Technical Manual

Colour-GENIE

LOWE
COMPUTERS

Foreword.

This technical manual is intended as a guide to programming, interfacing and servicing the Colour Genie computer. The layout is in sections for this purpose. Section 3, the block diagram, is intended as an introduction to the structure of the machine, and is recommended reading. Section 4 then continues with a detailed look at the circuit, and section 5 covers aspects of high and low level programming.

The information in this manual relates to the current version of the Colour Genie, with circuit board revision D-1 and ROM software sourced from Germany (ie New ROM's). Earlier versions of the computer PCB contain largely the same circuit, but minor changes in component layout will be noticed in some areas. Earlier versions of ROM software (which can be updated) have different CALL's to a few routines. CALL locations listed in this manual that are relevant only to New ROM's are indicated where possible.

Although circuit operation is described, the manual does not give comprehensive data on all the integrated circuits used in the Colour Genie. Reference to a data book on 74 series TTL logic circuits is suggested if further information is required. Similarly additional Z 80 programming information should be sought from one of the many good books on this subject.

I hope that you will find the technical manual interesting and informative, and that any mistakes that may have crept in do not cause too many problems. The technical staff at Lowe Computers will be happy to try and answer queries that arise from, or are not answered by, this manual.

John Thorpe. Jan 1984.

Text Copyright (c) 1984 by
Lowe Electronics.
Circuits reproduced by kind
permission of T.C.S. GmbH.

Lowe Electronics Ltd.,
Bentley Bridge,
Chesterfield Road,
Matlock, Derbyshire. DE4 5LE
Tel 0629 ~~4995~~ 580800 . Telex 377482

Contents.

Section.		Page.
1.	Technical specification.	3
2.	Memory and port map.	6
3.	Block diagram.	7
4.	Detailed circuit information.	
4.1	CPU and system timing.	9
4.2	Address decoding and keyboard.	12
4.3	Video display.	13
4.4	System RAM.	16
4.5	PAL colour encoder.	17
4.6	Serial, parallel, sound & cassette interfaces.	19
4.7	Expansion port.	21
4.8	Power supply.	23
5.	Detailed programming information.	
5.1	BASIC memory utilization, pointers etc.	24
5.2	Programming hint and tips.	27
5.3	Keyboard.	31
5.4	Screen and graphics.	32
5.5	Cassette.	36
5.6	PSG and parallel port.	38
5.7	Serial port.	40
5.8	Summary of ROM calls and dedicated RAM locations.	45
6.	List of components.	48
7.	Circuit diagrams and layouts.	
7.1	CPU and address decoding.	52
7.2	System timing.	54
7.3	CRTC and system RAM.	56
7.4	Character generator and colour RAM.	58
7.5	PSG, serial and cassette interfaces.	60
7.6	PAL colour encoder.	61
7.7	ROM board.	62
7.8	Power supply.	63
7.9	Keyboard.	64
8.	Accessories.	
8.1	EG 2011 Additional 16k RAM card.	65
8.2	EG 2012 Parallel printer interface.	65
8.3	EG 2013 Joysticks.	66
8.4	RGB output circuit.	67
8.5	Circuits and layouts.	68
9.	I/O connections.	71

Section 1. Specification.

Processor

Z 80 CPU running at 2.2 MHz with one wait cycle introduced during RAM access to avoid bus contention with the video display.

Memory space

16K of extended Microsoft BASIC is held in four 2532 type EPROMS located at addresses 0000H to 3FFFH. 32K of RAM is provided between 4000H and BFFFH, the lower 16K is on the main circuit board, the remaining 16K is on a plug-in card. This RAM is executed with 16K-bit dynamic chips, type 4116.
The lower 16K of RAM is shared between the processor and the video display controller, and memory space for the text display and the bit-mapped display is reserved in this region. The approximate allocation of RAM space is shown below:-

Use.	Number of reserved bytes. (approx)
BASIC workspace	1350
Text display memory	1000
Graphic display memory	4100 (Optionally 0)
Free space	26300 (Optionally 30400)

12K of address space from C000H to EFFFH is available for use in an external ROM cartridge. The remaining 4K of address space is reserved for device interfaces, text colour memory and programmable character memory.

Video display

The computer may be switched to one of two modes:-
LGR - Normal text display and
FGR - Bit-mapped high resoloution graphic display.
The video display is controlled by a 6845 CRTC chip, which is re-programmed by ROM routines to operate in the required mode.

LGR mode

Text is diplayed in 25 rows of 40 columns, each character comprises 8 by 8 pixels, and can be set to any one of 16 colours against a black background.
256 different characters can be displayed simultaneously, with character sets selected from 128 alpha-numeric characters, 128 graphic characters and 128 user-definable characters (read from RAM).

FGR mode

Graphic resolution is 160 by 102 pixels, each of which may be set to any one of 4 colours. Two alternative colour schemes are available, but may not be used simultaneously. Each pixel is represented by two bits in the graphic memory, hence the requirement of 4K of memory and the availabitity of 4 colours.

Keyboard

The keyboard provides the normal alpha-numeric keys with the following additions:-

Four function keys that are programmed with 8 common BASIC commands, but may be re-defined by the user.

A mode select key that switches key characters between alphabetic symbols and graphic symbols.
A pair of reset keys that produce a processor interrupt (NMI) if pressed simultaneously.
A break key to halt execution of BASIC programs.

All keyboard switches, with the exception of the reset keys, are connected in an 8 by 8 matrix, and mapped into a 256 byte area of address space. Firmware in the ROM scans the keyboard matrix and converts the resulting code to ASCII.

Sound

A comprehensive sound generating facility is provided by a PSG chip, type AY-3-8910. This contains 3 independent tone generators, a noise generator and an envelope shaper. The sound output is modulated onto the television output and is also available via a separate audio output.

Interfaces

T.V. UHF Channel 36, 625 lines, PAL colour with sound on a 6 MHz subcarrier. Connection is via a flying lead, terminated with a co-axial plug.

Video 1 V pk-pk composite video signal, including PAL colour information. Output impedance is 75 ohms and connection is via a phono socket.

Audio The sound signal is available at approx 250 mV via a phono socket, for use with an external amplifier.

Cassette An interface for a standard cassette recorder is provided for storing and retrieving programs and data. This interface operates at 1200 baud, transferring about 150 bytes per second. The output level is 100 mV and is suitable for feeding into the microphone input of a cassette recorder. The computer requires a playback signal of about 3 V pk-pk, and can be driven from the cassette recorder earphone output. Connection is via a 5-pin DIN socket, and a lead to two 3.5 mm jack plugs is provided.

Light Pen A light pen input provides a connection to the light pen strobe pin of the CRT controller chip. A +5V supply, and connections for an operator push-button are also provided via a 5-pin DIN socket.

Serial A serial interface that operates at V24 standard voltage levels (+12V / -12V) is provided for communication and connection to serial printers. Three lines are provided; Transmit data, Receive data and Carrier Detect. The CD line is used as a handshake input to prevent printer buffer overrun. The serial data format and baud rate are controlled by the port driving software, which is not resident in ROM, but must be loaded if required. Connection to the serial port is via a 5-pin DIN socket.

Parallel

Two 8-bit parallel ports, A and B, are provided that operate at TTL levels. Either port may be configured to be input or output. The interface uses the PSG chip, which supports the two ports as registers 14 and 15. All communication to the parallel ports is done via this chip. Please note that:-

The parallel printer and joysticks are operated through this interface.
In output mode, the data written remains latched on the output lines.
The default condition is for port A to be set to output and port B to be set to input.
No strobe, busy or acknowledge lines are present.

Connection to the parallel port is provided on a 20-way, 0.1 inch pitch indirect edge connector. Supplies at +5V, +12V and -12V are also provided.

Expansion

A 50-way double sided PCB edge socket provides signals from the Z 80 bus for external system expansion and for the connection of ROM cartridges.

Miscellaneous

Height	85 mm.
Depth	280 mm.
Length	443 mm.
Weight	Approx 4 kg
Power	220 / 240 V at 50 Hz.

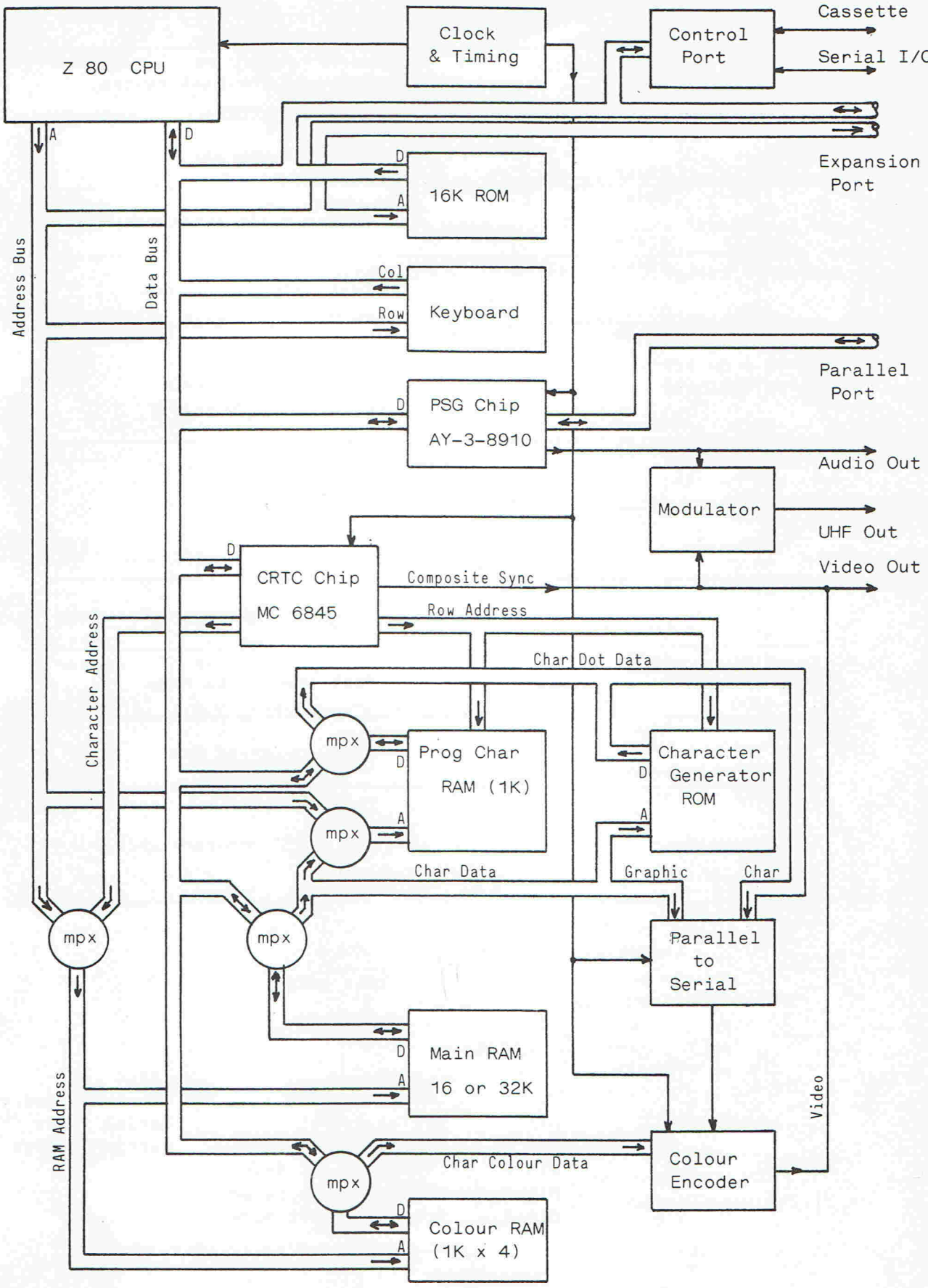
Section 2. Memory map.

Hex address	Dec value	PEEK/POKE value	Memory function
FFFF	65535	-1	
to			Reserved for disk system.
FC00	64512	-1024	
FBFF	64511	-1025	
to			Keyboard matrix.
F800	63488	-2048	
F7FF	63487	-2049	
to			Programmable character RAM.
F400	62464	-3072	
F3FF	62463	-3073	
to			Text mode colour RAM.
F000	61440	-4096	(Low order 4 bits only)
EFFF	61439	-4095	
to			ROM cartridge space.
C000	49152	-16384	
BFFF	49151	-16385	
to			
8000	32768	-32768	User RAM.
7FFF	32767	32767	
to			
5800	22528	22528	
57FF	22527	22527	
to			High resolution graphic RAM.
4800	18432	18432	
47FF	18431	18431	
to			Text mode video RAM.
4400	17408	17408	
43FF	17407	17407	
to			BASIC reserved RAM.
4000	16384	16384	
3FFF	16383	16383	
to			16K BASIC interpreter.
0000	0	0	

Ports.

Hex	Function		
F8	PSG register number (output only)		
F9	PSG register data		
FA	CRTC register number (output only)		
FB	CRTC register data		
FF	Control port:-	Output	Input
	Bit 0	Cassette write	Cassette read
	Bit 1	Serial output	Serial input
	Bit 2	Graphic background	Carrier Detect
	Bit 3	Character set select	- -
	Bit 4	Character set select	- -
	Bit 5	Graphic / text mode	- -

Section 3. Block diagram.



Section 3. Block diagram.

This block diagram outlines the major functional blocks comprising the Colour Genie computer. There follows an outline of operation and a more detailed description can be found in the next section.

The Z 80 CPU provides most of the bus timing and control signals, and its outputs are buffered by TTL gates to provide the main address and data busses that run through the machine. Connected directly to these busses, via address decoders and buffers, are the 16K BASIC ROM and the keyboard matrix. The busses are also extended to the expansion port. The two LSI chips, the sound generator (PSG) and the video controller (CRTC), both communicate with the processor via the data bus using the Z 80's I/O ports, as does the control port register, which determines graphic modes and character set selection in addition to providing the serial and cassette interfaces.

The main RAM in the system is not attatched directly to the bus, but instead a system of data and address multiplexers allows access to the RAM by either the CPU or the video display circuitry. Most important video signals are generated by the CRTC chip, principally:-

(1) the address in memory of the character currently being displayed on the screen,

(2) the row number in the character that is being displayed. This is sent to the character generator so it can produce the appropriate dot pattern,

and (3) the horizontal and vertical synchronization signals which are sent to the video mixing stage.

The character generator is best thought of as a section of read-only-memory, where addresses are determined by combining the code of the character being displayed with the row number from the CRTC. Data from the main RAM provides the character code, and this is fed, via a multiplexor, to the character data bus. Multiplexing is required to permit shared access to the RAM by the processor. The character generator ROM produces an 8-bit output representing the dot-pattern of the row of the character to be displayed. This information is presented to the parallel-to-serial converter, which outputs the dots one after another, in serial fashion, as required by a video display.

For high resolution graphic displays the character generator is not required, and the character data is fed directly into the parallel-to-serial converter as dot patterns.

The character address bus from the CRTC is also fed to a small amount of RAM which stores colour information for the characters that are displayed. This memory and the main memory are read simultaneously, and while the character data addresses the character generator, the colour data controls the PAL colour encoder. The colour data also is fed through a multiplexing circuit to allow the RAM to be read and written by the processor.

The remaining memory is the programmable graphic RAM. This can be connected in place of the character generator ROM, and allows character dot patterns to be programmed. Again multiplexing circuits allow access by both character data and processor information.

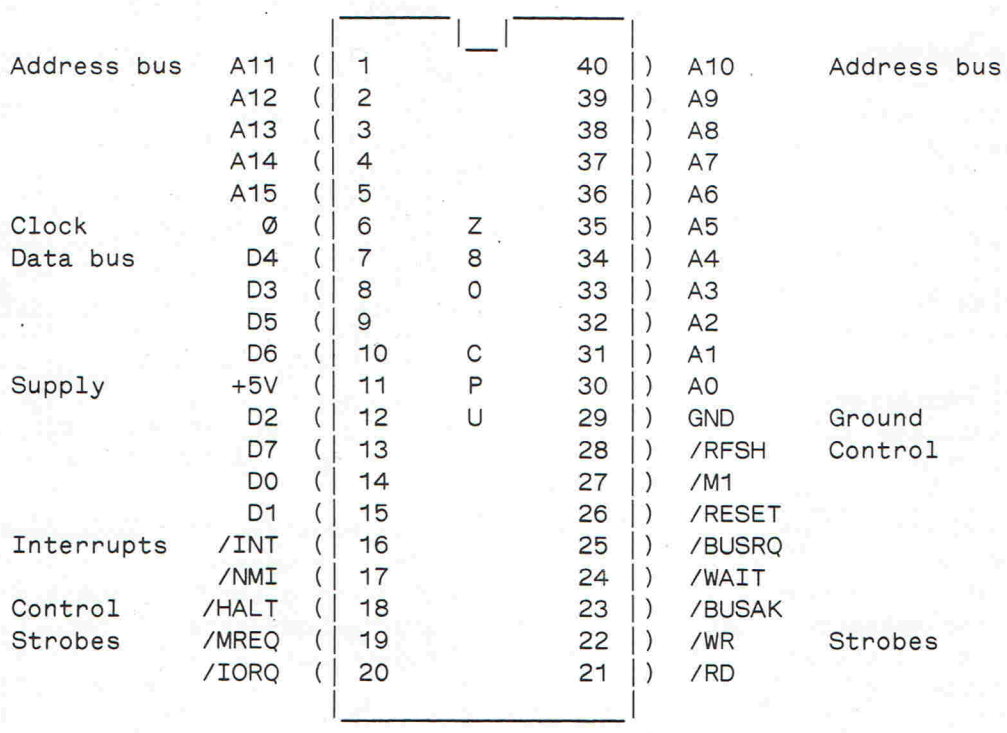
4.1 CPU and system timing circuits. (See diagrams on pages 52 and 54)

All system timing is derived from a single crystal oscillator, running at 17.734 MHz. This is chosen as being 4 times the PAL subcarrier frequency so that 4 signals in quadrature phase can be generated for the PAL encoder. The CPU clock is produced by dividing the oscillator frequency by 8, and a 1.1 MHz clock for the CRT controller (CCLK) by a division of 16. Clock division is done by a 4-bit counter, Z65.

RAM access cycles are initiated every cpu clock cycle by Z64 clocking Z10. Alternate cycles deal with processor and video access, and signal 'A' determines whether a cycle takes place - if either the CCLK line is high (video cycle) or if the CPU does a read or write to the RAM or the colour RAM. The CPU access is latched in Z5 (top half), with the other half of Z5 providing the WAIT signal, extending the Z80's memory cycle until the start of the next video cycle.

Once a RAM cycle is in progress, the RAS strobe is made active by Z10 (left half), and a MUX signal is produced by delaying this by about 70ns with the other half of Z10. The CAS signal to the RAM chips is derived from the MUX signal and the address decoding.

Pin connections and signal details of the Z80 CPU are shown below:-



Data bus D0-D7 CPU data bus (bi-directional).

Address bus A0-A15 CPU address bus (output).

Interrupts	/INT	Processor interrupt. A low level on this input causes the CPU to be interrupted if interrupts are enabled (by the program). The subsequent events depend on the interrupt mode that the processor is in, but generally a restart is executed from location 0038H.
	/NMI	Non-maskable interrupt (used by the keyboard reset keys). Similar in operation to the /INT input, but the non-maskable interrupt cannot be disabled, and a restart from location 0066H is executed.
	/RESET	Power-on reset. CPU starts executing instructions from location 0000H when this input is taken low then high.
Strobes	/MREQ	This output goes low when the processor is reading from or writing to the memory.
	/IORQ	The input/output request is taken low when the CPU executes an IN or OUT instruction, indicating that a port number is present on the bus, not a memory address.
	/RD	The read strobe output goes low when the CPU requires to read data from memory or a port.
	/WR	The write strobe output goes low when the CPU writes to memory or outputs data to a port.
Clock	Ø	The processor clock input at 2.2168 MHz.
Control	/HALT	When the CPU executes a HALT instruction, program execution is suspended and this output is taken low. The HALT instruction is not normally used in the Colour Genie, since execution can only be resumed by an interrupt or a reset. During a halt, the processor continues producing memory refresh cycles.
	/BUSRQ	A low level on this bus request input forces the processor into an idle state, with address and data buses in a high impedance state. This allows a device other than the CPU to control the bus in the computer.
	/BUSAK	This output is taken low when the processor is in idle state, and the address and data busses can be controlled externally.
	/WAIT	The wait input to the CPU suspends processor activity immediately it is taken to a low level. This is used in the Colour Genie to extend RAM memory cycles so that no conflict exists between the CPU and the video controller when sharing memory. The wait input should not be held low for more than a few clock cycles, since the Z80 ceases to refresh memory during a wait.
	/M1	The machine cycle one output goes low when the CPU is reading an op-code from the memory.
	/RFSH	This output is taken low along with the /MREQ output during a refresh cycle, when a 7-bit refresh address is placed on A0 - A6 by the processor.

Section 4. Circuit description.

Most of the bus and control signals from the Z80 proceed directly around the computer and to the expansion port. The exceptions to this are the read and write strobes, which are first buffered by Z52, and the data bus which is buffered by Z35. The data bus buffer direction is normally to write from the processor to the bus, but this is reversed by Z48 when either the /M1 or read strobes are active. The buffers are disabled by Z48 and Z41 when the ROM's are addressed, or when the PSG or CRTC registers are accessed, since these devices are connected directly to the CPU data bus and not to the buffered data bus.

A power-on reset is generated by holding the reset pin on the Z80 low whilst the capacitor C41 charges through resistor R28. The reset signal is squared by Z54 and Z34. Z54 also combines internal and external signals for reset and NMI.

4.2 Address decoding and keyboard. (See diagram on page 54)

The address decoding circuits can be split into two sections dealing with memory addresses and I/O ports. The port decoding is handled by Z40 and Z41 - using the 74LS138 as a one-of-eight decoder which is enabled when /IORQ is low and A3 to A7 are high, thus selecting ports F8H to FFH.

The majority of memory space decoding is handled by Z55 and Z56, each a dual one-of-four decoder. The top half of Z56 decodes A14, A15 and /MREQ to give four signals each representing 16K byte sections of memory. The section starting at addresses 4000H and 8000H are designated /RAM1 and /RAM2 and are used to enable the CAS strobes to the memory. The other two signals are decoded further.

