microbee 256TC

TECHNICAL REFERENCE

MANUAL

TECHNICAL REFERENCE

MAY 1987

Written by:
Garner Annett, Paul Antoine, Raymond Han, Gerard Hill, Max Maughan, Craig Southeren, Matthew Starr.

Edited by:
Max Maughan, Bruce McKinnon.

Published by
Honeysoft Publishing Company
1a Pattison Avenue,
Waitara, N.S.W. 2077

ISBN 0 949309 72 9

First published in Australia 1987

Printed by
Central Coast Printery
Debenham Road,
West Gosford, N.S.W. 2250

T A B L E   O F   C O N T E N T S

Introduction

1 Introduction

The Microbee 256 TC is culmination of five years of Microbee computer development The best features of previous design have been incorporated onto a single, four layer, printed circuit board

1 1 Technical spec - standard

CPU : Z80 3 375MHz clock rate

RAM : 256k Dynamic RAM

ROM . 16k Boot / Network ROM / Kernel

VDU Memory Mapped
2k Screen RAM
2k Colour RAM
2k Attribute RAM
4k Dual Font ROM
Formats 40x24 for Videotex
64x16 for Basic
80x24 for CP/M
Flashing (character ON/OFF
Inverse attribute bit 6 (LA14)
(does not need PCG)
16k Programmable Character Generator or 131,072 pixels

VIDEO : Monochrome standard 1 volt peak to peak composite video with negative sync

COLOUR : R G B intensity with positive or negative sync

KEYBOARD . Full sized 92 keys including numeric pad and 12 function keys

DRIVE : Single 3 5 inch, double density 800k

CASSETTE Built in to load and dump programs at 300 and 1200 baud

SERIAL RS232 Can be used to connect modems, printers and to network with other computers

PARALLEL : 8 bit I/O fully programmable with strobe pulse for centronic printers Connection for Star Network

SOUND Internal speaker with volume control for BELL under CP/M

REAL TIME CLOCK Using the Motorola RTC 146818 Battery backed with a nicad battery

DISK CONTROLLER : TMS2793NL Texas Instruments

POWER SUPPLY 10 to 12 volts at 1.5 amps. Standby current approx 700mA to 800mA. Peak current during a single disk access 1200mA

POWER SUPPLY UNITS For monochrome use, use the Microbee M15 monitor with the internal power supply. For Colour use, use the Microbee plug pack number 111-012 (10V, 1.5amps)

OPERATING TEMPERATURE : 5-45 degrees C

OPERATING HUMIDITY 20% 80% relative humidity

1.7 Technical spec options

MOST OPTIONS ARE NOT SUPPORTED WITH SOFTWARE AT PRESENT

VDU Expandable from 2k to 8k
Four pages of Memory mapped screen
8k Screen RAM
8k Colour RAM
8k Attribute RAM

Character ROM expandable from 4k to 8k or 16k
6k Font ROM expansion for other languages

GRAPHICS Expandable from 16k to 32k 32k Programmable Character Generator or 262,144 pixels full 80x24 graphics

KEYBOARD Can use serial keyboard instead of the internal Microbee 256 TC keyboard

DRIVE Second 3.5 inch, double sided double density 800k drive can be added

SERIAL True RS232 signal can be added for negative swing

EXPANSION Internal 50 way Z80 bus fully buffered available for external testing and further expansion

Internal 40 way for hard disk interface which can be used with internal 3.5 inch drives

SOUND Internal circuit can use either the BC548 transistor or the LM386 op amp. Also socket available for the TI SN76489AN sound generator

Technical Description

2 Technical Description

This chapter details the theory behind the operation of the Microbee 256 TC's hardware

2 1 The CPU

The Z80 CPU is an NMOS device and so is only capable of supplying one TTL load per output The buffering of its outputs increases its current drive capabilities and gives a better noise margin The Z80 CPU, U16, transmits and receives all data via bi-directional data lines, designated D0 through D7 D0 is the Least Significant Bit (L S B) of the data The data lines are buffered by U37, a 74HCT245 octal bus transceiver

Pin 1 of U37 controls the direction of data flow through the device This pin is set high when either a READ cycle is indicated by the CPU's ZRD* signal, or when the ZIORQ* and ZM1* signals are concurrently active, indicating an interrupt acknowledge read from a device (such as the PIO.

Pin 19 either enables (low) or disables (high) U37 When disabled, all outputs are tristate (high impedance) Thus, the CPU data lines are isolated from the external data bus

U37 will be disabled for one of two reasons The first is when an external device tries to gain direct access to the 256 TC internal buses by asserting the BUSREQ* signal When the Z80 detects that this signal is low, it completes the current bus cycle and responds with the ZBUSAK* signal This line is used to disable processor control of the data bus, address bus and all control signals It is inverted by one gate of U7 and used to disable the address and control buffers (U15,U17 & U27) This inverted line is then inverted again to give a buffered BUSAK* signal to external devices and drive pin 5 of U6, which disables the data bus buffer U37 when BUSAK* is low

The other occasion that U37 is disabled is in order to perform the power-on reset or keyboard reset This reset function is mainly performed by one half of U95, a 74HC74 dual D flip-flop This flip flop is reset using pin 13 whenever the processor is reset When this flip-flop is reset, the Q output, pin 9, goes low, and therefore disables U37 via one gate of U6

On the processor side of the U37 buffers are pull down resistors in resistor SIP RP1), forcing the CPU data lines low whenever the buffers are disabled Thus, the processor is forced to execute the instruction code 00h (a 'NOP') 'NOP' is the No OPeration instruction for the Z80 that does as the name suggests, nothing The Z80 accordingly increments through its addresses, starting at address 0000h and executing NOPs until it reaches the desired ROM address This address is determined by U39 and the signals VIDEO-S and ROM S

The active low signal (RCS) is gated with MREQ* by U40 and then goes to the SET input pin 10) of U95 The Q output on pin 9 is now high, enabling U37 and the first byte of the ROM appears at the CPU Normal CPU execution continues from this point Thus, the 256 TC avoids the need for having ROM at address 0000h

The reset circuitry also serves the purpose of protecting the RAM from spurious writes which the Z80 tends to do when resetting This is achieved by gating the WR* line through U32 so that writing is disallowed as soon as the reset is applied and up until the ROM is found

The Z80 address bus is buffered by two 74HCT541 octal buffer chips, U15 and U27 These chips are disabled by taking pins 1 and 19 high, as is done when the Z80 grants bus access to an external device by asserting the BUSAK* output

The various control lines that the Z80 asserts are buffered by a third 74HCT541, U17 These are also disabled when the Z80 acknowledges an external bus request by using the BUSAK* line

2 2 Clock circuitry

The 256 TC's crystal oscillator provides all the timing for both the VDU and CPU

A crystal oscillator is formed using inverters in a 74LS04 package (U4 with a further inverter gate that is used to 'square up' the signal Buffered output at the 13 5 MHz crystal frequency appears at pin 12 or U4

The 13 5MHz clock is fed to U5 and U3, the output of the U3 gate is fed to pin 2 of U14 (HC161) where it is divided by 4 to provide a clock frequency of 3 375MHz Additional outputs of U14 are 6 75MHz on pin 14 and 1 68MHz on pin 12

The clock circuit includes three latch circuits U5 and half of U82 These are used to latch various combinations of the seven clock frequencies generated in the clock circuitry These latches are latched by using port 9 and address lines 8 and 9 There are three combinations in all and they are

80 column	LD	a,0	or	POWER ON and
	IN	a,(9)		RESET
40 column	LD	a,1		
	IN	a,(9)		

Signal	IC Pin	80 col	40 col	J82 SET
Dot Clock	U3 6	13 5MHz	6 75MHz	13 5MHz
CD Clock	U3 3	13 5MHz	6 75MHz	13 5MHz
CCLK *	U4 6	1 68MHz	84MHz	1 68MHz
CCLK	U14 12	1 68MHz	84MHz	1 68MHz
LOAD	U8 6	1 68MHz	84MHz	1 68MHz
CO1	U14 13	3 375MHz	1 687MHz	3 375MHz
3 375MHz CPU)	U3 8	3 375Mhz	3 375Mhz	6 75MHz

The CPU clock 3 375MHz or 6 75MHz feeds the Z80 CPU on pin 6, the Z80 PIO on pin 25, and pin 12 of connector X12

The character clock CCLK, used by the 6545 CRTC (U100) and generated by U14 appears on pin 12 and is the master frequency divided by 8 The synchronous load pulse LOAD, used to load the video shift register U4 (74HC166 with dot information, is generated by NORing the outputs of pins 12, 13 and 14 of U14 using one gate of U8 (74HC10

2 3 I/O circuitry

The Z80 microprocessor is capable of addressing a total of 256 input/output ports Not all of these are used U88 and U46 decode address lines 0-4 and M1* (machine cycle 1) to give various port selects, which are used to drive the chip selects on the RTC chip (U104), the 6545 VDU controller, the LV DAT bus (see section on the VDU for details), etc

U64 latches data for the LV DAT bus. and U65 (a tri state buffer) allows it to be read by the CPU

U1, the PIO, is a Z80 family device which implements 16 bits of fully independent input or output. As a Z80 family device, the PIO's configuration is fully programmable by the CPU and this feature has been used to generate the RS232 serial interface, the cassette interface, sound generation and the 8 bit programmable I O port All the port A signals of the PIO are taken directly to a DB15 socket, X1, while individual bits of port B are used for cassette interface, RS232 interface, keyboard control and networking

Most of the Z80 CPU bus signals are used by the PIO to control direction of data, interrupts, etc The PIO requires only two address lines to decode the four internal control ports The chip select signal active low) from pin 15 of U88 is applied to pin 4 of the PIO The 'interrupt enable in' IEI line of the PIO is tied high and as such, it becomes the highest priority interrupting device in the 256 TC

Port A is suited to any kind of 8 bit I/O It is commonly used as a parallel printer port, but may be used, for instance, to interface an external keyboard This port plays an important role in providing the interface to the Microbee Star Network

Either of the ports on a Z80 PIO may be programmed to operate in what is known as 'BIT' mode In this mode each bit of the port is programmable as either an input or an output This flexibility has been put to great use in the 256 TC, with port B of the PIO on the motherboard configured to provide both cassette and RS232 I/O ports

The music chip (U24) is driven from the Z80 data bus whilst its chip select is driven from pin 11 of U88 RV1 provides volume control for the speaker output and RV2 reduces the level of the 'bit bashed tone' output of the PIO so that it suits the music chip The 'bit-bashed tone' output of the PIO is connected directly to the op amp if the music chip is not fitted In the standard 256 TC the op amp (U44) is replaced by a BC547 transistor Q2

With respect to the RS232C standard Australian standard V24/V28), the 256 TC is configured as data terminal equipment' which allows standard connection to modem devices The 256 TC does not, however, fully implement the RS232C standard The 256 TC does not support all standard signals, nor does it provide a negative voltage swing on output In general, this does not provide a problem with most RS232 equipment

A negative voltage swing is available by fitting U13 (751701) and U25 (7660) and a few additional passive components see circuit diagram A for values)

The RS232 data transmit signal (pin 2 of X2) is driven by gates of U4 and J102 R16 protects the output line from damage due to an external short circuit If U13 is fitted, R16 must be removed

The RS232 receive data line (pin 3 of X2) takes RS232 signals over a wide voltage range (-12V to +12V) The resistor and diode networks (D5, D6, etc) limit the voltage (0V-5V range) being fed to pin 31 (bit 4) of the PIO If J13 is fitted R14, R15, D5 and D6 are not needed and link 1 has to be connected

The Clear To Send (CTS) and Clock (CLK) inputs from the RS232 port pin 5) are voltage converted in a similar manner to the receive data line

The cassette interface of the 256 TC is software driven, a major advantage in realising far greater reliability than is often obtainable using hardware designs. This is because the interface software can actually 'track' the data coming in on a 'bit-by-bit' basis and can therefore cope with much greater speed irregularities between tape recorders.

The cassette data output consists merely of an RC network which accepts a signal from DB1 (pin 28 of the PIO). The signal is attenuated and then decoupled prior to sending it to the cassette recorder's 'aux' input. This signal appears on pin 3 of the 5 pin DIN socket. Microphone inputs may be used if the 1k5 (R20) resistor is decreased in value (approx 680R).

The cassette data input circuit is slightly more complicated. The input (pin 5 of the DIN socket) passes first to an attenuator/decoupler. Following this is an op-amp (U105) to allow a wide range of input levels to be squared up before the signal is passed to pin 27 of the PIO (DB0). The two diodes across the inverting and non inverting inputs to the op-amp clip any input signals greater than the diodes forward voltage in either direction. Two capacitors are required by the CMOS op-amp for pre-compensation.

2.4 The Keyboard

The keyboard is scanned by U60 and the output of U60 is controlled by U61 and U71. The keyboard can be used in one of two ways. It can be programmed to generate interrupts when a key is pressed or it can be polled and read when required.

The keyboard uses two software ports, port 2 bit 1 and port 18h, 8 parallel bits.

When a key is pressed, pin 7 of U60 changes state and sets a flip flop using two gates of U71. Pin 13 of U71 connects to bit 1 port B of the PIO (U1). When this pin goes high it indicates a key has been pressed and it will remain high until a read from port 18h is made. When bit 1 goes high, and bit 1 is programmed to generate an interrupt, the keyboard can be read by the interrupt routine. In polled mode you wait for bit 1 of port 2 to go high and then do an IN routine from port 18h to get the data.

When an IN port 18h is made, enabling U61 and putting the data from U60 onto the Data Bus, the flip flop, using U71, is reset U71 pin 1 goes high and tells U60, via pin 38, that the data has been read

U60 also has a keyboard buffer which can store up to nine down key strokes The 3870 (U60 generates two lots of data for each key pressed When a key is pressed a code is generated with bit 7 set and when the key is released the same code is generated with bit 7 reset see the following tables for the key codes)

For software programming of the keyboard, see chapter 4 of this manual

2 4 1 External keyboard

With an external keyboard fitted, U60 is not used RP4 must be fitted and D8 must replace D13 Data from the external keyboard is read in serial format via X15 (pin 11) to Port 2 bit 1

Port 58h and data lines 1, 2 & 3 are used as control lines for the serial keyboard and port 2 bit as the input data port U68 is used to latch the control data lines

RP4 is used to hold all data lines high when port 18h is read and returns FFh (code not generated by U60), indicating there is no internal keyboard fitted This then allows the software to use port 58h to find the external keyboard

2 4 [illegible] Keyboard code tables

ASCII	DOWN HEX	DECIMAL	UP [illegible]	DECIMAL
F1	80	128	00	0
ESC	81	129	01	1
TAB	82	130	02	2
BRK	83	131	03	3
	84	132	04	4
0 (num)	85	133	05	5
DECIMAL	86	134	06	6
SPACE	87	135	07	7
F2	88	136	08	8
1 - !	89	137	09	9
Q	8A	138	0A	10
A	8B	139	0B	11
	8C	140	0C	12
CAPS	8D	141	0D	13
LF	8E	142	0E	14
INS	8F	143	0F	15
F3	90	144	10	16
2 - @	91	145	11	17
W	92	146	12	18
[illegible]	93	147	13	19
ADD	94	148	14	20
2 (num)	95	149	15	21
3 (num)	96	150	16	22
Z	97	151	17	23
F4	98	152	18	24
3 - #	99	153	19	25
E	9A	154	1A	26
D	9B	155	1B	27
SUBTRACT	9C	156	1C	28
5 (num)	9D	157	1D	29
6 (num)	9E	158	1E	30
X	9F	159	1F	31
F5	A0	160	20	32
4 - $	A1	161	21	33
R	A2	162	22	34
F	A3	163	23	35
MULTIPLY	A4	164	24	36
8 (num)	A5	165	25	37
9 (num)	A6	166	26	38
C	A7	167	27	39
F6	A8	168	28	40
5 %	A9	169	29	41
T	AA	170	2A	42
G	AB	171	2B	43
7 (num)	AC	172	2C	44
1 (num)	AD	173	2D	45
4 (num)	AE	174	2E	46
V	AF	175	2F	47
F7	B0	176	30	48

ASCII	DOWN HEX	DOWN DECIMAL	UP HEX	UP DECIMAL
6 - ^	B1	177	31	49
Y	B2	178	32	50
H	B3	179	33	51
DIVIDE	B4	180	34	52
DOWN ARROW	B5	181	35	53
RIGHT ARROW	B6	182	36	54
B	B7	183	37	55
F8	B8	184	38	56
7 - &	B9	185	39	57
U	BA	186	3A	58
J	BB	187	3B	59
	BC	188	3C	60
LEFT ARROW	BD	189	3D	61
	BE	190	3E	62
N	BF	191	3F	63
F9	C0	192	40	64
8 - *	C1	193	41	65
I	C2	194	42	66
K	C3	195	43	67
	C4	196	44	68
	C5	197	45	69
UP ARROW	C6	198	46	70
M	C7	199	47	71
F10	C8	200	48	72
9 - (	C9	201	49	73
O	CA	202	4A	74
L	CB	203	4B	75
	CC	204	4C	76
BACK SPACE	CD	205	4D	77
CR	CE	206	4E	78
, - <	CF	207	4F	79
F11	D0	208	50	80
0 -)	D1	209	51	81
P	D2	210	52	82
; - :	D3	211	53	83
DEL	D4	212	54	84
` - ~	D5	213	55	85
\ - |	D6	214	56	86
> - .	D7	215	57	87
F12	D8	216	58	88
- -	D9	217	59	89
[- {	DA	218	5A	90
' - "	DB	219	5B	91
	DC	220	5C	92
= - +	DD	221	5D	93
] - }	DE	222	5E	94
/ - ?	DF	223	5F	95
SHIFT	E7	231	67	103
CONTROL	EF	239	6F	111
ALT	F7	247	77	119

2.4.3. Keyboard codes grouped.

FUNCTION.	DOWN	UP
F1	80	00
F2	88	08
F3	90	10
F4	98	18
F5	A0	20
F6	A8	28
F7	B0	30
F8	B8	38
F9	C0	40
F10	C8	48
F11	D0	50
F12	D8	58

CONTROL.		
ESC	81	01
TAB	82	02
BRK	83	03
SPACE	87	07
CAPS LOCK	8D	0D
LF	8E	0E
INS	8F	0F
BACK SPACE	CD	4D
CR	CE	4E
DEL	D4	54
SHIFT	E7	67
CONTROL	EF	6F
ALT	F7	77

NUMERIC.		
0 -)	D1	51
1 - !	89	09
2 - @	91	11
3 - #	99	19
4 - $	A1	21
5 - %	A9	29
6 - ^	B1	31
7 - &	B9	39
8 - *	C1	41
9 - (	C9	49

ALPHA		
A	8B	0B
B	B7	37
C	A7	27
D	9B	1B
E	9A	1A
F	A3	23
G	AB	2B
H	B3	33
I	C2	42
J	BB	3B

ALPHA.	DOWN	UP
K	C3	43
L	CB	4B
M	C7	47
N	BF	3F
O	CA	4A
P	D2	52
Q	8A	0A
R	A2	22
S	93	13
T	AA	2A
U	BA	3A
V	AF	2F
W	92	12
X	9F	1F
Y	B2	32
Z	97	17

NUMERIC PAD.		
0	85	05
1	AD	2D
2	95	15
3	96	16
4	AE	2E
5	9D	1D
6	9E	1E
7	AC	2C
[illegible]	A5	25
9	A6	26
DIVIDE	B4	34
MULTIPLY	A4	24
SUBTRACT	9C	1C
ADD	94	14
DECIMAL	86	06
UP ARROW	C6	46
LEFT ARROW	CF	4F
DOWN ARROW	B5	35
RIGHT ARROW	B6	36

SPECIAL FUNCTION.		
, - <	CF	4F
; - :	D3	53
\` ~	D5	55
\ - |	D6	56
- >	D7	57
	D9	59
[- {	DA	5A
' - "	DB	5B
= - +	DD	5D
] - }	DE	5E
/ - ?	DF	5F

2.4.4. Keyboard matrix

X14 X13	1	2	3	4	5	6		8
1		A						
2		CTRL						
3		SHIFT						
4	F		[illegible]	F5	$ 4	8n	9n	Prt Sc
5	G	V	T	F6	% 5	1n	4n	[illegible]
6	H	B	Y	F7	^ 6	\ /n	END>	[illegible]
7	J	N	U	F8	&	HOME<		
8	K	M	I	F9	* 8		[illegible]	
9	L	< ,	O	F10	(9	BS	[illegible]	
10	;	>	P	F11	) 0	_ -		[illegible]
11	" '	? /	{ [	F12		+ =		
12	D	X	E	F4	# 3	5n	[illegible]	[illegible]
13	S	Z	W	F3	@ 2	2n	[illegible]	[illegible]
14	A	INS	Q	F2	! 1	CAPS LOCK	[illegible]	
15	# BRK	SPACE	TAB	F1	ESC	0n	[illegible]	

n = numeric pad

2.5 Real time clock circuitry

The HD146818, U104, is a CMOS device which combines three unique features: a complete 'time of day' clock with alarm and a one hundred year calendar, a programmable periodic interrupt and square-wave generator and 50 bytes of low power static RAM.

