Project 685

A single-board computer using the 2650 on an S100 card

The S100 buss has become one of the most successful buss standards for both hobbyist and professional applications. Most past ETI computer projects have supported this buss. This project continues the line of succession and uses the popular 2650 microprocessor in a single-board computer design with many features not found elsewhere. It is compatible with our previous S100 projects (e.g: the 640 VDU and 681 PCG) and follow-up projects and articles are to come.

Ron Koenig

THE INTRODUCTION of the microprocessor to the electronics scene has brought with it many great possibilities and many new challenges for both the hobbyist and professional system builder. While the microprocessor is extremely flexible and is capable of excelling in most applications over 'discrete' circuit assemblies, it is unable to operate by itself. The microprocessor is only a system component and must be supported by a variety of additional components to be capable of performing any given task.

The basic microcomputer system is composed of three main units, or modules. These are: the Central Processing Unit (CPU), the Memory Unit and the Input/Output (I/O) Port Units. The CPU with its associated control circuitry performs all the processing and system control operations. The Memory Unit usually consists of several blocks of memory, each with its own address decoding circuitry. The memory blocks can consist of either non-volatile Read Only Memory (ROM) in which is stored permanent programs (or data), or Random Access Memory (RAM) in which is stored variable data that is subject to change during program execution. The Input/Output Units and the associated control circuitry provide the primary means by which the 'external world' can communicate with the CPU. All these units may be located simultaneously on one board or on separate boards, and are normally interconnected by a standardised system buss. (A 'buss' is a system or group of interconnections common to an assembly of different devices).

In many applications the microcomputer system is custom designed for the application required, but this is generally very expensive and quite inflexible to the needs of future expansion. Instead of designing individual cards which are dedicated to one application, it is often more cost effective to design small general purpose cards which can be used as building blocks for larger systems. This individual tailoring allows the finished system to suit any given application and, by its very nature of construction, is more flexible to the needs of future expansion.

Fortunately, the microprocessor lends itself naturally to the modular system approach. The concept of the buss-structured system can therefore be utilised to its maximum and provide the system designer with a means by which he can buy and develop small general purpose cards and interconnect them via the microprocessor buss.

The microprocessor buss forms the backbone to the microcomputer system. ▶

GENERAL SPECIFICATIONS — ETI 685

- Accommodates 2650A (1 MHz) or high speed 2650A-1 (2 MHz) CPUs.
- On-board 4 MHz crystal oscillator supplying the CPU and buss clocks.
- 4K of on-board RAM memory switchable to any 4K address boundary.
- 4K of on-board EPROM memory configured as:
  - -1K, 2K or 4K of EPROM memory commencing at '0000' hex.
  - -Supports multirail and single rail 2708 and 2716 EPROMs.
  - -Selected on-board EPROM has priority over RAM Memory.
  - -ROM memory can be enabled and disabled by software.
- The PHANTOM signal is generated when on-board RAM or EPROM is selected to disable any 'secondary' memory on the buss at the same address.
- One SERIAL PORT supported as EIA RS-232C and current loop.
- One latched 8-bit PARALLEL-IN 'keyboard' PORT.
- One Programmable Peripheral Interface (PPI) providing THREE PROGRAMMABLE PORTS. This PPI can provide combinations of static or strobed I/0, strobed bi-directional or serial I/0 and 16-bit timer operations.
- One Programmable Interrupt Controller (PIC) providing eight levels of programmable vectored interrupts.
- Non-vectored interrupts using pINT and sINTA.
- Full S100 processor and status signal generation:-
  - -pWR, pDBIN, sMEMR, sMWRT, sWO, sINP and sOUT for memory and I/0 data interchange.
  - -pSTVAL, pSYNC, Ø (1 MHz) and CLOCK (2 MHz) for buss timing.
  - -pOC for system initialisation.
- Fully buffered status, address and data lines.
- Direct Memory Address (DMA) capability using pHOLD and pHLDA.
- CPU can address up to 512K of memory using a full 16-bit S100 address buss and the on-board bank select logic.

It provides the communication 'highway' between the CPU and the systems memory and input/output modules. A great many microcomputer standard busses exist today. Some of these have thrived because of the *de facto* acceptance by large user groups, some by their ability to support a wide variety of regular devices, and others by their technical excellence. I have chosen to interface with the S100 buss as this is currently recognised as one of the industry leaders.

The S100 buss

This originated in the USA early in 1975 in a microcomputer system manufactured by MITS. The system was called the Altair 8800 and it used a 100-pin pc board connector (50 pins a side) to provide a communications buss for an Intel 8080 CPU. The Altair Buss later became known as the 'Standard 100 Pin Buss', or S100 buss.

Recently the S100 buss has attracted the attention of the Institute of Electrical and Electronic Engineers in the USA. They have now drafted the IEEE-696 Specification for the buss, which defines the electrical and buss timing specifications for the current generation of 16-bit microprocessors. Some changes include a 16-bit bi-directional data buss and an extended 24-bit address buss. Special signals have been designated to permit the combined operation of 8-bit and 16-bit hardware. These improvements will increase the useful life of the S100 buss well into the 1990s.

The S100 buss has become one of the most commercially successful buss standards ever produced, and the multitude of S100-compatible boards has attracted the interest of both the professional and hobby computerist. Several Australian companies are currently manufacturing S100 boards and several 'kit' projects have been published in ETI. The S100 RAM Card (Project 642 from Feb. 1979) and the S100 PROM Board (Project 682 from March 1981) are compatible with this project. The S100 VDU Board (Project 640 from April 1978) and the S100 Programmable Character Generator (Project 681 from June 1980) are compatible video interface units. A suitable keyboard and cassette tape interface will be published at a later date.

