8 kph computaware

CORTEX USER GROUP NEWSLETTER (August, September 1986)

Issue Number 8.

63 Highlands Road,
Andover, Hampshire,
England, SP10 2PZ

CONTENTS

Proprietor: K.P.Holloway BSc.

EDITORIAL

Welcome to the eighth issue of the Cortex Users Group Newsletter. Once again we would like to say thank you to everyone who has written and hope that you find this issue interesting.

Unfortunately, we regret that KPH Computaware will not be able to continue with the running of the User Group, or the newsletter publications. We will be producing one more newsletter this year, in which we will include all of the items which we receive before going to press. This decision is due to the fact that KPH Computaware is a relatively small business concern, and has recently been run mainly by myself (Kevin Holloway) in my spare time. Recent employment changes do not allow me sufficient time to keep the User Group going.

However, we are reluctant to let the group fold completely, and are of course willing to pass on all of the records and other documentation necessary to run the User group. Should anyone be interested then please send your phone number or address to our usual address, and we will try and detail what would be entailed. We would also be willing to pass on the remaining stock, and program master tapes subject to approval by their authors.

Yours faithfully,

K.P.Holloway

K.P.Holloway
Cortex Newsletter Editor

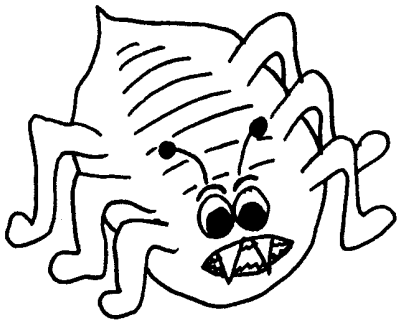
Now available: 64K Eprom Cards for the E.Bus.

This card has been designed and tested by Mr.E.Serwa, and is available from the address below for £27.

The card consists of a PCB fitted with turn pin EPROM sockets, and tested for use on a standard E.Bus. The capacity of the card is 64K which can be made up from 2764 EPROMS. NB.The EPROMS are not supplied with this card.

Please send orders/enquiries to;

Mr.E.Serwa
93 Long Knowle Lane
Wednesfield
Wolverhampton
West Midlands
WV11 1JG



BUG BYTES

Once again we present a collection of problems from Cortex Users. We are always willing to print questions about the Cortex and appreciate any answers or suggestions which are supplied. Please specifiy when writing whether or not you wish your address to be printed so that individual correspondence can be entered into.

Paul Sheridan wrote to us in reply to Mr.Holdaways query in newsletter 7 page 3. He had also noticed and corrected the same errors, and solved the problem in Tim Gray's 3D Graph by inserting RESTOR 2090 at line 115.

W.D.Eaves from Caithness wrote in with the following problem;

"I have a problem with the video section of my Cortex 2 computer. The fault occurs only when in graph mode and started off as an intermittent fault which has now become permanent. The fault affects the foreground colours and the shape tables as described below.

The colour fault only affects foreground colours in graph mode. Background colours are unaffected. Colours 4 to 7 and 12 to 15 are OK, however colours 0 to 3 and 8 to 13 are transformed to the colour 4 above. If the colour numbers are expressed in binary then it can be seen that the bit corresponding to 4 is permanently set to 1.

The shape fault is of a similar nature in the fact that the pixels are set permanently whether they are set by the software or not. Again the figure 4 appears to be the common factor. A shape which is set to 0,0,0,0 will appear as the shape 4040H,4040H,4040H,4040H which is a vertical line in the 2nd LHS column. The cortex manual glosses over the workings of the TMS 9929 and it's associated memory so I am at a loss as to what is happening.

I suspect a hardware fault in the section of circuitry involving IC's 48 to 56 and have checked for continuity and shorts in the data lines. Suspecting a possible failure in one of the memory chips (8818) I swapped them about to check if the fault would change but it remained exactly the same. I also checked the continuity of the pin connections on the TMS 9929 which are also OK.

I did suspect the data line corresponding to 4 being held high by a short but this did not check out. If so one would also expect the fault to have appeared in text mode and the graph background colours and also the shape parameter to be 4444H not 4040H.

Could the fault be the TMS 9929 itself. Being rather expensive I would prefer not to risk buying one without being sure whether or not it is responsible. Any ideas or comments on this subject would be very welcome."

FEATURE ARTICLE.

SWITCHING CONTROLLER/MONITOR FOR E-BUS

Paul Moyers

7 Philip Grove
Sutton
St Helens
Merseyside
WA9 3TD

Two useful interface circuits are featured here which allow the Cortex to be connected to the outside world. The Switching Controller can be used to drive lamps, motors, relays etc. Whilst the Switching Monitor can be used to feed positional infomation or external circuit status back to the Cortex. The circuits may be used in such applications as Model Railway, Robot, Burglar Alarm, Lighting control etc.

Both circuits will operate via the E-Bus in its simplest form without the 74LS2001 (as detailed in "Cortex Centronics Interface" E.T.I. June/August 1984 and The Cortex II manual Page 17 to 22). Their CRU base address is 0200H (512).

