CORTEX PART 1

Announcing the ETI Cortex, in all its 16-bit splendour. This advanced design uses up-to-the-minute technology and forms the basis of a powerful home or business system.

Setting records is getting to be a habit at ETI. We were the first magazine to publish a DIY computer (the Triton). And we were the first magazine to publish a full-feature 16-bit computer, and at a price that makes a lot of commercial machines look a bit sick. The Cortex forms the basis of a versatile computer system that expands with your imagination and is based on state-of-the-art VLSI technology; the 'why use five chips if one will do?' philosophy.

Processing Power

The processor is a high-speed 16-bit device which possesses a unique system of RAM-based registers. The Cortex kit is supplied with a full 64K bytes of dynamic RAM (ie 32K words of 16 bits), and 24K of BASIC and assembler in an overlaid memory organisation (we'll explain what that means later). The high definition colour VDU has a separate 16K of RAM outside the CPU's 64K memory map and some extraordinary features that result in superb graphics capabilities. Disc drives may be interfaced easily as the controller chips fit on the PCB, and the resident BASIC can be overwritten by disc-based languages; the first that will be available is UCSD Pascal. The latter features will make the system particularly attractive to business users, and Cortexes (Corti?) will be available ready-built as well as in kit form.

The heart of the Cortex is the TMS9995 CPU. As with all the components in this project it was selected from the wide range of currently available CPUs on a price/performance basis. The 9995 is based on the unique memory-to-memory architecture of the TMS9900. Thus it has the same architectural features of this powerful 16-bit processor and an enhancement of its rich, mini-computer style, instruction set. It is fabricated in state-of-the-art N-channel silicon gate MOS technology, enabling single 5 V operation and high speed (12 MHz) to be achieved in a compact silicon area.

As well as the 16-bit CPU, fabricated onto the same chip are a number of extra features that make this device the obvious choice for a large number of general purpose applications. 256 bytes (128 words) of on-chip RAM enables four complete, fast access register files (workspaces) to be implemented in full speed memory. A clock generator is also included to minimise the number of external components. Also available are a timer/event counter, a prioritised Interrupt Interface and a 16-bit flag register which can be output on the I/O control bus. The I/O bus is completely separate from the main memory map, and enables 32K of individual I/O bits to be manipulated individually or simultaneously in groups.

Artful Architecture

The term 'memory-to-memory' implies the fundamental difference between traditional eight-bit CPU architectures and that of the 9995. That is, all transfers in the machine are from one main memory location to another. Only three 16-bit registers exist on the CPU itself; the Program Counter, the Status Register and the Workspace Pointer.

The Workspace Pointer contains the address of the start of a block of RAM anywhere in the main memory map. This address is designated register zero; the next 15 contiguous memory locations are designated Registers 1–15. These registers may then be used by the

A complete Cortex system.

programmer as scratch registers. A large number of instructions exist that make maximum use of the reduced addressing needed to access these. For example MPY R4, R5, performs a 16-bit multiply of the contents of Register 4 with Register 5 and stores the 32-bit result in Register 5 (most significant word) and Register 6.

What advantage does this give a programmer over other architectures? Well, in addition to providing no restriction on the choice of addressing modes (register 0 can still be accessed as memory location 7XXXX) the power comes when an interrupt or other subroutine call occurs. It is obviously desirable, especially on receipt of an interrupt, to be able to react as rapidly as possible. On a register-oriented machine it is necessary to save the contents of the working registers in main memory, a slow operation requiring a large number of reads and writes to push the registers onto the stack. On the 9995 the context switch occurs by changing the value in the Workspace Pointer. This points to a new area of fresh registers ready to process the interrupt. The full context of the previous operation is retained in the previous workspace registers, which, being in main memory, are still preserved intact. A return is similarly implemented easily and rapidly by restoring the old Workspace Pointer value. In many real time control situations with a large number of external events occurring this architecture is the only one that allows for efficient processing.

External Affairs

The system clock is externally generated and fed to the CPU via the XTAL2 input. A clockout signal is also provided that is one-fourth of the crystal frequency (3 MHz). The 64K bytes of system memory are directly addressed by A0-A15. Here the convention is that A0 is the most significant bit. A0-A14 are also used to directly address the separate I/O structure of the Communication Register Unit (CRU). The bit to be accessed is held on A0-A14 and the data is present on A15. The data is then clocked out by $\overline{\text{CRUCLK}}$. Reading a CRU bit into the CPU is achieved in the same manner but is read into the CRUIN line.

The data bus is multiplexed from the internal 16-bit architecture to eight bits externally, D0-D7. A memory access is signalled by $\overline{\text{MEMEN}}$ and data direction is controlled by $\overline{\text{DBIN}}$. $\overline{\text{WE}}$ indicates a memory write. For the control of slow memories a READY line signals to the CPU if the memory is ready to complete the current access. If not, the CPU waits for another CLKOUT cycle before continuing.

In the Cortex all these features are used to provide the fastest BASIC available to a home constructor. Even then the speed is still limited by I/O transfers. This can be seen as evidenced by the amount of time the 'idle' LED is on. This LED is directly driven by a status decoder that indicates when the CPU is no longer executing any instructions but is waiting for an external interrupt.

Dynamic Designing

We don't doubt that all you digital dabblers have watched the plummeting price of memory in our advertisers' pages, and the most dramatic trend has been in dynamic memory (DRAM). This is partly due to the simplicity of the basic cell design in a DRAM; and this simplicity means in turn that the most dramatic cell density increases have been and will be made in DRAMs. With 16K DRAMs prices have pretty much levelled out at the bottom of the curve, but the 64K DRAMs are in hot pursuit.

More importantly from the technical point of view, certain 64K DRAMs offer upward compatibility to the next generation of 256K DRAMs. (Why are the steps quadrupling: 16K to 64K to 256K? Because DRAMs have a two-part

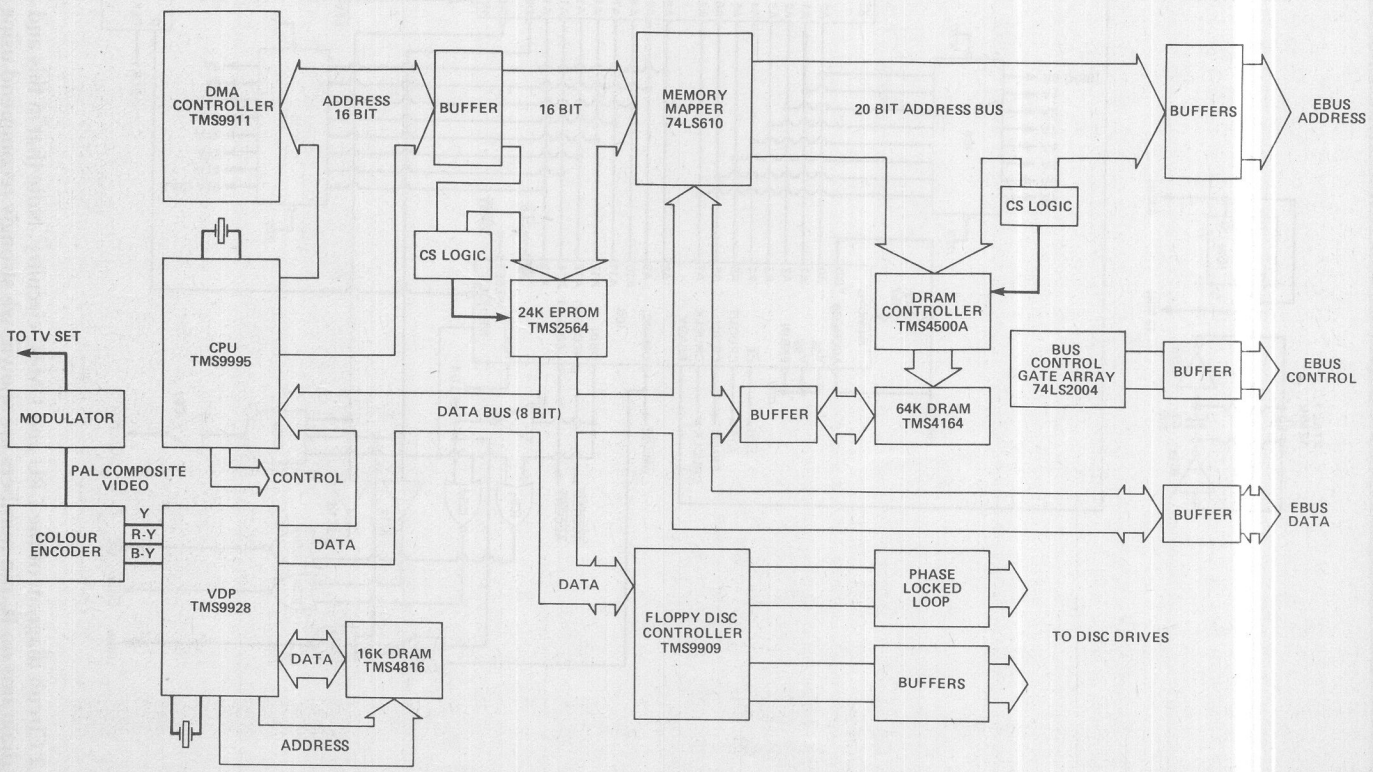
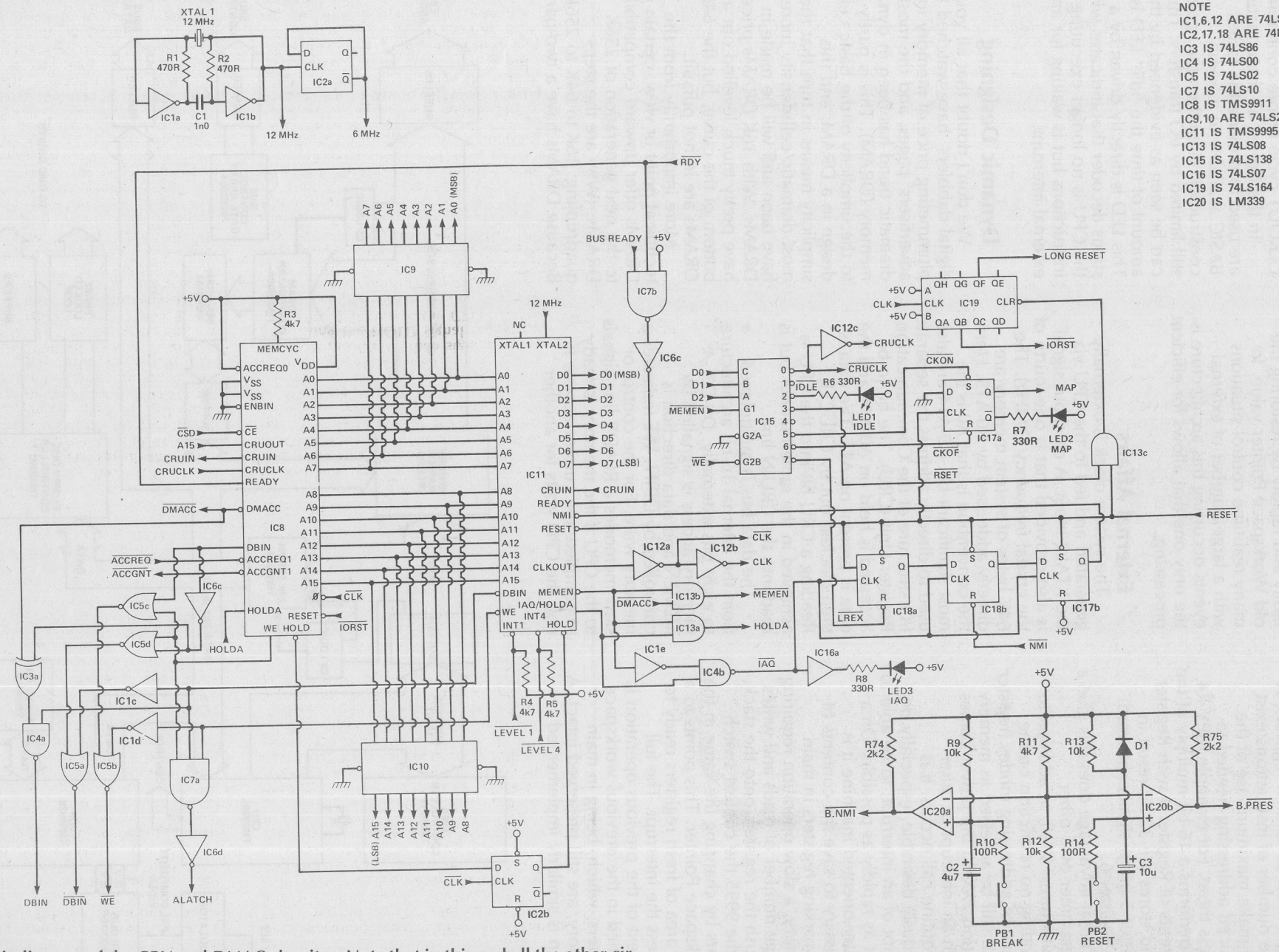


Fig. 1 Block diagram for the complete Cortex.



