Acorn Computers Ltd.
British Broadcasting Corporation Microcomputer
Functional differences between Project B and Models B & B+

Operating system changes:

New Series 3 OS for the Project B - 128/512/Sc.
New Series 4 OS for the Project B - Econet Terminal.

Memory map changes

The Snadow screen in the Model B+ from &3000 - &7FFF (Sideways) is present in the Project B machine. The 12K of additional Paged RAM in Model B+ is implemented differently in the Project B machine and is reserved for Operating and Filing system use as "Private RAM" (see below).

The minimum machine configuration is 128K, where 64K is allocated to Sideways RAM (4 by 16K pages) in a similar way to the 128K Model B+. Use of internally fitted ROM's in two of the three sockets will require changes to the link settings which will remove access to some of the Sideways RAM. (They are ROM locations 4,5,6 & 7).

Memory map below OSHWM, with changes from Model B/B+ indicated:

Page zero - &00-&FF
&00-&8F Language workspace. BASIC allows &70-&8F for the user.
&90-&9F ECONET workspace. Do not use.
&A0-&A7 NMI workspace. It must be claimed before use, the owner must:

a) Have a Filing System number allocated to it. Note the error in the Advanced User Guide (Bray, Dickens & Holmes) on P. 323. The NMI ID passed around by Service Calls &B and &C are Filing System numbers not ROM numbers.

b) Be able to process ROM Service Calls &B and &C i.e. they must be ROM's or intercept OSBYTE &8F.

&A8-&AF MOS scratch space.
&B0-&BF Filing System scratch space. Watch out for "hidden" filing system calls i.e. those produced by OSWRCH if *SPOOL used.
&C0-&CF Current Filing System PRIVATE workspace.
&D0-&FF MOS workspace only (see Advanced Reference Guide, published by Acorn, for accurate decription of the VDU Driver use).

Page one - &100-&1FF
Machine stack and error message buffer.

Page two - &200-&2FF
&200-&235 Vectors.
&236-&28F Main MOS variables.
&290-&2FF MOS workspace (for MOS only).

Page three - &300-&3FF
&300-&37F VDU variables (for use by graphics routines only).

In the Model B, B+ and this machine, Page three is used for VDU workspace. Most variables are the same, with the following exceptions:

	Model B & B+	This machine
&359 -	Foreground graphics colour	Plotting foreground/background
&35A -	Background graphics colour	Current graphics plot mode
&366 -	Mode 7 cursor	Cursor control flags (VDU 23,16)
&367 -	Exploded font flag	Dotted line pattern (VDU23,6)
&368 -	Exploded font location bytes	Current dotted line state
&369 -	" " " "	Plot colour (0-solid,Not 0-pattern)
&36A -	" " " "	F/grnd col. (0-solid,Not 0-pattern)
&36B -	" " " "	B/grnd col. (0-solid,Not 0-pattern)
&36C -	" " " "	Col. 81 flag (top bit set pending)
&36D -	" " " "	Foreground graphics colour
&36E -	" " " "	Background graphics colour

&380-&3DF Cassette Filing System workspace (do not use).

Page four, five, six and seven - &400-7FF

Language workspace. Do not use unless you are the current language or the current language has allowed you to use it.

Page eight - &800-&8FF

&800-&87F Sound workspace. DO NOT USE (unless you want strange noises!).
&880-&8BF Printer buffer. Useable if, and only if, no printing needed.
&8C0-&8FF Sound workspace (envelopes 1-4). Useable if, and only if, envelopes not needed.

Page nine - &900-&9FF

&900-&9BF RS 423 O/P buffer, or Cassette O/P buffer (1st part), or Sound workspace (envelopes 5 - 16). Do not use for anything else, as this area could be re-allocated.
&9C0-&9FF Speech buffer, or Cassette O/P buffer (2nd part). Do not use for anything else, as this area could be re-allocated.

Page ten - &A00-&AFF

RS 423 I/P buffer, or Cassette I/P buffer. Do not use for anything else, as this area could be re-allocated.

Page eleven - &B00-&BFF

ECONET workspace. Do not use.

In the Model B & B+ this is used for the soft key buffer. In this machine, the soft key buffer resides in the Private 12K RAM and should not be accessed by the user. Programs directly accessing Page eleven on the Model B & B+ for soft-key purposes will be incompatible with this machine

Note: Initial Page states are:
Model B, B+: &B00-&BFF - &10
This machine: &B00-&BFF - &00

Page twelve - &C00-&CFF

ECONET workspace. Do not use.

In the Model B & B+ this is used for user defined characters in the range ASCII 224-255. In this machine, all characters up to ASCII 255 are defined with a standard font. This definition resides in the Private RAM area and should not be accessed by the user (the user can, however, re-define all of them as before with VDU23). Programs directly accessing Page twelve on the Model B & B+ for user defined characters will be incompatible with this machine.

Note: Initial Page states are:
Model B, B+: &C00 - &0D &C01-&CFF - &00
This machine: &C00-&CFF - &00

Page thirteen - &D00-&DFF

&D00-&D5F NMI routine and workspace. NMI's must be claimed to use this area.
&D60-&D9E ECONET private workspace. Do not use.
&D9F-&DEF Expanded vector set.
&DF0-&DFF Paged ROM workspace, 1 byte per ROM.

12K Private RAM:

&8000-&83FF Soft-key buffer. Do not use directly.
&8400-&87FF VDU workspace. Reserved for routines that need a large area such as Flood-fill. Do not use directly.
&8800-&88FF VDU variables and workspace. Do not use directly.
&8900-&8FFF Character definitions. Do not use directly.
&C000-&DCFF Paged ROM workspace, claimed via a Service Call.
&DD00-&DFFF MOS workspace. Do not use.

Changes in the firmware are detailed as follows:

Operating System

As version 1.2 with following extensions:

* Shadow Screen - Use of MODE 128 to 135 instead of MODE 0 to 7 results in the use of a section of memory for the screen separate from the main area of RAM. This allows more room for user Progs.
* Real-time clock - the status of a Real-time clock can be called from the MOS. The OSWORD calls &E and &F Read & Write the clock in BCD or text formats.
* Default command line interpreter if no language present or *GO used.
* Configure CMOS RAM commands - the state of the CMOS RAM can be set for both rserved and unreserved Bits. Reserved Bits include:

* Start-up mode	MODE <0-7,128-135>
* File-Server station number	FS <0-255[.0-255]>
* Printer-Server station number	PS <0-255[.0-255]>
* Econet station ID	Transient Command
* Start-up language ROM number	LANG <ROM>
* Start-up filing system ROM number	FILE <ROM>
* *TV position & Interlace state	TV [<Dec>[,<Dec>]
* Auto-repeat delay	DELAY <0-255>
* Auto-repeat rate	REPEAT <0-255>
* Default printer	PRINT <0-255>
* Print ignore char.(no param=no ignore)	IGNORE <0-255>
* Tube selected (Internal/External)	EXTUBE/INTUBE
* Tube ON/OFF	NOTUBE/TUBE
* Serial baud rate (Both ways)	BAUD <1-8>
* Serial data format	DATA <0-7>
* Boot status	BOOT/NOBOOT
* Scroll state (on/off)	SCROLL/NOSCROLL
* Floppy-Drive params (speed etc)	FDRIVE <0-7>
* Floppy or Winchester start-up	FLOPPY/HARD
* ADFS or FADFS as default	DIR/NODIR
* Bell character volume	LOUD/QUIET
* Caps lock on/off	CAPS/NOCAPS

FS & PS status bits only operational when ANFS fitted.