8-WAY SWITCHING CONTROLLER (Fig:1)

IC1/IC2 Form an address decoder energising IC3 from base address 0200H on output cycle (CRUCLK Low). The data bit on A15 CRUOUT is then transferred to the output Q0 to Q7 selected by A12, A13, A14. When an output n (where n=0 to 7) is ON. Dn2 will light and Tn2 will be turned ON switching up to 500mA each. The supply for the devices to be controlled must not exceed 50V DC and must be isolated otherwise heavy currents would flow back through the computer via the 0V rail.

8-WAY SWITCHING CONTROLLER (Fig:2)

Address decoding is performed as above but IC3 is energised on the input cycle (CRUCLK High). The data bit on D0 to D7 is again selected by A12, A13, A14; inverted and transferred to CRUIN. When input switch between In and ICOMM is closed LED Dn1 will light and Dn goes low. Rn2 and Cn1 form low pass filters to prevent noise triggering IC3.

BASIC CONTROL COMMANDS

```
BASE 0200H or BASE 512----------Select interface base address
CRB(n)=1------------------------Turns selected output "n" on
CRB(n)=0------------------------Turns selected output "n" off
IN=CRB(n)----------Stores status (1 or 0) of input "n" in variable IN
```

I am prepared to produce PCB's or Kits for these modules if there is enough demand. If you are interested then let me know.

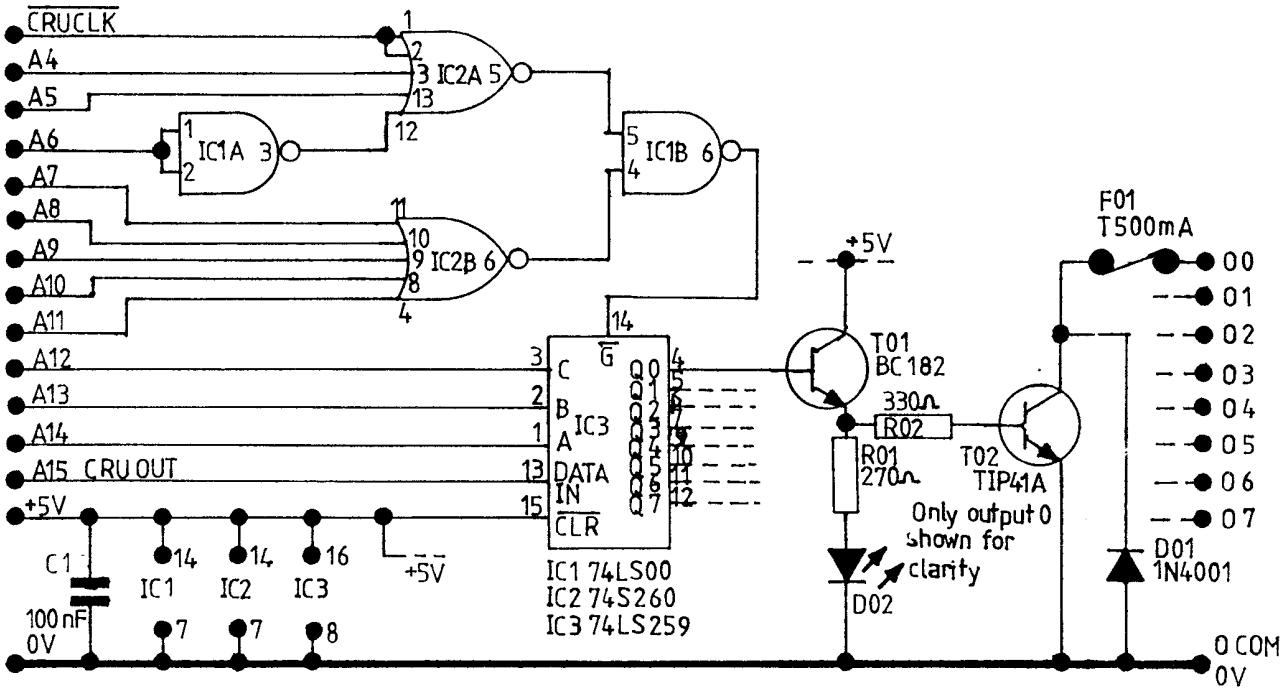


FIG1 CORTEX 8 WAY SWICHING CONTROLLER BASE 512 CRB[0] TO CRB[7]