NOTE
IC1,6,12 ARE 74LS04
IC2,17,18 ARE 74LS74
IC3 IS 74LS86
IC4 IS 74LS00
IC5 IS 74LS02
IC7 IS 74LS10
IC8 IS TMS9911
IC9,10 ARE 74LS244
IC11 IS TMS9995
IC13 IS 74LS08
IC15 IS 74LS138
IC16 IS 74LS07
IC19 IS 74LS164
IC20 IS LM339

Fig. 2 Circuit diagram of the CPU and DMAC circuitry. Note that in this and all the other circuit diagrams no IC pin numbers are shown; we strongly recommend using the special PCB for this project.

HOW IT WORKS — CPU AND DMAC

The heart of the system is the CPU, IC11 (a TMS9995). It has a 16-bit internal architecture and an eight-bit wide external data path. The master clock for the system is formed by IC1a,b and associated components; the 12 Mhz clock rate of the CPU enables it to complete a memory read or write in only 166 nS! This is too fast for present DRAM technology so the automatic wait state feature of the CPU is used. This automatically assumes that memory is not ready and extends the memory access to 500 nS. The cycle can be further extended by a low level on the READY input to the CPU; this occurs, for example, when the DRAM is not ready because a refresh cycle is taking place.

The CPU signals the type of memory cycle by driving either DBIN or WE (write) low after driving MEMEN low. If the memory cycle is an instruction fetch then the IAQ/HOLDA signal goes high until both bytes have been fetched. This condition is decoded by IC6e, IC14a and buffered by IC16a to light LED3 to provide a front panel indication.

The CPU has a bit-mapped I/O interface which is separate from the memory data bus; the process is carried out by a section of the CPU called the Communications Register Unit (CRU). The data transfers are serial, bit by bit, each bit having a unique address. This allows 32K bits to be accessed (not 64K since address line A15 carries the data). The value of the data bit is on CRUOUT (A15) for output cycles and a WE/CRUCLK pulse is generated to strobe the data into the I/O devices. On input cycles the data is sampled from the CRUIN line and a pulse is generated on the DBIN line to enable the bus buffers and so on. During all serial I/O operations the MEMEN signal stays high. Any number of bits from one to 16 can be transferred, each bit taking 500 nS to transfer if the READY input is high.

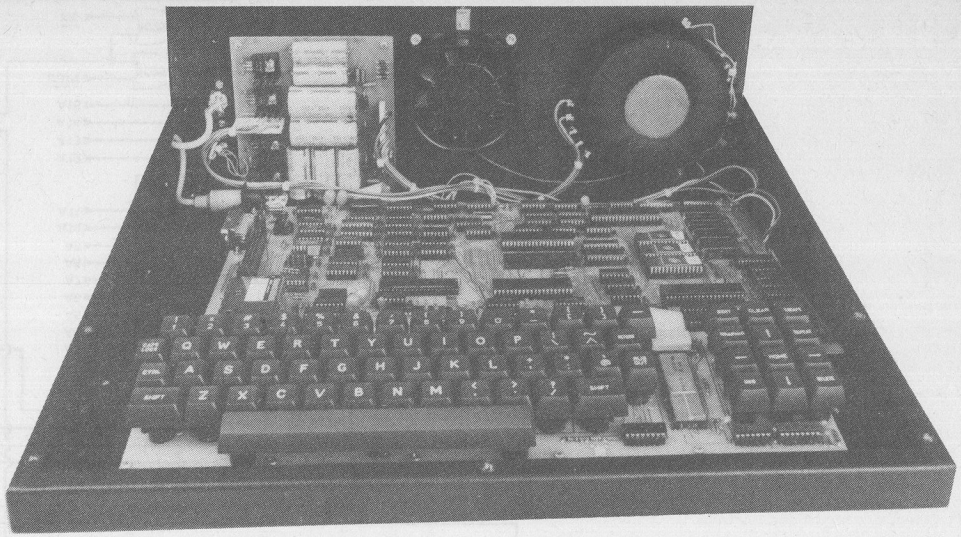
There are some special I/O signals for control use called IDLE, LREX, CKON, CKOF and RSET. IC15 decodes these separately from the normal I/O operations by using the three-bit code output on D0–D2 of the data bus. IDLE pulses continuously whenever the IDLE instruction is executed; it indicates that the CPU is in an internal loop waiting for interrupts (ie doing nothing). LED1 on the front panel lights to indicate this state. LREX is used for the single instruction execution logic which causes an NMI interrupt to occur two instructions after executing the LREX instruction. The two-instruction delay is generated by the series flip-flops IC18a, IC18b and IC2b.

CKON and CKOF are used to switch the memory mapper device from passive to active and vice versa: the signals set or reset the Q output of IC17a to enable or disable the memory mapper (IC26) via IC24a, 25a. When Q is high, Q is low, and LED2 lights to signal that external memory is being accessed.

The RSET signal causes all I/O devices to be reset and sets the CPU interrupt mask to disable all interrupts. Normally both RESET and RSET are high, so the output of IC13c is high and IC19, an eight-bit parallel-out serial shift register, clocks out a continuous series of 1s. A low on RSET or RESET sets all the outputs of IC19 low (specifically IORST and LONG RESET); when the CLR input returns high, 1s are clocked through the shift register first taking IORST then LONG RESET high.

The Direct Memory Access controller (IC8, a TMS9911) is used to provide transparent high speed data transfer to and from the floppy disc controller (FDC) into memory. The address bus of the CPU is tri-state, as are the address outputs of the DMAC. Only one device is in control of the address bus at any one time. When the FDC requires the memory it signals on ACCREQ (access request); the DMAC then signals to the CPU using the HOLD signal that it requires the bus. When the CPU reaches the end of the current memory cycle it tri-states all its outputs (except MEMEN) and signals HOLDA ('acknowledged'). The DMAC now takes over the bus, signals ACCGNT to the FDC, and transfers the data byte between the memory and the FDC.

After completing the memory cycle(s) the DMAC then relinquishes control back to the CPU by releasing HOLD. The CPU then continues as if nothing had happened. The TMS9911 was designed for use with the TMS9900 CPU; when it is used with the TMS9995, extra gating is required to make the signals compatible. Parts of ICs 3, 4, 5, 6 and 7 take care of this. In this application only one of two channels in the DMAC is used; the other is free for the user to experiment with.



multiplexed address bus, so adding one extra address pin is the equivalent of two extra address lines and thus four times the addressing range.) Since all 16K DRAMs operated on a 128-cycle refresh, some 64Ks followed suit. However, that extra address pin called for twice the number of sense amplifiers on the chip, which occupied valuable silicon area (see this month's article on Designing Micro Systems for more about DRAM structure). For 256K DRAMs the waste of chip area is intolerable and to get a product that is capable of manufacture, 256-cycle refresh is essential. The TMS4164 64K DRAM, the first commercially available production device, adopted 256-cycle refresh from the start, as well as following the JEDEC-approved pin-out. This means that the devices are not only upward pin-compatible from the TMS4116, but will also be upward compatible in the future to 256K devices. Simply plug new chips in the old sockets and you've got four times the memory!

For these reasons we chose the TMS4164 to provide a full 64K memory map using only eight chips. Not only is it compact, fast and very reliable, but it is the first 64K device to be successfully encapsulated in a low-cost plastic package.

In our application, refresh is achieved in a manner typical of the Cortex philosophy; a single-chip DRAM refresh controller is used, the TMS4500A. This device provides all the necessary control and arbitration functions for handling 64K DRAMs in a microprocessor system. It accepts the 16 address lines, A0–15, and multiplexes them to row address and column address (RAS and CAS) at the appropriate times, generates refresh signals for 128 or 256-cycle memories and arbitrates synchronously between access requests and refresh cycles. Synchronisation is important for achieving reliable arbitration, and the CPU clock is utilised as the main timing reference.

ROM To Manoeuvre

Similar design criteria were applied to the choice of EPROMs to store the system firmware. In order to economically store the large number of bytes required to provide

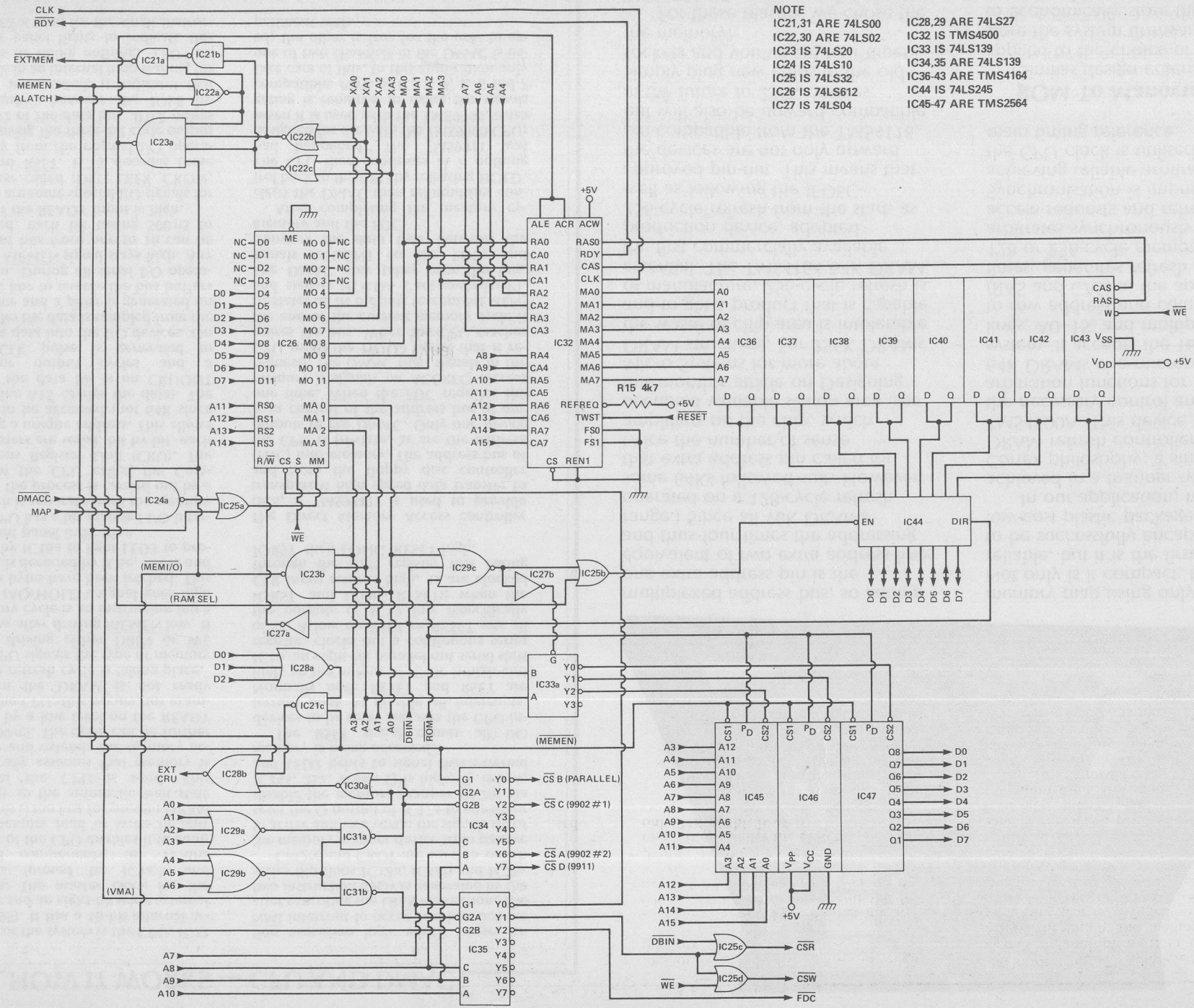


Fig. 3 Circuit diagram of the memory section.