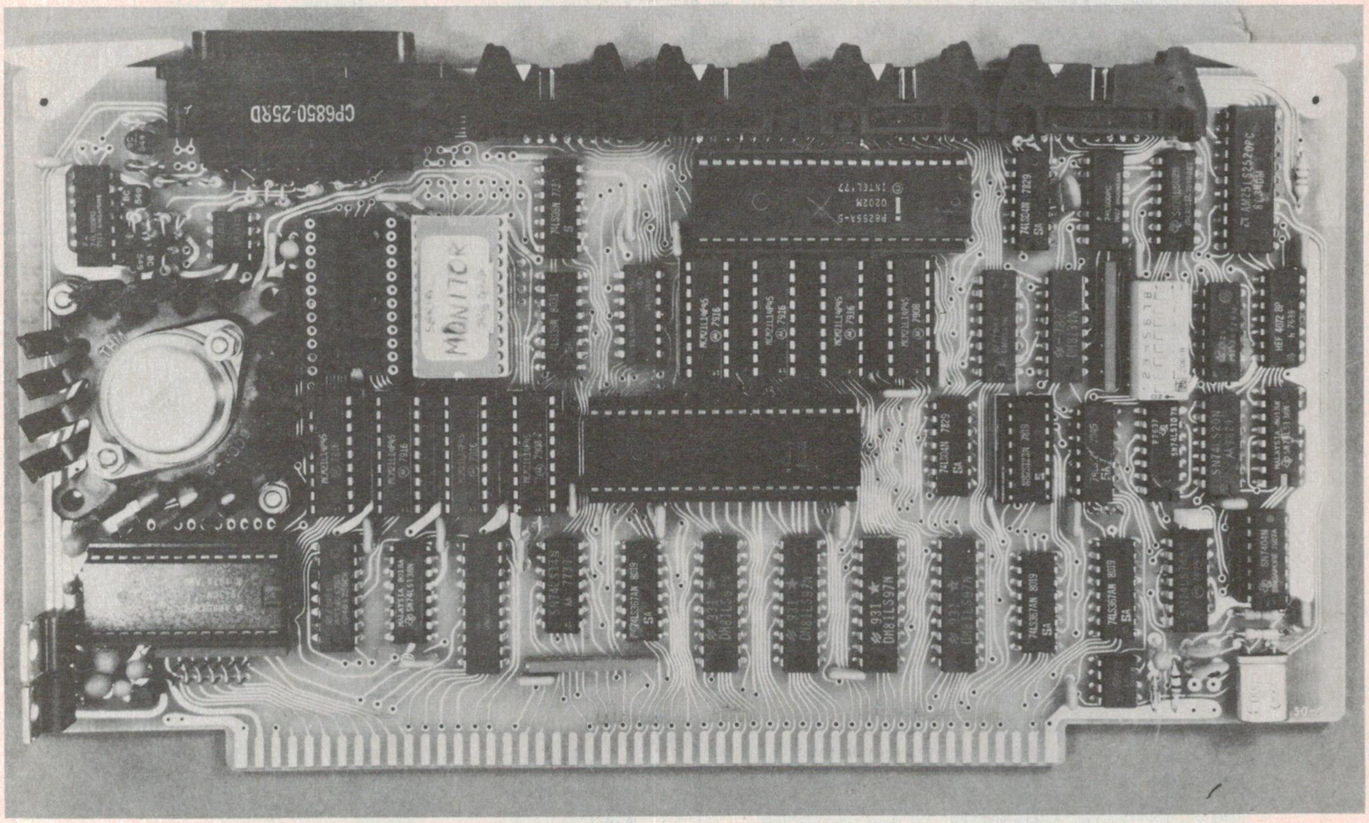
The project

The ETI-685 has been designed as a very flexible general purpose single-board computer incorporating the Signetics 2650 8-bit microprocessor. This 2650 CPU board has been designed to interface with the well-established S100 buss structure, and this enables the user to easily expand his system's input/output and memory capabilities. This easy-to-use single-board system is a very cost-effective CPU board for OEM

IEEE-696 S100 BUSS STANDARD PINOUTS

NOTE:
H = Active high signal. L = Active low signal
(O/C) = signal driven by open collector device

PIN	NAME	LEVEL	FUNCTION
1	+8 V	-	Positive system power supply
2	+16 V	-	+16 V power supply
3	XRDY	H	Buss ready; use with pin 72
4	VI0	L(O/C)	Vectored interrupt line 0
5	VI1	L(O/C)	Ditto
6	VI2	L(O/C)	Ditto
7	VI3	L(O/C)	Ditto
8	VI4	L(O/C)	Ditto
9	VI5	L(O/C)	Ditto
10	VI6	L(O/C)	Ditto
11	VI7	L(O/C)	Ditto
12	NMI	L	Non-maskable interrupt
13	PWRFAIL	L	Indicates power failure
14	DMA3	L(O/C)	Priority bit 3
15	A18	H	Extended address bit 18
16	A16	H	Extended address bit 16
17	A17	H	Extended address bit 17
18	SDSB	L(O/C)	Disable 8 status lines
19	CDSB	L(O/C)	Disable 5 control lines
20	GND		Common with pin 100
21	NDEF		Manufacturer specification
22	ADSB	L(O/C)	Disable 16 address lines
23	DODSB	L(O/C)	Disable 8 data lines
24	PH2	H	Master buss timing signal
25	PSTVAL	L	Status valid strobe
26	PHLDA	H	Use with pin 74 to buss
27	RFU	-	Reserved
28	RFU	-	Reserved
29	A5	H	Address bit 5
30	A4	H	Address bit 4
31	A3	H	Address bit 3
32	A15	H	Address bit 15
33	A12	H	Address bit 12
34	A9	H	Address bit 9
35	DO1	H	Data out bit 1 (bidirectional bit 1)
36	DO0	H	Data out bit 0 (bidirectional bit 0)
37	A10	H	Address bit 10
38	DO4	H	Data out bit 4 (bidirectional bit 4)
39	DO5	H	Data out bit 5 (bidirectional bit 5)
40	DO6	H	Data out bit 6 (bidirectional bit 6)
41	DI2	H	Data in bit 2 (bidirectional bit 10)
42	Di3	H	Data in bit 3 (bidirectional bit 11)
43	DI7	H	Data in bit 7 (bidirectional bit 15)
44	SMI	H	Status signal (op-code fetch)
45	SOUT	H	Status signal (data to output device)
46	SINP	H	Status signal (data to input device)
47	SMEMR	H	Status signal (data from memory to buss)
48	SHLTA	H	Status signal (halt executed)
49	CLOCK	-	2 MHz signal
50	GND	-	Common with pin 100
51	+8 V	-	Common with pin 1
52	-16 V	-	Negative 16 V supply
53	GND	-	Common with pin 100
54	SCLR	L(O/C)	Reset buss slaves
55	DMA0	L(O/C)	Priority bit 0
56	DMA1	L(O/C)	Priority bit 1
57	DMA2	L(O/C)	Priority bit 2
58	SXTRQ	L	Status signal (slave request)
59	A19	H	Extended address bit 19
60	SIXTN	L	Response by slaves to pin 58
61	A20	H	Extended address bit 20
62	A21	H	Extended address bit 21
63	A22	H	Extended address bit 22
64	A23	H	Extended address bit 23
65	NDEF		
66	NDEF		
67	PHANT	L(O/C)	Phantom to disable slave devices
68	MWRT	H	With PWR for write operation
69	RFU	-	Reserved
70	GND	-	Common with pin 100
71	RFU	-	Reserved
72	RDY	H(O/C)	With pin 3
73	INT	L(O/C)	Interrupt request
74	HOLD	L(O/C)	Used with pHLDA
75	RESET	L(O/C)	Master reset
76	PSYNC	H	Control buss cycle 1
77	PWR	L	Valid data on DO buss
78	PDBIN	H	Control signal data from DI buss
79	A0	H	Address bit 0
80	A1	H	Address bit 1
81	A2	H	Address bit 2
82	A6	H	Address bit 6
83	A7	H	Address bit 7
84	A8	H	Address bit 8
85	A13	H	Address bit 13
86	A14	H	Address bit 14
87	A11	H	Address bit 11
88	DO2	H	Data out bit 2 (bidirectional bit 2)
89	DO3	H	Data out bit 3 (bidirectional bit 3)
90	DO7	H	Data out bit 7 (bidirectional bit 7)
91	DI4	H	Data in bit 4 (bidirectional bit 12)
92	DI5	H	Data in bit 5 (bidirectional bit 13)
93	DI6	H	Data in bit 6 (bidirectional bit 14)
94	DI1	H	Data in bit 1 (bidirectional bit 9)
95	DI0	H	Data in bit 0 (bidirectional bit 8)
96	SINTA	H	Status after interrupt request (pin 73)
97	SWO	L	Status signal data (transfer master to slave)
98	ERROR	L(O/C)	Status signal error (in current cycle)
99	POC	L(O/C)	Power-on-clear signal
100	GND	-	System ground