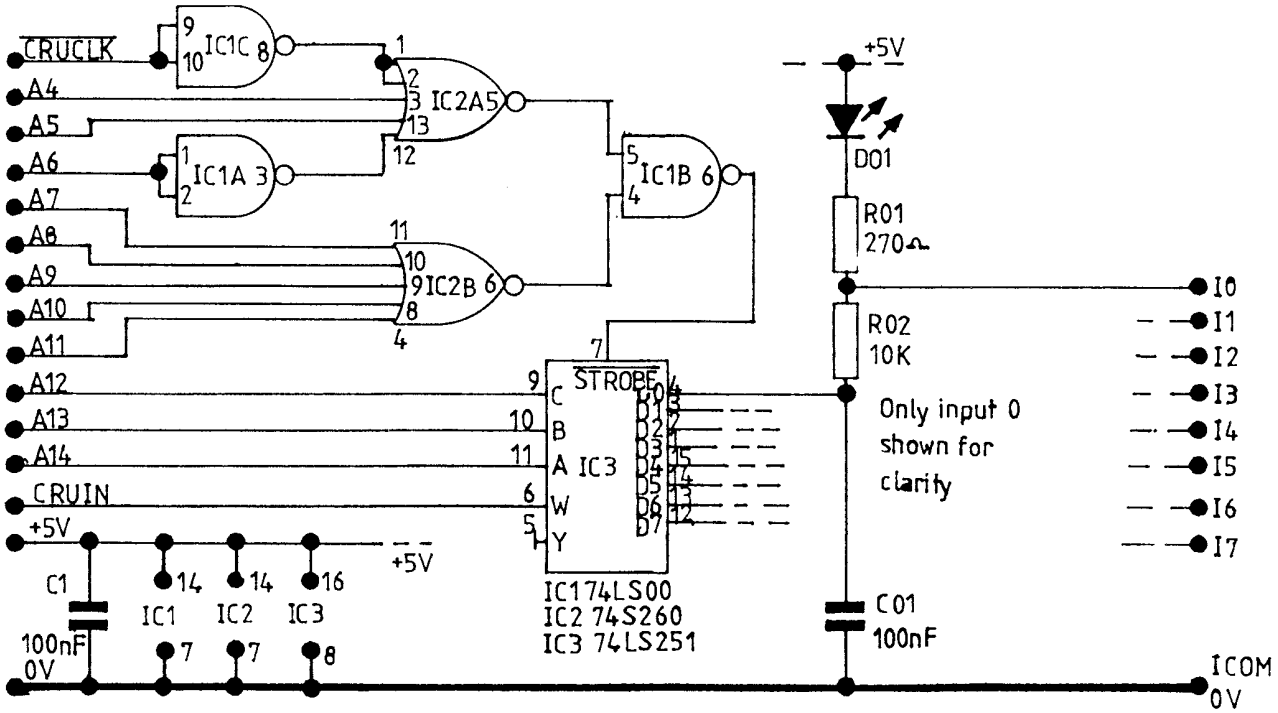


FIG2 CORTEX 8 WAY SWITCHING MONITOR BASE 512 CRB[0] TO CRB[7]

PROGRAMS.

The following programs and routines have been sent in by Cortex Users. Please keep on sending items to us, as we do feature them all eventually.

The first program is by Robert Lee, and provides a CDOS Track Verify Utility. The program prompts for a drive number, and then steps from track zero to the innermost track, reading each track and displaying the track number if correctly read. If a track cannot be read, the error is noted and the program steps to the next track. The DI utility can be used to patch a bad sector.

```
100 TEXT: PRINT"CDOS Track Verify Utility program"
110 ERROR 330
120 INPUT £1"Drive ",D
130 DIM X(4)
140 AX=ADR(X(0))
150 MWD(AX)=0420H: MWD(AX+2)=06180H: MWD(AX+4)=0D000H
160 MWD(AX+6)=01601H: MWD(AX+8)=0380H: MWD(AX+10)=0460H
170 MWD(AX+12)=06550H
180 DIM C(500)
190 A=0
200 T=0
210 NT=MWD(MWD(06382H+D*2)+2)
220 NT=NT/MWD(0636AH+D*2)-1
230 BPS=MWD(06362H+D*2)
240 SPT=MWD(0636AH+D*2)
250 BPT=SPT*BPS
260 W1=ADR(A)+2: W2=ADR(A)+4
270 FOR T=0 TO NT
280  A=(16*65536+D*256)*65536+T*BPT
290  CALL AX,0,MWD(W1),MWD(W2),ADR(C(0)),BPT
300  ? £;T,
310 NEXT T
320 END
330 X1=MEM(0EE36H): Y1=MEM(0EE37H)
340 ? @(0,20)"Disc read error no.";SYS(1);" at track ";£;T
350 ERROR 330
360 T=T+1
370 PRINT @(X1,Y1);"xx",
380 POP
390 GOTO 280
```

Our next program is also from Robert, and is a numerical simulation of living cells. The program is called LIFE and generates patterns based on three simple rules, starting with an initial pattern in a grid (the screen).

i) A new cell is generated if it has exactly 3 neighbours.
ii) A cell dies if it has less than 2 neighbours, or more than 3.
iii) A cell lives if it has 2 or 3 neighbours.