The top half of Z55 decodes two 8K byte sections of memory for the ROM's, producing the signals /ROM1 and /ROM2. This chip also prevents a memory access during a refresh cycle, and does not decode the ROM's if the external ROM disable input is held low.

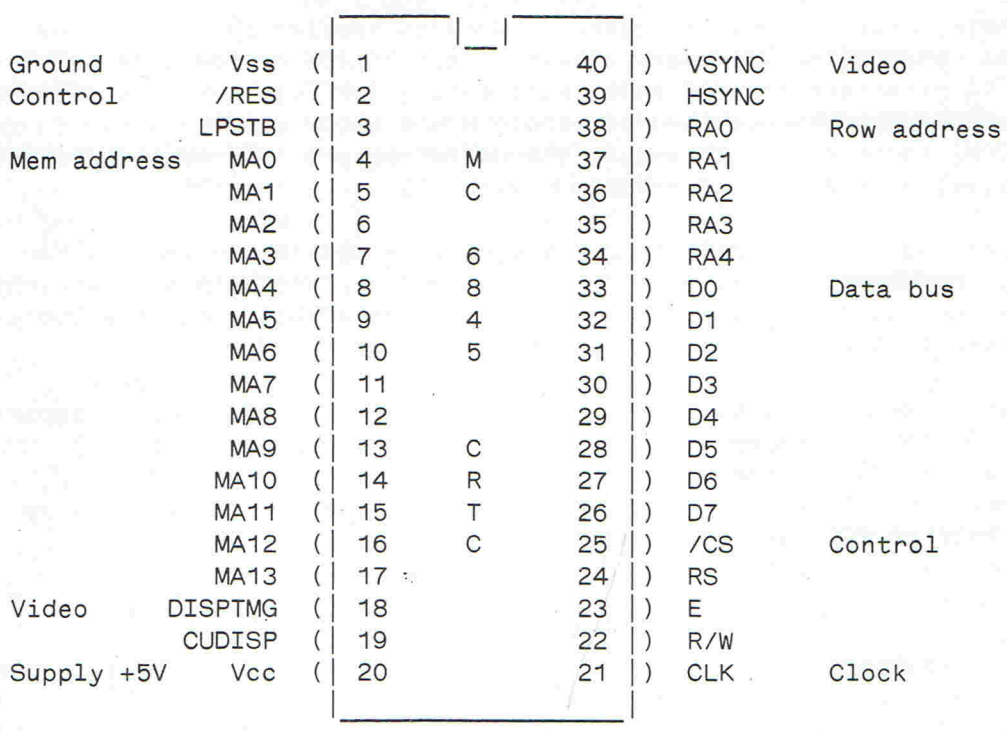
Addresses above C000H enable the lower half of Z56, which separates the top 16K of memory into four 4K blocks. Those addressed at C000H, D000H and E000H are the cartridge ROM enable lines /C1, /C2 and /C3. The top-most 4K block of memory is further split into 1K sections by the lower half of Z55, producing signals to enable the colour RAM (/CCM), the programmable character generator (/PCG), the keyboard (/KYBD) and the disk controller space (/C4).

The keyboard switches are arranged in an 8 by 8 matrix between address and data lines. The keyboard address lines AK0 to AK7 are driven through inverting buffers Z33 and Z34 from the address bus and through diodes D1 to D8 which are included to prevent output contention if several keys are pressed simultaneously. Keyboard data lines DK0 to DK7 are buffered back through inverting buffers Z24 and Z34 to the data bus when the keyboard address strobe (/KYBD) is active. There are pull-up resistors for each of these data lines (on the keyboard PCB) which ensure a high logic level on the data line unless it is pulled low through a key switch to a low address line. Keyboard operation relies on the driving software setting each address line A0 to A7 high in turn (hence AK0 to AK7 are set low in turn) and reading the data then presented. When this is done a one in a bit position represents a pressed key and a zero represents a key that is not pressed. In eight read operations the whole keyboard is scanned.

4.3 Video display circuits.

(See diagrams on pages 56 and 58)

Pin connections and signal descriptions for the MC6845 CRT controller chip are shown below:-



Data bus	D0-D7	Bi-directional data input/output lines connected to the processor data bus for loading and reading internal CRTC registers.
Mem address	MA0-MA13	Address output bus to the video memory, holding the address of the character to be displayed.
Row address	RA0-RA4	The row address outputs give the position of the current scan line in the row of characters being displayed. This number is fed to the character generator.
Clock	CLK	The character clock input is used to derive all the display timing within the CRTC. One clock cycle is the time between each displayed character.
Control	/RES	A low level on the reset input causes all counters in the CRTC to be set to zero, and display operation ceases until the input is taken high.
	LPSTB	A positive pulse on the light pen strobe causes the light pen register in the CRTC to be loaded with the address currently being sent to the memory on MA0-MA13.
	R/W	The read/write input controls the direction of data

flow on the bi-directional data bus. A high level indicates that data is being read from the CRTC to the CPU, a low level indicates that data is being written to the CRTC.

E — The enable input is used as a strobe for the data lines during processor communication.

/CS — The chip select input must be low to enable processor communication.

RS — The register select input determines whether a register address or register data is interpreted from the data lines. A low level selects a register address when a write operation is performed and a high level allows data transfer to and from the selected register. This signal is connected to A0 of the processor address bus.

Video

DISPTMG — The display timing output is high when the required picture area is currently being diplayed and low during the time when the 'border' is displayed. This is used as a video blanking signal.

CUDISP — The cursor display output provides the cursor video signal which is mixed with the video signal derived from the parallel-to-serial converter, before being fed to the display. The type of cursor is defined in the cursor control register.

HSYNC — The synchronization outputs produce horizontal and VSYNC vertical sync signals which are mixed with the video signal. A horizontal sync pulse occurs before the start of every scan line, and a vertical sync pulse occurs between frames.

Video memory in the Colour Genie is shared between the CPU and the video circuits, and a considerable number of the circuit elements are concerned with routing and selecting memory data and addresses.

Address selection is performed by four 2-to-1 line data selectors, Z42, Z49, Z59 and Z58. These select 14 address lines from either the CPU or the CRT controller to feed to the RAM, depending on the state of the CCLK clock signal. A low level on CCLK selects CPU addresses, whilst a high level selects character addresses. The addressing range of the CRTC is limited to 16K bytes by its 14-bit address bus, this range being between locations 4000H and 7FFFH since the RAM1 block is always selected during CRTC address cycles via Z9 (pins 4,5,6). The bottom-most section of selector Z42 ensures that only memory read cycles can occur during CRTC access.

Data routing is somewhat more simple, the video data being latched by Z8 at the end of a CRTC read cycle. Note however that this latch has tri-state outputs, and is disabled during a programmable character generator program operation.

The colour RAM is read in parallel with the main RAM, and for this reason Z39 is fed from the outputs of the address selectors. The colour data is latched by Z18, which is configured as two cascaded latches, four

bits wide, to produce the colour control signals CC0 to CC3. The cascaded latches delay the data by one clock cycle, compensating for the cycle taken by the character data to pass through the character generator, so that character colours and bit-patterns are correctly aligned when they are displayed. Colour data written from the CPU passes through Z29 to the RAM chip Z39, Z4 (two parts) producing the buffer enable and the write signal. When the CPU reads colour information, data is latched by Z17, which has tri-state outputs feeding directly onto the main data bus.

Character data from Z8 is used to address the character generator ROM and RAM, and also to generate graphic bit-patterns via two shift registers Z37 and Z38. These each accept four bits of the character data, and produce a two-bit signal, FC1 and FC2, defining one of the four graphic colours to be displayed. Character data bits 6 and 7 are processed with signals 'B' an 'C' from the control port (in Z44 and Z51) to select either the ROM or RAM character generators. A row address from the CRTC chip is fed to both generators as A0, A1 and A2 and the resulting bit-pattern data is fed into shift register Z28 to produce the video data /SROUT1.

Both character and graphic shift registers use the same timing signals; a shift/load strobe (S/L) occurs at the beginning of each character slot, loading new character data into the register; and a dot clock (QA) moves bit-patterns out of the shift register at eight times the character rate. Character or graphic mode is selected by the CTRG and /CTRG signals which enable the appropriate shift register.

The programmable character generator RAM can be programmed by the CPU, and thus needs to have access to address and data from the processor. The processor address is fed in two sections; A3 to A9 replaces the character data via tri-state buffers Z36 and Z29; and A0 to A2 is selected in place of the row address RA0 to RA2 by Z19. Data is transferred to or from the RAM via bi-directional buffer Z26.

The video bit-pattern is masked by the blanking signal from the CRTC (DISPL) in Z11 (pins 1,2,3) and mixed with the cursor in Z64 (lower part). The dot pattern is the fed into a pulse-stretching circuit (D19, R29, C58) to make single dots large enough to display on a TV, and to prevent spurious colour outbreaks. The other part of Z64 produces a video signal from the graphic signals FC1 and FC2 and the graphic background FGS, which is blanked by DISPL in Z32 before being presented as the VIDEO signal to the PAL encoder. Z1 serves to combine graphic and text colour information into four signals C0' to C3' which are fed to the encoder.

4.4 System RAM.

(See diagram on page 56)

The random access memory in the system consists of 16K or 32K of dynamic memory constructed from 16K by 1 bit chips, type 4116. These devices use multiplexed address connections, whereby a 14-bit address is reduced to 7-bit row and column addresses. These two addresses are latched into the RAM by strobe signals /RAS (row address) and /CAS (column address). When both of these signals are active (low) the chip is enabled to read or write, depending on the state of the /WR input.

Since the RAM's are dynamic they require refreshing periodically to maintain data storage. The refresh requires that all possible row addresses are presented to the chip at least every 2ms, but a read or write operation is not necessary. The Z80 generates refresh addresses following each op-code fetch, so no refresh timing or counting circuits are necessary, but a RAS-only memory cycle is generated whenever the processor's RFSH output is active.

Because the RAS strobe is needed for refreshing, the CAS strobe is used as the chip-select for the upper or lower 16K of memory - the CAS signal is only made active on the set of chips that is requested, the remaining set of chips receives a RAS-only cycle and remains unselected although refreshed.

The sequence of memory signals to access the RAM is as follows:-

(1) The row address (A0-A7) is presented to the RAM.
(2) The row address strobe goes active (low).
(3) The column address (A8-A13) is presented to the RAM.
(4) The column address strobe goes active (low).
(5) If the /WR input is low, data is written to the RAM.
Otherwise data from the RAM is read from Dout.
(6) The RAS and CAS lines go inactive (high).

Two data selector chips, Z43 and Z50, are used to multiplex row and column addresses to the RAM's. Z43 also derives the CAS signal, which is gated with RAM1 by Z9 to produce the /CAS to the lower 16K bytes of memory. The CAS to the upper 16K is similarly gated with RAM2 by Z32 and Z6. Note that an access by the video circuit will always activate the lower 16K (the /CCLK signal is low).

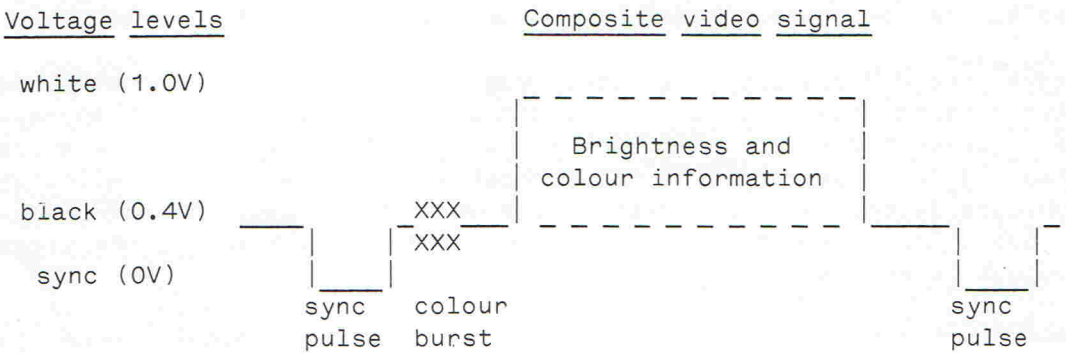
Data is passed into the RAM chips by direct connection of their Din pins to the buffered data bus. Data is read out to the bus via Z16, and to the video via Z8. Chips on the additional 16K RAM card are essentially wired in parallel with the chips on the main PCB, but separate data latches are provided to drive the data bus, enabled by the /RAM2KD signal from Z4.

The memory strobes, RAS and CAS, and the RAM address lines are required to work at high data rates, and for this reason 33R resistors are wired in line with some signals to damp reflections and ringing. Care should be taken when examining RAM timing that measuring instruments do not substantially modify the signals by capacitive loading.

4.5 PAL colour encoder.

(See diagram on page 61)

The PAL colour encoder board is mounted on the main PCB via a 14-way connector. It accepts sync, video and colour information, and produces a composite PAL colour output. One scan line of video signal is shown diagramatically below:-



The sync pulses shown are for horizontal syncronization, and occur at 64us intervals. There are additionally vertical sync pulses which occur for about 0.5ms every 20ms.

The transistors Q1 and Q2 are responsible for controlling the voltage levels at the video output. R2 and R3 provide the basic voltage level around which the video signal is generated, and Q1 clamps this to 0V during sync pulses, when the CSYNC line goes high. Q2 is used as an emitter follower to give a low impedance output suitable for driving a 75 Ohm monitor.

The colour-burst is used as a phase reference within the PAL system, and is produced by gating the Ø1 and Ø4 clocks through Z5. The duration of the burst is controlled by R10 and C6, and it is triggered at the end of the horizontal sync pulse from HSYNC. The burst signal is mixed by R14 and R15, and spurious harmonics removed by L1 before being fed to Q2.

The colour information in the video signal is contained in the colour subcarrier (at 4.43MHz). The amplitude of the subcarrier determines the intensity of colour and the phase difference between the subcarrier and the colour-burst reference determines the colour that is displayed. Z1 (right half) and Z5 (adjacent) produce four clock signals, all at the colour subcarrier frequency, but at 90 degree phase intervals. To produce varying colours the four clock signals are mixed by Z4 which is controlled by the colour data, C0' to C3'. The video and blanking signals are applied via D1 and D2, and the result is mixed into the output via R4/C5 with filtering applied by L2.

In the PAL system it is necessary for phase angles to be reversed on alternate lines. For this reason Z1 (left half) is made to change state every line (by dividing the HSYNC pulses by 2), producing the 'E' signal. This in turn reverses the phase of the Ø1 and Ø3 clocks through Z2 before they are used by the colour signal mixing circuits.

Also involved in the colour mixing is Z3. This is used to null any colour information from the mixer when no video output is required. This

improves character contrast on a TV display. Q3 is used to suppress any signal from the mixer that may interfere with the colour-burst signal.

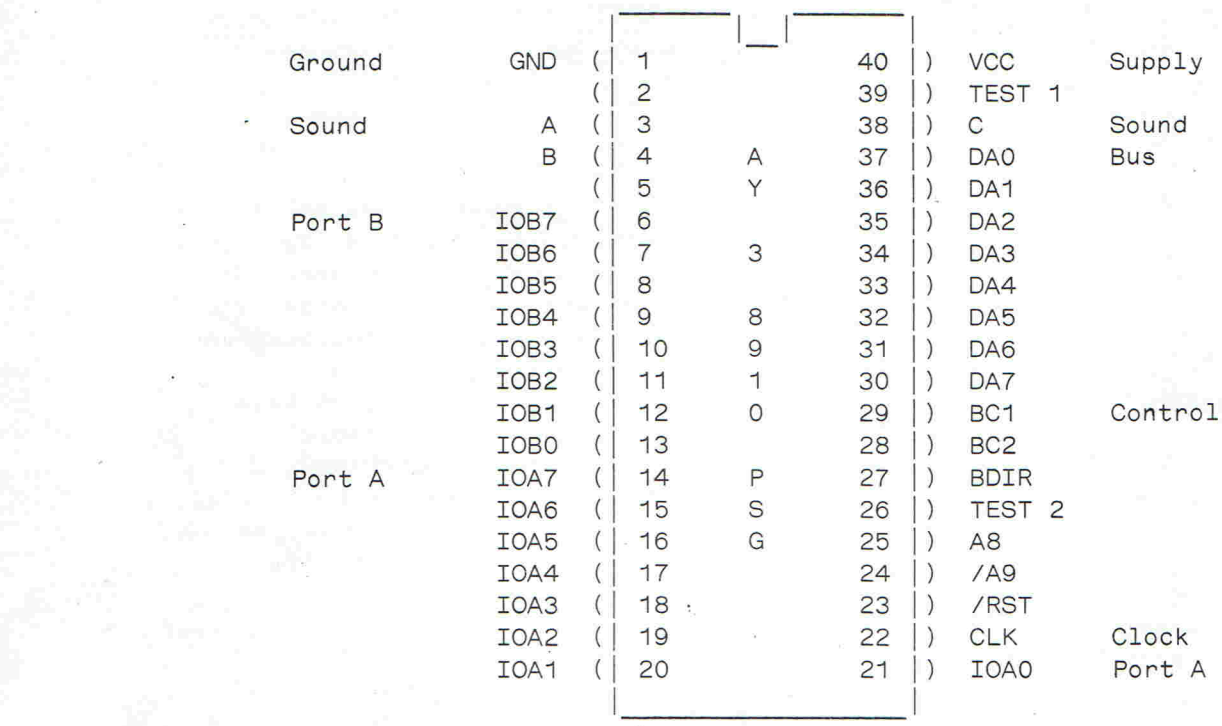
A table of displayed colours and colour data is shown below:-

Colour data				Colour	BASIC	Displayed colour
CC3	CC2	CC1	CC0	RAM data	colour no.	(approx)
0	0	0	0	0	10	Grey
0	0	0	1	1	7	Cyan
0	0	1	0	2	3	Red
0	0	1	1	3	1	White
0	1	0	0	4	4	Yellow
0	1	0	1	5	2	Green
0	1	1	0	6	5	Orange
0	1	1	1	7	11	Yellow/green
1	0	0	0	8	6	Blue
1	0	0	1	9	9	Dark blue
1	0	1	0	10	16	Violet
1	0	1	1	11	12	Blue
1	1	0	0	12	13	Grey
1	1	0	1	13	14	Turquoise
1	1	1	0	14	8	Magenta
1	1	1	1	15	15	Grey

4.6 Interface circuits.

(See diagram on page 60)

Pin connections and signal descriptions for the AY-3-8910 programmable sound generator chip are shown below:-



Bus	DA0-DA7	A bi-directional bus through which the PSG communicates with the processor. Data is stored in several internal registers, and both register data and addresses are sent through this bus.
Sound	Outputs A B C	The three analogue channel outputs are connected together to produce the sound signal.
Ports	IOA0-7 IOB0-7	Two 8-bit ports, A and B, may be separately configured for input or output.
Clock	CLK	The clock input controls all internal timing and frequency generation. In the Colour Genie this is fed with the CPU clock at 2.2MHz.
Control	/RST	A low level on the reset input clears all registers in the PSG and terminates any current operation. Both I/O ports are set to input mode.
	/A9	This input must be low for the PSG to communicate through the bus.
	A8	This input must be high for the PSG to communicate through the bus.
	BDIR	This bus direction input is taken high to write data or register addresses into the PSG via the bus, and low to read data from internal registers.
	BC1	The bus control input is low to write data to the

	PSG registers and high to set address or read data.
BC2	The second bus control input is permanently high.
TEST	The two test pins are not used.

The PSG chip bus is connected directly to the processor data bus. The bus control signals for the PSG, BDIR and BC1, are derived from the decoded ports F8H and F9H by Z53 and Z54 allowing the processor to set PSG register addresses via port F8H, and transfer data via port F9H.

4.6.1 Parallel interface.

The two 8-bit parallel interfaces are simply connected to the IOA and IOB ports of the PSG chip. Within the chip these ports are represented by register numbers 14 and 15 respectively. Details are given in the programming section on general use and in the accessories section on how the joysticks and parallel printer interface use the port.

4.6.2 Sound output.

The sound output from the PSG chip is buffered and amplified by half of Z62. The output is fed to the modulator and to the audio output jack via a protection resistor R13.

4.6.3 Serial interface.

The serial output signal is derived from bit 1 of the control port latch Z25. Op-amp Z62, fed from a +/- 12V supply is used as a V24 level converter. The serial input is converted to a TTL level by transistor Q3, zener diode Z3 being used to lift the threshold voltage above ground, and diode D13 protecting the transistor against the negative input voltage. The TTL signal is buffered to the processor data bus bit 1 through Z52, which is the control port read buffer.

The carrier detect input is processed in a similar way through Q1 to bit 2 of the data bus.

4.6.4 Cassette interface.

The cassette output signal is produced from the control port latch, Z25, and buffered through op-amp Z61 to produce a voltage swing of about +/- 4V. The output is then attenuated by R6 and R7 to about 100mV so that it can be fed into the microphone input of a cassette recorder.

Input from the cassette recorder is fed through D10 and D11 which remove some of the noise from the signal. The input is then passed through a high-pass filter, C6 and R11, and converted to a square wave by Z61 (lower half) acting with a small degree of positive feedback to further reduce noise effects. The output from the op-amp is converted to a TTL level by R10 and D9, then buffered to data bus bit 0 through Z52.

On models fitted with a level meter the cassette input signal is also fed through Q2 to a diode pump, D12 and D13, to detect peak level. C73 is charged, and drives the meter via sensitivity control R49. This should be set to give a mid-point deflection on the meter with an input from the cassette of about 3V peak-to-peak.

4.7 Expansion port signals.

(See page 71 for connections)

The 50-way expansion port provides access to the Z80 bus signals and can be used for general interfacing, memory extension or addition of ROM program cartridges. The signals available are listed and explained below.