U104 is battery backed with a nicad battery so that once the correct time is set it should never have to be adjusted again.

2.5.1 Main clock features

- * - Time of day clock and calendar
  - counts seconds, minutes, and hours of the day
  - counts days of week, date, month, and year
- * - Binary or BCD representation of the time, calendar and alarm
- * - 12 or 24 hour clock with AM and PM in 12 hour mode
- * - Automatic end of month recognition
- * - Automatic leap year compensation
- * - Interfaced with software as 64 RAM locations
  - 14 Bytes of clock and control register
  - 50 bytes of general purpose RAM

2.5.2. RTC memory address map

00h	Seconds	08h	Month
01h	Seconds alarm	09h	Year
02h	Minutes	0Ah	Register A
03h	Minutes alarm	0Bh	Register B
04h	Hours	0Ch	Register C
05h	Hours alarm	0Dh	Register D
06h	Days of the week	0Eh 3Fh	User RAM
07h	Date of month		

Full programming details can be [illegible] either the Hitachi Semiconductor Data [illegible] Microcomputer manual HMCS6 16 EM or the Motorola 8/[illegible] bit Microprocessor manual.

6 The VDU circuitry

The 256 TC's Visual Display Unit (VDU) is memory [illegible] character displayed is 8 bits wide by up to 16 scan [illegible] from two sources; the first is the masked font ROM (U53). This ROM contains the two standard character sets, one being used in the 64 character by 16 line mode (a character cell of 8 bits by 16 scan lines) while the other is used in 80 character by 24 line mode (8 bits by 11 scan lines). The display is not restricted to these display formats, as will be seen lat[illegible]

The second source of character dot information is the Programmable Character Generator (PCG) RAM. This allows selection of a pre programmed character from [illegible] most significant bit (bit 7) of the ASCII characte[illegible] value contained in the screen character memory.

The VDU logic is based around a 6545 Cathode Ray Tube Controller (CRTC). The CRT controller consists of [illegible] a linear address counter and cursor logic. The 6545 also has a light pen capture register connected to pin [illegible] of X2 and control circuitry for interface to the [illegible] and 0Dh, port 0Ch is the 6545 'address' port while port 0Dh is the 'data' port.

All timing for the CRTC is derived from the clock input on pin 21. This is the character rate clock (CCLK*) and is generated by dividing the 13.5MHz master clock by 8, the dot width of a character. The character clock of 1.68 MHz is generated by U14, a 74HC161 [illegible] synchronous counter. This clock is used by the 6545 horizontal and vertical counters to generate the display enable (DISPEN) signal, horizontal [illegible] (HSYNC), vertical sync (VSYNC) and scan line [illegible] signals RA0-RA4. The initial values of these co[illegible] [illegible] fully programmable to allow for different screen formats.

The 256 TC can be fitted with up to 32K ... (character generator) RAM capa... programmable ... number of ... characters ... screen ... values of the characters ... the screen) and 8K of attribute RAM

... banks is fitted ... 256 and ... screen ... attribute RAM is used ... selecti... each character ... which ... Basic ... resolutions of ... to 64 by 256 ... resolution

... insert ... wait states ... the ... without screen flicker

The data buses ... and ... from the CPU data bus. These are ... by means of tri-state buffers ... This allows the CPU to both read and write ... screen and attribute RAMs ... and ... so that the ... accesses the RAMs

... from the ... data selector ... foreground to background ... output

Output from the ... RAM and ... shift register ... and then used ... selection at U91

U52 is a PAL (programmable ...) ... takes care of all of the ... the flow of address ... the circuit is perhaps the most difficult part ... comprehend

The character row count from U100 is shown on the main circuit as the RA BUS. This cycles through the various parts of each character as they are to be sent to the shift register for display.

This is combined with the particular character's ASCII code from the screen RAM and the character set select from the attribute RAM. The MSB (most significant bit) of each character in the screen RAM determines whether a programmable character or a ROM character is to be used.

On the other side of the diagram, the CPU address bus is combined with the LV DAT bus, which originates from U64 (circuit A, main circuit). The LV BUS allows the CPU to select which chip/bank of the PCG RAM it is going to read from or write to. There is also a bit of the DAT bus called LV7 which is used by the software written for the Premium Microbee to turn on the chip bank select of the PCG. This means that the 256 TC [illegible] run older software without modification.

Both 16 bit buses are fed to a set of data selectors (U2, U83, U74 and U75) which determine whether the CPU or the video circuitry has access. The switching is controlled by U52.

The resulting address bus, called the PC BUS, is fed to the address inputs of the PCG RAM and character ROM. The LV7 signal is used to enable the chip bank selects.

The top bit of the data from the screen RAM is taken off the PC BUS and used to select whether the RAM or ROM characters are used. Address lines to the colour, screen and attribute RAM have a similar arrangement. Character select information from U100 (called the MA BUS) is fed into data selectors U86 U111 and U99 which select either the MA BUS or the A BUS (the CPU address bus) and LV DAT bus. The selected bus is used to feed the address inputs of the colour screen and attribute RAM.

2 The Memory Circuitry

The 256 TC uses eight 4256 dynamic RAMs. These are 256k by 1 bit RAMs (i.e. eight are required to supply the 8 data bits needed per byte.

The addressing of the RAMs is critical to the operation of the board, but its timing can be assumed to be correct if RAS and CAS can be observed at appropriate pins. Please note that the Data lines often in the tri state mode and wi ear to float.

The 17 bit address bus is multiplexed via three address multiplexers U29, U30 and U31 and is presented nine bits at a time to the RAM. The nine ROW address lines to the RAM are strobed with the RAS while CAS strobes the nine column address lines (refer to the data book for RAS and CAS timing requirements.

The PAL (U52) generates the RAS and CAS signals and one gate of U43 generates the MUX changeover delay.

At power on the 256 TC behaves a little differently from other Microbees. The data bus is disabled until the address of the ROM 1 (U38) is reached at 8000h (figure 2.1). The data bus is then enabled and code from the ROM executed. This code will first check for the presence of a disk controller and a disk in the disk drive. If no disk can be found then the presence of connection to a Microbee Star network is checked. If the 256 TC cannot detect connection to a network, it will continue to execute code from the ROM, the Kernel main menu being displayed on the screen as a result.

If a disk is found or if <F1> or <F2> is pressed from the Kernel main menu., the Kernel will attempt to detect the presence of either a disk in the disk drive or connection to a Star network control being returned to the Kernel if neither cannot be detected. If detection is successful then the CPU copies an image of either the disk boot code or the network boot code into RAM at 6000h (figure 2.1). Control is then passed to a small routine loaded into the PCG RAM which then switches the RAM from 0000h to 8000h, so that the image of the ROM just produced is located at E000h. Control is then passed to the ROM image at E000h (figure 2.1).

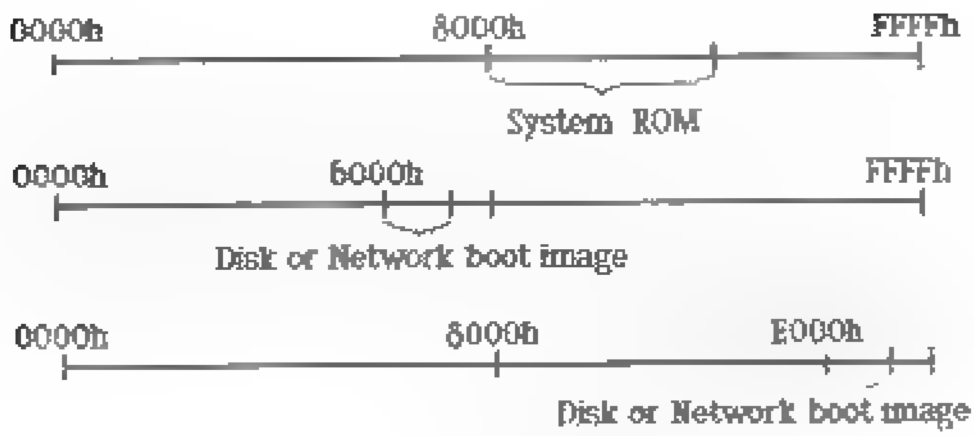


Fig 2.4: The 256 TC's power up memory map

This moving of the ROM image is another feature of the DRAM section. The PAL (U39) has a 6 bit latch (labelled S0 - S5) at port 50h. The outputs of the latch determine where each bank of memory is located in the memory map.

Many different combinations of memory can be selected, however, it must be remembered when switching blocks that the scratch areas of memory do not move with the selected blocks. This makes the movement of blocks to enable data storage somewhat difficult and with Basic, virtually impossible. Memory block selection and movement is discussed further in chapter 3.

2.8 Disk controller circuitry

The following table shows briefly the functions of each of the [illegible] in the controller [illegible]

Integrated Circuit	Function
U34 FD2793	The disk controller
U46 74HC175	Produces side select
U50 74HC139	Drive select decoder

There are three broad categories for communication between the computer and the disk drive

1) reading from disk
2) writing to disk
3) mechanical information

to the drive
- a) motor starting stopping
- b) head loading
- c) head stepping
- d) direction of head stepping

from the drive
- e) write protect signal
- f) index pulse
- g) track 0 detector
- h) ready signal (not used)

Before any reading or writing occurs the [illegible] information must be [illegible] with [illegible] to read or write [illegible] disk [illegible] read [illegible] This information must be [illegible] the disk [illegible] integrated circuit (FD2793)

The register select lines are A0 and A1 pins [illegible] 6 of the FD2793 [illegible] the appropriate [illegible] levels on RE* and W[illegible] [illegible] FD2793 [illegible] registers [illegible] RE* signals initiate the data transfer

When reading or writing, the FD2793 first reads in the track and sector details until the matching track sector is found (coupled with appropriate head stepping instructions, etc.). Then, providing the signals WG*, WE* or the RE* are correctly set, reading or writing may occur.

2.8.1. Reading data

The raw data pulses from the disk drive enter the FD2793 at pin 27. On the input side of the FD2793, when the RE* goes low, valid read data is output to the 256 TC no more than 350nS later. When the RE* goes high, valid read data is finished between 50nS and 150nS later.

2.8.2. Writing Data

This requires that the write gate (WG) to the disk drive should rise before the write data (WD), out of pin 30 of the FD2793, by a maximum of 8uS. The write data exits the FD2793 at pin 31 and then travels to the disk.

2.9 The Power supply

The 256 TC power supply is based around a special high current switching regulator, the L296. This power supply has the following features:

* +5.1 volts output
* 4 amps output current
* Overvoltage output protection
* Overcurrent shortcircuit protection

The power supply board is fitted to the disk drive support bracket, underneath the keyboard. It is connected to the main PCB with four wires via connector X6.

The input voltage to the L296 can vary over a wide range, the limit input voltage on the 256 TC being limited to the voltage level required for the RS232 circuit. The RS232 circuitry on the 256 TC operates on the input supply. This must not be greater than 15 volts.

The supply voltage passes through a 4 amp fuse before it reaches pin 3 of the L296. The fuse is used if the output voltage exceeds 5.1 volts. If the output voltage rises above 5.1 volts, the voltage sense pin turns on the SCR which shorts the input voltage to ground, causing the fuse to blow.

If you ever find the fuse blown don't just replace it and power up again. Check the power supply on a separate DUMMY LOAD CIRCUIT. DO NOT USE THE MAIN BOARD. Short the fuse and measure the output voltage. If it is greater than 5.1 volts then check the few passive components around the L296. If these are OK then try a new L296.

If the output voltage is shorted to ground, the L296 will shut down. Removing the short will allow the output to return to normal. Further information on the L296 regulator can be found in SGS power supply data books.

Waveforms

3. Waveforms.

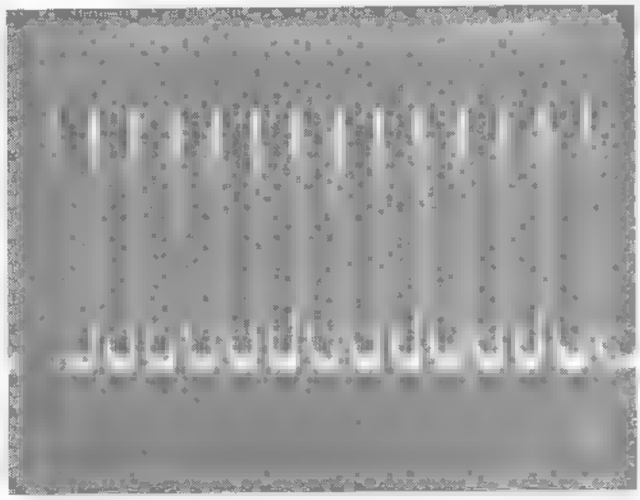
This chapter provides you with waveforms for important clock, data and video signals on a properly functioning 256 TC.

On these photographs, 0 volts is set to the second division, with the 256 TC powered up and the Kernel main menu displayed.

3.1. Dot clock.

Location - U3 pin 6

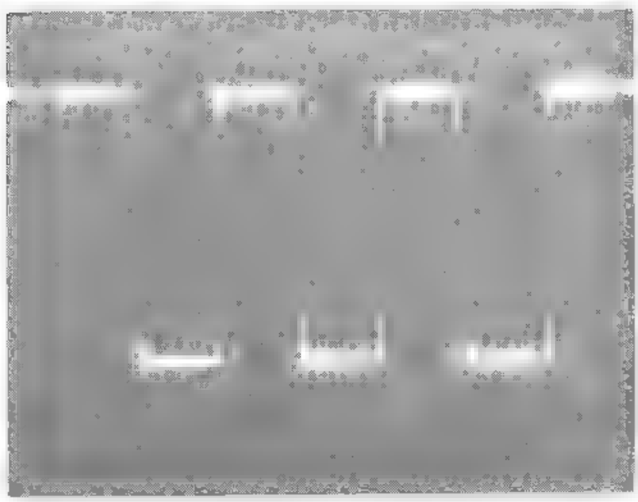
Time/Div - .1us.
Volt/Div - 1v/cm



3.2. CCLK*.

Location - U4 pin 6

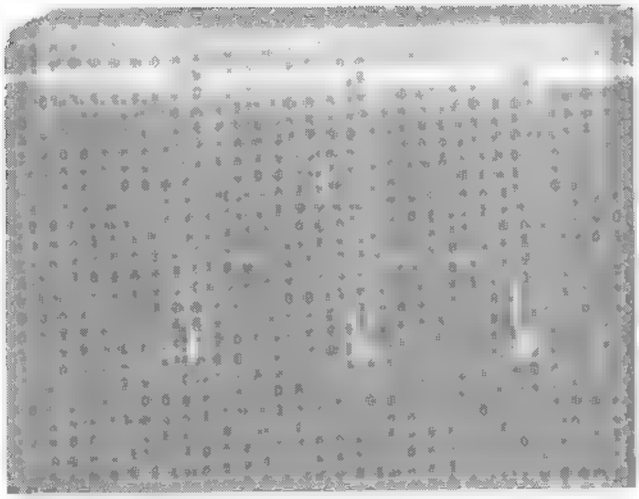
Time/Div - .1us.
Volt/Div - 1v/cm



3.3. Load.

Location - U8 pin 6

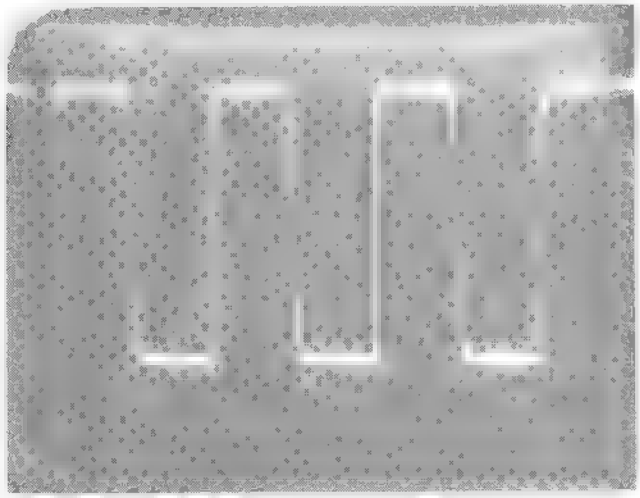
Time/Div - .2us.
Volt/Div - 1v/cm



3.4. CO1.

Location - U14 pin13

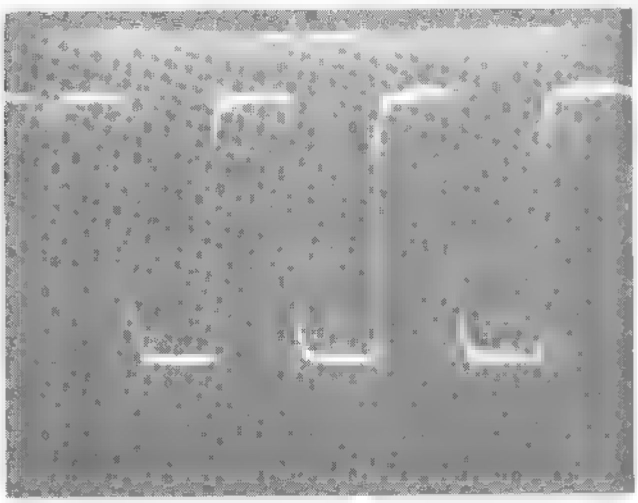
Time/Div - .1us.
Volt/Div - 1v/cm



3.5. CPU clock.

Location - U3 pin 8

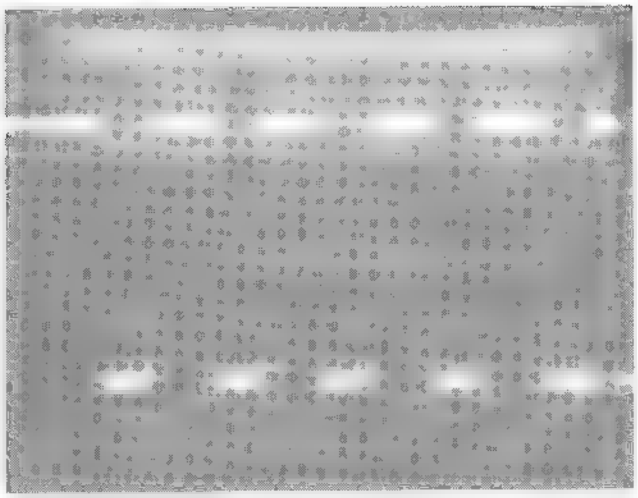
Time/Div - .1us.
Volt/Div - 1v/cm



3.6. Power supply.

Location - L296 regulator output.