HOW IT WORKS — MEMORY

The 24K of EPROM (IC45, 46, 47) contains the assembly language support and the BASIC interpreter. The EPROMs are switched in and out of the memory map by the I/O bit 'ROM' (see I/O section). This signal powers up in the active (low) state, with the EPROMs on. The DRAM (ICs 36–43) is also accessed during a read of the EPROMs but the data buffer IC44 is not enabled; this means that any write while the EPROMs are on is put into the DRAM, so that it behaves as a 'phantom' or ghost. The BASIC interpreter copies itself into the DRAM and then switches the EPROMs off. This has two advantages; first, during execution the interpreter overlays sections of code and then re-copies the relevant section of EPROM back to conserve memory. Second, to enable disc-based operating systems (eg Pascal) to be used, the system needs to be able to operate RAM-resident.

The addresses for DRAM accesses are passed through the memory mapper device, IC26. This segments the CPU's 64K address map into 16 pages of 4K; each page has a 12 bit register (MO0 to MO11) of which only eight bits are used (MO4 to MO11). The outputs replace the top four bits from the CPU and add four more. Thus each 4K block can be anywhere in the 1M total address reach (20 bits). The CPU at any one time still only has a 64K address map but by dynamically loading the mapper during program execution the full 1M can be used. The mapper registers are loaded or read as 16 memory locations in the memory-mapped I/O area in high memory.

The chip select logic defines the first 64K as on-board memory and the remaining memory map as off-board, using the E-BUS interface (see later). The bottom 32K of memory has the 'phantom' DRAM under the EPROM although only 24K of this is used. The DRAM occupies 60K of the memory map; the top 4K is sub-divided into 256 bytes of high-speed internal to the CPU, and then a memory-mapped I/O region for the Video Display Processor, the memory mapper and the floppy disc controller. Eight memory-mapped slots are decoded, leaving some spare chip selects for user experimentation and expansion.

The DRAM controller (IC32) takes the 16-bit least significant address signals and multiplexes them on outputs MA0–7 to supply the memory devices with the correctly-timed waveforms. Once an access request is made, the addresses are first latched and then the controller arbitrates between a refresh cycle and an access cycle. If the controller is busy refreshing the memory then it signals a 'not ready' state to the CPU on the READY line, which suspends operation until the signal returns high again. The controller has to use 'cycle-steal' refresh, as the CPU is nearly always accessing memory and not enough free time can be guaranteed. The refresh cycle obviously slows down the CPU, but by less than 10%, which is a small penalty to pay for the large amount of memory at a low cost.

a comprehensive and versatile BASIC language, high capacity devices were required. The cost and capacity trends in EPROM technology have followed a similar path to DRAMs; thus the TMS2564 was chosen to complement the TMS4164 in the system. These devices, like all the major devices in the project, operate off a single 5 V rail.

Each 2564 can store 8K of program organised in the now industry-standard 8K×8 format. Three chips are used to store the firmware: 24K in all. However, an important design feature of the Cortex should be mentioned here; the EPROMs phantom the DRAMs in the memory map. At power-up the EPROMs are enabled, and after checking the DRAMs the program then copies the full operating system from EPROM into RAM. Once this has been completed the EPROMs are disabled. Thus the operating system is running in RAM, allowing changes to be made or sections deleted to create extra space. This is most noticeable when you're only operating the assembler section, since the whole of the BASIC interpreter may then be eliminated from memory, freeing an extra 20K for the user! Thus the Cortex is the only computer offering a full 64K of RAM to the user as a standard feature!

Graphic Descriptions

No home computer is considered worthy of the name these days unless it features impressive graphics capabilities, and the Cortex certainly won't disappoint you in this respect (just look at our cover and the photos in this article for some examples of the display power). What's more, all this is controlled with just one IC, the TMS9929! This chip is a memory-mapped video controller, which produces all the relevant video signals internally; it's a very versatile device capable of producing extremely complex displays, including 3D simulations. In order to achieve this its mode of operation is very different to the majority of display devices currently in use.

For simple text displays the video output of a computer may be produced by using a small random access memory to address a larger read-only memory that stores the character shape. Teletext displays fall into this category. The next step up from this is to have a display where each displayed dot is a bit (or bit pattern) stored in RAM. The first method means that shapes can be positioned very rapidly but the repertoire of shapes is limited to those in the graphics/text ROM. The second scheme is slower but allows more complex shapes to be created and lines to be plotted. However, problems occur when trying to overlay shapes as everything must be done in software — making things even slower.

The 9929 uses variants on both these schemes to produce extremely complex shapes with the minimum of software overhead. Tables are designated in an area of RAM to define pattern shapes, characters or graphics. A separate table area is designated to define attributes to the shape, such as colour and size. Finally an area of RAM akin to the Teletext RAM contains pointers for each screen location that point to the desired shape and its associated attributes. The advantage of this system is that large, dramatic changes can be made to the display very rapidly by making simple changes in the pattern look-up table. Further, all displays of one particular shape can be modified simultaneously by updating the pattern generator table.

A distinction is made in hardware between graphics and text modes, though text is available in graphics mode. They may be interchanged using the BASIC commands TEXT and GRAPH. In GRAPH mode the resolution is 256×192 (48K pixels), and the colour of each pixel can be set to any of the 16 available display colours, the only limitation being that only two colours per eight pixels may be used horizontally. To put it another way, the background may be any one of the 16 colours, while the colours of the pixels (foreground) can be set to one of 16 colours in blocks of 32×192. Control is exercised from BASIC using the COLOUR and PLOT commands.

In TEXT mode the screen format is 24 rows of 40 characters per row, each character being defined by a 6×8 bit matrix. In this case an attribute table is not required; instead the character colour is defined by a colour register on the 9929 itself that stores one foreground and one background colour. The character shapes are held in RAM in the VDP's memory

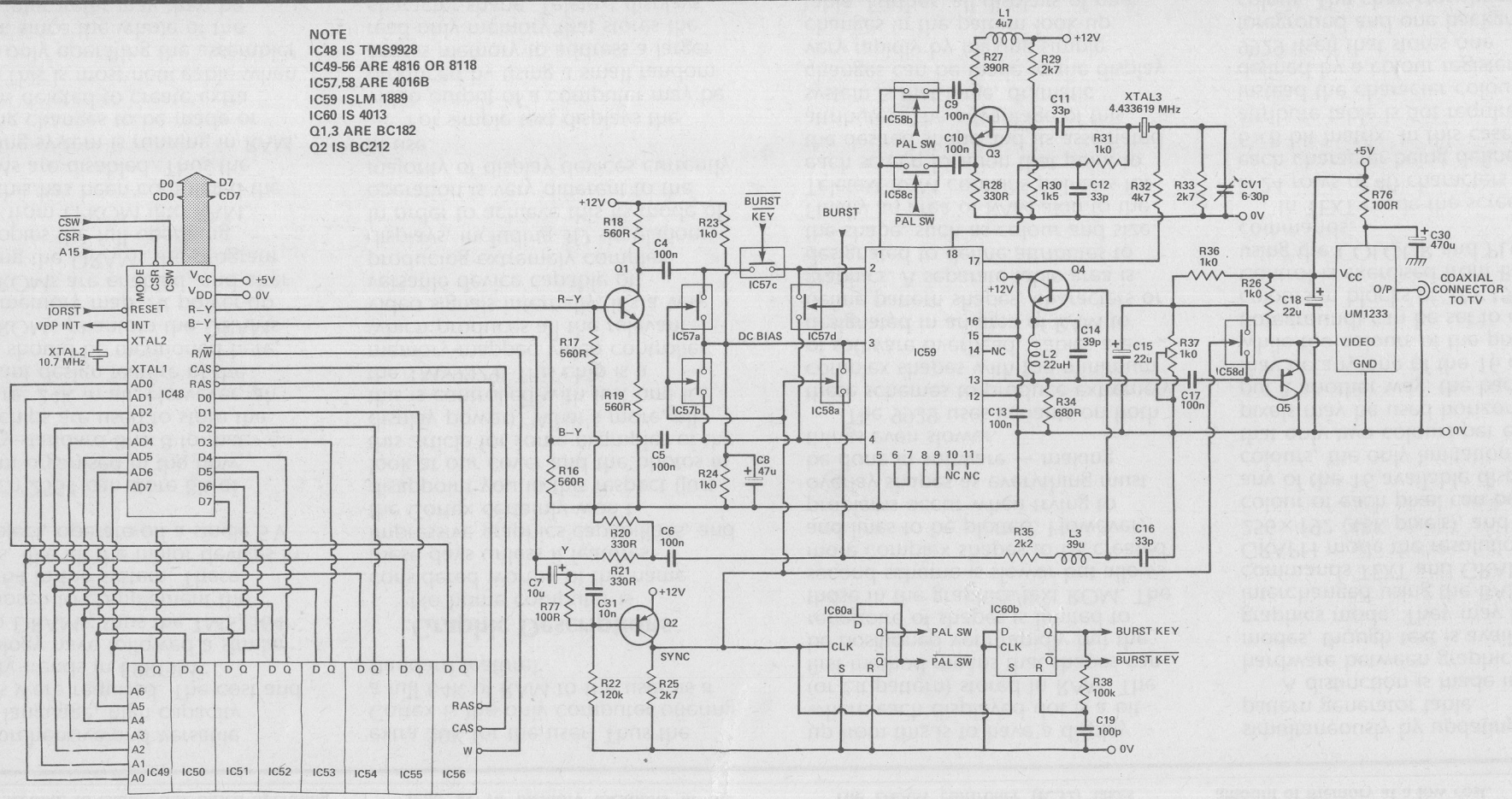


Fig. 4 Circuit diagram of the video display circuitry.

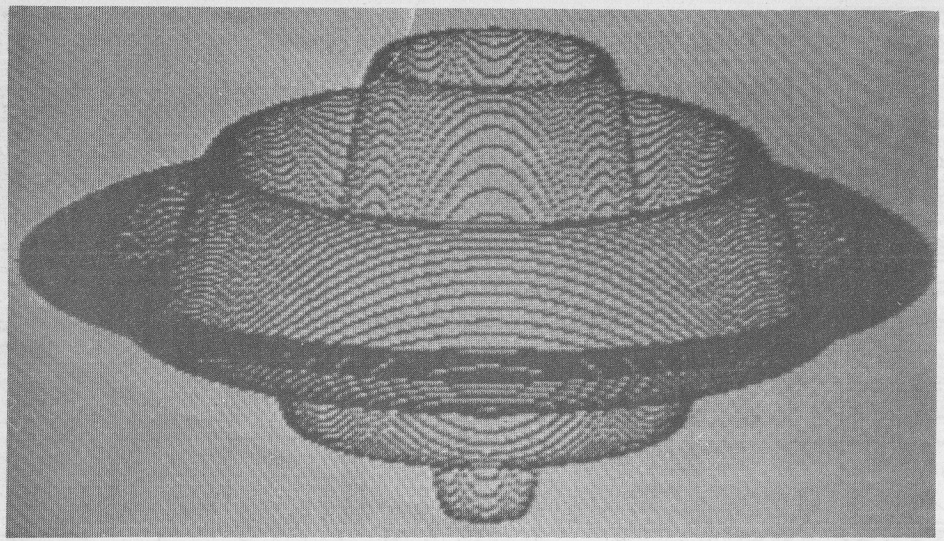
HOW IT WORKS — VIDEO DISPLAY PROCESSOR AND PAL ENCODER

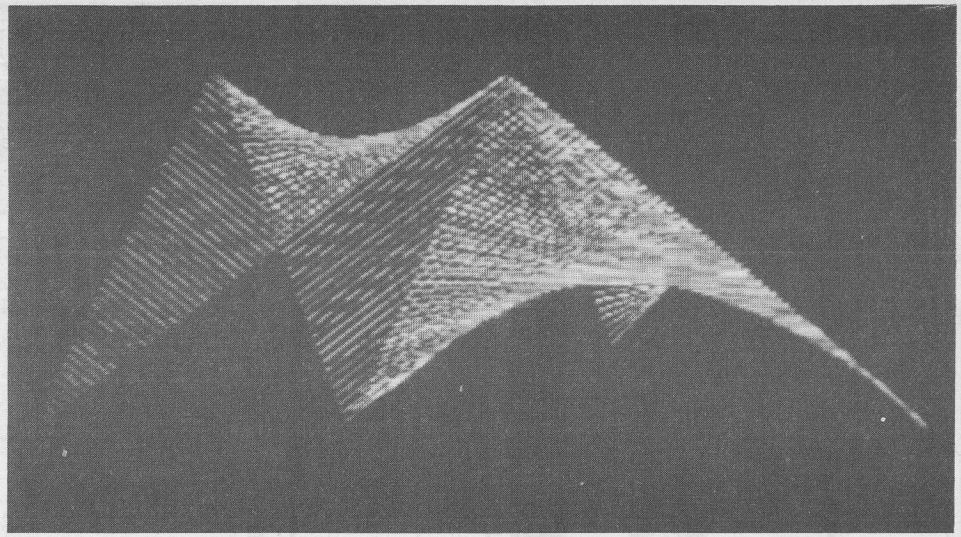
The TMS9929 (IC48) is a 625-line non-interlaced video display processor; it directly drives 16K of memory which is completely separate from the main CPU memory. The VDP fetches data from its DRAM (ICs 49–56) at such a rate that the DRAM is automatically refreshed many times over. There's very little else to say about this section of the circuitry — IC48 does everything internally! The VDP outputs a composite luminance and sync waveform on the Y output and colour difference signals on the R–Y and B–Y outputs. These signals contain all the information needed to produce either R–G–B or PAL-encoded colour. The VDP is controlled by a two-byte memory-mapped slot in high memory.