CPU address bus	A0 - A15	The address bus is un-buffered and should not be loaded with more than one TTL input for reliability.
CPU data bus	BD0 - BD7	The buffered data bus is bi-directional, and protocols with the read and write strobes should be observed when placing data onto the bus.
Bus control	/RD	The read strobe goes active (low) when the data bus transfers data to the CPU.
	/WR	The write strobe goes low when the CPU writes data onto the bus.
	/MREQ	The memory request line goes low when the processor needs memory access. During this time a valid address is present on the address bus.
	/IORQ	The input/output request line goes low when the processor needs to access an I/O port. An 8-bit port address is presented on A0 - A7 whilst the IORQ line is active.
	/M1	The machine cycle one line is taken low by processor during an op-code fetch cycle. The most common use of this signal is as an interrupt acknowledge, since when an interrupt is serviced by the CPU both the /M1 and /IORQ lines go low simultaneously.
	/RFSH	The refresh line is at a low level during a memory refresh operation. A 7-bit refresh address is placed on A0 - A6.
CPU status	/WAIT	The wait line can be taken low to suspend processor operation temporarily, for example during a read or write operation to slow memory. The /WAIT line should be driven from an open-collector gate, as it is used within the Colour Genie. Prolonged use of the wait line will cause problems of un-refreshed RAM in the computer.
	/HALT	The halt line is taken low by the CPU when it has executed a HALT instruction and is pending an interrupt.
	/BUSRQ	The bus request signal is passed to the Z80 processor, but the hardware in the Colour Genie does not support bus control by an external device, so its use is best avoided.
	/BUSAK	Is similarly useless.

Interrupts	/INT	This line should be pulled low by an open-collector device to generate a processor interrupt.
	/NMI	As above, but a non-maskable interrupt. This is equivalent to pressing the RESET keys on the keyboard.
	/RESET	The reset line can be pulled low to generate a power-up reset. Note that this connection is only an input, and it is not held low by circuitry inside the computer at power-up.
Address decoding	/ROMDIS	If this input is taken low, the internal 16K ROM is removed from the memory map, allowing any external memory to take its place.
	/Cn	Four outputs are available from the address decoders, each going low when a particular range of addresses is present on the address bus:-

/C1	from	C000H	to CFFFH
/C2	from	D000H	to DFFFH
/C3	from	E000H	to EFFFH
/C4	from	FC00H	to FFFFH

Supply	+5v	A five volt supply is available for external circuits. Current consumption should be limited to about 250mA.

4.8 Power supply circuit.

(See diagram on page 63)

The mains supply is fed directly from the input lead to the mains switch and then to the primary of the transformer. In some early models a filtering circuit was incorperated, consisting of a choke and two capacitors, mounted on the PSU circuit board. GREAT CARE should be taken when working on these versions since live mains is present on the underside of the power supply PCB.

There are two secondary windings on the transformer. One, giving about 8V, is rectified by G1 to provide a 10V DC supply which is regulated to +5V. The other winding is centre-tapped to ground, and is rectified by G2 to give +20V and -20V. These supplies are in turn regulated to provide +12V, -12V and -5V supplies.

The +5V regulator uses a 7805 regulator chip to control a larger current carrying transistor. Initially the regulator supplies the +5V rail drawing current through R1. When the voltage drop across R1 reaches about 0.6V, transistor Q1 is turned on, and supplies current to the +5V rail, around the regulator. The regulator is still in control, however, since if the rail should rise above 5V, then it will pass less current, and in turn shut down the transistor.

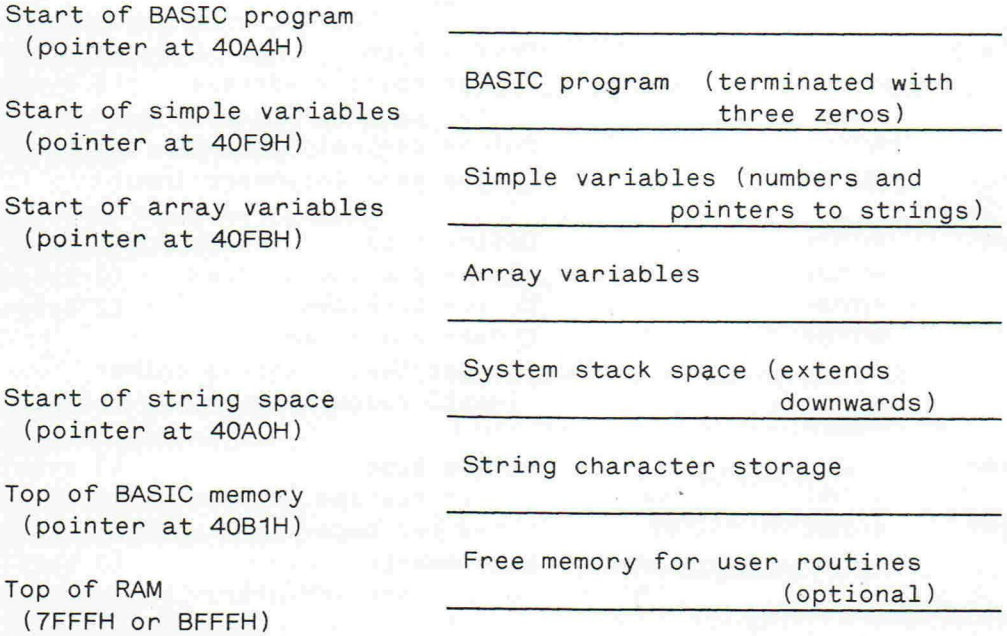
The 12V supplies are regulated entirely by 7812 and 7912 regulator chips. The -5V supply is derived from the -12V rail on the main PCB, and a circuit for this regulator can be found on page 56. Zener diode Z1 provides a constant voltage source at -5.6V with current through R1. The transistor, Q1, acts as an emitter-follower, providing a regulated supply at -5V.

5.1 BASIC memory utilization.

The BASIC interpreter in the Colour Genie reserves some areas of RAM for storing pointers, workspace, and dynamic program information. This section lists a few of the useful locations that may be modified by the user to achieve particular results. A rough memory map of the area of RAM called the 'Communication Region' is given below. Sections of this are discussed in detail later.

Address (hex)	Function
4000 to 4014	Z 80 restart vectors
4015 to 401C	Keyboard device control block (DCB)
401D to 4024	Screen DCB
4025 to 402C	Printer DCB
408E / 408F	Entry address to USR routine
40A0 / 40A1	Pointer to start of string space
40A4 / 40A5	Pointer to start of BASIC program
40A7 / 40A8	Pointer to keyboard input buffer
40B1 / 40B2	Pointer to top of BASIC memory space
40DF / 40E0	SYSTEM tape entry address
40F9 / 40FA	Pointer to end of BASIC program
40FB / 40FC	Pointer to array variables
41E2 to 41E4	Exit vector from SYSTEM command
41E8 to 42E7	Keyboard input buffer (256 bytes)
42F0 to 42FF	Text mode CRTC register values
4300 to 430F	Graphic mode CRTC register values
431C	Copy of byte output to system control port

The layout of BASIC program and variables in memory is shown below. A BASIC program starts at the first free location above the video memory, this being 5801H (22529) normally, and 4801H (18433) if the high resolution mode is de-selected.



5.1.1 Restart vectors.

The Z 80 restart instructions are vectored from ROM to an area at the bottom of RAM. Most of the restarts are used by BASIC. Their functions are listed below, and care should be taken if any of the vectors are changed. The most useful vectors are those for RST 30H, the BREAK vector, and RST 38H which is the processor interrupt vector.

Restart	Vector	Function
RST 8	4000H	The byte at (HL) is compared with the byte following the RST 8. If different a syntax error is produced.
RST 10H	4003H	Returns the next character in the A reg from a string pointed to by HL. Spaces are skipped, and the C flag is set if the char is numeric. HL register pair is incremented.
RST 18H	4006H	Compares register pairs HL and DE. The Z flag is set if they are equal, otherwise the C flag is set if DE is greater than HL.
RST 20H	4009H	C flag reset if double precision number is being processed, else C flag set.
RST 28H	400CH	BREAK key vector. Keyboard routine jumps here if BREAK key is pressed.
RST 30H	400FH	Used by disk system.
RST 38H	4012H	Processor interrupt. Also used by disk.

5.1.2 Device Control Blocks.

The keyboard, screen and printer have Device Control Blocks (DCB's) located in RAM. The three DCB's are set up as follows:-

DCB	Location	Value	Function	
Keyboard	4015H	1	Device type	(1 byte)
	4016H	03E3H	Driver routine address	(2 bytes)
	4019H	7	Cursor control R11 CRTC	(1 byte)
	401AH	64	Cursor control R10 CRTC	(1 byte)
	401BH	'KI'	Device name (Keyboard Input)	
Screen	401DH	7	Device type	(1 byte)
	401EH	30E4H	Driver routine address	(2 bytes)
	4020H	varies	Cursor location	(2 bytes)
	4022H	20H	Cursor character.	
	4023H	0 to 15	Current text printing colour (BASIC colour number -1)	(1 byte)
Printer	4025H	6	Device type	(1 byte)
	4026H	04E7H	Driver routine address	(2 bytes)
	4028H	67	Lines per page	(1 byte)
	4029H	varies	Line counter	(1 byte)
	402BH	'PR'	Device name (PRinter)	

Information in the DCB's can be changed if new device drivers are

written, for example a printer driver that uses the serial port instead of the normal parallel port. Setting the device type to zero will disable input or output through the device. For example, the keyboard can be turned off by setting location 4015H (16405) to zero, and the screen by setting 401DH (16413) to zero.

5.1.3 Format of stored BASIC program.

BASIC program is stored in memory in ascending line order, as it would normally list. Expressions and numbers are stored as ASCII characters, but reserved words, PRINT / FOR / NEXT etc, are stored as one or two byte tokens. All tokens have bit 7 set. Line numbers are stored as 16-bit integers in two bytes, and are immediately preceded by a 16-bit pointer to the start of the next line. (The layout is similar to an indexed-sequential file). Each line is terminated with a zero byte, and the last line is terminated with three zero bytes.

5.1.4 Pointers to memory areas.

As shown above, the memory is partitioned into defined storage areas with pointers indicating the bounds of each area. The pointers may be modified (with care) to provide free memory for machine language programs or allow merges of BASIC programs.

The top of memory pointer at 40B1H (16561) can be set at the MEMORY SIZE ? prompt at switch-on, or is set to the physical top of RAM if no value is entered. The string space pointer is set 50 bytes below this unless a CLEAR command is executed to reserve more string space.

5.2 Hints and tips.

5.2.1 Disabling the BREAK key.

The BREAK vector can be modified to disable the BREAK key on the keyboard. Location 400CH (16396) normally contains a RET instruction, 0C9H. Changing this to 17H (an RLA instruction), and the following byte to 0C9H will disable the BREAK.

Thus POKE 16396,23 : POKE 16397,201 disables BREAK
and POKE 16396,201 enables BREAK

5.2.2 Merging programs.

BASIC programs can be merged by first loading one program then moving the start of program pointer to the end of this and loading the second program. Restoring the pointer to its original place will yield a single program consisting of the two parts. Line numbers in the two programs must not conflict, so the second must start at a line number higher than the end of the first. The merger is possible because the CLOAD command re-locates the line link' values within a program as it loads.

5.2.3 Details of BASIC program merge.

1) Find the value of the start of program pointer at 40A4H (16548).
   ie PRINT PEEK (16548) , PEEK (16549)
   Record these two values, which I shall call A1 and A2.

2) CLOAD the first program.

3) Find the value of the end of program pointer at 40F9H (16633).
   ie PRINT PEEK (16633) , PEEK (16634)
   R(ecord these values as B1 and B2.

4) Decrement the end of program pointer value (B1 and B2) by two.
   ie Subtract 2 from B1
   If B1 is still positive or zero then all is well, but
   If B1 is negative then add 256 to B1 and subtract 1 from B2.

5) Set new start of program pointer from B1 and B2.
   ie POKE 16548,(value B1) : POKE 16549,(value B2)

6) CLOAD the second program.

7) Reset the start of program pointer to its original value.
   ie POKE 18548,(value A1) : POKE 16549,(value A2)

The two programs should now be merged into one.

5.2.4 Recovery of program.

If a NEW or CLOAD command is typed accidentally, then any BASIC program in the computer will be lost - it will not LIST or RUN. However,

the program is not erased, merely the first byte has been set to zero. To recover the program this byte should be set to one, and the pointer to the end of the program suitably set. The line link values in the program may also need setting, and a call to a ROM routine provides an easy way to do this. The process is detailed below:-

1) Set the first byte of the program to a 1.
   ie POKE PEEK (16548) + 256*PEEK (16549) , 1

2) Reset the end of program pointer and the line link values.
   ie CALL 2C83

5.2.5 Finding the name of system-load programs.

It is not possible to load a tape with the SYSTEM command unless the name of the tape is known. The following small BASIC program will print the name of a program on a system-load tape.

```
10 CLEAR 500 : INPUT #-1,A$
20 IF LEFT$ (A$,1)="U" THEN PRINT MID$(A$,2,6) ELSE
    PRINT "NOT A SYSTEM TAPE, OR BAD LOAD"
```

Type the program in, then run it and start the tape. The asterisks should flash briefly in the normal way, then the name should be printed.

5.2.6 BASIC with machine code.

Machine language programs can be safely located above the top of memory pointer without interference from BASIC. The pointer can be set at switch-on by entering a number at the MEMORY SIZE ? prompt, or may be set by the program itself when it is first run. If the program sets the top of memory pointer to a new value below itself, and requires to return to BASIC, a jump to location 06C0H should be used, which will cause a warm start in BASIC. Note that string variables will be corrupted by this action, but any resident BASIC program should be unaffected provided it is not over-written.

Location at the top of memory is recommended for machine code routines occupying more than 100 bytes or so. The program can be set up from a BASIC program using a READ / POKE loop, and storing the machine code program in DATA statements. Alternatively, if a monitor program or an editor/assembler is available to make a tape, the program can be loaded with the SYSTEM command. This provides a much quicker method of loading long programs.

Programs loaded with the SYSTEM command can be made to automatically start after loading by modifying the exit vector from the SYSTEM command, at location 41E2H (16866). This location contains a RET instruction by default, but this may be changed to a jump into the loaded program. For example if a program loaded into memory at 7000H, and started execution at this address then the following code could be loaded into 41E2H for auto-start:-

41E2H	C3H	Jump instruction
41E3H	00H	Entry point of routine (lsb, msb)
41E4H	70H	

5.2.7 Small machine code programs.

Small machine code routines can be located below or within BASIC programs. Two techniques are commonly used: packing a program into a REM statement, and packing into a string variable. Programs should be less than 255 bytes long or things become very difficult.

Packing into a string is convenient if the machine code program is required as a subroutine to the BASIC program. The method to be used is outlined below:-

1) A string variable is set to a constant string somewhere near the beginning of the BASIC program. The constant string should be longer than the machine language program.

```
50 PR$ = "12345678901234567890123456789012345678901"
```

2) The address of the string in memory can be found using the VARPTR function. Note that provided the string variable is never altered by a string expression, it will reside in the program space where the definition is and will not move.

```
100 AD = PEEK (VARPTR(PR$)+1) + 256*PEEK (VARPTR(PR$)+2)
```

This address may need to be determined before the machine code program is finally written, to find any absolute locations. If this is the case remember not to alter any BASIC program before the the string definition.

3) Put the machine code into the string using a READ / POKE loop.

```
110 FOR X = AD TO AD+(length of prog -1)
120 READ D : POKE X,D : NEXT X
```

4) Define the machine language program in DATA statements.

Cautions - Unusual things will happen to your BASIC if the machine code program contains any zero bytes. The following guidelines should be used:-

1) Write programs without zero bytes in them !

If (1) is not possible then

2) Ensure that the string assignment is executed only once, when the program is first run.
3) Do not expect the program to LIST or CSAVE properly after it has been run. This means save it on tape BEFORE you RUN it.

Using REM statements instead of strings for packing machine code programs works in a very similar manner. A complete example is given in section 5.7 with the serial printer driver program, so no more is said here.

5.2.8 Calling machine code from BASIC.

Two methods of executing machine language programs from within BASIC can be used. The CALL XXXX command passes control to a fixed location (hexadecimal value XXXX), and the machine code program should return with a RET instruction. Values can only be passed to or from the program by POKEing and PEEKing into memory locations within the program.

As an alternative to CALL, the USR(X) function allows a value to be passed to and from the machine code program. A further advantage is that the call address is not fixed, and can be calculated (eg from a VARPTR). The location called by the USR(X) function is stored at locations 408EH / 408FH (16526 / 16527), and two 8-bit values forming a 16-bit address must be POKEd into these locations. The value of the argument X can be recovered in the HL register pair if the machine code program calls location 0A7FH on entry. A value is returned to BASIC in the HL register pair by jumping to location 0A9AH at the end of the program. If a returned value is not required, the program may just RET.

5.2.9 Programs in cartridges.

Up to 12K of program can be put in a ROM cartridge. The cartridge memory area is from 0C000H to 0EFFFH. A provision is made for programs in cartridges starting automatically; a character 'C' (43H) should be put at location 0C000H, and the program entry point should be 0C001H. If the keyboard ROM calls are required by the cartridge, then the device driver address at 4016H should be set to 03E6H.

5.3 Keyboard routines.

The keyboard is presented as 256 bytes of memory, located from 0F800H to 0F8FF in the computers memory map. The keys are arranged in 8 rows each of 8 keys. The rows are at the 8 addresses shown in the map below, and each key in a row corresponds to a data bit. If a keyboard address is read, then keys that are pressed yield a one in their respective bit positions.

Keyboard map

Data bit No.		0	1	2	3	4	5	6	7
Decimal value		1	2	4	8	16	32	64	128
Address (Hex)	F801	@	A	B	C	D	E	F	G
	F802	H	I	J	K	L	M	N	O
	F804	P	Q	R	S	T	U	V	W
	F808	X	Y	Z		F1	F2	F3	F4
	F810	0	1	2	3	4	5	6	7
	F820	8	9	:	;	,	-	.	/
	F840	Ret	Clr	Brk	↑	↓	←	→	Space
	F880	Shift	Mod		Rpt	Ctrl			L.P.

There are three ROM routines that can be called to scan the keyboard matrix, these being for both single character and line input.

Routine address	Operation
002BH	An INKEY subroutine. The keyboard is scanned and if any key is pressed its ASCII code is returned in the A register. If no key is pressed, A will contain zero. Registers AF and DE are used.
0049H	A character input subroutine. This routine will wait for a key to be pressed, and then return its code in the A register. Registers AF and DE are used.
0040H	A line input subroutine. Before calling, the HL register pair should be set to point to an input buffer, and the B register should contain the maximum number of characters that may be entered. The subroutine will accept a line of input from the keyboard and store it in the buffer. Input is terminated with either a RETURN or when the BREAK key is pressed, or when (B) characters have been put into the buffer. On exit, the B register contains the number of characters entered and the C flag is set if BREAK was pressed. Registers AF and DE are used.

5.4 Screen and graphics.

5.4.1 The CRT controller chip.

The screen display format and timing is controlled by the CRTC chip, and in turn this is controlled by its internal registers which are set up by the processor. The 16 registers are loaded with data stored in tables in RAM, which are set up when the computer is reset. There are two tables, one for text mode display and one for graphic mode display, since the required data is different for the two cases. The location and contents of these tables is shown below, and a description of the register functions follows. Data is copied from the appropriate table to registers 0 to 15 whenever BASIC returns to command mode, or when FGR or LGR commands are executed.

CRTC setup tables.

Register No.	function	RAM location text mode	RAM location graphic mode	Value text	Value graphic
0	Horizontal total	42FFH (17151)	430FH (17167)	70	70
1	Horiz displayed	42FEH (17150)	430EH (17166)	40	40
2	Horiz sync posn	42FDH (17149)	430DH (17165)	52	52
3	Sync pulse width	42FCH (17148)	430CH (17164)	094H	094H
4	Vertical total	42FBH (17147)	430BH (17163)	38	126
5	Vert total adjust	42FAH (17146)	430AH (17162)	0	31
6	Vert displayed	42F9H (17145)	4309H (17161)	25	96
7	Vert sync posn	42F8H (17144)	4308H (17160)	30	108
8	Interlace & skew	42F7H (17143)	4307H (17159)	0A0H	20H
9	Max raster address	42F6H (17142)	4306H (17158)	7	7
10	Cursor control	42F5H (17141)	4305H (17157)	0C4H	20H
11	Cursor end raster	42F4H (17140)	4304H (17156)	7	0
12	Start addr (msb)	42F3H (17139)	4303H (17155)	04H	08H
13	Start addr (lsb)	42F2H (17138)	4302H (17154)	00H	00H
14	Cursor (msb)	42F1H (17137)	4301H (17153)	00H	00H
15	Cursor (lsb)	42F0H (17136)	4300H (17152)	01H	00H

The CRTC chip contains 18 registers, the first 16 being used as display parameters and are write-only, the remaining two forming the light-pen register which is read-only.

Register	Function
0 Horizontal total	Contains the number (-1) of character slots required to produce a 64us scan line. In the Colour Genie this number is 71.
1 Horiz displayed	The number of characters displayed on a line of the screen. Note that in graphic mode, 4 pixels represent a 'character'.
2 Horiz sync posn	The number in this register determines the position of the horizontal sync pulse in relation to the video signal (see page 17). Altering this number will move the picture sideways on the screen.

3	Sync pulse width	The duration of horizontal and vertical sync pulses is determined by this register.
4	Vertical total	The vertical total number specifies the number (-1) of character rows required to produce a 20ms frame time. This register has only 7 bits, so a maximum value of 128 can be chosen. For graphic mode a value of 140 is required here, and the resulting discrepancy causes rolling or movement of the graphic display on some televisions.
5	Vert total adjust	The vertical total adjust is a 5-bit register that allows the frame scan time to be fine tuned to exactly 20ms.
6	Vert displayed	This register determines the number of rows of characters to be displayed. In graphic mode, it determines the number of rows of pixels.
7	Vert sync posn	The position of the frame sync pulse in relation to the picture is determined by this register. Changing its value will move the picture vertically.
8	Interlace & skew	The contents of this register control the display blanking and cursor timing. Change is not recommended.
9	Max raster address	Contains the number (-1) of scan lines in a row of characters. This number is normally 8 for character mode (8 lines per char) and 2 for graphic mode (2 lines per pixel).
10	Cursor control	The cursor control register defines the cursor attributes and the position of the top line of the cursor. Bits 5 and 6 determine the blink rate. Bits 0 to 5 specify the start position scan line within the character row. Position 0 (normally set up) starts at the top of the character, position 7 would start at the bottom of the character. Bits 5 and 6 are used as follows:-

bit 6	bit 5	Cursor mode
0	0	Non blinking
0	1	Not displayed
1	0	Fast blink
1	1	Slow blink

11	Cursor end raster	This 5-bit register determines the position of the bottom of the cursor. A value of 7 selects full character depth.
12	Start addr (msb)	The two registers 12 and 13 combine to produce
13	Start addr (lsb)	a 14-bit address which is the start address of the screen memory. Hardware constrains this to be in the range 4000H to 7FFFH.
14	Cursor posn (msb)	Registers 14 and 15 similarly combine to give
15	Cursor posn (lsb)	the cursor position address.
16	Light pen (msb)	The light pen register contains the character
17	Light pen (lsb)	address where the light pen strobe input was last active.

