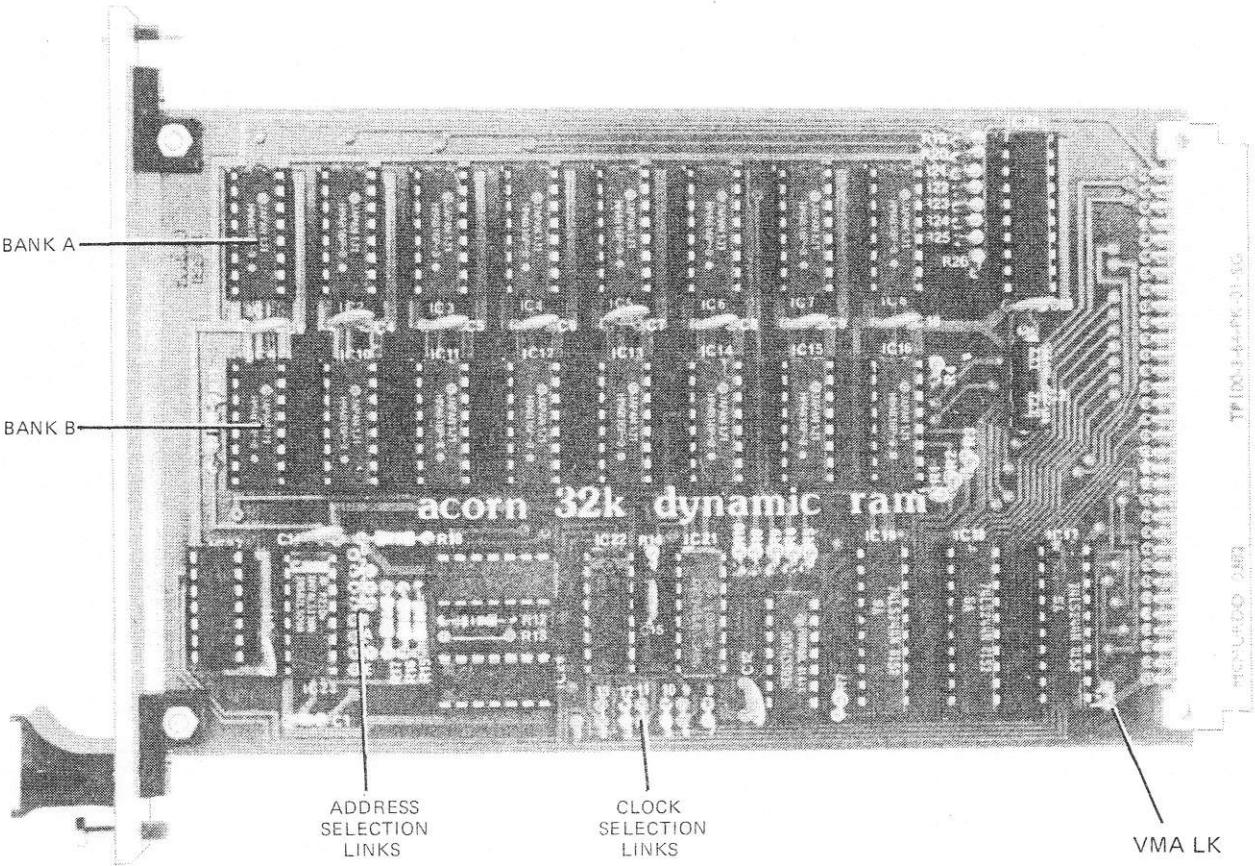


Figure 1. 32K Dynamic RAM Board Block Diagram



**Figure 2. 32K DRAM Bank and Link Location**

## 2.1 DYNAMIC RAM

The DRAM consists of two Banks of 4816 ICs, each IC having 16384 (16K) x 1 bit separately addressable memory locations. The 8 ICs in each Bank provide 16K 8 bit bytes of memory. The memory locations are addressed by seven address lines CA0–CA6, carrying multiplexed row and column addresses. The addresses are strobed into the ICs by the Row Address Strobe (NRAS) and Column Address Strobe (NCAS) signals. A Write (NW) enable signal determines whether a Write or Read operation is required. The DRAM ICs are refreshed by a row address on CA0–CA6 and a NRAS signal, the refresh address is cycled through the 128 row addresses by the Refresh Circuit, refer to para. 2. 4. The DRAM timing is given on Figure 3(a) 2MHz operation, and Figure 3(h) Local Oscillator operation.

## 2.2 MEMORY ADDRESS SELECTION

The Memory Address Selection Circuit decodes the address on the Acorn Bus lines A13—A15 to detect DRAM addresses, refer to Figure 4. The Row and Column Address Buffers are enabled in sequence by the Row Address Select (NROW) and Column Address Select (NCOL) signals, so that when a DRAM address

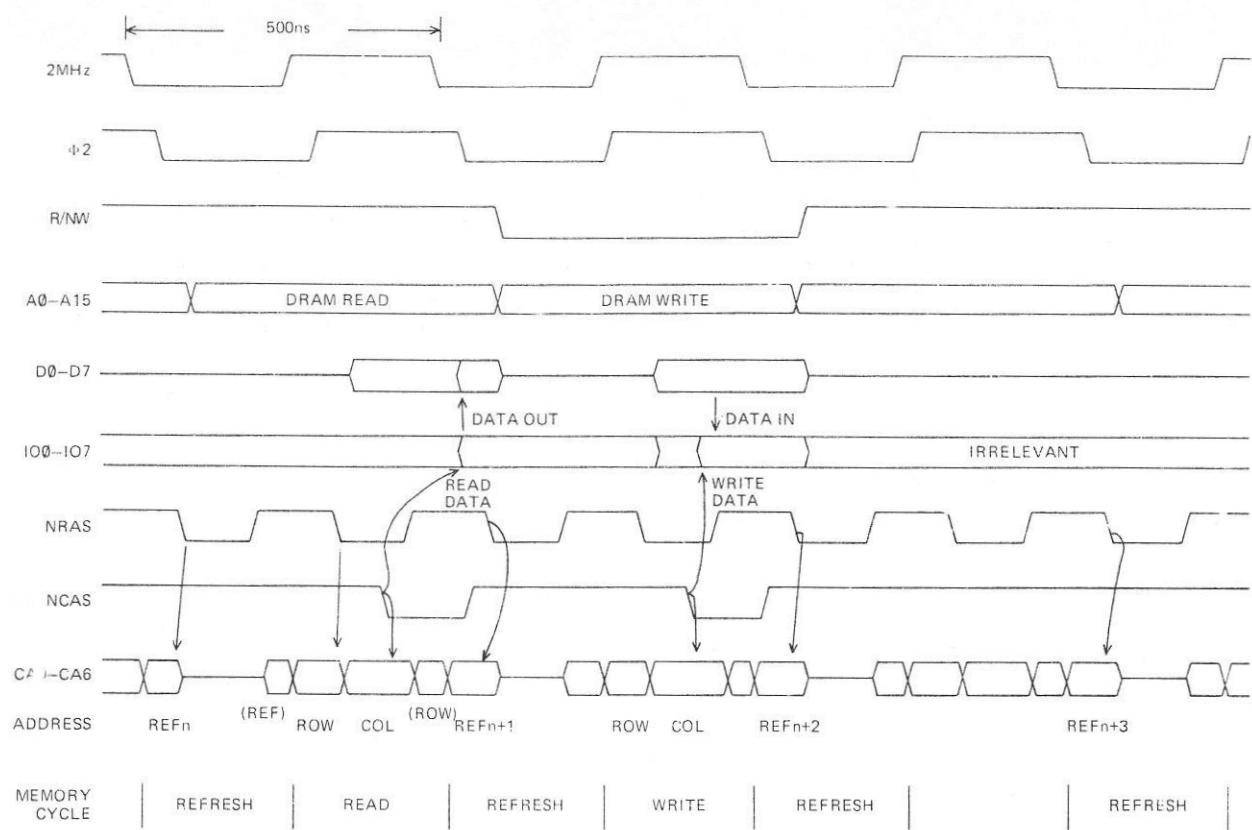
is detected the appropriate NCASA or NCASB signal is generated. The two and column addresses are output from the Buffers onto the DRAM IC address lines A0—A6.

The Memory Address Selection Circuit also provides the NRAS and NCAS signals to strobe the addresses into the ICs. A Data Enable signal (LADEN) is set low to the Data Buffer to enable the transfer of data to or from the DRAM ICs.

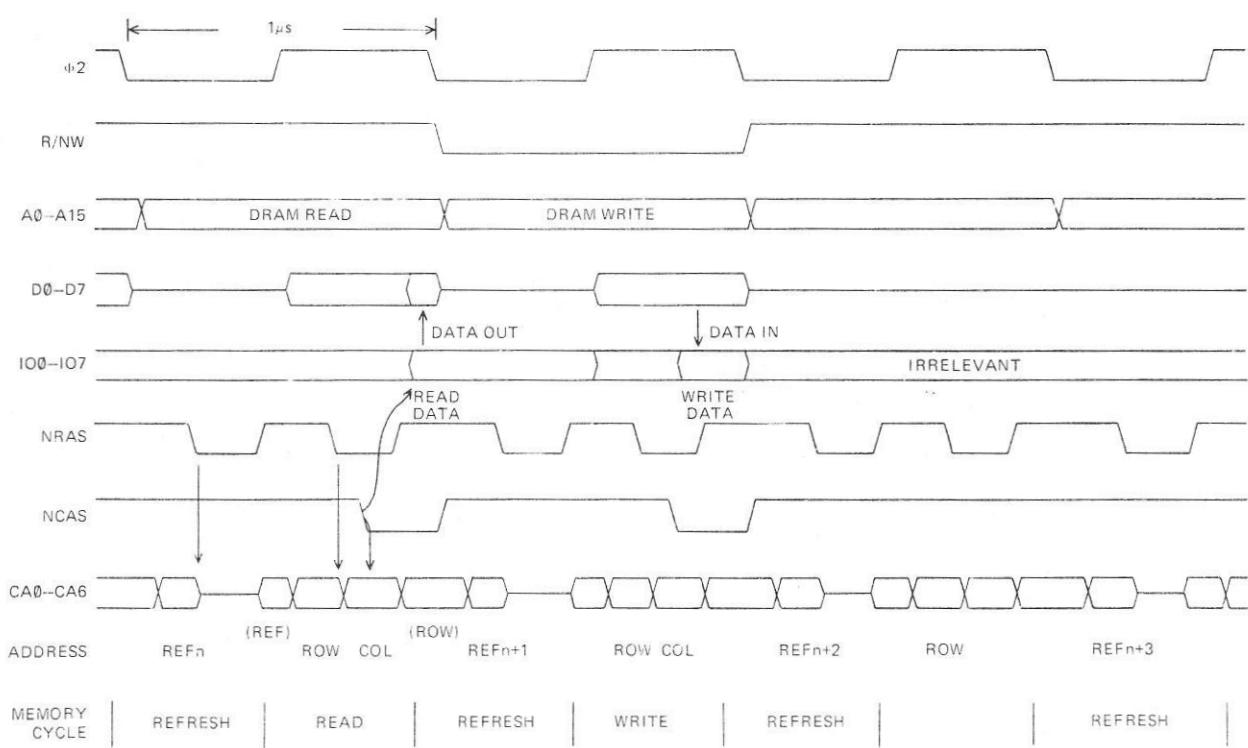
### 2.2.1 Address Decoding

The Acorn Bus address lines A15, A14 and A13 are decoded by IC23, during the CPU Phase 2 (\$2) clock time. A low on the Valid Memory Address (NVMA) Acorn Bus signal enables IC23, when the Board is configured for memory paging, track. Link LK is cut. Otherwise, IC23 is enabled by the 0V connected to IC23/5 by Link LK. Enable G1, IC23/6 is connected to +5V. The inverted \$2 signal at IC23/4 enables address decoding (provided NVMA is low), to provide a low on the selected output of IC23, refer to Table 1.

The DRAM addresses are determined by the configuration of the Address Selection Links. The normal



(a) 2MHz Operation



(b) Local Oscillator Operation

Figure 3. 32K DRAM Read, Write and Refresh Timing

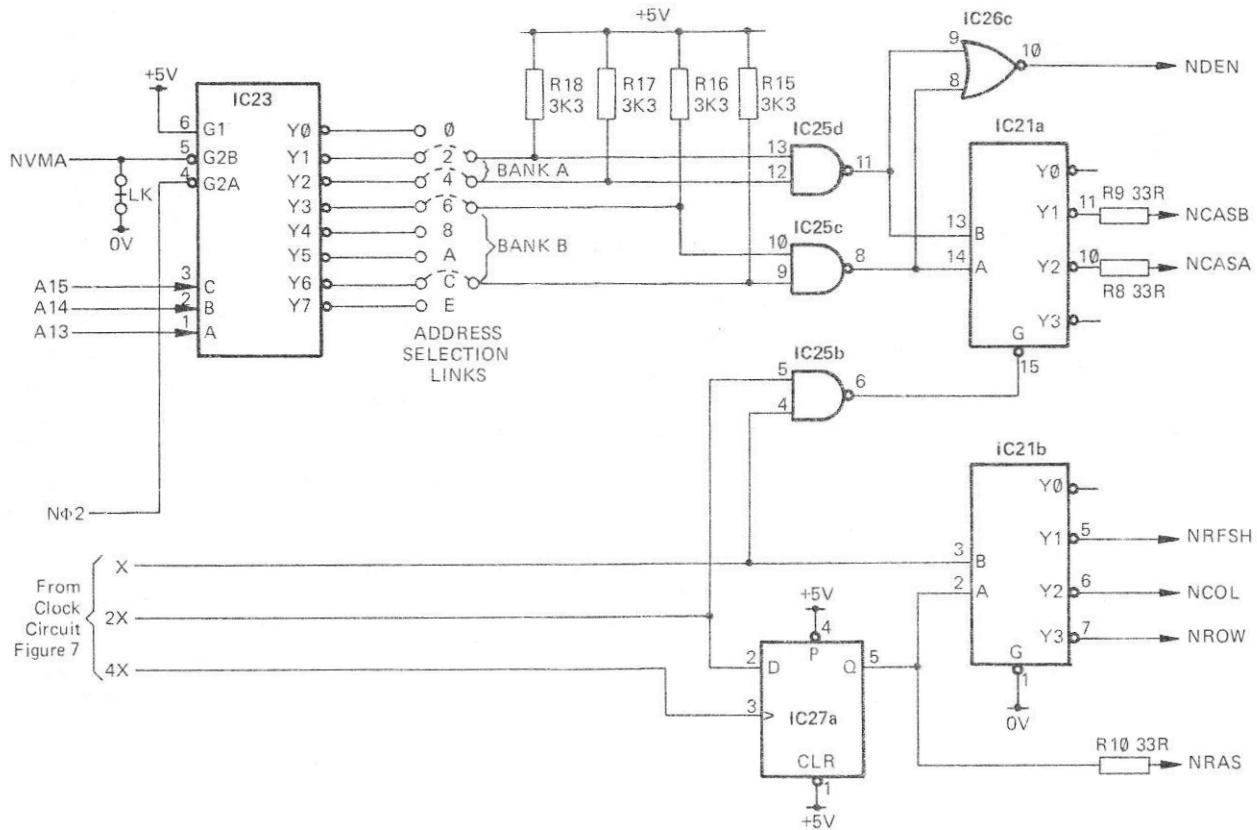


Figure 4. Memory Address Selection

§2	NVMA	A15	A14	A13	Selected Output '0'	Block Addressd (hex)	Link No.
0	X	X	X	X	None	None	
X	1	X	X	X	None	None	
1	0	1	1	1	Y7	E,F	E
1	0	1	1	0	Y6	C,D	C
1	0	1	0	1	Y5	A,B	A
1	0	1	0	0	Y4	8,9	B
1	0	0	1	1	Y3	6,7	6
1	0	0	1	0	Y2	4,5	4
1	0	0	0	1	Y1	2,3	2
1	0	0	0	0	Y0	0,1	0

Table 1. 32K DRAM Address Block Allocation

Acorn Computer configuration is given on the Memory Map Figure 5, alternatives can be configured by removing the links fitted and soldering new links to the required memory selection outputs, refer to Figure 5. Note that each link selects 8K (two Blocks) of memory addresses for one section of the DRAM.

Because the DRAM addresses are decoded to provide 4 starting points for RAM, and there are only 2 banks of IC's on the board, there is some redundancy in the address decoding. This means that the links associated with one bank must not be made so as to leave a gap of 1, 3 or 5 positions between them, If this restriction is

not observed then one section of memory will respond to two address ranges, and data will be overwritten.

The 16K DRAM option can be equipped with 8 DRAM IC; in Bank B. The Address Selection Links are then required to give DRAM addresses 8000 (hex) to BFFF (hex), refer to Figure 5(c),

## 2.2 Address Signal Timing

When a DRAM address is decoded, NAND gates IC25c and IC25d (refer to Figure 4) give a high on the A or B input to IC21a. Any detected DRAM address gives a high on an input to NOR gate IC26c, to generate the NDEN signal low to the Data Buffer. NAND gate IC25b output low enables IC21a, to generates the appropriate NCASA or NCASB signal at the Column Address Strobe time (X and 2X high), refer to Figure 3.

The clock signals from the Clock Circuit, refer to para 2.3, generates the NRAS, NROW, NCOL and NRFSH signals as shown on Figure 6. The X and 2X clock signals also enable IC21a to generate NCAS signals as described above. The NROW, NCOL and NRFSH signals enable the outputs of the Row, Column and Refresh Address Buffers respectively, onto the DRAM address lines CA0—CA6, The row and refresh addresses are strobed by the NRAS signal negative edge, When a DRAM address is decoded, the column address is strobed by the appropriate NCASA or NCASB signal negative edge.

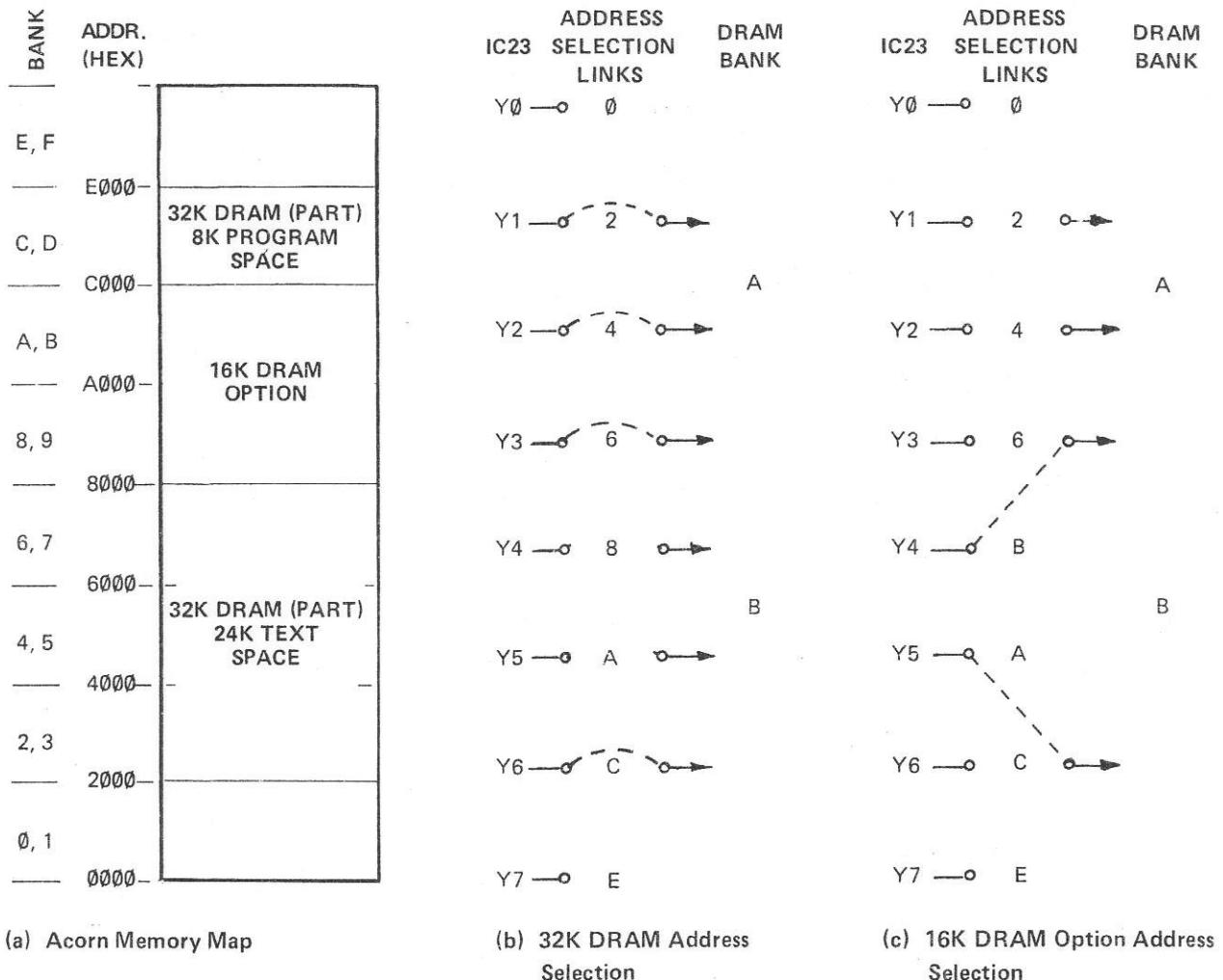


Figure 5. Acorn Memory Map and DRAM Address Selection

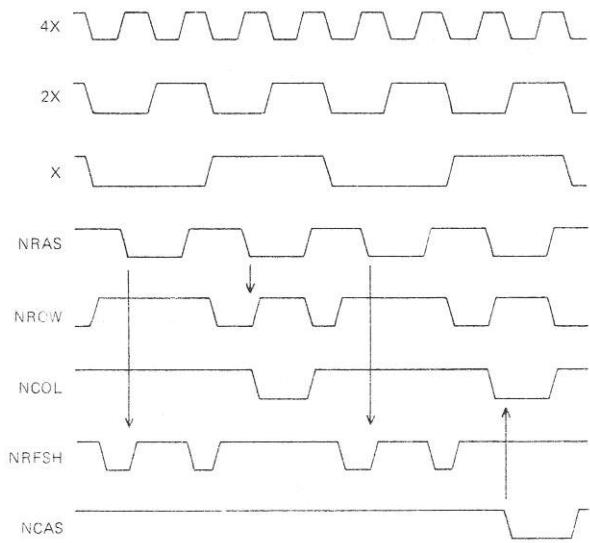


Figure 6. Address Signal Timing

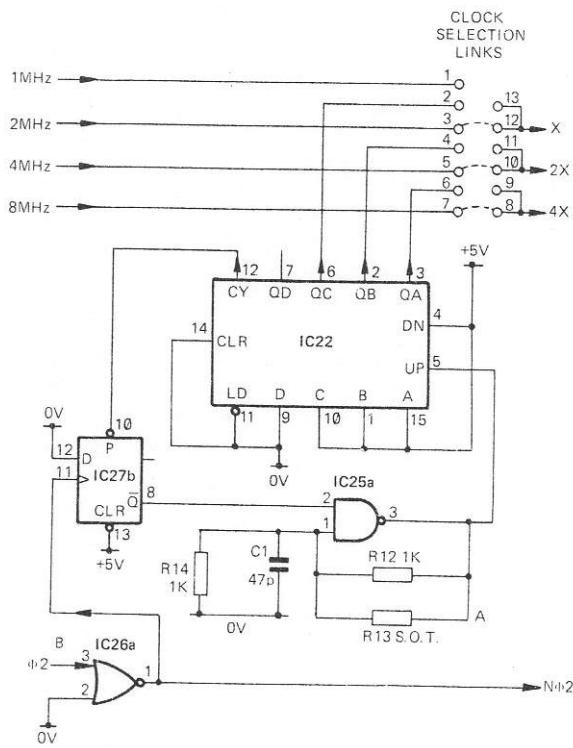
### 2.3 CLOCK CIRCUIT

The Clock Circuit, refer to Figure 7, provides three synchronized signal outputs to the Memory Address Selection Circuit, either from the clock signals on the Acorn Bus (e.g. System 5) or from a local oscillator that is synchronized by the Phase 2 (\$2) clock from a CPU (e.g. 6502 CPU).

Clock Selection Links are provided, refer to Figure 2, to select the appropriate clock signals as shown in Table 2. The clock signals are X, 2X and 4X, where X is the selected operating frequency: 1MHz or 2MHz.