The program first invites you to draw an initial pattern using the cursor keys, space bar and delete key. Robert recommends fairly simple patterns of say 4 or 5 cells to start with. When you are satisfied with the pattern press CTRL G and the program jumps into a loop displaying a continuous representation of succesive generations of cells. The only way to exit from this loop is to press ESCAPE.

```
 10  REM   ***              ***
 20  REM   ***    LIFE      ***
 30  REM   ***              ***
 40  REM   *** BY R.M.LEE ***
 50  REM   ***              ***
 60  DIM A[45]
 70  B=ADR[A[0]]
 80  PRINT @"C"
 90  GOSUB 240
100  COLOUR 10,1
110  X=20: Y=12
120  PRINT @(X,Y);"*<08>";
130  A=KEY[0]
140  IF A=047H THEN GOTO 220
150  IF A=08H THEN X=X-1: ;"<08>";: IF X<0 THEN X=0
160  IF A=09H THEN X=X+1: ;"<09>";: IF X>39 THEN X=39
170  IF A=0AH THEN Y=Y+1: ;"<0A>";: IF Y>21 THEN Y=21
180  IF A=0BH THEN Y=Y-1: ;"<0B>";: IF Y<0 THEN Y=0
190  IF A=32 THEN GOTO 120
200  IF A=017H THEN PRINT @(X,Y);" <08>";
210  GOTO 130
220  CALL B
230  GOTO 220
240  FOR N=B TO B+266 STEP 2: READ Q: MWD[N]=Q: NEXT N: RETURN
250  DATA 736,-4080,520,41,521,2,522,3,523,8192,524,0,514,-3808,515
260  DATA -3807,516,960,517,68,518,32767,-11067,1733,-11067,-8814,
          1540,5885,513,960
270  DATA 514,08000H,515,-31488,516,0,517,02A00H,-15998,550,1,-28330
          5633,1412,550
280  DATA 38,-28330,5633,1412,550,1,-28330,5633,1412,550,1,-28330,
          5633,1412,-15998
290  DATA 24968,-28330,5633,1412,550,1,-28330,5633,1412,550,1,-28330
          5633,1412,550
300  DATA 38,-28330,5633,1412,-28334,5642,-32188,5634,-11067,4107,
          -32124,5634,-11067,4103,-11061
310  DATA 4101,-32124,5634,-11067,4097,-11061,1220,1410,1411,1537,
          5825,514,-31449,515,25
320  DATA 516,8192,517,40,-11132,-24443,1539,5884,514,-3808,515,
          -3807,516,960,517
330  DATA 68,518,-31488,519,08000H,-11067,1733,-11067,-8746,-11082,
          1540,5884,1420,4246
```

The following program is by Syd Champkin, and is another variation of the autorunning of the CDOS directory at the time of `BOOT`. Disk files are loaded into a string and then displayed in blocks of eight filenames. The blocks can then be incremented or decremented until the required filename is found. The file can then be loaded by entering a single digit from 1 to 8.

Saving this program with a filename "DIR" and then adding an extra line to the `AUTOEXEC` utility (LOAD 0,"DIR") will allow the user to see the contents of the disk and load the required program at `BOOT` time.

NB In order to use this program it is necessary to make the `NEW` command modification described in the short tips section of this issue.

AUTO LOAD PROGRAM LISTING. (SYD CHAMPKIN)

```
10 TEXT: COLOUR 1,15
20 PRINT @(1,1); "CDOS Program load utility"
30 DIM B(100), #N(2), X(20), #PGM(40,2), #DOS(14,2), #S(2)
40 AX=ADR(X(0)): AB=ADR(B(0))
50 DATA 0420H, 06180H, 0D000H, 01601H
60 DATA 0380H, 0460H, 06550H, 04F2H
70 DATA 04D2H, 0C0F1H, 0704H, 0A13H
80 DATA 01701H, 0592H, 0600H, 01601H
90 DATA 0380H, 0A14H, 016F8H, 010F5H
100 FOR I=AX TO AX+38 STEP 2
110 READ IAQ: MWD(I)=IAQ
120 NEXT I
130 D=0
140 DC=MWD(06382H+D*2)
150 BS=MWD(DC): NB=MWD(DC+2)
160 DS=MWD(DC+4): ND=MWD(DC+6)
170 BPS=MWD(06362H+D*2)
180 CO=1
190 FOR E=1 TO ND-1
200 DA=DS*BPS+E*64
210 CALL AX,0,D*256,DA,AB,64
220 IF MWD(AB)=0 THEN GOTO 280
230 FOR II=1 TO 8
240 #N(0;II)=%MEM(AB+II+1)%0
250 NEXT II
260 #PGM(CO,0)=#N(0)
270 CO=CO+1
280 NEXT E
290 B=0: C=1
300 IF B<=0 THEN B=0
310 IF B>=32 THEN B=32
320 IF C<=1 THEN C=1
330 IF C>=5 THEN C=5
340 PRINT"<C>": PRINT @(14,1);"DISK INDEX ";C: PRINT: PRINT: PRINT
350 FOR A=1 TO 8
360 PRINT TAB(5);A; TAB(15);#PGM(A+B,0)
370 NEXT A
380 PRINT: PRINT
390 PRINT;" Enter 'I' to increment list": PRINT
400 PRINT;"       'D' to decrement list": PRINT: PRINT
410 INPUT %1" Select a program 1-8 : ";#S(0)
420 PRINT
430 A=ASC(#S(0))
440 IF A=73 THEN B=B+8: C=C+1: GOTO 300
450 IF A=68 THEN B=B-8: C=C-1: GOTO 300
460 IF A<49 OR A>56 THEN GOTO 560
470 IF C=1 THEN A=A-48
480 IF C=2 THEN A=A-40
490 IF C=3 THEN A=A-32
500 IF C=4 THEN A=A-24
510 IF C=5 THEN A=A-16
520 IF #PGM(A,0)="" THEN GOTO 550
530 PRINT"<C>": PRINT" Loading ";#PGM(A,0)
540 LOAD 0, #PGM(A,0)
550 PRINT" File not on disk": WAIT 300: GOTO 300
560 PRINT" Valid charactors only": WAIT 300: GOTO 300
```