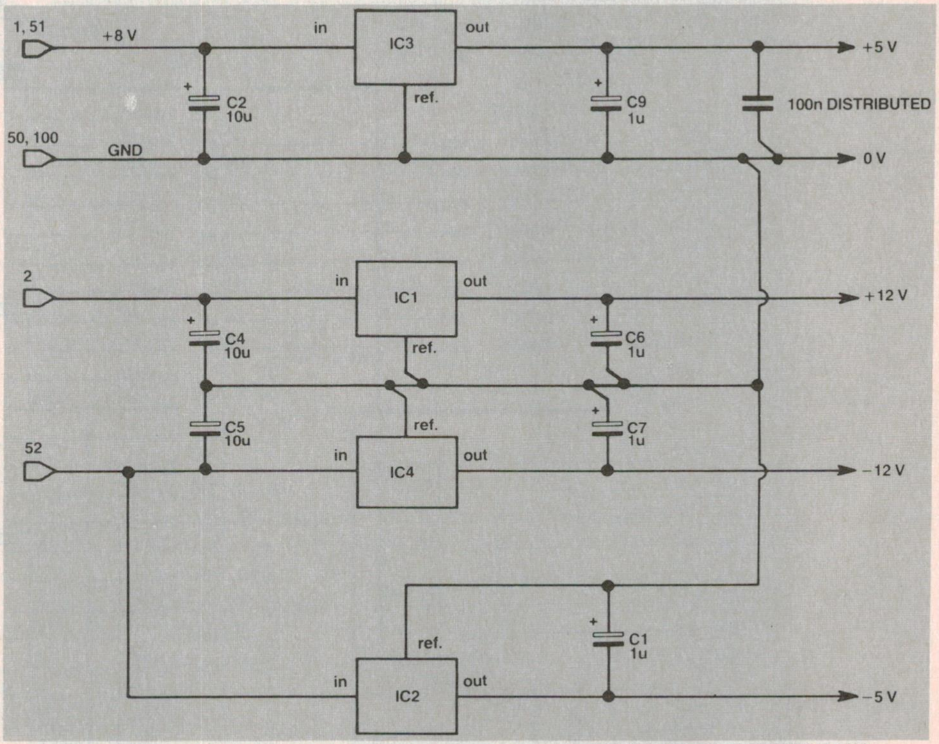
applications, 2650 enthusiasts, microprocessor students or the computer hobbyist after a powerful, expandable system well supported with projects and software.

The ETI-685 is an ideal microprocessor for the student or hobbyist who is just starting out in the world of microcomputing. Well-known author Adam Osborne describes the 2650 as "a very mini-computer-like device ... rich in memory-addressing modes and memory reference instructions". Memory addressing combinations available include absolute or relative direct addressing with optional indexing and auto-increment or decrement, and indirect addressing with optional post-indexing and auto-increment or decrement.

It may be seen from examination of the 2650 instruction set that there are many powerful instructions which are all easily understood and are typical of larger computers. This project has been designed to fully complement the capabilities of this very able microprocessor as *every mode* of memory or I/O addressing has been utilised.

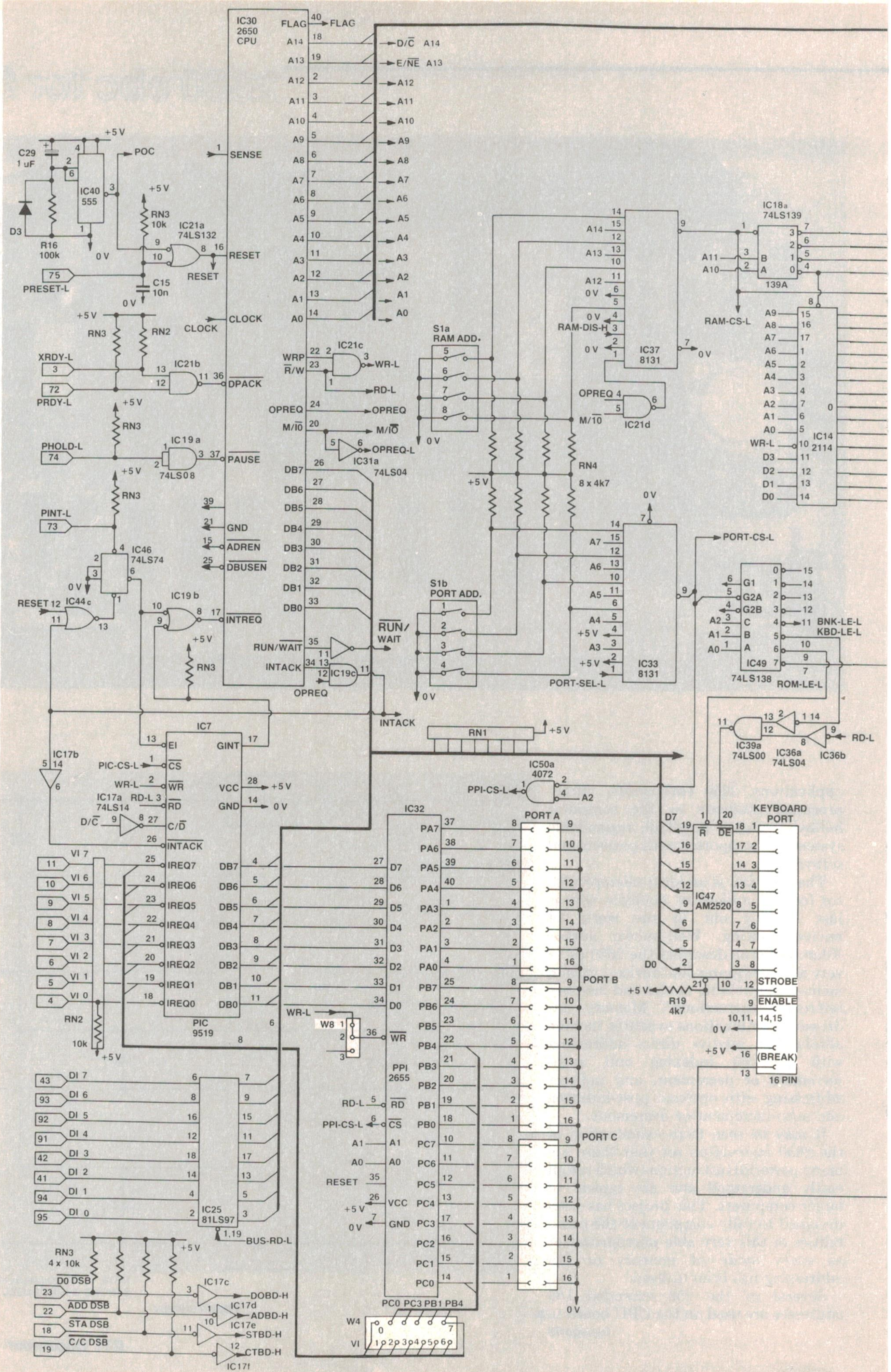
Several of the 256 extended I/O addresses are used on the CPU board to ▶