Some of the CRTC's registers are continually being changed by the computer as programs are run. These are R10, R11, R14 and R15, all responsible for cursor control. Although initially set from the tables shown above, the values normally in these registers are derived from other locations shown below.

Register	RAM Location	Value	Description
10 Cursor ctrl	401AH (16410)	64	Fast-blinking cursor.
11 Cursor end	4019H (16409)	7	Full depth cursor.
14 Cursor addr	4021H (16417)	varies	Cursor position (msb).
15 Cursor addr	4020H (16416)	varies	Cursor position (lsb).

Thus to change the form of the cursor on the screen, location 401AH (16410) should be changed. Try, for example, POKE 16410,102.

5.4.2 Screen ROM calls.

Routine address	Operation
0033H	Writes contents of A reg at current cursor position and in current colour. Cursor positioning codes are interpreted and acted on. On return the contents of the A reg are destroyed. The cursor position can be set in advance by writing a 16-bit cursor address to 4020H (16416), and the printing colour is held at 4023H (16419). The colour is stored as one less than the BASIC colour number, ie 1 is green.
01C9H	Clears the screen and sets the cursor to the top left corner.
38A9H	FGR Sets up the graphic screen.
38B0H	LGR Sets up the text mode screen.

5.4.3 Other points.

In graphic mode, the screen memory is used directly as bit image data. Each byte in the memory represents four adjacent pixels on a line, two bits being used for each pixel. Each 2-bit number specifies one of the 4 colours that the pixel may take. The pixels are displayed starting with bits 6 & 7, and ending with bits 0 & 1, then the next byte is displayed in a similar way.

If the ROM routines are not used for text display, then you should not forget about the colour memory at 0F000H to 0F3FFH. Each character on the screen requires a colour value (4 bits) in this RAM as well, at the appropriate location. A character at 4437H will have a colour value at 0F037H etc. - the addresses A0 to A9 are the same for the two memories.

The character set displayed in text mode can be changed (as with the CHAR command in BASIC) by sending suitable values to the control port (port 0FFH (255)). Bits 3 and 4 control character generator selection in the following way:-

bit 3 value	bit 4 value	Character codes 0 to 127	128 to 191	192 to 255
0	0	Alpha	Prog	Prog
0	1	Alpha	Prog	Graph
1	0	Alpha	Graph	Prog
1	1	Alpha	Graph	Graph

Alpha = Alphanumeric characters.
Prog = Programmable graphic characters.
Graph = Graphic characters.

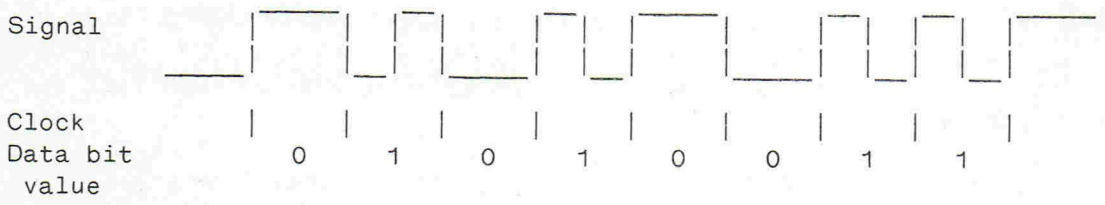
The remaining bits in the control port (other than bits 3 and 4) should be set in accordance with the image of the control port data at RAM location 431CH (17180), which should also be updated with the new value written to the port.

If necessary, the CRTC registers can be set directly from a program. To do this, first output the number of the register to change on port 0FAH (250), then send the required data out of port 0FBH (251). Remember that the computer will set register values back to those in its table whenever it returns to BASIC command mode.

5.5 Cassette details.

The cassette system stores data at 1200 baud on a standard audio cassette. The hardware in the Colour Genie simply operates as a level converter between the computer and the cassette recorder, and signal production and decoding is left to the software.

Data bits are encoded in a self-clocking system using the 'no change for zero' technique. The system is best understood as follows; imagine a clock running at the rate that bits are sent to the cassette, that is 1200 ticks per second. At every tick the state of the signal changes (from low to high or high to low). These ticks mark the beginnings and ends of data bits. Additionally the signal changes state half-way between two ticks if the data bit is currently a one. The diagram below shows a sample of cassette signal.



Data bytes are sent out on the tape end-to-end with no separating bits between them. There are three formats of tape; BASIC programs, SYSTEM tapes, and tapes holding data from PRINT#-1 statements. All tapes start with a leader, which is 256 bytes of zeros, followed by a sync byte which is 0A5H.

BASIC tapes then have a header of three 0D3H's, then the fist byte of the program name, then the memory image of the program ending in three zeros.

DATA tapes simply follow the header with the printed data in ASCII characters, terminated with a carriage return (0DH).

SYSTEM tapes consist of blocks. A block may be one of three types:-

A filename block, consisting of a filename header (55H) and a 6-byte name, padded with blanks if necessary.

A data block, with a data header (3CH) then a count byte specifying the number of data bytes in the block. This is followed by a loading address for the data, then the data, and then a checksum byte.

An entry block, with an entry header (78H) and an entry address. The entry block signifies the end of data on the tape.

All types of tape are read and written by the same cassette subroutines in ROM. These are:-

Routine address	Operation
023FH	Write leader and sync byte to the cassette. Uses registers AF and C.
021FH	Write byte in A register to cassette. No other registers used.
024CH	Read leader and sync byte, and put asterisks on screen. Register AF is used.
01EDH	Read byte from cassette into A register. No other registers are used, but the routine must be called sufficiently often to ensure that data is not lost. Essentially there is just time to put the data into memory and increment a pointer.

5.6 Sound generator and parallel port.

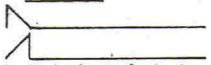
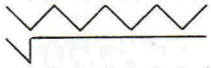
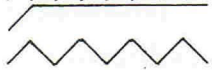
5.6.1 PSG chip registers.

The operation of the programmable sound generator chip is controlled by 15 internal registers. A description and explanation of these is given below.

Register No.	Function	No. of bits	Description
0	Channel A fine tune	8	The coarse and fine tune registers form a 12-bit period count which determines the pitch produced by the tone generator for each of the three channels. Frequency of generator = 138550 / period
1	Channel A coarse tune	4	
2	Channel B fine tune	8	
3	Channel B coarse tune	4	
4	Channel C fine tune	8	
5	Channel C coarse tune	4	
6	Noise tone control	5	A number from 0 to 31 controls the tonal quality of the white noise source.
7	Function control	8	This register controls both the sound channel outputs and the I/O port data directions. Bit assignments are:-

Bit	Function
0	Zero for tone on channel A
1	Zero for tone on channel B
2	Zero for tone on channel C
3	Zero for noise on channel A
4	Zero for noise on channel B
5	Zero for noise on channel C
6	Zero for port A input, one for port output.
7	Zero for port B input, one for port output.

8	Channel A volume	5	The volume registers provide a 4-bit level control (values 0 to 15) or select the envelope generator to control the channel amplitude (value 16).
9	Channel B volume	5	
10	Channel C volume	5	
11	Envelope fine tune	8	The envelope tune registers combine to a 16-bit period count that determines the speed at which the envelope generator runs.
12	Envelope coarse tune	8	
13	Envelope shape	4	The envelope shape register controls the output of the envelope generator. The shapes shown below are available:-

Value	Shape
0	
4	
8	
10	
11	
12	
13	
14	

Register No.	Function	No. of bits	Description
14	I/O port A	8	The I/O port registers represent the data that is presented at the parallel ports A and B. Data can be read or written depending on the direction the ports are set to by reg 7.
15	I/O port B	8	

The PSG registers are accessed through two ports from the Z 80. The required register number should be sent to port 0F8H (248), then data can be read or written through port 0F9H (249).

5.6.2 ROM routines.

The printer and joystick controllers operate through the parallel port, which is controlled by the PSG chip. Given here are details of ROM calls that can be used to operate these devices. It is important to note that the ROM calls listed for joystick and keypad use are available only in the new ROM's. The accessories section gives details of the operation of the routines for the joysticks, keypads and printer.

Routine address	Operation
003BH	PRINTER The byte in the A reg is sent to the printer. Note that certain bytes are not sent, ie line feed (10), top of form (11) and form feed (12). If A is zero on entry, then a status check is performed, and the Z flag set if the printer is ready to accept a character.
3AB3H	PSG WRITE The contents of the L reg are written into the PSG register number specified in H.
3ABBH	PSG READ On entry the H reg contains the number of the PSG register to be read. The result is returned in A.
* 3A5EH	JOYSTICK On entry the D reg contains 1, 2, 3 or 4 to specify the 1X, 1Y, 2X or 2Y joystick to be read. The position value from 0 to 63 is returned in A.
* 3A87H	KEYPAD Register D should contain 0FEH if keypad 1 should be read, or 0F7H if keypad 2. At return the A reg contains zero if no keys are pressed, or the key value. Keys 0 to 9 return 0 to 9, key 0 returns 10, and the other two keys return 11 and 12.

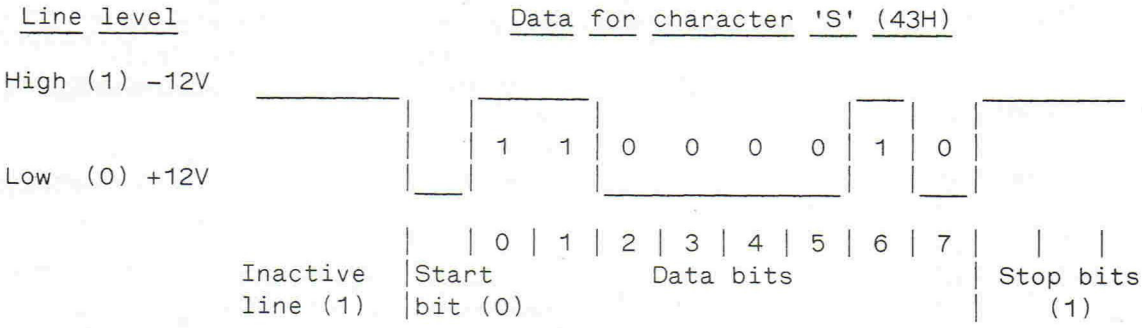
* New ROM's only.

5.7 Using the serial port.

The serial interface on the Colour Genie is via the control port, port OFFH (255). The state of the transmit data line is controlled by writing to bit 1 of the control port. If the port is read, the receive data input is presented on bit 1, and the carrier detect input on bit 2.

Details are given below of machine language subroutines to send and receive data through the serial port. The programs perform the serial-to-parallel and parallel-to-serial conversion and the necessary timing, and operate one character at a time. When using the receive routine it is important that the calling program does not take too long to process the characters as they are received, since subsequent characters will be missed or corrupted.

5.7.1 Serial data format.



Each character sent over the serial data line is preceded by a start bit, which is always zero. This provides a synchronization point for the receiver. The data is then sent as 8 serial bits, with bit 0, the least significant bit, sent first. After the data, one or two stop bits are sent. These are at the same level as an inactive line, and are there to ensure some gap between characters.

The rate at which data is sent is termed the BAUD RATE. A transmission rate of 300 bits-per-second is at 300 baud. Note that these bits include start and stop bits as well as data bits, so the data transmission rate will only be 70% to 80% of the baud rate.

5.7.2 Serial transmit subroutine.

The transmit subroutine transmits the byte in the A reg from the serial port. Before using this routine it should be noted that BASIC sets the state of the TXD line incorrectly (ie active) and this may cause errors to be generated by equipment connected to the line. To overcome this the following BASIC can be typed in to set the line to its correct value.

```
POKE &H431C,PEEK (&H431C) OR 2
```

or with new ROM's

```
SET ~~&H431C,1~~
SET 1,&H431C
```

The transmitted baud rate is determined by the delay BDRATE. This can be calculated from BDRATE = (85000/baud rate) -4.

The carrier detect input can be sensed by reading from the control port and examining bit 2. The sense of this bit is opposite to that of the CD input, ie a high level on the input (+12V) results in a zero on bit 2.

```
TXBYTE  PUSH BC          Save registers on stack.
        PUSH DE
        LD   E,A         Save byte to send in E.

        LD   A,(431CH)   Get control port value from image
        AND  0FDH         in RAM, and zero TXD bit.
        LD   C,A         Save this in C reg.

        LD   D,1         Put stop bit into D reg.
        SLA  E           Carry top bit from E into D.
        RL   D           Now DE holds 10 bits to send,
                          with data, start and stop bits.

        LD   B,10        Set bit counter.
SLOOP   XOR  A           Clear A
        SRA  D            and shift lsb out of DE
        RR   E            into carry flag.
        ADC  A,0         Put carry flag into A
        SLA  A            in bit 1.
        OR   C           Set up control port value
        OUT  (0FFH),A     and output.
        PUSH BC
        LD   BC,BDRATE   Wait for one bit period
        CALL 0060H        using ROM delay routine.
        POP  BC
        DJNZ SLOOP       Loop for all 10 bits.
        POP  DE          Restore registers
        POP  BC
        RET               and return.
```

5.7.3 Serial receive subroutine.

The receive subroutine provides the basis of a serial character input program, it returns with the received character in the A register. The timing of call is critical; if the routine is not called before or during the start bit, data will be lost. Operation above 1200 baud is not really practical. The baud rate delay variable BDRATE should be calculated from (85000/baud rate) -4, but this value may need to be changed slightly for successful results.

```
RXBYTE  PUSH BC                 Save registers on stack
        PUSH DE

STLOOP  IN   A,(0FFH)           Wait for start bit:
        AND  2                    Mask RXD bit from port
        JR   NZ,STLOOP            Loop until zero.

        LD   B,8                Set counter for bits

RDLOOP  PUSH BC                 Wait one bit period for next bit
        LD   BC,BDRATE
        CALL 0060H              Use ROM delay routine
        POP  BC                 Restore reg

        IN   A,(0FFH)           Read control port data
        RRA                     Rotate RXD bit into carry flag
        RRA
        RR   E                   and rotate this into E reg.
        DJNZ RDLOOP             Now go and get next bit.

RXDONE  LD   A,E                Put received char into A.
        POP  DE                 Restore register contents
        POP  BC
        RET                      and return.
```

5.7.4 Serial printer driver program.

The following BASIC program provides a machine code routine to drive a serial printer via the serial interface. The program illustrates how machine code can be stored and set up within BASIC, and how memory pointers and vectors can be modified. Since the program changes part of itself, it should be saved before being run.

The carrier detect input is also sampled by the program. Options allow this to be used to prevent printer buffer overflow if the printer is connected correctly. The CD input is used as a 'printer ready' signal to the computer to start sending data.

```
  0 REM 1234567890123456789012345678901234567890      This line must be
     1234567890123456789012345678901234567890          typed in - it is
                                                      overwritten with
                                                      machine code.
100 CLEAR 500 : PR=&H4026 : SB=&H40A4                 Pointers to printer
                                                      driver addr. and
                                                      start of BASIC.
110 CP=PEEK (&H431C) OR 2 : POKE &H431C,CP :          Set TXD line to
     OUT 255,CP                                       inactive state.
120 ST=PEEK (SB) + 256*PEEK (SB+1)                    ST is BASIC start.
130 B1=PEEK (ST) : B2=PEEK (ST+1)                     B is link address.
140 POKE SB,B1 : POKE SB+1,B2                         Move start of BASIC
                                                      pointer above m/c.
```

```
200 PS=ST+5 : PC=PS : READ H$

210 GOSUB 1000 : PRINT ".";
220 POKE PC,H : PC=PC+1
230 READ H$ : IF H$ <>"END" THEN 210
240 PRINT : PRINT
250 IF PC>(B1+256*B2) THEN PRINT "ERROR -
     LINE 0 NOT LONG ENOUGH" : STOP

300 INPUT "TRANSMIT 7 OR 8 DATA BITS (7/8)",I$
310 POKE PS+6,24 : POKE PS+7,19
320 IF I$="8" THEN 400 ELSE IF I$ <>"7" THEN 300

330 INPUT "PARITY EVEN OR ODD (E/O)",I$
340 IF I$="E" THEN P=0 ELSE IF I$="O" THEN P=47 ELSE 330
350 POKE PS+23,P

400 INPUT "SENSE CARRIER DETECT (Y/N)",I$
410 POKE PS+46,0 : POKE PS+47,0
420 IF I$="N" THEN 500 ELSE IF I$ <>"Y" THEN 400

430 INPUT "TRANSMIT WHEN HIGH OR LOW (H/L)",I$
440 IF I$="H" THEN P=32 ELSE IF I$="L" THEN P=40 ELSE 430
450 POKE PS+46,P : POKE PS+47,250

500 INPUT "ADD LINE FEED AFTER RETURN (Y/N)",I$
510 IF I$="N" THEN P=201 ELSE IF I$="Y" THEN P=192 ELSE 500
520 POKE PS+73,P

600 INPUT "BAUD RATE",BR
610 IF BR<45 OR BR>9600 THEN 600
620 BD = 85251.5/BR - 4.846
630 B2=INT (BD/256) : B1=INT (BD-B2*256)
640 POKE PS+50,B1 : POKE PS+51,B2

700 PRINT : PRINT
710 INPUT "DO YOU WISH TO RE-ENTER ANYTHING (Y/N)",I$
720 IF I$<>"N" THEN 300

800 B2=INT (PS/256) : B1=PS-256*B2
810 POKE PR,B1 : POKE PR+1,B2
820 PRINT : PRINT "SERIAL DRIVER NOW LOADED"
830 END

1000 H=0 : FOR X=1 TO LEN (H$)
1010 P=ASC (MID$ (H$,X,1))-48 : IF P>9 THEN P=P-7
1020 IF P<0 OR P>15 THEN PRINT "ERROR IN DATA
     STATEMENTS" : STOP
1030 H=16*H+P : NEXT X : RETURN

2000 DATA 79,FE,0D,F5,C5,D5,18,13,E6,7F,5F,07,AB,47,07,07
2010 DATA A8,47,07,07,07,07,A8,2F,E6,80,B3,5F,3A,1C,43,E6
2020 DATA FD,4F,16,01,CB,23,CB,12,06,0A,DB,FF,CB,57,28,FA
2030 DATA C5,01,02,03,CD,60,00,C1,AF,CB,2A,CB,1B,CE,00,CB
2040 DATA 27,B1,D3,FF,10,EA,D1,C1,F1,C0,3E,0A,B7,18,B4,END
```

PS is where m/c prog starts. Now read program, convert hex data to dec and poke into memory. Stop if code is too long for REM stmt.

Get options and modify program.

Required delay is calculated from baud rate.

Put driver address in printer DCB.

Convert hex in H$ to decimal in H.

M/C program in hex.

5.7.5 Terminal operation.

To use the Colour Genie as a terminal, the serial transmit and receive routines must be combined with the screen and keyboard routines to allow 'simultaneous' transmission of characters typed on the keyboard and screen display of characters received on the serial port.

The difficulty of simultaneous operation means that the routines outlined below will not give very satisfactory performance without a deal of fine tuning, and probably addition of a character receive buffer.

The basic technique is shown in this routine.

```
TERM    CALL 002BH          Call INKEY routine to get key.
        OR   A              Check if key pressed
        CALL NZ,TXBYTE       and transmit char if there is one.

        IN   A,(0FFH)       Check RXD line to see if active
        AND  2
        JR   NZ,TERM        If not then loop back to start

        CALL RXBYTE          otherwise read char coming in
        CALL 0033H           and display it on the screen.
        JR   TERM           The loop back to start.
```

The problem in the above routine is that part of the incoming serial data may be lost while the computer is transmitting data or servicing the screen or keyboard. A great improvement is to make use of the interrupt facility of the Z 80 processor. A hardware modification to the computer is required so that an interrupt is generated when the serial start bit is received. This is simply a diode (type 1N914 or 1N4148) connected between the /INT line and the RXD line. The anode of the diode should connect to pin 16 of the Z 80 and the cathode (band end) to pin 4 of Z52 (74LS367).

With this modification, the receive routine should be called whenever an interrupt is taken (the interrupt should be vectored through location 4012H). Received characters should be stored in a buffer, and the terminal program then written to transfer characters from the buffer to the screen, and from the keyboard to the transmit routine. The transmit routine should be run with interrupts disabled so that its timing is not compromised.

5.8 ROM calls and RAM locations.

5.8.1 List of ROM calls.

Keyboard routines

	002BH	INKEY Returns key code in A, or zero if no key pressed.
	0049H	INPUT CHAR Waits till key pressed, then returns code in A.
	0040H	INPUT LINE Accepts line of input. HL must point to buffer.

Screen routines

	0033H	DISPLAY CHAR Prints character in A at cursor position.
	01C9H	CLEAR SCREEN Clears screen and homes cursor.
	28A7H	DISPLAY TEXT Prints text pointed to by HL, until CR or 0.
	38A9H	FGR Set up high res graphic screen.
	38B0H	LGR Set up text screen.

Cassette routines

	023FH	WRITE HEADER and sync byte to cassette.
	021FH	WRITE BYTE in A to cassette.
	024CH	READ HEADER and sync byte.
	01EDH	READ BYTE from cassette, returned in A.

Miscellaneous

	003BH	PRINTER Send byte in A to printer.
	0060H	DELAY Waits for time in BC reg.
	06C0H	WARM START Entry point for BASIC warm start.
	3AB3H	PSG WRITE Byte in L sent to register H.
	3ABBH	PSG READ Byte in register H returned in A.
*	3A5EH	JOYSTICK Position of joystick in D returned in A.
*	3A87H	KEYPAD Keypad specified by D. Value returned in A.

* New ROM's only.

5.8.2 List of RAM locations.

Location	Function
400CH	BREAK key vector.
4012H	INTERRUPT vector.
4015H	Keyboard DCB.
4016H	Keyboard driver routine address.
4019H	Cursor control data (2 bytes).
401DH	Screen DCB.
401EH	Screen driver routine address.
4020H	Cursor position.
4023H	Current screen printing colour.
4025H	Printer DCB.
4026H	Printer driver routine address.
408EH	Entry point to USR routine.
40A0H	String space pointer.
40A4H	BASIC program pointer.
40B1H	Top of memory pointer.
40DFH	SYSTEM tape entry pointer.
40F9H	Simple variable pointer.