Configuration is set using *CONFIGURE <param> or read using *STATUS <param> or just *STATUS to read all.

* Soft characters fully exploded
* Improved soft-key structure with more buffer area.
* New resident commands:
    * APPEND - functions as BUILD but appends to end of file.
    * BUILD - as DFS but control codes input by | mechanism.
    * CLOSE - closes open files on current filing system.
    * CONFIGURE - To set-up start-up options.
    * CREATE - creates empty file using *SAVE params.
    * DELETE - as DFS delete.
    * SHOW - displays soft-key contents.
    * DUMP - <start in file> <start address on output>.
    * EX - Examine file. (similar to *INFO)
    * EXEC - as DFS.
    * GO - to enter address in language/IO processor.
    * GOIO - to enter program at address in IO processor.
    * IGNORE - as *FX6.
    * INFO - as DFS.
    * INSERT - inserts ROM number n into ROM map from reset.
    * LIBFS - Define FS where LIB is from current FS.
    * MOVE - copies files from one name to another incl between filing systems.
    * PRINT - as TYPE but gets/processes vdu codes.
    * REMOVE - as DELETE but no error msg if not found.

    * ROMS - lists ROM names, skts, vers & UNPLUG/PLUG.
    * SHADOW - with 0 or no param, gives shadow on next mode.
      With 1, drops shadow on next mode.
    * SHUT - Close open files on all filing systems.
    * SPOOLON - functions as SPOOL but appends to end of file.
    * STATUS - Lists status of start-up options.
    * TYPE - as DFS but control codes displayed as |D etc.
    * UNPLUG - removes " " " " " " " "

* Numeric keypad - options to set base for range of values returned and set whether SHIFT/CTRL etc affect codes.
* Cassette filing system upgraded to do OSGBPB (calls 1 & 3) in those cases where it is to write to or read from the current pointer.
  OSFSC extended for CFS to provide call 7 so that it returns legal file handle range.
  CFS will perform * EX (but not * INFO).
  CFS can be reselected by *FX143,18,<n> where n=1 (1200 baud), n=2 (300 baud, n=3 (ROM filing system),
* Multifiling system capability. Name of filing system can be prefixed to the file name itself. Files may be opened and kept open across several filing systems.
  Filing names currently defined include:
  CFS (or TAPE), ROM, DFS, ADFS, NET
* Extended graphics facilities:
    * Extended Fill commands using checkerboard pattern/stipple.
    * Mark-to-space ratio of dotted line.
    * New triangle algorithm for correct plotting under all conditions.
    * Parallelogram - Solid fill.
    * Rectangle - Solid fill.
    * Circle - Outline.
    * Circle - Solid fill.
    * Arc - Line fill.
    * Arc - Solid fill (Pie shape).
    * Arc - Solid fill (Chord segment).
    * Ellipse - Outline.
    * Ellipse - Solid fill.
    * Fill enclosing shape outline (flood-fill).
    * Enhanced speed of all fill operations.
    * Move/copy rectangle.
    * Extended horizontal line fill.
* Extension text:
    * Improved character draw speed (especially VDU5).
    * Clear block of text window.
    * Output character with no cursor move (dead key).
    * Advance left/right after drawing character.
    * Pending scroll to allow bottom right character to be drawn.
    * Cursor options held across mode changes.
    * Direct window scroll - left/right/up/down.
* Extra 128 standard characters in ASCI range 128 to 255.

Detailed changes:
Miscellaneous:

1) Sideways ROM headers which have been "illegal" previously on Models B & B+, but accepted by the MOS, may not be accepted by this MOS version.
2) The screen Paged Mode algorithm has been improved in this MOS, resulting in slight differences in the number of displayed lines in some modes compared with Model B & B+.
3) In the Econet Terminal, the User VIA chip (6522) is not normally

fitted. Application software intended for Econet use should not use the VIA timer.

4) The RAM latch at &FE34 has some changes and additions to the use of each bit. Refer to the Advanced Reference Guide for more information.

OSBYTE calls:

&0 (0) - Enter with X<>0, Returns X=3 for Project B New
Entering with X=0, (or *FX0) displays OS version

&14 (20) - Parameters for *FX20 are now ignored and *FX20
resets standard exploded font. Software writers
must add parameters as required for Model B & B+.

&16 (22) - Increment ROM polling semaphore. New
Used to request MOS polling with service call
21 every 10mS (polling with semaphore non-zero)

&17 (23) - Decrement ROM polling semaphore. New
Used to stop MOS polling with service call 21
every 10mS

&6B (107) - Switch Internal/External 1Mnz Bus New
*FX107,0 - Select external bus (default).
*FX107,1 - Select internal (cartridge) bus.

&6C (108) - Switch Main/Shadow memory into main map New
*FX108,0 - Switch Shadow memory into main
map area, &3000-&7FFF (immediate).
*FX108,1 - Switch Main memory into main map
area (immediate).

&6D (109) - Make temporary Filing system permanent New

&70 (112) - Write to Main/Shadow memory: New
*FX112,0 - Write to memory specified by mode
change.
*FX112,1 - Write to main memory (immediate)
*FX112,2 - Write to shadow memory (immediate)

&71 (113) - Display Main/Shadow memory: New
*FX113,0 - Display memory specified by mode
change.
*FX113,1 - Display main memory (immediate)
*FX113,2 - Display shadow memory (immediate)

&72 (114) - Write to/Display Main/Shadow memory (*SHADOW) B+
*FX114,0 - Use Shadow memory at next mode
change.
*FX114,n - Use mode defined at next mode change.
(where n is 1-255)

&81 (129) - Now extended to return new OS version:
Enter with X<>0 & Y=255
Returns X=3 for OS 3.00 New

Enter with X=0
Returns X=253 for Project B New

&84 (132) - Read top of user RAM (was display RAM start). Change

&85 (133) - Read top of user RAM for a given mode (was display
RAM start for a given mode). Change

&A1 (161) - Read CMOS RAM New
Enter with A=0, X=n, where n is the RAM
location number (30-49).
Returns with result in Y
Use *STATUS for locations 0-29.

&A2 (162) - Write CMOS RAM New
Enter with A=1, X=n, where n is the RAM
location number (30-49). *FX162 can be used.
Use *CONFIGURE for locations 1-29. Location
0 is protected.