The sync is separated from the Y (luminance) signal by Q2 and associated circuitry, and used to drive DC restoration clamps IC57a,b,58a; these charge capacitors C4, 5, 6 to a reference voltage during the sync pulse. The sync is also used to toggle the PAL (Phase Alternate Line) switch, IC60a, which gates an inverted or non-inverted chroma oscillator signal into one of the two analogue multipliers in IC59. The chroma oscillator is built around Q3 and XTAL3, a crystal whose resonant frequency is 4.433619 MHz, that of the PAL colour subcarrier. The non-inverted signal is taken from the collector of Q3 via C9 and IC58b, while the inverted signal is taken from Q3 emitter via C10 and IC58c.

The second flip-flop, IC60b, is used as a monostable which, at the beginning of each line, connects the burst pulse which occurs on the B–Y signal to the R–Y input of IC59. This switching is done by IC57d. The inverting amplifier Q1 on the R–Y line from the VDP, IC48, is to match the direction of the burst pulse with the direction of the video signal to yield the correct colours.

The luminance signal is low-pass-filtered by R28, L3, C16 and then summed with the chrominance output of IC59 via the chroma trap L2, C14; this filter removes colour fringing effects. The signal is then DC-shifted by another DC restoration clamp (IC58d) to feed the RF modulator.





and so may be easily user-modified. The CHAR command allows any of the 256 possible character definitions to be altered.

Table 1 shows the 16 colours which are available; this 'palette' has been arranged to give not only a good colour display, but also a good monochrome display, as the colours produce an even grey scale on a black-and-white TV. Eight grey levels are generated.

One peculiarity may have caught your eye in Table 1: what is the point of a transparent colour? A transparent object will allow you to see what's behind it, but in most graphic displays 'behind' is meaningless. However, the VDP in the Cortex considers its display to consist of 36 planes prioritised one above the other. When you look at the screen you're seeing an image which can be considered analogous to holding 36 colour slides, one above the other in a stack, and peering through them all.

TABLE 1

	Code Colour	Code	Colour
0	transparent	8	medium red
1	black	9	light red
2	medium green	10	dark yellow
3	light green	11	light yellow
4	dark blue	12	dark green
5	light blue	13	magenta
6	dark red	14	grey
7	cyan	15	white

The rearmost plane is black to allow images to be built up over it. The next plane is for external video and need not concern us here. On top of this is the backdrop plane which lies directly behind the text/graphic plane. This defines the border colour as well. Since this plane defines the colour of the whole screen, it is now obvious that the only way to see the external video input or the black rearmost plane is to set the backdrop to transparent. The text/graphic plane is written to by the TEXT and GRAPH commands discussed earlier.

This leaves another 32 planes sitting in front of the four mentioned above; these are called the sprite planes. A sprite is a graphic shape that can be user-defined from BASIC with the SPRITE command. Sprites can be displayed in a variety of sizes depending on the size and magnification flags; these give four possible modes. SIZE 0 means that a block of 8×8 bits is used to define the sprite, while SIZE 1 uses 16×16 bits (but reduces the total number of different shapes from 256 to 64). The display size can be varied with the MAG command; MAG 0 maps one bit in the shape onto one pixel while MAG 1 maps one bit onto a 2×2 block of pixels on the screen.

Each sprite has four attributes associated with it; its plane (or priority), its colour and its X and Y screen coordinates. Again, each sprite can be one of 16 colours, and those bits set to 1 in the sprite definition adopt the defined colour while the other bits are set to transparent. The screen coordinates define the position of the top left-hand corner of the sprite, and sprite positions can therefore be rapidly changed by simply altering two bytes in memory. The colour can be changed equally quickly by altering one byte. Because the planes are prioritised, 0 to 31, if any shape is positioned coincident with another shape, only the one with the highest priority plane will be displayed. This gives rise to simple 3D simulation. A status flag is set to indicate when any two sprites 'touch' each other. Any point in the text or graphic plane will only be seen if all the points directly above it on the 32 sprite planes are transparent.

All these features mean you can generate very versatile and complex displays; but they also use up a reasonable amount of memory. We don't believe that screen RAM should be stolen from the user RAM, so the VDP only occupies two bytes in the CPU memory map. These two registers are all that's required to write all the relevant information through the VDP chip and into its own 16K of DRAM.

ETI

CORTEX PART 2

Build yourself a better brain: in this article we explain the remaining Cortex circuitry and the construction of the main board.

Serial I/O on the Cortex is handled by a versatile UART, the 9902. The CPU communicates with the UART via its serial I/O bus, based on the Communication Register Unit or CRU, which requires only three wires; thus the device fits easily into an 18-pin package. The 9902 is fully programmable and the range of variations is so great that it's outside the scope of this article. In the Cortex the chip is configured to handle RS232 eight-bit codes with even parity and 1½ stop bits; the communication rate can be set from BASIC using the BAUD command and the device is activated using the UNIT statement. The parameter for UNIT is a 16-bit word, each bit corresponding to a channel that can be either on (1) or off (0).

Channel 0 is the keyboard/screen channel; channel 1 is the 9902 that is already wired into the PCB. Channels 2–15 are implemented in software and only require the addition of extra 9902s on the CRU bus. The Cortex powers up set to UNIT 1. Executing UNIT 2 disables the keyboard and passes control to the 9902. UNIT 3 enables both simultaneously.

TABLE 1

SIZE	DDEN	TRANSFER RATE (kHz)	DIVISION RATIO (IC87)	MONOSTABLE PERIOD (uS)	COMMENTS
0	0	125	12	3.0	5¼" single density
1	0	250	6	1.5	5¼" double density
0	1	250	6	1.5	8" single density
1	1	500	3	0.75	8" double density

Construction

The main board and the keyboard both have plated-through PCBs, ie there are tracks on both sides and connections between the sides are made by the copper that has been plated onto the sides of each hole. There are therefore no track-link pins; it is, however, good practice to apply solder to EVERY hole to reinforce the connections which in some cases carry power. This happens automatically when boards are 'flow soldered' by

HOW IT WORKS—I/O

The I/O map space is split into two regions; the bottom region is for on-board I/O devices and the top region causes an off-board access. (The CPU has an internal I/O area of 16 bits, some of which is reserved for specific hardware functions; the rest is free for the user.) The on-board I/O area of the Cortex is decoded by IC34 into eight 32-bit slots, of which only four are used. Two slots (CS A and CS C) are used for the Asynchronous Communications Controllers (ACCs), the third (CS B) for the parallel I/O for the keyboard data, flags and control lines (such as 'ROM', mentioned in the Memory section), and the fourth for the DMA controller IC8 (CS D).

NOTE
IC16 IS 74LS07
IC61 IS 74LS74
IC62,63 ARE 74LS251
IC64 IS 74LS259
IC65 IS 74LS32

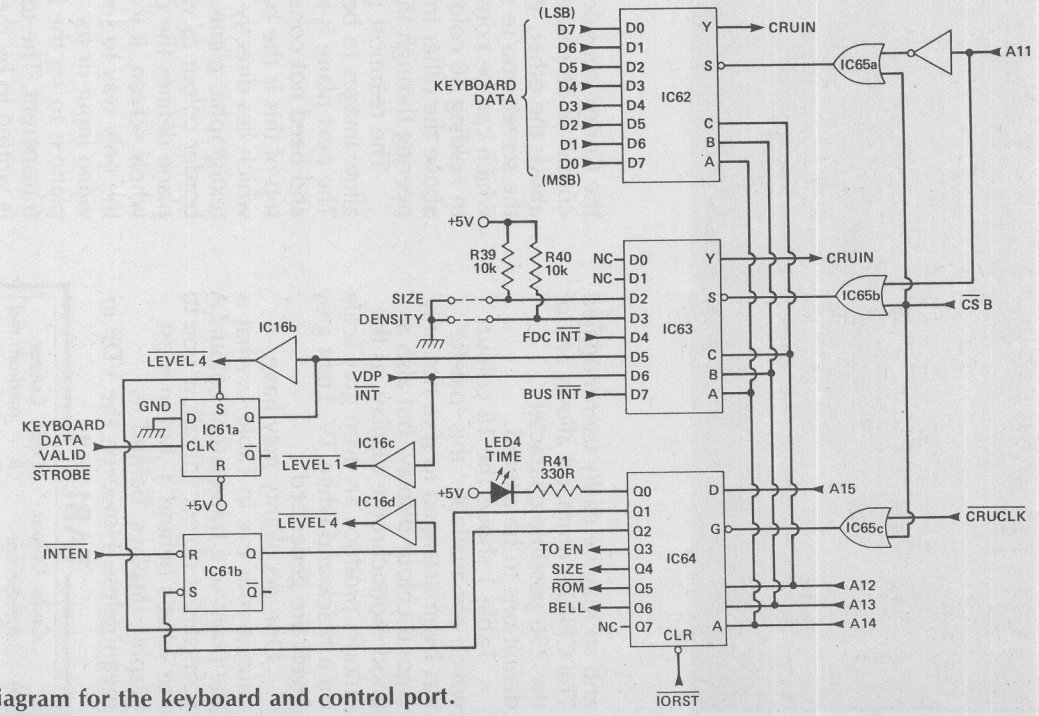


Fig. 1 Circuit diagram for the keyboard and control port.

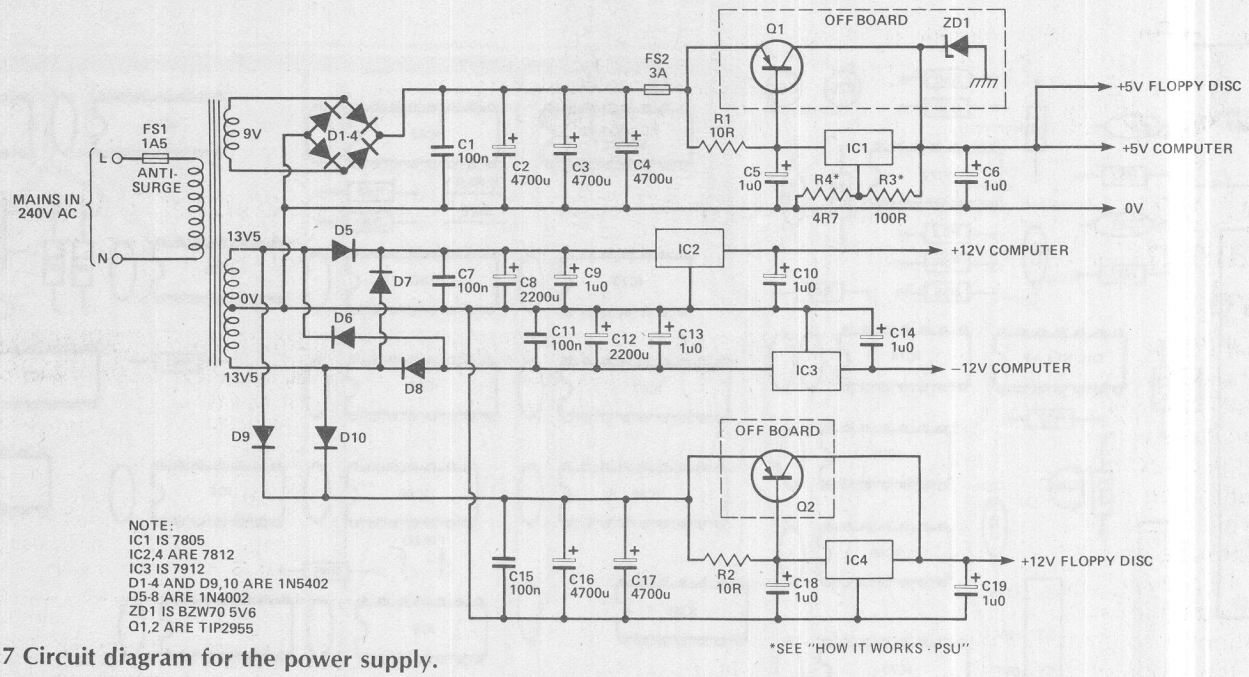


Fig. 7 Circuit diagram for the power supply.