#### 2.3.1 Acorn Bus Clock Operation

The 32K DRAM Board is supplied to operate at 2MHz from the Acorn Bus clock signals, the appropriate links in the Clock Selection Links are soldered, in the positions indicated in Table 2, to connect the 8MHz, 4MHz and 2MHz Clock signals to the appro-



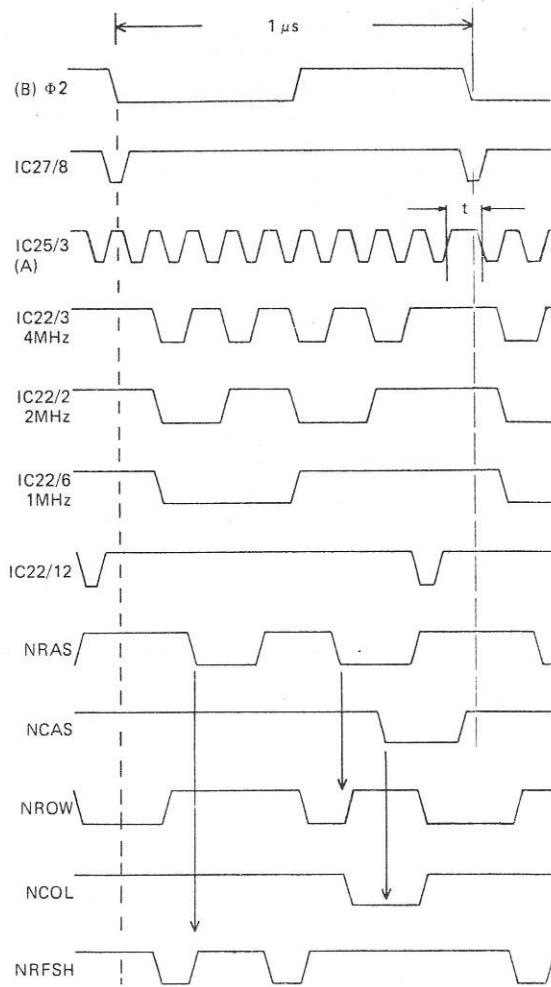
**Figure 7. Clock Circuit**

priate 4X, 2X and X outputs. The local oscillator circuit is not used in this case. To operate at 1 MHz the links must be removed and the required links soldered in place.

### 2.3.2 Local Oscillator Operation

To operate from the local oscillator the appropriate Clock Selection Links must be connected, refer to Table 2. The negative edge of the §2 signal from the CPU is inverted by NOR gate IC26a to clock D-type bistable IC27b, refer to Figure 7. The Q output of IC27b goes high since the D input is connected to 0V. The high on Q enables the local oscillator, NAND

gate IC25a, R12, R13, R14 and C1. The oscillator output clocks binary counter IC22, to produce the 4MHz, 2MHz and 1MHz Clock signals, refer to timing diagram, Figure 8.



**Figure 8. 32K DRAM Clock Timing  
(Local Oscillator/§2 Synchronization)**

INPUT		CLOCK SELECTION LINKS	OUTPUT SIGNAL
ACORN BUS	LOCAL OSC.		
8MHz	—	7 to 8	4X
4MHz	—	5 to 10	2X
2MHz	—	3 to 12	X
4MHz	—	5 to 9	4X
2MHz	—	3 to 11	2X
1MHz	—	1 to 13	X
—	4MHz	6 to 9	4X
—	2MHz	4 to 11	2X
—	1MHz	2 to 13	X

**Table 2. Clock Selection Links**

When the binary counter reaches a count of F (hex) the CY output goes low to the preset input of D-type bistable IC27b. IC27b Q output goes low to load 7 (hex) into the data inputs of IC22, and also to inhibit the local oscillator output from IC25a. The next §2 negative edge clocks IC27b to restart the local oscillator and count up IC22 from 7, thus synchronizing the local oscillator.

### 2.3.3 Local Oscillator Setting-Up Procedure

To set up the local oscillator the select on test resistor (R13) is used to adjust the frequency of the clock circuit. The value of R13 should be in the order of 1kOhm.

The following equipment is required:

- Dual-beam oscilloscope
- Variable +5V power supply
- Frequency Generator

Connect the oscilloscope as follows (refer to Figure 7):

- Channel A to right hand end of R13 or IC25/3(A).
- Channel B to S2, IC26/3(B).

Set the oscilloscope to:

- Amplitude 2V/cm
- Timebase 100ns/cm
- Trigger Channel B

Connect the power supply to the board edge connector as follows:

- +5V to Side A pin 1
- 0V to Side A pin 32

Set the power supply output to +5.0V.

Connect the frequency generator output to the board edge connector Side A pin 29 and 0V to Side A pin 32. Adjust the frequency generator output to 1.00MHz, amplitude +3.5V w.r.t. 0V.

Ensure that the length of the part "t" of the channel A trace in Figure 8 is 120ns +20ns at 50% amplitude.

-0ns

Adjust the power supply input to +4.5V and ensure that "t" is approximately 60ns.

Substitute different values of resistor R 13 until the above measurements are obtained.

#### 2.4 REFRESH CIRCUIT

The Refresh Circuit consists of a dual 4 bit binary counter IC20 and an octal buffer IC19. The counter is configured to produce the 128 refresh addresses on the seven DRAM IC address lines CA0—CA6. The X clock signal (1MHz or 2MHz) from the Clock Selection Links clocks the counter on each negative edge. The octal buffer is enabled by the NREFSH signal low from the Memory Address Selection Circuit, so that a row address is loaded onto CA0—CA6 to be strobed into the DRAM ICs by the NRAS signal, refer to timing diagram, Figure 3. The DRAM IC refresh cycle is thus completed in 256 $\mu$ s (1MHz operation) or 128 $\mu$ s (2MHz operation).

#### 2.5 DATA BUFFER

The Data Buffer is an octal buffer IC24. The buffer is enabled for data transmission by the NDEN signal low from the Memory Address Selection Circuit, refer to para 2.2.2. The direction of transmission of data is

determined by the R/W signal on the Acorn Bus. The R/W signal is inverted by IC26b to select output from the 32K Dynamic RAM Board for a CPU Read operation, R/W high. R/W is low for Write to DRAM operation.

### 3. SOFTWARE PARAMETERS

#### 3.1 MEMORY ADDRESSING

The 32K Dynamic RAM Board is provided with soldered links which give the Dynamic RAM the addresses 2000 to 7FFF (hex) and C000 to DFFF (hex), Acorn Memory Blocks 2 to 7, C and D. If alternative addresses are required, refer to para 2.2.1.

#### 3.2 MEMORY PAGING

The Link LK is supplied with a track connection on the 32K Dynamic RAM Board. When memory paging is required the track must be cut, the Valid Memory Address (NVMA) signal low can then be used to select the memory on the Board.

### 4. 32K DYNAMIC RAM BOARD CONNECTIONS

#### 4.1 32K DYNAMIC RAM BOARD TO ACORN BUS

Pin	Mnemonic	Meaning	I/O
Side A			
1	+5V	+5V Supply	I
2	A15		
3	A14		
7	A8		
8	A7		
9	A6		
10	A5	Address Lines	I
11	A4		
12	A3		
13	A2		
14	A1		
15	A0		
16	D7		
17	D6		
18	D5		
19	D4	Data Lines	I/O
20	D3		
21	D2		
22	D1		
23	D0		
24	A13		
25	A12		
26	A11	Address Lines	I
27	A10		
28	A9		

Pin	Mnemonic	Meaning	I/O
Side A 29	$\Phi_2$	Phase Two Clock	I
30	R/NW	Read/Write	I
32	0V	0V Supply	I

Pin	Mnemonic	Meaning	I/O
Side B 14	8MHz		
16	4MHz		
18	2MHz		
21	1MHz		
24	NVMA	System Clocks	I
32	0V	Valid Memory Address 0V Supply	I

## 5. PARTS LIST

### 5.1 MECHANICAL PARTS

ITEM	DESCRIPTION	VALUE	QTY	PART NO.
	PCB 200.100 Issue 4		1	
	PCB Terminals Vero 20-2137D (for R13)		2	
	20 pin IC Socket		4	
	16 pin IC Socket		19	
	14 pin IC Socket		4	

### 5.2 ELECTRICAL PARTS

ITEM	DESCRIPTION	VALUE	QTY	PART NO.
C1	Capacitor, Disc Ceramic	47pF	1	
C2	Capacitor, Electrolytic	22 $\mu$ F 16V	1	
C3...C14	Capacitor	47nF	12	
C15	Capacitor, Disc Ceramic	100pF	1	
PL1	Connector 64-way Plug (Right angle solder tails to DIN 41612)		1	
IC1...IC16	Integrated Circuit	4816-7 (for 1MHz)	16	
or	Integrated Circuit	4816-3 (for 2MHz)		
IC17...IC19	Integrated Circuit	74LS244	3	
IC20	Integrated Circuit	74LS393	1	
IC21	Integrated Circuit	74LS139	1	
IC22	Integrated Circuit	74LS193	1	
IC23	Integrated Circuit	74LS138	1	
IC24	Integrated Circuit	74LS245	1	
IC25	Integrated Circuit (Texas device only)	74LS132	1	
IC26	Integrated Circuit	74LS02	1	
IC27	Integrated Circuit	74LS74	1	
R1...R11	Resistor	33Ohm CR25	11	
R12	Resistor	1kOhm	1	
R13	Resistor	Select on test*	1	
R14	Resistor	100Ohm	1	
R15...R18	Resistor	3.3kOhm	4	
R19...R26	Resistor	68Ohm CR25	8	

\*refer to para 2.3.3

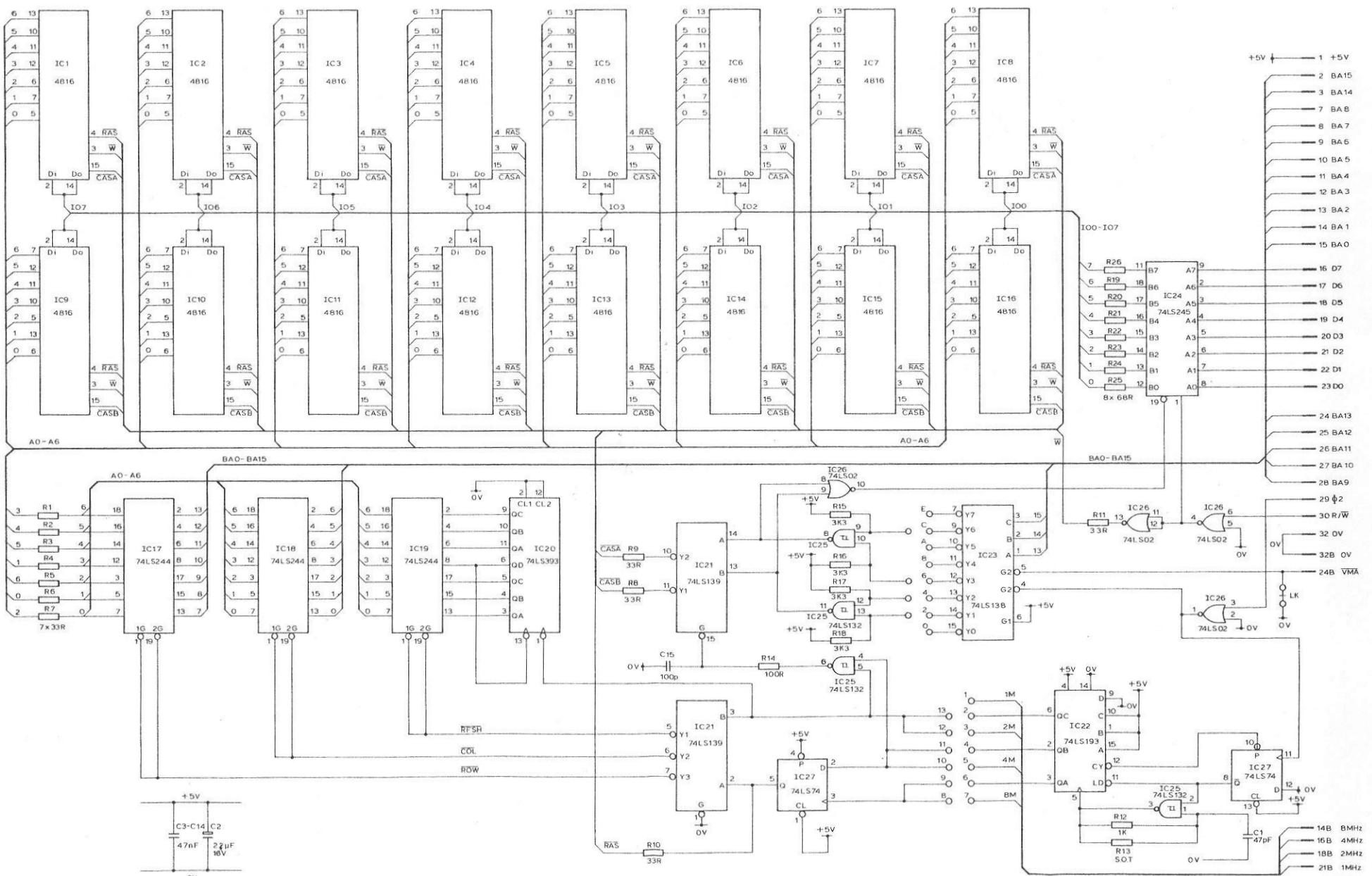


Figure 9. 32K Dynamic RAM Board Circuit Diagram

# ACORN 80 x 25 VDU INTERFACE BOARD

## UNIT DESCRIPTION

### CONTENTS

Paragraph		Page
1.	INTRODUCTION	1
1.1	General	1
1.2	Principles of Operation	1
1.3	Leading Particulars	2
1.3.1	Mechanical	2
1.3.2	Power Supplies	2
1.3.3	Connections	2
2.	CIRCUIT DESCRIPTION	2
2.1	Address Decoding	2
2.1.1	RAM Selection	2
2.1.2	VDU Controller Selection	4
2.2	VDU Controller	4
2.2.1	Register Functions	5
2.2.2	System Interface Signals	6
2.2.3	Video Interface Signals	6
2.2.4	Memory Address Signals	6
2.2.5	Light Pen	6
2.2.6	Controller Read/Write	6
2.2.7	Horizontal Scan	7
2.2.8	Vertical Scan	7
2.2.9	Scan interlace	7
2.2.10	Screen Scrolling	8
2.3	VDU Clock Circuit	8
2.3.1	2MHz Clock Signal Generation	8
2.3.2	Load Signal Generation	8
2.3.3	12MHz' Clock Signal Generation	9
2.3.4	VDU Clock Setting-Up Procedure	9
2.4	Character RAM	9
2.4.1	System Read or Write Operation	9
2.4.2	VDU Controller Read Operation	10
2.5	Character Generator	11
2.5.1	Character Latch	11
2.5.2	Character ROM	11
2.5.3	Dot Output	11
2.6	Video Output	12
2.6.1	Dot Display	12
2.6.2	Cursor Display	12

© 1982 Acorn Computers Limited, 12 Fulbourn Road, Cherry Hinton, Cambridge.  
Written and Produced for Acorn Computers Limited by Arch Technical Publications Limited,  
St. Lukes House, 5, Walsworth Road, Hitchin, Herts.

## CONTENTS (Cont'd)

Paragraph		Page
	2.6.3 Sync Signals	12
	2.6.4 Display Inversion	12
3.	SOFTWARE PARAMETERS	13
3.1	VDU Interface Addressing	13
3.1.1	Character RAM	13
3.1.2	VDU Controller	13
3.1.3	Paging Facility	13
3.2	VDU Controller	13
3.2.1	Register Addressing	13
3.2.2	Mode Control	13
3.2.3	Display Configuration	14
4.	80 x 25 VDU INTERFACE BOARD CONNECTIONS	14
4.1	VDU Board to Acorn Bus	14
4.2	VDU Board to VDU	14
5.	PARTS LIST	15
5.1	Mechanical Parts	15
5.2	Electrical Parts	15

## TABLES

Table No.		Page
1	VDU Board Address Decoding	4
2	VDU Controller Registers	4
3	VDU Clock Links	8
4	Register Settings	14

## FIGURES

Figure No.		Page
1	80 x 25 VDU Interface Board Block Diagram	1
2	VDU Scan (1 Frame 50Hz)	2
3	Character Dot Matrix	2
4	80 x 25 VDU Interface Board	3
5	Address Decoding Circuit	3
6	Interlace and Skew Register (R8) Bit Significance	5
7	VDU Controller Write and Read	6
8	VDU Controller Scan Signals	7
9	VS Signal Timing Interlace-Sync Mode	7
10	VDU Clock Circuit	8
11	VDU Clock Timing ( $\$2 = 1 \text{ MHz}$ )	9
12	Character RAM Circuit	9
13	Character RAM Timing	10
14	Character Generator Circuit	11

## **FIGURES (Cont'd)**

<b>Figure No.</b>		<b>Page</b>
15	Character Generator Timing	11
16	Video Output Circuit	12
17	Composite Video Output Signal	13
18	80 x 25 VDU Interface Board Circuit Diagram	17

## 1. INTRODUCTION

### 1.1 GENERAL

The 80 x 25 VDU Interface Board provides the interface between a Microcomputer System, such as the Acorn System 5 and a monochrome Visual Display Unit (VDU). A block diagram of the VDU Interface Board is given on Figure 1.

The VDU Board receives characters to be displayed in 8 bit bytes on the Acorn Bus. The characters are stored in the 2048 (2K) byte Random Access Memory (RAM). The VDU Controller IC reads the characters to be displayed from the Character RAM into a Read Only Memory (ROM). The ROM is programmed to convert the characters from the code used in the System, i.e. American Standard Code for Information Interchange (ASCII), to a dot output of '1's and '0's. The ROM output is converted to a serial dot pattern which produces the required characters on the VDU screen.

The VDU Controller also provides the Horizontal Synchronization (HS) and Vertical Synchronization (VS) signals, which are combined with the dot output to generate the composite video signal to the VDU.

The VDU Board can be operated directly from the Acorn Bus 6502A 2MHz and 12MHz clock signals. Alternatively, an on-board clock circuit may be used

to generate the clock signals from the Phase 2 (\$2) clock in a 1MHz or 2MHz CPU system (e.g. 6809).

Address Selection Links on the VDU Board are provided to allow two possible Character RAM address locations, either Block 1 1000 to 17FF hexadecimal (hex) or Block F F000 to F7FF (hex). The VDU Controller addresses are 1840 and 1841 (hex) for System 5 or E840 and E841 (hex) for 6809. Note that addresses 1840 to 187F (hex) or E840 to E87F (hex) must be reserved for the VDU Board, since the addresses are not completely decoded.

The Character RAM may be accessed at any CPU \$2 clock time by a Read or Write operation, without affecting the video output.

### 1.2 PRINCIPLES OF OPERATION

The characters to be displayed are written into the locations in the Character RAM that correspond to the display positions on the VDU screen. The Character RAM locations are addressed by the VDU Controller to obtain the characters during the Phase 1 (\$1) CPU clock time. The characters are loaded into (or read from) the Character RAM locations during the Phase 2 (\$2) clock time. The VDU Controller is used in the consecutive binary address mode. In this mode each character display location is identified by a binary number, starting with 0 in the top left hand display position, refer to Figure 2.

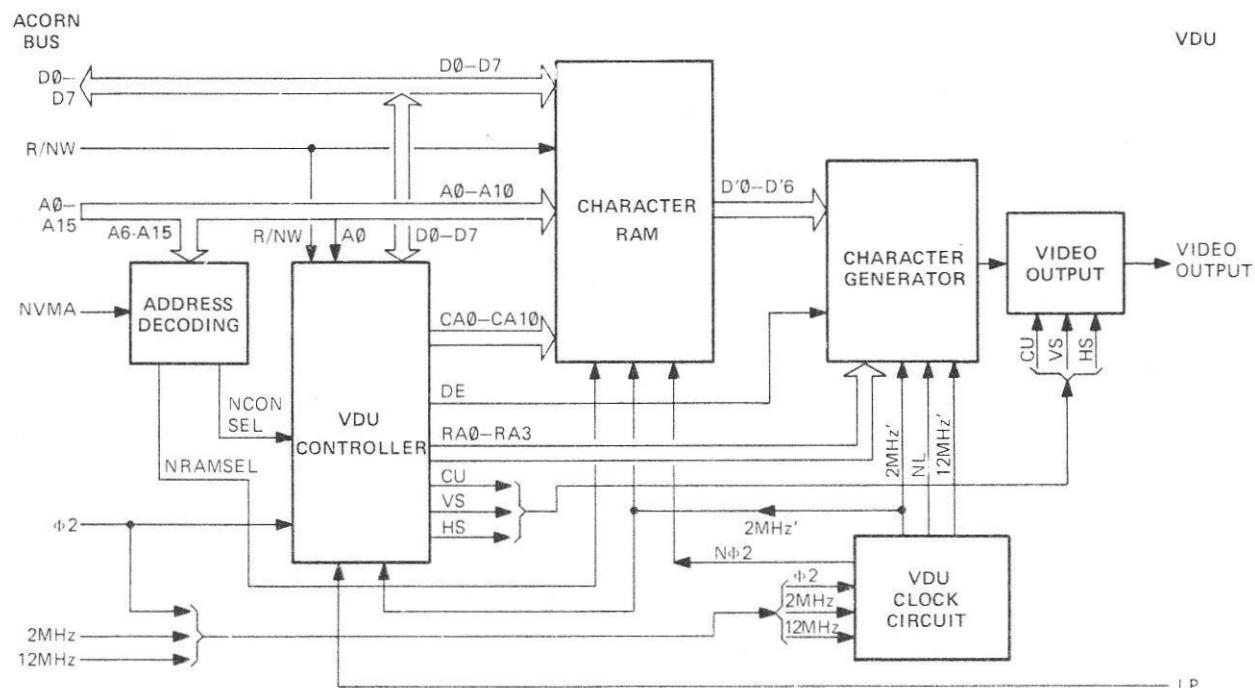
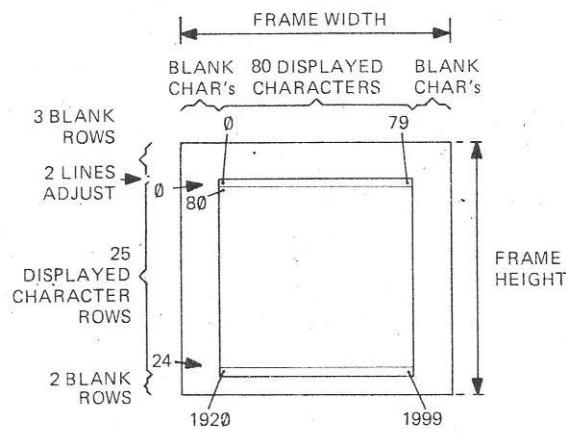


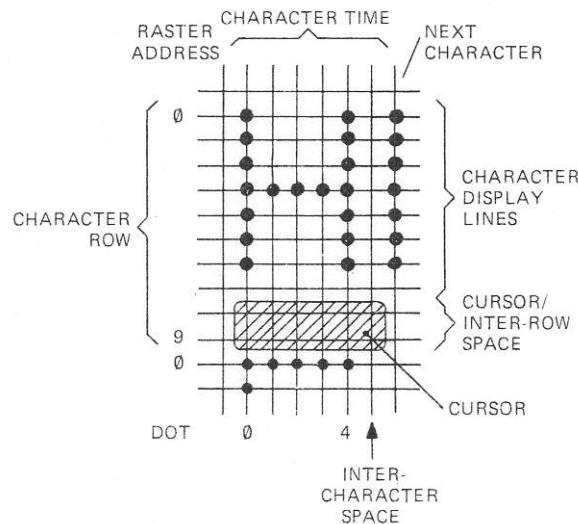
Figure 1. 80 x 25 VDU Interface Board Block Diagram



Note: Addresses shown are decimal

**Figure 2. VDU Scan (1 Frame 50Hz)**

The character timing and the number of scan lines per character row are programmable. The characters are displayed as a dot matrix on the VDU screen. Figure 3 shows an example of a letter 'H' displayed as a  $5 \times 8$  dot matrix. A character time of  $0.5\mu s$  with a dot clock of 12MHz is used to produce the five character dots plus the inter-character space. In the example, a scan of 10 lines per character row gives the 8 scan lines for the character, plus two inter-character scan lines for vertical spacing or the Cursor. The Cursor may be displayed as an underline or a character block. The Cursor is positioned by Writing the character number to the Cursor Position Register in the VDU Controller.



**Figure 3. Character Dot Matrix**

### 1.3 LEADING PARTICULARS

#### 1.3.1 Mechanical

Construction : Single Eurocard printed circuit board.  
Size : 100mm x 160mm

#### 1.3.2 Power Supplies

$+5V \pm 5\%$  at 450mA typically.

#### 1.3.3 Connections

Pin connections are given in Section 4.

Connectors : Double sided edge connector to Acorn Bus, TTL signal levels are used,  $0V$  to  $0.4V$  = logic '0',  $>=+2.4V$  = logic '1'.

5 way connector for ribbon cable to connector on the front panel. Composite video output at  $1V$  into  $75\Omega$ . Light Pen input, TTL.

## 2. CIRCUIT DESCRIPTION

Reference should be made to the circuit diagram, Figure 18 in conjunction with this description. The Character ROM and Link Location is given on Figure 4.