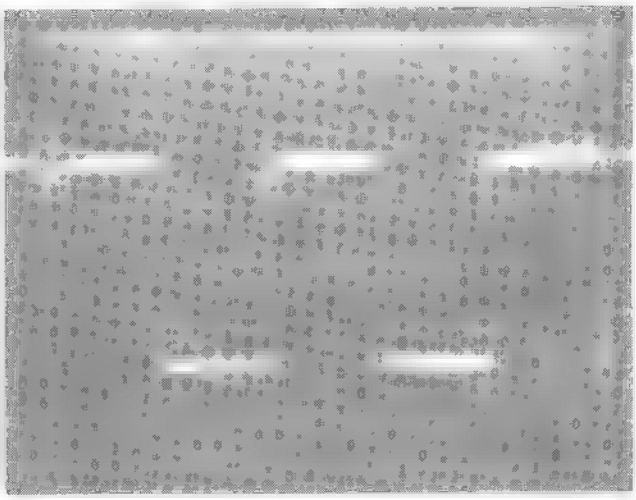
Time/Div - 5us.
Volt/Div - 2v/cm



3.7. Disk controller - CV1.

Location - CV1 Adj.

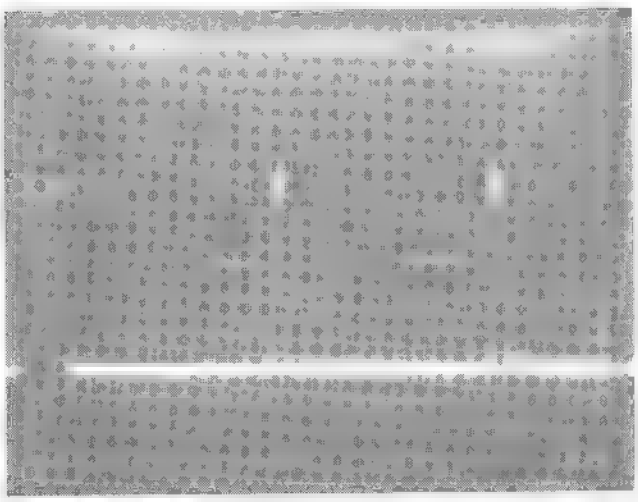
Time/Div - 1us.
Volt/Div - 1v/cm



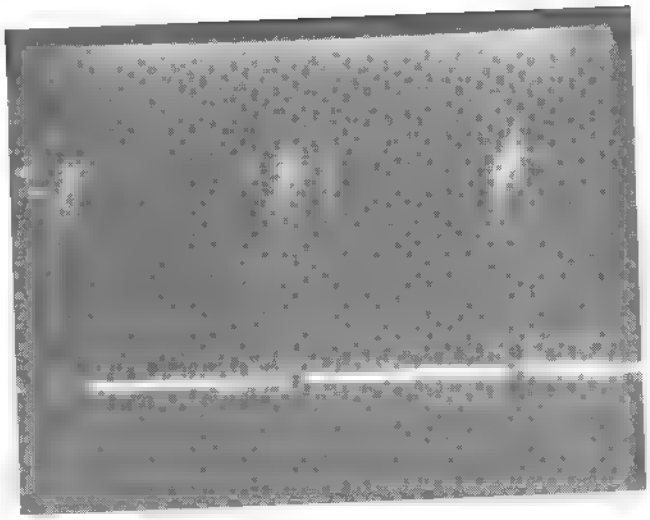
3.8. Disk controller - RV1.

Location - RV1

Time/Div - 1us.
Volt/Div - 1v/cm



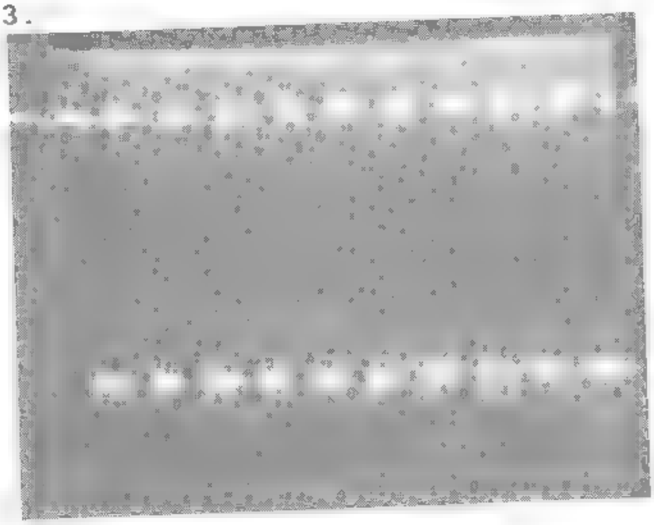
3.9. Disk controller - RV2.



Location - RV2

Time/Div - 1us.
Volt/Div - 1v/cm

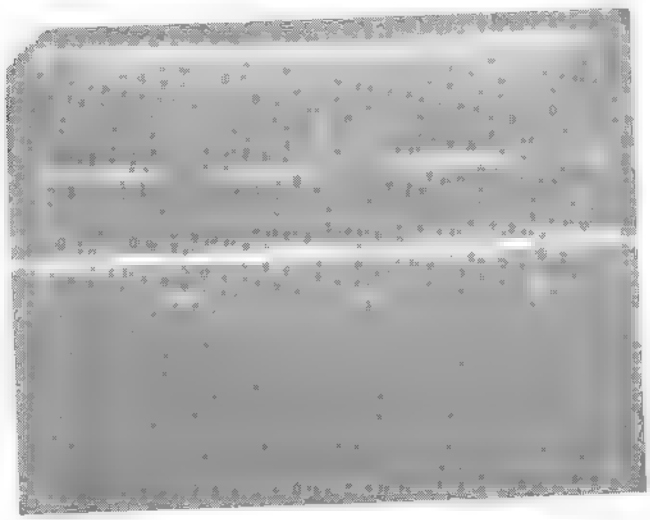
3.10. Disk controller - 2793.



Location U34 pin 24

Time/Div - .5us.
Volt/Div - 1v/cm

3.11. Composite video.



Location - Q3 (base)

Time/Div - 20us.
Volt/Div - 1v/cm

System Programming

4 System Programming

This chapter details some the changes and revisions that have been made to the operating system supplied with the 256 TC

4.1 The Kernel

In most cases, the Kernel will be by-passed completely as the 256 TC will access software from it's internal drive 99% of the time However the Kernel does perform a check upon bootup and power up It checks to see if there is a floppy disk controller disk drive or a disk in the drive If any one of these is not available, the main menu will appear

You will be greeted and informed of your last power up time this is the time your machine was last switched on

Four icons appearing in the middle of the screen invite you to select one of four functions which are available Each function is selected using a function key

F1 Boot disk from drive A

F2 Boot up Starnet system

F3 Set the system time

F4 Perform Self Test

Secondary functions are also available in this main menu They are

<CTRL ALT-DEL> - Perform a warm boot

<SHIFT F1> - Boot disk from drive B: and from then on, physical drive B will be drive A and physical drive A: will be drive B: (i e , drive codes swapped

<CTRL ALT M> - Jump into the system monitor

4.1.1 Boot disk from Drive A:

When the <F1> key is pressed, the boo[illegible] will flash and 5 attempts will be made to look [illegible] a disk in drive A

If all these attempts fails, 3 short beeps [illegible] follow and a DISK READ ERROR warning will be displa[illegible]e with the disk icon still flashing You will have a choice of either exiting to the main menu or re attempting to boot the disk

If you choose to exit to the main menu, press the <F2> key If you wish to re-attempt to boot the disk [illegible] <F1> key should be pressed a[illegible] the [illegible] process is repeated

If, however, a disk is found, control is passed onto a disk boot image Upon receipt of control the screen is re-programmed in 64x16 mode and a block cursor will appear on the top left corner of the screen Attempts to boot the disk will be made and if successful, the familiar Shell based CP/M operating system will greet you If, however, the attempts were unsuccessful, the system will keep retrying and, in some cases the system will 'hang'. Regretfully, the only way to exit from this situation is to press the RESET button (located on the back panel).

4.1.2 Network

The <F2> key will immediately put the 256 TC in Starnet mode and, if connected to an active [illegible] Server, the system should boot up quite happily and [illegible] any program

Please remember that if you have a program that uses the hardware to scan the keyboard, such as games they will not run until the software is updated to scan the new keyboard

4.1 Set Time

To set the time, press the <F3> key from the main menu and the Set Time menu will appear. You should see several fields which form the time and date. Please note that in this mode the time will not be updated. A flashing block cursor should also appear on the adjustment fields.

You can move the cursor into any fields and directly manipulate the fields from the keyboard. Use the right and left cursor keys to move the cursor to the field you want.

The time is based on a 24 hour clock although the system clocks operates on a 12 hour cycle. Simply adjust the time and set it. The field is laid out in a hh:mm:ss fashion where

hh = hour (0-23)
mm = minute (0-59)
ss = second (0-59)

The date is can also be adjusted in the same manner as the time, however it does not check for validity of the dates and may set the clock even if such as date is not available (e.g., 30th February 87). The field is laid out in a dd:mm:yy fashion where

dd = date (1-31)
mm = month (1-12)
yy = year (00-99)

To set the time according to the parameters in the fields, press the <F5> key or <F1> to abort the operations completely and return to the main menu.

4 1 4 Self Test

The system Self Test is fairly straight forward It simply does a self test of the computers main operations and gives a visual status report either with a TICK or a CROSS for pass or failure If a failure is reported, the number of cycles which it has passed through the self test is reported alongside the CROSS

4.1 4.1. Keyboard test

Press the key corresponding to that which appears on the screen If you press the wrong key, a beep will follow and more beeps will follow until you press the correct key If a key is faulty however, you will never be able to get to the next key, In this case press the <ESC> key and unless the <ESC> key is the one that is faulty, the self test will continue on

You can press the <ESC> key at any time to abort the keyboard test and continue on with the other tests or press the <BRK> key to bypass the self test completely

4 1 4 2 Parallel Printer test

This test simply prints out.

Microbee 256 TC Parallel Printer Test

on the first run and just checks for the parallel printer on each subsequent iteration If the printer is not selected, the test will fail

4 1.4.3. Serial Port test

An RS232 mini tester should be connected to the serial port with these pins connected to pass the test

2 3
6 24 or 20

To abort the Self Test, press <F2> and wait for the current test to be completed. If you want to re-run all the tests, press <ESC> again. To return to the main menu, press the <BRK> key.

4.1.5 Kernel function calls

The Kernel has provided emulations of XBIOS calls 15-18 for programmers writing software for Starnet student stations. These functions are located in memory at a location defined by value of EFFAh.

Jump Vector	Description
Scn in	Keyboard Scan In routine. Returns Z with keycode in A or NZ when no is key available.
Is_closed	Emulates extended BIOS call 15
Pkey Set	Emulates extended BIOS call 16. Buffer can contain up to 256 bytes
Pkey Read	Emulates extended BIOS call 17. Buffer can contain up to 256 bytes
Ext key	Emulates extended BIOS call 18

4.1.6. Machine code monitor commands

The machine code monitor is entered by pressing <CTRL ALT-M> from the kernel's main menu The following control keys have been implemented

<CTRL E> - reverse cursor 16 bytes

<CTRL X> - forward cursor 16 bytes

<CTRL S> reverse cursor 1 byte

<CTRL D> forward cursor 1 byte

<CTRL R> or
<SHIFT> UP arrow - reverse cursor 16*4 bytes

<CTRL [illegible]> or
<SHIFT> DOWN arrow - forward 16*4 bytes.

<CTRL-T> - toggles between text and hex write mode

<ESC> - return control to command line

4.1.6.1. A xxxx - Alter memory at address

Allows hand alteration of memory The byte changed is indicated by the position of the cursor

4.1.6.2. B xx - Boot a disk

Bootup the disk from drive xx (currently only 0 and 1)

4.1.6.3. C aaaa bbbb 1111 Compare memory

Compare two 1111 bytes long blocks starting at aaaa and bbbb

4.1.6.4. E xxxx - Examine memory

Command is similar to A command, except that <M> key needs to be pressed to invoke MODIFY mode

4.1.6.5. F ssss ffff xx) - Fill memory

Fill block of memory from ssss to ffff with xx. If xx is not specified, the block of memory will be filled with 00

4.1.6.6. G aaaa - Goto address

Execute a machine code routine at memory address aaaa

4.1.6.7. I aa - Input from port

Input a byte from port AA, and display it on the screen

4.1.6.8. M ssss dddd llll - Move memory

Move llll bytes from the ssss source address, to the dddd destination address

4.1.6.9. N - Network

Jump to the network routines, auto boot into the Starnet system

4.1.6.10. O aa bb - Output to port

Output byte bb, to port address aa

4.1.6.11. P - Clear screen

Clear the screen

4.1.6.12. S ssss ffff aa bb cc - Search for memory

Search the block of memory between ssss and ffff for the sequences of up to six bytes (aa bb cc ...

4.1.6.13. XR ttss dddd llll - Read from A: drive

Read llll bytes from the A: drive into memory at address dddd (no sector skewing). ttss, track and sector addresses

4.1.6.14. XW ttss dddd llll - Write to A: drive

Similar to XR, except this function write instead of reads

4.1.6.15. YR ttss dddd llll - Read from B: drive

Similar to XR, except this function performs a read from drive B

4.1.6.16. YW ttss dddd llll - Write to B: drive

Similar to XW, except this function performs a write to drive B

4.1.6.17. Z - Execute code

Execute program at location 0000h

4.2 BIOS calls

The 256 TC BIOS has all the standard CP/M 80 BIOS function calls implemented. In this section we will present a brief summary of each. For further details see 'Inside CP/M' by David E. Cortesi, CBS College Publishing, 1982.

Details on standard CP/M 80 BDOS calls can also be found in 'Inside CP/M'. Calling BIOS functions can be made easier by inserting the following macro at the start of your program and using to perform call the functions.

```
Bios call MACRO  function
          ld   hl,(1)            ;read the BIOS start
          push de
          ld   de,3*function     ;get to the entry point
          add  hl,de
          pop  de
          ld   ix,$+7            ;return address
          push ix
          jp   (hl)
          ENDM
```

This allows BIOS functions to be called simply by using the label BIOS CALL and specifying the call number (i.e., BIOS CALL 3 will call BIOS function 3).

Please note that these function calls were originally intended to be implemented on CP/M systems that used serial terminals and as such, some of the calls are either unnecessary or made redundant by the XBIOS calls.

The following program demonstrates the use of this macro by implementing a glass typewriter. Any characters you type at the keyboard will be echoed to the screen. You may press <CTRL-C> to exit the program and it is presented in M80 format.

```
 TV typewriter program
  (c) 1987 Microbee Systems Ltd

                aseg
                 z80
                org       100h


Bios call MACRO  function
                ld   hl,(1)          ;read the BIOS start
                push de
                ld   de,3*function   ;get to the entry point
                add  hl,de
                pop  de
                ld   ix,$+7          ;return address
                push ix
                jp   (hl)
                ENDM

loop
;get a key from the keyboard
                bios call       2

;check for CTRL C
                cp        'C'-64
                jp        z,0                  ;boot

;display the character on the console
                ld        c,a
                bios call       3
                jr        loop

                end
```

4.2.1. Function 0 - Warm boot CP/M

This function causes the calling program to be terminated and CP/M re-initialised (warm booted). If a $$$.SUB file exists on the logged drive then the next command will be read from it.

4.2.2. Function 1 - Check console status

This call causes the BIOS to check the status of the keyboard. If a non zero value is placed into the A register, then the key has been pressed and a character is ready to be read. A zero value will be returned if no character is ready be read.

4.2.3. Function 2 - Read from the keyboard

This call causes the BIOS to read a character from the keyboard. The ASCII value of the key is placed into the A register and the calling program will be suspended until the key has been read.

4.2.4. Function 3 - Send to the screen

Function 3 is used to send a character to the screen. The character to be sent must be placed into the C register before calling the function and the calling program will be suspended until the character is displayed.

4.2.5. Function 4 - Send to the list device

This function causes the ASCII equivalent of the value in the C register to be sent to the list device. The calling program is suspended until transmission is complete. The list device is normally the parallel port.

Assumes Bit 7 = 0

4.2.6. Function 5 - Send to the punch device

This function causes the ASCII equivalent of the value in the C register to be sent to the punch device. The calling program is suspended until transmission is complete. The punch device is normally the serial port.

Assumes Bit 7 = 0

4.2.7. Function 6 - Read from reader device

This function returns a character from the reader device into the A register. The calling program is suspended until the character has been read. The reader is normally the serial port.

4.2.8. Function 7 - Home the drive head

Function 7 is used to move the heads of the currently selected drive to track 0 of the disk (the outermost track).

4.2.9. Function 8 - Select drive

This call takes the drive number in the C register and logs it for any further operations. The value in C must be in the range of 0..15 representing drives A to P.

4.2.10. Function 9 - Move the drive head

This function allows you to move the drive head to any track on the disk. Once called, CP/M will read the value in the BC register pair and move the head to that track.

The 256 TC double sided disks use 80 tracks per side, tracks 0-3 being reserved as the CP/M system tracks. On single sided 256 TC disks, 80 tracks are used on side 0 of the disk, tracks 0 and 1 being reserved as the CP/M system tracks.

4.2.11. Function 10 - Set sector for I/O

Function 10 is used to inform CP/M on which sector of the current track it will perform I/O. On the 256 TC, ten 512 byte sectors are stored per track, numbered 1 to 10. The sector number must be loaded into the BC register pair before calling this function.

4.2.12. Function 11 - Set file buffer address

This function sets the value in the BC register pair as the address of the file buffer. The file buffer stores one complete sector (512 bytes).

4.2.13. Function 12 - Read a sector

This function will read the sector (512 bytes) specified by BIOS calls 8, 9 and 10 and place the data in the disk buffer, the address of which is defined by BIOS call 11. If a read has been successful then a zero value is placed into the A register. A non zero value is returned if an error occurs. The calling program is suspended until either the sector is read or an error is reported.

4.2.14. Function 13 - Write a sector

This function will write the data in the disk buffer (the address of which is defined by BIOS call 11) to the sector specified by BIOS calls 8, 9 and 10. The C register should be loaded with a value indicating the type of data being written.

- 00 - Normal write, pre-read if necessary, defer if convenient
- 01 - Directory write, pre-read if necessary, do not defer
- 02 - Initial write to sector, no pre-read necessary, defer if convenient

4.2.15. Function 14 - Poll list device

This function allows polling of the list device. When it is ready to accept a character a non-zero value will be returned in the A register.

4.2.16. Function 15 - Translate sector number

The sector number in the BC register pair is translated by the skew table whose address must appear in the DE register pair. The translated sector number will be returned in the HL register pair.

4.3 Extended BIOS calls

As the 256 TC has no boot ROM permanently mapped in memory, the functions otherwise found in ROM have been replaced by extensions to the BIOS. [illegible] extended BIOS (or XBIOS) function call, place the function number in the A register and [illegible] a RST [illegible] for [illegible] M [illegible] systems [illegible] and greater [illegible] special 256 TC calls, otherwise known as 'Subset C' functions.

4.3.1 Function 0 - Set CCP

```
Usage:    ld    a,0
          rst   28h
```

This function sets the CCP (ZCPR2 command line interface). The file 'CCP.SYS' must be present on the disk. This function has no effect until a BDOS function 0 (warm reset) is executed.

4.3.2. Function 1 - Set Shell

```
Usage:    ld    a,1
          rst   28h
```

This function sets the Shell. The file 'SHELL.SYS' must be present on the disk. This function has no effect until a BDOS function 0 (warm reset) is executed.

4.3.3 Function 2 - Number of drives

```
Usage:    ld    a,2
          rst   28h
```

This function returns into the accumulator the number of consecutively connected drives to the system starting with the A drive. This does not include either the M or L drives.