HOW IT WORKS — PSU

The computer main board and keyboard together require a 5 V at 3 A supply, together with low current ± 12 V rails. One amp plastic voltage regulators on small finned heatsinks are used for the 12 V supplies; for the 5 V supply a 1 A regulator is also used but the current-carrying capacity is boosted by bypassing it with a 15 A power transistor, the base current of which passes through the regulator. R1 prevents the off-load input current of the regulator from turning on the transistor when there is no load during testing. The resistor also increases the speed of operation of the transistor. The 1uf capacitors are for the stability of the regulator and the 100nF capacitors are used to remove fast transients orginating from the mains. The zener will clamp any spikes that reach the output.

R3 and R4 increase the output voltage of the 5V supply and may be needed if the volts at IC48 are low (less than 4.8V); otherwise omit R3 and replace R4 with a link.

To simplify the addition of floppy discs these are powered from the same board. The drivers require about 0A7 at 5 V which is also supplied by Q1; they also require + 12 V at 1A6 with higher surges at switch-on, and this is provided by a separate section using Q2 controlled by IC4.

PARTS LIST — MAIN BOARD

Resistors (all ¼W 5% except where stated)

R1,2	470R
R3-5,11,15,-32	4k7
R6-8,20,21,-28,41	330R
R9,12,13,39-,40,46,52,55-,61	10k
R10,14,45,4-7,58,63,77	100R
R16-19	560R
R22	120k
R23,24,26,3-1,36	1k0
R25,29,33,68	2k7
R27	390R
R30	1k5
R34	680R
R35,60,74,75	2k2
R37	1k0 vertical preset pot
R38,53,54	100k
R42	6k8
R43	3k9
R44	39k
R48-50	8k2
R51	M0
R56,59,68	4k7 resistor array
R57	22k
R62	27k
R64-67,71	150R resistor array
R69	5k6
R72,73	3k3
R76	10R

Capacitors

C1	1n0 ceramic
C2,18	16 V PCM mounting electrolytic
C3,7,25,26	10u 16 V PCB electrolytic
C4-6,9,10,1-3,17	100n polycarbonate
C8	47u 10V PCB electrolytic
C11,12,16	33p ceramic
C14	39p ceramic
C15,27	22u 16V PCB electrolytic
C19	100p ceramic
C20	22n polycarbonate
C21,31	10n polycarbonate
C22	330n polycarbonate
C23	2n2 polycarbonate
C24	5n6 polystyrene
C28	100u 16V PCB electrolytic
C29	330p ceramic
C30	470u 10V PCB electrolytic

Semiconductors

IC1,6,12,27,81	74LS04
IC2,17,18,61,69,88,92	74LS74
IC3	74LS86
IC4,21,31,93	74LS00
IC5,22,30	74LS02
IC7,24	74LS10
IC8	TMS9911
IC9,10,84,94-96,98,99	74LS244
IC11	TMS9995
IC13,77,90	74LS08
IC15,34,35	74LS138
IC16,66,80,82,83	74LS07
IC19	74LS164
IC20,79	LM339
IC23	74LS20
IC25,65,78,91	74LS32
IC26	74LS612
IC28,29	74LS27
IC32	TMS4500
IC33,85	74LS139
IC36-43	TMS4164
IC44,97	74LS245
IC45-47	TMS2564
IC48	TMS9929
IC49-56	4816 or 8118
IC57,58	4016B
IC59	LM1889
IC60	4013
IC62,63	74LS251
IC64	74LS259
IC67,68	TMS9902
IC70	74LS123
IC71	75189A
IC72	TL084
IC73	74LS73
IC74	75188
IC76	TMS9909
IC86	74LS297
IC87	74LS163
IC89	74LS2001
Q1,3,4	BC182
Q2,5	BC212
Q6	BC212
D1-4	1N4148
LED1-4	LEDs to choice

Miscellaneous

L1	4u7H
L2	22uH
L3	33uH

PCB (see Buylines); case (see Buylines); IC sockets; I/O connectors to suit; UHF modulator (UM1233 or UM1286).

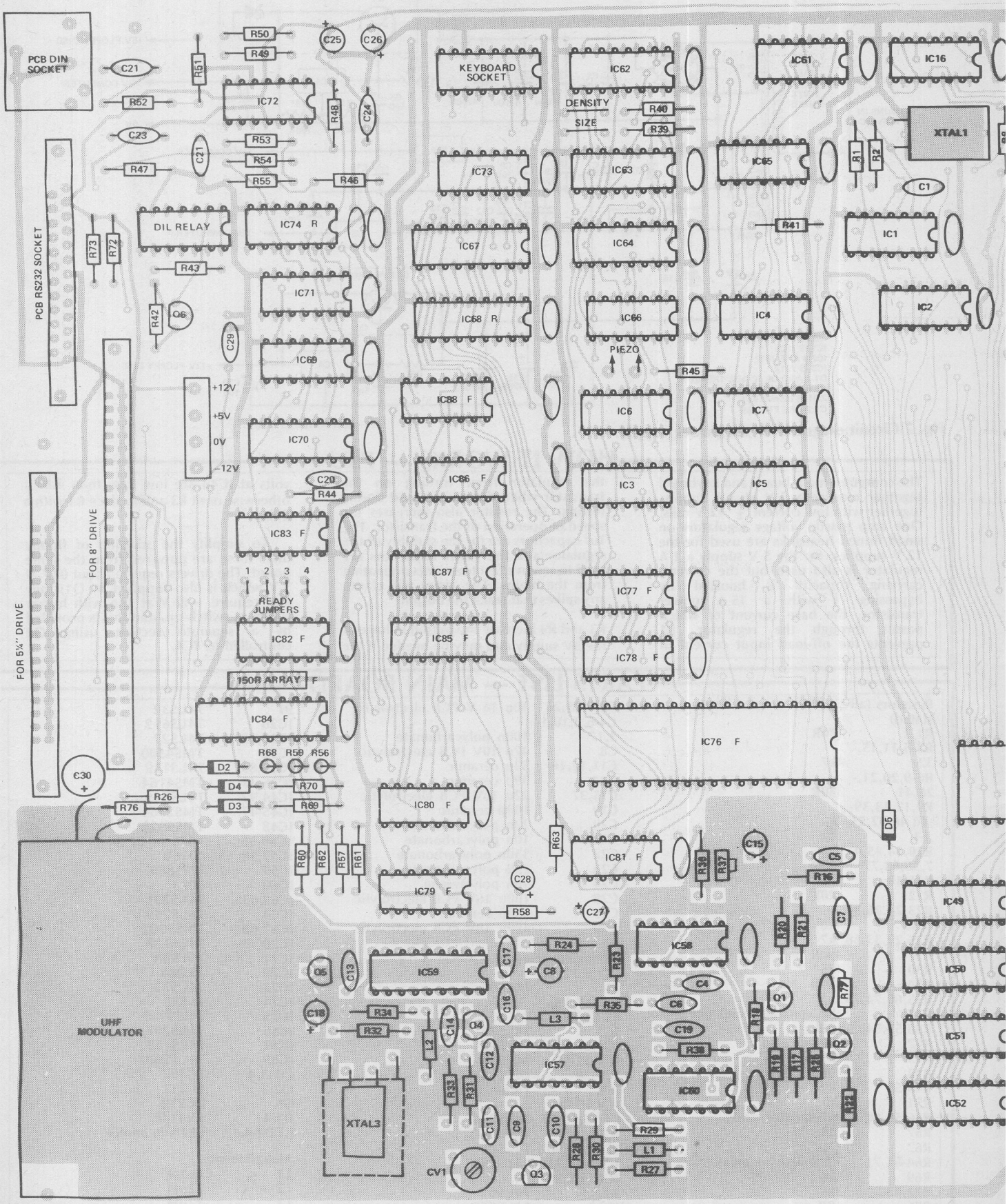
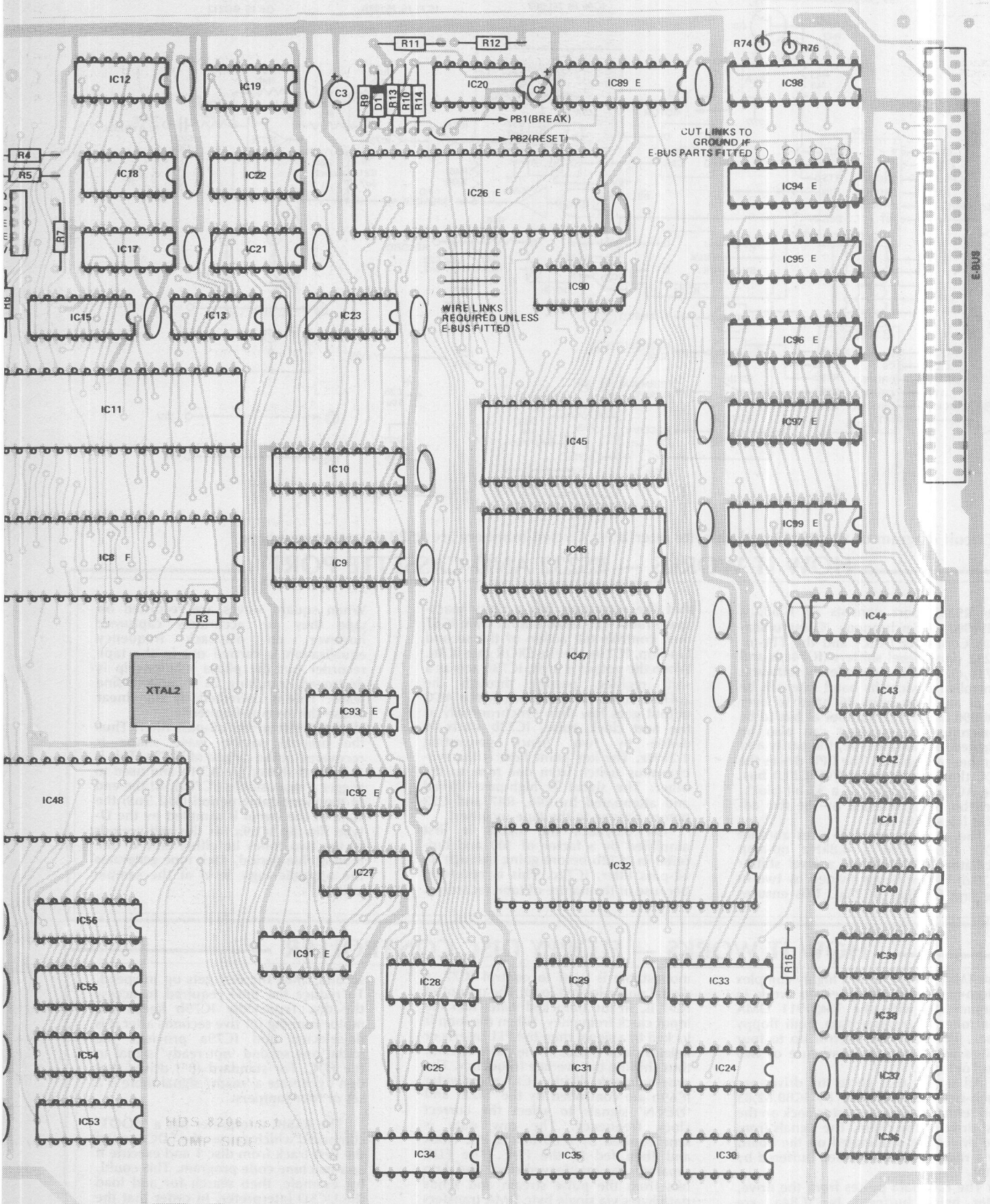


Fig. 4 Component overlay for the Cortex main board. The numerious unmarked unpolarised capacitors are all supply line decoupling capacitors and are 47n ceramic. The tracking shown is of the top foil of the earliest version and may differ on later generation boards (as may the patterns shown for the PSU and keyboard). ICs marked with various letters are for expansion options and are



not supplied with the most basic version of the kit. The expansion options are as follows: R = RS232; F = floppy disc interface; E = E-BUS interface.

