ACORN COMPUTER
Acorn Computers Limited, 4a Market Hill, Cambridge CB2 3NJ, England. Telephone 0223 312772

ACORN TECHNICAL MANUAL

Visual Display Unit Interface Board 200.002

Issue 1 October 1979

Introduction

The Acorn Visual Display Unit Controller Board connects to the Standard Acorn Computer Bus and contains a memory mapped character storage R.A.M. which is transparently written to or read from by the C.P.U.

An MC 6845 programmable controller I.C. provides all the synchronisation signals to drive a 625 line 50 fields per second V.D.U. together with read addresses for the character R.A.M. Characters are then fed to an SAA5050 character generator IC which produces the necessary dot patterns to create the characters to refresh the V.D.U.

The SAA5050 produces Teletext standard characters and has Red, Green and Blue drive outputs giving coloured characters or graphics.

The R.G.B. and sync outputs may be used to drive a colour encoder and modulator for a U.H.F. Television; also provided is a 1 volt/75 ohm composite sync and video output which can directly drive a Monochrome Monitor on which the different colours will appear as different scales of grey.

The V.D.U. controller P C B is supplied in kit form with a full set of I.C. sockets. It is easily assembled using a small soldering iron and useful hints on assembly may be found in the Acorn Micro-computer system 1 Technical Manual. The board operates from a single +5v supply from which it draws not more than 500 mA.

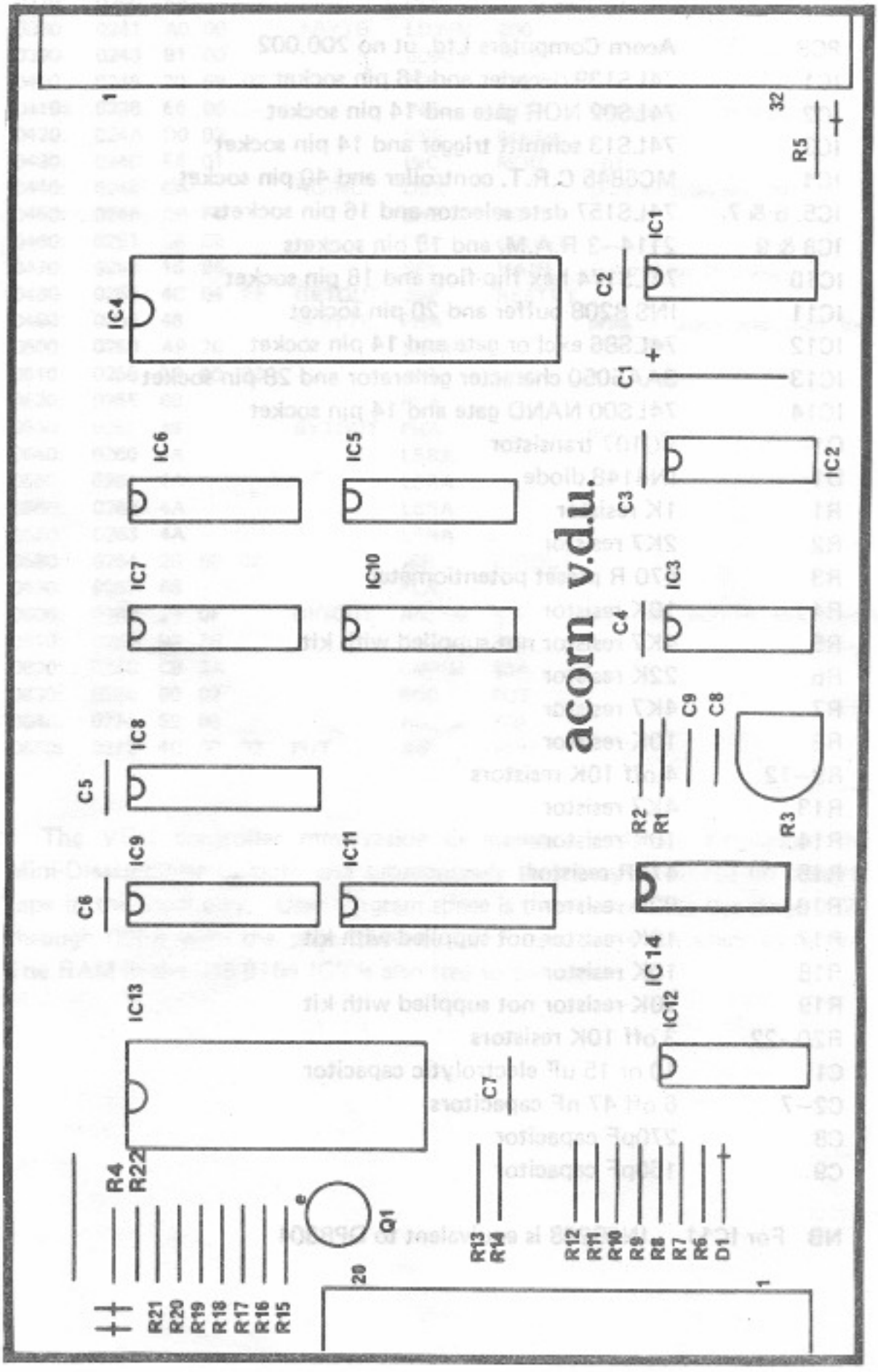
Also provided are listings for programs which set up the MC6845, display 25 instructions in hex on the V.D.U. (with double or treble byte instructions on a single line) and allow the drawing or graphics or characters on the V.D.U. These programs may be loaded and run using the Acorn system 1 Monitor. A new monitor R.O.M. will shortly be available for linking the V.D.U. and an ASC II keyboard to Acorns' 4K Fast BASIC.