4.3.4 Function 3 - M: drive fitted

```
Usage       ld    a,3
            rst   28h
```

The function will return a 1 value into the A register if an M: drive is fitted, otherwise a 0 value will be returned.

4.3.5 Function 4 - Interrupt vector table base

```
Usage       ld    a,4
            rst   28h
```

This function returns the base address of the interrupt vector table into the HL register. The structure of the vector table is shown below.

```
pioa_vect    defw   par_int    ;pio a int vector
piob_vect    defw   int_rtn    ;pio b int vector
             defw   0          ;reserved for future use
             defw   0          ;reserved for future use
             defw   0          ;user defined interrupt
             defw   0          ;user defined interrupt
```

4.3.6 Function 5 - Pointer to command buffer

```
Usage       ld    a,5
            rst   28h
```

This function returns a pointer to an internal 128 byte extended command buffer. This buffer is only ever used by the Shell and CCP. The Z flag will be set if successful.

4.3.7. Function 6 Purge drive buffer

```
Usage.    ld   c,DRIVE CODE
          ld   a,6
          rst  28h
```

Will purge all the buffering for the drive indicated by DRIVE_CODE. The DRIVE_CODE is expressed as an integer (i.e. A drive = 0, B drive = 1, etc.

This function is useful when drives are being accessed directly from software of it is necessary to verify that data has been saved correctly

The function will only work on a hard drive when the FLUSH_FLAG byte (located at BIOS BASE+132h) is set to FFh. It should be reset to 0 after use

4.3.8. Function 7 Purge directory buffers

```
Usage     ld   c,DRIVE_CODE
          ld   a,7
          rst  28h
```

This allows for the directory buffers to be purged when disk swapping is taking place. The DRIVE CODE is expressed as an integer, (i.e. A: drive = 0, B: drive = 1, etc.. This function has no effect on a partitioned hard disk

4.3.9. Function 8 Return floppies

```
Usage     ld   a,8
          rst  28h
```

This function returns into the accumulator the number of floppy drives connected to the system

4.3.10. Function 9 Resense media

```
Usage:     ld    a,9
           rst   28h
```

Forces the BIOS to re-examine the disk media on the next access.

4.3.11 Function 10 Set baud rate

```
Usage:     ld    c,BAUD RATE CODE
           ld    a,10
           rst   28h
```

This function is used to set the baud rate of the RS232 port. The BAUD RATE CODEs are as follows:

```
0     300 baud
1     600 baud
2    1200 baud
3    2400 baud
4    4800 baud
```

This function may be used in conjunction with function 11 and 12. Upon booting the BIOS performs an internal initialisation of the RS232 port according to the parameters set by the INIT program.

4.3.12 Function 11 RS232 receive

```
Usage:     ld    a,11
           rst   28h
```

This function returns a received character in the A register if available and sets the Z flag, otherwise NZ.

4 3 13 Function 12 - RS232 send

```
Usage:     ld    c CHAR_TO SEND
           ld    a,12
           rst   28h
```

Once a character has been loaded into the C register, executing this function will cause it to be sent out the RS232 port

4 3 14 Function 13 - Set error mode

```
Usage.     ld    c ERROR_MODE
           ld    a 13
           rst   28h
```

This function allows you determine the method by which errors will be detected If ERROR MODE is equal to 0, then physical disk errors are detected but not passed to the BDOS avoiding BDOS errors Function 14 can then be used to check after each disk operation that no errors have occurred

If ERROR MODE is equal to FFh then all disk errors are passed to the BDOS and a BDOS error message will be returned this is the default state

4 3 15 Function 14 - Return disk status

```
Usage      ld    a 14
           rst   28h
```

This function is used exclusively with function 13. If a 0 value was returned to the A register then no errors on the last disk access If a non zero value is returned into A then a disk error has occurred

4 3 16 Function 15 - Keyboard test

```
Usage     ld    c,key code
          ld    a,15
          rst   028h
```

This function checks to see if the key whose code is in C (from table below) is currently down on the keyboard If it is down it returns with the Z flag set if it is up it returns with NZ status

should be noted that the fact that this function is available does not necessarily mean th the keyboard being used is a 256 TC style Therefore, ne should be careful not to make programs incompatible with existing keyboards by relying on keys which only exist on the new keyboard (e g , the <ALT> key. This does not mean that these keys should not be offered as an alternative for users with new keyboards, it is simply a matter of checking, for instance, both <CTRL-E> and the UP arrow code. Please note that if a key hich only exists on the new keyboard is checked on a machine using an old keyboard the key-up (NZ) status will always be returned

So that characters are not lost when a program is alling the KB TEST routine on a system with a new

put in a queue and returned via the standard BIOS/BDOS console input calls The minimum queue length implemented is 20 characters Because of this it would be a good idea to empty the key buffer by calling BDOS console in or something until no more keys are found before prompting the user for input if the KB_TEST cal. has been used previously - otherwise the user could get string of 20 or more of the last characters that he has been using to move the cursor around the screen with etc

Remember not to rely too heavily on keys b returned through console in that have been detected with KB TEST as on an old keyboard the queueing doesn't occur - so the key will only be returned by console-in if this is called whilst the key is actually down

Below is a list of the keys on the different keyboards and the corresponding codes to pass KB TEST to test if they are down. You will note that the first 64 codes are the same as the matrix codes for the keys on the old keyboard for obvious reasons of compatibility.

Keycode	Standard KB	256 TC KBD
0	@	
1	A	A
26	Z	Z
27	[{	
28	\ [illegible]	
29	] }	
30	^ ~	
31	DEL	DEL *
32	0	0
33	1 !	1
34	2 "	2 @
35	3 #	[illegible]
36	4 $	4 $
37	5 %	5 %
38	6 &	6
39	7 '	7 &
40	8 (	8 *
41	9)	[illegible]
42	: *	
43	; +	=
44	, <	, <
45	- =	[illegible]
46	. >	. >
47	/ ?	/ ?
48	ESC	ESC
49	BACK SPACE	CORRECT
50	TAB	TAB BACKTAB
51	LINE FEED	SELECT
52	RETURN	RETURN
53	LOCK	CAPS LOCK
54	BREAK	BRK #
55	SPACE	SPACE
56	ARROW_UP	ARROW UP
57	CTRL	CTRL
58	ARROW DOWN	ARROW DOWN
59	ARROW LEFT	ARROW LEFT
60	RESERVED	RESERVED
61	RESERVED	RESERVED
62	ARROW RIGHT	ARROW_RIGHT
63	SHIFT	SHIFT

(New Keyboard Only)

64	ALT
65	F1
66	F2
67	F3
68	F4
69	F5
70	F6
71	F7
72	F8
73	F9
74	F10
75	F11
76	F12
77	ESC
78	UP_ARROW
79	INS
80	DEL *
81	LEFT ARROW
82	DOWN ARROW
83	RIGHT ARROW
84	PRTSC #
85	7 (NUM PAD)
86	8 (NUM PAD)
87	9 (NUM PAD)
88	DIVIDE
89	4 (NUM PAD)
90	5 (NUM PAD)
91	6 (NUM PAD)
92	MULTIPLY
93	1 (NUM PAD)
94	2 (NUM PAD)
95	3 (NUM PAD)
96	SUBTRACT
97	0 (NUM PAD)
98	. (NUM PAD)
99	ADD

4.3.17. Function 16 - Program function keys

```
Usage      ld    hl,key table addr
           ld    a,16
           rst   28h
            AF, BC, DE and HL are corrupted
```

This function copies the table at (HL) to the internal programmable key buffer. On the standard [illegible] Microper keyboard [illegible] there are 12 [illegible] The data [illegible] all 12 function keys [illegible]

The format for the data table is as follows

```
key_table_addr
     defb   length_of_F1_data  ;4 in this case (test)
     defb   'test'             ;actual data for this key
     defb   length of F2 data  ;13 for 'fred was here'
     defb   'fred was here'


     defb   length of F12 data    ;last entry
     defb   [illegible]
```

4.3.18. Function 17 - Copy function key codes

```
Usage      ld    hl,func table buffer
           ld    a,17
           rst   28h
            AF, BC, DE and HL corrupted
```

This function copies the internal programmable key buffer to the [illegible] the program should provide at least 1.2K bytes [illegible] buffer [illegible] size [illegible] [illegible] returned in HL [illegible] a tempting [illegible] ensure the new key data is [illegible] than the value returned in HL

4.3.19 Function 18 Extended key check

Usage: ld a,18
rst 28h

This function checks the keyboard and returns an ASCII code into the A register [illegible] Z [illegible] the Z flag will be set if the key code is unavailable. This function [illegible] may [illegible] used [illegible] of keyboard is being used. If a 256 TC style keyboard is being used then I will be returned in the B register [illegible] 0 is returned for [illegible] returns all the normal keyboard codes except for:

Key	Unshifted	Shifted
Function keys	[illegible]	[illegible]h-DCh
Numeric keypad	B0h-B9h	B0h-B9h
Add (num)	normal (HBS)	normal (HBS)
Subtract (num)	normal (HBS)	normal (HBS)
Division (num)	normal (HBS)	normal (HBS)
Multiply (num)	normal (HBS)	CTRL-P (HBS)
Right arrow	CTRL-D (HBS)	CTRL-F (HBS)
Left arrow	CTRL-S (HBS)	CTRL-A (HBS)
Up arrow	CTRL-E (HBS)	CTRL-R (HBS)
Down arrow	CTRL-X (HBS)	CTRL-C (HBS)
SELECT	CTRL-J (HBS)	CTRL-J (HBS)
INS	CTRL-V (HBS)	CTRL-V (HBS)
TAB	[illegible]	8[illegible]h
[illegible]	7Fh	[illegible]h
[illegible]	[illegible]h	[illegible]Eh
CTRL-BRK	DFh	

HBS = High bit (bit 7) set

4.3.20 Function 19 Select speed

Usage: ld a,19
rst 28h

This function is only applicable to dual speed 256 TC's. Dual speed machines are not currently being offered by [illegible] Systems [illegible] this function will have no effect on the speed of the machine.

This function is only applicable until the next warm boot when the default speed (defined by Init) is set.

Function 19 requires one of the following values to be placed in the the C register

00h   set speed to 3.375 MHz

01h   set speed to 6.75 MHz
      (sets carry flag if switch not possible)

FFh - return current speed
      returns 0 in the A register if
      3.375MHz, 1 if 6.75MHz

4.4 Memory bank switching

The 256 TC uses 256k of dynamic random access memory (DRAM) with one 27128 (16k) EPROM. To enable all of these memory devices and the VDU to be used to advantage, the 256 TC uses a write only bank select port at 50h to control the memory map.

The 256k of DRAM memory is effectively broken to 4 x 64k banks, named banks 0

[illegible] accessed program/data memory. Under the BIOS bank 0 memory contains the CP/M program being run, the BDOS and the BIOS.

Banks 1 and 2 is intended to be used for other storage such as the M drive while bank 3 is used for disk caching data and tables.

The 27128 EPROM always resided from 8000h-BFFFh if the ROM is enabled. The VDU memory is also controlled by the bank select port. The VDU memory always occupies 4k of memory and can be disabled, enabled at 8000h-8FFFh or enabled at F000h-FFFFh.

The bank switching port (port 50h) uses 6 bits named S0-S5 which correspond to the data lines D0 through D5 in the data being written to the port. The meanings of these bits are as follows:

S0 This bit selects between DRAM bank 0 and bank 1 at 0000h-7FFFh. If S0 = 0 then bank 0 is selected, if S0 = 1 then bank 1 is selected.

S1 This bit selects the upper or lower 32k of DRAM bank 1 to appear at 0000h-7FFF when S0 = 1.

S2 This bit selects either the ROMs or the upper half of DRAM bank 0 at 8000h-FFFFh. When S2 = 0 the ROMs are enabled at 8000h-FFFFh. When S2 = 1, the upper 32k of DRAM bank 0 is enabled at 8000h-FFFFh.

S3 This bit enables/disables the 4k VDU memory block. If S3 = 0, video is enabled and overrides DRAM or ROM at the address. If S3 = 1 then the VDU memory block is disabled and any hidden RAM or ROM is made accessible.

S4 - This bit selects the address of the video between 8000h-8FFFh and F000h-FFFFh. If S4 = 0 (and S3 = 0) then video is enabled at F000h-FFFFh. If S4 = 1 (and S3 = 0) then video is enabled at 8000h-8FFFh.

S5 This bit is used to select between banks 0 and 1 and banks 2 and 3. With S5 = 1 it is simply a matter of using bits S0 and S1 to select any of the four 32k blocks from the second 128k at 0-7FFFh. Please note that this is the only bit that is not compatible with Premium 128k Microbees.

These bit definitions give rise to a set [illegible] useful values which set various memory maps.

00h Reset map - this configuration has DRAM bank [illegible] at 0000h-7FFFh, ROM at 8000h [illegible] and the VDU at F000h-FFFFh.

0Ch - BIOS map - This setup has DRAM bank 0 at 0000h-FFFFh so that VDU is switched out of the memory map. This is called the BIOS map because BIOS usually occupies the memory exposed at F000h when the VDU is switched out.

[illegible] VDR map - This has DRAM bank 0 at 0000h-FFFFh except that VDU is located at 8000h-8FFFh. This is how the BIOS routines above F000h access the screen.

0Dh - Bank A map - This configuration has the [illegible] 32k of DRAM bank [illegible] at [illegible] and the upper 32k of DRAM bank 0 at 8000h-FFFFh.

0Fh - Bank B map - This configuration has the upper 32k of DRAM bank 1 at 0000h-7FFFh and the upper 32k of DRAM bank 0 at 8000h-FFFFh.

4.5 Programming the keyboard

WARNING Attempting to run any program on the 256 TC that reprograms bit 1 of the PIO port B to an output bit will stop the 256 TC keyboard system software working. When reprogramming the PIO port B make sure you leave bit 1 (pin 28) as an input bit generating an interrupt with a logic 1.

The keyboard includes features such as two key rollover, function keys, a numeric keypad and an ergonomic design.

Because of differences between this keyboard and the previous keyboard, changes will have to be made in any currently existing, or future, software releases to avoid hardware dependency on the 6545 keyboard scanning system. We realise that many programmers have used routines to check if a particular key is pressed and that these routines depend on the current keyboard hardware. Unfortunately, the only way for these programs to work with the new keyboard system is for the hardware dependent pieces of code to be removed.

To facilitate easy change over and to avoid problems, should keyboard interfacing change in the future, Microbee Systems is providing calls in the BIOS for checking whether or not keys are down and for programming function keys. These extra calls are accessed via the currently existing extended BIOS (XBIOS) call structure. Obviously any program wishing to use these calls will need first to determine whether the calls are present under the particular version of BIOS and if so use them or if not use present methods of scanning etc (i.e., all users with new keyboards will be running BIOS versions with these new calls).

In order to determine if the BIOS your program is running under supports these calls (known as subset C of the XBIOS calls), the version number byte can be checked. In any given system the BIOS version number is found at BIOS_START+033h. Therefore, in an assembler program (XBIOS calls MUST be called from assembler), you would use code similar to the following to determine whether or not subset C if XBIOS is present.

```
Peek_t 05
            push      af
            push      de
            push      hl

            ld        hl,(0001h) ;address of warm start
                                 ;entry
            ld        de,( ...
            add       hl,de
            ld        e, ...
            ld        d,0          ;initial flag value=false
; 0x0 ... < 0x06
            cp        ...
            jr        c,test2
            cp        06h
            jr        nc,test2
            jr        Use_XBIOS...
; > 0x06 AND < 0x17
test2
            cp        07h
            jr        c,test3
            cp        17h
            jr        nc,test3
            jr        Use_XBIOSsc
; > 0x1D AND < 0x2F
test3
            cp        2eh
            jr        ...test4
            cp        2fh
            jr        nc,test4
            jr        Use_XBIOSsc
; > 0x42 AND < 0xBF
test4
            cp        44h
            jr        c,nxt
            cp        0bfh
            jr        nc,nxt
; If we get to here we must use XBIOS call 15 to check
; for key closed
Use_XBIOSsc:
            ld        d,0ffh       ;flag= ...
                                   available
; If we have jumped to this mark we use the 6545 h w to
;check for a key down
Use_6545
            ld        a,d
            ld        (New_BIOS),a ;=ffh if BIOS subset C
                                   ;=0 if not
            pop       hl
            pop       de
            pop       af
            ret
```

As you can see the code carries out this simple test on the version number

```
IF BIOS TYPE is      > 01h AND < 06h OR
                     > 06h AND < 17h OR
                     > 1Dh AND < 2Fh OR
                     > 41h AND < BFH

    THEN
            XBIOS SUBSET 6 IS SUPPORTED
    ELSE
            S   ARTWARE IS CLOSED ROUTINE TO SCAN 65
```

4.6.1 Basic and the Keyboard

Most Basic programs, whether they be in Basic or in assembler running with Basic, should not notice difference when running under the new keyboard, except for the fact that a few keys are different on the keyboard.

If your Basic program uses machine code sub-routines that directly scan the keyboard for pressed keys you will most likely need to make a couple of small changes. Some programmers call a non-standard location inside the Basic code itself (i.e., not in the jump table) which takes a key matrix code in the A register and returns the Z flag if this key is down or NZ if it is up. We have decided to retain this system and will be supporting this call location on all versions of Basic past, present and future; please note that this is the ONLY internal call that will be supported in future versions of Basic.

If your program calls this internal routine (address A50Ah) then you don't need to change your code. If your program does its own hardware scan then all you need do is change it to call A50Ah instead. Whilst you could call the XBIOS call detailed above this is not recommended as all the version checking is done automatically and the correct keyboard test is formed whether it be on an old or new keyboard.

The first version of Basic to support this A50Ah call and to support access to ALL keyboard types is version 6.31. If your program needs to access the programmable function keys please call the extended BIOS functions.

4.6. Screen programming.

One area where the 256 TC has improved [illegible] earlier release Microbees is in the screen area, [illegible] increased PCG and screen attributes

The 256 TC can provide programmers with up to 32K of PCG [illegible] The initial production versions of attribute RAM [illegible] [illegible] bit mapped gra[illegible] Basic [illegible] only possible with 32K of PCG RAM

The operation of the screen [illegible] you will be familiar [illegible] have [illegible] Microbee before the screen RAM contains the ASCII code [illegible] each character to be displayed on the screen and the [illegible] fore[illegible] and backg[illegible] is to be displayed in The attribute RAM, however, is exclusive to the Premium and the 256 TC

4.6.1 The 6545 CRT controller

The 6545 CRT controller is responsible for interfacing the 256 TC to cathode ray tube monitors The 6545 is a dedicated video processor responsible for the generation of horizontal and vertical timing signals

The 6545 has 23 internal registers 20 of which are used [illegible] aspects of the screen display, see figure 4.2 [illegible] To program a 6545 register one must first output the number of the register to be programmed to port 0Ch and then output to port [illegible] the value that is to be loaded into the register

As most programmers will only ever access the 6545 to change the screen format, we have presented tables giving values for common screen formats below and a summary of the registers implemented in the 6545 (figure 4.1).

Format	Register Values (hex)															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
64x16	6B	40	51	37	12	09	10	12	48	0F	2F	0F	00	00	00	00
64x17	6B	40	51	37	12	09	11	12	48	0F	0E	0F	00	00	00	00
40x25 wide	35	28	2D	24	1B	05	19	1A	48	0A	2A	0A	20	00	00	00
40x25 narrow	6B	28	46	37	1B	05	19	1A	48	0A	2A	0A	20	00	00	00
80x16	6B	50	59	37	12	09	10	12	48	0F	2F	0F	00	00	00	00
80x24	6B	50	58	37	1B	05	18	1A	48	0A	2A	0A	20	00	00	00
80x25	6B	50	58	37	1B	05	19	1A	48	0A	09	0A	20	00	00	00

Please note that all values presented above are hexadecimal. For the 40 column wide screen format to be enabled, the programmer must first halve the frequency of the video clock by issuing the instructions

```
LD    A
IN    A,(9
```

When re-entering any of the 80 or 64 column formats or the narrow 40 column format, the video clock frequency must be doubled by issuing the instructions

```
LD    A,0
IN    A,9
```

Full details of the 6545's internal registers can be found in either the Synertek data book (1983) or the Rockwell data book (1984).