40FBH Array space pointer.
41E2H Exit vector for SYSTEM command.
41E8H Start of input buffer.
42F0H Register values for CRTC, text mode.
4300H Register values for CRTC, hi res mode.
431CH Control port value.
4400H Start of text screen memory.
4800H Start of hi res screen memory.
5801H Normal start of BASIC program.

5.8.3 List of BASIC reserved word tokens.

Word	Decimal	Hex	Word	Decimal	Hex	Word	Decimal	Hex
ABS	217	D9	FRE	218	DA	RESTORE	144	90
AND	210	D2	GOSUB	145	91	RESUME	159	9F
ASC	246	F6	GOTO	141	8D	RETURN	146	92
ATN	228	E4	IF	143	8F	RIGHT$	249	F9
AUTO	183	B7	INKEY$	201	C9	RND	222	DE
BGRD	255 152	FF 98	INP	219	DB	RUN	142	8E
CALL	255 150	FF 96	INPUT	137	89	SCALE	255 138	FF 8A
CDBL	241	F1	INT	216	D8	SET	131	83
CHAR	255 146	FF 92	JOY	255 131	FF 83	SGN	215	D7
CHR$	247	F7	KEYPAD	255 130	FF 82	SHAPE	255 139	FF 8B
CINT	239	EF	LEFT$	248	F8	SIN	226	E2
CIRCLE	255 137	FF 89	LEN	243	F3	SOUND	255 145	FF 91
CLEAR	184	B8	LET	140	8C	SQR	221	DD
CLOAD	185	B9	LGR	255 134	FF 86	STEP	204	CC
CLS	132	84	LIST	180	B4	STOP	148	94
COLOUR	255 128	FF 80	LLIST	181	B5	STR$	244	F4
CONT	179	B3	LOG	223	DF	STRING$	196	C4
COS	225	E1	LPRINT	175	AF	SWAP	255 148	FF 94
CPOINT	255 143	FF 8F	MEM	200	C8	SYSTEM	174	AE
CSAVE	186	BA	MID$	250	FA	TAB(	188	BC
CSNG	240	F0	NBGRD	255 153	FF 99	TAN	227	E3
DATA	136	88	NEW	187	BB	THEN	202	CA
DEFDBL	155	9B	NEXT	135	87	TO	189	BD
DEFINT	153	99	NOT	203	CB	TROFF	151	97
DEFSNG	154	9A	NPLOT	255 144	FF 90	TRON	150	96
DEFSTR	152	98	NSHAPE	255 140	FF 8C	USING	191	BF
DELETE	182	B6	ON	161	A1	USR	193	C1
DIM	138	8A	OR	211	D3	VAL	245	F5
EDIT	157	9D	OUT	160	A0	VARPTR	192	C0
ELSE	58 149	3A 95	PAINT	255 142	FF 8E	VERIFY	255 151	FF 97
END	128	80	PEEK	229	E5	XSHAPE	255 141	FF 8D
ERL	194	C2	PLAY	255 136	FF 88	+	205	CD
ERR	195	C3	PLOT	255 132	FF 84	-	206	CE
ERROR	158	9E	POKE	177	B1	*	207	CF
EXP	224	E0	POS	220	DC	/	208	D0
FCLS	255 135	FF 87	PRINT	178	B2	↑	209	D1
FCOLOUR	255 129	FF 81	RANDOM	134	86	<	212	D4
FGR	255 133	FF 85	READ	139	8B	=	213	D5
FIX	242	F2	REM	147	93	>	214	D6
FKEY	255 149	FF 95	RENUM	255 147	FF 93	'	58 147	3A 93
FOR	129	81	RESET	130	82		251	FB

Additional reserved words for disk BASIC

CLOSE	166	A6	GET	164	A4	MKD$	238	EE
CMD	133	85	INSTR	197	C5	MKI$	236	EC
CVD	232	E8	KILL	170	AA	MKS$	237	ED
CVI	230	E6	LINE	156	9C	NAME	169	A9
CVS	231	E7	LOAD	167	A7	OPEN	162	A2
DEF	176	B0	LOC	234	EA	PUT	165	A5
EOF	233	E9	LOF	235	EB	RSET	172	AC
FIELD	163	A3	LSET	171	AB	SAVE	173	AD
FN	190	BE	MERGE	168	A8	TIMES$	199	C7

Section 6. Component lists.

Circuit Designation	Component Type
Integrated Circuits.	
Z1	74LS02
Z2	74LS04
Z3	74LS86
Z4	74LS32
Z5	74LS74
Z6	74LS00
Z7	74LS32
Z8	74LS374
Z9	74LS00
Z10	74LS74
Z11	74LS08
Z12 - Z15	4116 RAM
Z16	74LS374
Z17	74LS173
Z18	74LS273
Z19	74LS157
Z20 - Z23	4116 RAM
Z24	74LS368
Z25	74LS174
Z26	74LS245
Z27	2716 C.GEN
Z28	74LS166
Z29	74LS367
Z30 & Z31	2114 RAM
Z32	74LS08
Z33 & Z34	74LS368
Z35	74LS254
Z36	74LS367
Z37 & Z38	74LS166
Z39	2114 RAM
Z40	74LS138
Z41	74LS20
Z42 & Z43	74LS157
Z44	74LS00
Z45 & Z46	ROMS
Z47	Z80 CPU
Z48	74LS08
Z49 & Z50	74LS157
Z51	74LS10
Z52	74LS367
Z53	74LS32
Z54	74LS132
Z55 & Z56	74LS139
Z57	MC 6845 CRTC
Z58 & Z59	74LS157
Z60	AY-3-8910 PSG
Z61 & Z62	LM747
Z63	74LS74
Z64	74LS27
Z65	74LS161
Z66	74LS04

Circuit Designation	Component Type
Semiconductors.	
Q1	BC327
Q2 - Q4	BC337
Z1	BZY88 5V6
Z2 - Z4	BZY88 6V2
D1 - D9	1N60
D10 & D11	1N4148
D12 - D15	1N60
D16 & D17	1N4148
D18 & D19	1N60
Resistors. 1/4 W 5 %	
R1	560R
R2	120R
R3	2K7
R6	10R
R7	820R
R8	2K2
R9	68K
R10	2K2
R11	22K
R12	100R 1/2 W
R13	2K2
R14 & R15	4K7
R16 & R17	10K
R18	1K0
R19	4K7
R20	1K0
R21	4K7
R22	8K2
R23	10K
R24	4K7
R25	39K
R26 - R34	4K7
R35 - R37	33R
R38	220R
R39	10K
R40	220R
R41	1K5
R42 & R43	560R
R44	33R
R45	4K7
R46 & R47	68K
R48	390R
R49	50K preset

Section 6. Component lists.

Capacitors.

Circuit Designation	Component Type
C1	150pF ceramic
C2	220pF ceramic
C3	0.1uF disc ceramic
C4	100uF 10V
C5 - C8	0.1uF disc ceramic
C9	10uF 16V
C10 - C21	0.1uF disc ceramic
C22 - C23	10uF 16V
C24 - C28	0.1uF disc ceramic
C29	10uF 16V
C30 - C34	0.1uF disc ceramic
C35	100pF ceramic
C36	0.1uF disc ceramic
C37	10uF 16V
C38 - C39	0.1uF disc ceramic
C40	10uF 16V
C41	100uF 10V
C42 - C57	0.1uF disc ceramic
C58	150pF ceramic
C59	47pF ceramic
C60	22pF trimmer
C61	100pF ceramic
C62 - C67	0.1uF disc ceramic
C68	150pF ceramic
C69	1nF ceramic
C70	33uF 16V
C71 & C72	1uF 50V

ROM board.

Z1 - Z4	2532 BASIC ROM
Z5	74LS139
C1 - C5	0.1uF disc ceramic

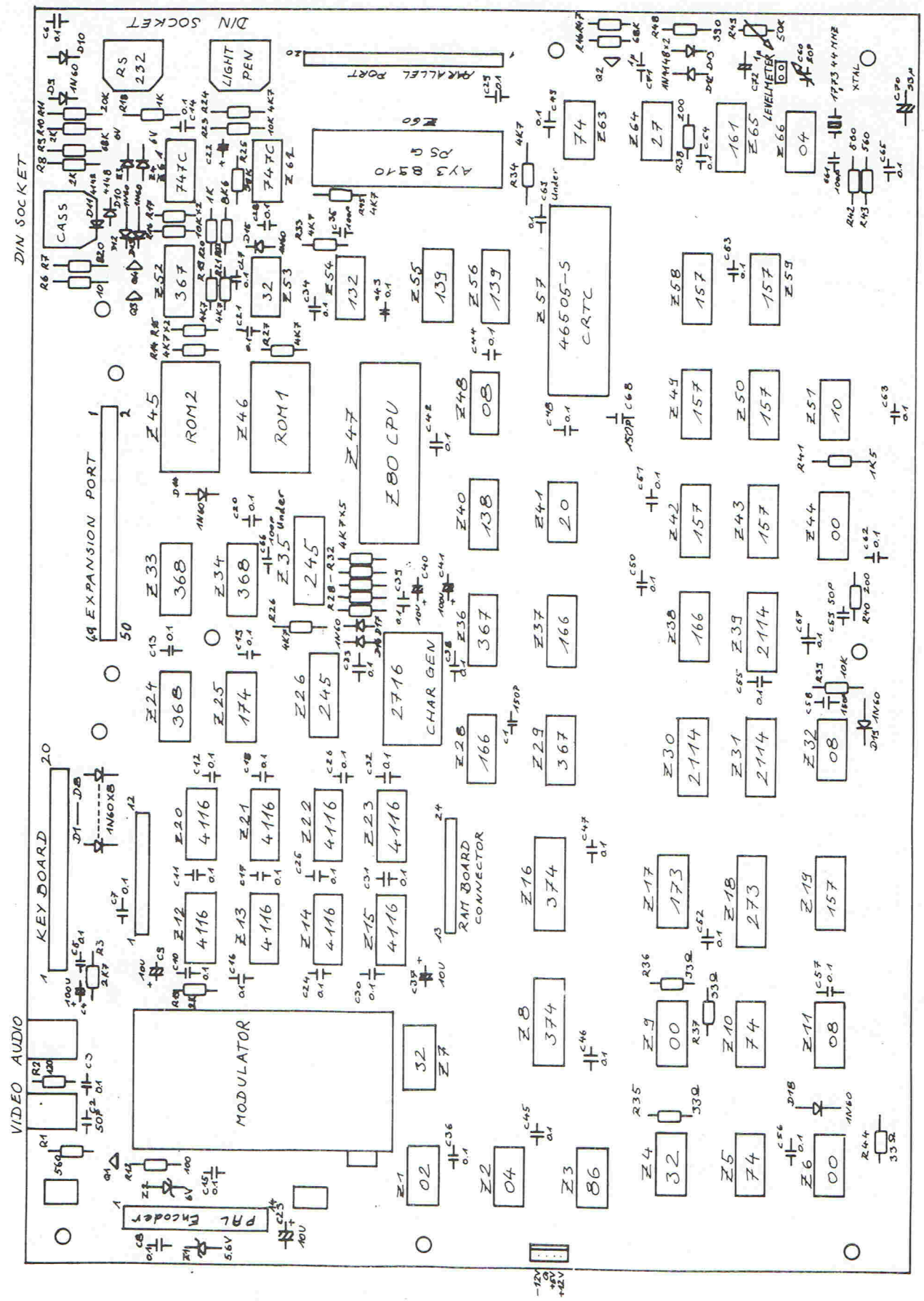
PAL colour encoder.

Z1	74LS74
Z2	74LS86
Z3 & Z4	74LS125
Z5	74LS02
Q1 - Q3	BC337
D1 & D3	1SS99
D2	1N60
L1 & L2	10uH
R2	2K7
R3	3K3
R4	2K7
R5 - R8	2K2
R9	3k3
R10	1K8
R11	56K
R12	3K3
R13	47R
R14 & R15	10K
C1 & C2	0.1uF disc ceramic
C3	47pF ceramic
C4	0.1uF disc ceramic
C5	22pF ceramic
C6	2.7nF ceramic
C7 - C9	0.1uF disc ceramic

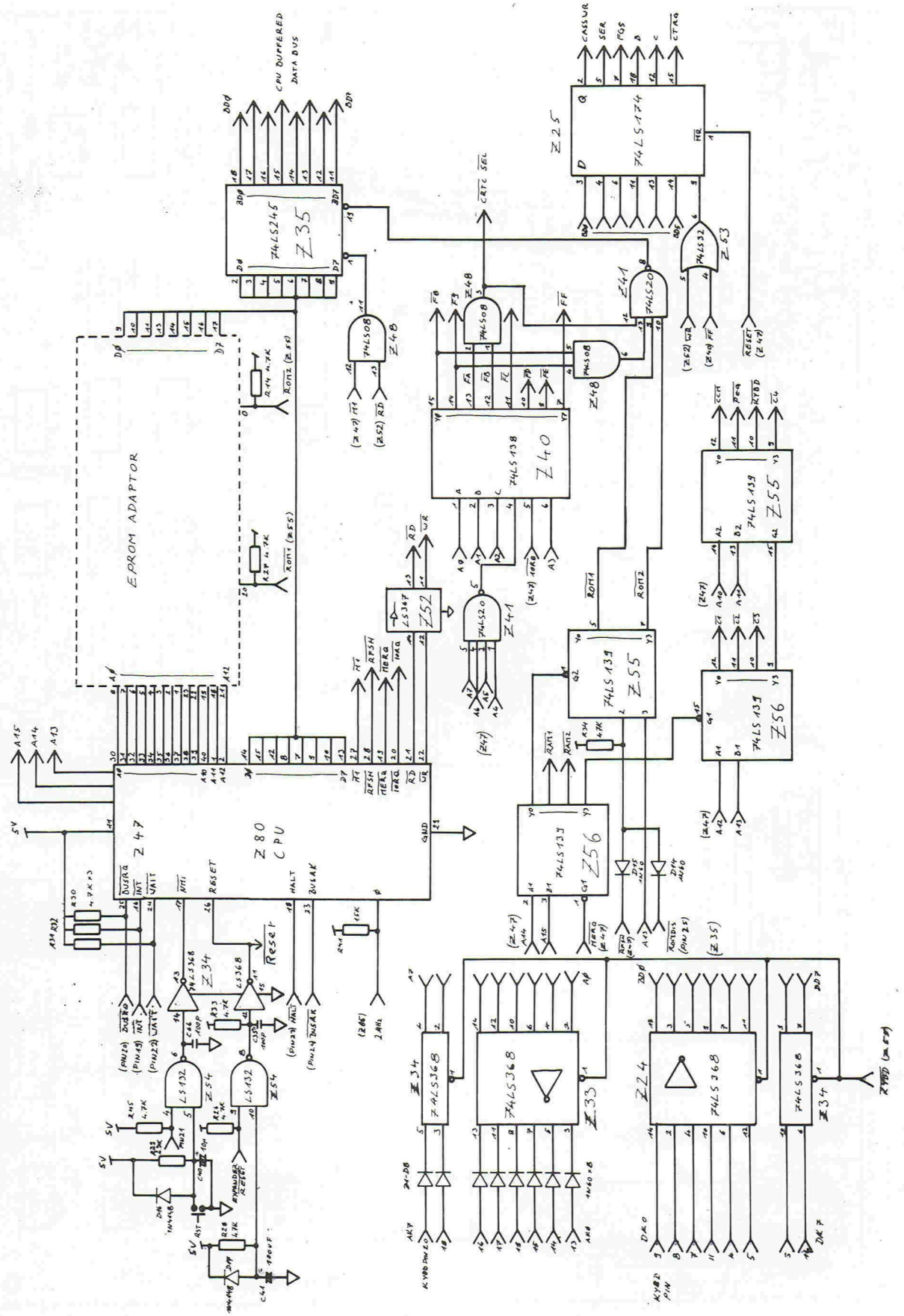
Power supply.

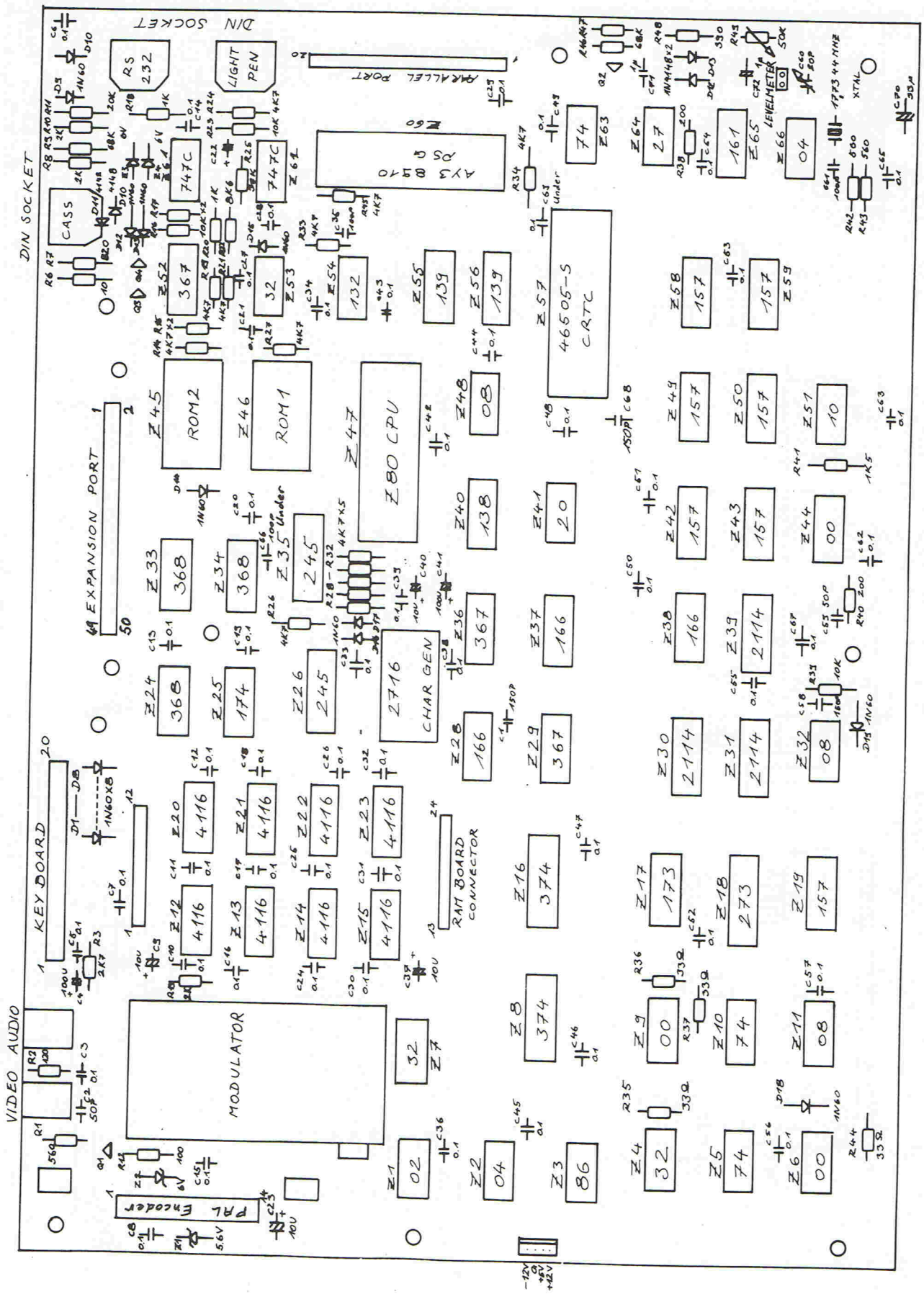
Z1	7805 regulator
Z2	7812 regulator
Z3	7912 regulator
Q1	MJ2955
G1	4A 50V bridge rect
G2	1A 50V bridge rect
R1	4R7
C1 - C3	0.1uF disc ceramic
C4	10 000uF 16V
C5	220uF 10V
C6	47uF 25V
C7 - C9	0.1uF disc ceramic
C10	2200uF 25V
C11	47uF 25V
C12	0.1uF disc ceramic
C13	2200uF 25V
C14	47uF 25V

Section 7. Circuit diagrams.