545 CALL 01F8H

The next program is based on a technique for solving simultaneous equations, using matrices. It is called the Gauss-Jordan matrix reduction method for solving simultaneous equations. The actual method will not be described here, partly because it is not necessary to understand it in order to use this program, and the description would have to be in a mathematical form in order to fit it on the page. Anyone wishing to look up this topic should be able to find it in most degree level mathematics texts, or books on numerical computation.

Simultaneous equations of the form shown below can be represented by a matrix (a 2D array). The example shown below illustrates how this program is set up from the equations you wish to solve.

equation 1) $x_1 + 2x_3 = 4$
" 2) $x_1 + x_2 + x_4 = 4$
3) $x_2 + 3x_3 = 7$
4) $x_1 + x_2 + x_3 + x_4 = 6$

$$\Rightarrow \begin{pmatrix} 1 & 0 & 2 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 3 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 4 \\ 4 \\ 7 \\ 6 \end{pmatrix}$$

A B

To set up the arrays A and B, the values are stored as data in lines 620 to 650 of the program. The answers (values of x_1, x_2, x_3, x_4) are to be found in array B after the program has been run. Both arrays are printed on the screen before and after processing.

```
10  REM ******************************
20  REM G-J SIMULTANEOUS EQN SOLVING
30  REM WITH FULL PIVOT SEARCH
40  REM K.P.HOLLOWAY (11-6-86)
50  REM ******************************
60  GOSUB 500: REM SET UP A,B
70  DIM D[N]
80  FOR I=1 TO N: D[I]=0: NEXT I
90  REM ****************
100  REM START MAIN LOOP
110  REM ***************
120  GOSUB 690: REM PRINT INITIAL ARRAYS
130  ? : ? : ?
140  FOR C=1 TO N
150   REM ****************************
160   REM FIND LARGEST ELEMENT IN A
170   REM ****************************
180   PV=0: PI=0: PJ=0
190   FOR I=1 TO N
200    IF D[I]=1 THEN GOTO 250
210    FOR J=1 TO N
220     IF D[J]=1 THEN GOTO 240
230     IF A[I,J]*A[I,J]>PV*PV THEN PV=A[I,J]: PI=I: PJ=J
240    NEXT J
250   NEXT I
260   REM ********************************************************
270   REM MOVE PIVOT ONTO DIAGONAL AND DIVIDE PIVOT ROW BY PIVOT
280   REM ********************************************************
290   FOR J=1 TO N
300    T=A[PJ,J]: A[PJ,J]=A[PI,J]: A[PI,J]=T: A[PJ,J]=A[PJ,J]/PV
310   NEXT J
320   T=B[PJ]: B[PJ]=B[PI]: B[PI]=T: B[PJ]=B[PJ]/PV
```

```
330  REM ***********************************
340  REM REDUCE PIVOT COLUMN TO 0's, EXCEPT PIVOT (=1)
350  REM ***********************************************
360  FOR I=1 TO N
370   IF I=PJ THEN GOTO 430
380   M=A[I,PJ]
390   FOR J=1 TO N
400    A[I,J]=A[I,J]-M*A[PJ,J]
410   NEXT J
420   B[I]=B[I]-M*B[PJ]
430  NEXT I
440  D[PJ]=1: REM MARK PIVOT ROW/COL
450 NEXT C
460 GOSUB 690: REM PRINT ANSWERS
470 STOP
480 REM ****************************
490 REM ****************************
500 REM SET UP ARRAYS
510 RESTOR 610
520 READ N
530 DIM A[N,N],B[N]
540 FOR I=1 TO N
550  FOR J=1 TO N
560   READ A[I,J]
570  NEXT J
580  READ B[I]
590 NEXT I
600 RETURN
610 DATA 4
620 DATA 1,0,2,0,4
630 DATA 1,1,0,1,4
640 DATA 0,1,3,0,7
650 DATA 1,1,1,1,6
660 REM ********************
670 REM PRINT ARRAYS A AND B
680 REM ********************
690 FOR I=1 TO N
700  ? " (";
710  FOR J=1 TO N
720   ?  TAB (J*4);A[I,J];
730  NEXT J
740  ?  TAB (J*4);") (X";I;") = (";(INT[B[I]*100])/100; TAB ((J+4)*4);")"
750 NEXT I
760 RETURN
```

NB. The program as listed deals with sets of 4 equations; to deal with a different number is necessary to adjust the value at line 610.