Reg	Register Name	Information	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Address	Reg st No	W				A4	A3	A2	A1	A0
	Status		R	U	L	V					
R0	Horiz Total 1	# Chars	W								
R1	Horiz Displayed	# Chars	W								
R2	Horiz Sync Pos	# Chars	W								
R3	VSync HSync Widths	# Scan Lines # Char Times	W	V3	V2	V	V0	H3	H2	H1	H0
R4	Vert Total -1	# Chars Row	W								
R5	Vert Total Adj	# Scan Lines	W								
R6	Vert Displayed	# Char Rows	W								
R7	Vert Sync Pos	# Char Rows	W								
R8	Mode Control		W	U	[illegible]	[illegible]	[illegible]	T	RA	[illegible]	[illegible]
R9	Scan Lines 1	# Scan Lines	W								
R10	Cursor Start	Scan Line No.	W		B1	B0					
R11	Cursor End	Scan Line No.	~~R~~ W								
R12	Display Start Addr (H)		W								
R13	Display ~~End~~ START Addr (L)		W								
R14	Cursor Position (H)		R/W								
R15	Cursor Position (L)		R/W								
R16	Light Pen (H)		R								
R17	Light Pen (L)		R								
R18	Update Addr (H)		W								
R19	Update Addr (L)		W								
R31	Dummy Location										

NOTES: [ ] Designates an unused bit

Fig 4 1. 6545 internal register summary

4.6.2 The Attribute RAM

The Attribute RAM contains data which influences how the corresponding byte in the character RAM is displayed. Most importantly, it contains the bank number that contains the PCG data if the corresponding location in the character RAM is selecting a PCG character.

To explain this more fully, let assume that bank 0 of the character RAM, location 123h contains the code 85h which is for a PCG character (bit 7 is set). Let us also assume that bank 0 of the attribute RAM, at location 123h contains a 04h. The address for the data actually displayed would be obtained as:

```
 Character RAM                              Attribute RAM
Bank 0, Loc 123h                          Bank 0, Loc 123h

   Value 85h                                  Value 04h

 0 1 2 3 4 5 6 7                          0 1 2 3 4 5 6 7

 | | |                                    | | |
 | |           |  <-- Set to "1"          | |
 |       | | |        for PCG             | |
     | | | | |                            |
     | | | | |                            |   | |

   Value 05h                              Value 04h
```

This means that the data actually displayed on the screen would come from PCG bank 4, and would be the fifth character in that bank, which is at location 50h (each character is 16 bytes long). If bit 7 of the byte in the character RAM is not set, (i.e., PCG operation is not selected), then the attribute RAM is ignored and the character displayed comes from the character generator ROM.

Please note that it is possible to disable the function of the attribute RAM in hardware and make all PCG characters used come from bank 0. This is done by resetting bit 7 of the "Video Memory Latch" (VML) located at port 1Ch. This is the default mode upon power up or reset. This was included to allow emulation of an earlier version Microbee machine from the power up condition. The same effect could be done in software by filling the attribute RAM with the same bank number for all bytes on the screen.

As each bank of PCG RAM can contain up to 128 characters [illegible] there are [illegible] 128 = [illegible] banks of PCG RAM [illegible] versions of the 256 [illegible] This means that to select a [illegible] bank for each character only requires [illegible] bits of each attribute RAM character [illegible] RAM [illegible] the bit is reserved [illegible] what are the other [illegible] are used for hardware attributes such as [illegible] and flashing

4 6 2 Banking and memory locations

In the 256 TC, locations 0F000h to 0FFFFh are reserved for use by the VDU RAM. As there is obviously more than 4k of RAM that has to be accessed [illegible] banking arrangement is necessary. The banking [illegible] on the 256 TC has been made as compatible as [illegible] with earlier Microbee models

There are three banks of VDU memory that can appear [illegible] fig 4 12. The first is the character RAM and this is the default on power up [illegible] The second is the character generator ROM. This is selected by setting [illegible] port 0BA to a one. It is deselected by [illegible] the same bit to a zero. The third bank of RAM is the [illegible] RAM. This is selected by setting bit 4 of port [illegible] the [illegible] latch. It is deselected by resetting the same bit

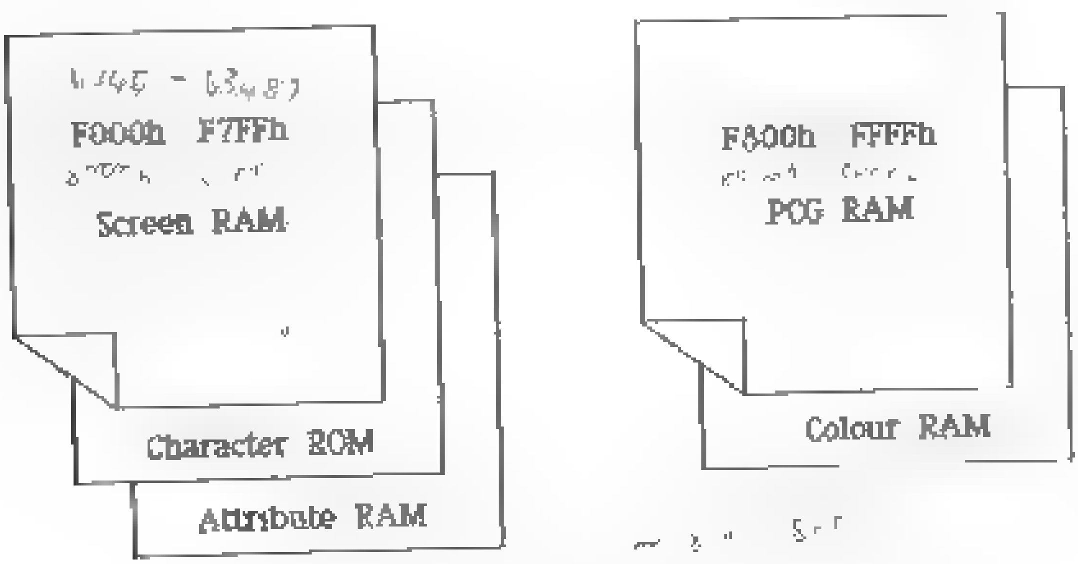


Fig 4 2 VDU memory banking

Between 0F800h and 0FFFFh, two banks of VDM RAM can be selected (figure 6.1). Firstly, the PCG RAM normally resides here. The colour RAM, however, can be made to overlay this area by setting bit 6 of port 08h to a one, and can disabled by setting the same bit to a zero.

The case of either the attribute, screen, colour or PCG RAM being enabled on a 256 TC, there is also a mechanism for selecting which bank of the particular type of RAM you are talking to (e.g., you may want to access the third bank of PCG RAM). Also, in the case of having 8K of screen, attribute or colour RAM, you may want to access another of the four possible banks of these RAMs. To do this, the bank that you wish to access should be selected by setting the number on the bottom four bits of the port 1Ch (the VML latch).

At this point, please note that the bank of screen [illegible] attribute [illegible] PCG RAM [illegible] [illegible] those that are displayed but those accessible by the processor. If 256 TC mode is enabled (i.e., bit 7 of the VML is set), the 6545 screen start address will determine the first address in screen memory being displayed on the screen.

In the case of only having a single 2k bank of screen [illegible] attribute RAM [illegible] be the case, as the addressing will "wrap" around, and use the same locations in any case.

6.4 Hardware attributes

The 256 TC has additional circuitry to allow programmers to set flashing and inverse attributes to characters, without the need of using extra PCG characters. Hardware inverse and flashing may be attributed to characters by setting bits 6 and 7, respectively, of the corresponding byte of attribute RAM.

The Turbo Pascal program listed below demonstrates the use of bits 6 and 7 of the attribute RAM to set inverse and flashing attributes.

```
{256 TC hardware inverse/flashing demonstration}

var
  i  : byte;
  ch : char;

begin
  port[$1C] := 144;            {Latch the Attribute RAM}
  for i := 0 to 23 do
    mem[$f000+i] := 64;       {Set Bit 6 for 1st message}
  for i := 0 to 24 do
    mem[$f000+80+i] := 128;  {Set Bit 7 for 2nd message}
  for i := 0 to 32 do
    mem[$f000+2*80+i] := 192;
                              {Set bits 6&7 - 3rd message}
  port[$1C] := 128;       {Latch Screen RAM - a+ enabled}
  clrscr;
  writeln('This is hardware INVERSE');
  writeln('This is hardware FLASHING');
  writeln('This is hardware INVERSE/FLASHING');
  read(kbd,ch);
  port[$1C] := 0;                         {Disable a+ mode}
end.
```

4.6.5 Colour RAM

The 256 TC uses a single byte of colour information per character location on the screen, with the colour RAM [illegible] at [illegible] by setting bit 6 of port 08h to a one. The function of the colour RAM bits are as follows

Foreground		Background	
Bit 0 -	Red	Bit 4 -	Red
Bit 1 -	Green	Bit 5 -	Green
Bit 2 -	Blue	Bit 6 -	Blue
Bit 3 -	Intensity	Bit 7 -	Intensity

The function of the intensity bit is not as one might expect. Half intensity is enabled by setting the bit to a zero and full intensity by setting it to a one.

4.6.6 Summary

Video Memory Latch - port 1Ch

Bit 0 - 3	PCG Char Attr/Col Bank Select
Bit 4	Attribute RAM Enable
Bit 5	BC85 ROM "A" Select (not Used
Bit 6	Reserved - always set to zero
Bit 7	Alphat mode enable

Attribute RAM

Bit 0 - 3	PCG Bank Enable	
Bit 4	Char ROM select 4K blocks	Link L16
Bit 5	Char ROM select 8k blocks	Link L6 7
Bit 6	Inverse (hardware select	
Bit 7	Flashing (hardware select	

Colour RAM

Bit 0 - Foreground Red
Bit 1 - Foreground Green
Bit 2 - Foreground Blue
Bit 3 - Foreground Half Int
Bit 4 - Background Red
Bit 5 - Background Green
Bit 6 - Background Blue
Bit 7 - Background Half Int

Colours

Half Intensity Colours	Full Intensity Colours
0 - Black	8 - Dark Grey
1 - Red II	9 - Red
2 - Green II	10 - Green
3 - Yellow II (Brown)	11 - Yellow
4 - Blue II	12 - Blue
5 - Magenta II	13 - Magenta
6 - Cyan II	14 - Cyan
7 - Light Grey	15 - White

4.7 Programming the PIO

The PIO is a Z80 family device, used to provide a TTL-compatible interface between the Z80 data bus and peripherals.

The PIO is a dual port device and is used in the 256 TC to interface the parallel, serial and cassette ports to the CPU. Port A of the PIO is reserved exclusively for the parallel port. The parallel port is connected directly to PIO and so is available for use by the programmer in any of the four PIO operating modes. Port B is configured permanently by the 256 TC's hardware for 'bit mode' operation, providing the serial and cassette ports and keyboard activity detection. Further details can be obtained from the 'Zilog Components Data Book' (1983) or the SGS 'Z80 Microprocessor Family' data book (1982).

4.7.1 Operating modes

The PIO offers the programmer four different operational modes. Mode 0 (or 'Output mode') allows bytes of data to written to the port. An active (high) Ready output is used by the PIO to indicate to the peripheral that data is ready to be transmitted. Once data has been transferred, the peripheral should respond with an active (low) Strobe input, which will in turn generate an interrupt if interrupts have been enabled.

Mode 1, 'Input mode', uses an active (high) Ready signal to indicate to the peripheral that data is ready to be read. The peripheral should then place data onto the I/O lines and strobe the port by taking the Strobe signal low. This causes the data to be latched into the input registers, resets the Ready line and triggers the interrupt request, if enabled.

Mode 2 allows 'bi-directional' data I/O. This mode requires that port B handshake and interrupt signals be used when data is being read. As all versions of the Microbee use port B to interface RS232 devices and a cassette recorder, it is not possible to support this mode. Instead, the spare port B data line (DB7) is connected to the parallel port (pin 15) and may be used, with additional hardware, as a data direction indicator (this is implemented with the Starnet network! The Starnet sets DB7 high to output data to bi-directional data buffer connected to parallel port. When DB7 is sent low, the direction latch on the buffer is changed and data can be read.

Mode 3 is 'Bit' mode. This mode allows you to configure any of the data bits for either input or output. 'Bit' mode does not use either the Ready or Strobe signals; instead it can generate an interrupt whenever the state of an input bit is changed to a programmable level. The interrupt mode is configured when mode 3 is selected.

4.7.2 PIO configuration

The 256 TC has dedicated ports 00h and 02h as the PIO port A and B control ports respectively. The control ports are used to configure the operation of the PIO's data ports.

When configuring a port for modes 0, 1 or 2, the PIO requires that you send to its relevant control port a 'mode control word' (figure 4.3) which may be followed by an 'interrupt vector word', if interrupts are to be enabled (figure 4.3).

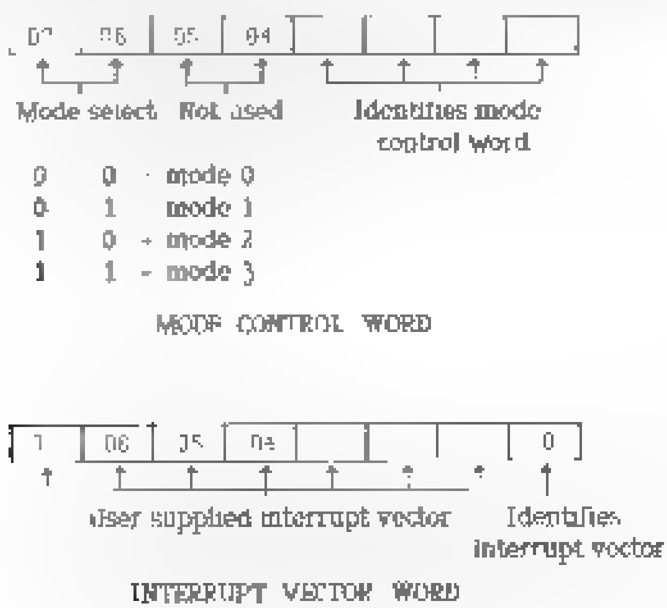


Fig 4.3 Mode control & interrupt vector words

When programming a PIO port for 'bit' mode operation (mode 3), the PIO requires that you send it three additional words to those described above.

The 'mode control word' must be followed immediately by an I/O register control word. This byte defines the data direction of each of the eight data lines (figure 4.4).

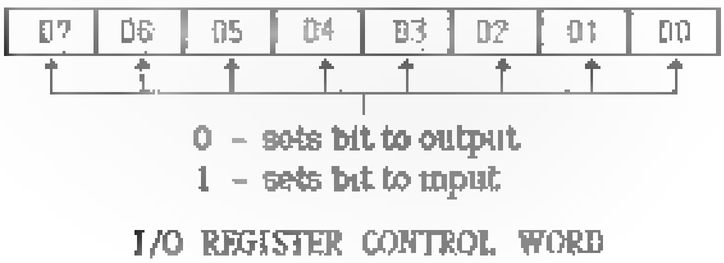
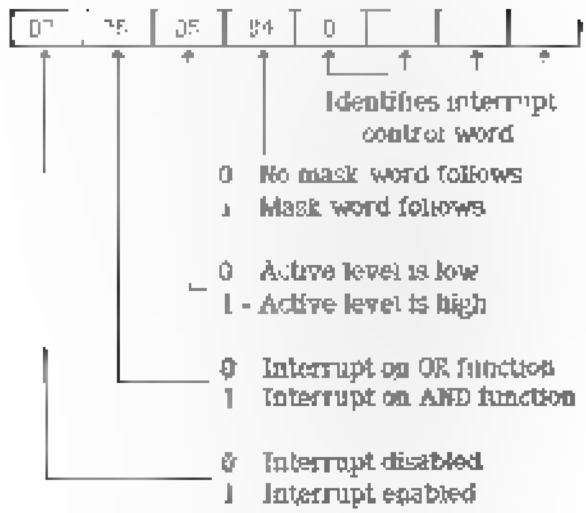


Fig. 4.4 The I/O register control

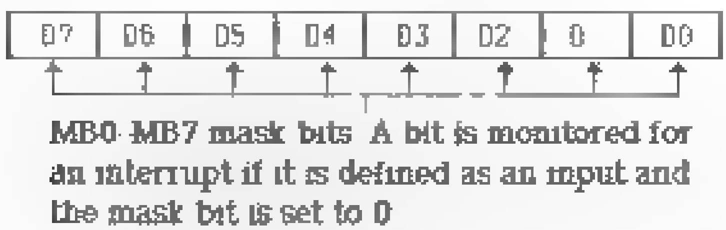
The 'interrupt control word' defines the conditions when interrupts will be generated (figure 4.5). Because handshaking signals are not used in mode 3, it is necessary to generate interrupts as a logic function of the input signal levels. When the AND function is selected, an interrupt will be generated if all input bits change to an active level. Selecting the OR function will cause an interrupt to be generated when any of the input bits change to an active level.



INTERRUPT CONTROL WORD

4 5 The Interrupt control word

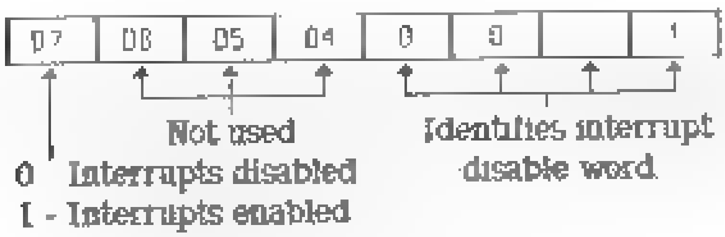
If you are not using all the data lines, then set D4 in the 'interrupt control word' to 1 This indicates to the PIO that a mask control word' is about to follow A 'mask control word' simply indicates to the PIO which data lines are not going to be used (figure 4 6 Any bits containing a zero value in the 'mask control word' will be regarded by the PIO as being masked



MASK CONTROL WORD

Fig 4 6 The mask control word

There is one final control word that may be used with the PIO This is 'interrupt disable word' (figure 4 7) This word allows you to enable or disable interrupts, at any point of operation, without having to change any portions of the 'interrupt control word



INTERRUPT DISABLE WORD

Fig. 4 7. The interrupt disable word

As we have mentioned previously, port B of the 256 TC's PIO has additional hardware attached to it and so it is not recommended that any programmer attempt to change the direction of any of the bits other than DB1 (bit 1) This bit is now used by the keyboard circuitry to detect keyboard activity and is set as an input If you wish to use the cassette output line, you may reconfigure this bit to an output but you must re program it to an input once you have finished sending data to the cassette The keyboard will not operate unless DB1 is set as an input

The default configuration of port B on the 256 TC is described below

PIO port B default 'bit mode' configuration

BIT	SIGNAL	CONNECTOR	DIRECTION
DB0	Cassette IN	X11/pin 5	IN
DB1	Keyboard	X11/pin 3	IN
DB2	Clock	X2 pins 20&24	OUT
DB3	CTS	X2 pin 5	IN
DB4	RxD	X2 pin 3	IN
DB5	TxD	X2/pin 2	OUT
DB6	Sound	X6/pin 1	OUT
DB7	Direction	X11/pin 14	OUT

4.8 The disk formats

The Microbee 256 TC BIOS provides for two types of disk formats on 3.5" disks. The first format is the 80 track format (400k) which uses SSDD diskettes. This format is provided to maintain compatibility with disks created on earlier versions of the Microbee Computer-In A Book, etc.