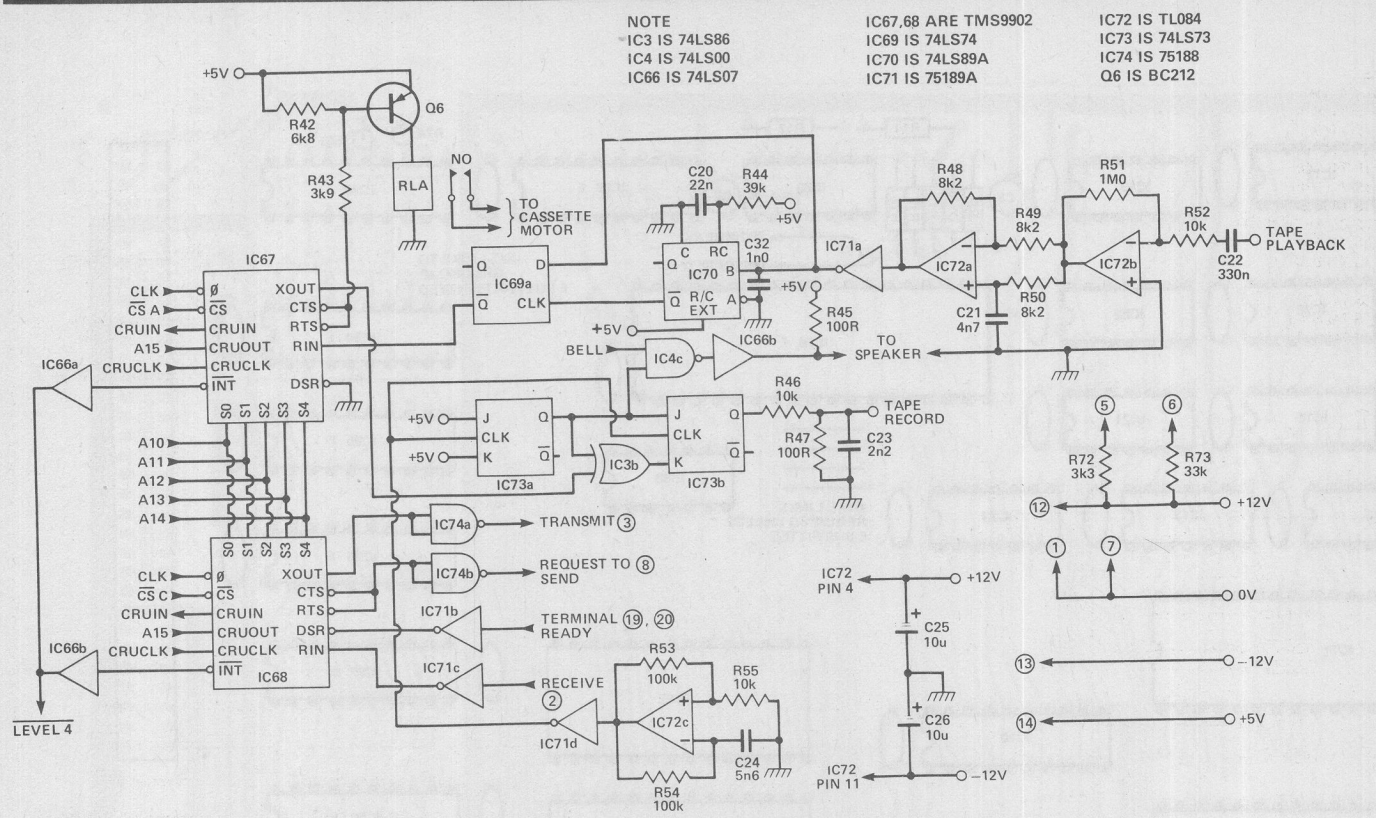


Fig. 2 Circuit diagram of the RS232 and cassette interfaces. Circled numbers are RS232C socket numbers.

HOW IT WORKS — RS232 AND CASSETTE PORT

The RS232 port consists of IC68, a TMS9902 Asynchronous Communications Controller (ACC) and the TTL-to-RS232 signal level shifters (IC74a,b and IC71b,c). IC68 is a completely software-controlled device; its baud rate can be set at anything from 46 baud to over 100,000 baud. The number of bits to be transmitted or received can also be changed, as can the type, the parity and number of stop bits. The CPU drives the ACC through the serial I/O (CRU) bus. The ACC is decoded as a 32-bit block, each bit being selected by the five address lines A10–A14.

The cassette interface uses another ACC, IC67. First a 4.8kHz op-amp oscillator (IC72c) drives a level shifter (IC71d) before being divided by two in the first flip-flop (IC73a). This ensures that the waveform has a unity mark-space ratio. The serial output from IC67 then controls the action of the second flip-flop, IC73b, via the EXOR gate IC3b. When the output is high, IC73b acts as a shift register, passing through the 2.4kHz tone; however, when the ACC output goes low then synchronously at the next clock pulse, IC73b starts to divide by two, hence generating 1.2kHz. The key point here is the synchronous switch from one tone to the other. The signal is high-pass-filtered and attenuated by R46, R47 and C23 before passing to the tape recorder.

On playback the signal is first amplified by a factor of 100 and buffered in IC72b before going through an all-pass filter, IC72c. This is necessary because of the nature of tape recording. When square waves are recorded on tape they are accurately captured; however, on playback frequency equalisation is carried out in the tape recorder but the phase relationship is destroyed, resulting in a 'spiky' sine wave. This is corrected by the linear phase-shift-versus-frequency characteristic of the all-pass filter. Thus the original square wave shape is recovered at the output of IC72a. This is then level-shifted by IC71a and used to trigger a monostable (IC70a). At the end of the monostable period (312.5 uS) the state of the signal is sampled by the D-type flip-flip IC69a. As the half-periods of the two tones lie either side of the monostable period, each tone generates the opposite logic level at the sample point.

HOW IT WORKS — FLOPPY DISC CONTROLLER

The TMS9909 (IC76) is a highly complex micro-controller, designed to work in conjunction with the TMS9911 DMA controller to transfer data from floppy discs. The FDC can control up to four drives which can be a mixture of two sizes or types.

All signals that go to the drives are open-collector buffered by IC80,82,83 and terminated by a resistor pack on the last drive in the chain. The signals from the drives are terminated on the board by a resistor pack and then buffered by IC84.

The raw data pulses from the drive, after being buffered by IC84a, are stretched by a monostable (IC70b) by an amount dependent on the data transfer rate selected by the 'SIZE' I/O bit and the 'DDEN' (double density enable) signal (see Table 1). The output of the monostable is used to control IC77, a digital phase-locked loop. The output of IC77 is, in the unlocked state, half the input clock frequency. When the loop is locked to a signal then the PLL inserts or deletes clock pulses in the pulse stream, thus shifting the average frequency. The programmable divider IC87 and divider IC69b are controlled by the 'SIZE' and 'DDEN' signals to select the correct clock frequency. The raw data is synchronised by IC88 to the PLL clock and then fed to the FDC. The FDC separates the interleaved clock and data bits from the pulse stream and sends data bytes via single byte DMA transfers to main memory.

Mini-floppy ($5\frac{1}{4}$") drives require a motor control signal to start and stop the disc rotating. Upon starting, the disc will not be ready for data transfers for one second while the disc gets up to speed. To reduce the time required to access the disc repeatedly IC79b keeps the motor running for five seconds after it is de-selected and IC79a provides the initial one second 'not ready' signal to the FDC. For standard (8") drives that don't generate a 'ready' signal there is a set of four jumpers.

The BASIC interpreter has a 'BOOT' command which causes the FDC to read the first track from disc 1 and execute it as a machine code program. This could, for example, then search for and load the UCSD interpreter. In order that the system can boot from any type of disc there are two jumpers called 'SIZE' and 'DENSITY' which are read by IC63. This enables the BASIC interpreter to set up the FDC correctly.

NOTE
IC3 IS 74LS86
IC69 IS 74LS74
IC70 IS 74LS89A
IC76 IS TMS9909
IC77 IS 74LS08
IC78 IS 74LS32
IC79 IS LM339
IC80,82,83 ARE 74LS07
IC81 IS 74LS04
IC84 IS 74LS244
IC85 IS 74LS139
IC86 IS 74LS297
IC87 IS 74LS163
IC88 IS 74LS74