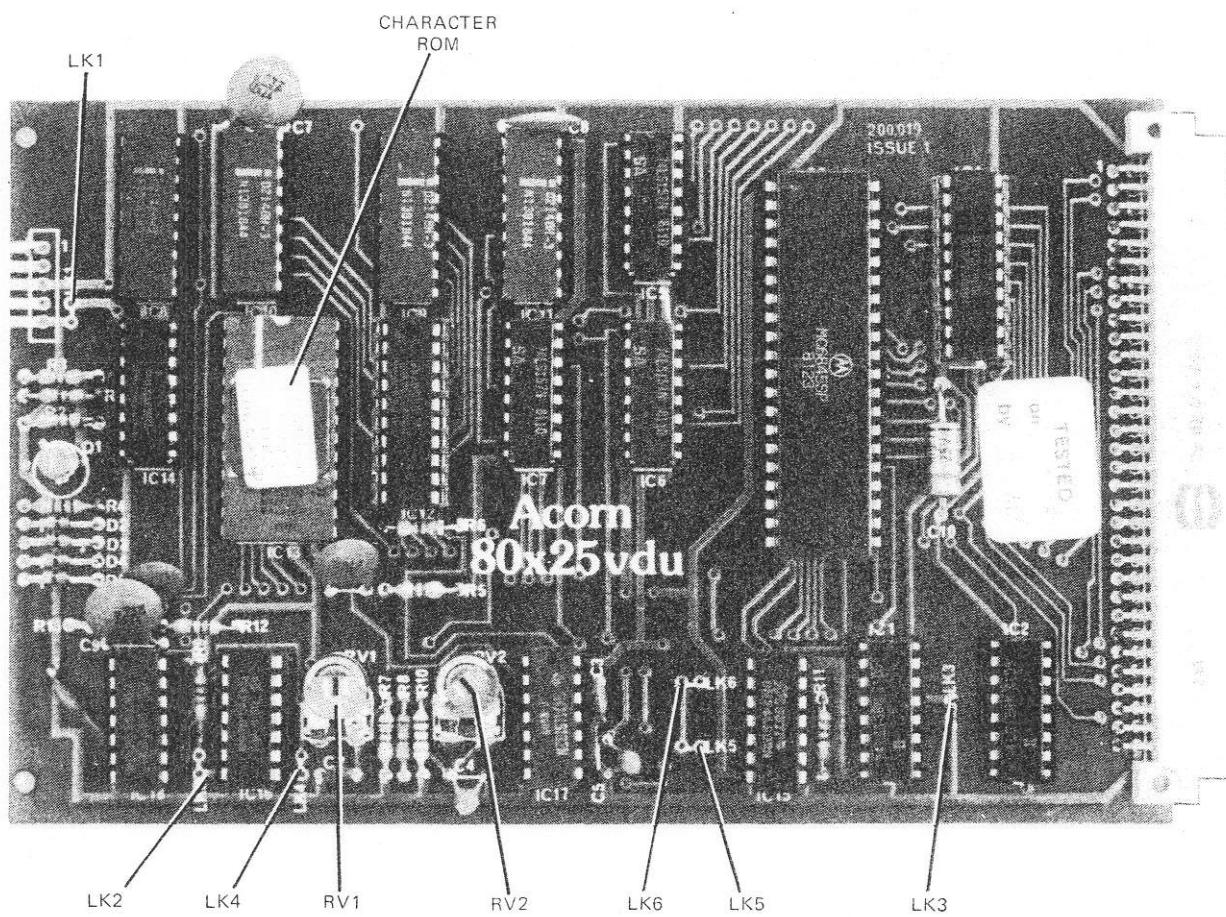
### 2.1 ADDRESS DECODING

The Acorn Bus address lines A6 to A15 are decoded by IC1 and IC2 to select the Character RAM or the VDU Controller, refer to Figure 5. The Valid Memory Address (NVMA) signal may be used for paging the VDU Board in the System memory; to use this facility the track link LK3 must be cut.

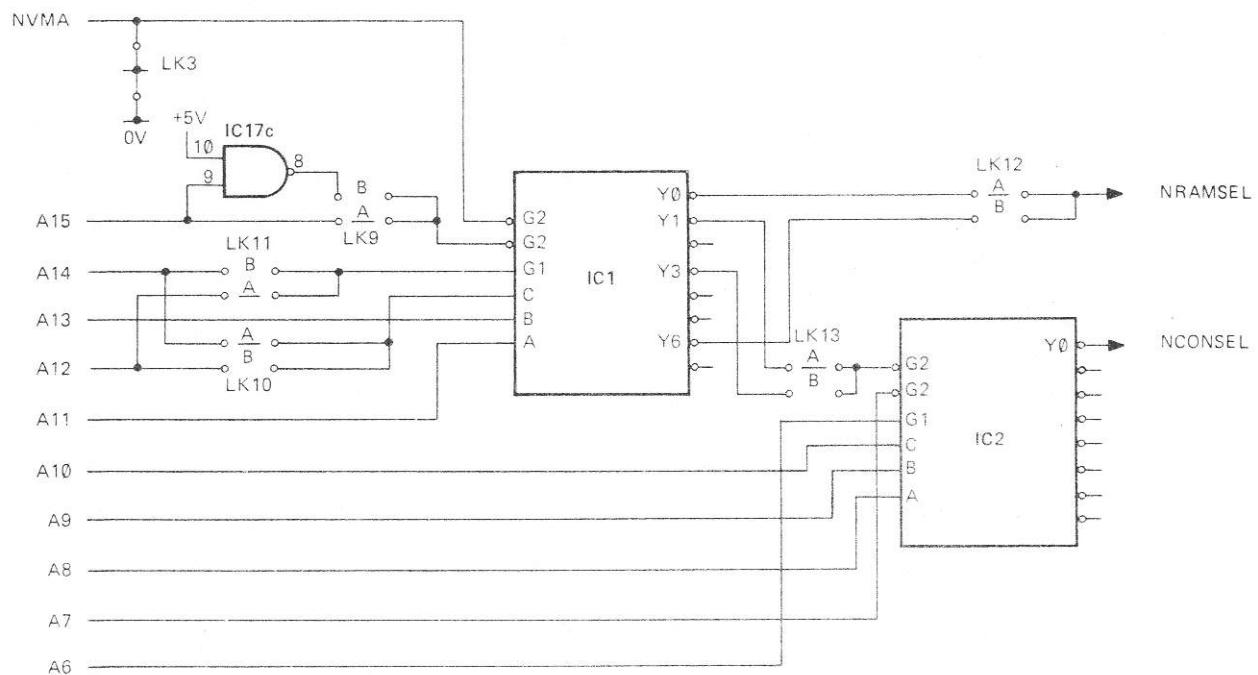
#### 2.1.1 RAM Selection

The address lines A11 to A.15 are decoded by IC1 when it is enabled by either the NVMA signal low, or  $0V$  via LK3. The address Block for the RAM is selected by Links LK9, LK10 and LK11, refer to Table 1. The VDU Board is supplied with the links in position A to give addresses in Block 1. To select addresses in Block E the links in position A must be removed and links soldered in position B.

The appropriate outputs of IC1 are similarly selected by the positions of LK12 and LK13. Outputs Y0 or Y6 generate signal NRAMSEL low to select the Character RAM. Outouts Y1 or Y3 enable IC2 to select the VDU Controller.



**Figure 4. 80 x 25 VDU interface Board**



**Figure 5. Address Decoding Circuit**

Links	NVMA	A						IC1/Y	Address (Hex)	Select
		15	14	13	12	11	0			
	1	X	X	X	X	X	1	1	1	
A	0	0	0	0	1	0	0	1	1	1000-17FF
B	0	1	1	1	1	0	1	1	1	F000-F7FF
A	0	0	0	0	1	1	1	0	1	1800-18FF
B	0	1	1	1	0	1	1	1	0	E800-E8FF

(a) RAM Select

Links	IC1		A				IC2 Y0	Address (Hex)	
	Y1	Y3	10	9	8	7	6		
A	0	X	0	0	0	0	1	0	1840-187F
B	X	0	0	0	0	0	1	0	E840-E87F

(b) VDU Controller Select

X = irrelevant

Table 1. VDU Board Address Decoding

### 2.1.2 VDU Controller Selection

The selected Y1 or Y3 output of IC1 enables IC2, which decodes Acorn Bus address lines A6—A10 to generate the NCONSEL signal low and select the VDU Controller for Read or Write operations. Note

that the VDU Controller address is not fully decoded, so that any addresses in the range 1840 to 187F (hex) or E840 to E87F (hex) will select the Controller.

## 2.2 VDU CONTROLLER

The VDU Controller is a HD46505SP-2 IC operating under program control. The VDU Controller has 19 addressable Registers. The Acorn Bus address line A0 is used to select a Register when the Controller is selected. A Write operation to the Controller with a '0' on A0 selects the Address Register. The Register number on data bits D0—D4, will then determine which Register will be selected by a subsequent Read or Write operation to the VDU Controller, with A0 = '1'. Refer to Table 2 and paragraph 2.2.1 for the Registers and their functions.

The VDU Controller generates the VDU screen character positions as consecutive binary numbers on the Character Address lines CA0—CA10. The position of a character is completely defined by the address, refer to Figure 2. The display is organized into character rows, the Vert. Displayed Register specifies the number of rows, the Horiz. Displayed Register the number of characters (char.) per row to be dis-

R/NW	A0	ADDRESS REG. BITS					No.	REGISTER		FUNCTION
		4	3	2	1	0		Name		
0	0	—	—	—	—	—	AR	Address	Contains selected Register No.	
0	1	0	0	0	0	0	R0	Horiz. Total	No. of char./line — 1	
0	1	0	0	0	0	1	R1	Horiz. Displayed	No. of char./line displayed	
0	1	0	0	0	1	0	R2	HS Position	Start of HS (char. No. — 1)	
0	1	0	0	0	1	1	R3	Sync Width	Sync pulse width D0—D3 = Char. times (HS) D4—D7 = Scan lines (VS)	
0	1	0	0	1	0	0	R4	Vert. Total	No. of char. rows/frame — 1	
0	1	0	0	1	0	1	R5	Vert. Total Adjust.	No. of scan lines to complete frame	
0	1	0	0	1	1	0	R6	Vert. Displayed	No. of char. rows displayed/frame	
0	1	0	0	1	1	1	R7	VS Position	Start of VS (char. row No. — 1)	
0	1	0	1	0	0	0	R8	Interlace & Skew	Select Interlace Mode	
0	1	0	1	0	0	1	R9	Max. Taster Address	No. of scan lines/char. — 1	
0	1	0	1	0	1	0	R10	Cursor Start Raster	D0—D4 = Cursor start (scan line No.) D5, D6 = Cursor Mode	
0	1	0	1	0	1	1	R11	Cursor End Raster	Cursor end (scan line No.)	
0/1	1	0	1	1	0	0	R12	Start Address (H)	Address of start of display (char. No.)	
0/1	1	0	1	1	0	1	R13	Start Address (L)		
0/1	1	0	1	1	1	0	R14	Cursor (H)	Address of Cursor position (char. No.)	
0/1	1	0	1	1	1	1	R15	Cursor (L)		
1	1	1	0	0	0	0	R16	Light Pen (H)	Address of Light Pen position (char. No.)	
1	1	1	0	0	0	1	R17	Light Pen (L)		

Table 2. VDU Controller Registers

played. The 80 x 25 VDU Interface Board can display up to 2000 char., arranged as 25 rows with 80 char. in each row.

The number of scan lines per char. row is selected by program. The VDU Controller generates the scan line addresses on Raster Address lines RA0—RA3 for each char. row.

### 2.2.1 Register Functions

The functions of the VDU Controller Registers are as follows:

- ADDRESS (AR): Contains the vector number (hex) of the specified internal Control Register in bits 0-4. Accessed by a Write operation when A0 = '0'r
- HORIZ. TOTAL (R0): Contains the total number of char. in a row, minus one. (Determines HS frequency).
- HORIZ. DISPLAYED (R1): Contains the total number of char. to be displayed in a row, i.e. 80.
- HS POSITION (R2): Contains the number of the char. position in a row, minus one, at which the HS signal starts.
- SYNC WIDTH (R3): Divided into two parts, Bits 0-3 define length of HS signal in char. time units in hex (bits 0-3 = 0000 not valid). Bits 4-7 define length of VS signal as number of scan lines index (except, bits 4-7 = 0000 given 16 scan lines duration).
- VERT. TOTAL (R4): Total number of char. rows, minus one. Bit 7 not used.
- VERT. TOTAL ADJUST (R5): Contains the number of additional scan lines to complete scan field. Bits 5-7 not used.
- VERT. DISPLAYED (R6): Contains the total number of rows to be displayed on the VDU Screen. Bit 7 not used.
- VS POSITION (R7): Contains the number of the char. row, minus one, at which the VS signal starts. Bit 7 not used.
- INTERLACE & SKEW (R8): Selects the VDU Controller operating mode, refer to Figure 6.
- MAXIMUM RASTER ADDRESS (R9): Contains the total number of scan lines per char. row, including spacing. When Non-interlace' Mode or Interlace Sync Mode selected the number of scan lines per char. row, minus one, is required. For Interlace Sync and Video Mode the number of scan lines per char. row, minus two, is required. Bits 5-7 not used.
- CURSOR START RASTER (R10): Bits 0-4 contain the number of the first scan line of the

Cursor in a char. row, Bit 7 nis not used. Bits 5 and 6 control the Cursor Mode as follows:

#### BIT 6 BIT 5 CURSOR MODE

0 0	No blinking
0 1	No Cursor
1 0	Blink at 1/16 field rate
1 1	Blink at 1/32 field rate

NOTE: For Block Cursor set R10 to 0 and R11 to last char. scan line number.

- CURSOR END RASTER (R11): Bits 0-4 contain the number of the last scan line of a Cursor in a char. row. Bits 5-7 not used.
- START ADDRESS (R12, R13): A two byte Register that contains the char. number from which the display starts. When scrolling is required the address of the first char. in the top row to be displayed is loaded into R12 and R13. R12 bits 6 and 7 not used (00 when read).
- CURSOR POSITION (R14, R15): A two byte Register that contains the char. number at which the Cursor is to be displayed. R14 bits 6 and 7 not used (00 when read).

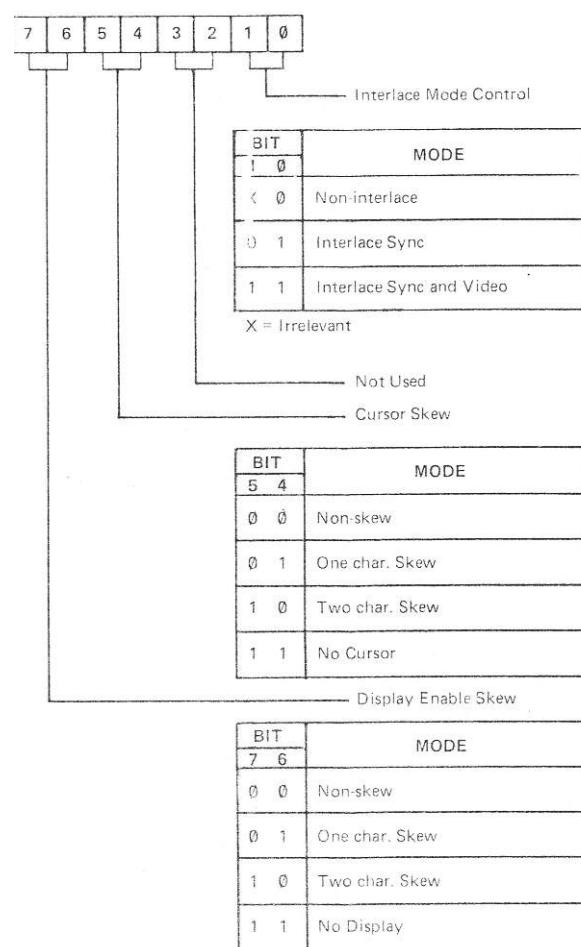


Figure 6. Interlace and Skew Register (R8) Bit Significance

- LIGHT PEN (R16, R17): A two byte Register that is loaded with the char. number, when a positive edge is detected on the LP input. R16 bits 6 and 7 are always 00. (Refer to para 2.2.5 for further details).

### 2.2.2 System Interface Signals

The following inputs and outputs are used to control the operation of the VDU Controller:

- S2: The CPU Phase 2 (S2) clock input controls the timing of data transfers between the System and the VDU Controller.
- R/NW: This signal input from the System selects the direction of data transfers.
- NCS: The Chip Select signal input is set low by the NCONSEL signal from the address decoder, when the VDU Controller address is detected.
- RS: The Register Select input is connected to address line A0. A '0' on A0 selects either the Address Register for a Write operation or the Status Register for a Read operation. When A0 is '1' the address in the Address Register determines which Register is accessed by the System.
- D0—D7: The Data input/output lines are enabled by a low on the NCS input. The lines are high impedance when NCS is high.

### 2.2.3 Video Interface Signals

The following inputs and outputs are used by the VDU Controller to control the video output from the VDU Board:

- HS: This signal generates the horizontal (lines) sync pulse to the Video Circuit. The timing of the signal is programmable, refer to para 2.2.1.
- VS: This signal generates the vertical (frame) sync pulse to the Video Circuit. The timing of the signal is programmable, refer to para 2.2.1.
- DE: The Display Enable (DE) signal is set high during the transmission of the display characters, refer to para 2.2.7.
- CU: The Cursor (CU) signal is set high when the character address corresponds to the Cursor address held in the Cursor Address Register. The Cursor may be either a block, or an underline, the mode of operation is selected by program.
- CK: The Clock (CK) input is driven by the 2MHz' clock, generated by the Clock Circuit, refer to para 2.3.1.

### 2.2.4 Memory Address Signals

The following memory address signals are used to select the display characters:

- CA0—CA10: The Character Address (CA) lines provide the binary addresses of the locations in the RAM, which contain the characters to be displayed.
- RA0—RA3: The Raster Address (RA) lines are used to select the appropriate dot output from the Character ROM on the current scan line to produce the characters read from the RAM location by CA0—CA10.

### 2.2.5 Light Pen

The Light Pen (LP) input stores the current character address in the Light Pen Register, when a positive edge is detected on the input. LK1 is used to connect the input to 0V when not used.

Since there is a two char. time delay between the output of a char. address from the VDU Controller and the display of that char. on the VDU screen and an internal one char. delay, the value of the char address stored in the LP Register will typically be three char. times later than the actual Light Pen location.

### 2.2.6 VDU Controller Read/Write

When the VDU Controller is addressed, the selected Register is read or written to by a Read or Write operation from the System via the Acorn Bus. The timing of the operations is given on Figure 7. For details of the Register bit significance refer to para 2.2.1.

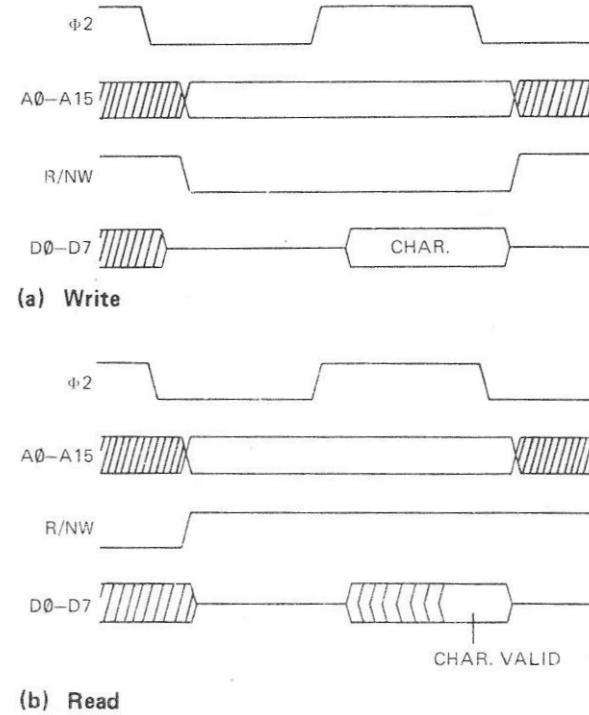
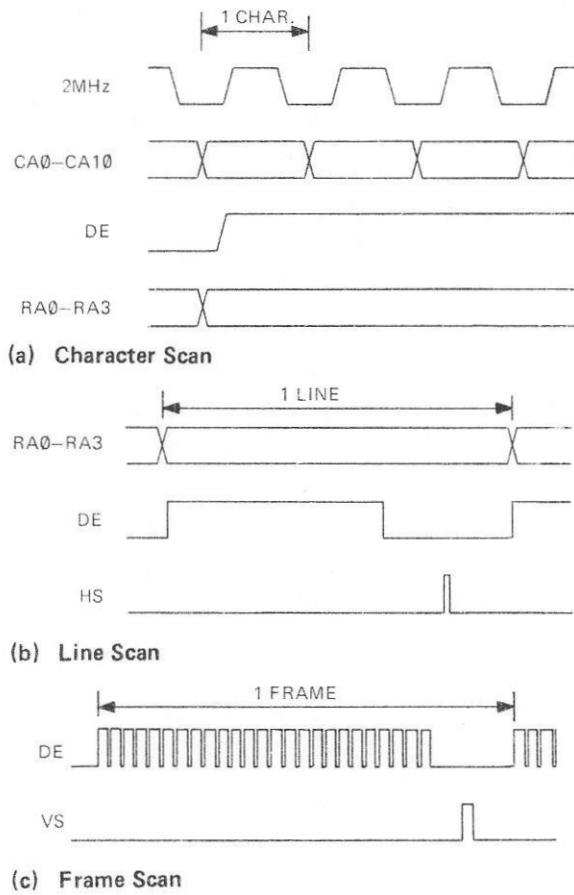


Figure 7. VDU Controller Write and Read

### 2.2.7 Horizontal Scan

The VDU Controller generates the Character Addresses CA0—CA10 and Raster Addresses RA0—RA3 together with HS and VS signals to produce the composite video output to the VDU.

The CA0—CA10 signals are connected via Multiplexers IC5, IC6 and IC7 to the RAM address inputs. The RA0—RA3 signals are connected directly to the Character Generator ROM IC13. The RA0—RA3 signals are all set to '0' at the beginning of the first scan line of a character row. The CA0—CA10 signals are set to the first character address in the row and then incremented at 2MHz, until the last character address in the row is reached. The Data Enable (DE) signal is set high during this time. Refer to timing diagram, Figure 8.



**Figure 8. VDU Controller Scan Signals**

When the address of the last character to be displayed in the row is reached (specified by the Horiz. Displayed Register), DE is set low. The character address count continues. When the character number in the row, specified as the start of the HS by the HS Position Register, is reached, the HS signal is set high. HS remains high for the number of character times specified in the Sync Width Register.

The character address count continues until the number specified by the Horiz. Total Register plus one is reached. The character number is then reset to the address of the first character in the row, and the scan address is incremented to the next scan line. The horizontal scan process is then repeated for the next scan line and for the other scan lines in the character row.

### 2.2.8 Vertical Scan

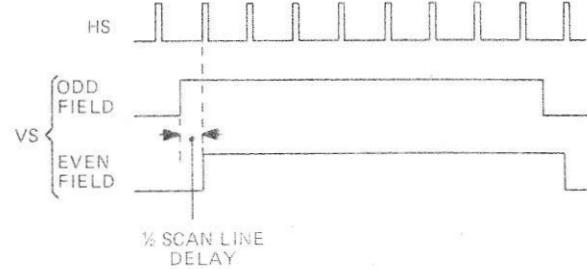
When the last scan line in the character row has been completed, the character address is set to the first character in the next row and the RA0—RA3 signals are all set to '0'. The horizontal scan process is then repeated.

When all the displayed character rows have been scanned, the horizontal scan process is continued for the non-display rows. The VS signal is set high when the row count reaches the value specified in the VS Positions Register. VS remains set high for the number of scan lines specified in the Sync Width Register.

The horizontal scan process continues until the number of character rows specified in the Vert. Total Register plus one have all been scanned. The VDU Controller continues the scan for the number of scan lines specified in the Vert. Total Adjust Register. On completion of the scan of the last line, the scan of one field is completed. The VDU Controller then sets the character address to the address specified in the Start Address Register and starts to scan the next field.

### 2.2.9 Scan Interface

The VDU Controller can be programmed for non-interlaced or interlaced scan modes. In the non-interlaced mode the VDU Controller produces fields at the programmed rate, e.g. at 50Hz field rate, a field of 312 scan lines can be produced. In the interlaced scan mode alternate odd and even fields are produced to generate frames, e.g. at 50Hz field rate at a frame of 625 scan lines can be produced. The odd fields are displaced by half a scan line time with respect to the even fields, refer to Figure 9. Thus the raster scan of an odd field starts at the centre top of the VDU screen and interlaces between the even field lines previously scanned.



**Figure 9. VS Signal Inning Interlace-Sync Mode**

### 2.2.10 Screen Scrolling

To provide the scrolling facility, the VDU Controller counts from the address in the Start Address Register, to the address determined by the Horiz. Displayed and Vert. Displayed Registers, e.g. 80 char./row by 25 rows = 1999 (7CF hex). The VDU Controller then sets the character address to '0' and counts up to the address in the Start Address Register — 1, to complete the display.

### 2.3 VDU CLOCK CIRCUIT

The VDU Clock Circuit is used to generate the NL, 2MHz' and 12MHz' clock signals from the Phase 2 ( $\Phi_2$ ) clock input, refer to Figure 10. When 12MHz and 2MHz clock signals are available in the System on the Acorn Bus, the VDU Clock Circuit 2MHz' and 12MHz' Clock Generator Circuits are not used. The Links LK4—LK8 are provided to select the required mode of operation, refer to Table 3.

MODE	$\Phi_2$	LINKS
Local Clock Generator	1MHz	LK4, LK7
Local Clock Generator	2MHz	LK5, LK7
External Clocks	—	LK6, LK8

Table 3. VDU Clock Links

The dot Load (NL) clock signal is generated from whichever 2MHz clock source is selected by the Links. The Phase 2 ( $\Phi_2$ ) clock signal is inverted by IC15c to produce the N $\Phi_2$  clock signal irrespective of the selected clock source.

### 2.3.1 2MHz' Clock Signal Generation

The 2MHz' Oscillator Circuit consists of gates IC16d and IC17b together with associated resistors and capacitors. The  $\Phi_2$  clock input is inverted by IC15c to generate N $\Phi_2$ . The N $\Phi_2$  clock signal edge is delayed by R 11 and C5, so that exclusive OR gate IC16d generates a short negative pulse, refer to timing diagram Figure 11. The pulse from IC16d synchronizes the 2MHz' Oscillator Circuit IC17b, RV2, R10 and C4. The Oscillator output is connected via LK4 to NAND gate IC18c, which inverts the signal to produce the 2MHz' VDU Clock Circuit output.

When the 2MHz' clock is derived from a 2MHz  $\Phi_2$  clock signal, LK5 connects N $\Phi_2$  from IC15c to IC18c to produce the 2MHz' VDU Clock Circuit output.