&A4 (164) - Check processor type. New
&A5 (165) - Read output cursor position. New
&B3 (179) - Read/Write ROM polling semaphore. Change
(was Read/Write OSHWM).
A=179, X=n, Y=0 reads semaphore into X and sets state to n.
Setting state directly with this call will interfere with OSBYTE 22 & 23 use.
A=179, X=0, Y=255 reads semaphore into X
&B6 (182) - Read NOIGNORE state (was Read font explosion) Change
&EE (238) - Changes numeric pad base: New
"0" key is based at 48, *FX238,<base> with base from 0-255 will alter key characters.
&FA (250) - Read memory area used for writing to New
&FB (251) - Read memory area used for reading from New
&FE (254) - Controls effect of SHIFT on numeric pad Change
*FX254,0 - makes SHIFT have effect
*FX254,<1-255> - deletes effect of SHIFT
(NOTE: This call returned RAM size in Model B & B+)

OSWORD calls:

&E (14) - Read CMOS clock New
&F (15) - Write to CMOS clock New

New Service calls to Sideways ROM's:

&15 (21) - Polling interrupt. Made 100 times per Sec. if OSBYTE 22 issued.

&18 (24) - Interactive HELP. Made by MOS when it executes a *HELP command, after service call 9.

&21 (33) - Indicate Static Workspace in Hidden RAM. Call is made on a Reset. Workspace starts at &C000 in Hidden RAM and can only be used by a Filing System, and only one at a time. Workspace has an upper limit of &DBFF.

&22 (34) - Claim Dynamic Workspace in Hidden RAM. ROM's should ideally ignore Call 2, which takes workspace in main memory.

&23 (35) - This is top of Static Workspace. Tells ROM's where top of Static Workspace in Hidden RAM is.

&24 (36) - Dynamic Workspace requirements. ROM's should indicate how much memory they will each claim through Call 34.

&25 (37) - Inform MOS of Filing System name and info.

&26 (38) - Close all files. Issued at a Reset. Filing systems should select themselves, close open files and then de-select.

&27 (39) - Reset has occurred. Call made after hard reset. Mainly for Econet Filing System so that it can claim NMI's.

&28 (40) - Unknown CONFIGURE option. Used to extend range of commands.

&29 (41) - Unknown STATUS option. Used to provide extra commands.

&2A (42) - ROM based language starting up.

New VDU commands:

VDU 18,m,c - Define graphics colour:
m = 0 to 4 same as 1.2 MOS.
m = 5. Leave screen colour unchanged.
For each of n = 1,2,3,4 (ecf pattern numbers):
m = 16n. Overwrite the colour on the screen
m = 16n + 1. OR the colour of the screen.
m = 16n + 2. AND the colour of the screen.
m = 16n + 3. EOR the colour of the screen.
m = 16n + 4. Invert the colour of the screen.
m = 16n + 5. Leave screen colour unchanged.

VDU 22,m - Select screen mode:
m = 0 to 8. As 1.2 MOS.

m = 128 to 135 covers shadow screen modes.

VDU 23,0,r,v,0,0,0,0,0,0 - Control 6845 CRTC directly
As 1.2 MOS i.e. writes value of v to 6845 register r.

VDU 23,1,n,0,0,0,0,0,0,0 - Turn cursor on/off.
As 1.2 MOS (but with additions of n=2 & n=3) i.e.:
n = 0. Stops cursor appearing.
n = 1. Cursor appears on screen (Default case).
n = 2. Cursor is steady.
n = 3. Cursor flashes at approx 1.5 times/sec (Default case).
Flash rate is doubled in cursor edit mode.

VDU 23,2-5,a,b,c,d,e,f,g,h - Set ecf pattern:
ecf patterns can be set to pixel groups of 8*8, 4*8 or 2*8 if mode has 2, 4 or 16 colours respectively. VDU 23,2 thru VDU 23,5 sets patterns 1 thru 4 respectively.
Integers a thru h define pattern rows from top to bottom. If the integer is derived from stuvwxyz in binary, then:
For 2 colour mode, logical colours from left to right are:
s, t, u, v, w, x, y, z
For 4 colour mode, logical colours from left to right are:
sw, tx, uy, vz
For 16 colour mode, logical colours from left to right are:
suwy, tvxz

VDU 23,6,n,0,0,0,0,0,0,0 - Set dotted lines pattern:
n = &FF. Solid line as in 1.2 MOS.
n = &AA. Dotted line as in 1.2 MOS.(Default-Reset every mode change)
n = &EE. Dashed line (dot-dot-dot-space repeated)
n = &E4. Dash-dotted line (dot-dot-dot-space-dot-space-space rept)

VDU 23,7,m,d,z,0,0,0,0,0 - Scroll window directly.
Allows text window or arbitrary rectangle to be scrolled without cursor movement:
m = 0. Scroll text window
m = 1. Scroll entire screen
d = 0. Scroll right d = 1. Scroll left
d = 2. Scroll down d = 3. Scroll up
d = 4. Scroll in positive X direction (defined by VDU 23,16, etc).
d = 5. Scroll in negative X direction (" " " " " ").
d = 6. Scroll in positive Y direction (" " " " " ").
d = 7. Scroll in negative Y direction (" " " " " ").
z = 0. Scroll by 1 character cell.
z = 1. Scroll by 1 character cell vertically, 1 byte horizontally.
(i.e. 8 Pixels in 2 colour modes, 4 in 4 colour modes, 2 in 16 colour modes, and 1 character in mode 7). This is the minimum distance that can be scrolled and still be able to do a hardware scroll if the full screen is scrolled.

VDU 23,8,t1,t2,x1,y1,x2,y2,0,0 - Clear block of text window.
This causes a block of the text window to be cleared to the text background colour. The parameters indicate where the two ends of the block (i.e. string start & string finish) are, with t1,x1 and y1 relating to the start of the block and t2,x2 and y2 to the end of the block. In each case, ti indicates a base position, to which (xi,yi) is added to get the true position. The character postion at the start of the block is generally included in the clear, that at the end is not.
ti = 0. Base position is "top left of window"
ti = 1. Base position is "top of cursor column"
ti = 2. Base position is "off top right of window"
ti = 4. Base position is "left end of cursor line"
ti = 5. Base position is cursor position

ti = 10. Base position is "off bottom right of window"

Other values of ti have undefined effects. (The quotes are to indicate that all of these positions are calculated taking the cursor movement controlsset by VDU 23,16 into account - e.g. after VDU 23,16,2,0,0,0,0, 0,0,0, "left" above right etc.)

The results of this function are undefined if the absolute values of the coordinates of the two ends go outside the range -128 to 127. This is best avoided by not using values of xi and yi outside the range -128 to 47. Should the end point of the block lie before the start point, no clearing will be done.

VDU 23,9,n,0,0,0,0,0,0,0 - Set 1st flash time.
Same spec as 1.2 MOS

VDU 23,10,n,0,0,0,0,0,0,0 - Set 2nd flash time.
Same spec as 1.2 MOS

VDU 23,11,0,0,0,0,0,0,0,0 - Set default ecf patterns.