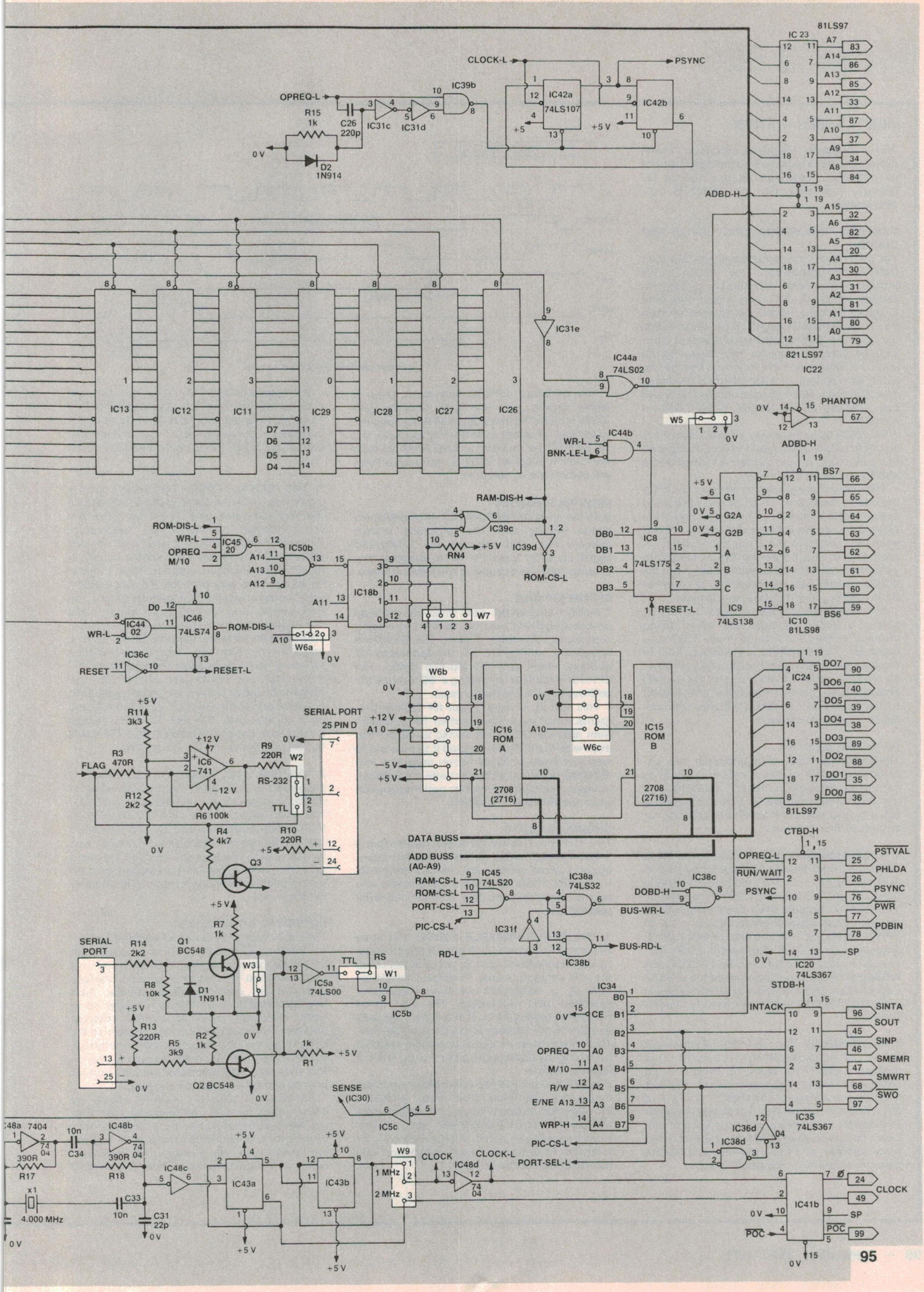
To page 99



NOTE: IC2 IS NOT REQUIRED IF 2708 EPROMS ARE NOT USED.

POWER SUPPLY


IC30
2650
CPU
FLAG
D/C̄ A14
E/N̄Ē A13
SENSE
RESET
CLOCK
D̄P̄ĀC̄K̄
P̄ĀŪS̄Ē
GND
ĀD̄R̄ĒN̄
D̄B̄ŪS̄ĒN̄
ĪN̄T̄R̄ĒQ̄
RUN/W̄ĀĪT̄
INTACK
WRP
R̄/W
WR-L
RD-L
OPREQ
M/ĪŌ
OPREQ-L
R̄ŪN̄/WAIT
C29 1 uF
IC40 555
POC
D3
R16 100k
RN3 10k
IC21a 74LS132
PRESET-L
C15 10n
XRDY-L
PRDY-L
IC21b
PHOLD-L
IC19a 74LS08
PINT-L
IC46 74LS74
IC44c
IC19b
IC31a 74LS04
IC19c
IC7
PIC 9519
EI
GINT
C̄S̄
PIC-CS-L
W̄R̄
R̄D̄
C/D̄
ĪN̄T̄ĀC̄K̄
IREQ7
IREQ0
IC17b
IC17a 74LS14
VCC +5 V
GND 0 V
RN2 10k
IC25 81LS97
BUS-RD-L
DI 7
DI 0
RN3 4 x 10k
D̄0̄ DSB
ĀD̄D̄ DSB
S̄T̄Ā DSB
C̄/C DSB
IC17c DOBD-H
IC17d ADBD-H
IC17e STBD-H
IC17f CTBD-H
IC32
PPI 2655
PA7
PA0
PB7
PB0
PC7
PC0
PORT A
PORT B
PORT C
W8
W4
PC0 PC3 PB1 PB4
PPI-CS-L
IC50a 4072
RN1
S1a RAM ADD.
S1b PORT ADD.
RN4 8 x 4k7
IC37 8131
RAM-DIS-H
IC21d
IC33 8131
PORT-SEL-L
PORT-CS-L
IC18a 74LS139
139A
RAM-CS-L
IC14 2114
IC49 74LS138
BNK-LE-L
KBD-LE-L
ROM-LE-L
IC39a 74LS00
IC36a 74LS04
IC36b
IC47 AM2520
KEYBOARD PORT
STROBE
ENABLE
(BREAK)
16 PIN
R19 4k7



OPREQ-L
R15
1k
C26
220p
0 V
D2
1N914
IC31c
IC31d
IC39b
CLOCK-L
PSYNC
IC42a
74LS107
IC42b
+5
+5 V
IC 23
81LS97
A7
83
A14
86
A13
85
A12
33
A11
87
A10
37
A9
34
A8
84
ADBD-H
A15
32
A6
82
A5
20
A4
30
A3
31
A2
81
A1
80
A0
79
821LS97
IC22
IC13
IC12
IC11
IC29
IC28
IC27
IC26
D7
D6
D5
D4
IC31e
IC44a
74LS02
PHANTOM
67
W5
0 V
ADBD-H
WR-L
BNK-LE-L
IC44b
+5 V
G1
G2A
G2B
A
B
C
IC9
74LS138
BS7
66
65
64
63
62
61
60
59
BS6
IC10
81LS98
DB0
DB1
DB2
DB3
IC8
74LS175
RESET-L
RAM-DIS-H
IC39c
RN4
IC39d
ROM-CS-L
ROM-DIS-L
WR-L
OPREQ
M/10
IC45
20
IC50b
A14
A13
A12
A11
IC18b
W7
D0
IC44
02
IC46
74LS74
ROM-DIS-L
A10
W6a
RESET
IC36c
RESET-L
W6b
+12 V
A1 0
−5 V
IC16
ROM
A
2708
(2716)
W6c
IC15
ROM
B
2708
(2716)
IC24
DO7
90
DO6
40
DO5
39
DO4
38
DO3
89
DO2
88
DO1
35
DO0
36
81LS97
R11
3k3
SERIAL PORT
25 PIN D
FLAG
R3
470R
IC6
741
+12 V
−12 V
R9
220R
W2
RS-232
TTL
R12
2k2
R6 100k
R4
4k7
R10
220R
Q3
DATA BUSS
ADD BUSS
(A0-A9)
CTBD-H
OPREQ-L
RUN/WAIT
PSYNC
PSTVAL
25
PHLDA
26
PSYNC
76
PWR
77
PDBIN
78
SP
IC20
74LS367
RAM-CS-L
ROM-CS-L
PORT-CS-L
PIC-CS-L
IC45
74LS20
IC38a
74LS32
IC31f
IC38c
DOBD-H
BUS-WR-L
IC38b
BUS-RD-L
RD-L
R7
1k
SERIAL
PORT
R14
2k2
Q1
BC548
W3
IC5a
74LS00
TTL
RS
W1
R8
10k
D1
1N914
IC5b
R13
220R
R2
1k
R5
3k9
Q2 BC548
R1
SENSE
(IC30)
IC5c
IC34
CE
A0
A1
A2
A3
A4
B0
B1
B2
B3
B4
B5
B6
B7
OPREQ
M/10
R/W
E/NE A13
WRP-H
PIC-CS-L
PORT-SEL-L
STDB-H
INTACK
SINTA
96
SOUT
45
SINP
46
SMEMR
47
SMWRT
68
SWO
97
IC35
74LS367
IC36d
04
IC38d
IC48a 7404
10n
C34
IC48b
390R
R17
R18
IC48c
x1
4.000 MHz
C33
10n
C31
22p
IC43a
IC43b
W9
1 MHz
2 MHz
CLOCK
CLOCK-L
IC48d
74
04
IC41b
Ø
24
49
CLOCK
SP
POC
99

HOW IT WORKS — ETI 685

This is a detailed functional description of the project and not a 'pulse-by-pulse' description of its operation. Reference to data books for relevant ICs (especially the 2650) is recommended.