An 80 track (800k) format, using DSDD diskettes, is also provided. Please note that the 256 TC BIOS allows either format to be used in any combinations (i.e., it is possible to use a 400k disk in one drive whilst having an 800k drive in the other drive.

The 256 TC CP/M disk formats use the MFM recording scheme with a system 34 (WD2793) organisation. All sides on one cylinder use the same track number. The outermost track is numbered as 0, the innermost track is either 39 or 79 decimal). The format sets the 'side' byte to zero on both sides of the diskette.

Sector numbering is dependent upon the format being used. The single sided 400k format uses 10 sectors of 512 bytes. These are formatted linearly with values 1 through 10. The double sided (800k) format uses sector numbers 1-10 on cylinder 0 and sector numbers 21-30 on cylinders 1-79.

The physical sectors on the disk are interleaved by CP/M using the following translation tables.

```
Sector numbering 1-10

xltd        db    2,5,8
            db    1,4,7,10
            db    3,6,9

Sector numbering 21-30

      sector skew with offset of 20 for 80 track disks
x  [illegible]   db    22,25,28
            db    21,24,27,30
            db    23,26,29
```

To access the physical sectors of a disk, BIOS calls should be used in preference to to direct disk controller access.

A list of the disk parameter blocks for both formats are listed below

D.P.B. for 3.5" 80 track single sided disks

```
  format d. Double density, 10*512 byte sectors
  for most work giving 410k per drive
  this one is a double sided cylinder format
  shifted 3 sector skew should allow 6 revs per seek
  for sequential reading writing
  This is spread over 80 tracks on one side for 3.5"
  drives and 40 tracks on two sides for 5.25" drives
          db    xltd xlt base    ;sector translate table
;0x02 is the "don t correct for 40 track IBM style" bit
          db    0x02              organised on side 0 only,
          db    3                ;sector mask

          dw    40               ;sectors per track
          db    4                ;block shift factor
          db    15               ;block mask
          db    1                 extent mask
          dw    194               blocks per disk DS drive
          dw    127              ;# directory entries
          db    192               alloc 0
          db    0                ;al   1
          dw    32               ;dir check vector size
          dw    2                ;system track offset
```

D.P.B. for 3.5" 80 track double sided disks

```
  format b  Double density, 10*512 byte sectors
  80 tracks per side  double-sided format using
  78 data tracks to allow same organisation for system
  track storage as SS
          db    xltb xlt_base     offset from xlt base
          db    5                ;80 track + up/down
          db    3                ;sector mask

          dw    40               ;sectors per track
          db    5                ;block shift factor
          db    0x1f             ;block mask
          db    0x01             ;extent mask
          dw    194              ;blocks per disk  DS
          dw    127               # directory entries
          db    0x80             ;alloc 0
          db    0                 alloc 1
          dw    32               ;dir check vector size
          dw    4                ;system track offset
```

The actual DPB starts after the semicolon in each listing. The three items before the start of the DPB used by the BIOS to control various modes of operation. The main item of interest is the mode byte, the second of the three bytes. The various bits in this flag byte have the following meanings:

Bit 0   Up/Down format

If this bit is set the disk is organised in cylinder format, alternating between sides as the disk is traversed from the outside to the inside.

Logical track	0	1	2	3	4	5	6	7
Cylinder	0	0	1	1	2	2	3	3
Side	0	1	0	1	0	1	0	1

If the 'up/down format' bit is zero then the tracks are numbered first inwards on side 0 and outwards on side 1.

Logical track	0	1	2	3		36	37	38	39
Cylinder	0	1	2	3	..	3	2	1	0
Side	0	0	0	0	..	1	1	1	1

Bit 1   Disable 40 track folding

If this bit is set then the folding at track 40 will be disabled allowing access to all 80 tracks on a single sided disk.

Bit 2   Enable access to 800k disk

If this bit is set then the format is an 800k format disk and access to both sides of an 80 track disk will be possible if the up/down bit is set.

4.9 The DGOS tape format

The 256 TC, like other versions of Z80 based Microbee computers, supports an extended vers[illegible] of the DGOS (David Griffiths Operating System) cassette [illegible] format.

This extension of the DGOS format is iden[illegible] original except for [illegible] which are used to indicate tape speed.

Bytes	Name	Description
16	NULLS	At least 16 NULLS (usually mo[illegible]
1	SOH	Start of header character [illegible]
6	NAME	Left justified name. Maximum of 6 chars, padded with NULLS
1	TYPE	File type character
2	LENGTH	Length of file in Z80 format (i.e., LSB, MSB format)
2	LADDR	Load address of file
2	AADDR	Auto start address [illegible] FL EXEC is <> 0
1	SPEED	0 = 300 baud, non zero = 1200 baud
1	FL EXEC	FFh = M.L. to auto start, else 0
1	SPARE	
1	CK	Checksum byte for header
256	DATA	256 bytes for each full block
1	CK	Checksum for block
	.... DATA BLOCKS	
?	DATA	Last block, length <= 256 bytes
1	CRC	CRC for last block

The checksum is calculated using the following formula:

```
c = 0
For each data byte "d" in a block do
c = CPL d - c
```

4.9 The DGOS tape format

The 256 TC, like other versions of Z80 based Microbee computers, supports an extended version of the DGOS (David Griffiths Operating System) cassette tape format.

This extension of the DGOS format is identical to the original except for two bytes [illegible] the original which are used to indicate tape speed.

Bytes	Name	Description
16	NULLS	At least 16 NULLS (usually more)
1	SOH	Start of header character (01h)
6	NAME	Left justified name. Maximum of 6 chars, padded with NULLS
1	TYPE	File type character
2	LENGTH	Length of file in Z80 format (i.e., LSB, MSB format)
2	LADDR	Load address of file
2	AADDR	Auto start address if FL EXEC is <> 0
1	SPEED	0 = 300 baud non zero = 1200 baud
1	FL EXEC	FFh = M L to auto start, else 0
1	SPARE	
1	CK	Checksum byte for header
256	DATA	256 bytes for each full block
1	CK	Checksum for block
	..... DATA BLOCKS	
?	DATA	Last block, length <= 256 bytes
1	CRC	CRC for last block

The checksum is calculated using the following formula:

```
c = 0
For each data byte "d" in a block do
c = CPL d - c
```

Connections

5 Connections

This chapter details the signals connected to each of the physical connectors and the port maps

5 1 Finding Pin 1

Looking at the board from the I,O edge with the mask overlay the correct way up

Connectors X1, X2, X5, X6 X7 X8, X9, X10 X11 X13, X14 are numbered from right to left, the reverse of the IC's

6 5 4 3 2 1

X X X X X second row X1 X2 & X10

Connectors X3 X4 X12, X16 are numbered from left to right

2 4 6 8 10

1 3 5 7 9

Connector X15 numbered the same as a 14 pin IC

5 2, Parallel port DB15 X1

1	+5 VOLTS	9.	+12 same as input voltage
2	Data A7	10	Data A6
3	Data A5	11	Data A4
4	Data A3	12	Data A2
5	Data A1	13	Data A0
6	Printer strobe	14.	Data B7 for Starnet
7	ARDY	15	ASTB
8	0V or ground		

5 3 Serial port DB25 - X2

1	N C	14	N C
2	Transmit TXD	15	N C
	Receive RXD	16	Light Pen * 6545
4	N/C	1	N C
5	Clear to send CTS	18	N C
6	N C	19	N C
7	0 volts (ground)	20	[illegible]
8	Pullup resistor +5 volts	21	N/C
	+12 volts	22	N/C
10	N C	23	N C
1	N/C	24	CLK link to [illegible]
12	N/C	25	N C
13	N/C		

5 4 Hard disk interface 40 way - X3

1	D0	2	0 Volts
3	D1	4	
5	D2	6	
7	D3	8	
9	D4	10	
11	D5	12	
13	D6	14	
15	D7	16	
17	A0	18	
19	A1	20	
21	A2	22	"
23	Port 40 *	24	"
25	ZWR *	26	"
27	RD *	28	"
29	System clock 3 375MHz	30	"
31	M1 *	32	"
33	IORQ *	34	"
35	HALT *	36	INT *
37	F SIDE	38	[illegible] *
39	RESET *	40	I.C to [illegible]

5.5 Floppy disk interface 34 way - X4

1	0 volts	2	N/C
3		4	link to ready X4
5		6	D select
[illegible]		8	Index Pulse
9		10	A select
11		12	B select
13	"	14	C select
15	"	16	Head Load
[illegible]	"	18	Direction
19	"	20	Step
21	"	22	Write Data
23	"	24	Write Gate
25		26	Track 00
27	"	28	Write Protect
29		30	Raw Read
31	"	32	Side Select
[illegible]	"	34	N/C

5.6 Floppy drive B, power connector X5

1 +5 Volts
2 0 Volts
3 +12 Volts
4 N/C

5.7 Speaker connector X6

1 Audio signal
2 0 Volts

5.8 Floppy drive A, power connector

1 +5 Volts
2 0 Volts
[illegible] +12 Volts
4 N/C

5 9 Floppy drive alignment test points X8

	0 Volts			
2	TP1	adjust RV2	U34 pin 29 and 1	5us
3.	TP2	adjust RV1	U34 pin 31	250ns
4	TP3	adjust CV1	U34 pin 16	2us
5.	TEST	connect to pin 1 to make adjustments		
6	+5 Volts			

Please refer to chapter 3

5 10 Power connector to power supply - X9

1 +5 volts from power supply
2 0 volts
3 +12 volts to power supply
4 0 volts

5 11. Colour port DB9 X10

1 0 volts
2 0 volts
3 RED TTL
4 GREEN TTL
5 BLUE TTL
6 Intensity
7 Comp Sync (Neg
8 Horizontal Sync (pos)
9 Vertical Sync pos

5 12 Main plug 5 pin DIN X11

1 +12 volts @ 1.5 amps
2 0 volts
3 Cassette "SAVE" @
4 Composite Video (monochrome)
5 Cassette "LOAD" @
SHIELD connected to 0 volts

5.13. Expansion port 50 way   X12

1 2	+12V Input/Output	Power to or from Unit
3 4	0 volts	system ground
5	Port 68H or 14H * (O	select with link L14
6	IEO or IEI * (I or O	select with link L13
7	WAIT * (I	used to slow down CPU
8	[illegible] * (O	
9	INT * (I	interrupt input
10	HALT * (O	indicates CPU halted
11	NMI *	non maskable interrupt
12	CPU CLOCK (O	3.375MHz
13	BUSAK * (O)	acknowledge bus request
14	BUSRQ * (I	request use of bus
15 16	+5 volts	from main supply
17	RFSH * (O)	refresh on DRAM
18	RESET * (I/O	open collector driven
19	M1 * (O)	instruction fetch
20	IORQ * (O)	indicates port access
21	RD * (O	read strobe
22	AWR * (O)	deglitched write stb
23	PHANTOM * (	
24	MREQ * (O	indicates memory access
25	Address 14	
26	Address 15	
27	Address 12	
28	Address 13	
29	Address 10	
30	Address 11	
31	Address 8	
32	Address 9	
33	Address 6	
34	Address [illegible]	
35	Address 4	
36	Address 5	
37	Address [illegible]	
38	Address 3	
39	Address [illegible]	
40	Address 1	
41	DATA [illegible]	
42	0 volts	
43	DATA 5	
44	DATA 6	
45	DATA 3	
46	DATA 4	
47	DATA [illegible]	
48	DATA 2	
49	0 volts	
50	DATA 0	

Notes : (O) = output (I) = input (*) = active low

5.14. Keyboard 15 way - X13

1	U60 3870 pin 17
2	" " pin 18
3	" " pin 19
4	" " pin 22
5	" " pin 23
6	" " pin 24
7	" " pin 25
8	" " pin 3
9	" " pin 4
10	" " pin 5
11	" " pin 6
12	" " pin 34
13	" " pin 35
14	" " pin 36
15	" " pin 37

5.15. Keyboard 8 way - X14.

1	U60 3870 pin [illegible]
2	" " pin [illegible]
3	" " pin 3[illegible]
4	" " pin 3[illegible]
5	" " pin 3[illegible]
6	" " pin 28
7	" " pin 27
8	" " pin 29

5.16. Serial keyboard 14 pin DIP header X15

1	0 volts	14.	+5 volts
2	0 volts	13	N/C
3.	N/C	12	KCOM * IO
4	LOAD EN * O	11	KSTB I
5	N/C	10	DATA EN IO
6	N/C	9	N/C
7	N/C	8	N/C

5.17. Serial port internal connection X16

1	0 volts	2	clock CLK
3	0 volts	4.	+12 volts
5	0 volts	6	clear to send CTS
7	0 volts	8.	receive RXD
9	0 volts	10.	transmit TXD

5 18. Port maps

The following details the port maps of the 256 TC. Some of these ports are only used if the related optional parts are fitted

Port	Function
00h	PIO port A data port
01h	PIO port A control port
02h	PIO port B data port bit 0 Cassette data in bit 1 a. Keyboard interrupt internal b. Keyboard data external c Cassette data out bit 2 RS232 CLOCK or DTR line bit 3 RS232 Clear To Send (CTS 0 = true bit 4 RS232 Receive Data (RXD 0 = mark) bit 5 RS232 Transmit Data (TXD 1 = mark bit 6 Speaker bit (1 = on bit 7 a Network data direction b RTC interrupt bit c VSYNC interrupt
03h	PIO port B control port
04h	RTC address port
06h	RTC write port
07h	RTC read port
08h	COLOUR control port: bit 6 COLOUR RAM enable (0 = PCG, 1 = colour, F800h FFFFh
09h	40 column video mode INPUT from port

```
LD   a 1
IN   a,(9     = 40 column

LD   a,0
IN   a,(9)    = 80 column
```

0Ah — Decoded but not Used
Used in ROM Microbees for PAKs

0Bh — Character ROM CPU access makes char[illegible]ter generator ROM appear from F000h to F7FFh when bit 0 of this port is set

0Ch — 6545 CRTC control port

0Dh — 6545 CRTC data port

0Eh [illegible] — not USED

10h [illegible] — Sound chip 76489AN

14h [illegible]h — Decoded - possible use for second

18h [illegible]Bh — Internal Keyboard

1Ch [illegible]h — Video Memory Latch

- bit 0 3 — PCG RAM screen RAM colour RAM at [illegible]bute RAM bank select
- bit 4 — Enable reading [illegible] attribute RAM [illegible] bute RAM appear from F0[illegible]h [illegible] F7FFh
- bit 5 — Do Not Use Should always be zero PC85 ROM 'A' select
- bit 6 — Reserved always set to 0
- bit 7 — Extended PCG character sets enable

40h — Disk controller command/status register

41h — Disk controller track register

42h — Disk controller sector register

43h — Disk controller data register

48h — Disk controller select/side DD latch bit assignments (write only)

- bit 0 — LSB of drive address
- bit 1 — MSB of drive address
- bit 2 — Side select 0=side 0 1=side [illegible]

bit 3 — DD select
0 = single density

bit 7 — Controller TRANSFER status bit - gives INTRQ or DRQ

50h — DRAM active block selection (all set to 0 on reset). Bits 0, 1 select which 32K bank of RAM appears from 0000h to 7FFFh

Bit 1	Bit 0	RAM at 0000h-7FFFh
0	0	bank 0 block 0
0	1	bank 0 block 1
1	0	bank 1 block 0
1	1	bank 1 block 1

bit 2 — Disables ROMS. When low, ROM (U38) appears from 8000h to 8FFFh

bit 3 — Disables video RAM when high

bit 4 — When low, video RAM appears at F000h to FFFFh irrespective of RAM select bits 0 and 1. When high, video RAM is from 8000h to 8FFFh, again irrespective of bits 0 and 1)

bit 5 — When set, enables second 128k DRAM into the low 32k area of memory

58h — bit 0 — Low selects internal floppy disk drive
High selects external floppy or hard disk drive

bit 1 — External keyboard control
High latches KCOM *

bit 2 — High latches LOAD EN *

bit 3 — High latches DATA EN *

60h — File server selection

68h — Reserved for 8530 SCC

70h — Decoded - not USED

78h — Decoded - not USED

5.19 Link selections

With the standard model 256 TC, only one link has been changed L8 the flashing rate link. Links will only have to be changed when certain options are fitted.

Legend : -- or = Linked with track
= Permanently connected

L1. A B Join A B for RS232 rec with U1.

L2. A B Cut A,B if sound IC U24 installed

L3. A. B. C A=VSYNC B=RTC INT C=NET D.R
F . E . .D FED = Pin 34 U1 PIO

L4 A B. Link READY signal U34 to X4, Pin 4

L5. A B C D ADJ KLT timing on the 279. U34
H G F E

L6 A C CHARACTER ROM SELECT 4732/2764/27128
D F 4732 link D G, C--F standard
G J 2732/2764 link A D, F J
27128 link A D F--J + L7

L7 A -- B. C CHARACTER ROM 27128
CUT A-B join B C

L8 A B . C D . E Change flashing rates
F G H J K

L9 A B -- C REMOVE BUFFER cut B C join B—A

L10 A B. C RESET CPU only on POWER ON
D E -- F Change HSYNC Colour port to COMP sync

L11 A B A=NMI B=Delayed Reset from SW.

Connections

L12	A -- B	C	Cut A,B and join B- C for a 6264 Colour RAM
			Cut D E and join E- F for a 6264 Screen RAM
	G · H	J	Cut G,H and join H - J for a 6264 Attribute RAM
L13	A	B -- C	A=IEI U1, pin 24 B=X12 pin 6 (50 way) C=IEO U1, pin 22
L14	A	B - C	A=Port 14 B=X12 pin 5 (50 way Port 68
L16	A	B	C	A=Attribute Bit A4 (LA12) B=U53 pin 2 A12 C=+5 Volts (High 4K block)
		D		D=0 Volts (Low 4k block)

5.20 Factory modifications

This sections lists all of the modifications being made to all 256 TCs as of the date of publication

5.20.1 Dual speed disable

As the dual speed option cannot be supported at the present time boards should have the track connected to pin 15 of [illegible] component side. The print through hole should then be connected to pin 8 of [illegible]

5.20.2 PCG write delay

Disconnect pin 12 of U86 from the board and join the print through hole closest to pin 3 of X15 to the print through hole [illegible]

5.20.3. M1 delay line

Cut the track under R58 (solder side) next to the leg closest to pin 1 of U76

5.20.4. L296 ground strap

A ground strap must be connected from the [illegible] screw of the L296 regulator to [illegible] 1[illegible]5 [illegible] CB2

5.20.5. F-354L drive earth strap

The 0V line and frame ground of the disk drives must be shorted. Use a short length of wire between pin P[illegible] on the disk drive PCB and the track connected [illegible] the nearest mounting screw

5.20.6. PCG chip select

Some early release had an 74HC157 fitted as U63. This should be replaced by a 74ALS157

Component Reference

6. Component Reference

This chapter provides you with a complete reference for every component used in the 256 TC.