When the 2MHz clock input is used, LK6 connects the inverted 2MHz output from IC15a to IC18c, to produce the 2MHz' VDU Clock Circuit output.

### 2.3.2 Load Signal Generation

The Load (NL) signal is generated by IC17d from the selected 2MHz clock signal. R8 and C3 delay the negedge of the 2MHz clock input to IC18c, so that the positive edge on the output of IC18c is input to IC17d, before C3 discharges below the Schmitt-trigger threshold. A negative pulse, signal NL, of typically 20ns is generated at the output of IC17d at 2MHz frequency.

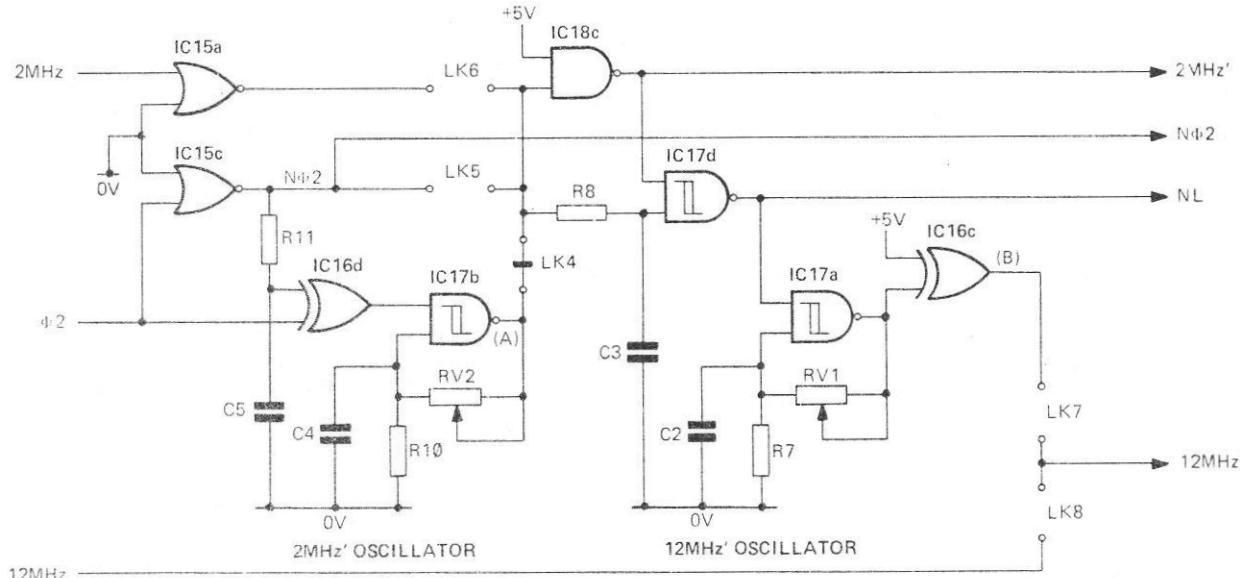
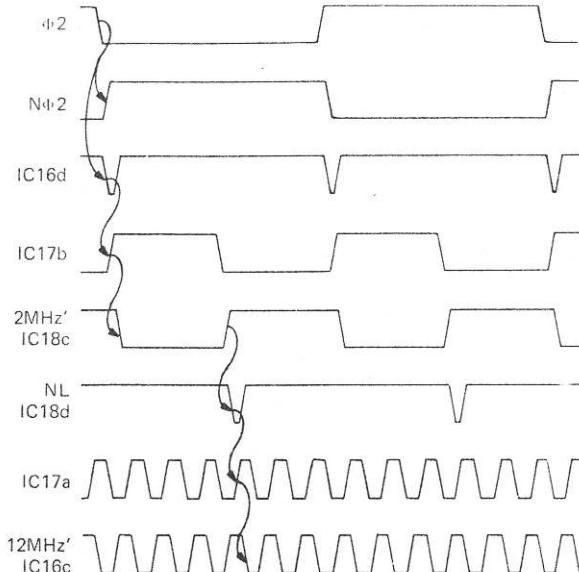


Figure 10. VDU Clock Circuit



**Figure 11. VDU Clock Timing ( $\$2 = 1\text{MHz}$ )**

### 2.3.3 12MHz' Clock Signal Generation

The 12MHz' Oscillator Circuit consists of IC17a, IC16c, RV1, R7 and C2. The NL signal from IC17d synchronizes the 12MHz' Oscillator circuit. The output is connected via inverter IC16c to LK7. When the 12MHz clock from the Acorn Bus is required this is selected by LK8.

### 2.3.4 VDU Clock Setting-Up Procedure

To set up the VDU Clock Oscillator Circuits the following equipment is required:

Dual-beam oscilloscope  
+5V = 1% power supply  
Frequency Generator

#### 2MHz' OSCILLATOR

Connect the oscilloscope as follows (refer to Figure 10):

Channel A to IC17/6 (A)  
Channel B to  $\$2$ , IC16/13 (B)

Set the oscilloscope to:

Amplitude	2V/cm
Timebase	100ns/cm
Trigger	Channel B

Connect the +5V supply to the board edge connector as follows:

+5V to Side A pin 1  
0V to Side A pin 32

Connect the frequency generator output to the board edge connector Side A pin 29 and 0V to Side A pin 32. Adjust the frequency generator output to 1.00MHz, amplitude +3.5V w.r.t. 0V.

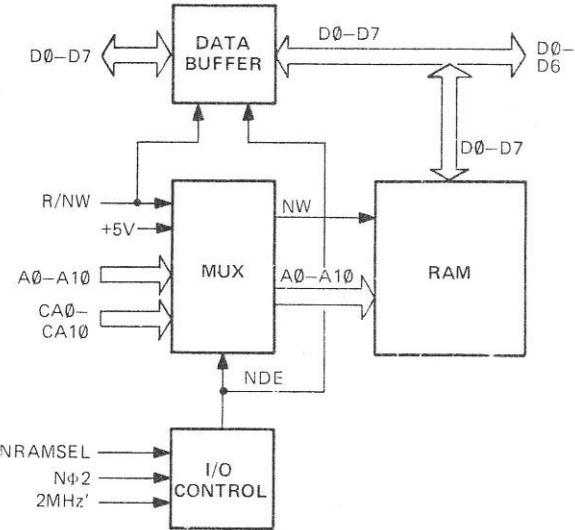
Ensure that the trace on Channel A has the relationship to the  $\$2$  signal (Channel B) shown in Figure 11. Adjust RV2 to obtain a symmetrical trace.

### 12MHz' OSCILLATOR

Connect the oscilloscope Channel A to IC16/8 and ensure that the trace on Channel A is as shown in Figure 11. Adjust RV1 to obtain a symmetrical trace.

## 2.4 CHARACTER RAM

The Character RAM Circuit consists of the four RAM ICs IC8-IC11, the address Multiplexers IC5, IC6, IC7, the Data Buffer IC4 and the I/O Control Circuit, refer to block diagram Figure 12.



**Figure 12. Character RAM Circuit**

The D2114 AL-2 RAM ICs are arranged in pairs, each pair provides 1024 (1K) x 8 bit memory locations. The locations are addressed by the Character RAM address lines A0—A10 from the Multiplexers (MUX). The Character RAM Write line (NW) is selected from either the Acorn Bus, R/NW line, or +5V by the MUX.

### 2.4.1 System Read or Write Operation

The I/O Control Circuit IC15d and IC18b is enabled by a low on the NRAMSEL signal from the Address Selection Circuit, refer to para 2.1.1.

The N $\$2$  clock signal derived from the Acorn Bus Phase 2 ( $\$2$ ) clock signal ie IC15c, is delayed by R14 and C11 and gated with the 2MHz' clock signal from the VDU Clock Circuit at NAND gate IC18b, refer to timing diagram, Figure 13. The Data Enable (NDE)

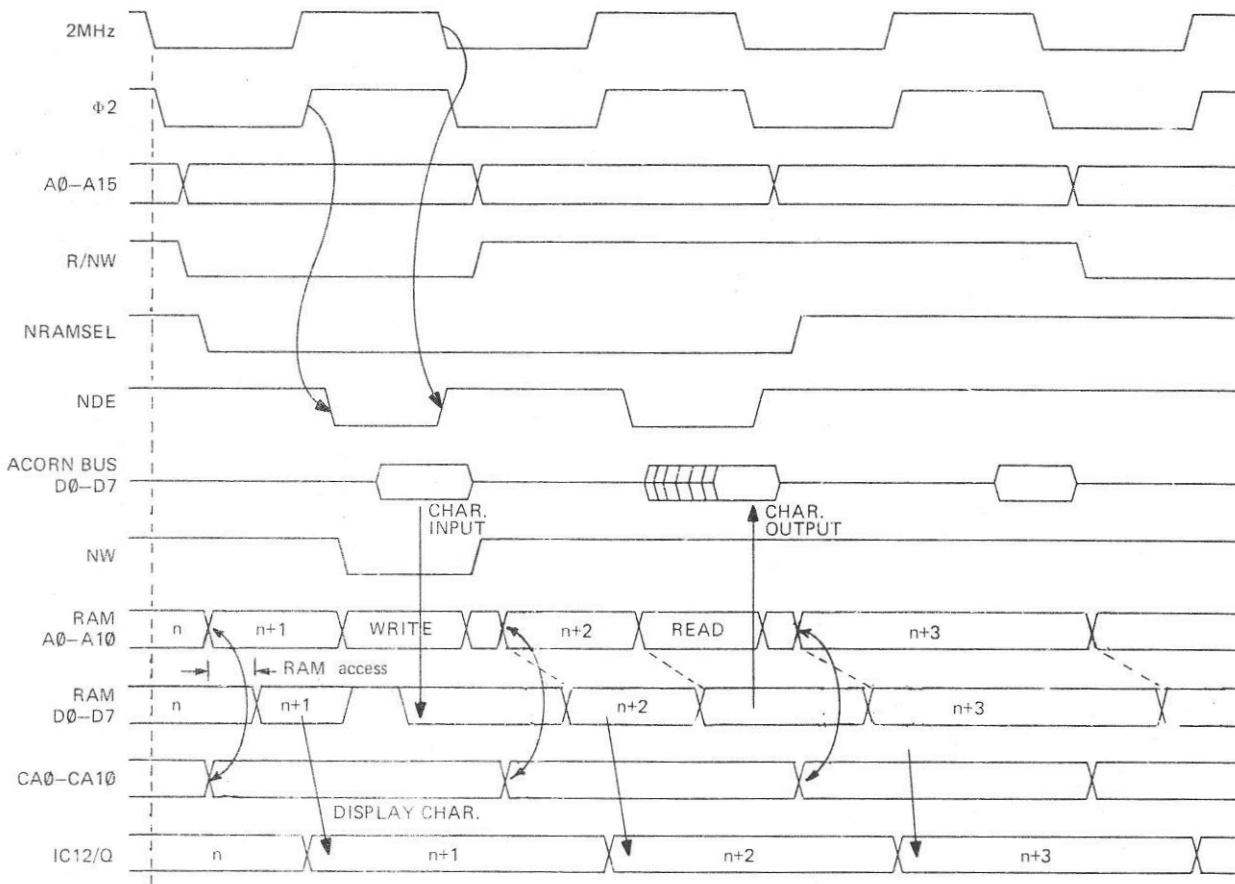


Figure 13. Typical Character RAM Timing

signal from IC18b enables the Data Buffer IC4 for data transfer between the Acorn Bus data lines D0—D7 and the Character RAM data lines D'0—D'7. The direction of transfer is selected by the R/NW signal:

The NDE signal is also used to switch the MUX ICs, to select Acorn Bus address lines A0—A10 as inputs to the Character RAM. The R/NW line is also connected to the Character RAM NW signal line.

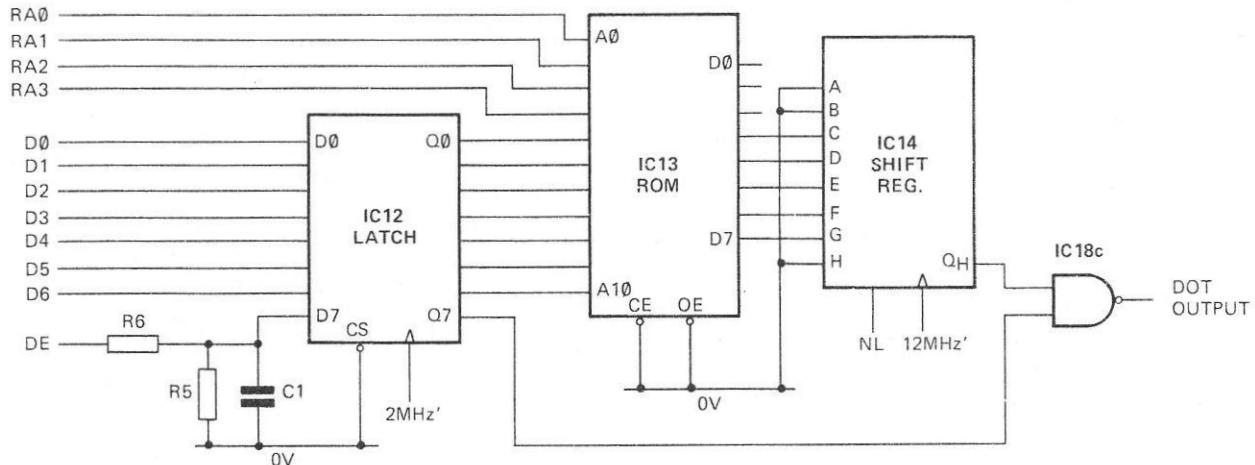
For a Write operation the character on D'0—D'7 is written into the Character RAM location addressed by A0—A10 when the NW signal is low.

For a Read operation the character in the location addressed by A0—A10 is output on D'0—D'7.

#### 2.4.2 VDU Controller Read Operation

When the Character RAM is not selected for a System Read or Write operation, the CA0—CA10 address lines from the VDU Controller are connected via the multiplexers to RAM address lines A0—A10. The Character RAM NW line is connected to +5V to inhibit writing to the RAM in this mode of operation.

The character in the location addressed by CA0—CA10, is output onto the data lines D'0—D'7. The Character RAM access time is typically 120ns.



**Figure 14. Character Generator Circuit**

## 2.5 CHARACTER GENERATOR

The Character Generator Circuit consists of the octal latch IC12, Programmable ROM IC13 and shift register IC14, refer to Figure 14. Characters stored in the Character RAM are loaded into the octal latch. The latch output addresses the Programmable ROM, which is programmed to decode the character into the appropriate dot pattern for the scan line selected by RA0—RA3. The parallel Character ROM output is converted to a bit serial output by the shift register.

### 2.5.1 Character Latch

The character code read from the Character RAM on data lines D'0—D'6 (D'7 not connected) is latched into IC12 by the positive edge of the 2MHz' clock signal from the VDU Clock Circuit. In place of the unused D' 7 bit signal line, the DE output from the CRT Controller is latched into the eighth bit of IC12. DE is delayed by R5, R6 and C1, so that a change in DE is not latched until one character time after the VDU Controller output changes state.

### 2.5.2 Character ROM

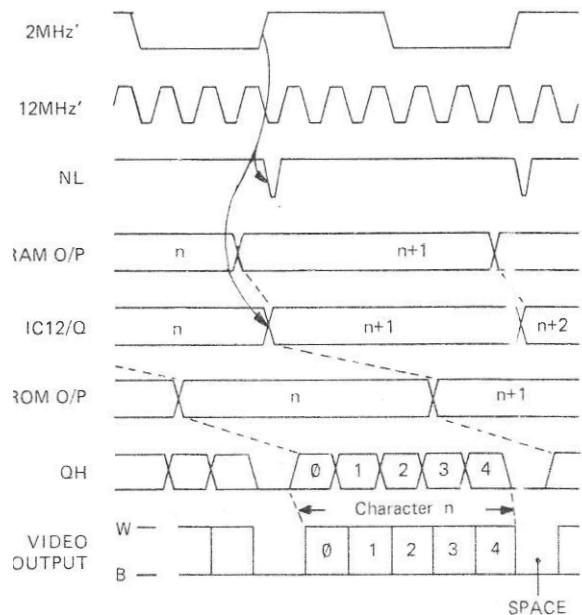
The RA0—RA3 signals from the VDU Controller determine which is the current scan line in the character row. The character code from the latch IC12 is connected to ROM address inputs A4—A10, so that the programmed dot pattern for a particular character is read from the addressed ROM location onto outputs D3—D7 (D0—D2 are not used).

### 2.5.3 Dot Output

The dot pattern from the ROM is loaded into the shift register IC14 by the NL signal from the VDU Clock Circuit. The dot pattern bits are loaded into the C, D, E, F and G inputs, while '0's are loaded into the A, B and H inputs. The bits in the shift register

are then clocked out by the 12MHz' clock signal, starting with bit H. Since the NL signal is generated from the 2MHz' clock only six bits will be shifted out of IC14 before the next NL signal, refer to timing diagram, Figure 15. The next dot pattern is then loaded into IC14.

The QH output of IC14 is connected to NAND gate IC18d. The delayed DE signal latched in IC12, enables IC18d, when a character is to be displayed. The character dots are then inverted by IC18d, so that a '0' represents a dot and a '1' a space on the VDU screen. This output is connected to the Video Output Circuit.



**Figure 15. Character Generator Timing**

## 2.6 VIDEO OUTPUT

The Video Output Circuit translates the binary dot pattern from the Character Generator into the analogue video output, refer to Figure 16. The circuit also mixes in the HS, VS and Cursor signals to provide the composite video for the VDU, refer to Figure 17.

### 2.6.1 Dot Display

When characters are to be displayed, signal DE from the VDU Controller is high enabling dot output, refer to para 2.5.1. The CU signal from the VDU Controller is low except for the character time when the Cursor is displayed. Link LK2 is open circuit for the display of white characters on a black background. These conditions produce a high output from exclusive OR gate IC16a, so that a dot bit low on the dot output from the Character Generator produces a high at the output of exclusive OR gate IC16/6. The output transistor TR1 is turned on by the high on IC16/6 to supply +1V on the VDU connector pin 1. When the dot bit output is high (no display dot), IC16/6 goes low. The low on IC16/6 pulls down the voltage on TR1 base, but diodes D2, D3 and D4 limit the voltage excursion. TR1 then supplies the black level output voltage of +0.3V approximately on the VDU connector pin 1.

### 2.6.2 Cursor Display

The VDU Controller may be programmed for either Block or Underline Cursor display. For Block Cursor the CU signal is high for all scan lines during the selected character time. For Underline Cursor the CU

signal is high for the selected number of scan lines during the selected character time. The CU signal can be delayed by one or two character times under program control, to allow for RAM access delay.

The CU signal high is delayed by R12, R13 and C6 to allow for dot signal propagation times. The CU signal high at IC16/1 and the +5V via R9 at IC16/2 generate a low output from the exclusive OR gate or IC16/3. This low at IC16/4 inverts the dot output. In Block Cursor mode the character displayed during the Cursor time will appear as black on a white background. In Underline Cursor mode the character position will be underlined by a solid bar.

### 2.6.3 Sync Signals

When the HS or VS signals are high IC15/4 output goes low. This low pulls down the voltage on the base of TR1 via diode D1 to turn off the transistor. When TR1 is off the voltage output on the VDU connector pin 1 falls to 0V, the video signal sync level.

### 2.6.4 Display Inversion

Link LK2 is provided to allow display inversion when required. Connecting a link in the LK2 position connects 0V to IC16/2. The output on IC16/3 is then low except when the CU signal from the VDU Controller is high. The dot output will generate a black level signal for a dot or a white level for spaces. The display on the VDU screen will consist of black dots on a white background. The Block Cursor will also be inverted to produce white dots on a black block.

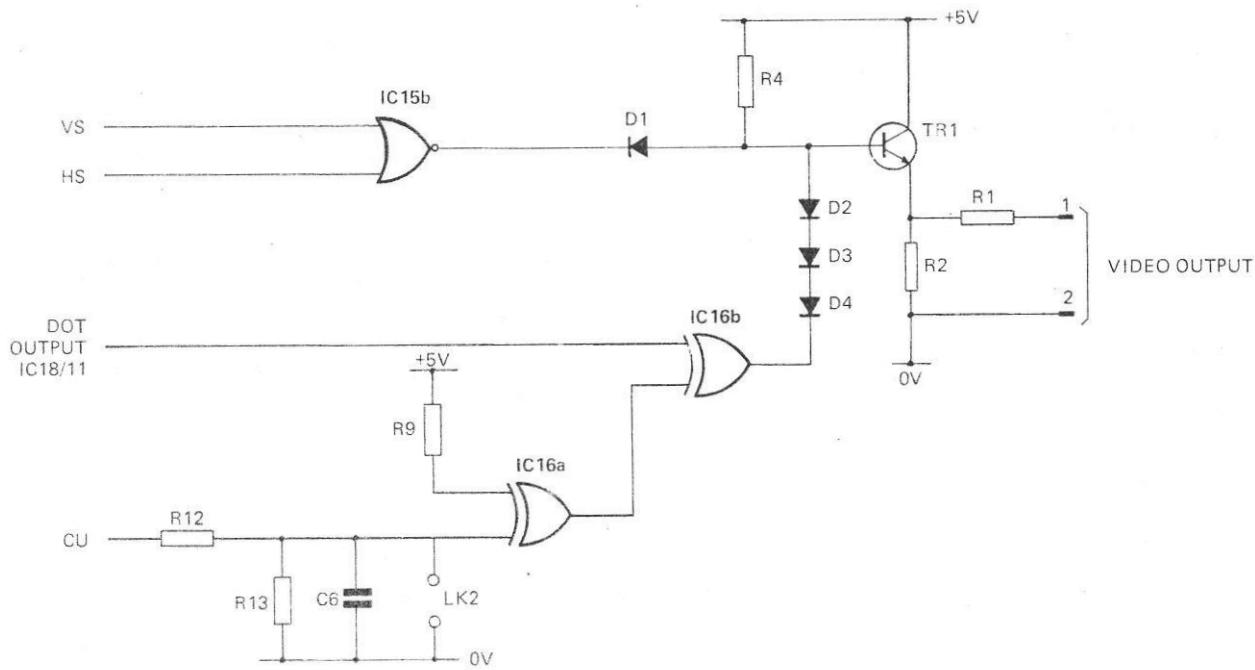


Figure 16. Video Output Circuit

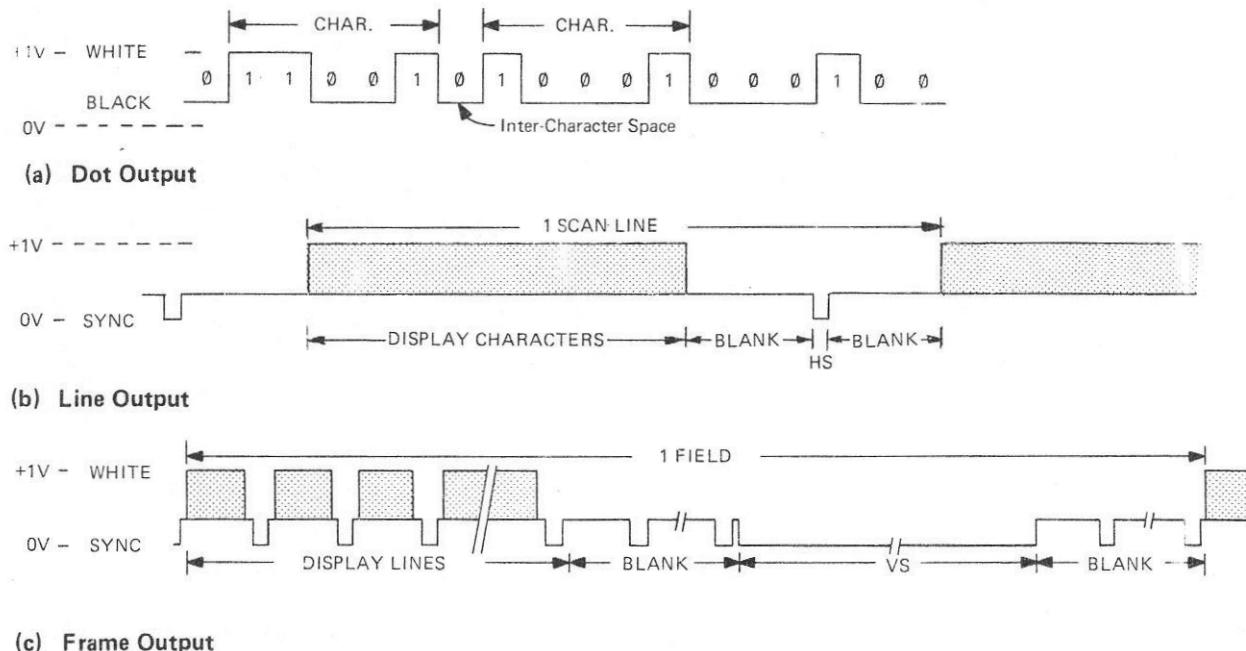


Figure 17. Composite Video Output Signal

### 3. SOFTWARE PARAMETERS

#### 3.1 VDU INTERFACE ADDRESSING

The Acorn Bus address lines A11—A15 are decoded for the Character Ram and VDU Controller addresses.

##### 3.1.1 Character RAM

The Character RAM can be located at one of two locations by links on the VDU Board. Links LK7—LK10 in the A position select RAM addresses 1000—17FF (hex). In the B position addresses F000—F7FF (hex) are selected.