USER INFO

Mr.P.R.Kenny from Australia wrote to us asking the following questions. He cannot locate sufficient details about the 74LS2001 IC for the E.Bus, and is looking for a full functional specification so that he can design his own replacement.

Mr.Kenny would also be interested to hear from anyone who knows of additional documentation on the Cortex, apart from the compilation of ETI extracts and the programming manual.

Mr D.Raison from Waterlooville wrote in response to our request for information on printers. He has recently purchased a citizen LSP-10 and makes the following comments.

The LSP-10 is a very versatile printer that can emulate either the IBM or Epsom standards. It can be obtained with either a centronics or RS232 port cartridge, and further cartridges can also be bought.

A complete set of page formatting commands are available including four margins, full text justification, proportional print and 8 widths of character. Print modes supported are emphasised, doublestrike, italics, underlining, overscoring, reverse print, subscripts, superscripts and double height- most of which are also available in correspondence quality(CQ).

Full bit graphics, variable line feeds, user defined characters and foreign charcter sets are also available. The price of this printer is about £179 which includes a tractor feeder and 2 year guarantee.

SHORT TIPS

R.M.Lee reponds to Julian Terry's tip concerning the cursor in the last issue, by saying that printing character 1DH turns the cursor OFF, and character 1CH turns it on again.

Rick Packer sent in the following tip and also mentions that interference in video circuitry can arise from local airport radar. In response to video problems described in recent issues, Rick says that some of these may be because there are 2 types of polycarbonate capacitors fitted in the Cortex, and shorting of tracks can occur if they are too close to the board. The affected components are C4, 5, 6, 9, 10, 13, 17, 20, 21, 22 and 23.

ERRATA

page 6.5 The three lines near label NEXTFILE should read;

```
7168            AI R3,>0000
716C NEXTFILE:  MOV *R3,*R3
716E            JEQ NEX
```

page 7.9 The following line should be ammended;

```
14 6022 C1A0 5FC0  MOV @>5FC0,R6
```

page 7.5 The following line should be ammended;

```
540 IF N[I,1]*WX+N[I,2]*WY+N[I,3]*WZ<0 : GOTO 650
```

page 7.11 On IC 70 the ceramic capacitor across pins 1 and 2 should be 1nF.

FEATURE Into, onto and out of the E.Bus.

This article was kindly written by Tim Gray in response to requests for more information about the Cortex E.Bus. This is the first of a series of articles which Tim is currently writing. We are sure that this will be of great interest to all cortex users who wish to use the E.Bus expansion system. In connection with this, please see the advert in the editorial (page 2) about an E.Bus PROM card which is being sold by Mr.E.Serwa.

This is a series of articles to describe the E.Bus system with special reference to its use use with the CORTEX computer . In the first article I hope to describe the the E.Bus system and how to implement it in the Cortex with follow up articles on how to add extra RAM ROM and CRU expansion cards including tested circuit diagrams and use of interupts .

E.Bus capabilities

The T.I. E.Bus system is a multiplexed address data and interupt code bus . This means that at any one time address , data or an interupt code can appear on the same bus lines with A.D.I. control signals controling which is active . The bus supports 8 or 16 bit wide data , although on the cortex only 8 bit is used , and allows use of more than one main proccessor on the same bus on a priority access system .

The Cortex E.Bus interface supports only 8 bit wide data and does not use the interupt code system so if the cortex is the only or main microproccessor accessing the bus a simplified E.Bus system can be used .

The E.Bus specification allows for a total of 1M byte of memory on the bus , the Cortex cannot access the lower 64K externally , and a total of 4096 CRU I/O bits . Table 1 shows the pin connections for the bus with a note of which lines are open collector and which are tristate . Fig 1 shows a replacement circuit for the unobtainable LS2001 and will be refered to in the following decriptions of bus use . The E.Bus manual specifies a maximum of 500mm line length be used on the bus with a termination network of a 560R resistor to ground and a 330R resistor to +5 volts on each of the tristate lines,however in order to drive such line terminations high power drivers would be necessary in the form of 74LS645-1 IC's which are expensive . If care is taken in the construction of the bus and the length kept to a minimum the termination network can be omitted and 74LS244/245 IC's used . The open collector lines however still need a 1KR pullup resistor to +5V for the bus to function correctly .

A ribbon cable can be used to connect a small bus system to the Cortex if every other cable is connected as an earth to prevent crosstalk between signal lines . A suitable P.C.B to construct an 8 slot E.Bus is the Elector number 83102 available for about £12 . A seperate E.Bus power supply may be required depending on the number of extra cards added,in this case the +5 and +/-12 volt lines should not be connected to the Cortex .