6.1 Mainboard - 970 755

Part no.	Component	Description	Comp. Ref	
650 508	MBR508	256 TC PCB		
600 081	Z80 PIO	I O interface	U1	SOCKET
610 008HC	74HC08	AND gate	U2	
610 032HC	74HC32	OR gate	U3	
[illegible]	[illegible]	Hex inverter	U4	
610 074HC	74HC74	Dual D type Flip-Flop	U5	
610 000HC	74HC00	NAND gate	U6	
610 004HC	74HC04	Hex inverter	U7	
[illegible]	[illegible]	Three Input NAND	U8	
618 256	4256	256K DRAM	U9	
618 256	4256	256K DRAM	U10	
618 256	4256	256K DRAM	U11	
618 256	4256	256K DRAM	U12	
618 701	751701	RS232 TxD/RxD	U13	*RS232
610 161HC	74HC161	4-bit Binary Counter	U14	
[illegible]	[illegible]	Octal BUFFER	U15	
600 080	Z80 CPU	Central Processor	U16	SOCKET
[illegible]	[illegible]	Octal BUFFER	U17	
618 256	4256	256K DRAM	U18	
618 256	4256	256K DRAM	U19	
618 256	4256	256K DRAM	U20	
618 256	4256	256K DRAM	U21	
		[illegible]	U22	
		[illegible]	U23	
611 489	76489AN	Sound Generator	U24	*Sound
600 660	TSC7660	DC to DC converter	U25	*RS232
610 004	74LS04	Hex inverter	U26	
610 541HCT	74HCT541	Octal BUFFER	U27	
610 174HC	74HC174	Hex D Flip Flop	U28	
610 157HC	74HC157	Multiplexer	U29	
610 157HC	74HC157	Multiplexer	U30	
[illegible]	[illegible]	Multiplexer	U31	
610 032HC	74HC32	OR gate	U32	
610 125HC	74HC125	3-State BUFFER	U33	
600 793	2793	Disk Controller	U34	SOCKET
610 074HC	74HC74	Dual D type Flip-Flop	U35	
610 175HC	74HC175	Quad D Flip Flop	U36	
610 645HCT	74HCT645	Octal Transceiver	U37	
		BOOT ROM 27128	U38	SOCKET
610 014S	PAL 14L8	PAL Silver	U39	SOCKET
610 032HC	74HC32	OR gate	U40	

Component Reference

Store No	Component	Description	Comp	Ref
610 011HC	74HC11	Three Input AND	[illegible]	
610 008HC	74HC08	AND gate	4[illegible]	
610 032HC	74HC32	OR gate	[illegible]	
613 386	LM386	Oper Amplifier	U44	*Sound
611 006	7406	Hex Inv Buffer	[illegible]	
610 139HC	74HC139	Decoder	46	
610 166HC	74HC166	Shift Register	[illegible]	
610 004HC	74HC04	Hex inverter	48	
610 074HC	74HC74	Dual D-type Flip Flop	U[illegible]9	
610 139HC	74HC139	Decoder	[illegible]	
620 521	4521	Binary Counter	[illegible]	
618 1210G	PAL12L10	Gold PAL	U52	SOCKET
604 516	TM4732	Character Generator	U53	SOCKET
600 264S	6264-15	64K CMOS RAM	U54	*P C G
600 264S	6264-15	64K CMOS RAM	U55	*P C G
600 264S	6264-15	64K CMOS RAM	[illegible]6	
600 264S	6264-15	64K CMOS RAM	U[illegible]	
611 007	7407	Hex Buffer/Driver	[illegible]	
610 032HC	74HC32	OR gate	[illegible]	
618 870	3870	Keyboard CPU	U60	SOCKET
610 541HCT	74HCT541	Octal BUFFER	6[illegible]	
610 157HC	74HC157	Multiplexer	[illegible]	
610 157ALS	74ALS157	Multiplexer	63	
610 273HC	74HC273	Octal D Flip-Flop	64	
610 541HCT	74HCT541	Octal BUFFER	[illegible]	
610 645HCT	74HCT645	Octal Transceiver	[illegible]6	
610 032HC	74HC32	OR gate	6[illegible]	
610 175HC	74HC175	Quad D Flip Flop	[illegible]68	
610 393HC	74HC393	Counter	[illegible]69	
610 138HC	74HC138	1 of 8 Decoder	[illegible]	
610 002HC	74HC02	NOR gate	[illegible]	
610 645HCT	74HCT645	Octal Transceiver	[illegible]2	
610 574HCT	74HCT574	Octal D Flip Flop	[illegible]3	
610 157HC	74HC157	Multiplexer	[illegible]4	
610 157HC	74HC157	Multiplexer	[illegible]5	
610 004HC	74HC04	Hex inverter	[illegible]	
610 175HC	74HC175	Quad D Flip Flop	[illegible]	
610 008HC	74HC08	AND gate	[illegible]8	
610 174HC	74HC174	Hex D Flip Flop	[illegible]9	
610 086HC	74HC86	Exclusive OR Gate	80	
610 004HC	74HC04	Hex inverter	81	
610 074HC	74HC74	Dual D type Flip Flop	82	
		n/a	83	
610 645HCT	74HCT645	Octal Transceiver	[illegible]64	
610 574HCT	74HCT574	Octal D Flip Flop	85	
610 157HC	74HC157	Multiplexer	[illegible]	
610 157HC	74HC157	Multiplexer	U87	*P C G
610 138HC	74HC138	1 of 8 Decoder	U88	
610 032HC	74HC32	OR gate	U89	
610 574HCT	74HCT574	Octal D Flip-Flop	U90	
610 157HC	74HC157	Multiplexer	U91	
610 574HCT	74HCT574	Octal D Flip Flop	92	

Component Reference

Store No	Component	Description	Comp Ref
610 574HCT	74HCT574	Octal D Flip Flop	U93
610 645HCT	74HCT645	Octal Transceiver	U94
610 074HC	74HC74	Dual D type Flip Flop	U95
6[illegible] [illegible]65	6116 15ASP	2K CMOS RAM OR 6264S	U96
[illegible]10 [illegible]	6116 15ASP	2K CMOS RAM OR 6264S	U97
[illegible]65	6116 15ASP	2K CMOS RAM OR 6264S	U98
[illegible] [illegible]57	74HC157	Multiplexer	U99
[illegible] [illegible]545	6545E	CRT Controller	U100 SOCKET
610 573HC	74HC573	Octal latch	U101
6[illegible] 007	7407	Hex Buffer/Driver	U102
6[illegible] 000HC	74HC00	NAND gate	U103 RTC
6[illegible] 818	MC146818	Real Time Clock	U104 RTC
6[illegible] 140	CA3140	Oper Amplifier	U105
6[illegible] [illegible]8HC	74HC08	AND gate	U106
[illegible] 014HC	74HC14	Hex Inv Schmitt Trig	U107
[illegible] 000HC	74HC00	NAND gate	U108 RTC
6[illegible] 032HC	74HC32	OR gate	U109
		n/u	U110
610 157HC	74HC157	Multiplexer	U111
		n/u	U112
300 101	100R	Resistor 1/4 watt	R1
300 103	10k	" " "	R2
300 102	1k	" "	R3
300 102	1k	" " "	R4
300 472	4k7	" " "	R5
300 152	1k5	" " "	R6
300 102	1k	" " "	R7
300 102	1k	" " "	R8
300 470	47R	" " "	R9 *RS232
300 102	1k	" " "	R10
300 152	1k5	" " "	R11
300 183	18k	" " "	R12
300 152	1k5	" " "	R13
300 183	18k	" " "	R14
300 152	1k5	" " "	R15
300 471	470R	" " "	R16
300 471	470R	" " "	R17
300 472	4k7	" " "	R18
300 103	10k	" " "	R19
300 152	1k5	" " "	R20
300 100	10R	" " "	R21 *Sound
300 102	1k	" " "	R22
300 181	180R	" " "	R23 *Sound
	LINK	" " "	R24
300 100	10R	" " "	R25 *Sound
300 102	1k	" " "	R26
300 331	330R	" " "	R27
300 152	1k5	" " "	R28
300 103	10k	" " "	R29
300 153	15k	" " "	R30
300 152	1k5	" " "	R31

Component Reference

Store No	Component	Description	Comp Ref
300 152	1k5	Resistor 1/4 watt	R32
300 221	220R	" " "	R33
300 102	1K	" " "	R34
300 331	330R	" " "	R35
300 820	82R	" " "	[illegible]
300 331	330r	" " "	[illegible]
300 472	4k7	" " "	R38
300 102	1R	" " "	R39
300 103	10k	" " "	R40
300 823	82k	" " "	R41
300 681	680R	" " "	R42
300 331	330R	" " "	R43
300 331	330R	" " "	R44
300 472	4K7	" " "	R45
300 333	33K	" " "	R46
300 105	1M	" " "	R47
300 104	100k	" " "	R48
300 472	4k7	" " "	R49
	N/L	" " "	R50
300 182	1k8	" " "	R51 [illegible]
300 105	1M	" " "	R52 [illegible]
300 154	150K	" " "	R53 *RTC
300 475	4M7	" " "	R54 [illegible]
300 102	1K	" " "	R55
300 470	47R	" "	R56 [illegible]
	N/L	" " "	R57
300 472	4k7	" " "	R58
300 102	1k	" " "	R59
310 472	9x472	9*4k7 resistor Sip	RP1
310 102	9x102	9*1k " "	RP2
310 472	9x472	9*4k7 " "	RP3
320 010	10k pot	10k trimpot	RV1
320 050	50k pot	50k trimpot	RV2
320 002	2k pot	2k trimpot	RV3 *Sound
320 006	5k Vol	5k Volume Cont XM296RE	RV4
420 547	4.7uf	Tag Cap 16v	C1
410 301	.01uf	.01uf Ceramic Cap	C2 to C9
412 401	.1uf	.1uf Monolithic Cap	C10 to C13
420 601	1uf	Tag Cap 16v	C14
410 301	.01uf	.01uf Ceramic Cap	C15
410 210	.001uf	.001uf ceramic cap	C16
410 301	.01uf	.01uf Ceramic cap	C17,C18
412 401	.1uf	.1uf Monolithic Cap	C19 to C21
420 601	1uf	Tag Cap 16v	C22
		n/u	C23 C24
420 547	4.7uf	Tag Cap 16v	C25
410 301	.01uf	.01uf Ceramic Cap	C26
420 610s	10uf	Tag Cap 16v	C27 *RS232
410 122	220pf	Ceramic Cap	C28
412 401	.1uf	.1uf Monolithic Cap	C29
410 301	.01uf	.01uf Ceramic Cap	C30 to C35

Component Reference

Store No	Component	Description	Comp. Ref
410 301	.01uf	.01uf Ceramic Cap	C38 to C39
410 301	.01uf	.01uf Ceramic Cap	C40 to C41
420 610s	10uf	Tag Cap 16v	C42 *RS232
410 301	.01uf	.01uf Ceramic Cap	C43 to C50
420 601	1uf	Tag Cap 16v	C51
410 301	.01uf	.01uf Ceramic Cap	C52 to C53
410 301	.01uf	.01uf Ceramic Cap	C54 *Sound
410 301	.01uf	.01uf Ceramic Cap	C55 to C56
412 401	1uf	1uf Monolithic Cap	C57 *Sound
422 101	100uf	16V Electro	C58 *Sound
410 301	.01uf	.01uf Ceramic Cap	C59 to C64
420 601	1uf	Tag Cap 16v	C65
410 301	.01uf	.01uf Ceramic Cap	C66
		" "	C67 to C69
410 301	.01uf	.01uf Ceramic Cap	C70 to C78
420 601	1uf	Tag Cap 16v	C79
410 301	.01uf	.01uf Ceramic Cap	C80
	100uf	Electro	C81
410 301	.01uf	.01uf Ceramic Cap	C82
	100uf	Electro	C83
410 301	.01uf	.01uf Ceramic Cap	C84
420 547	4.7uf	Tag Cap 16v	C85
410 301	.01uf	.01uf Ceramic Cap	C86 to C96
420 601	1uf	Tag Cap 16v	C97
410 301	.01uf	.01uf Ceramic Cap	C98
420 547	4.7uf	Tag Cap. 16v	C99
410 301	.01uf	.01uf Ceramic Cap	C100 to C109
420 601	1uf	Tag Cap. 16v	C110
410 301	.01uf	.01uf Ceramic Cap	C111 to C116
420 601	1uf	Tag Cap. 16v	C117
410 301	.01uf	.01uf Ceramic Cap	C118
		" "	C119
410 301	.01uf	.01uf Ceramic Cap	C120 to C121
410 122	220pf	Ceramic Cap	C122
420 468	68uf	Tag Cap 16v	C123
410 301	.01uf	.01uf Ceramic Cap	C124
410 033	33pf	Ceramic Capacitor	C125 *RTC
410 247	4700pf	Ceramic Capacitor	C126 *RTC
420 610	10uf	Tag Cap. 16v	C127 *RTC
420 547	4.7uf	Tag Cap 16v	C128
410 047	47pf	Ceramic Capacitor	C129
412 401	1uf	1uf Monolithic Cap	C130
420 547	4.7uf	Tag Cap. 16v	C131
410 301	.01uf	.01uf Ceramic Cap	C132 to C133
420 522	2.2uf	Tag Cap 16v	C134
410 301	.01uf	.01uf Ceramic Cap	C135-C136
410 033	33pf	Ceramic Capacitor	C137 *RTC
410 301	.01uf	.01uf Ceramic Cap	C138 to C140
420 601	1uf	Tag Cap 16v	C141
410 301	.01uf	.01uf Ceramic Cap	C142 to C144
420 601	1uf	Tag Cap. 16v	C145
410 301	.01uf	.01uf Ceramic Cap	C146

Component Reference

Store No	Component	Description	Comp Ref
420 547	4.7uf	Tag Cap. 16v	C147 C148
410 018	6pF	Ceramic capacitor	C144-C152
410 022	22pF	Ceramic Capacitor	C[illegible]
425 565	trim cap	5.5 to 65pf Trim Cap	[illegible]
425 565	trim cap	5.5 to 65pf Trim Cap	C[illegible] [illegible]
580 [illegible]4	1N4148	small signal diode	D1 to D6
580 914	1N4148	small signal diode	D7 *RS232
580 914	1N4148	diode EXT KB ONLY	D8
580.740	1N4740	10 volt zener diode	D9 *RS232
580 914	1N4148	small signal diode	D10 to D23
590 548	BC548	Transistor	TR1
590 548	BC548	Transistor	TR2
590 548	BC548	Transistor	TR3
[illegible] 140	xtal	4.0Mhz	XL1
711 135	xtal	1[illegible].5Mhz	XL2
711 327	xtal	32.768KHz	XL3 *RTC
510 211	DB15	R/A female nut inserts	X1
510 252	DB25	R/A female nut inserts	X2
500 408	40 way	vertical mount	X3 *EXT
500 342	34 way	vertical mount	X4
710 026	4 pin	6410-4A	X5
710 025	2 pin	pins for speaker	X6
710 026	4 pin	6410-4A	X7
710 025	6 pin	pins for TEST points	X8
710 016	4 pin	socket 3502[illegible]	X9
510 097	DB9	R/A female nut inserts	X10
7[illegible]0 005	Din Socket	R/A PCB mounting	X11
500 502	50 way	vertical mount	X12 *EXT
710 290	15 way	flat cable socket	X13
710 291	8 way	flat cable socket	X14
530 014	14 pin	IC socket	X15 *EXT
	10 way	vertical mount	X16 *EXT
530 040	40 pin	IC socket .6 pitch	U1
530 040	40 pin	IC socket .6 pitch	U16
530 040	40 pin	IC socket .6 pitch	U34
530 040	40 pin	IC socket .6 pitch	U60
530 040	40 pin	IC socket .6 pitch	U[illegible]
530 025	24 pin	IC socket .3 pitch	U39
530 025	24 pin	IC socket .3 pitch	U52
530 028	28 pin	IC socket .6 pitch	U38
530 028	28 pin	IC socket .6 pitch	U58
770 023	switch	push button 8023L	SW1
750 500	battery	nicad battery	B1 *RTC

6 2 Off board components single drive

Store No.	Component	Description	Comp Ref
[illegible] 034#		34 way header socket	2
[illegible] 528		Case for 256 TC	1
68[illegible] 705		Back panel for 256 TC	1
[illegible] 706		Drive support bracket	1
6[illegible] 707		Drive gap panel 256 TC	1
[illegible] 0272		drive 3 5" F3541 single 5 volts	1
[illegible] 009!		3 way NR connector 0.NR E4K	1
050*		crimp pins M2759-TL	4
508		Keyboard Module 92 KEY	1
[illegible] 020!		Speaker Wire 5 0 2m	3
[illegible] 2[illegible]4*		Red cable 7/0 054TC	4
[illegible]40 215*		Green cable 7/0 054TC	2
740 340#		34 way ribbon cable	150
[illegible]41 2[illegible]3		moulded mb lead assembly	1
745 205		S. 5012 black rubber bumpons	4
750 020		speaker 2" 50 ohms	1
[illegible]45 145		Decal 256 TC	1
755 146		Decal Back Panel 256 TC	1
78[illegible] 009		Screw lock without nut	6
781 155		4#x3/8PAN PHIL s/t nickle plat	8

* Sub-ass 970 786 # Sub-ass 970 765 Sub-ass 970 788

6 3. Power supply - 970 760

Store No	Component	Description	Comp Ref
300 103	10k	Resistor 1/4 watt	R3
300 153	15k	" " "	R2
300 472	4k7	" " "	R1
410 222	2200pf	ceramic or green cap	C6
410 333	33nf	" " " "	C5
420 522	2 2uf	tag 16vw	C4
420 610	10uf	" "	C7
422 703	100uf	16vw Electro pig tail	C2
422 703	100uf	16vw Electro pig-tail	C1
422 8250	2500u6	RI2500/16 Soanar Cap	C3
580 810	diode	RUR810 Fast acting	D1
590 560	SCR	TIC106A	
602 296	L296	SM Regulator IC	IC1
650 511	MB85101 $	Power supply PCB	
700 631	Coil	250 uH inductor Horz	L1
41.210	cable	power supply cable	X1
750 222	elephanide	120mm*40mm	1
750 233	fuse holder	PCB fuse holders	2
750 406	4amp	4 amp fast M205 fuse	F1

6.4 RS232 option · 970 757

Store No	Component	Description	Comp Ref
300 470	47R	Resistor 1/4 watt	R9
410 301	.01uf	Ceramic Cap	C40 41
420 610	10uf	Tag Cap 16vw	C42
580 740	1N4740	10 volt zener diode	D9
580 914	1N4148	small signal diode	D7
600 660	TSC7660	DC to DC converter	U25
618 701	751701	RS232 txd/rxd	U13