Mode	Pattern	Colour	VDU 23,2-5 type definition
0	1	Dark grey	&CC,&00,&CC,&00,&CC,&00,&CC,&00
	2	Grey	&CC,&33,&CC,&33,&CC,&33,&CC,&33
	3	light grey	&FF,&33,&FF,&33,&FF,&33,&FF,&33
	4	Hatching	&03,&0C,&30,&C0,&03,&0C,&30,&C0
1,5	1	Red-orange	&A5,&0F,&A5,&0F,&A5,&0F,&A5,&0F
	2	Orange	&A5,&5A,&A5,&5A,&A5,&5A,&A5,&5A
	3	Yellow-orange	&F0,&5A,&F0,&5A,&F0,&5A,&F0,&5A
	4	Cream	&F5,&FA,&F5,&FA,&F5,&FA,&F5,&FA
2	1	Orange	&0B,&07,&0B,&07,&0B,&07,&0B,&07
	2	Pink	&23,&13,&23,&13,&23,&13,&23,&13
	3	Yellow-green	&0E,&0D,&0E,&0D,&0E,&0D,&0E,&0D
	4	Cream	&1F,&2F,&1F,&2F,&1F,&2F,&1F,&2F
4	1	Dark grey	&AA,&00,&AA,&00,&AA,&00,&AA,&00
	2	Grey	&AA,&55,&AA,&55,&AA,&55,&AA,&55
	3	Light grey	&FF,&55,&FF,&55,&FF,&55,&FF,&55
	4	Hatching	&11,&22,&44,&88,&11,&22,&44,&88

Mode 0 patterns are different from 4 to avoid TV effects.

VDU 23,12-15,a,b,c,d,e,f,g,h - Set simple ecf pattern.
This sets a simple 2*4 (or double for mode 0) pattern. Patterns 1 thru 4 are set by VDU 23,12 thru VDU 23,15 respectively. The logical colours from left to right are:

Top row - a,b
next row - c,d
next row - e,f
last row - g,h

Mode 0 has double pixels to avoid TV patterning.

VDU 23,16,x,y,0,0,0,0,0,0 - Cursor movement control
Allows control of cursor after a character has been printed. This control sequence replaces the current flag byte as follows:

((Current byte) AND x) EOR y

If the byte of flags is abcdefgh in binary, then:

a = 0. Normal.
a = 1. Undefined.
b = 0. In VDU 5 mode, cursor movement outside of a window cause special actions i.e. Carriage returns generated
b = 1. In VDU 5 mode, cursor movement outside of a window does not cause special actions.
c = 0. Cursor moves in positive direction. d & h define action if cursor moves outside of window.

d = 1. As above but cursor always moves to opposite edge.

efg=000. Text X direction is right, Y direction is down.

efg=001. Text X direction is left. Y direction is down.

efg=010. Text X direction is right, Y direction is right.

efg=011. Text X direction is left, Y direction is up.

efg=100. Text X direction is down, Y direction is right.

efg=101. Text X direction is down, Y direction is left.

efg=110. Text X direction is up, Y direction is right.

efg=111. Text X direction is up, Y direction is left.

h = 0. If movement would go outside of window, cursor moves to negative edge and one step in positive Y direction. If this goes outside of window, d defines behaviour. This is '80' column mode.

h = 1. If movement would go outside of window, a 'pending cursor movement' is generated. It is released before next character is printed (or another control code). This is '81' column mode.

VDU 23,17-27,a,b,c,d,e,f,g,h - Unassigned (but reserved).

VDU 23,27,a,b,c,d,e,f,g,h - Acornsoft sprites.

VDU 23,28-31,a,b,c,d,e,f,g,h - Unassigned (for user application progs).

Reserved for use by application programs. Results in a call to the unknown Plot codes vector &226,&227. Call can be recognised as follows:

* C = 1 on entry to the vector.
* A contains the VDU 23 code (i.e. the first number following 23). All of the sequence except the 23 can be found in ascending order starting at the location: (Start of VDU variables) + &1B, i.e. at &31B in MOS version 1.2.

VDU 23,32-255,a,b,c,d,e,f,g,h - Define character

Spec as 1.2 MOS.

VDU 24,11,1h,bl,bh,rl,rh,tl,th - Set graphics window.

Spec as 1.2 MOS.

VDU 25,p,xl,xh,yl,yh - Plot.

VDU 25,0-63 - Plot line. Spec as 1.2 MOS, but some improvements.

VDU 25,64-71 - Plot point. Spec as 1.2 MOS.

VDU 25,72-79 - Horizontal line fill. Spec as 1.2 MOS.

VDU 25,80-87 - Plot triangle. Spec as 1.2 MOS.

VDU 25,88-95 - Horizontal line fill. Spec as 1.2 MOS.

VDU 25,96-103 - Plot rectangle.

Plots a filled axis aligned rectangle with opposite corners at the current graphics cursorand the new point.

VDU 25,104-111 - Horizontal line fill

Similar to VDU 25,72-79..., with the difference that the word "non-background" should be replaced by "foreground"

VDU 25,112-119 - Plot parallelogram.

Plots a filled parallelogram with vertices at the old graphics cursor, the current graphics cursor, the new point, and at (new point)-(current graphics cursor)+(old graphics cursor) in cyclic order. the 4th point is calculated in terms of internal pixel co-ordinates to ensure that the sides are parallel.

VDU 25,120-127 - Horizontal line fill.

Similar to VDU 25,88-95..., with the difference that the word "background" should be replaced by "non-foreground"

VDU 25,128-143 - Flood fill.

This flood fills the screen starting from the new point and continuing until non-background (plot codes 128-135) or foreground (plot codes 136-143) pixels are found. These sequences

- * The colour being used to fill can itself be filled
- * An escape occurs

VDU 25,144-159 - Plot circle.

Plots a circle outline (plot codes 144-151) or a filled circle (plot codes 152-159) with its centre at the current graphics cursor and the new point on its boundary.

VDU 25,160-183 - Plot circular arc.

Plots a circular arc (plot codes 160-167) the filled chord segment between a circular arc and the chord joining its endpoints (plot codes 168-175) or the filled pie sector between a circular arc and the two radii joining its endpoints to the centre of the circle (plot codes 176-183). In all three cases, the centre of the circle is at the old graphics cursor, the first endpoint of the arc is at the current graphics cursor, the second endpoint of the arc is on the circle and in the same direction from the centre of the circle as the new point is, and the circular arc is taken to be the arc going clockwise from the first endpoint to the second one.

VDU 25,184-191 - Move/copy rectangle.

Causes the axis aligned rectangle with opposite corners at the old and current graphics cursors to be moved (plot codes 185,189) or copied (plot codes 186,187,190,191) so that its new bottom left hand point is at the new point (plot codes 184 and 188 simply move the graphics cursor to the new point, like other plot codes which are 0 mod 4).

Any part of the source rectangle which lies outside the current graphics window is assumed to contain the current graphics background colour for the purposes of the copy or move. The difference between copying and moving is that moving sets any part of the source rectangle which lies outside the destination rectangle to background, whereas copying leaves such parts of the source rectangle unchanged.