##### 3.1.2 VDU Controller

The VDU Controller is normally assigned addresses 1840 and 1841 (hex), Links LK7—LK10 in the A position. Since the address lines are not completely decoded, the addresses 1840-187F (hex) must be assigned to the VDU Controller. In the B position addresses E840—E87F (hex) are assigned to the VDU Controller.

##### 3.1.3 Paging Facility

The VDU Board is provided with a paging facility. The Board is supplied with a track link in LK3, to use the facility the link must be cut. The Valid Memory Address (NVMA) signal input on the Acorn Bus connector pin 24b, will then select the VDU Board when '0'.

### 3.2 VDU CONTROLLER

The VDU Controller is provided with internal programmable Registers. The Registers are listed in Table 2, and described in para 2.2.1.

#### 3.2.1 Register Addressing

The Address Register is addressed when the A0 address bit is '0'. To select the required Control Register, the Register number is loaded into the Address Register from data bits D0—D4 by a Write operation. Subsequent Read or Write operations to the VDU Controller with A0 = '1' will then be vectored to the required Control Register.

#### 3.2.2 Mode Control

The VDU Controller Interlace and Skew Register (R8) selects the mode of operation required as follows (bits 2 and 3 are not used):

##### INTERLACE MODE CONTROL

Bits 1 and 0 select the VDU scan interlace required.

- Non-interlaced Mode (bit 0 = '0') provides a non-interlaced scan, e.g. 312 lines at 50Hz.
- Interlaced Sync Mode (01), is used to enhance readability of characters. The same information is displayed in odd and even fields to produce a higher quality character display. The scan is interlaced so that each frame is two complete fields, e.g. a 50Hz 625 line frame will comprise two fields of 312½ lines superimposed, refer to para 2.2.9.
- interlaced Sync and Video Mode (11) not used:

#### CURSOR SKEW

Bits 5 and 4 select cursor operation and the amount of cursor skew as follows:

- Non-skew (00). Cursor operation selected with no skew delay.
- One char. skew (01). Cursor operation selected with a skew of one char. time.
- Two char. skew (10). Cursor operation selected with a skew of two char. times.
- No Cursor (11). Cursor operation not selected.

#### DISPLAY ENABLE SKEW

Bits 7 and 6 select Display Enable (DE) signal output and the amount of skew as follows:

- Non-skew (00). Display enabled with no skew.
- One charr skew (01). Display enabled after a delay of one char. time.
- Two char. skew (10). Display enabled after a delay of two char. times.
- No Display (11). DE signal output inhibited.

#### 3.2.3 Display Configuration

The Register settings used in System 5 to program the VDU Controller for operation at 50Hz field rate with non-interlaced scan (312 lines+field) are given in Table 4.

REGISTER	CODE (HEX)	SIGNIFICANCE
R0	7F	123 char./row
R1	50	80 char. displayed/row
R2	66	HS starts at char. 102
		VS = 6 lines scan time
R3	62	HS = 2 char. times
R4	1E	31 rows/field
R5	02	2 additional lines/field
R6	19	25 rows displayed/field
R7	18	VS starts at row 27
R8	40	Mode selected: — Non-interlace — 1 charr Cursor delay — No Display Enable delay
R9	09	10 scan lines/charr row
R10	68	Cursor blink at 1/32 field rate
		Cursor start: line 8
R11	09	Cursor end: line 9
R12	10	Display start address char. 0000
R13	00	(RI2 bit 4 not used by VDU Controller).

Table 4. Register Settings

#### 4. 80 x 25 VDU INTERFACE BOARD CONNECTIONS

##### 4.1 VDU BOARD TO ACORN BUS

Pin	Mnemonic	Meaning	I/O
<b>Side A</b>			
1	+5V	+5V Supply	I
2	A15		
3	A14		
7	A8		
8	A7		
9	A6		
10	A5	Address Lines	I
11	A4		
12	A3		
13	A2		
14	A1		
15	A0		
16	D7		
17	D6		
18	D5		
19	D4		
20	D3	Data Lines	I/O
21	D2		
22	D1		
23	D0		
24	A13		
25	A12		
26	A11	Address Lines	I
27	A10		
28	A9		
29	Φ2	Phase Two Clock	I
30	R/NW	Read/Write	I
32	0V	0V Supply	I
<b>Side B</b>			
13	12MHz	System Clocks	I
18	2MHz		
24	NVMA	Valid Memory Address	I
32	0V	0V Supply	I

##### 4.2 VDU BOARD TO VDU

Pin	Mnemonic	Meaning	I/O
1		Video Output	O
2		Video Output (0V)	O
3		+5V Supply	O
4		Light Pen	I
5		Light Pen (0V)	O

## 5: PARTS LIST

### 5.1 MECHANICAL PARTS

ITEM	DESCRIPTION	VALUE	QTY	PART NO.
	PCB 200, 019 issue 1	1		
	24 pin IC Socket	1		
	20 pin IC Socket	3		
	18 pin IC Socket	4		
	16 pin IC Socket	6		
	14 pin IC Socket	4		

### 5.2 ELECTRICAL PARTS

ITEM	DESCRIPTION	VALUE	QTY	PART NC.
C1	Capacitor	330pF	1	
C2	Capacitor	47pF	1	
C3	Capacitor	150pF	1	
C4, C5	Capacitor	470pF	2	
C6	Capacitor	330pF	1	
C7...C9	Capacitor	47nF	3	
C10	Capacitor, electrolytic	10µF 25V	1	
C11	Capacitor	150pF	1	
	Connector 64-way Plug (Right angle solder tails to DIN 41612)		1	
D1...D4	Diode	IN4148	4	
IC1, IC2	Integrated Circuit	74LS138	2	
IC3	Integrated Circuit	HD46505SP-2	1	
IC4	Integrated Circuit	8208	1	
IC5...IC7	Integrated Circuit	74LS157	3	
IC8...IC11	Integrated Circuit	2114-2	4	
IC12	Integrated Circuit	74LS374	1	
IC13	Integrated Circuit	MM52116 F DW*	1	
IC14	Integrated Circuit	74LS165	1	
IC15	Integrated Circuit	74LS02	1	
IC16	Integrated Circuit	74LS86	1	
IC17	Integrated Circuit	74LS132	1	
IC18	Integrated Circuit	74LS00	1	
RV1, RV2	Potentiometer, Miniature Skeleton	1kOhm	2	
R1	Resistor Carbon	470Ohm 0.25W	1	
R2	Resistor Carbon	1000Ohm 0.25W	1	
R3	Resistor Carbon	47kOhm 0.25W	1	
R4	Resistor Carbon	1kOhm 0.25W	1	
R5	Resistor Carbon	4.7kOhm 0.25W	1	
R6	Resistor Carbon	3.3kOhm 0.25W	1	
R7	Resistor Carbon	1kOhm 0.25W	1	
R8	Resistor Carbon	1500Ohm 0.25W	1	
R9	Resistor Carbon	10kOhm 0.25W		
R10	Resistor Carbon	1.2kOhm 0.25W		
R11	Resistor Carbon	820Ohm 0.25W	1	

\*Programmed by Acorn.

ITEM	DESCRIPTION	VALUE	QTY	PART NO.
R12	Resistor Carbon	1kOhm 0.25W	1	
R13	Resistor Carbon	2.7kOhm 0.25W	1	
R14	Resistor Carbon	82Ohm 0.25W	1	
TR1	Transistor	2N2369	1	

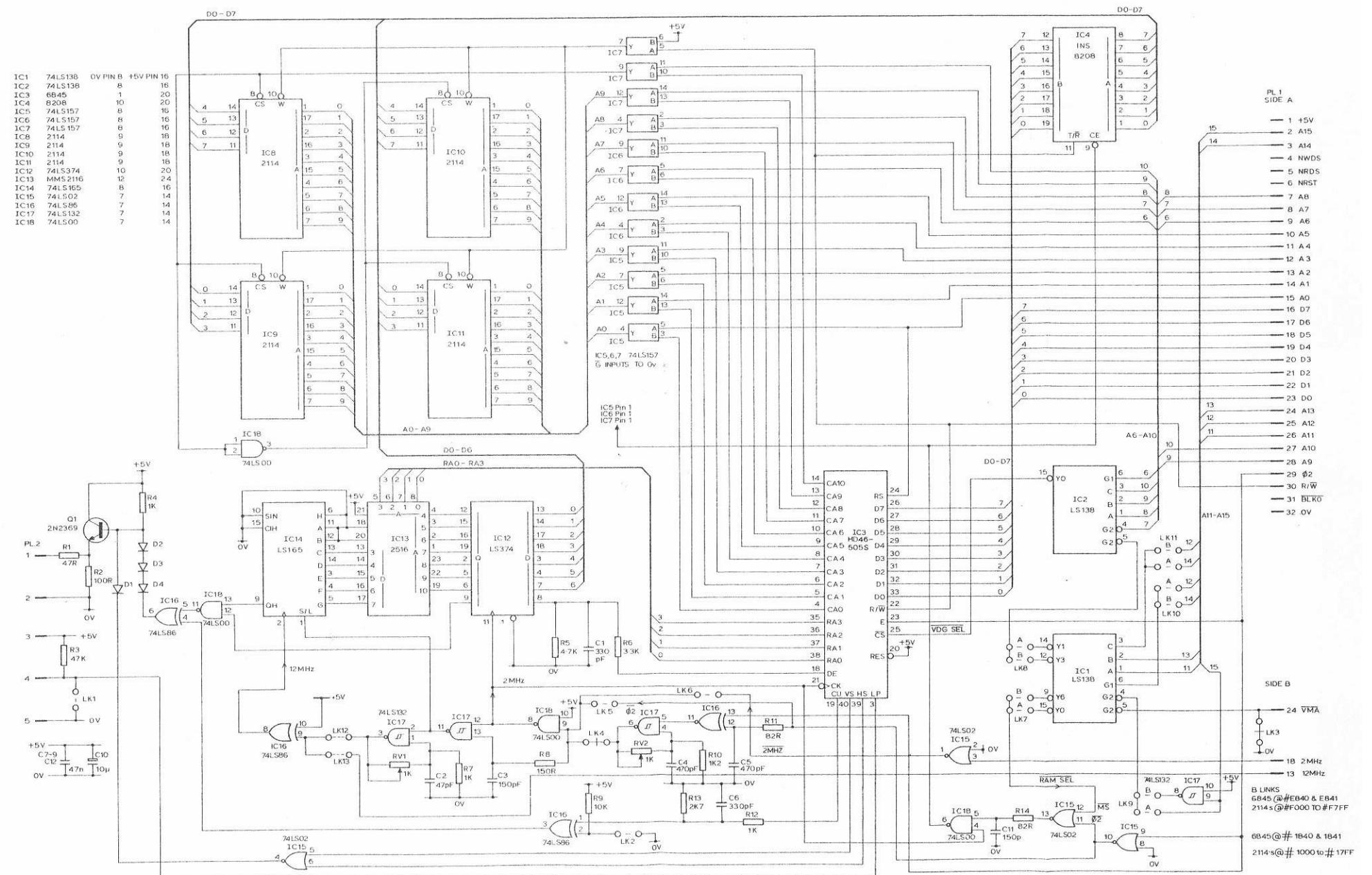


Figure 18. 80 x 25 VDU Interface Circuit Diagram

## ACORN FLOPPY DSC CONTROLLER BOARD

### UNIT DESCRIPTION

### CONTENTS

Paragraph		Page
1.	INTRODUCTION	1
1.1	General	1
1.2	Principles of Operation	1
1.3	Leading Particulars	3
1.3.1	Mechanical	3
1.3.2	Power Supplies	3
1.3.3	Connections	<b>3</b>
2.	CIRCUIT DESCRIPTION	3
2.1	CPU Interface	3
2.1.1	Addressing	3
2.1.2	Input/Output	3
2.2	8271 Floppy Disc Controller	3
2.3	Clock Signals	4
2.3.1	4MHz Operation	4
2.3.2	Local Oscillator Operation	4
2.3.3	Local Oscillator Setting-Up Procedure	4
2.4	Data Separator	5
2.5	Ready Signals Generator	6
2.6	Floppy Disc Controller/Drive Interface	6
2.6.1	Outputs from Floppy Disc Controller	6
2.6.2	Inputs from Floppy Disc Drive Unit	8
3.	SOFTWARE PARAMETERS	8
3.1	DOS	8
3.2	Addressing	9
3.3	Disc Format	9
3.4	Disc Errors	10
4.	FLOPPY DISC CONTROLLER CONNECTIONS	11
4.1	Floppy Disc Controller Board to Acorn Bus	11
4.2	Floppy Disc Controller Board to Floppy Disc Drive (Daisy Chained to All Drives)	11
5.	PARTS LIST	12
5.1	Mechanical Parts	12
5.2	Electrical Parts	12

01982 Acorn Computers Limited, 12 Fulbourn Road, Cherry Hinton, Cambridge.  
Written and Produced for Acorn Computers Limited by Arch Technical Publications Limited,  
St. Lukes House, 5, Walsworth Road, Hitchin, Herts.

**TABLES**

<b>Table No.</b>		<b>Page</b>
1.	Addressing	9
2.	Disc Drive Selection	9
3.	Error Messages	10

**FIGURES**

<b>Figure No.</b>		<b>Page</b>
1.	Floppy Disc Controller Block Diagram	2
2.	Floppy Disc Controller Write and Read	4
3.	Data Separator	5
4.	Ready Signals Generator	6
5.	Drive Timing	7
6.	Data Timing	8
7.	Data Format	10
8.	Floppy Disc Controller Circuit Diagram	13

## 1. INTRODUCTION

### 1.1 GENERAL

The Floppy Disc Controller Board provides an interface to one or two, single or double-sided floppy disc drive units. The board connects to the Acorn Bus and, via a ribbon cable, to the floppy disc drive; multiple floppy disc drives are daisy chained.

Throughout this document, the terms disc and disc drive refer respectively to the flexible plastic disc recording media ("floppy disc", "mini floppy", "flippy", etc) and to the unit containing the drive mechanism and read/write/control electronics into which the disc is inserted.

The board uses the 8271 programmable floppy disc controller device (FDC). The FDC is a high level controller that relieves the CPU of many tasks associated with floppy disc control. It supports a soft sectored format that is IBM 3740 compatible.

The Acorn Disc Operating System (DOS) provides comprehensive commands for the control of disc drives. This is available as ROM pre-programmed to suit a number of floppy disc drive types.

### 1.2 PRINCIPLES OF OPERATION

A block diagram of the Floppy Disc Controller Board is given on Figure 1.

The FDC executes each data record transfer, writing to or reading from the disc, in three phases. It accepts commands from the CPU, executes the command (transfers the data record byte by byte), and provides a result (e.g. transfer completed) at the end of the execution phase.

The FDC is controlled by the CPU using various command instructions. Each command is executed autonomously by the FDC. At the end of the execution phase, the FDC flags the CPU (NNMI) which reads the current states (result and status register) before issuing the next command.

Each command defines the mode of operation required, such as Initialization, Read Data, Write Data, etc. and may be supplemented by up to five parameter bytes which further define the command.

At power up or system reset (NRST), the FDC is reset. This halts any commands in progress, clears the registers, and forces the FDC into an idle state. The FDC can also be reset by software using a Reset Command.

The FDC is initialized using commands and parameters which specify the various disc drive characteristics such as step rate between tracks, head settling time (pause between final step signal to drive and the writing/reading of data), and head load time.

The head is loaded when a disc is selected, and remains in contact with the disc while stepping over the surface. The head load time is the time between head load and the writing/reading of data.

The final phase of initialization consists of setting the head(s) on each drive equipped to track 0 using a Seek Command. This command causes the FDC to locate the head of the first drive to track 0 and to interrupt the CPU which reads and resets the Result Register. The FDC then tests the second disc drive position. If a drive is present, its head is located to track 0 and the CPU interrupted to read and reset the Result Register for the second drive.

The locating of the head(s) to track 0 ensures that the head position is known, particularly after power up.

Each data transfer (write to or read from disc) consists of three phases. Each phase is initiated by the CPU with all further actions controlled by the FDC. These phases are:

#### COMMAND PHASE

The CPU addresses the FDC to enable its input and output buffers and the command and parameter registers as appropriate and transfers a command (Table 1) and from zero to five parameter instructions via the Data Bus, to the selected register. The FDC is addressed for each command or parameter transfer.

The command defines the type of transfer required and the parameters define the data, e.g. location (disc, surface, track, sector) and length of data record.

On completion of the Command Phase, the FDC enters the appropriate Execute Phase.

#### WRITE EXECUTE PHASE

During this phase, each byte of the data record to be written is loaded onto the Data Bus by the CPU and the NWDS signal set active to initiate the writing of that byte.

The CPU addresses the FDC to enable the data input buffers and sets the NWDS signal active to initiate a write transfer. The data byte on the Data Bus is loaded into the FDC, converted to a serial format, clock pulses added and, under control of the FDC transferred to the disc drive at the correct time to ensure loading to the correct sector. When the data

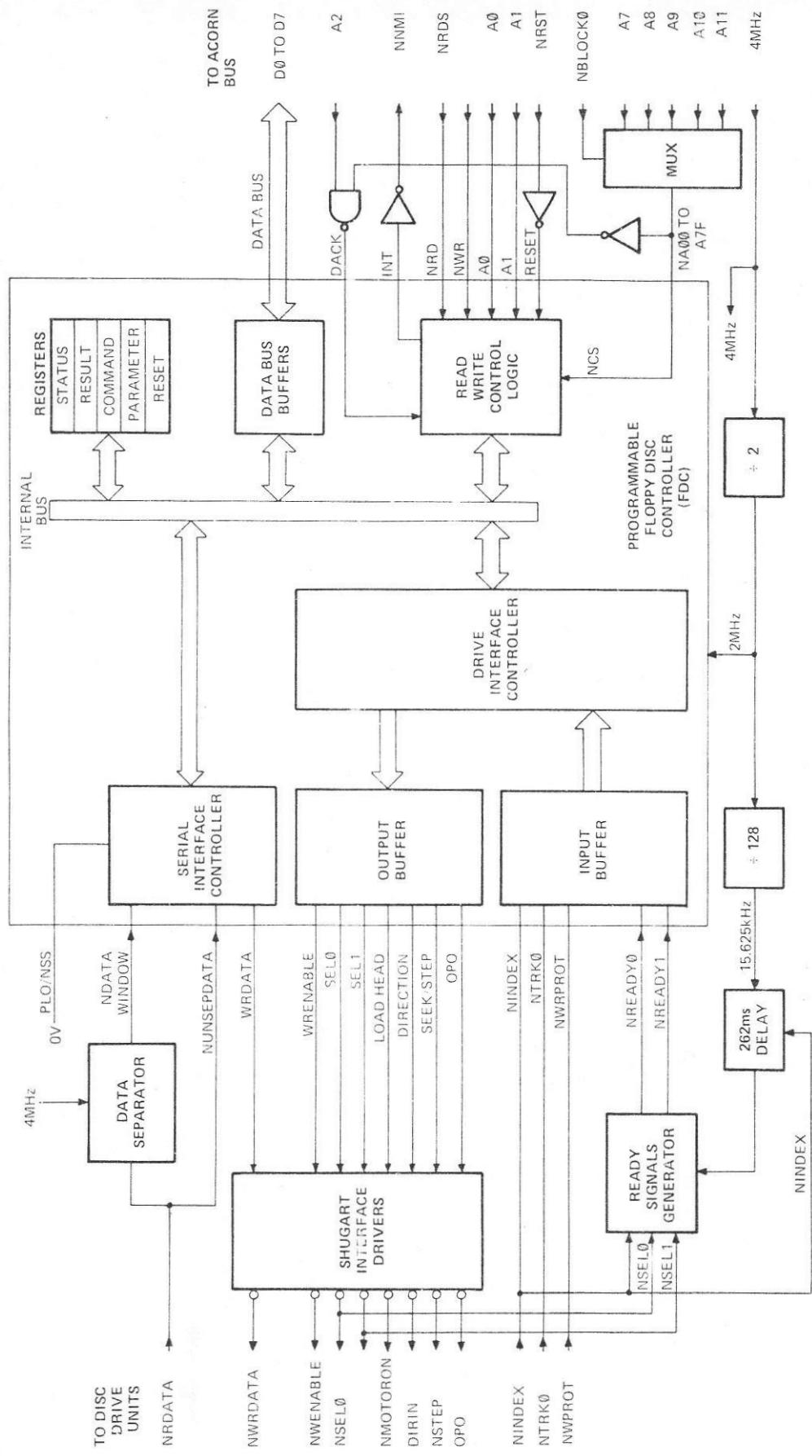


Figure 1. Floppy Disc Controller Block Diagram

byte has been transferred, the FDC flags the CPU (NNMI) to indicate that it is ready to receive the next data byte. The CPU reads the Status Register and the process is repeated for the remaining bytes of the data record. These transfers are synchronized by the FDC setting the NNMI interrupt flag active once every 64µs.

#### READ EXECUTE PHASE

The FDC searches for the required data record (defined by the command and parameter instructions). When the first byte is detected, it is read off the disc to the FDC which removes the clock pulses, converts it to a parallel form and flags the CPU using NNMI. The CPU reads the Status Register and then sets NRDS active enabling the transfer of the data byte onto the Acorn Bus. This procedure is repeated for each data byte with each data transfer between the disc, the FDC, and Acorn Bus being synchronized by NNMI once every 64µs.

#### RESULT PHASE

At the end of the Execute Phase, the Result Register Full bit is set in the Status Register, to indicate that the Command and Execute Phases of the data transfer have been completed.

During each data record transfer, the FDC maintains a tally on the number of data bytes transferred and any errors detected during the transfer. On completion of the data transfer, this data becomes valid in the Result Register. Following the next Status Register read operation, the CPU also reads the Result Register which indicates a successful completion of the command or, if not, the type of error or errors detected. These errors are more fully defined in Section 3.

When the Result Register has been read by the CPU, it is reset and the Status Register is updated to idle.

The FDC automatically unloads the disc head(s) if another command is not received from the CPU within ten revolutions of the disc (2s) after completion of a command.

### 1.3 LEADING PARTICULARS

#### 1.3.1 Mechanical

Construction : Single Eurocard PCB  
Size : 100mm x 160mm

#### 1.3.2 Power Supplies

+5V at 150mA.

#### 1.3.3 Connections

Pin connections are given in Section 5.

Connectors : Double-sided edge connector to Acorn Bus. TTL signal levels are used, OV to +0.4V = logic 0, +2.4V to +5.25V = logic 1.

32-way ribbon cable to disc drive unit. TTL signal levels are used, OV to +0.4V = logic 0, +2.4V to +5.25V = logic 1. Outgoing signals are fed via drivers, providing up to 48mA at the logic 0 level.

## 2. CIRCUIT DESCRIPTION

The circuit diagram for the Floppy Disc Controller Board, Figure 8, is filed at the end of this section.

### 2.1 CPU INTERFACE

#### 2.1.1 Addressing

The Floppy Disc Controller Board is addressed by the CPU on address lines A7 to A11 and the Block 0 signal OXXX. These signals are decoded by IC4 to provide the chip select signal A<sub>00</sub>-A<sub>7F</sub> to the FDC (IC1). A<sub>00</sub>-AFF is gated with address bit A2 and IC2/11 and IC2/8 to provide a second chip select signal, DACK, to the FDC.

#### 2.1.2 Input/Output

The transfer of data between the FDC and the CPU is controlled by the read and write signals NRDS and NWDS from the CPU. These signals set the direction of data transfer and synchronize the transfer to or from the data bus lines, D0 to D7 (Figure 2).

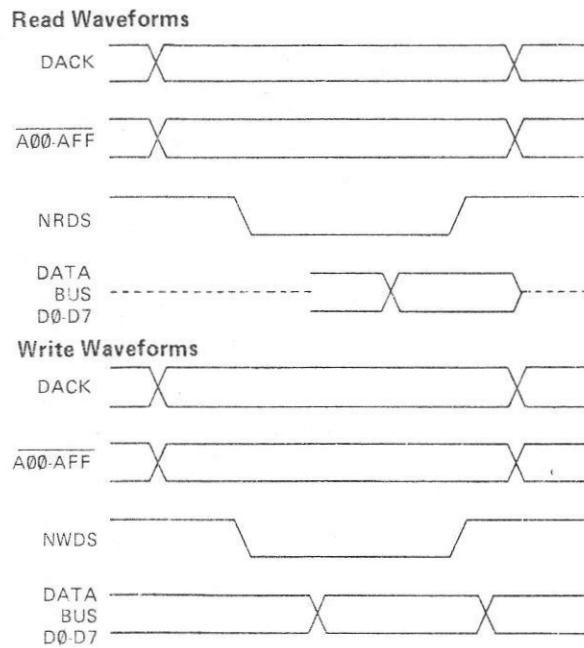
### 2.2 8271 FLOPPY DISC CONTROLLER

The internal structure of the Type 8271 FDC device, ICI is shown in block diagram form on Figure 1.

#### READ/WRITE CONTROL LOGIC

This logic provides address recognition from the CPU via address lines A0 to A2 and A7 to A11, the Block 0 signal OXXX and the Read and Write signals NRDS and NWDS. Commands and any associated parameters on the data bus lines D0 to D7 are passed to the Serial and Drive Interface Controllers and the register via the internal bus.

This logic also generates the Interrupt signal INT (inverted by IC3 to produce NNW) when the FDC is ready to receive or transmit data over the Data Bus.



!

## REGISTERS

The command and parameter data from the CPU, the status of the current data transfer, and the result of each transfer (successful or error detected) are held in these registers. The Status and Result Registers can be read by the CPU and the Reset Register set by a hardware or software Reset Command.

## DRIVE INTERFACE CONTROLLER

This logic receives information on the type of transfer between disc and FDC (read or write) and the location of the data to be read or stored. This is transferred to the disc drive via output buffers.