PARTS LIST FOR ACORN V.D.U. CONTROLLER

PCB	Acorn Computers Ltd. pt no 200,002
IC1	74LS139 decoder and 16 pin socket
IC2	74LS02 NOR gate and 14 pin socket
IC3	74LS13 schmitt trigger and 14 pin socket
IC4	MC6845 C.R.T. controller and 40 pin socket
IC5, 6 & 7	74LS157 data selector and 16 pin sockets
IC8 & 9	2114–3 R.A.M. and 18 pin sockets
IC10	74LS174 hex flip-flop and 16 pin socket
IC11	INS 8208 buffer and 20 pin socket
IC12	74LS86 excl or gate and 14 pin socket
IC13	SAA5050 character generator and 28 pin socket
IC14	74LS00 NAND gate and 14 pin socket
Q1	BC107 transistor
D1	IN4148 diode
R1	1K resistor
R2	2K7 resistor
R3	470 R preset potentiometer
R4	10K resistor
R5	4K7 resistor not supplied with kit
R6	22K resistor
R7	4K7 resistor
R8	10K resistor
R9–12	4 off 10K resistors
R13	4K7 resistor
R14	10K resistor
R15	470R resistor
R16	82R resistor
R17	18K resistor not supplied with kit
R18	10K resistor
R19	10K resistor not supplied with kit
R20–22	3 off 10K resistors
C1	10 or 15 uF electrolytic capacitor
C2–7	6 off 47 nF capacitors
C8	270pF capacitor
C9	150pF capacitor

NB For IC11 INS8208 is equivalent to DP8304

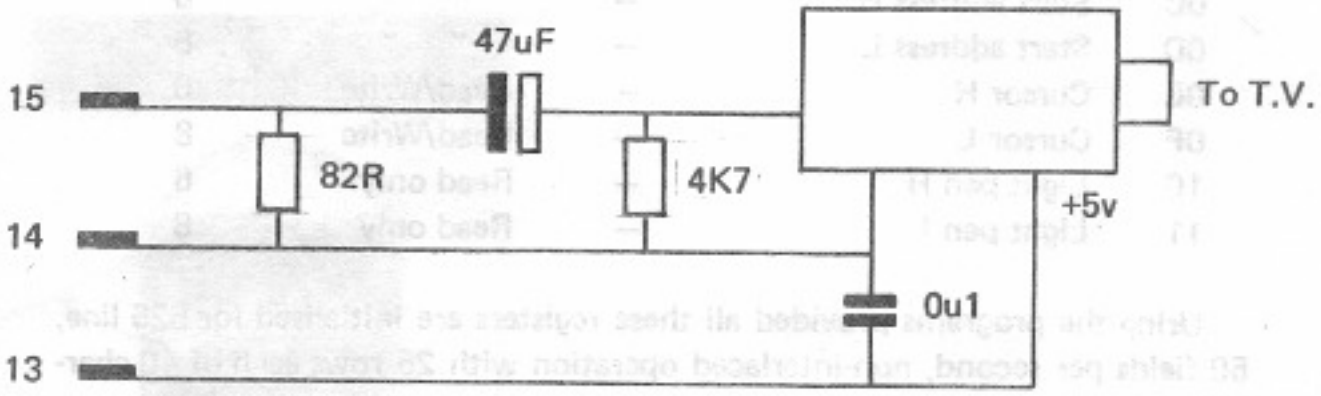
P.C.B.Layout


acorn v.d.u.
200.002 Iss.1
1
32
20
1
IC1
IC2
IC3
IC4
IC5
IC6
IC7
IC8
IC9
IC10
IC11
IC12
IC13
IC14
C1
C2
C3
C4
C5
C6
C7
C8
C9
R1
R2
R3
R4
R5
R6
R7
R8
R9
R10
R11
R12
R13
R14
R15
R16
R17
R18
R19
R20
R21
R22
D1
Q1
e