CENTRAL PROCESSOR UNIT (CPU) AND CLOCK

The CPU used is the Signetics single-chip 8-bit NMOS microprocessor, the 2650A. This processor has been designed to closely resemble conventional binary computers and executes variable length instructions of from one to three bytes in length. This CPU contains a total of eight general purpose registers, each eight bits long. Any register may be used as the source or destination for arithmetic operations, as index registers, and for I/O data transfers. The 2650 has a 15-bit parallel address buss and can address up to 32 768 bytes (32K) of memory.

The 2650 includes a very versatile set of I/O instructions which provide it with 256 extended I/O addresses, two non-extended ports and a special single-bit I/O facility.

The project comprises an internal 8-bit bidirectional data buss, 15-bit address buss and several control signals which interconnect the 2650 CPU to the on-board RAM and ROM memory, ports and the S100 buss buffers. A programmed 32 x 8 fuse-link PROM is used to generate the S100 buss control signals from the 2650's control signals, required for external S100 memory and I/O data interchange.

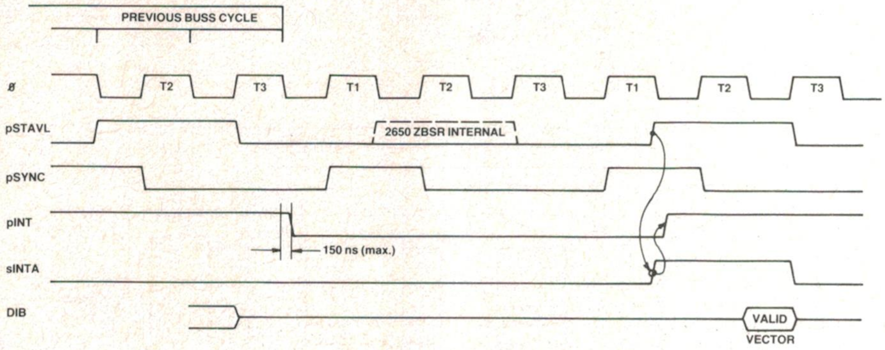
A 4 MHz quartz crystal oscillator formed by IC48a, b and c provides the basic timing element for the CPU and the entire S100 buss computer. This frequency is divided by IC43 to produce the 2 MHz and 1 MHz clocks required for the CPU and the S100 buss CLOCK (pin 49) and ϕ (pin 24) signals. The wire link, W9, is used to change the CPU clock to 2 MHz for the high-speed 2650A-1 processor.

ON-BOARD MEMORY

The Signetics 2650 microprocessor can address directly 32K of memory via its 15 address lines and, following a 'reset', reads its first instruction from address '0000'H. It is therefore customary to locate the system EPROM monitor to start at this address.

1. EPROM MEMORY: Provision has been made on the board for the use of either single (+5 V) or multirail (+5, +12, −5 Volt) type EPROMs in either 1K (2708/2758) or 2K (2716) increments. Two EPROM sockets have been provided, so the board can accommodate 1K, 2K or 4K of EPROM memory, with the first EPROM (IC16) addressed to '0000'H. The wire link set, W7, selects the location of the second EPROM, and W6 adjusts the pin configuration for the EPROM family in use. These links are preset for a single 2708-type EPROM, which will carry the monitor program.

2. RAM MEMORY: The ETI-685 has provision for 4K of on-board RAM using eight 2114 memory chips. This block of RAM can be addressed to any 4K boundary within the 32K 2650 memory map by switches 5, 6 and 7 of SW1, or it can be disabled completely by switch 8. These four switches work in conjunction with the octal comparator (IC37) to select the RAM block address, and the 1-of-4 decoder (IC18a) generates the RAM chip-select signals.



INTERRUPT (NON-VECTORED) TIMING DIAGRAM

The on-board EPROM has been given a higher priority than the RAM, and the gate IC39b inhibits the 'reading' of all RAM switched to occupy the same address. For example, if a 1K EPROM (i.e: monitor) and the RAM are both switched to start at '0000'H, the usable RAM will commence at '0400' H.

MEMORY PHANTOM

Both the on-board EPROM and RAM exercise a higher priority over the remaining system memory map and generate the PHANTOM signal on S100 buss pin 67 to deselect any external memory occupying the same address.

EPROM DISABLE

A facility has been provided where the selected on-board EPROM may be disabled, providing continuous RAM from '0000'H. The EPROM is disabled by writing '01'H to the applicable I/O address, which sets the flip-flop IC46b. The EPROM is returned by writing a '00'H to the I/O address or by a processor reset.

This feature is very useful for testing programs which have been written for operation from address '0000'H. It should be noted that even with the EPROM enabled it is possible to load (i.e. Write or Block Move) programs into the RAM, which is co-resident with the EPROM; however, you can only read (or run) the program when the EPROM is disabled.

ON-BOARD PORTS

The ETI-685 has been provided with five ports to give the user a wide variety of interface input-output devices without the need for additional I/O boards. Three ports are programmable, one port is serial and the fifth is a latched 8-bit parallel port.

1. THE PROGRAMMABLE PORTS: The three programmable ports (A, B and C) are provided by way of a single 40-pin LSI device (IC32) called the PPI (Programmable Peripheral Interface). This IC is addressed as four consecutive extended port addresses. The first three addresses access the A, B and C port registers and the fourth address is the Control Register. The PPI has three modes of operation, selected by writing the appropriate control word into the Control Register.

In MODE 0 the PPI provides simple input and output for the three 8-bit ports. MODE 1 provides for strobed input and output data transfer from ports A and B with 'handshaking' signals supplied by port C. In MODE 2 the A port is structured as an 8-bit bi-directional port with handshaking supplied by port C.

The Signetics 2655 PPI provides two additional features over the standard 8255 PPI. The B port may be configured for 8-bit serial-to-parallel or parallel-to-serial communications and, in the 2655, the B port contains a 16-bit timer.

2. THE SERIAL PORT: The serial port is supported by the 2650's single bit I/O facility via the Flag and Sense pins. These two CPU pins are connected directly to the processor's Program Status Word register and can be processed by software to provide a variety of serial communication formats. The monitor (Multibug) uses this port for 300 Baud ASCII serial communications, for keyboard input and CRT output, and for 300 Baud binary serial communications to the cassette tape interface. With suitable software, the port can be used to perform any form of data communication including *music and Morse code generation!*

The Flag and Sense pins of the CPU are buffered on the board and are made available at the serial port at EIA RS-232C voltage levels and as a current loop. The EIA RS-232C voltage levels can optionally be converted to TTL levels by rewiring the wire links W1 and W2.