It also receives status information from the disc drives, via input buffers, on the initial position of the disc drive head(s), NTRK0, write protected discs, NWRPROT, and the Index signal NINDEX. It also receives the Drive Ready signals NREADY0 and NREADY1 from the Ready signals generator. This information is used to control the transfer of data via the Serial Interface Controller.

## SERIAL INTERFACE CONTROLLER

This logic provides the serial/parallel conversion of data between the 8-bit internal bus and the serial read and write lines to the Disc Drive as follows:

- Data to be written to disc is converted from an 8-bit parallel form to a serial stream with clock pulses. It is inverted and buffered by IC6/6 before being fed to the Disc Drive.

- Data read from disc consists of a serial stream with clock pulses. The Data Separator (Section 2.4) provides timing pulses which enable the Serial Interface Controller to remove the clock pulses and convert the serial data to an 8-bit parallel form.

## 2.3 CLOCK SIGNALS

The timing for the Floppy Disc Controller Board is provided by the 4MHz clock input from the CPU Board, or may be provided by the local oscillator.

### 2.3.1 4MHz Operation

The 4MHz clock is divided by IC8 to provide a 2MHz clock signal to the FDC and by IC8 and IC9 to provide a 15.625kHz signal to the Ready Signals Generator Circuit.

### 2.3.2 Local Oscillator Operation

The 4MHz oscillator (IC13 and its associated components), is utilized when operation from the §2 signal is required. IC13/8, RV1, and C5 form a 4MHz, free-running oscillator. The 4MHz output is synchronized to the rest of the system timing by the §2 signal. This is inverted by IC13/6 and fed via the differentiating circuit of R9, R10 and C6 to trigger IC13/8 at every second cycle of the 4MHz output.

### 2.3.3 Local Oscillator Setting-Up Procedure

The local oscillator, IC13/8 is set up by adjusting RV1 to give an output of 4MHz.

The following equipment is required:

Dual-beam oscilloscope  
Variable +5V power supply  
Frequency Generator

Connect the oscilloscope Channel A to IC13 pin 8. Connect the oscilloscope Channel B and the frequency generator to Side A pin 29.

Set the oscilloscope to:

Amplitude	2V /cm
Timebase	100ns/cm
Trigger	Channel B

Connect the power supply to the board edge connector as follows:

+5V to Side .4 pin 1  
0V to Side A pin 32

Set the power supply output to +5.0V.

Set the frequency generator to 1 MHz and at an output level which provides a useable display.

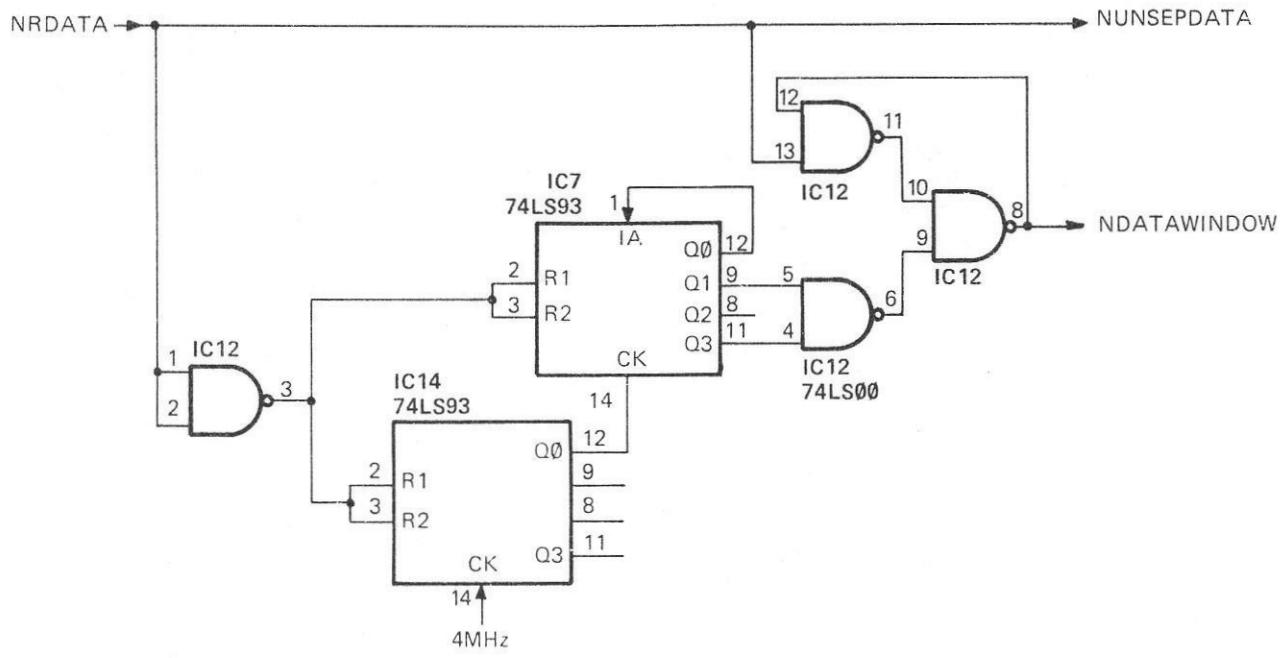
Adjust RV1 to give a stable 4MHz signal on IC13 pin 8 as seen on Channel A, in synchronization with Channel B.

In a 1MHz system the adjustment may be carried out using the system §2 clock signal instead of a signal generator.

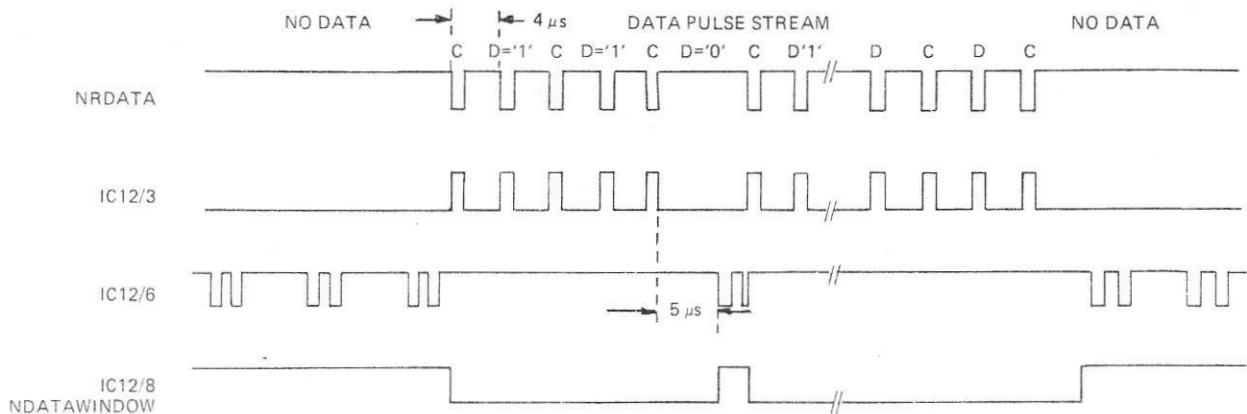
## 2.4 DATA SEPARATOR

The Data Separator Circuit is illustrated on Figure 3a. It provides identity pulses to the FDC, enabling it to extract the data bits from the combined clock and data pulse stream read off the disc.

The ripple counter, IC14 divides the 4MHz clock by 2 to provide a 2MHz clock signal to a second ripple counter, IC7. The  $\div 4$  (pin 11) and  $\div 16$  (pin 9) outputs of IC7 are NANDed by IC12/6 and inverted by IC12/8 to provide a positive going pulse to the FDC NDATAVVWINDOW input once every 5 $\mu$ s in the absence of clock and data pulses on the Unseparated Data line (Figure 3b).



(a) Circuit



(b) Timing

**Figure 3. Data Separator**

The first dock pulse in a data byte from the disc resets the ripple counters. These counters are reset again by each data logic '1' bit and all the clock pulses, holding the output of the Data Separator Circuit low. Data logic '0' bits do not reset the counters via IC12/3. This allows the counters to run on, producing a positive-going pulse, to the FDC after 5µs. IC12/11 latches the positive-going NDATAWINDOW signals, until the negative-going edge of the next clock pulse. At the end of a data pulse stream, IC12/11 inhibits the negative-going NDATAWINDOW signal until the next data stream is received.

## 2.5 READY SIGNALS GENERATOR

This circuit produces a ready signal to IC1 to indicate that the selected drive has a disc fitted and is rotating at approximately the correct speed. (Figure 4).

The 4MHz clock is divided by 256 by the dual counter IC8. The resulting 15.625kHz output is fed to the 14-stage ripple counter IC9. Thus, after approximately 262ms from power on the count will have reached 4096 and the Q12 output of IC9 (pin 2) is high.

When a disc drive with a disc installed is switched on (SEL0 or SEL1), the first NINDEX pulse, inverted by IC2/6, resets IC9 to zero. Its Q12 output goes low and the count recommences. If the next NINDEX pulse arrives before the count reaches 4096 (less than 262ms), the low output from IC9 (Q12) is clocked through the dual bistable IC11 to produce NREADY0

for disc drive 1 or through IC10 to produce NREADY1 for disc drive 2.

This process continues while the disc drive is running at the correct speed to produce a continuous Ready signal. As soon as the disc drive is deselected, the appropriate Select signal goes high, setting the corresponding bistable IC10 or IC11, thus disabling the Ready signal.

## 2.6 FLOPPY DISC CONTROLLER/DRIVE INTERFACE

Unless stated otherwise, all signals are active low (0V). The signals from the FDC to the drive unit(s) are driven via open collector NAND buffer gates IC5 and IC6 to achieve the power requirements of the interface. Absolute timing of signals across this interface depends on the type of drive in use. Typical signal timings are given on Figures 5 and 6.

### 2.6.1 Outputs from Floppy Disc Controller

DRIVE SELECT — NSEL0 AND NSEL1

These two lines are multiplexed to provide control of up to four disc drive units. In this application NSEL0 and NSEL1 are used to select the one or two drives equipped.

NMOTORON

This signal, in conjunction with NSEL0 and NSEL1 switches on the d.c. drive motors in the disc drive units. Writing to and reading from the discs are delayed by 250ms after NMOTORON becomes

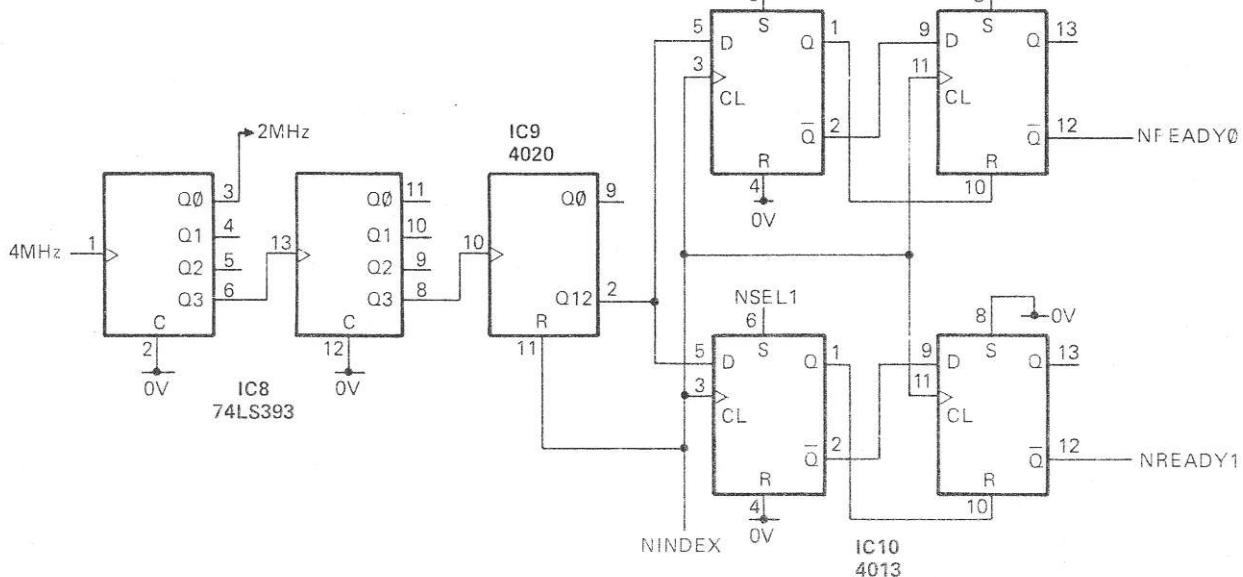
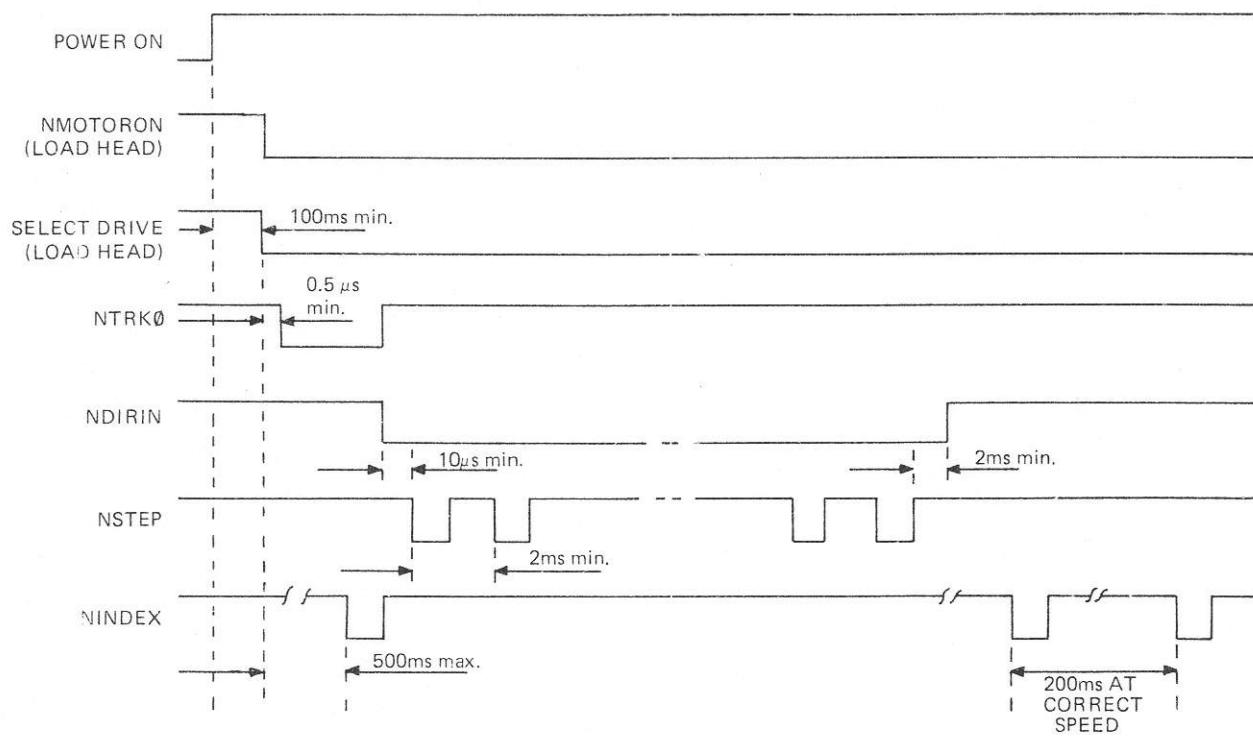
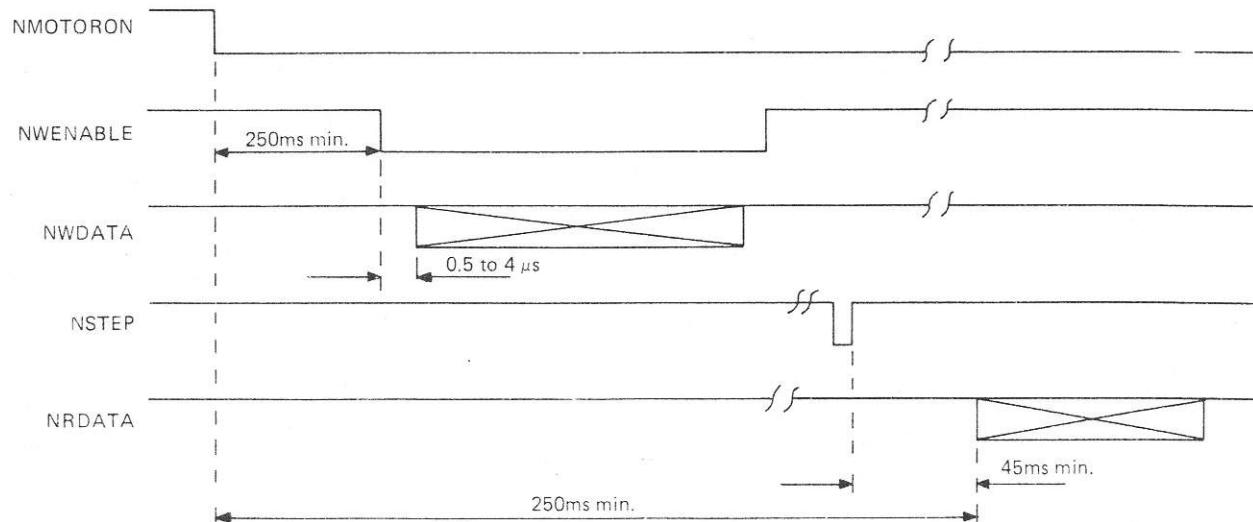


Figure 4. Ready Signals Generator



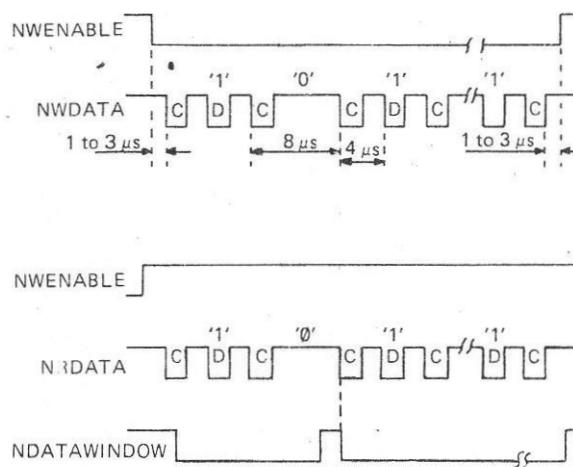
(a) Head Positioning



N.B. All timings quoted are typical and may vary between drive types.

(b) Write and Read

**Figure 5. Drive Timing**



Note: C = Clock Pulse; D = Data Pulse

**Figure 6. Data Timing**

active. During idle periods, 2s or more without a data transfer, the NMOTORON signal is disabled to increase motor life.

#### HEAD MOVEMENT DIRECTION — DIRIN

This Signal determines the direction of head movement when seeking a track. DIRIN low, in conjunction with NSTEP, causes the head to move towards the centre of the disc. DIRIN high and NSTEP cause the head to move out towards the edge of the disc.

#### NSTEP

Each negative-going, double NSTEP pulse causes the head to move one track position across the disc in the direction determined by the DIRIN signal.

#### READ/WRITE — NWENABLE

NWENABLE low enables the disc write circuits and, when high, enables the disc read circuits.

#### WRITE DATA — NWDATA

This line carries the serial write data to the disc drive units. Write data is enabled onto the selected disc drive unit by the NSEL0 and NSEL1 lines.

#### FAULT or OPO

This output is programmable by DOS to either reset a fault condition signalled by the disc drive or to select a disc surface on a double sided disc. In this application, it is used to select a disc surface. When FAULT is high, the left-hand surface (as viewed from the front of System 5) is selected: when low, FAULT selects the right-hand surface.

## 2.6.2 Inputs from Floppy Disc Drive Unit

### TRACK 00 — NTRK0

This signal is low when the head is precisely positioned on track 00.

### NINDEX

This line is pulsed low each time the disc index mark passes the index detector, once every 200ms. It should be noted that NINDEX is continuously active (low) while a disc is not fitted in the disc drive unit

### READ DATA — NRDATA

This line carries the raw serial read data, that is unseparated clock and data signals. Signal NRDATA goes low for each 1 bit (clock and data) read off the disc.

### WRITE PROTECT — NWPROT

The disc drive unit senses the insertion of a write protected disc and sets signal NWPROT low to prevent overwriting.

## 3. SOFTWARE PARAMETERS

The storage and retrieval of data on floppy disc is controlled entirely by the System 5 DOS supplied with the system. This control, the formatting of data, and the location of data on a particular disc is unalterable in use. Therefore only a brief description is given.

It should also be noted that the DOS resident in ROM supplied is specific to a particular manufacturer's disc drive unit and, that, where two disc drives are used, they must be of the same type and produced by the same manufacturer.

### 3.1. DOS

The operation of DOS (Disc Operating System) is fully described in the Disc Operating System document.

As soon as a disc drive with a disc loaded is selected, the disc catalogue held in Sectors 0 and 1 of Track 1 is loaded into the CPU's random access memory. This catalogue information is assumed to be valid while the disc is rotating. If the disc drive is stopped and restarted for any reason (e.g. disc not required or for a change of disc) the catalogue for the disc in use is reloaded to RAM.

The catalogue contains the disc identity, and the names, size (number of bytes) and location (track and sector) of the files currently held on the disc; double-sided discs contain separate catalogues on each sur-

face. This enables the CPU to control efficiently the transfer of data (files) to and from disc without having to search the whole disc for a file or the next vacant file position.

### 3.2 ADDRESSING

The Floppy Disc Controller Board is addressed by the CPU at  $A_0$  to  $A_7$  (hex) via address lines A7 to A11 and signal  $\overline{OXXX}$ . The various registers within the FDC device are accessed (command from CPU) using address lines  $A_0$  to  $A_2$  and the NRDS and NWDS signals as shown in Table 1. DACK corresponds to the chip select signal  $\overline{A_0-A_7}$  and A2, and NCS is the chip select signal  $\overline{A_0-A_7}$  generated from A7 to A11 and  $\overline{OXXX}$ .

NDACK	NCS	A1	A0	NRDS	NWDS	OPERATION
1	0	0	0	0	1	Read Status
1	0	0	0	1	0	Write Command
1	0	0	1	0	1	Read Result
1	0	0	1	1	0	Write Parameter
1	0	1	0	1	0	Write Reset Reg.
0	1	X	X	1	0	Write Data
0	1	X	X	0	1	Read Data
0	0	X	X	X	X	Not Allowed

Table 1. FDC Addressing

Up to five command parameters are transferred over the data bus lines D0 to D7. These parameters further define the type of command and identify the required drive disc surface, track and sector. This information is decoded by the FDC to provide the drive and surface select signals SEL and OPO to the disc drives. Further decoding provides the head positioning control signals DIRIN (head direction) and STEP (head stepping) as required.

### 3.3 DISC FORMAT

The storage of data on floppy disc is organised by disc and surface, track and sector (Figure 7) as follows:

- *Disc (and disc drive)*. Single or double-sided; one or both surfaces of a disc are used for recording data depending on the type of disc drive fitted.
- *Surface*. Each surface is formatted in 40 or 80 concentric tracks using the Utilities Disc FORM40 or FORM80 formatting program supplied depending on the type of disc drive in use.

- *Track*. Each track is organised in ten sectors. Track 0 is reserved for the Catalogue File and tracks 1 to 39 or 1 to 79 are used for data storage.
- *Sectors*. Each sector consists of identity and data fields. A data field holds up to 256 bytes of data. The sectors are recorded on a +3 stagger to obtain file transfers in the minimum number of disc revolutions (Figure 7).

Thus each surface can hold up to 100K bytes of data in 40 track format or 200K bytes of data in 80 track format.

Data is organized in files (data records) with up to 31 files to a disc surface. The number of files is limited by the size of the Catalogue to 31.

A file may be of any length up to the maximum capacity of the disc surface. A file always commences at the beginning of a sector and any unused space at the end of the last sector of a file is filled with random information.

The disc index hole together with a photo-electric detector in the disc drive provides the INDEX signal to the FDC once per disc revolution and also locates the beginning of the first sector on each track.

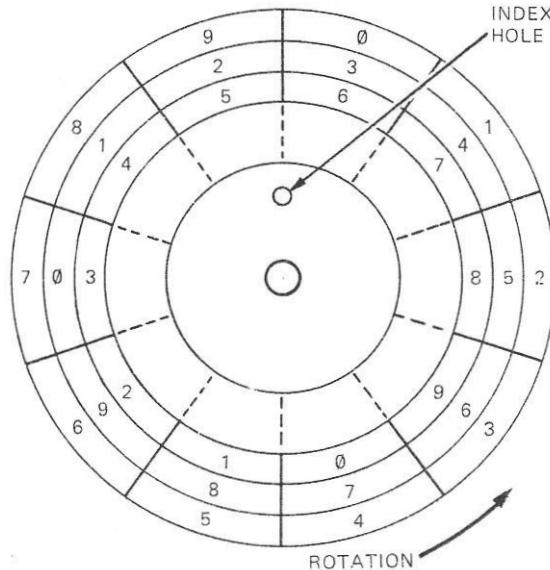


Figure 7. Data Format

### 3.4 DISC ERRORS

The FDC is capable of detecting many error conditions which could arise during the transfer of data to and from disc. These errors are interpreted by software which produces a disc error to the user, either on the VDU or printer, as shown in Table 2.