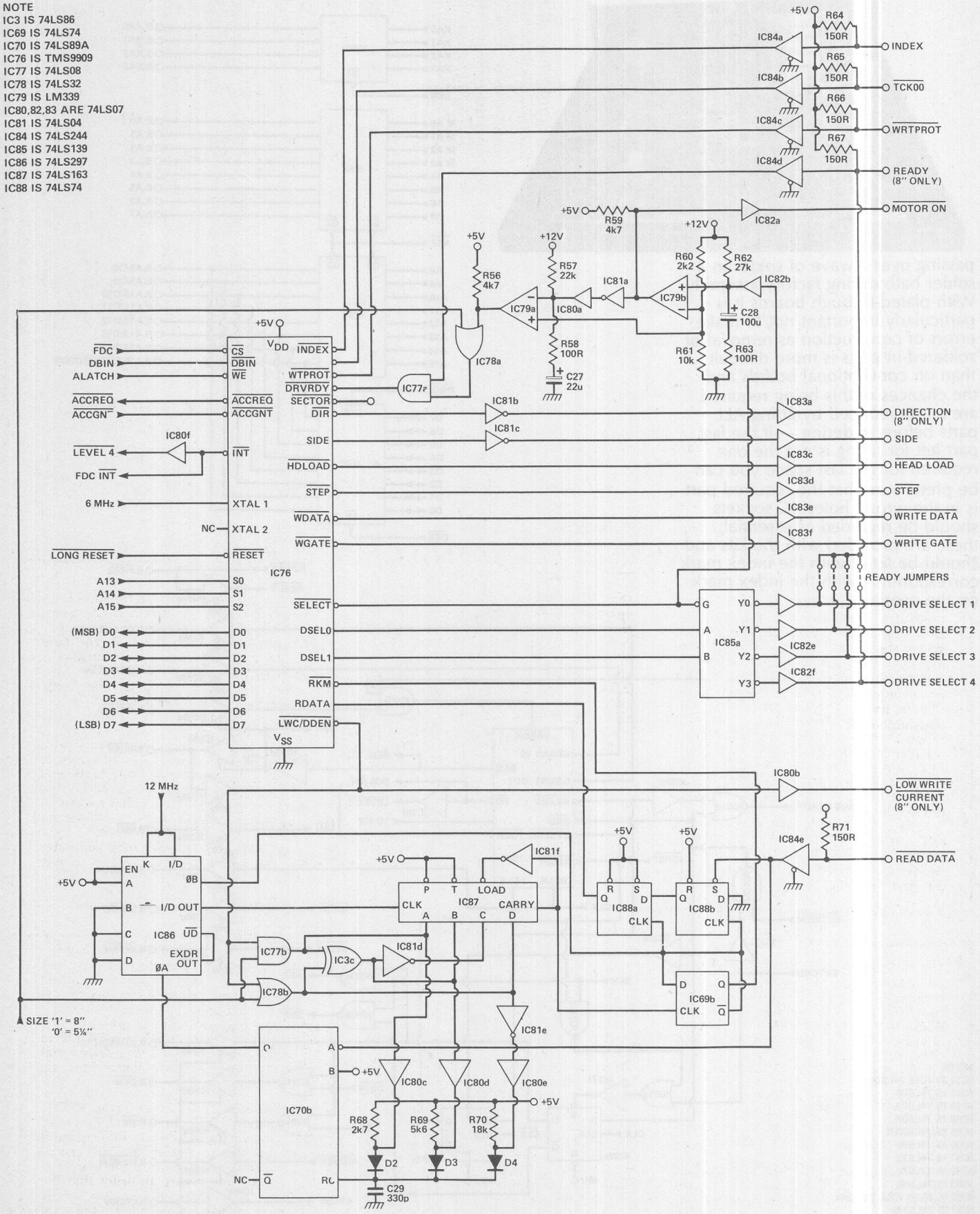
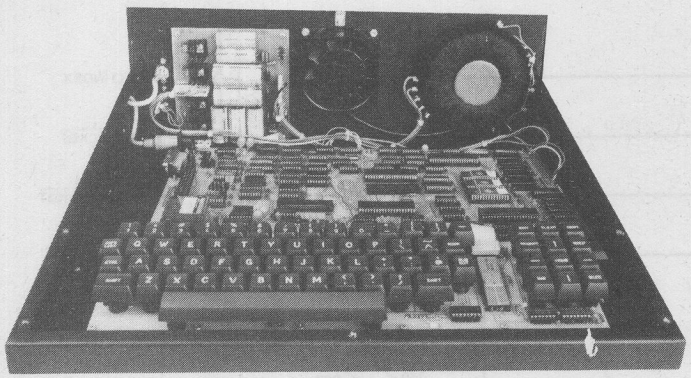
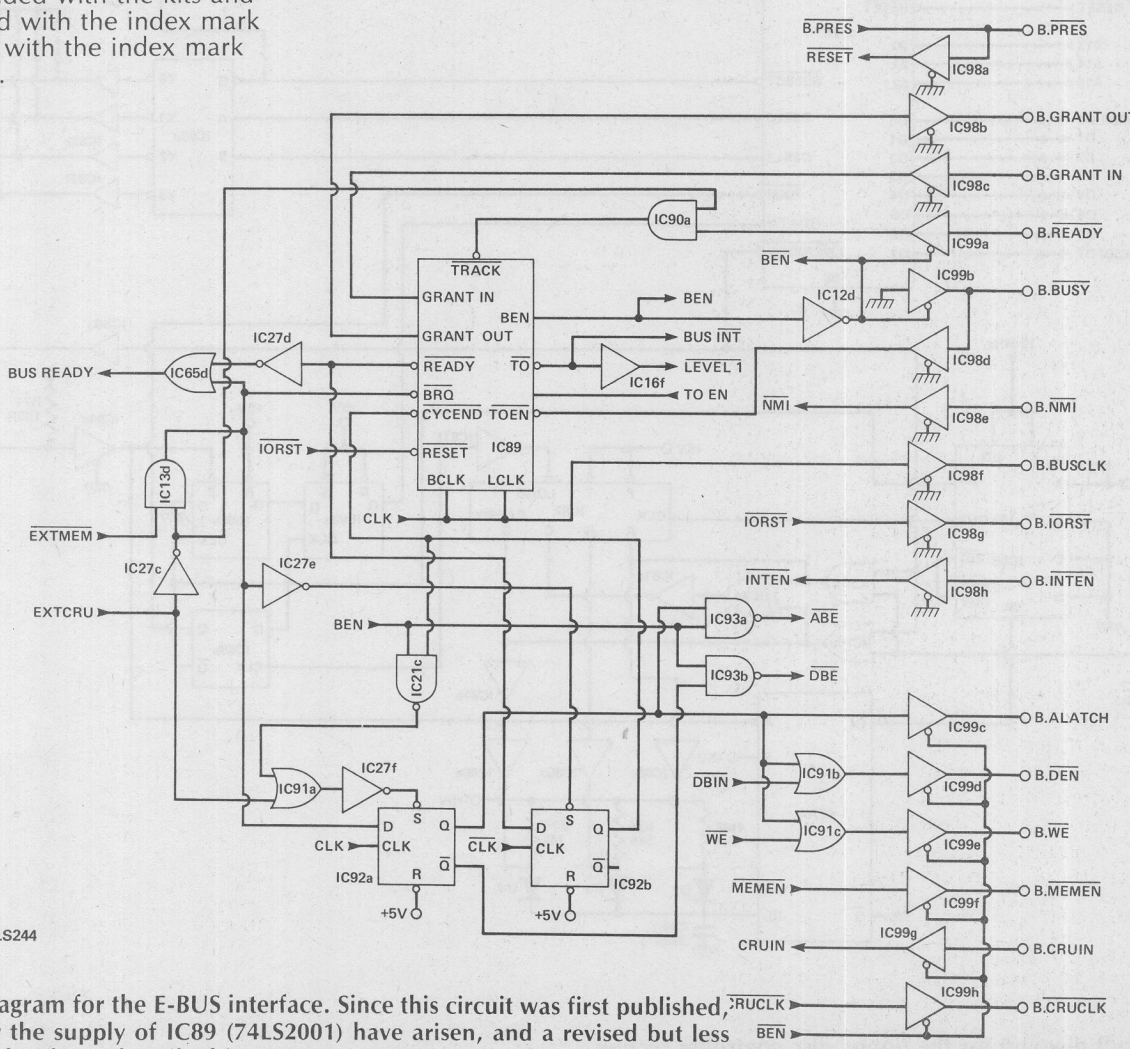


Fig. 3 Circuit diagram for the floppy disc controller section.



passing over a wave of solder in a solder bath during factory assembly. With plated-through boards it is particularly important not to make errors of construction as removal of soldered-in parts is more difficult than on conventional boards and the chances of this being required are much reduced by fitting ALL parts before soldering — if the last part left for fitting is not the one required for the last space you can be pretty sure that the required part is in the wrong holes! IC sockets should be regarded as essential; these are provided with the kits and should be fitted with the index mark corresponding with the index mark on the overlay.



NOTE
IC12,27 ARE 74LS04
IC13 IS 74LS08
IC16 IS 74LS07
IC66 IS 74LS07
IC89 IS 74LS2001
IC90 IS 74LS08
IC91 IS 74LS32
IC92 IS 74LS74
IC93 IS 74LS00
IC94-96,98,99 ARE 74LS244
IC97 IS 74LS245

Fig. 5 Circuit diagram for the E-BUS interface. Since this circuit was first published, difficulties over the supply of IC89 (74LS2001) have arisen, and a revised but less versatile circuit has been described in ETI, June 1984.

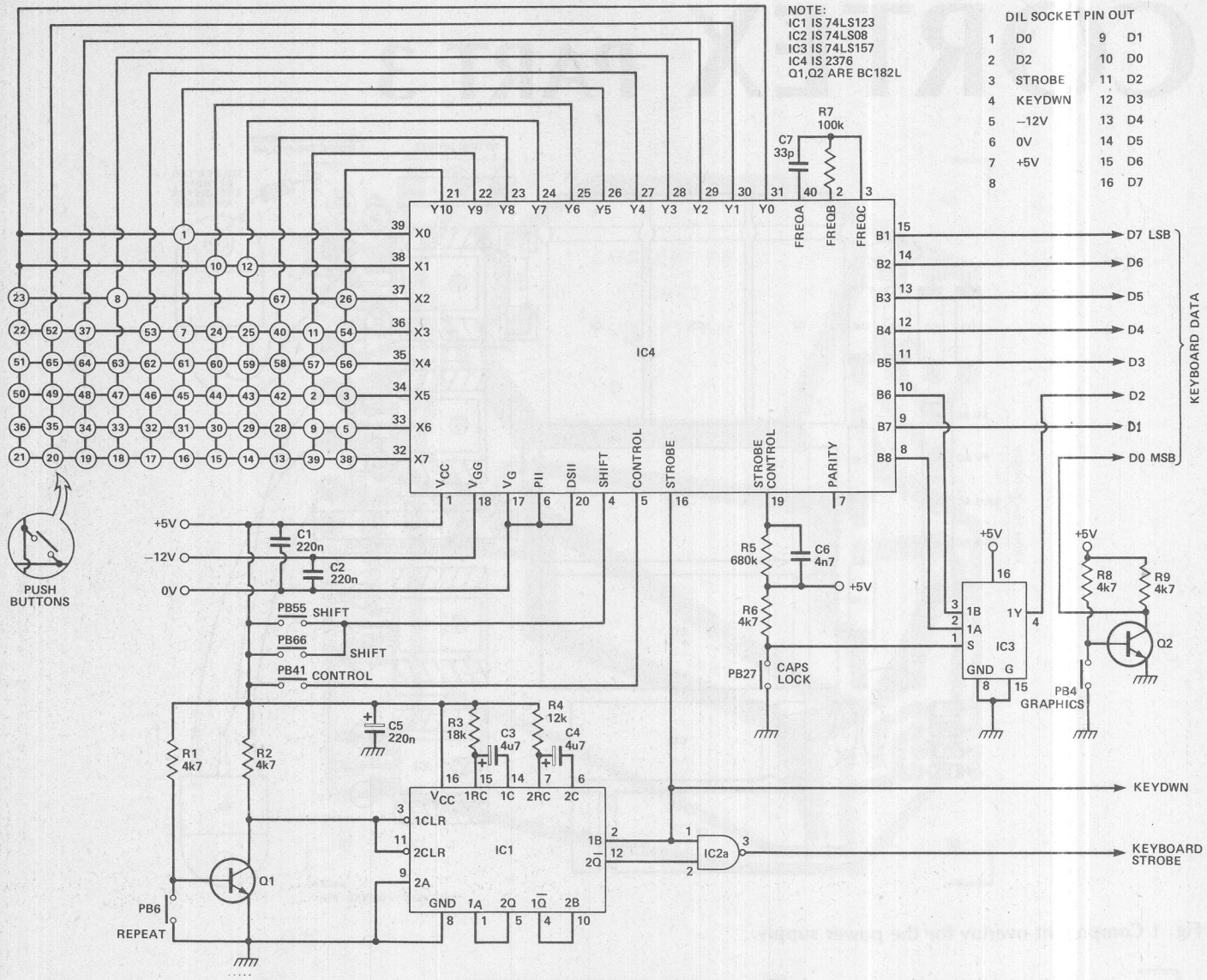


Fig. 6 Circuit of the keyboard; note that the component numbering is separate from all the previous circuitry.

HOW IT WORKS — E-BUS

The E-BUS is a powerful and compact bus which allows many intelligent cards to share a common resource of memory and I/O cards. In order to share out the resources on the bus, each card has a priority according to its position. This is done by passing a signal down the bus which goes into each card as GRANTIN and comes out as GRANTOUT to form the GRANTIN of the next card. A second signal, BUSY, tells each card if the bus is in use or free. If the bus is free and a card requires the bus, it disables the lower priority cards with the GRANTOUT signal and if the GRANTIN signal and BUSY are OK it asserts BUSY and enables its data and address bus buffers.

Once the bus transfers are complete or if a higher priority card requires the bus, then the card will relinquish control. All these events are synchronised by a backplane clock, BUSCLK. Each data transfer that takes place must signal its completion using READY.

The 74LS2001 gate array (IC89) contains the bus arbitration and control logic to gain and release the bus with timeouts upon error conditions. If the card cannot gain control of the bus after 128 clock cycles, it aborts with a timeout interrupt. Also, if after 16 clock cycles the transfer has not been signalled as complete using the READY line, the controller completes and issues a timeout interrupt.

The E-BUS has provision for a multibit interrupt code signalled by the INTEN signal. This interface only provides a single interrupt level using the INTEN signal. The data, address and interrupt signal are multiplexed onto the same pins to conserve connections. The ALATCH signal is used to enable the address latches when the address is on the bus. Then either DEN or WE will be signalled, to show that either a data read or write is occurring and that data is now on the bus. The INTEN signal can be used to latch the interrupt code.

KEYBOARD

The keyboard is a separate unit providing a fully encoded output. Most of the work is carried out by the 2376 keyboard encoder (IC4). This IC contains a 50 kHz oscillator and two ring counters of eight and 11 stages, the outputs of which form an XY matrix across which the switches are connected. By this means each key is sequentially scanned. The closing of one of the switches for a sufficient length of time for switch bounce to be completed causes the scanning to stop; a 'valid' signal now appears on the strobe output. The encoder also contains a 2376-bit ROM (hence the IC name) arranged as three groups of 88 words of nine bits. The shift and control inputs select one of the three groups and the individual word is addressed by the ring counters.

IC3 is a data selector. D2 is either the output B6 or B8 depending on whether upper or lower case characters are selected by the CAPS LOCK switch. Repeated entry of a character is accomplished by multiple strobe signals from IC1, which is a dual monostable arranged as an oscillator and is enabled by a high level on the clear inputs.