Applications

To use this board in the Acorn system a low for block zero signal must be present on pin 31A of the Acorn bus. This is provided by connecting pin 8 of IC9 to pin 31A of the connector on the 6502 C.P.U. board. The 5 volt regulator and the electrolytic capacitor must be removed from the C.P.U. board and an external 5 volt supply at 3 amps is required on pin 1A of all boards for a system using the C.P.U., V.D.U. interface, 8K memory board and cassette interface.

A 1 volt at 75 ohm video monitor may be driven directly from connections 14 and 15 on the front of the board. Alternatively a UHF modulator may be driven which connects to the aerial input of a domestic television set using the circuit shown



the component values given suit the Astec UM 1233E36 modulator although other types should work satisfactorily.

IC3 on the VDU board forms a 6MHz dot clock which is adjusted using the preset resistor R3 to give characters of the right width on a steady picture after MC6845 has been initialised using the Applications programmes.

The 1024 byte character memory appears in block zero of the Acorn address map filling pages 04, 05 06 and 07. Two registers, which set up the 6845, are at addresses 0800 and 0801 and these repeat throughout page 08. The register at 0800 is a 5 bit write only register used to determine which of the data registers is accessed at 0801. The data registers, their addresses and their functions are tabulated here.

Register Address	Function	Program Unit	Type	No of Bits Used
00	Horizontal total	char	Write only	8
01	Horizontal displayed	"	" "	8
02	H Sync Position	"	" "	8
03	H Sync Width	"	" "	4
04	Vertical total	char row	" "	7
05	V total adjust	scan line	" "	5
06	Vertical displayed	char row	" "	7
07	V sync position	char row	" "	7
08	Interlace mode	—	" "	2
09	Max scan line address	scan line	" "	5
0A	Cursor start	scan line	" "	5 + 2
0B	Cursor end	scan line	" "	5
0C	Start address H	—	" "	6
0D	Start address L	—	" "	8
0E	Cursor H	—	Read/Write	6
0F	Cursor L	—	Read/Write	8
10	Light pen H	—	Read only	6
11	Light pen L	—	Read only	8

Using the programs provided all these registers are initialised for 625 line, 50 fields per second, non-interlaced operation with 25 rows each of 40 characters.

Hardware scrolling is used in the system which, with transparent access, gives a clean display and uses only a very small proportion of C.P.U's time. The character memory may be read from or written to at any time like any other piece of R.A.M.

The light pen input to the board may be used after breaking the link to 0v and fitting a 4K7 resistor for R5. When the light pen input goes high the contents of the character refresh address counter are strobed into registers 10 and 11 i.e. high byte and low byte. Thus the screen co-ordinates of the pen position may be determined.

The SAA 5050 character generator produces the following characters represented by the hexbytes shown.

20	space	30	0	40	@	50	P	60	–	70	p
21	!	31	1	41	A	51	Q	61	a	71	q
22	"	32	2	42	B	52	R	62	b	72	r
23	£	33	3	43	C	53	S	63	c	73	s
24	$	34	4	44	D	54	T	64	d	74	t
25	%	35	5	45	E	55	U	65	e	75	u
26	&	36	6	46	F	56	V	66	f	76	v
27	'	37	7	47	G	57	W	67	g	77	w
28	(	38	8	48	H	58	X	68	h	78	x
29	)	39	9	49	I	59	Y	69	i	79	y
2A	*	3A	:	4A	J	5A	Z	6A	j	7A	z
2B	+	3B	;	4B	K	5B	←	6B	k	7B	¼
2C	,	3C	<	4C	L	5C	½	6C	l	7C	‖
2D	–	3D	=	4D	M	5D	→	6D	m	7D	¾
2E	.	3E	>	4E	N	5E	↑	6E	n	7E	÷
2F	/	3F	?	4F	O	5F	≠	6F	o	7F	delete

Note that there are a few exceptions to the usual ASCII, the most noticeable of which is at 23 where £ replaces ≠which is moved from 23 to 5F.