Table 1

PIN	ROW A	TYPE	ROW C	TYPE
1	GND	--	GND	--
2	-PRES	O/C	-BUSCLK	T/S
3	+12V	--	-12V	--
4	-IORST	O/C	-NMI	O/C
5	+5V	--	+5V	--
6	+5V BATT	--		
7				
8				
9				
10	-INTEN	O/C	ALATCH	T/S
11	XA0	T/S	XA1	T/S
12	XA2	T/S	XA3	T/S
13	A0/D0/INT0	T/S	A1/D1/INT1	T/S
14	A2/D2/INT2	T/S	A3/D3/INT3	T/S
15	A4/D4/INT4	T/S	A5/D5/INT5	T/S
16	A6/D6/INT6	T/S	A7/D7	T/S
17	A8/D8	T/S	A9/D9	T/S
18	A10/D10	T/S	A11/D11	T/S
19	A12/D12	T/S	A13/D13	T/S
20	A14/D14	T/S	A15/DA5/CRUOUT	T/S
21	-AREADY	O/C	-MEMEN	T/S
22	-DEN	T/S	-READY	O/C
23	GRANTIN	D	GRANTOUT	D
24	-PWRFAIL	--	-BUSY	O/C
25	GND	--	GND	--
26	+15V	--	ANALOGUE BUS	--
27	ANALOGUE COM	--	ANALOGUE BUS	--
28	-15V	--	CRUIN	O/C
29	-WE	T/S	+5V STANDBY	--
30	+5V	--	+5V	--
31	MEMWIDTH	T/S	-CRUCLK	T/S
32	GND	--	GND	--

Lines shown -- are connected direct
Row A 7 to 9 and row C 6 to 9 are unused
GRANTIN and GRANTOUT should be linked daisy chain fashion if used

In order to describe the function of the variouse control lines I will describe the use of the bus for memory and CRU transfers explaining the function of IC's as required .

Memory transfers

If the mapper has been programmed with an external address and the proccessor tries to access that address a signal -EXTMEM is generated by IC21a this arrives at IC13d and generates a -BRQ at IC102 pin 2 and hence a BEN at IC102 pin 3 . IC92 a & b have been previously set so BEN via IC93a becomes -ABE and enables

the address bus buffers IC94a , IC95 & IC96 . The address is now on the bus . The address is recived on the bus memory card with an address latch type 74LS373 or equivelent . After one clock cycle , 330ns , IC92a clocks the low at its D input through to its output setting B.ALATCH low via IC99a . The reciving memory card uses this to latch the address into the LS373 so that the bus is no longer needed for address use . At the same time IC93a dissables the address bus buffers and IC93b enables the data bus buffer IC97 , The direction of the data bus buffer is controlled by -DBIN from IC5a . Also at this time IC91a or b pass -DBIN or -WE to the bus via IC99 so that the memory can be read from or written to depending on the direction of transfer requested . As per normal on TI systems a memory ready signal is required to come back from the external memory card , this arrives via IC99 and IC90a , is syncronized by IC101 to form -READY at IC27d . The proccessor performes the transfer and on the next clock cycle IC92b resets the bus to address by setting IC92a via IC21c , IC91a & IC27f .

The only problem with this system is that due to propogation delays in IC99 and down the bus lines the reciving address latch may not get the ALATCH signal untill the address and data bus buffers have begun to change over causing a corrupted address to be latched . This can somtimes be avoided by carefull selection of LS373's and IC99 but it may be better to delay the local -ABE and -DBE signals by including two sections of a 74LS04 between each of IC93a & b outputs and the respective buffers .

CRU Transfers

CRU transfers are a little less complicated , when an external CRU request is made IC27c and IC13d generate a -BRQ . BEN is asserted as in memory cycles but this time IC92a is held set by IC91a so the address bus stays enabled . IC90a passes -BRQ to IC101 which is reclocked to form -READY to IC27d . A CRU clock is output via IC99 if the transfer is output or CRUIN is read via IC99 if input .

More on the LS2001 replacement

There is more in the LS2001 replacement circuit than I have mentioned upto now Firstly the problem with trying to access an external memory location that has no memory present . In such a case no -B.READY signal would be recived and the Cortex would just sit waiting for it forever the only way out being a RESET . The solution is to generate a -READY signal after 16 clock cycles by IC103 in the absence of a -B.READY appearing beforehand this results in a false read from such a none existant location but it is no worse than the false read that would occur from an unused CRU location anyway . Any valid memory would have produced a -B.READY before this time so corruption of valid transfer is not likely . The LS2001 offers not much more as all it would do in such a situation is to cause a level 1 interupt to the microproccessor . The level one interupt just resets the LS2001 timeout and branches back to the programme . One problem with the LS2001 is that it has to be enabled by setting CRU bit 3 before it can produce the timeout interupt and also it wont work if it is the DMA requesting the bus , refer to my DMA E.Bus mod in issue 7 , so the bus hangs up anyway . The other function performed by the LS2001 is concerned with multiproccessor use on the bus , Before access is made to the bus the Cortex should check if it is free by inspecting GRANTIN , which means a higher priority card has access to the bus , and -B.BUSY which means that the bus is in use . If the LS2001 cant gain access to the bus because of one of these signals being present it generates a -TO interupt after 128 clock cycles . If the cortex is the Highest priority card then GRANTIN can be ignored , -B.BUSY could be checked with software if it was