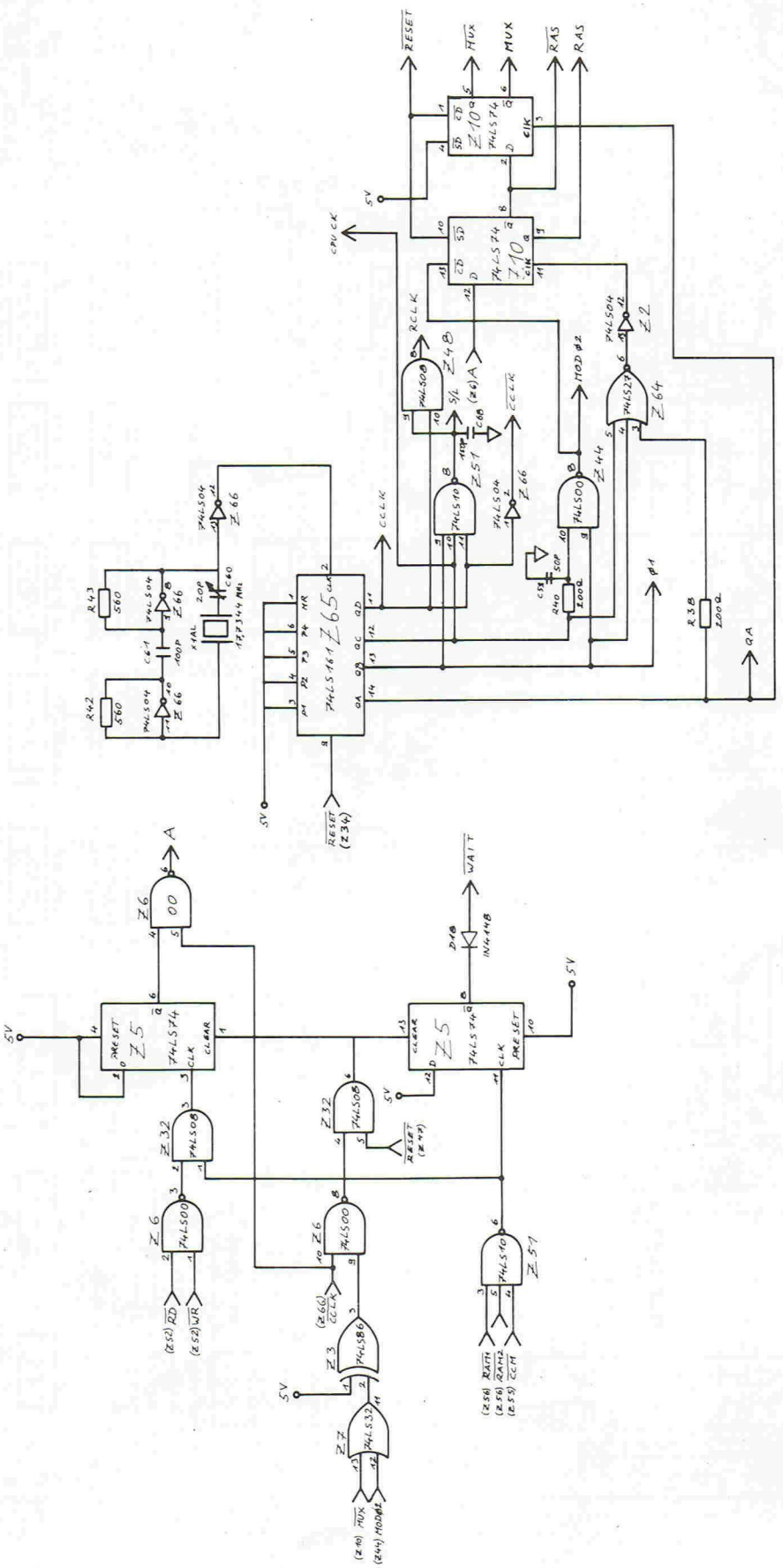


7.1 CPU and address decoding.

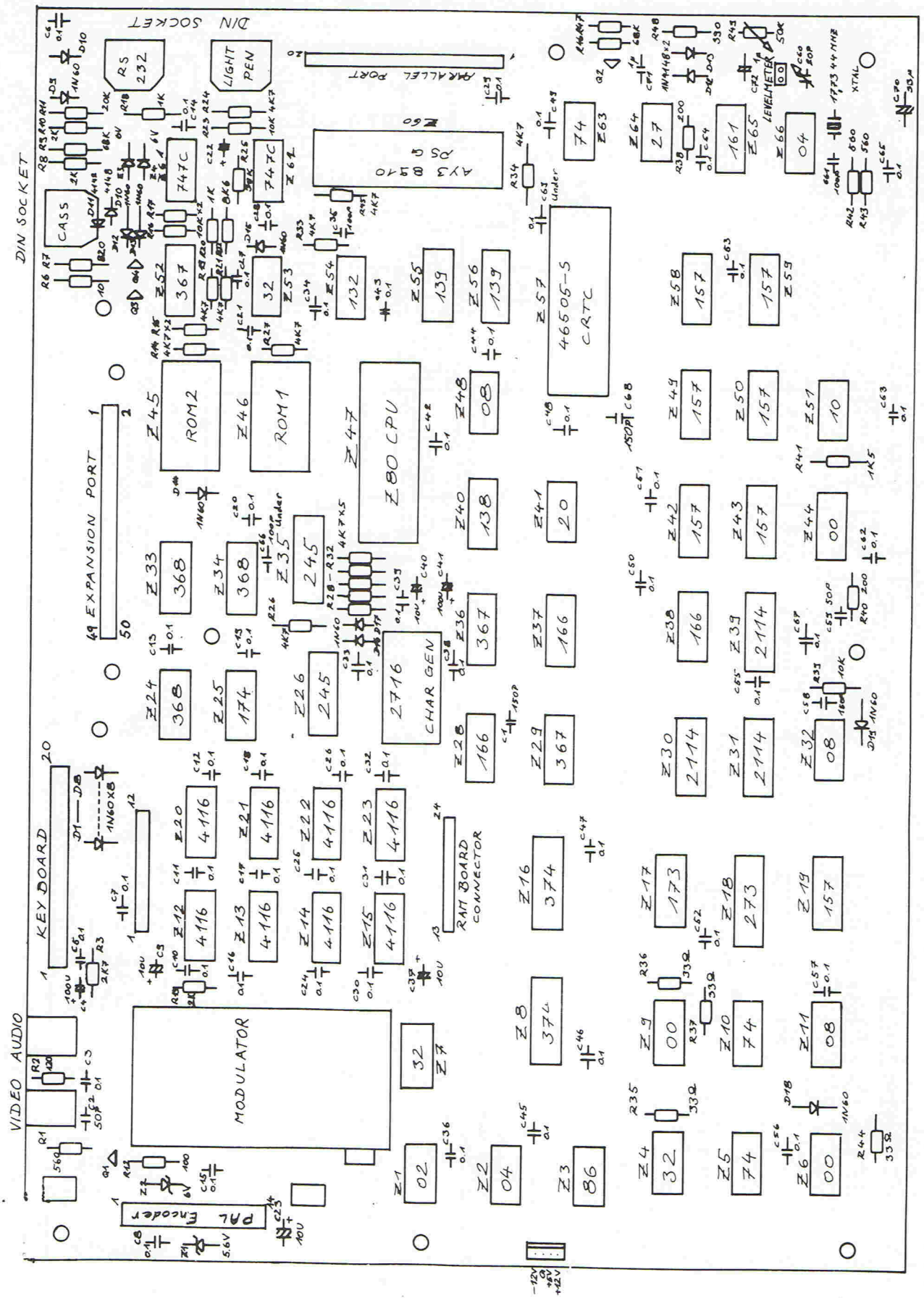



DIN SOCKET
CASS
RS 232
LIGHT PEN
PARALLEL PORT
EXPANSION PORT
ROM2
ROM1
Z80 CPU
CHAR GEN
MODULATOR
KEY BOARD
RAM BOARD CONNECTOR
PAL Encoder
VIDEO AUDIO
46505-S CRTC
LEVELMETER
XTAL

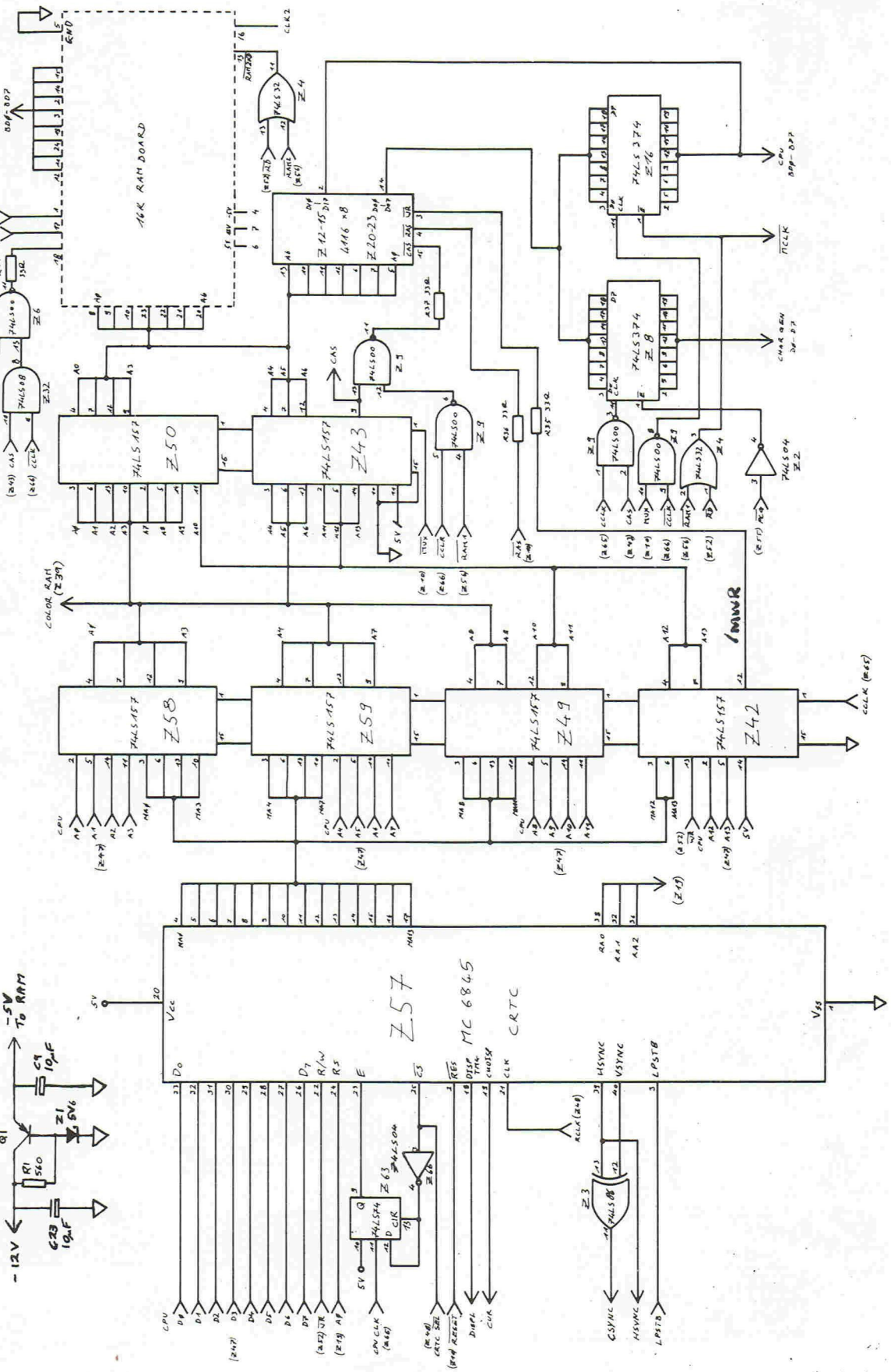
7.2 System timing.



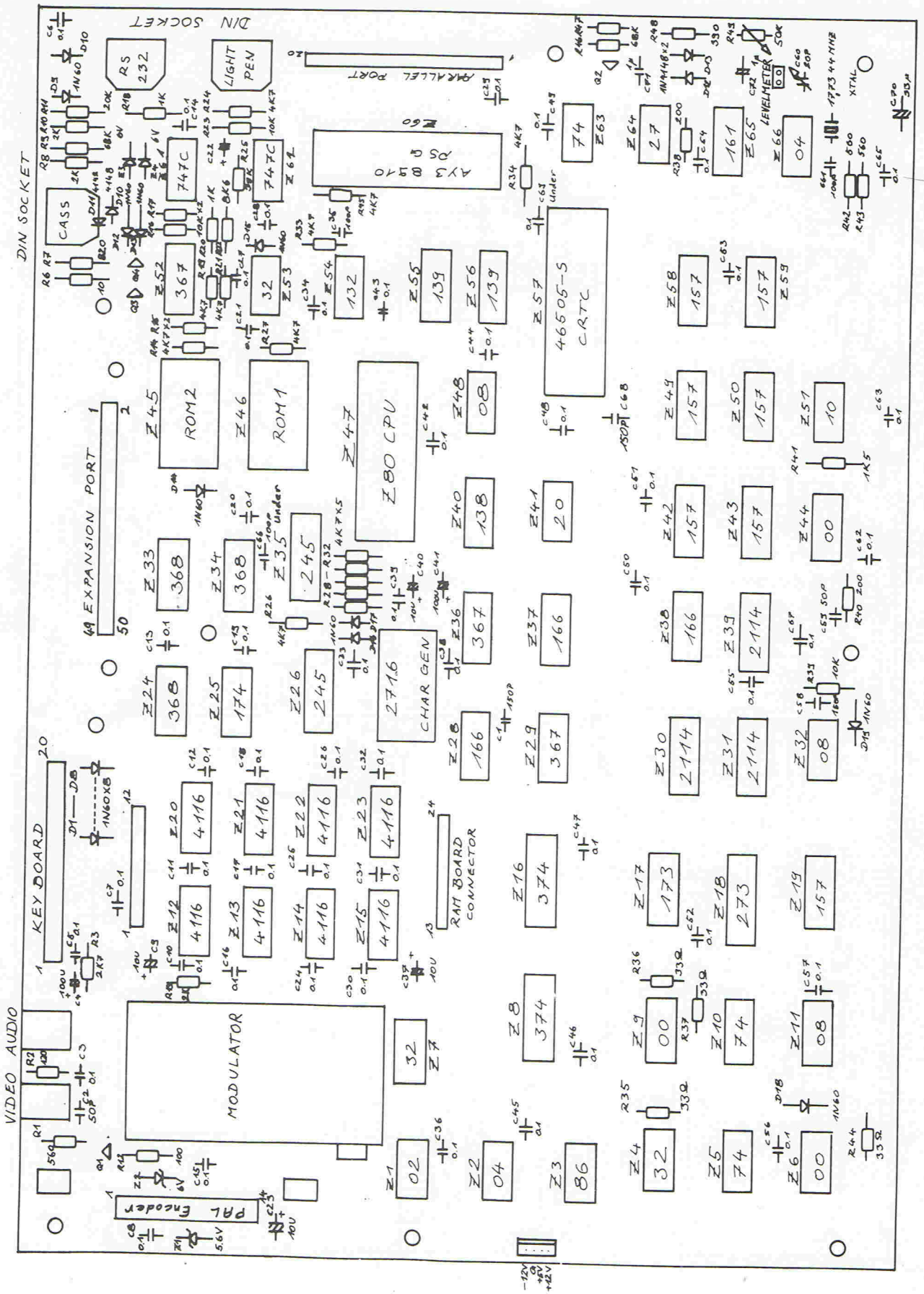
Section 7. Circuit diagrams.



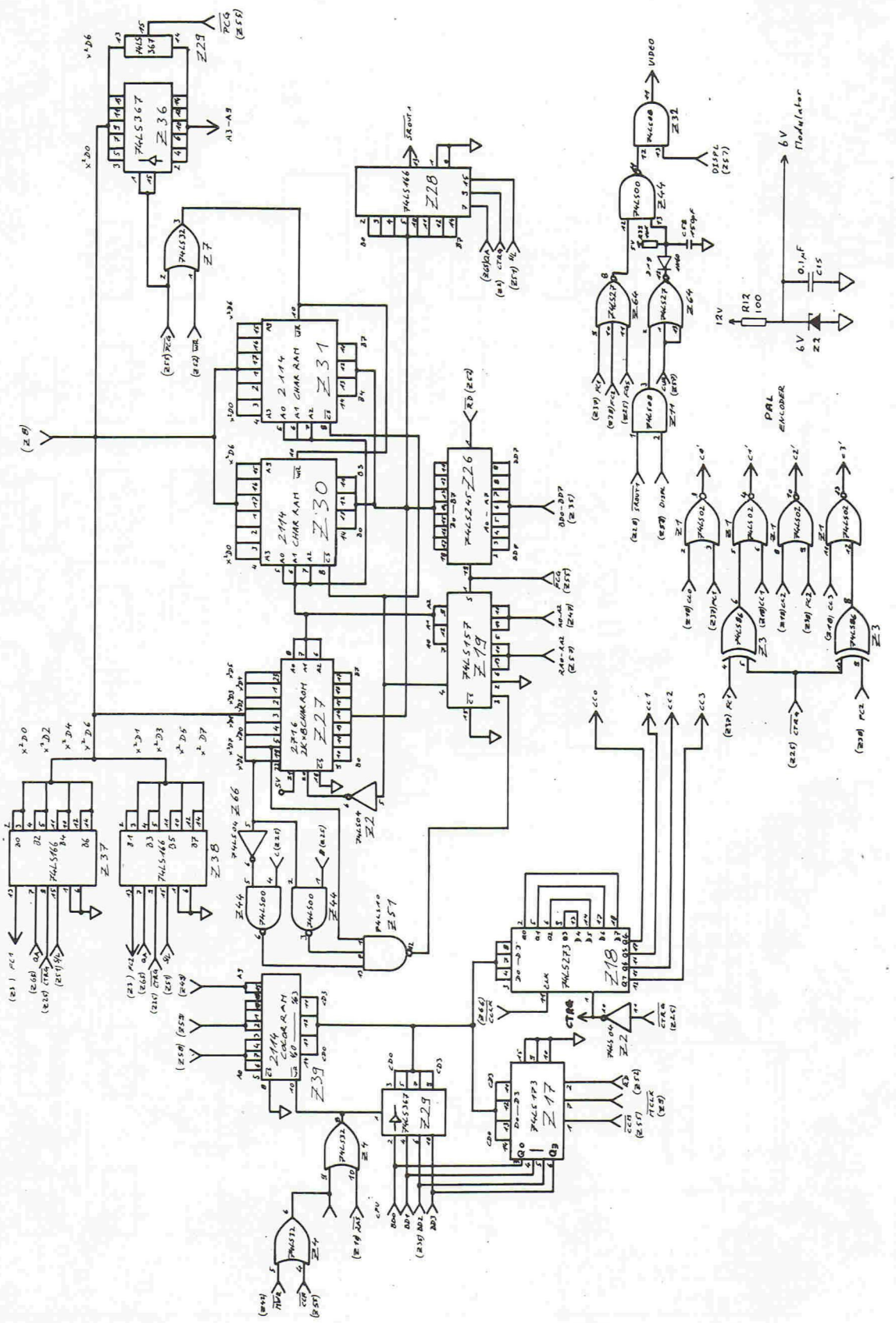
7.3 CRTC and system RAM.

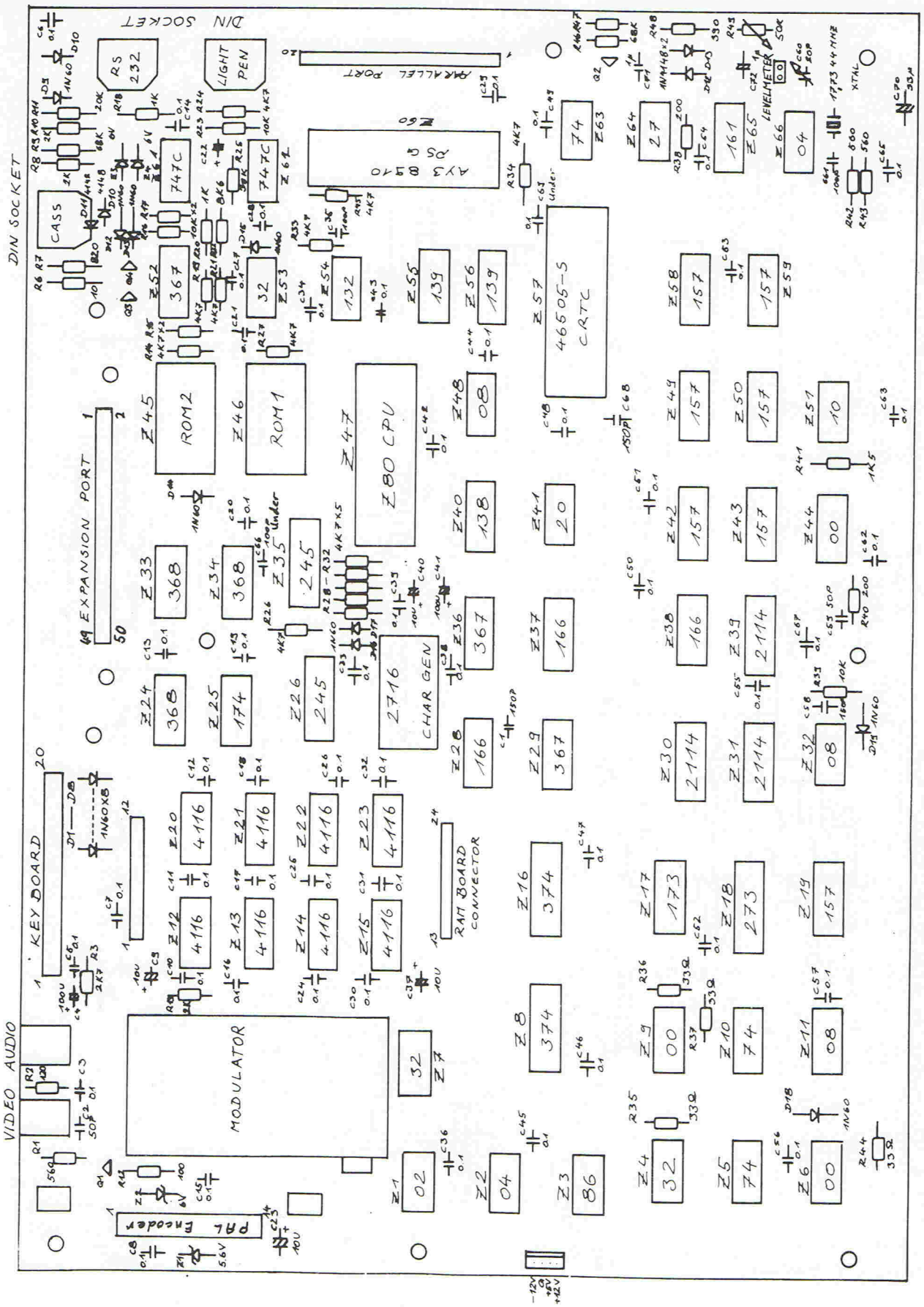


Section 7. Circuit diagrams.