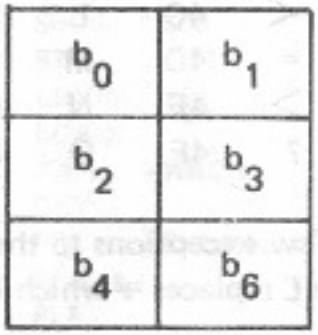
Also available are the following teletext control characters with their hexadecimal codes shown:—

01	Alpha	Red	11	Graphics	Red
02	"	Green	12	"	Green
03	"	Yellow	13	"	Yellow
04	"	Blue	14	"	Blue
05	"	Magenta	15	"	Magenta
06	"	Cyan	16	"	Cyan
07	"	White	17	"	White
08	Flash		18	Conceal Display	
09	Steady		19	Contiguous Graphics	
OA	End Box		1A	Separated Graphics	
OB	Start Box				
OC	Normal Height		1C	Black Background	
OD	Double Height		1D	New Background	
			1E	Hold Graphics	
			1F	Release Graphics	

These characters affect the characters displayed to the right of them on a line, and provide the following options:–

ALPHA (COLOUR) causes following characters on the line to be in the colour specified.

GRAPHICS (COLOUR) In Graphics mode each character space displayed is divided into 6 cells. Each cell is illuminated, in the specified colour, if a corres-ponding bit in the ASCII code stored at that location is set. The bit assignments are:–



Bit 5 in the byte is always set for a graphics character, if bit 5 is clear then the upper case, alphabet and characters ←, ½, →, ↑, #, @ are available in the same colour.

FLASH causes the following characters on the line to flash.

STEADY terminates the flash option on a line.

END BOX & START BOX Are options for using the circuit to superimpose text onto a normal TV picture.

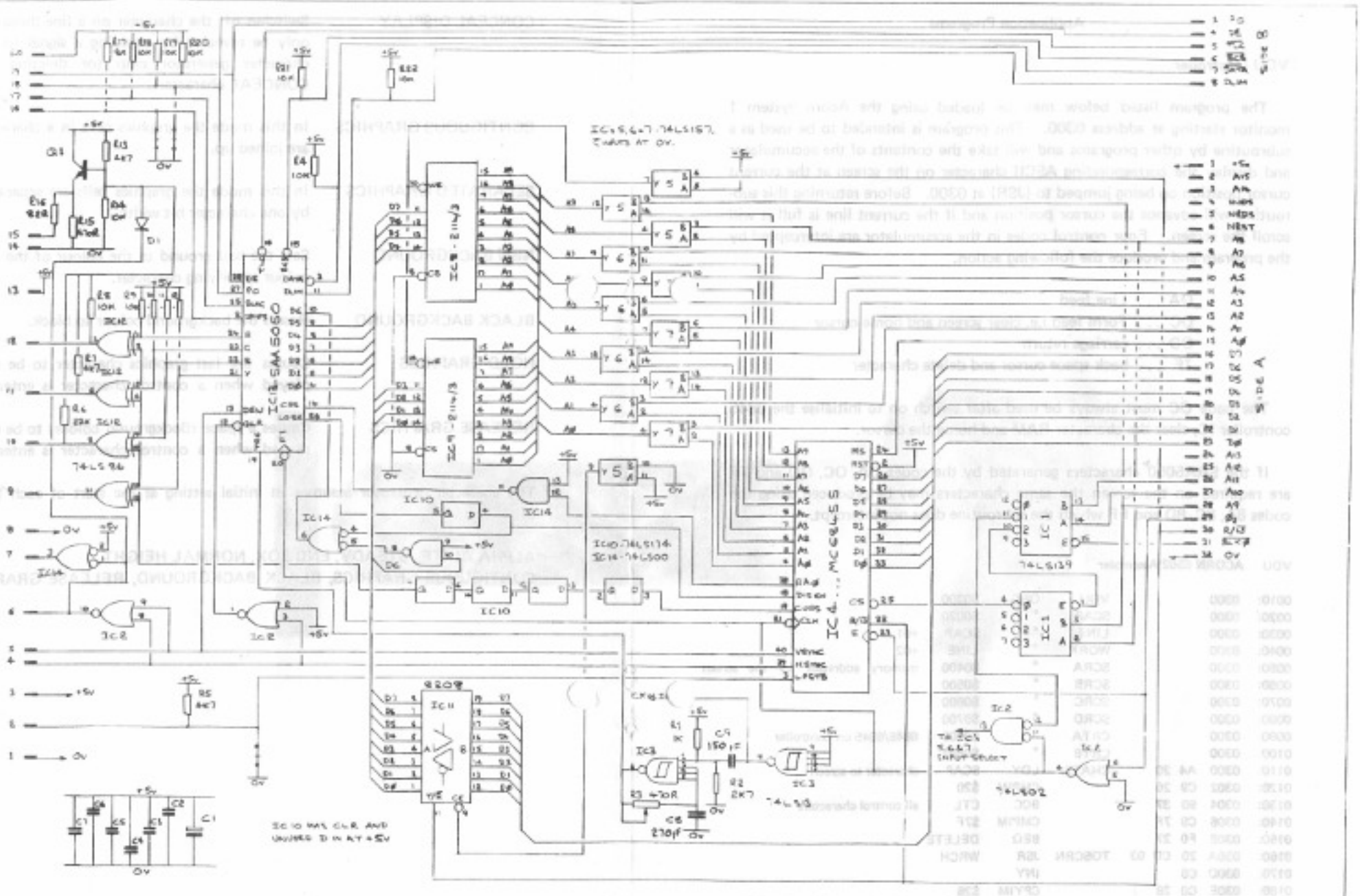
DOUBLE HEIGHT Must appear on two consecutive lines followed by the same characters in which case the characters fill the two lines specified.