USER ERRORS		SYSTEM ERRORS
Crunch	: An attempt has been made to format a Write Protected disc.	Disc error 08 : During a disc Read operation a clock bit was missing.
Disc Full	: A block large enough to hold the file cannot be found. Periodic housekeeping with the COMPACT utility will minimise the occurrence of this error.	Disc error 0A : During a disc transfer the processor did not respond fast enough (probably due to a faulty Floppy Disc Interface Board).
Disc Prot	: A Write operation has been attempted to a Write Protected disc.	Disc error 0C : The CRC (cyclic redundancy check) derived from the ID data read back, differed from that which was originally loaded to the disc.
Drive ?	: Drive number is incorrectly (or not specified).	Disc error 0E : The CRC derived from the data read back on a disc read differed from that loaded to the disc.
File ?	: File name cannot be found.	Disc error 10 : During a transfer the disc stopped rotating (probably due to a badly inserted disc).
File Prot	: A Write operation has been attempted to a Write Protected file.	Disc error 14 : The Floppy Disc Controller Interface Board failed to find Track 0. This error usually results from attempting to access an unformatted disc.
Full	: 31 files already exist on the current disc.	Disc error 18 : The Floppy Disc Controller Interface Board failed to find the required sector. This error usually results from either using an unformatted disc, or the disc being corrupted by magnetic fields etc.
Syntax ?	: Command is recognized but Syntax error exists.	

Table 2. Error Messages

#### 4. FLOPPY DISC CONTROLLER CONNECTIONS

##### 4.1 FLOPPY DISC CONTROLLER BOARD TO ACORN BUS

Pin	Mnemonic	Meaning	I/O
<b>Side A</b>			
1	+5V	+5V Line	I
2	A15	Address Lines	I
3	A14	15 Not Used	I
4	NWDS	Write Select	I
5	NRDS	Read Select	I
6	NRST	Reset	I
7	A8		
8	A7		
9	A6		
10	A5	Address Lines	I
11	A4	6 Not Used	I
12	A3	5 Used	I
13	A2		
14	A1		
15	A0		
16	D7		
17	D6		
18	D5		
19	D4		
20	D3	Data Lines	I/O
21	D2		
22	D1		
23	D0		
24	A13		
25	A12	13 Not Used	I
26	A11	12 Used	I
27	A10		
28	A9		
29	Φ2	System Φ2 Clock	I
30	R/NW	Read/Write Enable – Not Used	I
31	NBLOCK0	Board Enable from CPU	I
32	0V	Earth	I
<b>Side B</b>			
16	4MHz	4MHz Clock	I
29	NNMI	Interrupt from FDC	O

##### 4.2 FLOPPY DISC CONTROLLER BOARD TO FLOPPY DISC DRIVE (DAISY CHAINED TO ALL DRIVES)

Pin	Mnemonic	Meaning	I/O
2	—	—	—
4	—	—	—
6	—	—	—
8	NINDEX	Index Signal from Drive	I
10	NSEL0	Multiplexed Drive	O
12	NSEL1	Select Signals (1 of 4)	O
14	—	—	—
16	NMOTORON	Motor Drive Enable	O
18	DIRIN	Head Movement Direction	O
20	NSTEP	Moves Head 1 Track in Direction Set by DIRIN	O
22	NWDATA	Data to Disc	O
24	NWENABLE	Enable Disc Write Circuits	O
		Enables Disc Read Circuits when High	
26	NTRK0	Indicates Head is precisely positioned on Track 00	I
28	NWPROT	Disc is Write protected	I
30	NRDATA	Data from Disc	I
32	OPO	Selects Disc Surface for Double Sided Drives. User Programmable.	O
34	—	—	—

Odd numbered Pins connected to Earth.

## 5. PARTS LIST

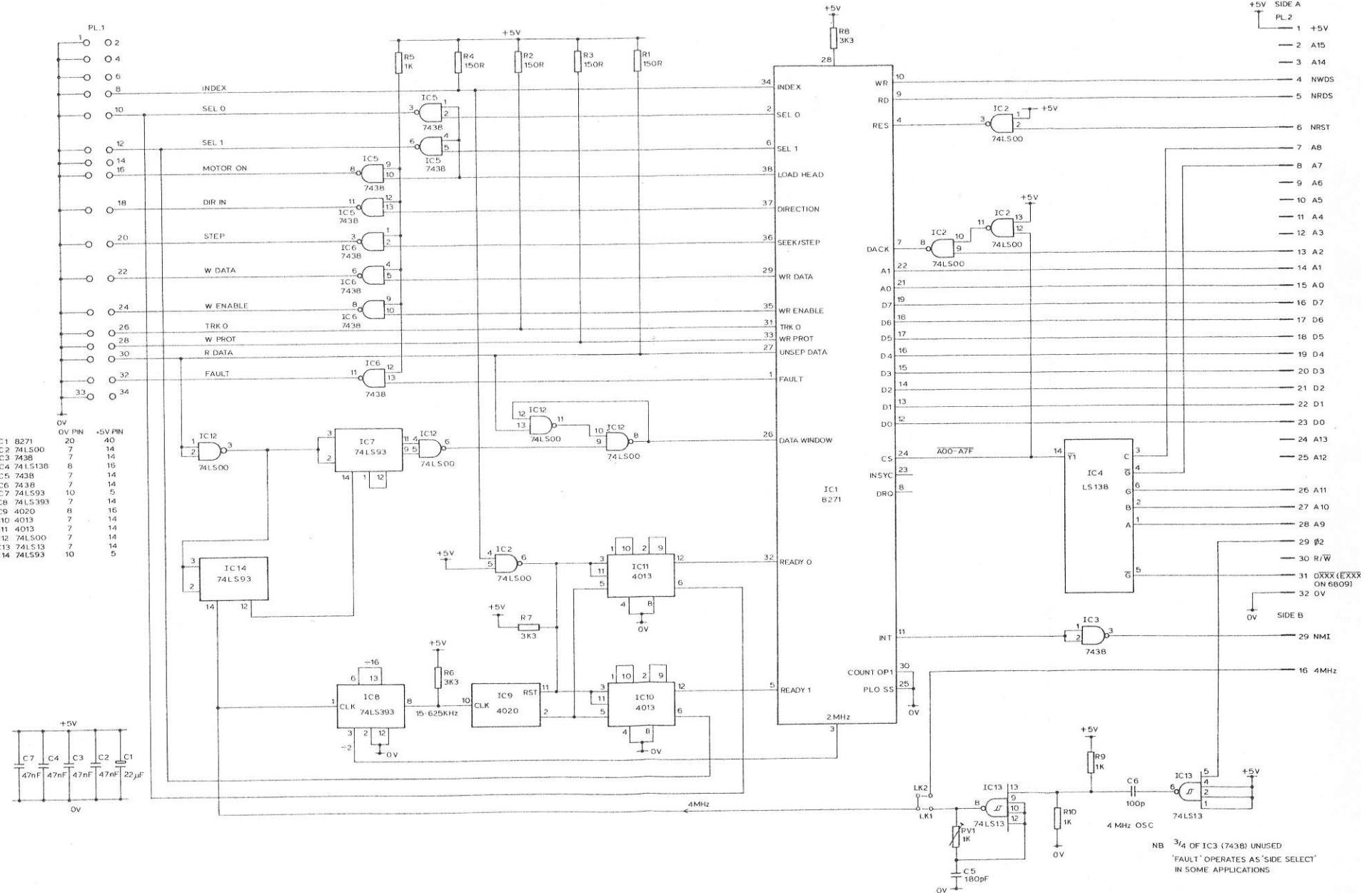
### 5.1 MECHANICAL PARTS

ITEM	DESCRIPTION	VALUE	QTY	PART NO.
	Printed Circuit Board, 200.004		1	
	Connector, 34-way		1	
	Connector, 64-way		1	
	40-pin IC Socket		1	
	16-pin IC Socket 2			
	14-pin IC Socket 10			

### 5.2 ELECTRICAL PARTS

ITEM	DESCRIPTION	VALUE	QTY	PART NO.
C1	Capacitor	15µF	1	
C2...C4	Capacitor	47nF	3	
C5	*Capacitor	180pF	1	
C6	*Capacitor	100pF	1	
C7	Capacitor	47nF	1	
IC1	Integrated circuit	8271	1	
IC2	Integrated circuit	74LS00	1	
IC3	Integrated circuit	7438	1	
IC4	Integrated circuit	74LS138	1	
IC5,IC6	Integrated circuit	7438	2	
IC7	Integrated circuit	74LS93	1	
IC8	Integrated circuit	74LS393	1	
IC9	Integrated circuit	4020	1	
IC10,IC11	Integrated circuit	4013	2	
IC12	Integrated circuit	74LS00	1	
IC13	*Integrated circuit	74LS13	1	
IC14	Integrated circuit	74LS93	1	
R1..R4	Resistor	150	4	
R5	Resistor	1K	1	
R6...R8	Resistor	3K3	3	
R9, R10	*Resistor	1K	2	
RV1	* Resistor, variable	1K	1	

\*These components are not used in the Acorn System 5 application.



**Figure 8 Floppy Disc Controller Circuit Diagram**

## ACORN KEYBOARD

### UNIT DESCRIPTION

#### CONTENTS

Paragraph		Page
1.	INTRODUCTION	1
1.1	Principles of Operation	1
1.2	Leading Particulars	2
1.2.1	Mechanical	2
1.2.2	Power Supplies	2
1.2.3	Signals	2
2.	CIRCUIT DESCRIPTION	2
2.1	Encoder	2
2.2	Reformat Circuit	3
2.3	Repeat Circuit	3
2.4	System Reset	4
2.5	User Programmable Key	4
3.	SOFTWARE PARAMETERS	4
4.	KEYBOARD CONNECTIONS	4
5.	PARTS LIST	6
5.1	Mechanical Parts	6
5.2	Electrical Parts	6
5.3	Power-on Reset Option	6

#### TABLES

Table No.		Page
1.	Key Codes	5

#### FIGURES

Figure No.		Page
1.	Keypad	1
2.	Keyboard Block Diagram	2
3.	Keyboard Circuit Diagram	7

©1982 Acorn Computers Limited, 12 Fulbourn Road, Cherry Hinton, Cambridge.  
Written and Produced for Acorn Computers Limited by Arch Technical Publications Limited,  
St. Lukes House, 5, Walsworth Road, Hitchin, Herts.

## 1. INTRODUCTION

The Acorn Keyboard provides a 62-key mechanism (Figure 1) using a QWERTY format. The keyboard is supplied in an injection moulded plastic case, and can be connected directly to a dedicated port on the Acorn 6502A CPU Board using a 20-way ribbon cable. Depressed key identity is transmitted to the CPU in 7 bit ASCII code.

Four special function keys provide the following facilities:

- ↑↓, referred to as the Caps key: This key functions as a shift key for alphabetic characters only so that only upper case letters are produced, as required when working in languages such as BASIC.
- REPEAT: The output lines always carry the code for the last key that was pressed, even if the key has been released. The REPEAT key gives repeated codes for that key to the CPU. An auto repeat feature starts the repeat circuit automatically if a key is held down for longer than 0.5 seconds.
- BREAK: This key provides a system reset function.
- ⇨, referred to as the Blank key: This key is undefined and can be user programmed via the CPU port.

Two other keys, CTRL (Control) and SHIFT, change the character set produced by the keyboard encoder circuit to give ASCII control characters and upper/ lower case selection respectively.

Three LED indicators on the keyboard provide an indication of Power On, Caps Lock mode and Shift Lock mode.

An optional timer circuit provides a Power-on Reset function.

### 1.1 PRINCIPLES OF OPERATION

A block diagram of the Acorn Keyboard is given on Figure 2.

The keyboard encoder device is capable of encoding up to 112 keys on a 16 x 8 matrix. It detects a depressed key and outputs a 10 bit code, representing the identity of the depressed key, and a flag signal for the duration of the key depression. Further circuits on the keyboard reformat the 10 bit code into an ASCII, 7 bit code. The key depressed flag signal is fed through a repeat key code circuit before being added to the seven ASCII data bits to form an 8 bit key data byte.

The key data byte in Acorn Systems is fed to Port A on the CPU Board. This port is periodically polled and if the flag bit (NSTROBE) is set, the key code is read. However, the key code remains valid on the port until reset by the next key depression. Thus, it is only necessary to set the NSTROBE signal again (by means of the Repeat Circuit) to obtain the Repeat facility.

The Caps key forms part of the reformatting circuit, automatically amending the ASCII code to ensure that, in the Caps mode any lower case alphabetic codes are converted to upper case.

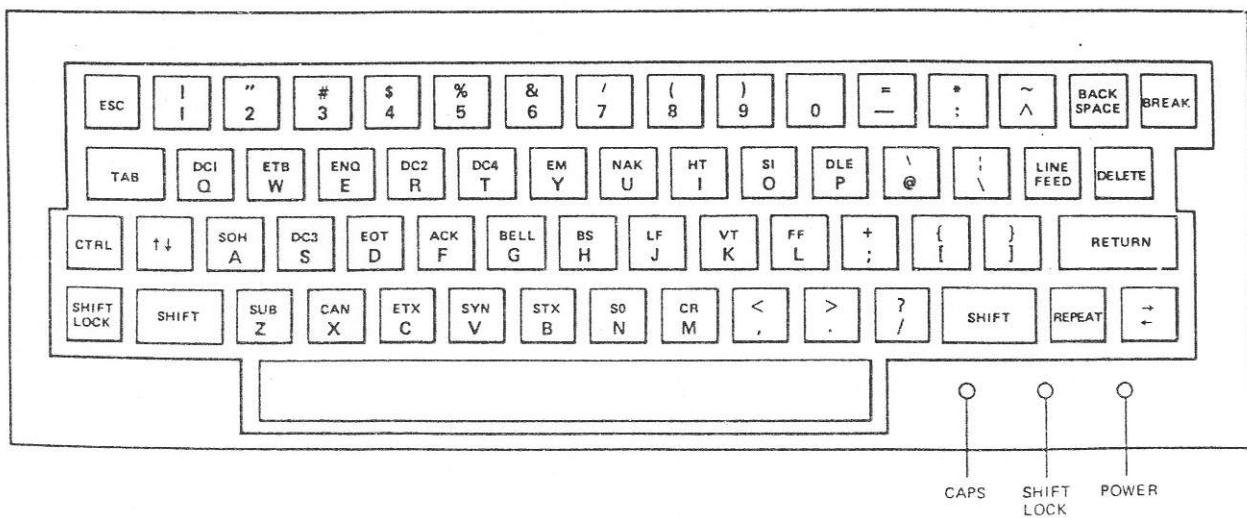
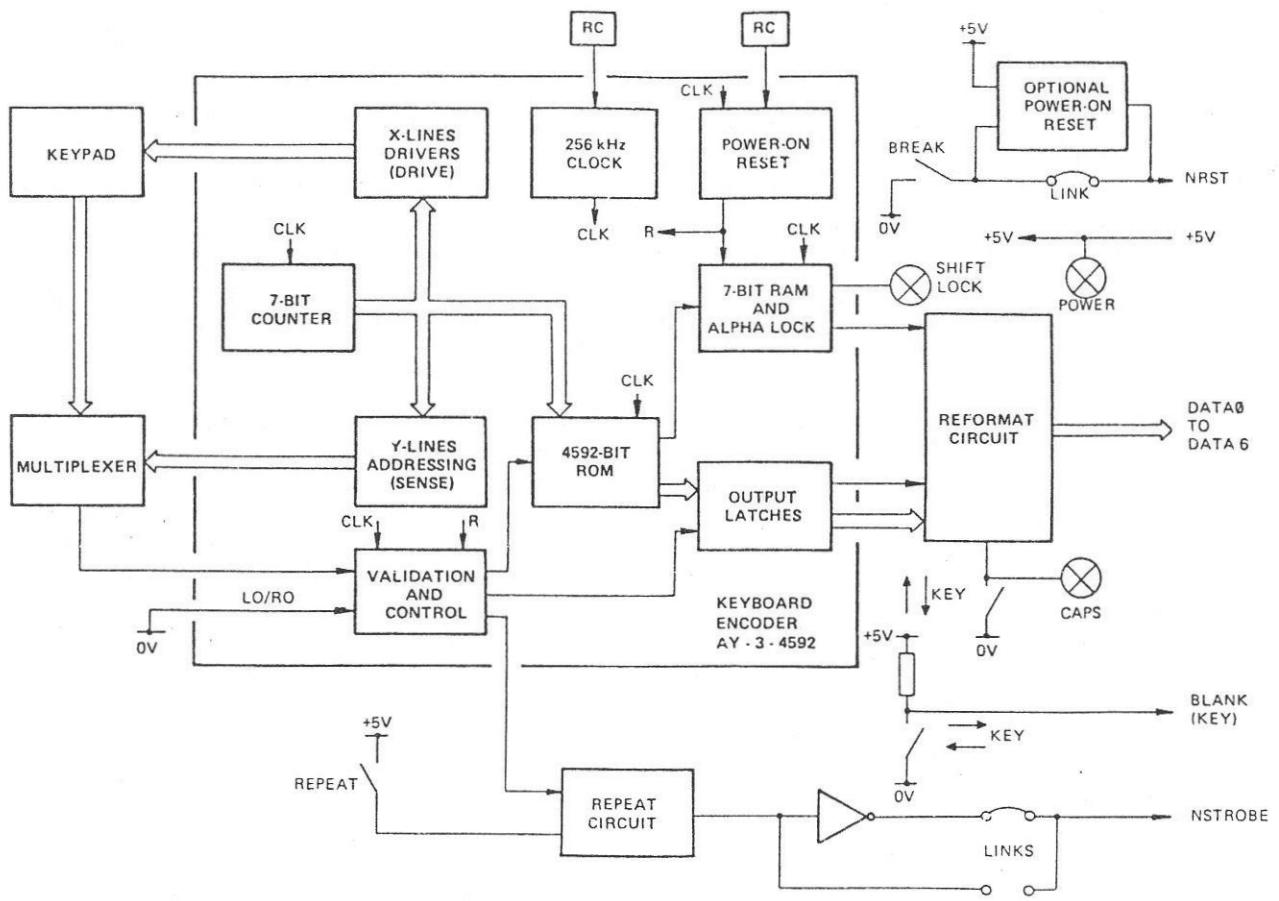


Figure 1. Keypad



**Figure 2. Keyboard Block Diagram**

The BREAK key forms part of the optional Reset Circuit when fitted. The Blank to 10 port bit user programmable key is connected directly to Port A on the CPU Board.

## 1.2 LEADING PARTICULARS

### 1.2.1 Mechanical

Construction	:Special printed circuit board carrying 62 keys.
Housing	:Injection moulded ABS case.
Size	:PCB and keypad: 355mm x 150mm, overall height 30mm. Case: 380mm x 240mm, overall height 60mm.

### 1.2.2 Power Supplies

+5V at 100mA maximum.

### 1.2.3 Connections

Output Signals : 20-way flying lead ribbon header-and power supplies.

Alternatively, connection may be made to the 20-way PCB edge connector.

Output Signals : Logic '0' = 0V to +0.4V  
Logic '1' = +2.4V to +5.25V.

## 2. CIRCUIT DESCRIPTION

The circuit diagram for the Acorn Keyboard is at the end of this section.

### 2.1 ENCODER

The keys are connected to the keyboard encoder device, IC1 on a 13 x 8 matrix. Scanning of the matrix is performed by the encoder in conjunction with an

external 3 to 8 multiplexer, IC2. The encoder provides a 3 bit binary address (YA, YB and YC) which is used cyclically to scan each of the eight possible sense lines (Y lines). The drive lines (X lines) are cyclically pulsed low by the encoder. If a key is depressed, the pulse is coupled from the drive line to the sense line at that key's matrix crosspoint and via IC2 to the NMATIN input of the encoder.

Each matrix crosspoint is given a unique binary code that is determined by internal scan counters. This code is loaded into a 7 bit latch and compared with the matrix code generated by the depressed key on the next scan. If the two codes match, a valid key depression, the matrix code is used to address the 4592 bit, ROM which holds the encoder output codes. The output code is loaded into the output holding register (output pins D1 to D10). At the same time, the code match generates the Any Key Depressed flag (AKD).

If a second key is depressed before the first key has been released, the AKD output goes low (inactive) for two clock cycles (8 $\mu$ s) and the code for the second key is latched onto the output lines.

A two key rollover facility is provided by connecting pin 13 (LO/RO) to 0V. This modifies the foregoing key detection action to encode the second key depression when two keys are depressed almost simultaneously (first key miskeyed) within the space of one matrix scan. When the matrix crosspoint codes are compared, a mismatch occurs and the code generated at the second scan is assumed to be the valid code.

Timing for the encoder is provided by an internal clock generator whose frequency is set to 256kHz by the external components R1 and C1. This results in an effective (double) keyboard scantime of 8ms. The keyboard inhibit facility (pin 12, KBINH) is not used.

At Power On, pin 37 (POR) is held low for approximately 100ms until C2 charges through R2. This action clears all the internal registers and bistables. The SHIFT LOCK key, when depressed, produces an additional output, SLI (pin 34). This is used to light the SHIFT LOCK indicator via inverter, IC/8.

Circuits within the encoder device guard against errors due to simultaneous depression of two or more keys and provide key debouncing.

## 2.2 REFORMAT CIRCUIT

This circuit takes eight bits (D1 to D5, D7 and D6 or D9) from the 10 bit output code and reformats them into a 7 bit ASCII/ISO/CCITT No. 5 Telegraphy/Data Transmission code.

Output code bits D1 to D5 and D7 are inverted by the hex inverters IC5 and IC6/6 to produce the ASCII code bits DATA0 to DATA4 and DATA6 respectively.

In the Caps Lock mode (Caps key depressed), the low input to IC4 pin 12 from the Caps key inhibits IC4/11 and thus output code data bit D6 (alphabetic lower case indicator bit). The same low input is inverted by IC4/6, enabling IC4/8 and the feeding of output code data bit D9 (alphabetic upper case indicator bit) via IC4/8 and IC4/3 (double inversion) to the output inverter IC5/12 as DATA5 bit. The Caps key depressed also lights the CAPS LOCK indicator (LED D2) via resistor R8.

In all other modes, the inputs to IC4/6 and pin 12 of IC4 are pulled high via R8 and D2. This inhibits output code bit D9 via IC4/8 and enables output code bit D6 via IC4/11 and IC4/3 (double inversion) and the output inverter IC5/12 (DATA5).

## 2.3 REPEAT CIRCUIT

This circuit (IC3) provides two Repeat Last Key facilities. It causes the key depressed coded to be retransmitted at a rate of 10Hz if the depressed key is held down for longer than 0.5s or if the REPEAT key is operated after the release of the depressed key and before operation of another key.

Initially, signal AKD from IC1 is low, C3 is discharged to AKD through D1, and the input to pin 2 of IC3 is high.

When a key is depressed, AKD goes high, is inverted by IC3/3, and fed via IC6/2 and IC6/4 (double inversion) to the output pin 18 (NSTROBE). For applications requiring a positive strobe, the printed circuit track from IC6 pin 4 to output pin 18 should be cut and a link made between IC6 pin 2 to output pin 18.

While a key is depressed, capacitor C3 begins to change towards AKD through R6 and R3. This is ineffectual unless the key is held down for longer than 0.5s, the time constant of C3 and R6.

If a key is held down for longer than 0.5s, C3 charges fully and the pin 8 input to IC3 goes high. The pin 9 input to IC3 is already high due to the action of C4.

R5/VR1, and IC3/4 and the output of IC3/10 goes low inhibiting IC3/3, and setting NSTROBE high. Since IC3/10 output is low and IC3/3 output is high, C3 reverse charges (in 100ms), the output of IC3/4 goes low enabling IC3/3 again via IC3/10 and resetting NSTROBE low. This oscillatory action of C4, IC3/4 and IC3 at 10Hz (100ms) continues until the depressed key is released.

Depression of the REPEAT key immediately charges C3, repeating the foregoing operation without the initial 0.5s delay required to charge C3.

#### 2.4 SYSTEM RESET

The System Reset pulse (NRST) is generated by operation of the BREAK key which connects the NRST output pin to 0V. The link between the BREAK key and the NRST output pin must be connected for this type of operation.

Alternatively, the NRST pulse can be generated at power on by the optional timer circuit of IC7 and at other times by the action of the BREAK key in conjunction with the timer. The link is disconnected for this type of operation.

The timer is connected in the monostable mode with R12 and C9 providing the timing, and the BREAK key connected to the trigger input (pin 2).

At power on, the trigger input is low (C8 discharged), setting the internal circuits so that the output (pin 3) is high inverted by IC6/10 to produce NRST low) and allowing C9 to charge through R12. After approximately 100ms C9 (and C8) will have charged, resetting the internal circuits to give a low output and discharging C9. Thus the NRST line is held low for 100ms to provide the System Reset at power on.

When the BREAK key is depressed to generate a user controlled system Reset, C8 is discharged, and the trigger input goes low. This results in the generation of the 100ms NRST pulse as described above.

#### 2.5 USER PROGRAMMABLE KEY

The Blank user programmable key is connected directly to pin 5 of the edge connector. With the key in the OFF state, this pin is pulled high by resistor R7.

### 3. SOFTWARE PARAMETERS

The following table defines the ASCII 7 bit codes produced by depressed keys under the five basic modes of operation:

<i>Normal</i>	Unshifted (lower case) character set.
<i>Shifted</i>	Upper case character set.
<i>Control</i>	Control codes.
<i>Shift/Control</i>	The character set and control codes produced when the SHIFT and CONTROL keys are operated.
<i>Caps Lock</i>	The numeric/upper case alphabetic and symbol character set produced by operation of the Caps key.

### 4. KEYBOARD CONNECTIONS

The Acorn Keyboard outgoing signals and incoming power supply are connected via a 20-way header cable. The remote end of this cable is identified as CPU Pin in the following list.