VDU 25,192-207 - Plot ellipse

Plots an ellipse outline (plot codes 192-199) or a filled ellipse (plot codes 200-207). The centre of the ellipse is at the old graphics cursor.

VDU 25,208-239 - Unassigned

Not reserved for application programs. Following assigned:
VDU 25,232-239,xl,xh,yl,yh - Acornsoft sprites

VDU 25,240-255 - User program calls

Reserved for application programs. Will result in a call to the unknown plot codes vector (&226,&227). Call recognised by:

- * C=0 on entry
- * Computer is in a graphics mode (can test location (start of VDU variables) + &61 i.e. &361 on 1.2 MOS. This contains (number of pixels/byte)-1 (i.e. 1,3 or 7) in graphics modes, and 0 in non-graphics modes.
- * A contains the VDU 25 code (i.e. the first number following the 25). The coordinates can be found in ascending order starting at the location (start of VDU variables) + &1F i.e. &31F on MOS 1.2

VDU 26 - Restore default windows.

Spec as 1.2 MOS.

VDU 27 - Null.

Spec as 1.2 MOS.

VDU 28,lx,by,rx,ty - Define text window

Spec as 1.2 MOS.

VDU 29,xl,xh,yl,yh - Define graphics origin

Spec as 1.2 MOS.

VDU 32-126 - Print a character
  Spec as 1.2 MOS.
VDU 127 - Backspace and delete.
  Spec as 1.2 MOS.
VDU 128-255 - Print a character
  Prints characters from the extended character set in a similar manner to VDU 32-126

BASIC IV

Incorporates changes from BASIC II to BASIC III as follows:

* Provides some formatting of Assembly listings
* COLOR is accepted as an alternative to COLOUR
* SAVE A$+B$ works correctly
* USE of ! & ? as formal parameters works correctly
* A USA version listing COLOR instead of COLOUR is available

Includes the following additional changes:

* The version number in ROM is 4.
* Incorporates all the 65C02 (65C12) instructions in Assembler:
    * DEC A may be represented as DEA
    * INC A may be represented as INA for compatibility with MASM
    * STZ may be represented as CLR
* ASL ALFRED or similar is accepted i.e. the 'A' indication of accumulator addressing mode for ASL LSR ROL ROR DEC INC no longer affects symbol recognition.
* X,Y or A in the Assembler may be in lower case. EQUB,EQUW & EQUD may also be in lower case.
* Trailing spaces will always be stripped from lines entered into the interpreter.
* Leading spaces will be stripped from lines entered into the interpreter when a non-zero LISTO is set. The assumption is that there will be a formatted listing on screen when cursor editing is used when LISTO is non-zero.
* LISTO indents loops correctly.
* Cross-reference/Search output is available from LIST. Lines will be LISTed IF the specified string is present e.g.:

  ```
  LIST IF DEF
  LIST 10,1000 IF =
  LIST ,2000 IF A%
  ```

  It is not possible to search for TIME=90 , for example, as a statement. It will only be checked for as a boolean expression. PTR£, HIMEM, PAGE, LOMEM are similarly affected.
* RENUMBER or LIST will not be affected by &8D in comments or strings. In addition, LIST will not be confused by coloured comments.
* A statement to update an open file's extent 'EXT£chan=length' has been added. This uses OSARGS and so will not work until there are suitable filing systems.
* A display real-time clock pseudo variable TIME$ has been introduced. It fetches a fixed 24 byte string from the OS in response to PRINT TIME$ (or similar). The string looks like Wed,31 Dec 1900.23:59:59. Assigning to TIME$="fred" merely passes the string directly to the OS with the length in the first byte.
* AUTO no longer outputs a space after the line number.
* General recursion is now allowed in 'FOR' loops e.g.:

  ```
  DEF FNQ FORJ=1TO10:P.J;:N.:=10
  FORI=FNQ-9 TO FNQ STEP FNQ/10
  ```

  now works. In previous versions only the first FNQ or FNQ's without the for loop would work current program (or section of it e.g. EDIT 10,100). It then issues the command "*EDIT hh,hh" where the hex addresses are addresses in zero page of the start of the in-memory text and the address in zero page of the end of the in-memory text plus one LISTO 0 is set before conversion begins. If there is not enough space to convert the entire file, the error message 'No room' will be given together with a line number which shows how far through the program it had got: at this stage either CLEAR or a different EDIT command should be used. ESCAPE will behave similarly, stopping the conversion to text.

* The use of the "|" character at the end of VDU parameters can be used to insert the correct number of remaining zeros.

EDITOR

* Text file editing with ability to go to defined line numbers.
* Includes a formatter to provide reasonable presentation on printed documents.
* Can display on-screen help info. State of this HELP info is kept in CMOS RAM.
* Cursor keys used to move cursor around screen, with screen scrolling up or down as necessary.

*

Shift mode	Action
NONE + up/down arrow	cursor moves one line
NONE + left/right "	" " " character
SHIFT + up/down arrow	" " " screen
SHIFT + left/right "	" " " word
CTRL + up/down arrow	" " top/bottom doc
CTRL + left/right "	" " start/end line

* ESCAPE safely abandons most operations.
* Function key operations:

f0 - Goto specified line number

SHIFT f0 - Toggles between invisible display of carriage returns (default) and small reverse video 'M's. User can easily control trailing spaces.

f1 - Access to OSCLI e.g. <f1>CAT for Disk Catalogue.

SHIFT f1 - Toggles entry mode between insert (default) and overtype. State is displayed at bottom left of screen.

f2 - Load text from a named file.

SHIFT f2 - Insert text from named file at the cursor position.

f3 - Save text to named file.

SHIFT f3 - Remove top and bottom scroll margins.

f4 - Enter interactive find and replace option. A prompt of FIND/REPLACE will appear at bottom of the screen The text until the next '/' or RETURN is the string to be searched for.

Some characters represent powerful operators:

$ - Carriage return.
. - ANY character except carriage return.
- A digit 0 to 9.
@ - An alphabetic (a to z, A to Z) or digit.
~ - 'NOT' the following character e.g. ~£ means not a digit.
* - Any number of the following character.
^ - As many as possible of the following character.

[]- Is a set of options for this character e.g. [0123] would allow 0,1,2, or 3.
| - Control characters as MOS e.g. |J is CTRL + J.

If the string is ended in a RETURN i.e. no /, then the string will be found (if possible) and a prompt of R(eplace),C(ontinue) or ESCAPE will be displayed. C moves on to the next occurence, R replaces and moves on, ESCAPE stops altogether. The text after the second '/' is the string which may replace the found string. If there is no / then you may type in the text and move on when R is used. There are special characters here:

$ - Again represents carriage return.
\ - As find string.
| - As find string.
& - Represents the found string.
%n- (n is 0 to 9) represents the nth wild section of the found string.

SHIFT f4 - Return to language, text being replaced into lang. Clear text buffer.

f5 - Global replace. Syntax exactly as find string. Displays the number of found objects after working. Will operate either over whole file or from mark to cursor.