CORTEX PART 3

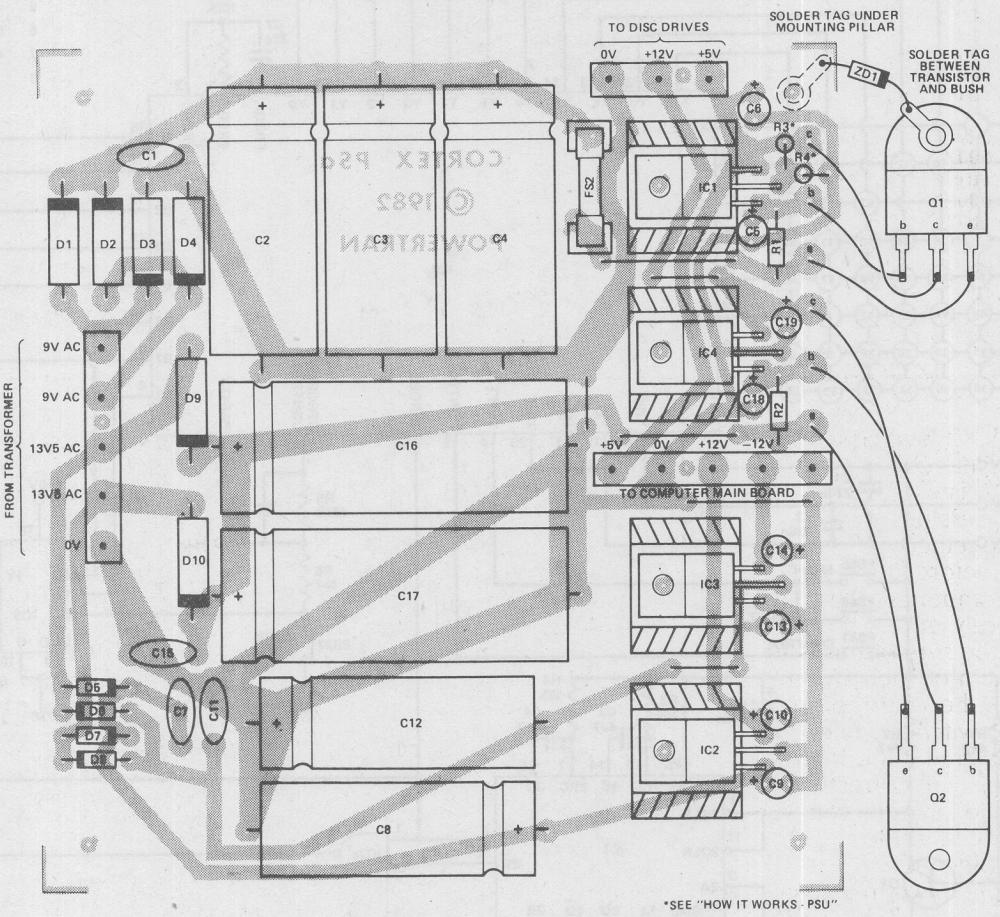


Fig. 1 Component overlay for the power supply.

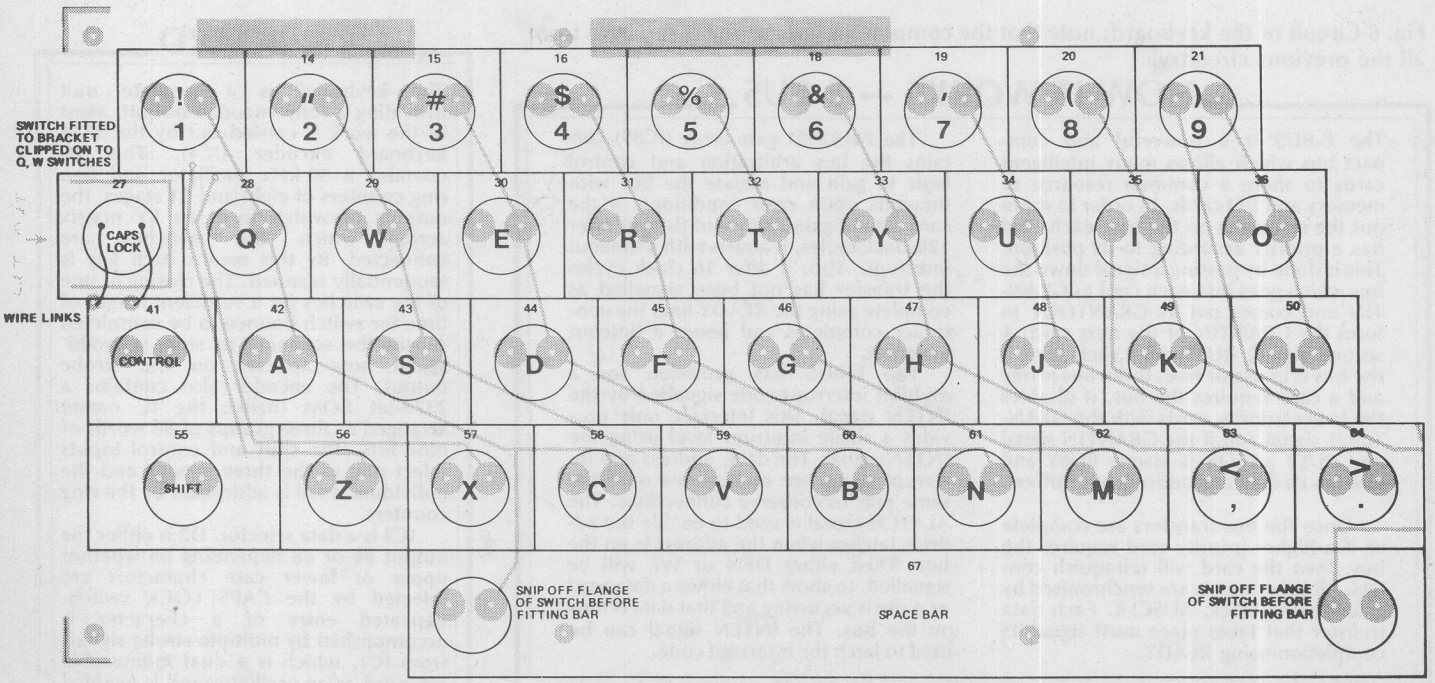


Fig. 2 The keyboard overlay, in two halves so it doesn't get stapled into illegibility.

PARTS LIST

POWER SUPPLY

Resistors (all ¼W, 5%)

R1,2	10R
R3,4	see "How it works — PSU"

Capacitors

C1,7,11,15	100n polyester
C2-4	4700u 16 V axial electrolytic
C5,6,9,10, 13,14,18,19	1u0 35 V tantalum
C8,12	2200u 25 V axial electrolytic
C16,17	4700u 25 V axial electrolytic

Semiconductors

IC1	7805
IC2,4	7812
IC3	7912
Q1,2	TIP2955
D1-4,9,10	1N5402
D5-8	1N4002
ZD1	BCW70 5V6

Miscellaneous

PCB (see Buylines); one off three way connector; two off five way connectors; transformer (13.5-0-13.5 V at 3 A, 9 V at 4 A; fuseholder clips; four off TV5 heatsinks.

KEYBOARD

Resistors (all ¼W, 5%)

R1,2,6,8,9	4k7
R3	18k
R4	12k
R5	680k
R7	100k

Capacitors

C1,2,5	220n 35 V tantalum
C3,4	4u7 16 V tantalum
C6	4n7 ceramic
C7	33p ceramic

Semiconductors

IC1	74LS123
IC2	74LS08
IC3	74LS157
IC4	2376
Q1,2	BC182L

Miscellaneous

PCB (see Buylines); 67 off momentary push-to-make switches; one off latching push-to-make switch; set of 67 double shot moulded key tops; IC sockets; switch mounting bracket; mounting pillars etc.

The CAPS LOCK switch is physically different from the rest and is wired in with wire links. Press this switch into its bracket together with the Q and W switches before fitting to the board. It is most important that the switches fit squarely on the board. The best way to be sure of this is to solder only one pin of each switch and then holding the board, press in turn each of the switches while reheating the soldered joint. If any are misaligned this will correctly position them. The key tops can now be pressed on and the remaining pins soldered.

The power supply is on a single-sided PCB with six wire links and it is best to fit these before any components. Connections to and from this board are made via connectors and to ensure that their pins are soldered in squarely, fit the sockets on to them during soldering. The power supply is all on the back panel and the power transistors use this as a heatsink, being fitted to it with mica insulating washers.

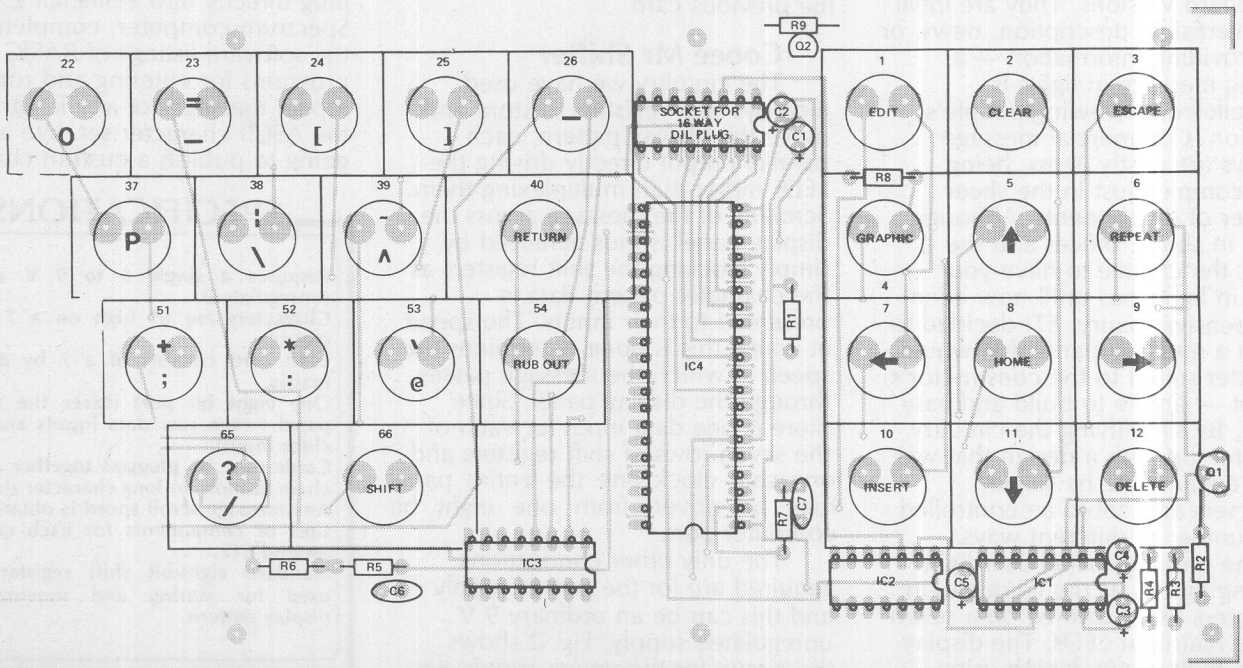
As well as holding the input and output sockets, the rear panel has provision for a cooling fan and one should be fitted when disc drives are used.

The disc drives pass through the front panel and are screwed onto a mounting plate. Plates on the sides of the drives press against the panel, thereby making a rigid sub-assembly which fits into the cover of the computer. The standard kit has a panel with no cut-outs for disc drives and a new panel is provided with the drives when purchased.

There are two positions in which the main board can be fitted. The board has provision for a Eurocard connector for expansion purposes and there is a cut-out in the side of the computer through which the connector passes; for external expansion the board fits at the far right hand side. However, if the add-on units are to be fitted internally then the position to the left is used.

BUYLINES

Powertran are supplying complete kits of parts and component packs for the Cortex. A complete 64K Cortex kit will cost £295 plus VAT, carriage free. A ready-built 64K Cortex will cost £395 plus VAT, carriage free. Prices for add-ons (eg floppy discs, RS232C interface, memory expansion etc) and for component packs (eg PCB, semiconductors etc) can be found in Powertran's brochure. Powertran Cybernetics, Portway Industrial Estate, Andover, Hants SP10 2NM. Telephone 0264 64455.



ETI