3. THE PARALLEL PORT: The parallel port is supported by a single TTL octal latch which is similar in operation to the standard 74LS373. The latch used, however, is the AM25LS2520 (IC47) which features an additional asychronous 'clear input' signal. This port is read and reset by addressing the appropriate 2650 extended port. The monitor uses this port as the keyboard input when the 'memory-mapped VDU' monitor is being used.

INTERRUPT CONTROLLER

The project uses the powerful AM9519 Universal Interrupt Controller (IC7) to process eight maskable interrupt inputs. This controller has been designed as a general purpose interface and can be used by most popular 8-bit microprocessors. The AM9519 manages up to eight maskable interrupt inputs, resolves priorities and issues an 'interrupt request' to the CPU. When the CPU responds with an 'interrupt acknowledge' the controller outputs a one-to-four byte response associated with the highest priority unmasked interrupt request.

For the 2650 CPU the AM9159 should be programmed for only one response byte, and the eight response bytes are pre-loaded into only eight locations within the 8 x 32 internal read/write response memory. All communication with the Interrupt Controller is by way of the 2650's non-extended C and D ports. The C port addresses the Control input for loading the Command Register and reading the Status Register, and the Data read or write transfers to or from the selected internal registers or memory locations are performed via the D port.

NON-VECTORED INTERRUPTS

The S100 non-vectored Interrupt Request (pINT) on buss pin 73 is also supported on the board. This input, when pulled low, sets the flip-flop IC46a and generates a 2650 INTREQ. When this interrupt occurs, the 2650 will complete its current instruction, set the interrupt inhibit bit in the PSW and generate an S100 sINTA ('interrupt acknowledge') signal on buss pin 96. On receipt of sINTA the interrupting device must output the 8-bit vector onto the data-in buss. The flip-flop IC46a is reset automatically when the CPU generates sINTA.

This interrupt request (pINT) has been allotted a higher priority than the eight vectored interrupts managed by the PIC.

MONITOR

A 2716 2K, single-rail EPROM containing two 1K monitor programs is available for use with this project. The EPROM has been written to support either serial communications or memory-mapped video terminals by selecting the appropriate 1K monitor. The EPROM-type link field is set up for a 2758 single-rail 1K EPROM, and the A10 address pin is wired to either 0 V or +5 V to select the required 1K monitor.

Both monitor programs have been styled on the BINBUG monitor and their commands are compatible to BINBUG and the Signetics PIPBUG monitors. The SERIAL MONITOR communicates at 300 Baud via the 2650 Flag and Sense pins and contains a few new subroutines to erase the VDU screen and print a sign-on message. The MEMORY-MAPPED MONITOR differs only from BINBUG in the keyboard-in subroutine, which now utilises the SBC 8-bit parallel port.

BANK SELECT

The ETI-685 incorporates a bank select facility to extend the 2650's maximum address range to 512K of memory. This is accomplished by first generating a sixteenth address bit to provide an address range of 64K, and then providing a one-of-eight bank (of 64K) select. A quad latch, addressed as one of the on-board decoded extended addresses, is used to store four bits of data. The least significant bit is used as the sixteenth S100 address bit (A15), and the other three bits are presented to a 74LS138 one-of-eight decoder. The eight outputs of the decoder are buffered and appear on the S100 buss on pins 59 to 66 inclusive. If the bank select feature is not required the 74LS138 and the tri-state buffer can be omitted.

S100 SIGNAL GENERATION AND TIMING

An 82S123 fuse-link PROM (IC34) is used to synthesise seven of the eleven S100 control and status signals generated on the board. This PROM has as its inputs the five 2650 control signals — OPREQ, WRP, $\overline{R}$/W, M/$\overline{IO}$ and E/$\overline{NE}$.

The 2650 'operation request' (OPREQ) output signal is the coordinating signal for all external CPU operations. As this signal validates (or qualifies) all data, address and control lines from the 2650 it is 'ANDed' in the PROM with the other four CPU control signals. The OPREQ signal is used to generate the S100 control signal pSTVAL (Processor Status Valid).

The Write Pulse (WRP) output is a timing signal from the CPU that provides a positive-going pulse in the middle of each memory or I/O write operation. It is designed to be used as a timed Write Strobe generated after the address and data lines have stabilised. In the fuse-link PROM this signal is used to generate the S100 control signal, PWR.

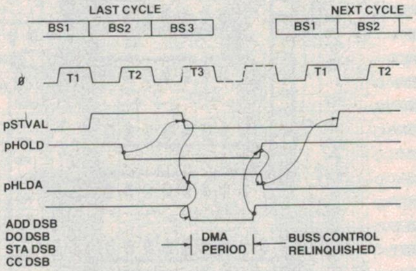
The processor Read/Write ($\overline{R}$/W) output defines whether the external operation is a read or a write, and the Memory I/O (M/$\overline{IO}$) output defines whether the operation is for memory or I/O. These signals are gated in the PROM to produce the S100 sMEMR, sINP, sMWRT and sOUT signals. As $\overline{R}$/W also indicates in which direction the data flow is, it is also used to generate pDBIN and to control the on-board data-buss transceivers.

The Extended/Non-Extended (E/$\overline{NE}$) 2650 output is the operation control signal that is used to discriminate between the two-byte extended and the one-byte non-extended I/O operation. On the ETI-685 the C and D non-extended I/O addresses are used for communicating with the on-board Interrupt Controller and they are not presented to the S100 buss. The extended signal is used in the PROM to qualify the S100 I/O status signals sINP, sOUT and sWO.

The S100 control signal pSYNC is defined as indicating the start of a new buss cycle and was initially used to strobe the status latches of external circuitry. These latches stored the 8080's status information which was present on its data lines at this time. As several modern memory boards use pSYNC pulses (e.g: to generate 'wait states') this S100 control signal has been synthesised on the ETI-685. A modulo-3 counter synchronised by OPREQ is used to generate a pulse every three CPU clock cycles. This pulse is one clock period long, and is timed to rise midway through the first buss cycle.

The S100 CLOCK signal is a 2 MHz clock, and the phi (ϕ) signal is the same frequency as the CPU clock (as selected by W9). A Power-On-Clear (POC) signal is generated onboard and can be used to reset 'slave' devices. The Status signal sINTA is the 2650 INTACK, and the pHLDA is the 'WAIT' CPU signal.

Refer to the *S100 buss timing diagram* for a graphical representation of the timing of these Status and Control signals.



DMA SEQUENCE TIMING DIAGRAM

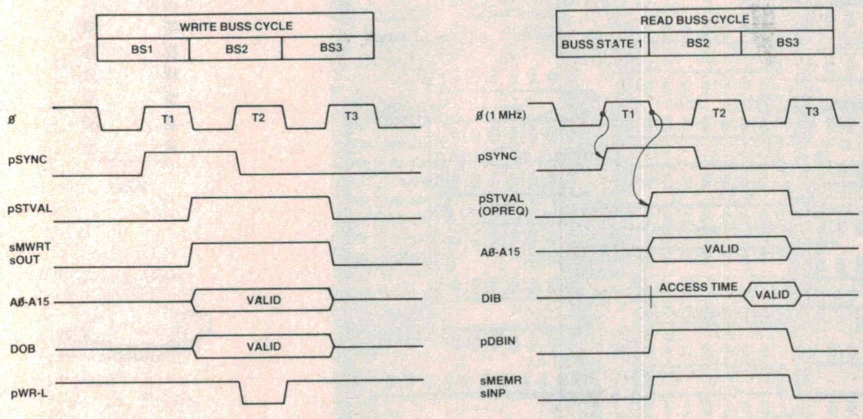
DIRECT MEMORY ACCESS (DMA)

The temporary transfer of buss control from the Buss Master to a Temporary Buss Master for that device to execute a direct memory read or write is referred to as 'Direct Memory Access'. In order to avoid conflict during this transfer of buss control, a predetermined sequence of events exists which is called the DMA Cycle. The exchange of buss control to the DMA device and the subsequent return of control to the CPU (Buss Master) is accomplished by the S100 pHOLD and pHLDA signals.