SHIFT f5 - Set HELP display mode.

f6 - Mark position. When pressed the character that the cursor is on will be replaced by an inverted 1 or 2 and the number of matks displayed at the bottom left of the screen will be increased. Marked positions can be used to delete blocks of text, copy and move them, and perform restricted searches. Up to two marked positions may be set, an attempt to set a third will produce an error message. No editing may be done while marks are set

SHIFT f6 - Remove all marks.

f7 - Copy text delimiting by two marks to where the cursor is now. The marks are not cleared so the operation can be repeated. The cursor must not be in the marked area.

SHIFT f7 - Move text delimited by two marks to where the cursor is now. The marks are cleared. The cursor must not be in the Marked area.

f8 - Print out text

SHIFT f8 - Delete text between single marked position and current cursor position. An error message will occur if there is not exactly one mark present.

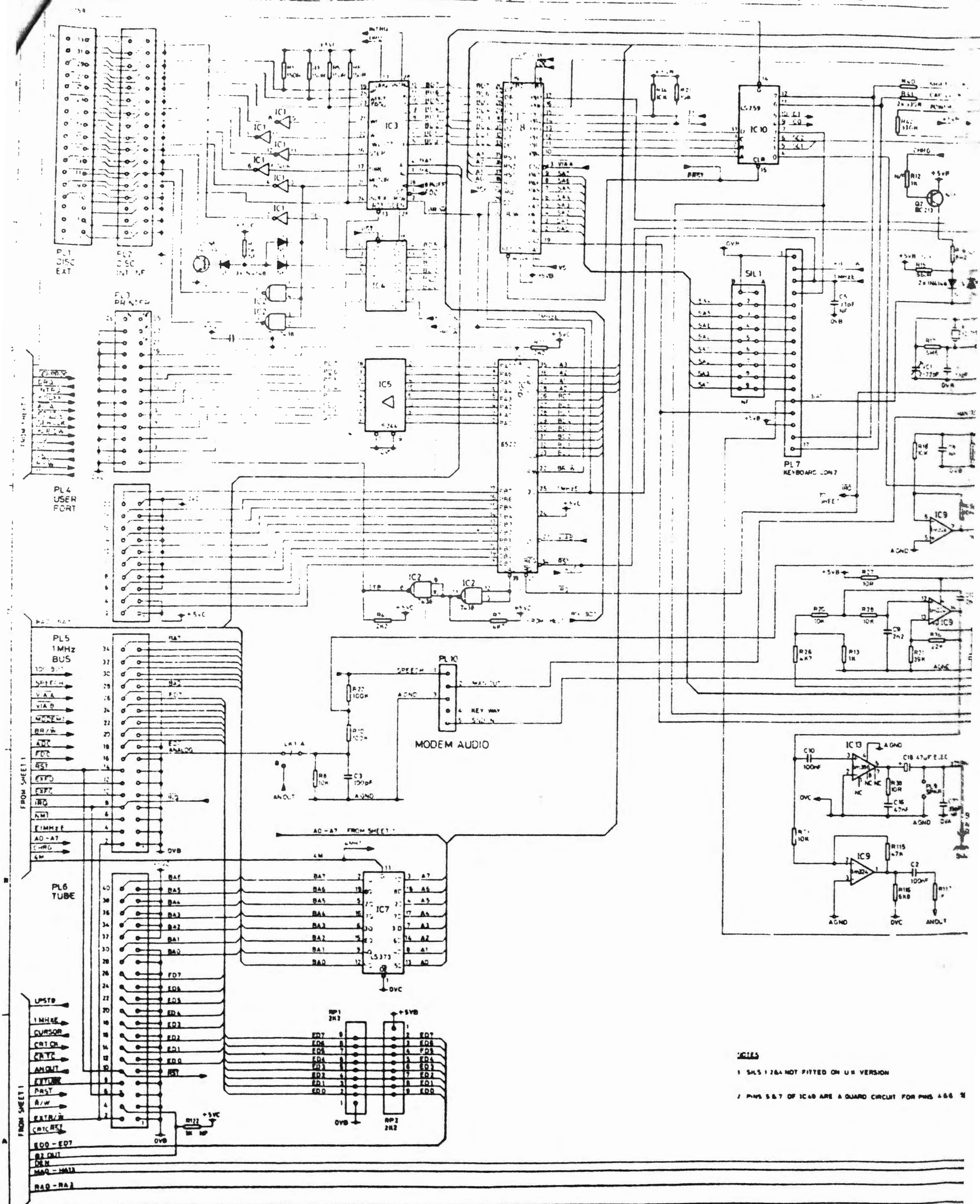
f9 - Get old text back. Works if SHIFT f9 has just been pressed or EDITOR has been broken out of and re-entered.

SHIFT f9 - Delete all text.

This information is subject to change without notice. No responsibility can be taken for any errors or ommisions. The user or program writer should verify that any application program is suitable for the intended environment(s).