NORMAL HEIGHT Clears the double height option on a line.

CONCEAL DISPLAY	Switches off the character on a line these can only be revealed by applying a signal to the character generator chip (or deleting the CONCEAL character).
CONTIGUOUS GRAPHICS	In this mode the graphics cells in a character are joined up.
SEPARATED GRAPHICS	In this mode the graphics cells are separated by one character bit width.
NEW BACKGROUND	Sets the background to the colour of the last colour specifying character.
BLACK BACKGROUND	Resets the background colour to black.
HOLD GRAPHICS	Causes the last graphics character to be displayed when a control character is entered.
RELEASE GRAPHICS	Causes a space (Background colour) to be displayed when a control character is entered.

The character generator assumes an initial setting at the start of each line of:—

ALPHA WHITE, STEADY, END BOX, NORMAL HEIGHT.
CONTIGUOUS GRAPHICS, BLACK BACKGROUND, RELEASE GRAPHICS.



ISSUE	C	D	1
DATE	12/6/79	9/08/79	29/08/79
	CBT	CBT	CBT

VDU Controller

The program listed below may be loaded using the Acorn system 1 monitor starting at address 0300. This program is intended to be used as a subroutine by other programs and will take the contents of the accumulator and display the corresponding ASCII character on the screen at the current cursor position on being jumped to (JSR) at 0300. Before returning this subroutine will advance the cursor position and if the current line is full it will scroll the screen. Four control codes in the accumulator are intercepted by the program and produce the following action.

OA ... Line feed
OC ... Form feed i.e. clear screen and home cursor
OD ... carriage return
7F ... back space cursor and delete character

The code OC must always be used after switch on to initialise the 6845 controller IC, clear the character RAM and home the cursor.

If the SAA5050 characters generated by the codes OA, OC, OD and 7F are required on the screen the same characters may be produced using the codes 8A, 8C, 8D and FF which the subroutine does not intercept.