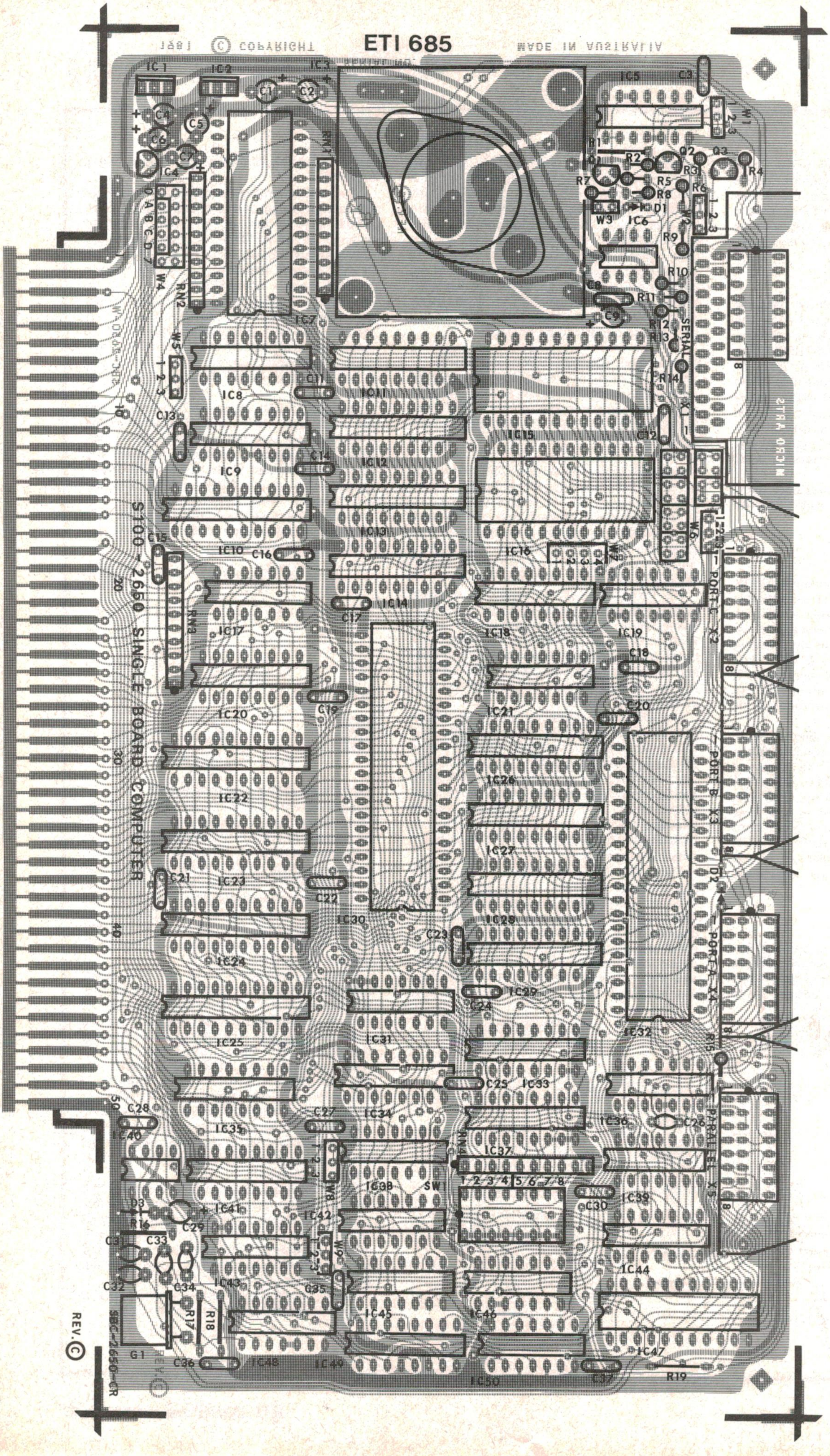
On the ETI-685 board the pHOLD S100 signal (pin 74) is connected to the 2650 Pause line. When this signal is active the CPU completes its current instruction and enters the WAIT state. It indicates when this condition exists by sending the RUN/$\overline{WAIT}$ status output 'low', and this action generates the S100 pHLDA signal (pin 26). The receipt of pHLDA by the DMA device indicates that it may assert ADSB, DODSB and SDSB, which disables (tri-states) the CPU address, data-out and status buss-buffers. The final transfer of buss control is effected with the assertion of CDSB, which disables the CPU control buss-buffer.

The Temporary Buss Master is now in full control of the buss, and will maintain this condition to the end of its DMA cycle. Return of control to the Buss Master is almost the 'mirror image' of events, with the final transfer of control accomplished with the removal of the pHOLD signal by the DMA device.

Refer to DMA *sequence timing diagram* for a graphical representation of a DMA Cycle.

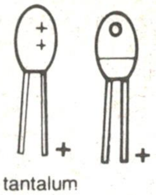


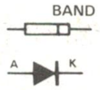
S100 BUSS TIMING DIAGRAM



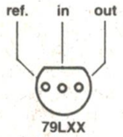
COMPONENT PINOUTS

Capacitors



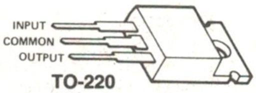


DIODES

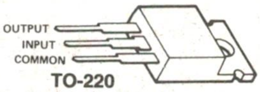


79LXX
bottom view

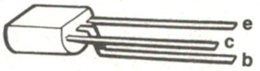
TO-92



TO-220
78xx +ve voltage reg.

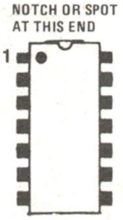


TO-220
79xx −ve voltage reg.

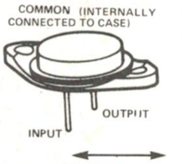


TO-92

BC548, BC108



ICs



TO-3
+ve voltage reg.

From page 93

access the on-board programmable ports, and the control and data non-extended I/O instructions are used to communicate with the programmable Interrupt Controller. This Interrupt Controller will provide the user with a full understanding of interrupt handling procedures.

Construction

The pc board designed for this project is a double-sided type with plated-through holes. We recommend you use a commercially made board for no other reason than that it goes a long way towards ensuring success with the project. If you have access to the appropriate equipment and have enough experience to feel confident in making your own double-sided board, then prints of the pc board pattern are available from us — with the usual proviso that you will only be making one for your own use and not for resale. Note that breach of copyright is now a criminal offence. The board design is copyright to the author, who has licensed Applied Technology to manufacture them. Apart from selling them retail, we understand Applied Technology will wholesale boards to other suppliers.

If you want to make your own board, then send a large (at least 250 x 300 mm) stamped, addressed envelope to:

ETI-685 PCB
ETI Magazine
15 Boundary St
Rushcutters Bay NSW 2011

We will return a same-size positive print of the front and rear pc board patterns.