connected to an unused CRU input bit on IC63 , -B.INT could be used if IC16f were removed to prevent level 1 interupt being generated , and a test made of the B.BUSY signal before every access to the bus . Connecting -BEN directly to GRANTOUT prevents lower priority cards trying to access the bus while the Cortex has control .

The next artical will look more closely at the memory card end of the E.Bus and describe a 64K ROM card .

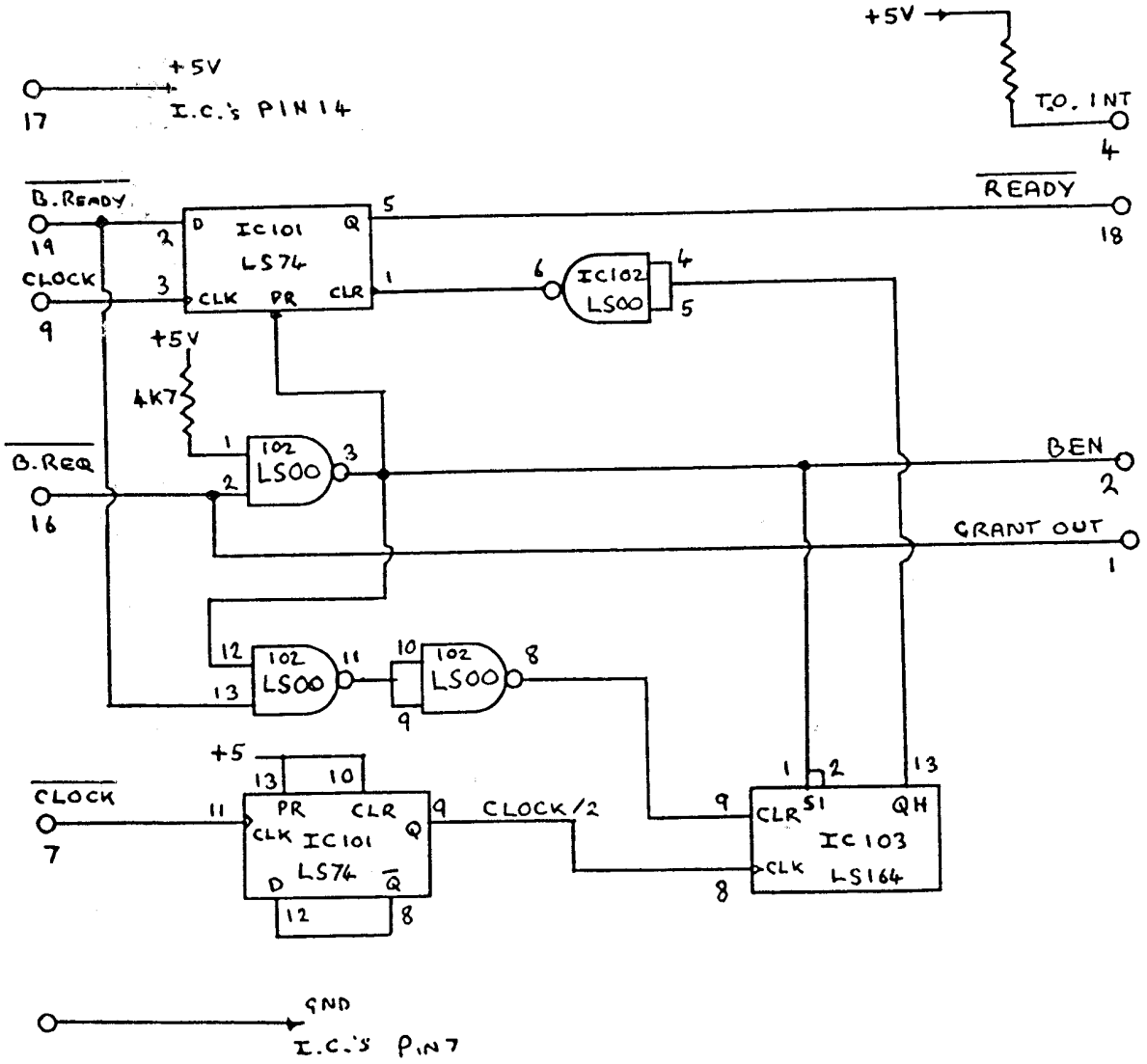


FIG 1. LS 2001 REPLACEMENT TIM.GRAY