VDU ACORN 6502 Assembler

```
0010:  0300                    VDU      ORG     $0300
0020:  0300                    SCAP     *       $0020
0030:  0300                    LINE     *       SCAP    +01
0040:  0300                    WORK     *       LINE    +02
0050:  0300                    SCRA     *       $0400   memory addresses for the screen
0060:  0300                    SCRB     *       $0500
0070:  0300                    SCRC     *       $0600
0080:  0300                    SCRD     *       $0700
0090:  0300                    CRTA     *       $0800   6845/6545 crt controller
0100:  0300                    CRTB     *       $0801
0110:  0300  A4 20             CHATS    LDY     SCAP    character to screen
0120:  0302  C9 20                      CMPIM   $20
0130:  0304  90 37                      BCC     CTL     all control characters
0140:  0306  C9 7F                      CMPIM   $7F
0150:  0308  F0 27                      BEQ     DELETE
0160:  030A  20 CD 03          TOSCRN   JSR     WRCH
0170:  030D  C8                         INY
0180:  030E  C0 28                      CPYIM   $28
0190:  0310  90 05                      BCC     VDUB    automatic scroll when line filled
0200:  0312  20 80 03          FILLED   JSR     SCROLL
```

```
0210:  0315  A0 00      VDUA    LDYIM   $00
0220:  0317  20 B5 03   VDUB    JSR     CALCN
0230:  031A  84 20              STY     SCAP
0240:  031C  A0 0F              LDYIM   $0F     rewrite cursor position
0250:  031E  8C 00 08           STY     CRTA
0260:  0321  A4 23              LDY     WORK
0270:  0323  8C 01 08           STY     CRTB
0280:  0326  A0 0E              LDYIM   $0E
0290:  0328  8C 00 08           STY     CRTA
0300:  032B  A4 25              LDY     WORK    +02
0310:  032D  8C 01 08           STY     CRTB
0320:  0330  60         VDUC    RTS
0330:  0331  88         DELETE  DEY
0340:  0332  30 FC              BMI     VDUC    refuse to delete before line start
0350:  0334  A9 20              LDAIM           write in a blank
0360:  0336  20 CD 03           JSR     WRCH
0370:  0339  A9 7F              LDAIM   $7F
0380:  033B  D0 DA              BNE     VDUB
0390:  033D  C9 0D      CTL     CMPIM   $0D     carriage return?
0400:  033F  F0 D4              BEQ     VDUA
0410:  0341  C9 0A              CMPIM   $0A     line feed?
0420:  0343  F0 06              BEQ     SCROL
0430:  0345  C9 0C              CMPIM   $0C     form feed?
0440:  0347  F0 09              BEQ     CLEARS
0450:  0349  D0 BF              BNE     TOSCRN
0460:  034B  20 80 03   SCROL   JSR     SCROLL  scroll screen and rewrite cursor
0470:  034E  A4 20              LDY     SCAP
0480:  0350  B0 C5              BCS     VDUB
0490:  0352  48         CLEARS  PHA             clear entire buffer
0500:  0353  A0 00              LDYIM   $00
0510:  0355  A9 20              LDAIM
0520:  0357  99 00 04   CLEAR   STAAY   SCRA
0530:  035A  99 00 05           STAAY   SCRB
0540:  035D  99 00 06           STAAY   SCRC
0550:  0360  99 00 07           STAAY   SCRD
0560:  0363  C8                 INY
0570:  0364  D0 F1              BNE     CLEAR
0580:  0366  84 20              STY     SCAP
0590:  0368  A0 0F              LDYIM   $0F
0600:  036A  8C 00 08   SETCRT  STY     CRTA    set up all the crt parameters
0610:  036D  B9 D9 03           LDAAY   CRTTAB
0620:  0370  8D 01 08           STA     CRTB
0630:  0373  88                 DEY
0640:  0374  10 F4              BPL     SETCRT
0650:  0376  A9 C0              LDAIM   $C0
0660:  0378  85 21              STA     LINE
0670:  037A  A9 07              LDAIM   $07
0680:  037C  85 22              STA     LINE    +01
0690:  037E  68                 PLA
0700:  037F  60                 RTS
0710:  0380  08         SCROLL  PHP             scroll subroutine
0720:  0381  48                 PHA
0730:  0382  D8                 CLD
```

```
0740:  0383  A0 28          LDYIM  $28
0750:  0385  20 B5 03       JSR    CALCN
0760:  0388  A5 23          LDA    WORK
0770:  038A  85 21          STA    LINE
0780:  038C  A5 25          LDA    WORK   +02
0790:  038E  85 22          STA    LINE   +01
0800:  0390  A0 0D          LDYIM  $0D
0810:  0392  8C 00 08       STY    CRTA
0820:  0395  A5 21          LDA    LINE
0830:  0397  38             SEC
0840:  0398  E9 C0          SBCIM  $C0
0850:  039A  8D 01 08       STA    CRTB
0860:  039D  88             DEY
0870:  039E  8C 00 08       STY    CRTA
0880:  03A1  A5 25          LDA    WORK   +02
0890:  03A3  E9 03          SBCIM  $03
0900:  03A5  8D 01 08       STA    CRTB
0910:  03A8  A0 27          LDYIM  $27
0920:  03AA  A9 20          LDAIM
0930:  03AC  20 CD      CLEARL JSR    WRCH
0940:  03AF  88             DEY
0950:  03B0  10 FA          BPL    CLEARL
0960:  03B2  68             PLA
0970:  03B3  28             PLP
0980:  03B4  60             RTS
0990:  03B5  08         CALCN  PHP          do calculation to make sure that the
1000:  03B6  48             PHA            processor and crt controller agree on
1010:  03B7  D8             CLD            position of screen
1020   0388  18             CLC
1030:  03B9  98             TYA
1040:  03BA  65 21          ADC    LINE
1050:  03BC  85 23          STA    WORK
1060:  03BE  A5 22          LDA    LINE   +01
1070:  03C0  69 00          ADCIM  $00
1080:  03C2  85 25          STA    WORK   +02
1090:  03C4  29 07          ANDIM  $07
1100:  03C6  09 04          ORAIM  $04
1110:  03C8  85 24          STA    WORK   +01
1120:  03CA  68             PLA
1130:  03CB  28             PLP
1140:  03CC  60             RTS
1150:  03CD  20 B5 03   WRCH   JSR    CALCN
1160:  03D0  84 25          STY    WORK   +02
1170:  03D2  A0 00          LDYIM  $00
1180:  03D4  91 23          STAIY  WORK
1190:  03D6  A4 25          LDY    WORK   +02
1200:  03D8  60             RTS
1210:  03D9  3F         CRTTAB =      $3F   total number of characters per line
1220:  03DA  28                =      $28   40 characters displayed
1230:  03DB  33                =      $33   position of horizontal sync
1240:  03DC  05                =      $05   width in uS of horizontal sync pulse
1250:  03DD  1E                =      $1E   total number of character rows
1260:  03DE  02                =      $02   additional no. of lines for 312 total
```

```
1270:  03DF  19     =   $19    25 displayed character rows
1280:  03E0  1B     =   $1B    position of vertical sync pulse
1290:  03E1  00     =   $00    set non-interlace mode
1300:  03E2  09     =   $09    set 10 lines per character row
1310:  03E3  68     =   $68    slow blink cursor from line 9
1320:  03E4  09     =   $09    to line 10
1330:  03E5  04     =   $04    high address of VDU ram
1340:  03E6  00     =   $00    low address of VDU ram
1350:  03E7  07     =   $07    high address of initial cursor position
1360:  03E8  C0     =   $C0    low address of intial cursor position
```