DJBELL

DRG No 143,000 C | FIRST USED ON PROJECT B | THIRD ANGLE PROJ



NOTES

1 SILS 1 2&4 NOT FITTED ON U K VERSION

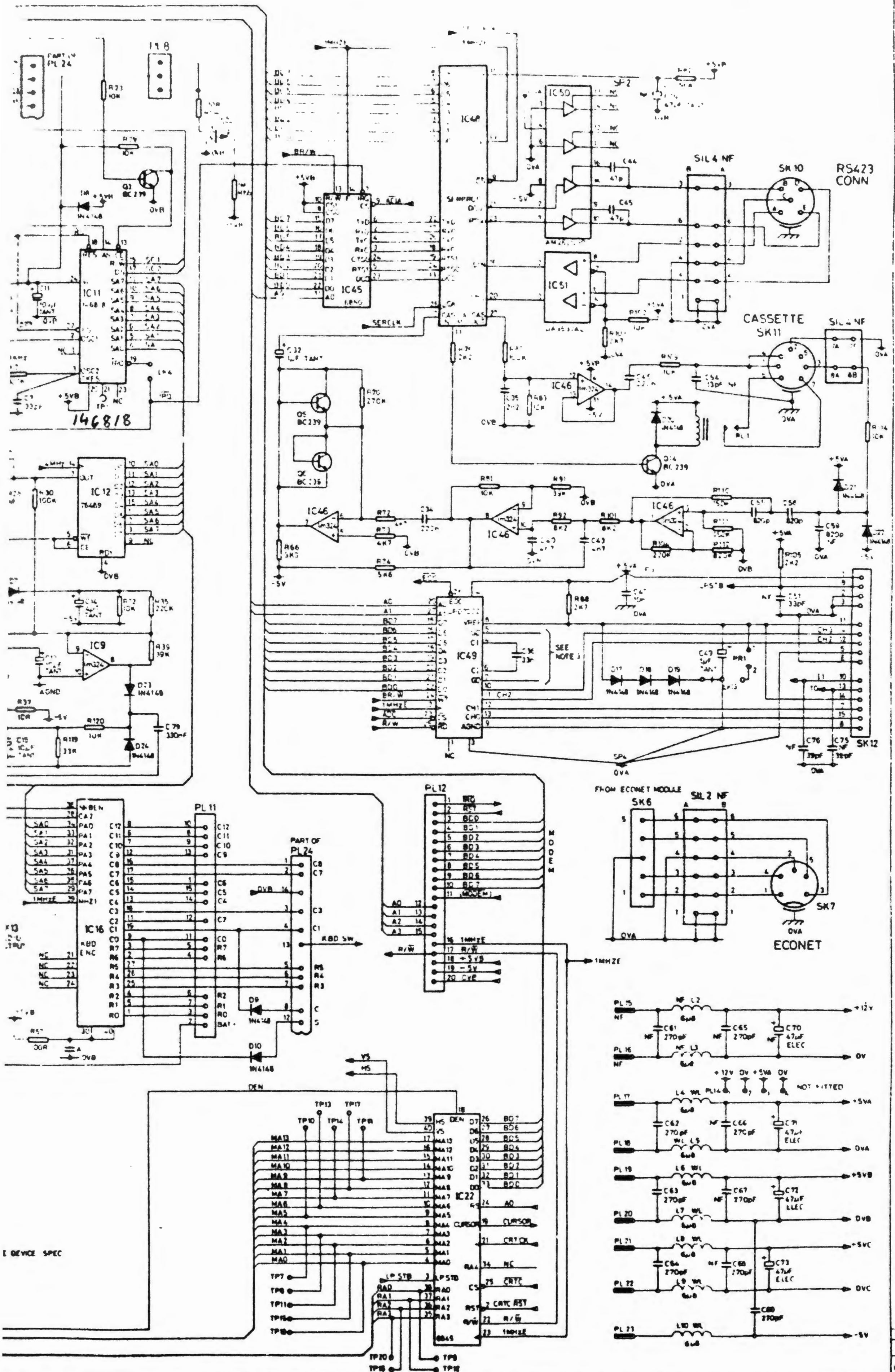
2 PINS 5 & 7 OF IC40 ARE A GUARD CIRCUIT FOR PINS 4 & 6

SCALE		ALL DIMENSIONS IN mm UNLESS OTHERWISE STATED
MATERIAL		
FINISH		
WHOLE No. TOL	DECIMAL No. TOL.	ANGLE TOL.

TRAFALGAR

IF IN DOUBT ASK

DO NOT SCALE



1		AMREO75		
F	1398			
D		REDRAWN ON TWO SHTS	B C	
ISS	DATE	DESC OF CHANGE	DRN	CHKD

ACORN COMPUTERS
not copied, reproduced
wholly or in part,
IN COMPUTERS LTD.

TITLE PROJECT B PCB CIRCUIT DIAGRAM

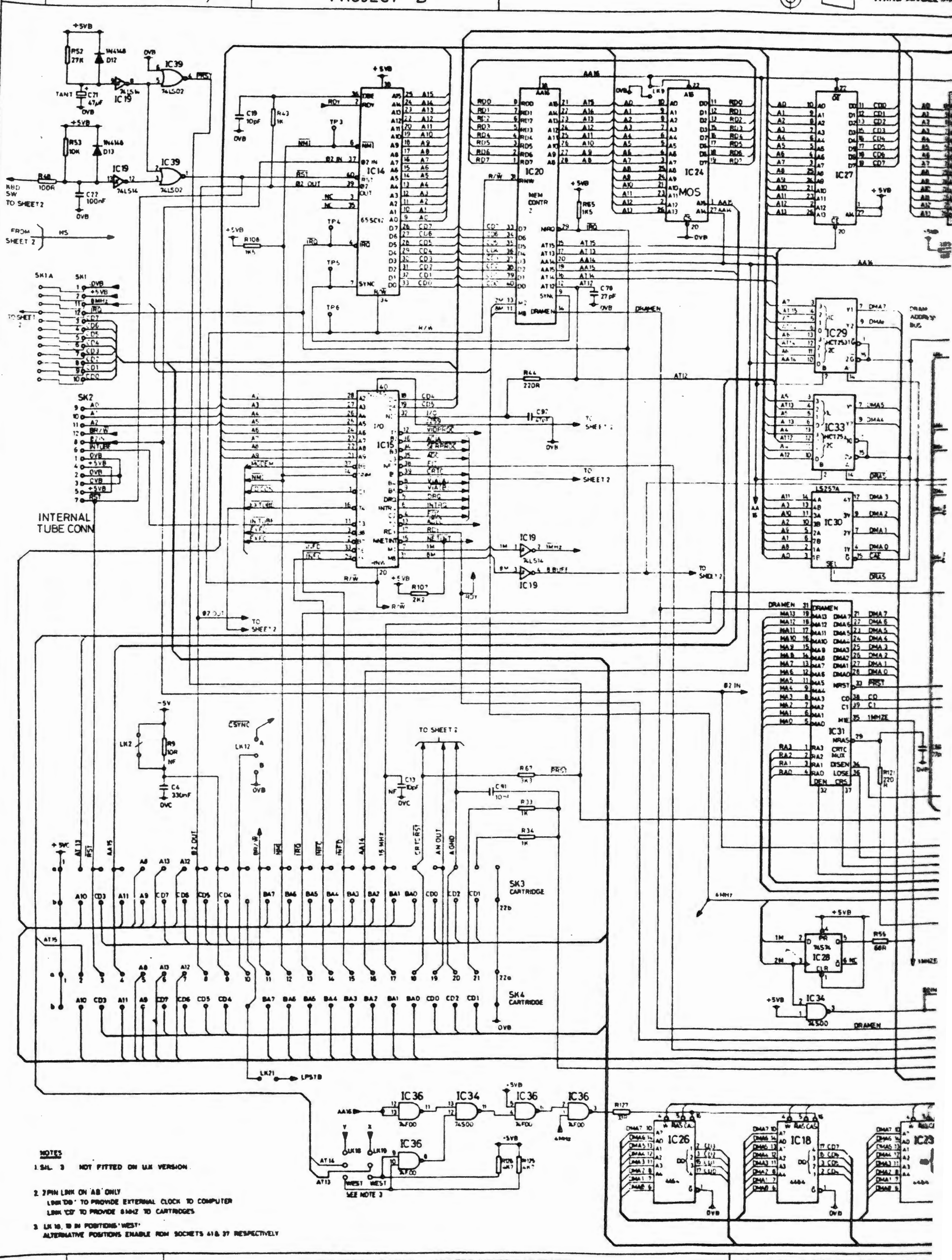
SHEET 2 of 2

ACORN COMPUTERS Ltd

A0 DRG. No. 0143,000/C

MICROFILM LTD

A0 | DRG. No. 0143,000/C | FIRST USED ON PROJECT B | THIRD ANGLE PR



NOTES

1. SK. 3 NOT FITTED ON U.K VERSION.

2. 7PIN LINK ON 'AB' ONLY
LINK 'DB' TO PROVIDE EXTERNAL CLOCK TO COMPUTER
LINK 'CB' TO PROVIDE 8MHZ TO CARTRIDGES

3. LK 18, 19 IN POSITIONS 'WEST'
ALTERNATIVE POSITIONS ENABLE ROM SOCKETS 41& 37 RESPECTIVELY

SCALE

MATERIAL

FINISH

WHOLE No. TOL.