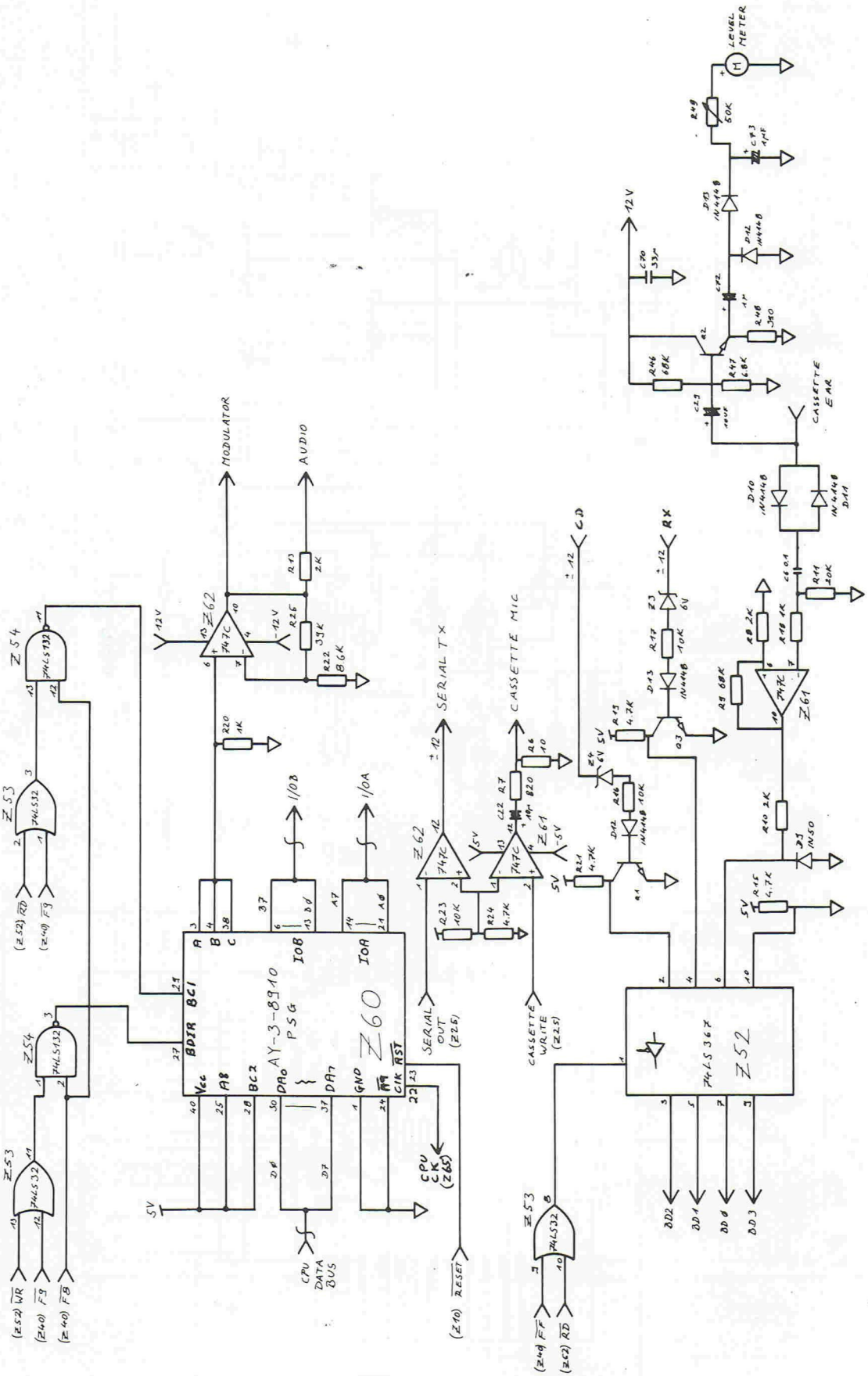


7.4 Character generator and colour RAM.

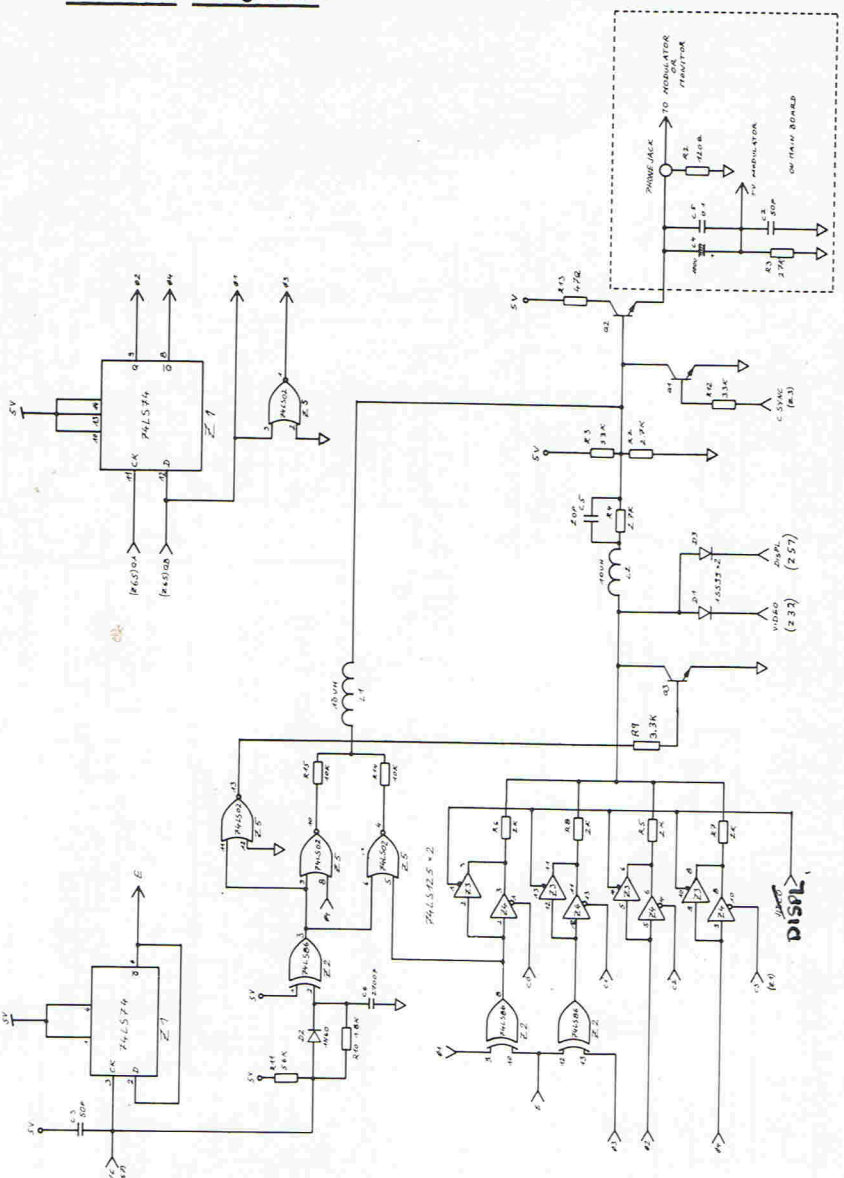


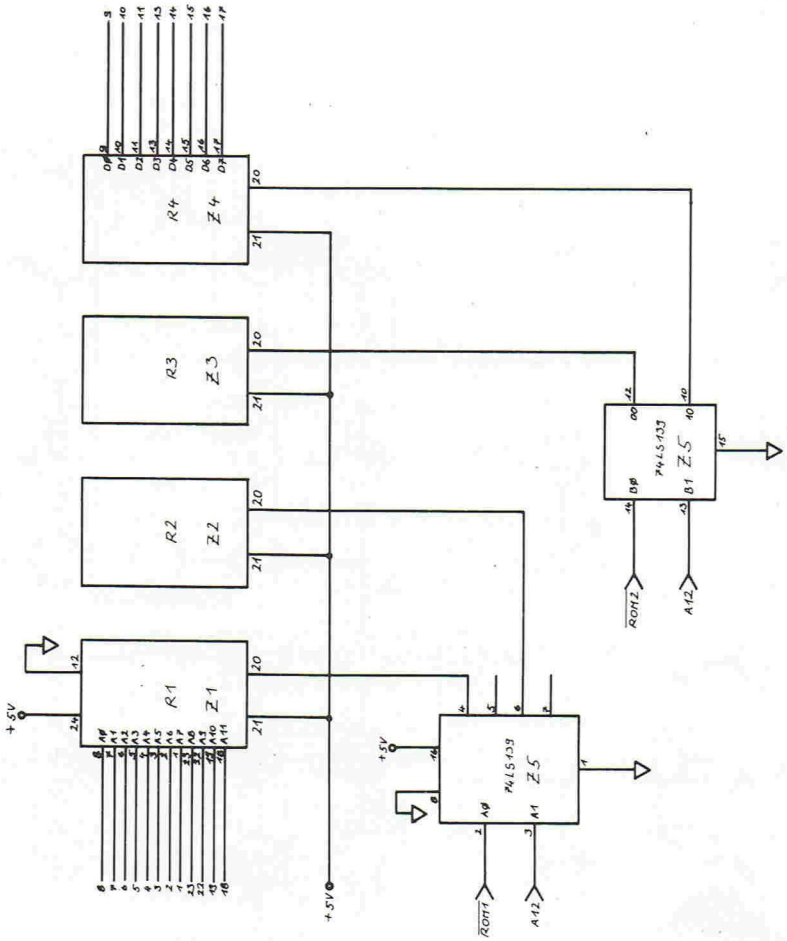

DIN SOCKET
RS 232
LIGHT PEN
CASS
PARALLEL PORT
EXPANSION PORT
KEY BOARD
MODULATOR
RAM BOARD CONNECTOR
CHAR GEN
Z80 CPU
ROM2
ROM1
4605-S CRTC
LEVELMETER
VIDEO AUDIO
DIN SOCKET

7.5 PSG, serial and cassette interfaces.

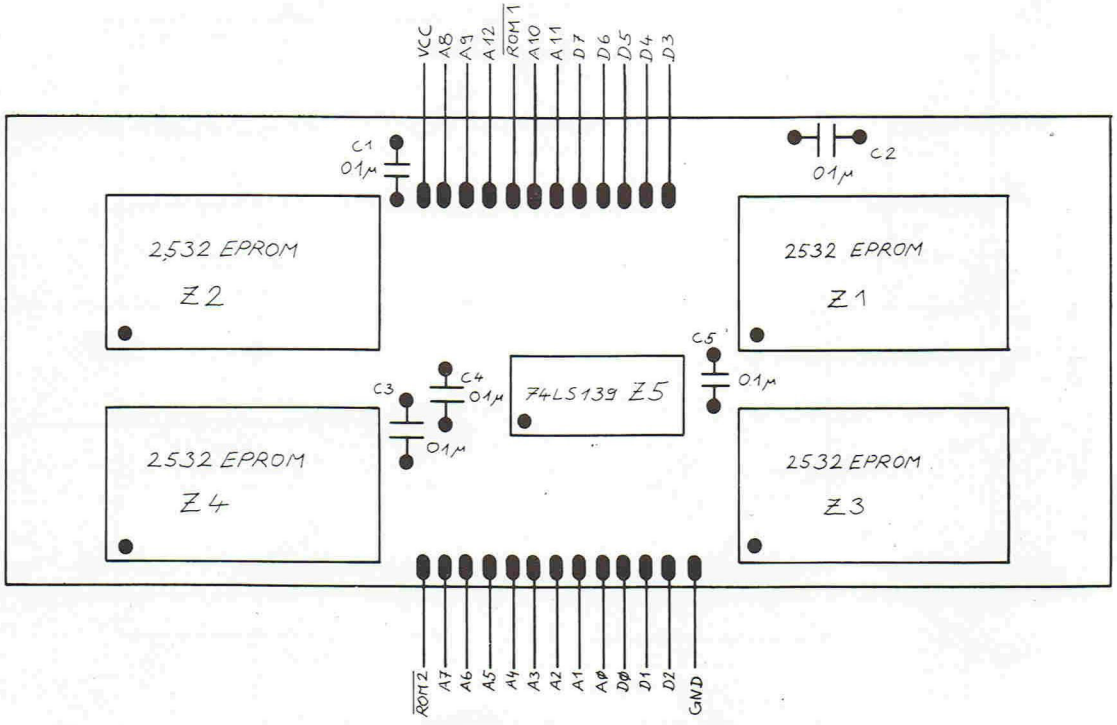


7.6 PAL colour encoder.

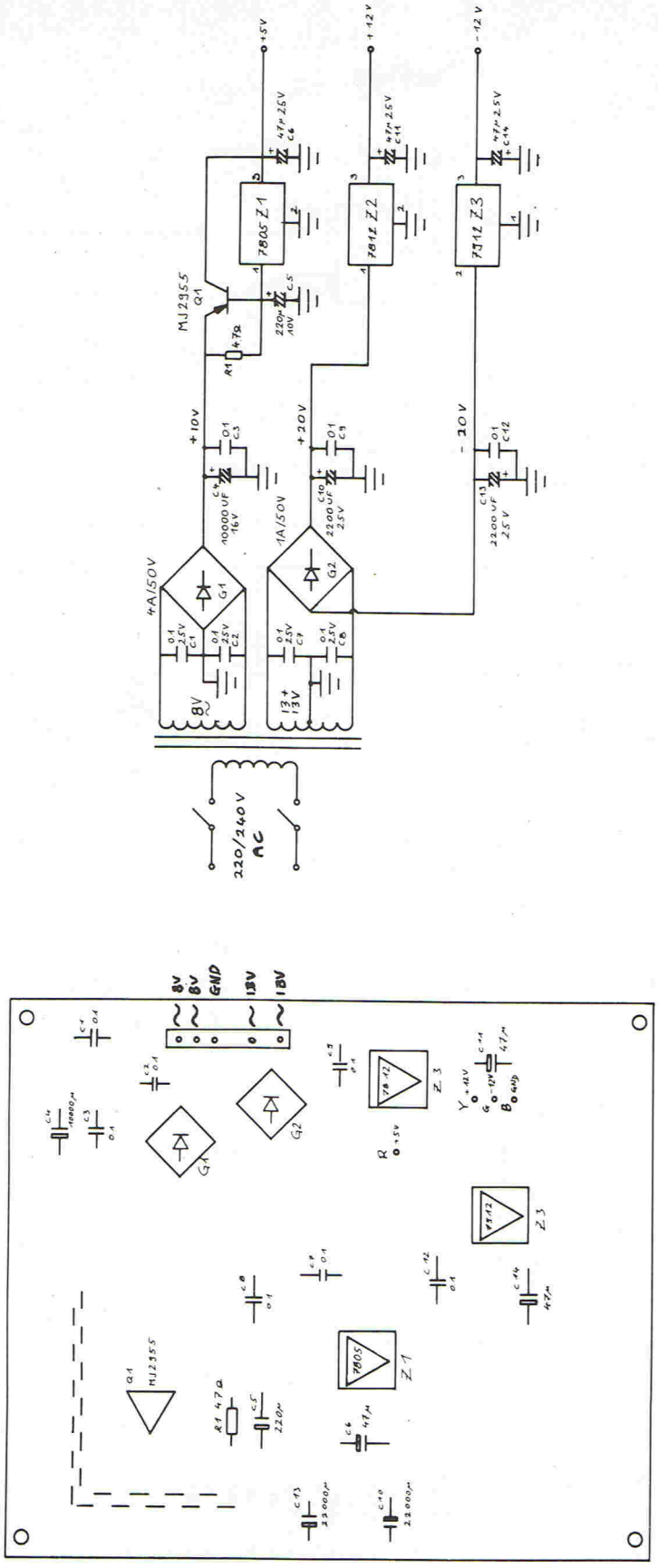



R4 Z4
R3 Z3
R2 Z2
R1 Z1
74LS139 Z5
+5V
ROM2
A12
ROM1

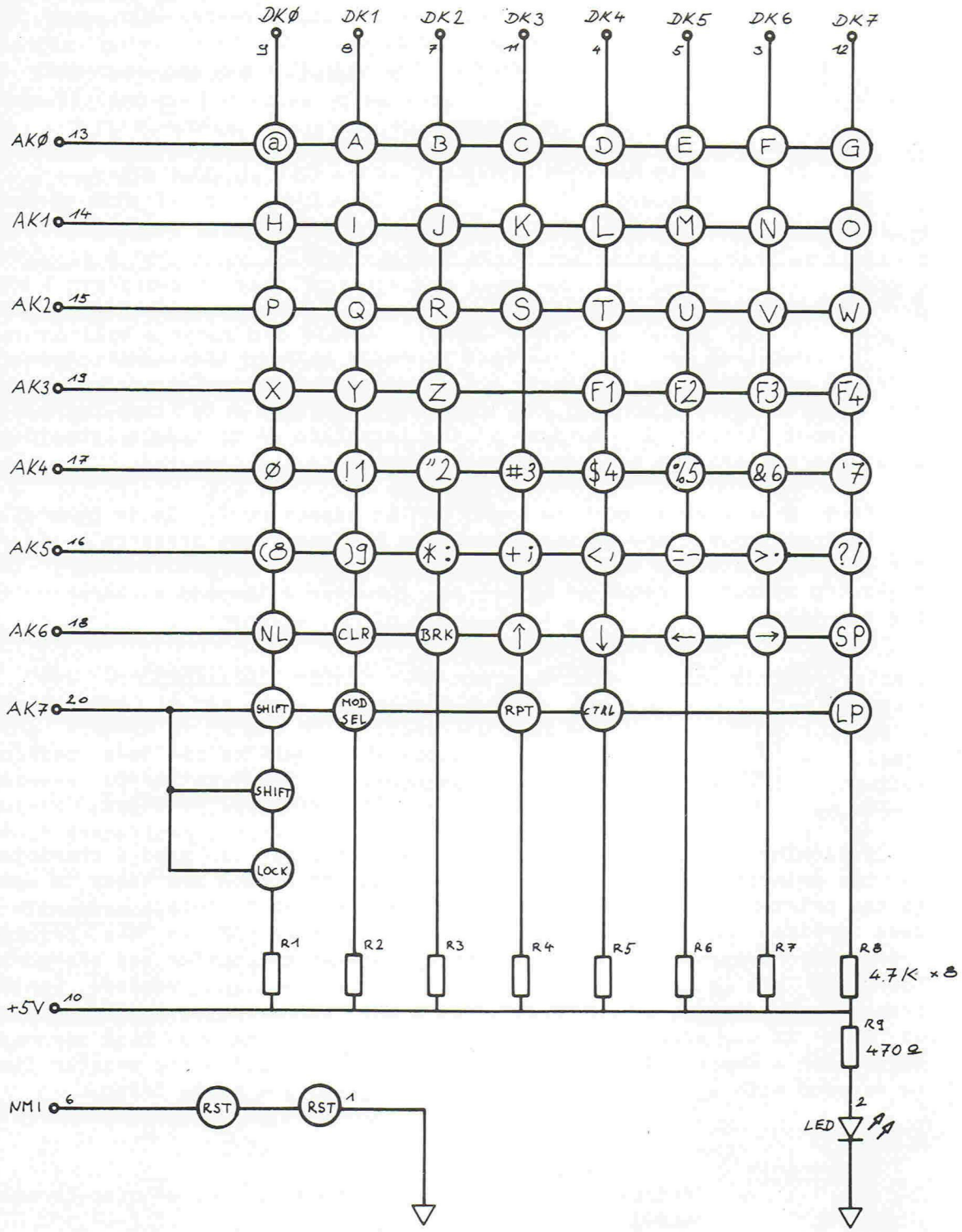
7.7 ROM board.


VCC A8 A9 A12 ROM1 A10 A11 D7 D6 D5 D4 D3
C1 01µ
C2 01µ
2532 EPROM Z2
2532 EPROM Z1
C3 01µ
C4 01µ
C5 01µ
74LS139 Z5
2532 EPROM Z4
2532 EPROM Z3
ROM2 A7 A6 A5 A4 A3 A2 A1 A0 D0 D1 D2 GND

7.8 Power supply.



7.9 Keyboard.


DKØ
DK1
DK2
DK3
DK4
DK5
DK6
DK7
9
8
7
11
4
5
3
12
AKØ 13
AK1 14
AK2 15
AK3 19
AK4 17
AK5 16
AK6 18
AK7 20
@ A B C D E F G
H I J K L M N O
P Q R S T U V W
X Y Z F1 F2 F3 F4
Ø !1 "2 #3 $4 %5 &6 '7
(8)9 *: +; <, =- >. ?/
NL CLR BRK ↑ ↓ ← → SP
SHIFT MOD SEL RPT CTRL LP
SHIFT
LOCK
R1 R2 R3 R4 R5 R6 R7 R8
4.7K x8
+5V 10
R9
470Ω
2
LED
NMI 6
RST
RST
1

Section 8. Accessories.

8.1 Additional RAM card.

The RAM card is fitted to the main PCB by two twelve-way connectors, which supply signals and power to the RAM card. The memory on the card consists of eight 4116-type dynamic RAM's, supplemented with two TTL chips - an address buffer and a data latch. All the timing signals required by this card are generated on the main PCB and are described in section 4.

Components.

Z1 - Z8	4116 RAM	C1 - C3	33uF 16V
Z9	74LS374	C4 - C16	0.1uF disc ceramic
Z10	74LS244		

8.2 Parallel printer interface.

The parallel printer interface connects between the Colour Genie's parallel port and a printer with a Centronics-type interface. Data lines are passed straight through the interface from port A to the printer's data input lines. The purpose of the interface is to handle the data strobe to the printer and handshaking signals to the computer.

Port B is used as both an input and an output port. It is generally set to input mode, and in that condition the interface presents a four-bit printer status on IOB4-IOB7, which can be read by the computer. The following status is required before the computer will send a character to the printer:-

Status bit	State	Signal description	
IOB4	1	Printer ready	(true)
IOB5	1	Unit selected	(true)
IOB6	0	Out of paper	(false)
IOB7	0	Printer busy	(false)

Following the acceptance of status, the computer can send a character to the printer. The required data is output to port A and hence is sent to the printer's data input. The B port is then set to output mode with a zero written into IOB1. This closes the buffer Z1 in the printer interface, preventing level contention between the buffer and the port. The IOB0 bit is then toggled in a high - low - high sequence, which triggers monostables in Z3. This sends a data strobe pulse to the printer so that it can accept the data presented to it. The busy line is made active for a short time by the other half of Z3 to allow the printer time to respond with a busy state, and prevent characters being lost.

Components.

Z1	74LS367	C3 & C4	0.1uF disc ceramic
Z2	74LS02		
Z3	74LS123	R1 - R3	3K9
		R4	22K
C1	22nF ceramic	R5	100K
C2	220pF ceramic		

8.3 Joysticks.

The joystick and keypad interface connects to the parallel port on the Colour Genie, and provides two numeric keypads and two analogue joysticks.

The keypads are connected in matrix form between port A (used as an output) and port B (used as an input). Each keypad has 12 keys arranged in a 3 by 4 matrix, together forming a 6 by 4 matrix. Outputs IOA0 to IOA5 are taken low in turn, and inputs IOB0 to IOB3 are used to sense the keys that are pressed.