Two further programs may be loaded which emply the V.D.U. control subroutine.

The keywrite program is entered at MAIN (0280). When this program is first used after switching on a form feed can be sent to the screen to program the 6845 by pressing the 'r' key.

The program then accepts pairs of Hex characters before sending them to the screen, by-passing the control interpreter to allow any code to be sent. The control keys provide the following functions

1	Line feed	g	space
Λ	delete	p	space
V	carriage return	s	space
r	form feed	m	return to monitor

KEYWRT ACORN 6502 Assembler

```
0010:  0280                    KEYWRT  ORG     $0280
0020:  0280                    TEMP    *       $0026
0030:  0280                    RESTRT  *       $FF04
0040:  0280                    DISPLA  *       $FE0C
0050:  0280                    VDU     *       $0300
0060:  0280  20 0C FE  MAIN    JSR     DISPLA
0070:  0283  90 0E             BCC     SEND    hex key ?
0080:  0285  29 07     CONTRL  ANDIM   $07
0090:  0287  F0 1B             BEQ     RETURN
0100:  0289  A8                TAY             look up control keys in table
0110:  028A  B9 A6 02          LDAAY   TABLE   -01
0120:  028D  20 00 03  SENDER  JSR     VDU
0130:  0290  4C 80 02          JMP     MAIN
0140:  0293  0A        SEND    ASLA
0150:  0294  0A                ASLA
0160:  0295  0A                ASLA
0170:  0296  0A                ASLA
0180:  0297  85 26             STA     TEMP
0190:  0299  20 0C FE          JSR     DISPLA
0200:  029C  B0 E7             BCS     CONTRL
0210:  029E  05 26             ORA     TEMP    mix in low digit
0220:  02A0  09 80             ORAIM   $80     fool control character check
0230:  02A2  30 E9             BMI     SENDER  forced branch to sender
0240:  02A4  4C 0A FF  RETURN  JMP     RESTRT  M key returns to monitor
0250:  02A7  20        TABLE   =       $20     G key gives space bar
0260:  02A8  20                =       $20     P key gives space bar
0270:  02A9  20                =       $20     S key gives space bar
0280:  02AA  0A                =       $0A     L key gives line-feed
0290:  02AB  0C                =       $0C     R key gives form-feed
0300:  02AC  7F                =       $7F     /\ key gives delete
0310:  02AD  0D                =       $0D     \/ key gives carriage return
```

Acorn Minidisassembler

The minidisassembler provides a formated listing of 25 lines of 6502 program instructions in machine code presented in Hexadecimal.