Key-board Pin	CPU Pin	Circuit Mnemonic	Meaning	I/O
1	11	0V	Earth	-
2	-	-		-
3	-	-		-
4	-	-		-
5	1	'BLANK'	Customer/Software specified	O
6	--	-		-
7	-	-		-
8	--	-		-
9	-	-		-
10	-	-		-
11	10	DATA0		O
12	12	DATA1		O
13	14	DATA2	Key Identity	O
14	16	DATA3	7 bit ASCII	O
15	18	DATA4	Code	O
16	20	DATA5		O
17	19	DATA6		O
18	17	NSTROBE	Key Operation flag	O
19	15	NRST	System Reset	O
20	13	+5V	+5V Line	1

NORMAL		SHIFT		CONTROL		SHIFT & CONTROL		TTY CAPS	
CHARACTER	HEX	CHARACTER	HEX	CHARACTER	HEX	CHARACTER	HEX	CHARACTER	HEX
TAB	09	TAB	09	TAB	09	TAB	09	TAB	09
LINE FEED	0A	LINE FEED	0A	LINE FEED	0A	LINE FEED	0A	LINE FEED	0A
RETURN	0D	RETURN	0D	RETURN	0D	RETURN	0D	RETURN	0D
ESC	1B	ESC	1B	ESC	1B	ESC	1B	ESC	1B
SPACE BAR	20	SPACE BAR	20	SPACE BAR	20	SPACE BAR	20	SPACE BAR	20
0	30	0	30	0	30	0	30	0	30
1	31	!	21	1	31	!	21	1	31
2	32	"	22	2	32	"	22	2	32
3	33	#	23	3	33	#	23	3	33
4	34	\$	24	4	34	\$	24	4	34
5	35	%	25	5	35	%	25	5	35
6	36	&	26	6	36	&	26	6	36
7	37	'	27	7	37	'	27	7	37
8	38	(	28	8	38	(	28	8	38
9	39	)	29	9	39	)	29	9	39
a	61	A	41	SOH	01	SOH	01	A	41
b	62	B	42	STX	02	STX	02	B	42
c	63	C	43	ETX	03	ETX	03	C	43
d	64	D	44	EOT	04	EOT	04	D	44
e	65	E	45	ENQ	05	ENQ	05	E	45
f	66	F	46	ACK	06	ACK	06	F	46
g	67	G	47	BELL	07	BELL	07	G	47
h	68	H	48	BS	08	BS	08	H	48
i	69	I	49	HT	09	HT	09	I	49
j	6A	J	4A	LF	0A	LF	0A	J	4A
k	6B	K	4B	VT	0B	VT	0B	K	4B
l	6C	L	4C	FF	0C	FF	0C	L	4C
m	6D	M	4D	CR	0D	CR	0D	M	4D
n	6E	N	4E	SO	0E	SO	0E	N	4E
o	6F	O	4F	S1	0F	S1	0F	O	4F
p	70	P	50	DLE	10	DLE	10	P	50
q	71	Q	51	DC1	11	DC1	11	Q	51
r	72	R	52	DC2	12	DC2	12	R	52
s	73	S	53	DC3	13	DC3	13	S	53
t	74	T	54	DC4	14	DC4	14	T	54
u	75	U	55	NAK	15	NAK	15	U	55
v	76	V	56	SYN	16	SYN	16	V	56
w	77	W	57	ETB	17	ETB	17	W	57
x	78	X	58	CAN	18	CAN	18	X	58
y	79	Y	59	EM	19	EM	19	Y	59
z	7A	Z	5A	SUB	1A	SUB	1A	Z	5A
,	2C	<	3C	,	2C	<	3C	,	2C
- (1)	2D	- (1)	5F	US (1)	1F	- (1)	5F	- (1)	2D
.	2E	>	3E	.	2E	>	3E	.	2E
/	2F	?	3F	/	2F	?	3F	/	2F
:	3A	*	2A	:	3A	*	2A	:	3A
;	3B	+	2B	;	3B	+	2B	;	3B
@	40	\	5C	NUL	00	NUL	00	@	40
[	5B	(	7B	ESC	1B	ESC	1B	[	5B
\	5C	:	7C	FS	1C	FS	1C	\	5C
]	5D	)	7D	GS	1D	GS	1D	]	5D
^	5E	~	7E	RS	1E	RS	1E	^	5E
- (2)	2D	= (2)	3D	- (2)	2D	= (2)	3D	- (2)	2D
DELETE	7F	DELETE	7F	DELETE	7F	DELETE	7F	DELETE	7F

NOTES: 1. BACKSPACE key produces - (minus), \_ (underline), and US (Unit separation).

2. = key produces - (minus), and = (equals). (2)

Table 1. Key Codes

## 5. PARTS LIST

### 5.1 MECHANICAL PARTS

ITEM	DESCRIPTION	VALUE	QTY	PART NO.
	Printed Circuit Board 200,013		1	
	Keyboard, Hi-Tech 873-10492		1	
	Ribbon cable, 20-way		2 ft	
	IDC Socket, 20-way		1	
	Strain relief for above		1	
	Case assembly		1	

### 5.2 ELECTRICAL PARTS

ITEM	DESCRIPTION	VALUE	QTY	PART NO.
C1	Capacitor, disc ceramic	1nF	1	
C2	Capacitor, disc ceramic	100n F	1	
C3	Capacitor, tantalum	470nF	1	
C4...C6	Capacitor, disc ceramic	47nF	3	
C7	Capacitor, electrolytic	22μF	1	
D1	Diode	1N4148	1	
D2...D4	Light emitting diode	TIL 228, Red	3	
IC1	integrated circuit	AY-3-4592	1	
IC2	Integrated circuit	CD4051B	1	
IC3	Integrated circuit	CD4093B	1	
IC4	integrated circuit	74 LS00	1	
IC5, IC6	Integrated circuit	74 LS04	2	
R1	Resistor	10K	1	
R2	Resistor	100K	1	
R3	Resistor	10K	1	
R4	Resistor	100K	1	
R5	Resistor	1M	.1	
R6	Resistor,variable	1M	1	
R7	Resistor	4K7	1	
R8... R10	Resistor	270	3	

### 5.3 POWER—ON RESET OPTION

The following additional components are required to implement the Power-on Reset Option.

ITEM	DESCRIPTION	VALUE	QTY	PART NO.
C8...C10	Capacitor, disc ceramic	100n	3	
IC7	Integrated circuit	555	1	
R11, R12	Resistor	1M	2	

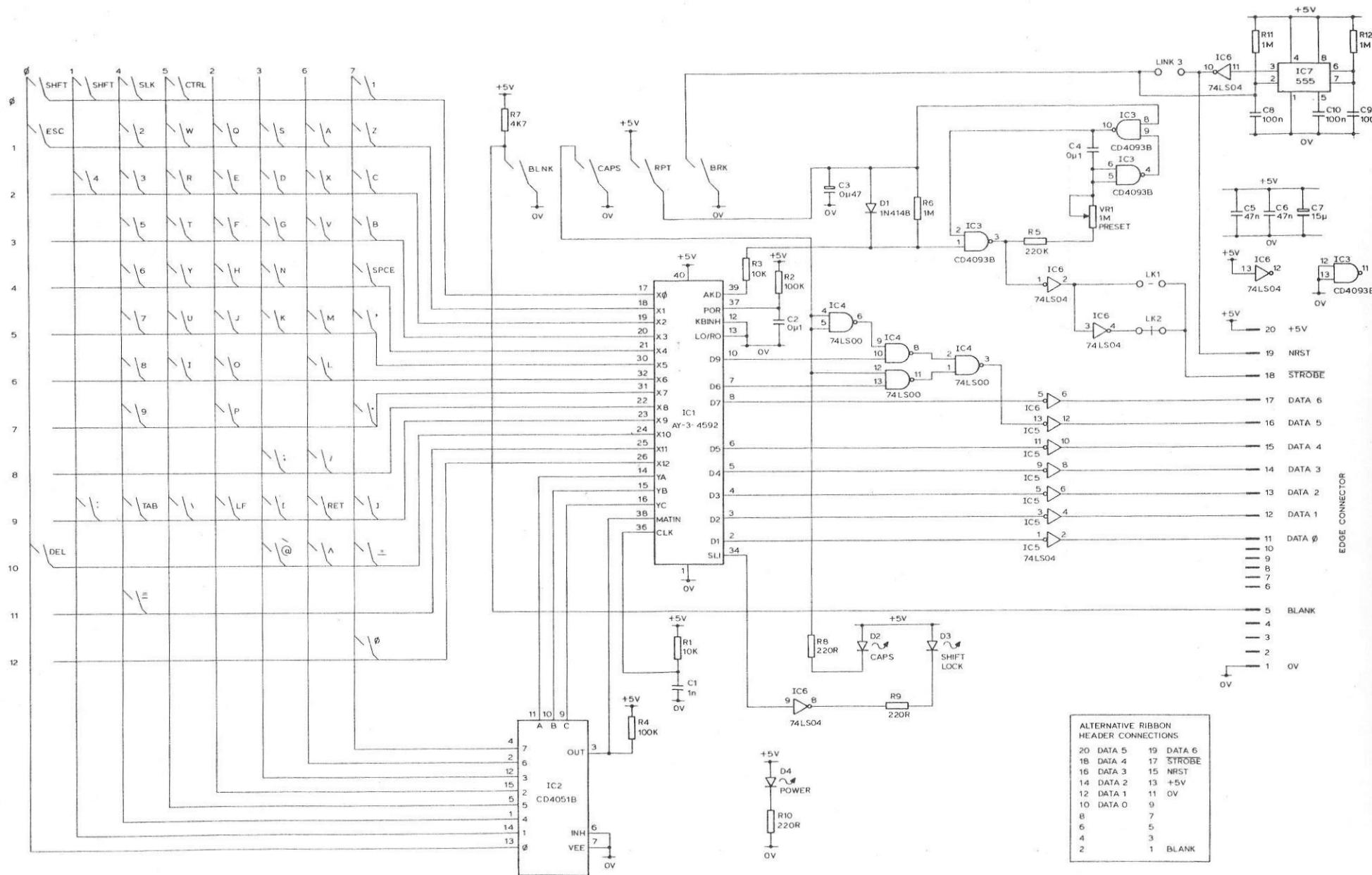


Figure 3. Keyboard Circuit Diagram

# ACORN ECONET BOARD

## UNIT DESCRIPTION

### CONTENTS

Paragraph		Page
1.	INTRODUCTION	1
1.1	General	1
1.2	Principles of Operation	1
1.3	Leading Particulars	1
1.3.1	Mechanical	1
1.3.2	Power Supplies	1
1.3.3	Connections	1
2.	CIRCUIT DESCRIPTION	1
2.1	Station Identification and Selection	1
2.2	Line Driver	2
2.3	Line Receiver	2
2.4	Collision Detect	2
2.5	Clock Valid Circuit	2
2.6	ADLC Serial Control Clip	2
2.7	Address Decoding	2
3.	SOFTWARE PARAMETERS	2
3.1	Econet Software	2
3.2	Econet Enabling	2
4.	ECONET BOARD CONNECTIONS	3
4.1	Econet Board to Acorn Bus	3
4.2	Econet Board to Line Connector	3
5.	PARTS LIST	3
5.1	Mechanical Parts	3
5.2	Electrical Parts	3

### TABLES

Table No.		Page
1.	Station Selection	2

### FIGURES

Figure No.		Page
1.	Econet Eurocard Circuit Diagram	5

©1982 Acorn Computers Limited, 12 Fulbourn Road, Cherry Hinton, Cambridge.  
Written and Produced for Acorn Computers Limited by Arch Technical Publications Limited,  
5, Walsworth Road, Hitchin, Herts.

## 1. INTRODUCTION

### 1.1 GENERAL

The Acorn Econet board allows System 2/3/4/5 to be connected to the Econet local area network. The Acorn Econet allows computers to communicate and therefore to share expensive resources such as printers and disc filing systems. The Econet system is primarily intended for use in schools and colleges but also lends itself to many office and business applications.

The Econet is a bus network and transmission of information uses two twisted pairs of wires. Data is transmitted in serial bit form over one pair and the other pair transmits a clock signal which is shared by all stations. All stations on the network have equal status and there is no central controller. The network is therefore tolerant against failure since a failure in one computer still allows other computers to use the network. Access to the network is by carrier sense multiple access with collision detect (CSMA/CD). The network can be 500 metres in length and computers simply tap into the main network cable.

This section describes the main details relevant to the Econet eurocard interface. For installation of an Econet network the user is referred to the Econet system and user manual.

### 1.2 PRINCIPLES OF OPERATION

The operation of the eurocard board can be divided into six main areas:-

- (1) The ADLC control chip – This handles the serializing and deserializing of information for the main CPU. The ADLC also handles such complex functions as automatic flag detection and synchronization, zero insertion and deletion, automatic CRC generation and checking, and idle detection.
- (2) The line driver – this is responsible for driving the differential twisted pair data lines.
- (3) The line receivers – These take serial differential signals on the clock and data lines for input to the ADLC control chip.
- (4) Collision detect circuit – This has a dual purpose. First it indicates when the network data lines are undriven. Second it indicates when a current transmission has interference from other simultaneous transmissions.
- (5) Clock valid circuit – This circuit provides an input to the Econet software to indicate the main network clock signal is present.
- (6) Station selection – this allows the selection of a unique 8-bit network address for the Econet interface.

Essentially the Econet board is an interface between the parallel Acorn bus and the serial Econet network. The ADLC chip controls much of the serial bit stream needed to implement the Econet protocols. For more information on the ADLC chip the user is referred to the Motorola MC6854 technical manual.

### 1.3 LEADING PARTICULARS

#### 1.3.1 Mechanical

Construction : Single Eurocard printed circuit board.  
Size : 100mm x 160mm

#### 1.3.2 Power supplies

+5V +/-5% at 250 mA typically.

#### 1.3.3 Connections

Connectors : Double sided Eurocard edge connector to Acorn Bus, TTL signal levels are used, OV to 0.4V = logic '0', +2.4V = logic '1'.  
180 Deg. 5 pin Din socket for connection to Econet.

## 2 CIRCUIT DESCRIPTION

The circuit diagram for the Econet eurocard board is shown in Figure 1.

### 2.1 STATION IDENTIFICATION AND SELECTION

The network number of each station is set up by hardware links on the eurocard board. The links are connected to the octal buffer IC 7 and when this is enabled the CPU is able to read the station ID. The links are in two parallel rows of molex pins and make up an eight bit binary station address from 0-255. Each station on the network must have a unique address. Note station numbers 0 and 255 are reserved and must not be used. Stations 235 and 254 are normally reserved for the printer server and file server respectively. The Econet software will examine the address of every message on the network and compare it with the hardwired address in its own station. A station number is selected by placing molex links on LK 0 to LK 7. Note LK 0 is the least significant bit and LK 7 is the most significant bit. Place a molex shunt in

the positions that require a '0' bit and leave the other positions empty. Table 1 shows a station with address 205 (decimal) selected.

LK	7	0	0
LK	6	0	0
LK	5	0	---
LK	4	0	---
LK	3	0	0
LK	2	0	0
LK	1	0	---
LK	0	0	0

**Table 1. Station Selection**

## 2.2 LINE DRIVER

The transmission of data onto the network is via IC1 a 75159 dual differential line driver. The line driver is to the RS 422 specification and can source or sink 40 mA. Differential line techniques ensure both minimal radiation and high noise immunity for the network. The line driver is enabled by Not Request To Send output from the ADLC when Data is transmitted. When disabled or powered down the line driver outputs are high impedance. Stations can therefore be left connected to the network even when they are not in use.

## 2.3 LINE RECEIVER

The reception of data and clock signals from the network is accomplished using IC2 an LM 319 dual fast comparator. The receive circuits for the clock and data are identical and are designed to give good common mode signal rejection and hysteresis thus providing noise immunity. The reference voltage for the line receivers is two volts provided by R20, R21, and C3. Incoming signal voltages are divided by ten by R4, R5, R8, R11 and R2, R3, R7, R10 for the clock and data respectively. Hysteresis is provided by R15 and R14 for the clock and data signals respectively.

## 2.4 COLLISION DETECT

The collision detect circuit uses IC4, an LM 319 dual fast comparator. Essentially the circuit is a window comparator and the collision signal is fed into the ADLC Not Clear To Send input. This signal indicates when the data lines are driven by a differential signal greater than one volt. The collision threshold voltage is set by R1. During idle conditions the lines are driven by a 0.6 volts line bias from the network terminator units. This voltage allows the collision detect circuit to indicate the line is free but also feeds a constant stream of binary ones into the ADLC control chip to signal that the line is idle. In the driven state the differential voltage on the data lines is above two volts.

## 2.5 CLOCK VALID CIRCUIT

The clock valid circuit provides an input to the Not Data Carrier Detect input of the ADLC. The Econet software is therefore able to check that a valid clock exists on the network before transmissions take place. The valid clock condition is sensed using a retriggerable monostable IC 5. The minimum clock rate of the Econet is therefore determined by this monostable time constant and is currently set at 70 kbits/sec.

## 2.6 ADLC SERIAL CONTROL CHIP

The ADLC control chip is the main component of the Econet board and it contains most of the functions needed to implement the bit oriented protocol of the Econet. The ADLC chip handles such functions as bit serializing and deserializing of data for the main CPU. The ADLC also handles such complex funtions as automatic flag detection and synchronization, zero insertion and deletion, automatic CRC generation and checking, and also idle detection. In addition to relieving the software from such time and code consuming functions the ADLC contains a number of useful input and outputs that simplify the hardware. The Not Request To Send output is used to enable the Econet data line driver, the. Not Clear To Send input is used to detect an undriven line, and the Not Data Carrier Detect is used to detect an absence of clock signal.

## 2.7 ADDRESS DECODING

The Econet eurocard board takes up five bytes of address space from 1940 to 1944 (hex). The ADLC is mapped from 1940 to 1943 and the station identification is mapped at 1944 (hex).

## 3. SOFTWARE PARAMETERS

### 3.1 ECONET SOFTWARE

The Econet software is 4kbytes long and is supplied in EPROM form. On System 5 computers the Econet is conveniently combined with the DOS (Disc Operating System) and is supplied in an 8kbyte EPROM. The EPROM is fitted on the main CPU card and is mapped from E000 to FFFF (hex).

### 3.2 ECONET ENABLING

Before the Econet can be used on system 5 computers it must be enabled. The enabling action allows the Econet software to initialize the ADLC chip, internal variables, and setup interrupt jump addresses. Initialization is achieved using \*GO command which can be typed by the user or executed from user software. There are three main addresses depending on what Econet resources are required.

\*GOE000 — init. only the Econet low level primitives.

\*GOE003 — init. low level, Econet printer, the Econet filing system.

\*GOE009 — init. low level, Econet printer, but allow use of a Disc filing system.

#### 4. ECONET BOARD CONNECTIONS

##### 4.1 ECONET BOARD TO ACORN BUS

Pin	Mnemonic	Meaning	I/O
Side A			
1	+5V	+5V Line	I
2	A15	Address Lines	I
3	A14		
6	NRST	Reset	I
7	A8		
8	A7		
9	A6		
10	A5		
11	A4	Address Lines	I
12	A3		
13	A2		
14	A1		
15	A0		
16	D7		
17	D6		

Pin	Mnemonic	Meaning	I/O
18	D5		
19	D4	Data Lines	I/O
20	D3		
21	D2		
22	D1		
23	D0		
24	A13		
25	A12		
26	A11	Address Lines	I
27	A10		
28	A9		
29	Φ2		
30	R/W	Read/Write Enable	I/O
32	OV	Earth	
Side B			
24	VMA	Valid Memory Address	I
28	IRQ	Interrupt Request	I

##### 4.2 ECONET BOARD TO LINE CONNECTOR

Pin	Mnemonic	Meaning	I/O
1	DATA	—	I/O
4		—	—
2	GROUND	—	—
3	CLOCK	—	I
5		—	—

#### 5. PARTS LIST

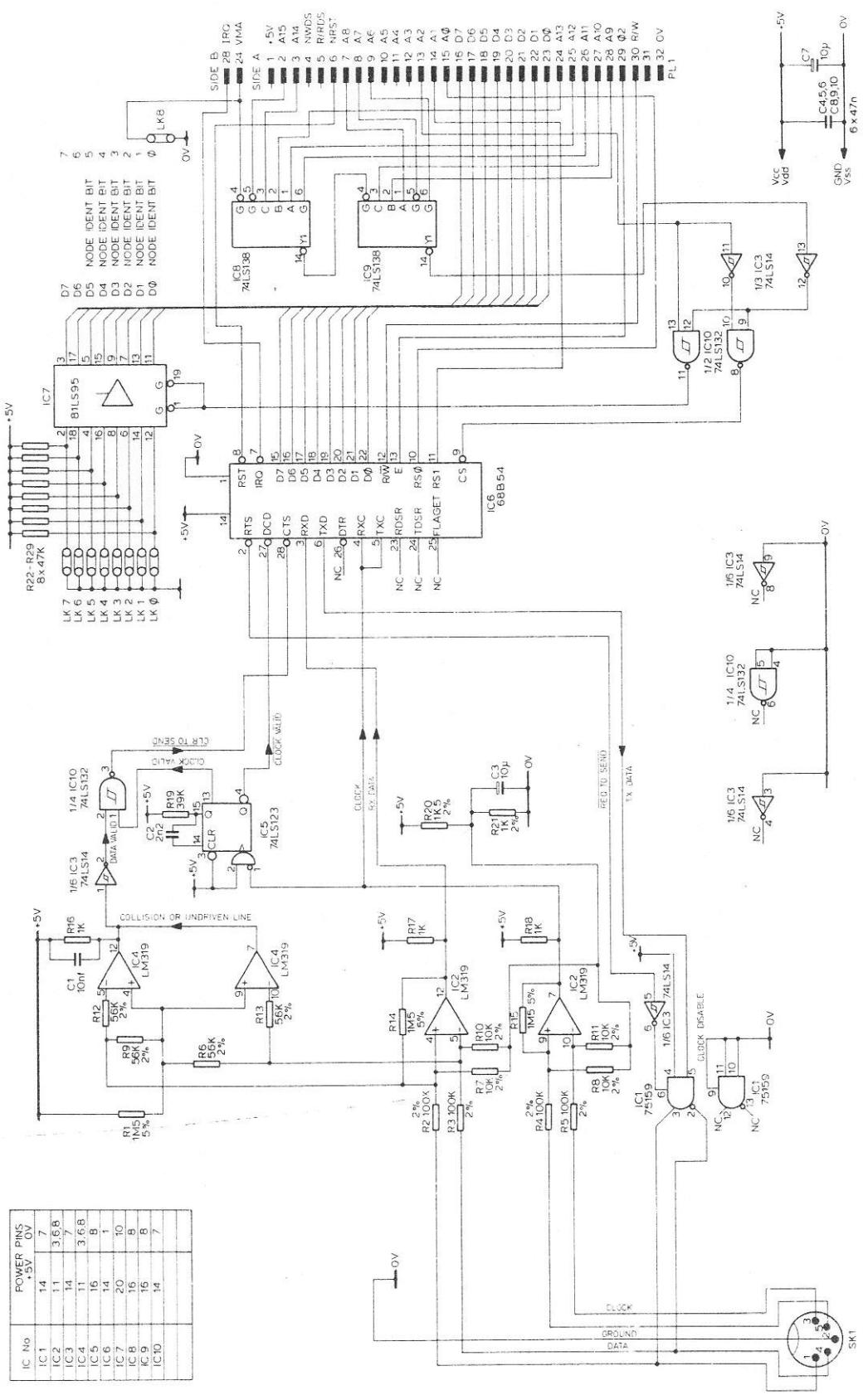
##### 5.1 MECHANICAL PARTS

ITEM	DESCRIPTION	VALUE	QTY.	PART NO.
	Front Panel Kit	1		100,524
	Shunt 2 Way	7		800,070
	Molex Pins 8 Way	2		800,054
	28 Pin IC Socket	1		800,128
	14 Pin IC Socket	3		800,114
	Printed Circuit Board	1		200,024
PL1	Connector, 64 Way	1		800,401
SK1	DIN Socket, 5 Way	1		800,004

##### 5.2 ELECTRICAL PARTS

ITEM	DESCRIPTION	VALUE	QTY.	PART NO.
IC1	Integrated Circuit	75159	1	735,159
IC2/IC4	Integrated Circuit	LM319	2	770,319
IC3	Integrated Circuit	74LS14	1	742,014

ITEM	DESCRIPTION	VALUE	QTY.	PART NO.
IC5	Integrated Circuit	74LS123	1	742,123
IC6	Integrated Circuit	68B54	1	706,854
IC7	Integrated Circuit	81LS95	1	738,095
IC8/IC9	Integrated Circuit	74LS138	2	742,138
IC10	Integrated Circuit	74LS132	1	742,132
C1	Capacitor	10nF	1	629,010
C2	Capacitor	2n2	1	629,002
C3/C7	Capacitor, Electrolytic	10 $\mu$ 1 6V	2	635,100
C4.....C6,C8.....C10	Capacitor	47nF	6	680,001
R1, 14, 15	Resistor	1M5	3	
R2...R5	Resistor 2%	100K	4	505,104
R6, 9, 12, 13	Resistor 2%	56K	4	505,563
R7, 8, 10, 11	Resistor 2%	10K	4	505,103
R16, 17, 18	Resistor	1K	3	500,102
R19	Resistor	39K	1	500,393
R20	Resistor 2%	1K5	1	505,152
R21	Resistor 2%	1K	1	505,102



**Figure 1. Econet Eurocard Circuit Diagram**