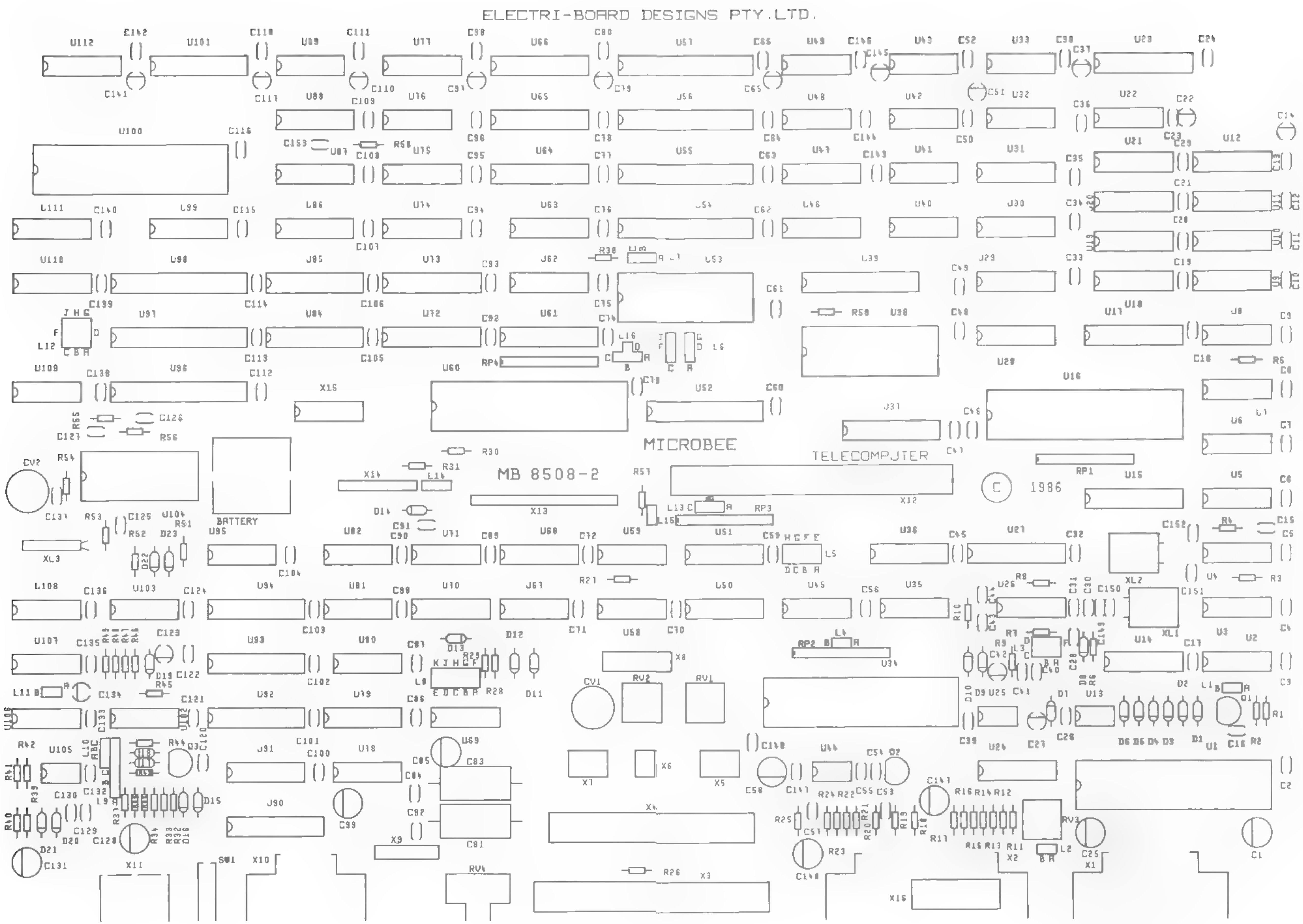
6.5. Sound circuit – 970 758

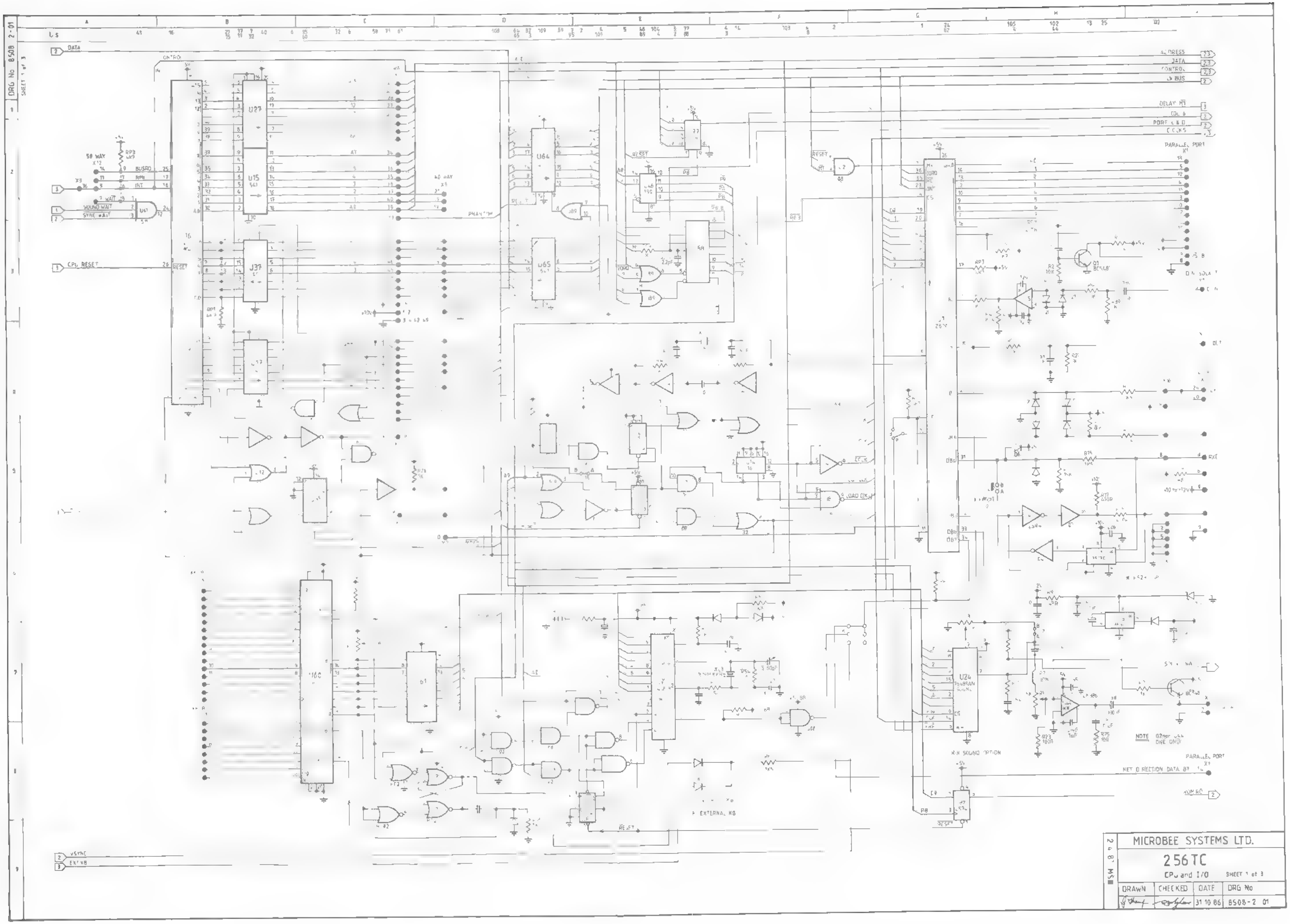
Store No	Component	Description	Comp Ref
300 100	10R	Resistor 1/4 watt	R21
300 181	180R	" " "	R23
300 100	10R	Resistor 1/4 watt	R25
320 002	2k pot	2k trimpot	[illegible]
410 301	.01uf	ceramic cap	[illegible]
412 401	1uf	mono cap	C5
422 701	100uf	16vw electro	C58
611 489	76489AN	Sound generator	U24
613 386	LM386	Oper. Amplifier	U44

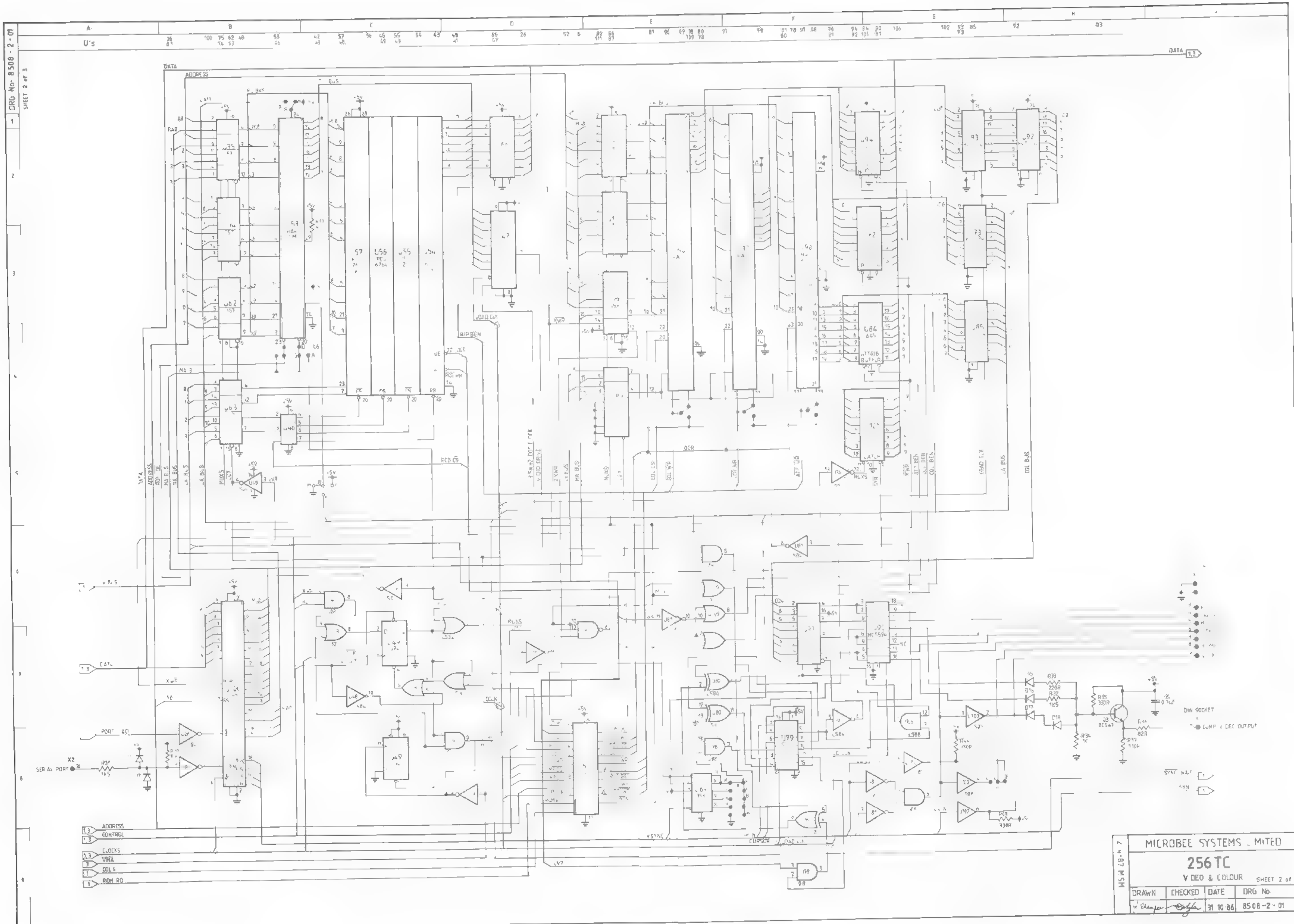
6.6 Other options 970 756

Store No.	Component	Description	Comp Ref
500 408	40 way	vertical mount	X3
500 502	50 way	vertical mount	X 2
530 014	14 pin	IC socket .3 pitch	X 5
600 2645	6264s	HM6264asp-15 CMOS RAM	U54,55 96,97,98
610 157hc	74HC157	Multiplexer	U87
740 160	14way	Ribbon Cable	300mm
500 014	14 pin	dip header	1 ext RS
510 096	DB9	male cable mount	1 ext KP

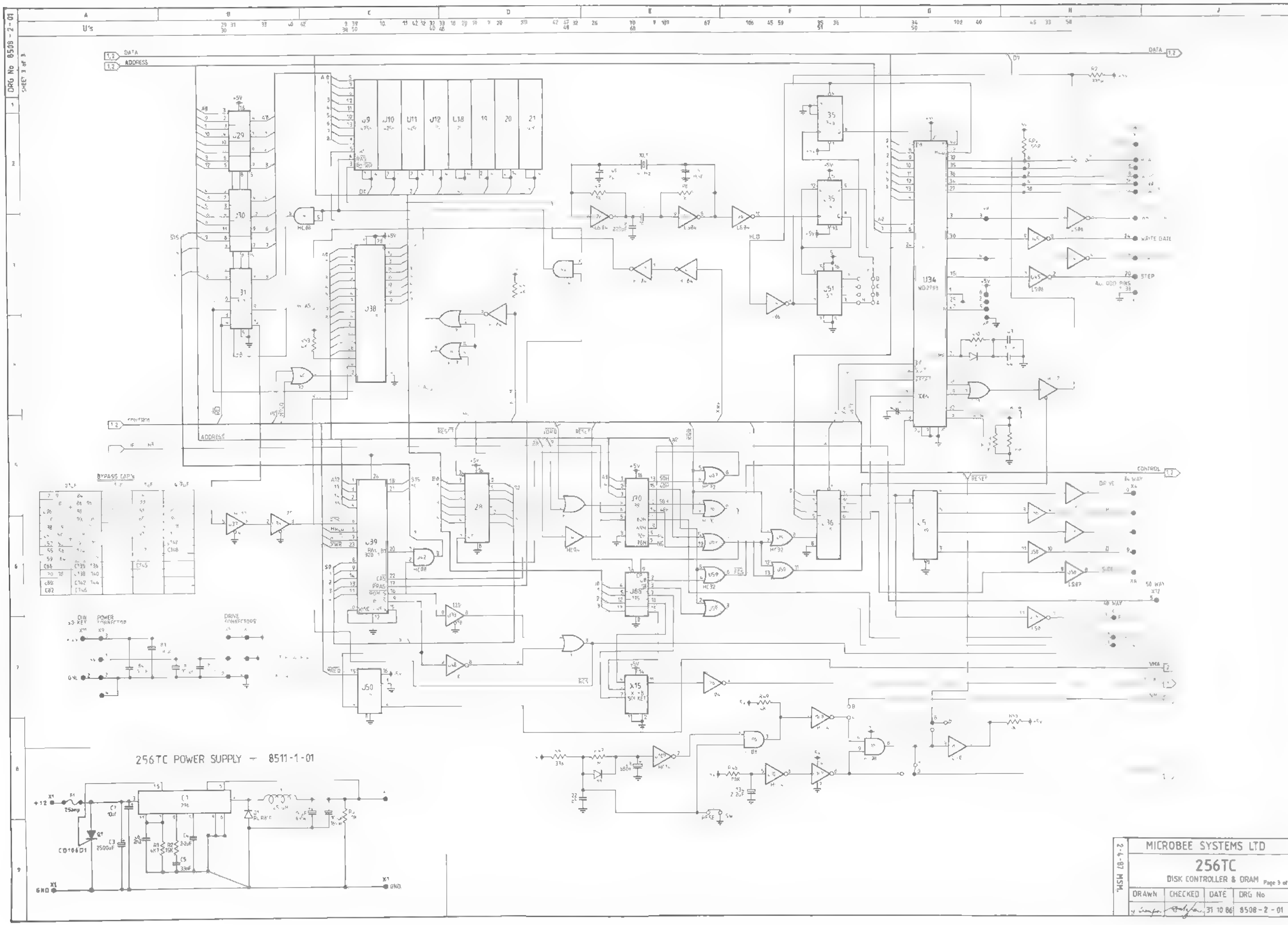
Circuit Diagrams


ELECTRI-BOARD DESIGNS PTY.LTD.
MICROBEE
TELECOMPUTER
MB 8508-2
© 1986
BATTERY

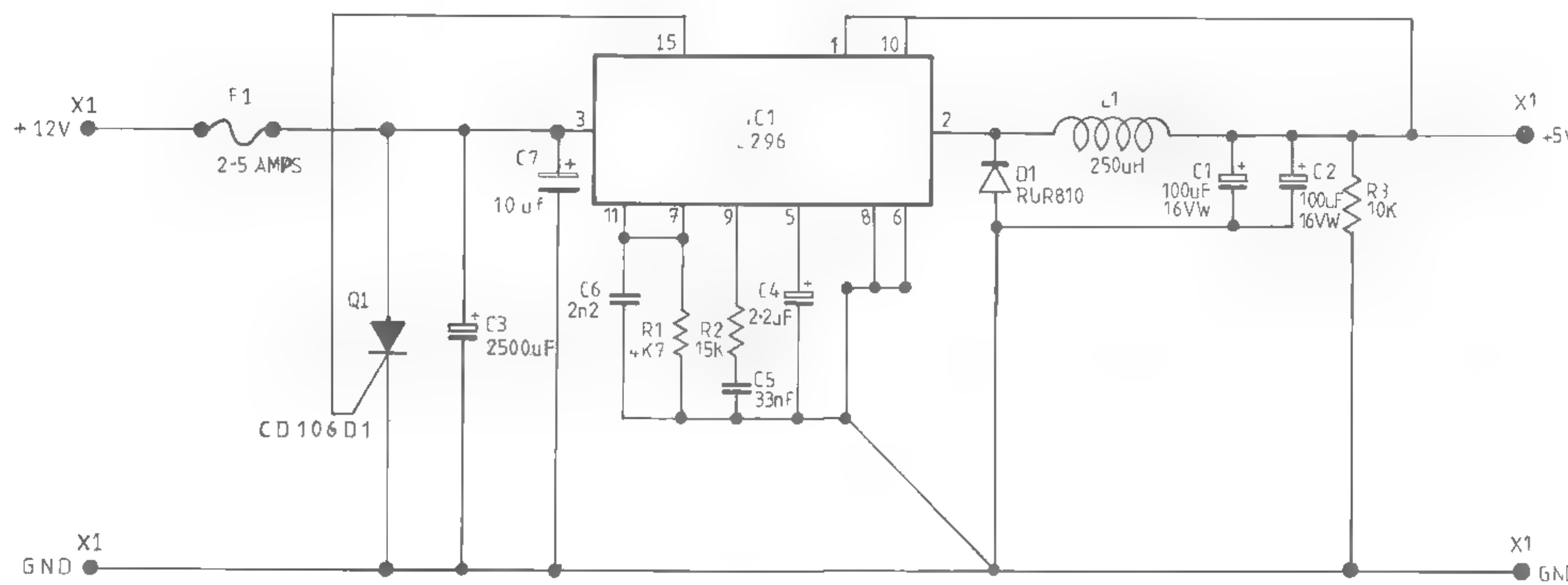

DRG No B508 2-01
SHEET 1 of 3
DATA
ADDRESS
DATA
CONTROL
IO BUS
DELAY
PORT
CLOCKS
PARALLEL PORT
U27
U15
U37
U64
U65
U24
50 WAY
40 WAY
BUSRQ
NMI
INT
SOUND WAIT
SYNC
CPU RESET
RESET
VSYNC
EXT KB
EXTERNAL KB
NOTE
PARALLEL PORT
NET DIRECTION DATA
MICROBEE SYSTEMS LTD.
256TC
CPU and I/O
SHEET 1 of 3
DRAWN
CHECKED
DATE
DRG No
31 10 86
B508-2 01



DRG No. 8508-2-01
SHEET 2 of 3
U's
DATA
ADDRESS
SERIAL PORT
PORT 40
ADDRESS
CONTROL
CLOCKS
VMA
COL 6
ROM RD
DIN SOCKET
COMP & DEC OUTPUT
MICROBEE SYSTEMS LIMITED
256TC
VIDEO & COLOUR
SHEET 2 of 3
DRAWN
CHECKED
DATE
DRG No.
31 10 86
8508-2-01


DRG No 8508-2-01
SHEET 3 of 3
DATA
ADDRESS
CONTROL
RESET
BYPASS CAP'S
DIN 5-KEY POWER CONNECTOR
DRIVE CONNECTORS
WRITE DATE
STEP
ALL ODD PINS
34 WAY
50 WAY
VMA
WD2793
256TC POWER SUPPLY - 8511-1-01
+12
GND
CD106D1
2500uF
MICROBEE SYSTEMS LTD
256TC
DISK CONTROLLER & DRAM
Page 3 of 3
DRAWN
CHECKED
DATE
DRG No
31 10 86
8508-2-01
2-4-87 MSM

DRG No: 8511-1-01



2-4-87 MSM.

MICROBEE SYSTEMS LTD.

256TC POWER SUPPLY

DRAWN	CHECKED	DATE	DRG No:
J Thompson		22·7·86	8511-1-01.

Index

256TC

Prepared and published by
Honeysoft Publishing Company
1 Pattison Avenue, Waitara, 2077
© 1987

Cat. No. 217-301
ISBN 0 949309 72 9

Microbee Systems Limited
Koala Crescent,
West Gosford
N.S.W. 2250