The start address of the program to be disassembled is entered into the m address of the Acorn Monitor using the m key and the dis-assembler program is then executed from 0200 using the g key in the normal way. A formated 25 lines of instructions will be displayed on the screen and the program ends in a jump back to the monitor. The m address is stepped forwards so re-running the program will display the next 25 program lines.

```
MINDIS          ACORN 6502 Assembler

0010:  0200                    MINDIS   ORG     $0200
0020:  0200                    MOD      *       $0000
0030:  0200                    COUNT    *       $000E
0040:  0200                    VDU      *       $0300
0050:  0200                    RESTRT   *       $FF04
0060:  0200  A9 18             START    LDAIM   $18     disassemble 25 lines
0070:  0202  85 0E                      STA     COUNT
0080:  0204  D8                         CLD
0090:  0205  A9 0C                      LDAIM   $0C     start with a form-feed
0100:  0207  20 00 03                   JSR     VDU
0110:  020A  A9 0D             MAIN     LDAIM   $0D     carriage return/line feed for each line
0120:  020C  20 00 03                   JSR     VDU
0130:  020F  A9 0A                      LDAIM   $0A
0140:  0211  20 00 03                   JSR     VDU
0150:  0214  A5 01                      LDA     MOD     +01 display current address
0160:  0216  20 58 02                   JSR     $PBYTE
0170:  0219  A5 00                      LDA     MOD
0180:  021B  20 5F 02                   JSR     BYTOUT
0190:  021E  A0 00                      LDYIM   $00
0200:  0220  A2 01                      LDXIM   $01     X will be the byte count of the opcode
0210:  0222  B1 00                      LDAIY   MOD     fetch opcode, find it's no. of bytes
0220:  0224  C9 20                      CMPIM   $20     'jsr' is an anomaly and is done first
0230:  0226  F0 17                      BEQ     CBYTE
0240:  0228  29 9F                      ANDIM   $9F
0250:  022A  F0 15                      BEQ     ABYTE   binary 0XX00000 is 1 byte
0260:  022C  29 1D                      ANDIM   $1D
0270:  022E  C9 19                      CMPIM   $19
0280:  0230  F0 0D                      BEQ     CBYTE   binary XXX110X1 is 3 bytes
0290:  0232  29 0D                      ANDIM   $0D
0300:  0234  C9 08                      CMPIM   $08
0310:  0236  F0 09                      BEQ     ABYTE   binary XXXXX0X0 (now) is 1 byte
0320:  0238  29 0C                      ANDIM   $0C
0330:  023A  C9 0C                      CMPIM   $0C
0340:  023C  F0 01                      BEQ     CBYTE   binary XXXX11XX is 3 bytes
0350:  023E  CA                         DEX             all others are 2 bytes
```

```
0360:  023F  E8          CBYTE   INX
0370:  0240  E8                  INX
0380:  0241  A0 00       ABYTE   LDYIM  $00
0390:  0243  B1 00               LDAIY  MOD
0400:  0245  20 58 02            JSR    SPBYTE
0410:  0238  E6 00               INC    MOD     increment the byte pointer
0420:  024A  D0 02               BNE    NOINC
0430:  024C  E6 01               INC    MOD     +01
0440:  024E  CA          NOINC   DEX             print all bytes required
0450:  024F  D0 F0               BNE    ABYTE
0460:  0251  C6 0E               DEC    COUNT
0470:  0253  10 B5               BPL    MAIN    finished the 25 lines?
0480:  0255  4C 04 FF    GETOUT  JMP    RESTRT
0490:  0258  48          SPBYTE  PHA             print a space and then the byte
0500:  0259  A9 20               LDAIM
0510:  025B  20 00 03            JSR    VDU
0520:  025E  68                  PLA
0530:  025F  48          BYTOUT  PHA             print a byte
0540:  0260  4A                  LSRA
0550:  0261  4A                  LSRA
0560:  0262  4A                  LSRA
0570:  0263  4A                  LSRA
0580:  0264  20 68 02            JSR    DIGOUT
0590:  0267  68                  PLA
0600:  0268  29 0F       DIGOUT  ANDIM  $0F      print the bottom Hex digit in A
0610:  026A  09 30               ORAIM  $30
0620:  026C  C9 3A               CMPIM  $3A
0630:  026E  90 02               BCC    PUT
0640:  0270  69 06               ADCIM  $06
0650:  0272  4C 00 03    PUT     JMP    VDU
```

The VDU controller may reside in memory with the Keywriter, the Mini-Disassembler or both and subsequently they may be stored on cassette tape in the usual way. User program space is then reduced to the range 0020 through 00FF with the page 0100 to 01FF used for stack space as before. The RAM in the INS 8154 IC's is also free to the user.