ALL DIMENSIONS IN mm UNLESS OTHERWISE STATED

DECIMAL No. TOL.

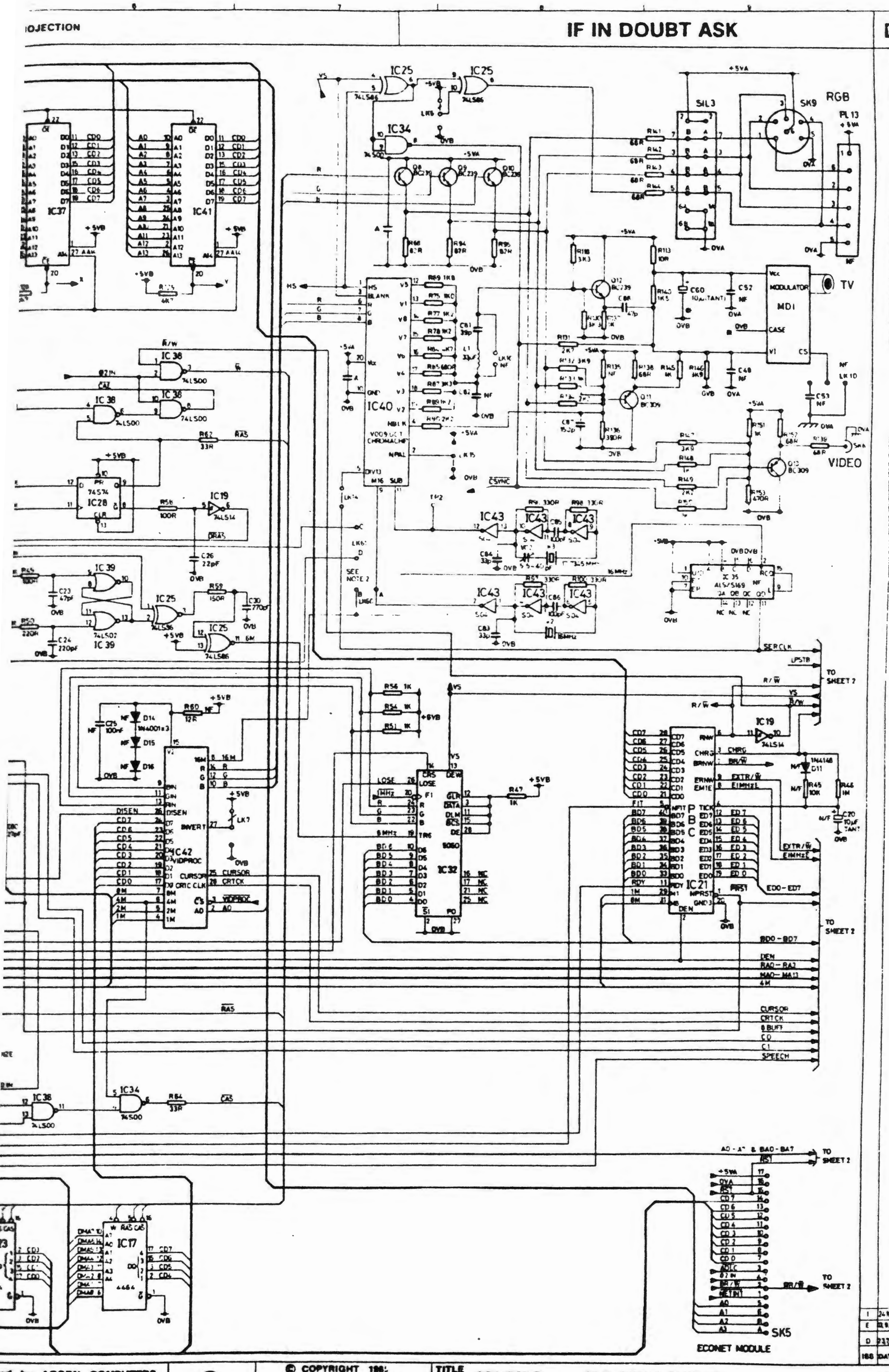
ANGLE TOL.

MASTER

IOJECTION

IF IN DOUBT ASK

DO NOT SCALE



SUB'Y	IC Nº	OV	+5V	TYPE Nº
C	1	7	14	7406
C	2	7	14	7438
C	3	14	15	WD1770
C	4	8	16	74LS174
C	5	10	20	74LS244
C	6	1	20	6522
C	7	10	20	74LS 373
B	8	1	20	6522
B	9	●	●	LM 324
B	10	8	16	74LS 259
B	11	12	●	146818
B	12	8	16	76489
B	13	●	6	LM 386
B	14	1 5,21	8	65SC12
B	15	10	30	I/O CONT CF30050
B	16	10 20	●	KEYBOARD CF30047
B	17	18	9	4464
B	18	18	9	4464
B	19	7	14	74LS14
B	20	10, 32	30	MEM CONT 2 CF3005[illegible]
B	21	10, 32	30	PBC CF30048
B	22	1	20	6845
B	23	18	9	4464
B	24	14	28	EROS
B	25	7	14	74LS86
B	26	18	9	4464
B	27	14	28	ROM/EPROM
B	28	7	14	74S74
B	29	8	16	74HCT253
B	30	8	16	74LS257A
B	31	10, 20	30 40	CRTC MUX CF30040
B	32	1	28	SAA 5050
B	33	8	16	74HCT253
B	34	7	14	74S00
B	35	8	16	74ALS169
B	36	7	14	74F00
B	37	14	28	ROM/EPROM
B	38	7	14	74LS00
B	39	7	14	74LS02
B	40	10	20	CHROMA CF30060
B	41	14	28	ROM/EPROM
B	42	1	16	VID PROC
B	43	7	14	74S04
	44	NOT	USED	
B	45	1	12	6850
B	46	●	●	LM 324
	47	NOT	USED	
B	48	14	28	SERPROC
A	49	●	●	µPD7002
A	50	5	1	26LS30
A	51	4	1	µA9637AC

● SEE CIRCUIT

ISS	DATE	DESC. OF CHANGE	DRN.	CHKD.
I	24.10.85	AMR E075	IRS	
E	13.9.85	64K / CHROMA ETC	IRS	
D	23.7.85	REDRAWN ON TWO SHTS	B C	T Mc

...ed by ACORN COMPUTERS
... it is not copied, reproduced
...ther wholly or in part,
... ACORN COMPUTERS LTD.

© COPYRIGHT 198:
ACORN COMPUTERS Ltd.
FULBOURN ROAD
CHERRY HINTON
CAMBRIDGE CB1 4JN

ACORN COMPUTERS Ltd

TITLE PROJECT B P.C.B. CIRCUIT DIAGRAM

SHEET 1 of 2

A0

DRG. No. 0143,000 /C