The A port also drives a 6-bit digital-to-analogue converter, that produces a reference voltage against which the voltages produced by the X and Y positions of each joystick are compared. To determine the position of a particular joystick, a number is sent from port A, and the appropriate B input bit sensed. If the input bit is low then the number is higher than the joystick position. If the bit is high then the number is lower. The software in the ROM uses a binary chop search to find the joystick position in the minimum number of operations. Port B bits IOB4 to IOB7 are used for the comparison inputs as follows:-

Bit	Function
IOB4	Joystick 2 Y
IOB5	Joystick 2 X
IOB6	Joystick 1 Y
IOB7	Joystick 1 X

The D-A converter uses an R-2R resistor ladder to produce a voltage proportional to the number presented to it. Z1 is a CMOS buffer, providing outputs at 0V and 5V to drive the ladder. The resulting voltage is shifted and scaled in Z2 so that numbers 0 to 63 produce a voltage between 0V and 4V. This is compared with the output from each joystick potentiometer in a quad comparator Z3. The joystick pots are fed from a zener stabilized supply, ie R19 and Z1.

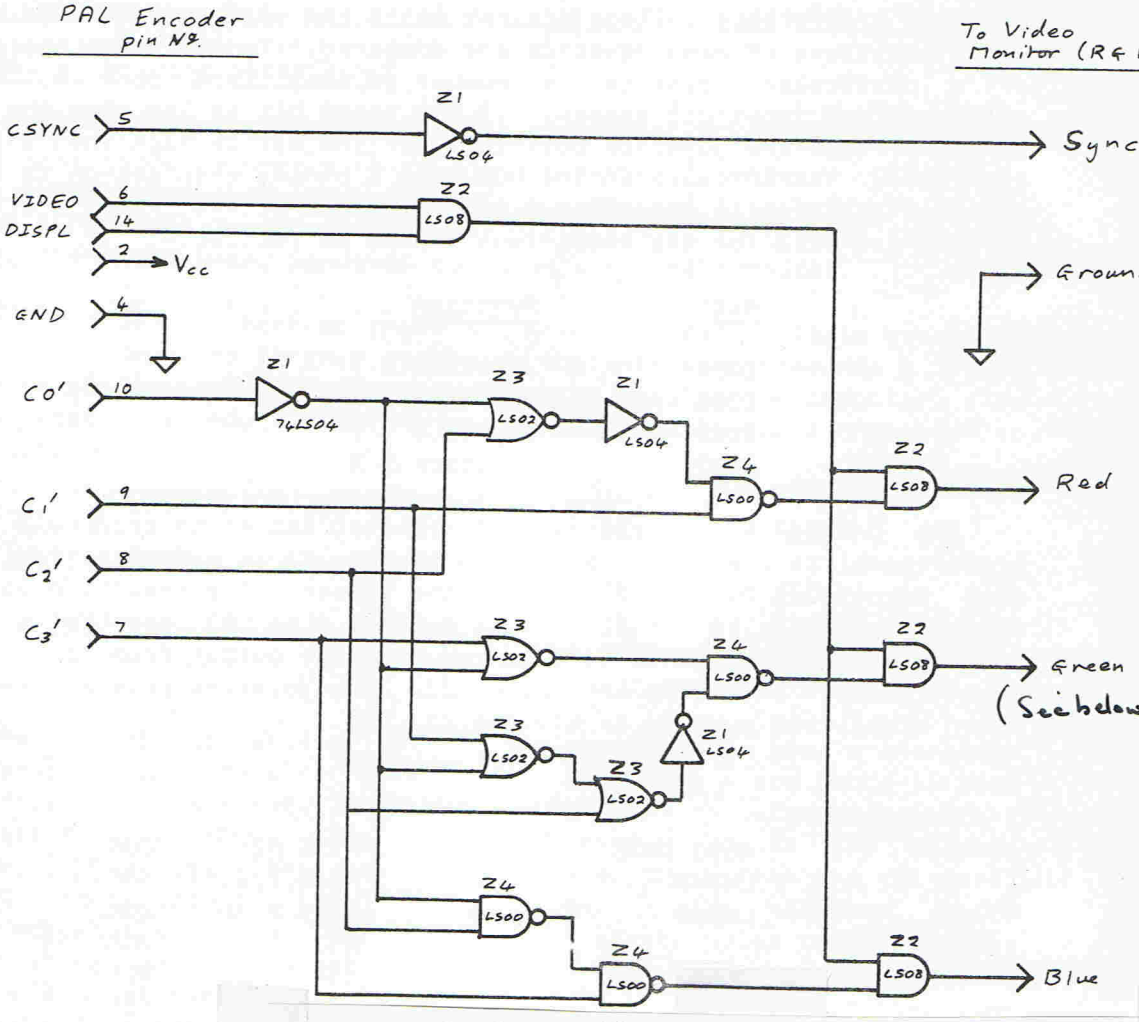
Components.

Z1	4050 CMOS	R1 & R2	20K
Z2	LM747	R3,5,7,9,11	10K
Z3	LM339	R4,6,8,10,12	20K
		R13	20K
D1 - D6	1N60	R14	30K
		R15	15K
C1 - C5	0.1uF disc ceramic	R16	4K7
		R17	12K
Z1	BZY88 4V3	R18	8K2
		R19	75R

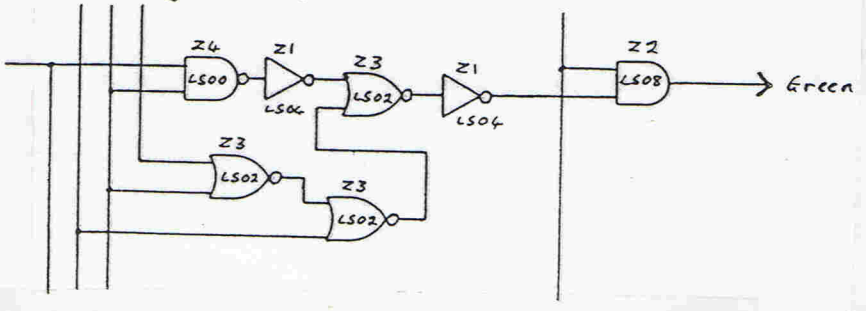
8.4 RGB output circuit.

The colour information supplied to the PAL encoder board can be used to derive red, green and blue colour signals that can be fed into an RGB colour monitor. With only 3 colour lines it is not possible to have more than 8 colours, so the circuit shown below attempts to produce the nearest colour on the RGB monitor to that which would display on a colour television.

All connections to this circuit can be taken from the connector between the main circuit board and the PAL encoder, and connections are referenced to this with pin 1 at the rear and pin 14 at the front.

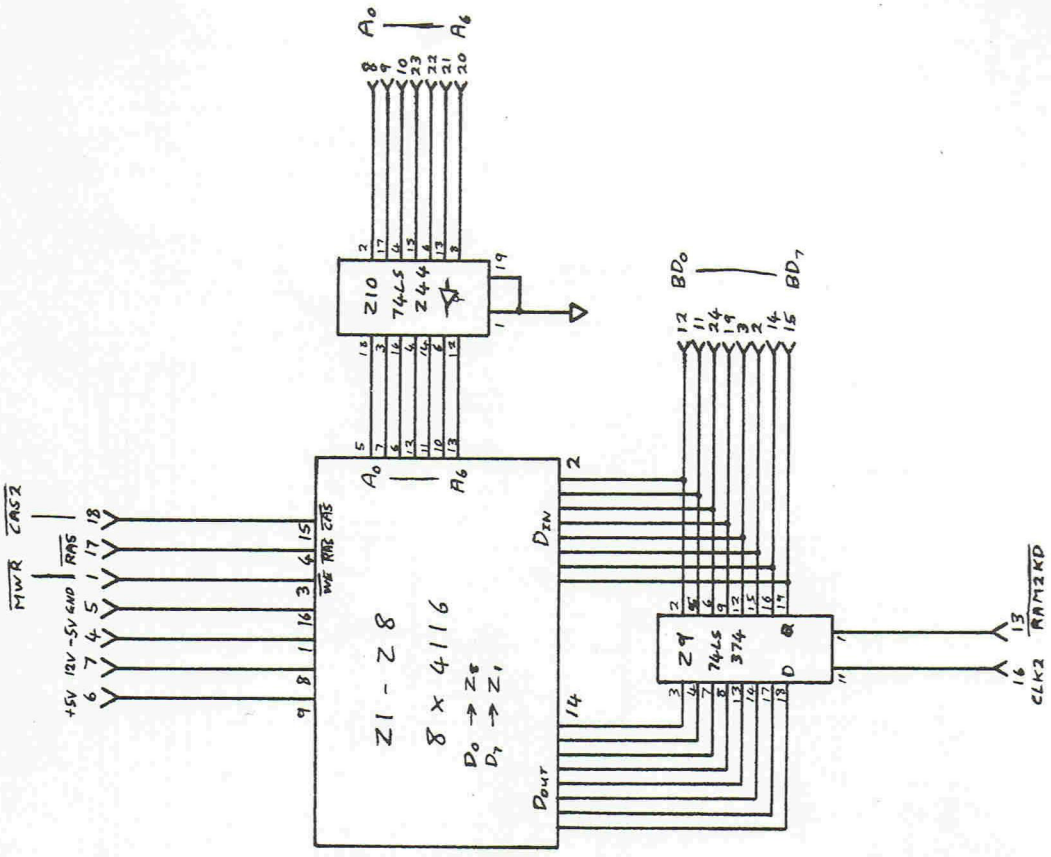


Correct circuit for green output.

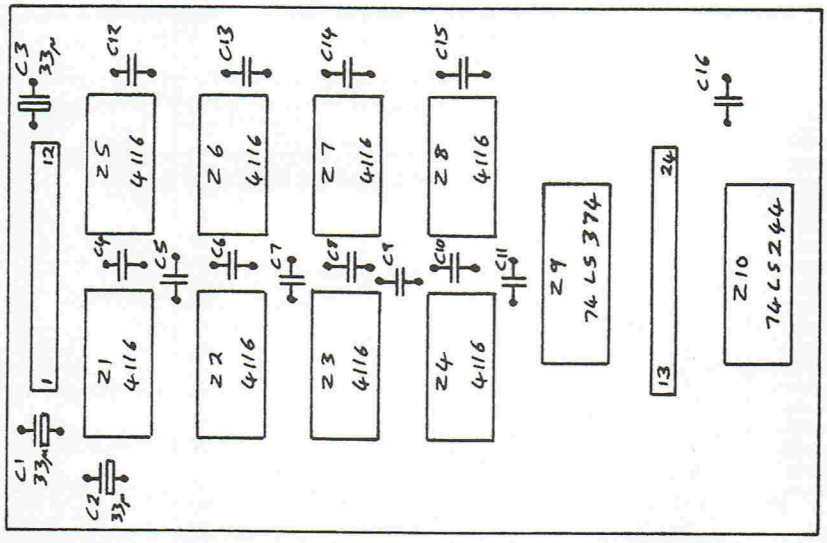


Components.

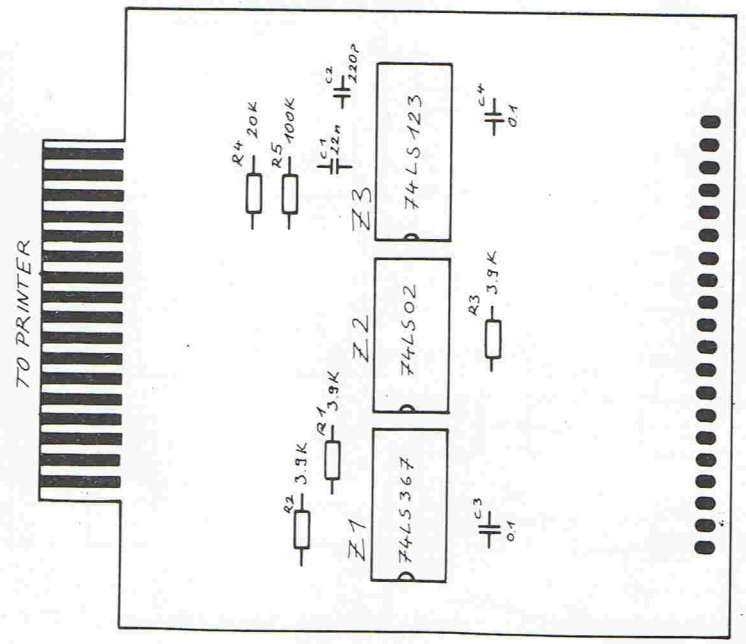
Z1	74LS04
Z2	74LS08
Z3	74LS02
Z4	74LS00

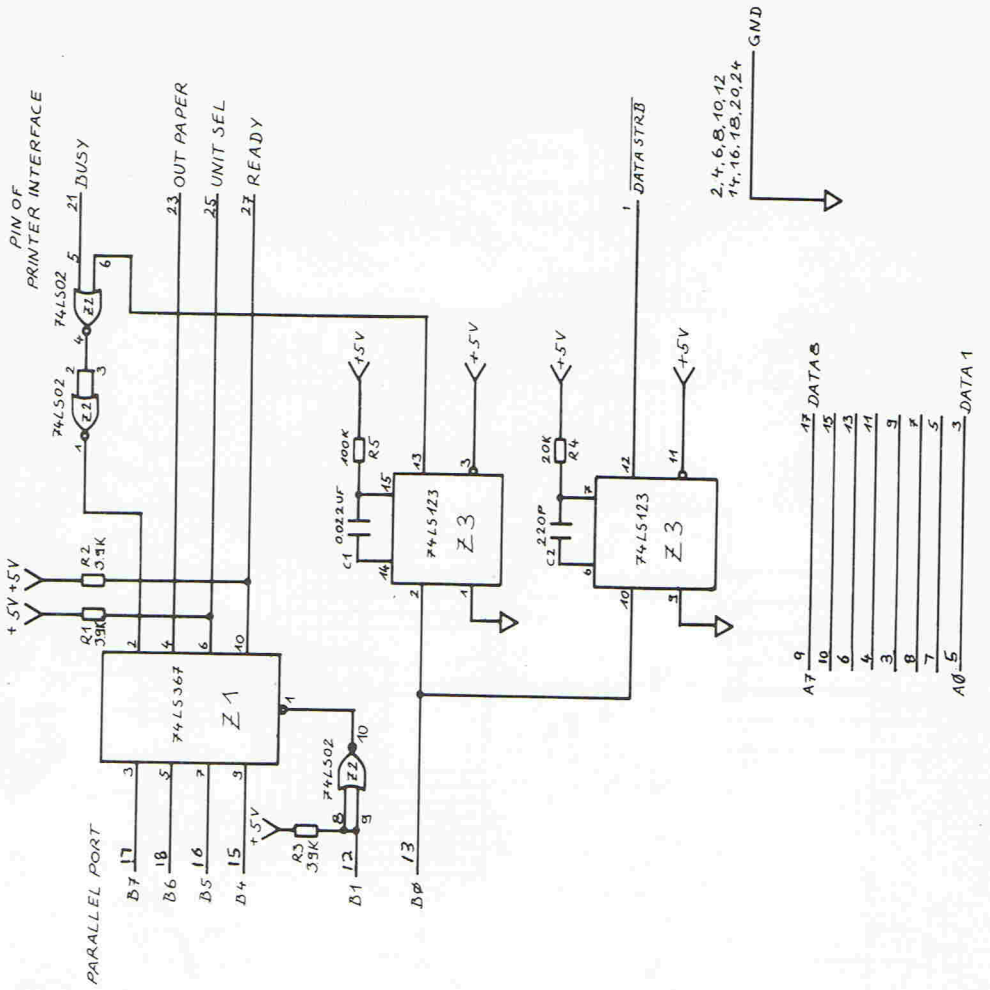

Z1 - Z8
8 x 4116
Z10
74LS
244
Z9
74LS
374
MWR
CAS2
RAS
+5V
12V
-5V
GND
WE
RAS
CAS
DIN
DOUT
BD0
BD7
RAM2KD
CLK2

16K RAM card.

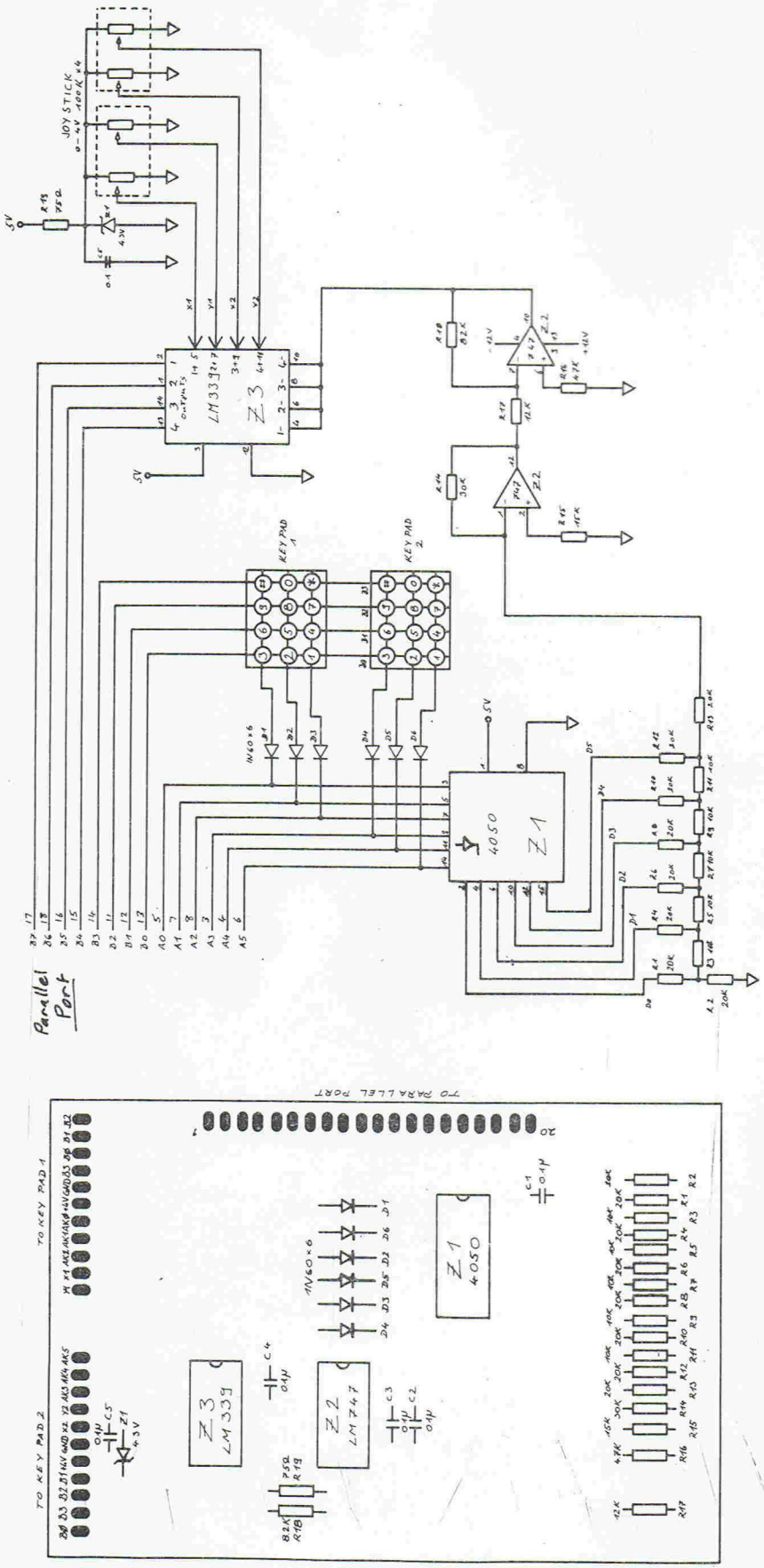

Z1
4116
Z2
4116
Z3
4116
Z4
4116
Z5
4116
Z6
4116
Z7
4116
Z8
4116
Z9
74LS374
Z10
74LS244

Parallel printer interface.

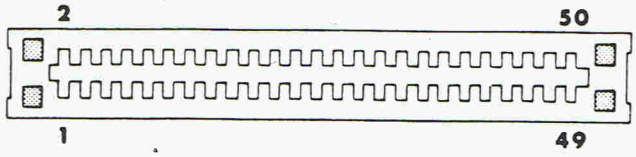




Joystick interface.

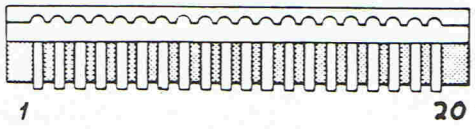


Expansion port.



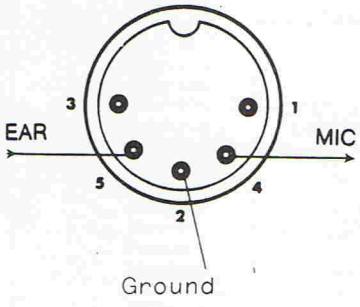
1	Ground	11	A11	21	/NMI	31	BD3	41	BD5
2	A8	12	A1	22	/WAIT	32	/C3	42	-
3	A7	13	A0	23	/HALT	33	-	43	BD0
4	A6	14	A12	24	/BUSAK	34	/C2	44	-
5	A9	15	A14	25	/ROMDIS	35	DB6	45	BD2
6	A5	16	A13	26	/MREQ	36	/RD	46	/RESET
7	A4	17	/RFSH	27	/WR	37	BD4	47	/M1
8	A3	18	A15	28	/C4	38	-	48	/IORQ
9	A10	19	/INT	29	-	39	BD7	49	BD1
10	A2	20	/BUSRQ	30	/C1	40	-	50	+5V

Parallel port.



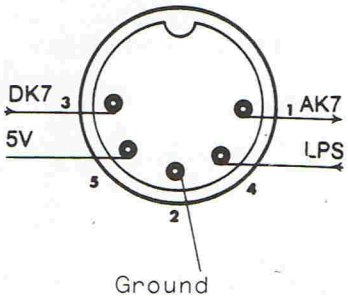
1	Ground	6	IOA5	11	IOB2	16	IOB5
2	+12V	7	IOA1	12	IOB1	17	IOB7
3	IOA3	8	IOA2	13	IOB0	18	IOB6
4	IOA4	9	IOA7	14	IOB3	19	+5V
5	IOA0	10	IOA6	15	IOB4	20	-12V

Cassette port.



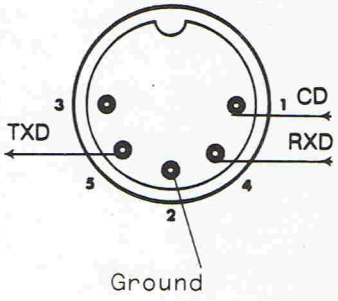
MIC To cassette rec.
EAR From cassette.

Light pen.



DK7 Keyboard switch
AK7 connections.
LPS Strobe input.

Serial port.



TXD Transmitted data.
RXD Receive data.
CD Carrier detect input.