With the pc board and all the components in your possession, the first step is to install all the IC sockets in their correct positions. It is recommended that you use sockets for the two EPROM positions (ICs 15 and 16), the 2650 CPU (IC30), all the RAM chips (ICs 11, 12, 13 and 14 plus ICs 26, 27, 28 and 29), IC7 (the PIC), IC32 (the 8255 PPI), IC43 (though one is not shown in our picture) and IC48. These are all located on the component side ('front'). Take care to orient them correctly. On those oriented 'vertically', pin 1 faces 'down' (toward the S100 connector). Those oriented 'horizontally' face the right hand side of the board, when viewed from the component side with the S100 connector facing down.

For the ports — marked X1, X2, X3, X4 and X5 — you have the option of installing dual-in-line sockets or the appropriate right-angle connectors (as shown in the photograph of the prototype).

A little tip — when installing IC sockets, solder one pin on each end of the socket and check that the socket is flat against the board. If necessary, reheat the solder and push the socket against the board. When all sockets are 'tacked' in flat, finish soldering all the other pins.

Install all the resistors next. Pre-bend the leads of each resistor using a pair of long-nosed pliers before inserting them into the board. Note that R2 to R15 are mounted vertically.

Install the four resistor networks (RN1, 2, 3 and 4). Note that pin 1, identified by a 'dot' on the resistor network, is located as indicated on the component overlay.

Now install all the capacitors. Take note of the polarity of the tantalum capacitors.

Follow with diodes D1, D2 and D3; D1 and D2 are mounted vertically. Take note of their polarity, also.

Next comes the crystal. Carefully preform the leads with a pair of long-nosed pliers and apply an 8 mm-square piece of double-sided tape to the back of the crystal before installing it on the board. When soldering, do not apply excessive heat.

At this stage, check with a multimeter that there is not a short circuit between any voltage rail and 0 V. Measure at the regulators IC1, 2, 3 and 4 between the input and output to ground for any short circuits. Locate and rectify any 'shorts' found before proceeding any further.

Now you can install the voltage regulators IC1, IC2, IC3 and IC4. IC3 is mounted on a large, finned heatsink which must be spaced above the board. Mount four M3 10 mm screws onto the pc board, with four nuts on the top of the board acting as spacers. Fit the heatsink onto the four screws and, after checking that the IC holes are the right way, secure the heatsink with two nuts to the outside screws. Apply heat conductive silicon paste to the underside of IC3 and mount it onto the heatsink. Secure with two nuts and solder the two pins.

Now it is prudent to check power supply operation. Apply power and verify with your multimeter that the outputs of the voltage regulators are within ± 5% (e.g: 5.2 to 4.8 V for IC3). ▶

PARTS LIST—ETI-685

Resistors	all ½W5%
R1, 2, 7, 15	1k
R3	470R
R4, 19	4k7
R5	3k9
R6, 16	100k
R8	10k
R9, 10, 13	220R
R11	3k3
R12, 14	2k2
R17, 18	390R
RN1, 2, 3, 4	4k7 x 9 SIP resistor networks
Capacitors	
C1, 6, 7, 9, 29	1u/6 V tant.
C2, 4, 5	10u/6 V tant.
C3, 8, 10-25, 27, 28, 30, 35-37	100n ceramic
C26	220p ceramic
C31, 32	22p (see text)
C33, 34	10n ceramic
Semiconductors	
D1, 2, 3	1N914A, 1N4148A
Q1, 2, 3	BC108, BC548
IC1	7812
IC2	7905
IC3	LM323K
IC4	79L12
IC5, 39	74LS00
IC6	741
IC7	AM9519PC (PIC)
IC8	74LS175
IC9, 49	74LS138
IC10	81LS98
IC11-14, 26-29	2114L-4
IC15, 16	2708/2716 (one monitor, one spare)
IC17	74LS14
IC18	74LS139
IC19	74LS08
IC20, 35, 41	74LS367
IC21	74LS132
IC22-25	81LS97
IC30	2650A (CPU)
IC31, 36	74LS04
IC32	2655:8255A (PPI)
IC33, 37	8131
IC34	82S123
IC38	74LS32
IC40	555
IC42	74LS107
IC43, 46	74LS74
IC44	74LS02
IC45	74LS20
IC47	AM25LS2520
IC48	7404
IC50	4072
Miscellaneous	
X1	4.000 MHz crystal
SW1	8-way SPST DIP switch

ETI-685 pc board (see text); pc-mount T0-3 heatsink — Thermalloy type THM6051B or 6001B-2 or similar; DIP sockets — 2 x 8-pin, 15 x 14-pin, 15 x 16-pin; 8 x 18-pin, 5 x 20-pin, 1 x 22-pin, 2 x 24-pin, 1 x 28-pin, 2 x 40-pin; nuts, bolts etc.; two pc board ejectors — e.g: Cambion No. 415 7036 01 00 20 or similar.

Optional connectors: The following connectors may be used in lieu of 16-pin DIP sockets: 1 x 25-pin 90° pc-mount 'D' connector (CP6850-25RD); 3 x 10-pin 90° headers (e.g: Hirose HIF3-10P-2.54DS or sim.); 1 x 16-pin 90° header (e.g: Hirose HIF3-16P-2.54DS or sim.).

Price estimate

We estimate the cost of purchasing all the components for this project will be in the range:

$200—$230

Note that this is an **estimate** only and **not** a recommended price. A variety of factors may affect the price of a project, such as — quality of components purchased, type of pc board (fibreglass or phenolic base), type of front panel supplied (if used), etc — whether bought as separate components or made up as a kit.

If all is well, remove power and clean the flux off the rear of the board with flux cleaner or methylated spirits.

Before proceeding with installation of the ICs onto the board it is advisable to check the data and address buss lines for shorts. Any shorts on these lines will prevent the board from operating and can make fault finding very difficult. An ohmmeter or small buzzer can be used to check for shorts. Place one probe on the first data or address line at the CPU socket (IC30) and touch the other probe across the remaining lines in turn. No continuity should be found. Move the fixed probe to the next line and repeat the process until all lines have been checked.

Install IC43 and 48 into their respective sockets. Apply power to the board and, with the aid of a logic probe or CRO, verify that the 1 MHz clock appears at pin 38 of IC30 and pin 6 of IC41.

Install the following ICs: 5, 13, 17, 18, 19, 21, 28, 30, 33, 36, 37, 40, 44, 45, 46, 49 and 50. The board now contains sufficient components to operate as a 'minimum component system' with 1K of user RAM available from '0400'H to '07FF'H. The board can be operated and tested using a serial VDU with communications via the current loop. A serial VDU monitor in 2708 EPROM must be installed at IC16. If the 2716 dual SBC monitor is to be used, the W6 wire link field must be rewired, as illustrated in the wire-link diagrams. The SBC Monitor requires the keyboard to be wired to the parallel port at socket X5.

The successful operation of the project at this point will indicate that all the internal data, address and control busses are without fault and you may proceed to install the on-board ports, extra RAM and the S100 buss buffers.

If a serial VDU is not available, this intermediate test cannot be performed, so proceed to the next step.

Install the following ICs: 6, 11, 12, 14, 20, 22, 23, 24, 25, 26, 27, 29, 31, 34, 35, 38, 39, 41, 42 and 47. The board is now complete and can be installed onto an S100 buss mother board.

A 'BINGUG' 2708 Monitor can be isnerted into IC16 to enable the use of the 640 VDU. If the dual SBC monitor or any other program in 2716 EPROM is to be used, ensure that the W6 link field has been rewired accordingly.

The ETI-685 can now be tested on the S100 buss with additional RAM memory. To verify that the system is fully operational, load and execute the RAM-TEST program given on page 104.

Next, you can install the PPI (IC32) and the PIC (IC7 — optional). Verify that the PPI is operational by loading and executing the PPI-TEST Program on page 104. Note that external hardware (an octal DIP switch and pull-up resistors) will be required to connect to the port under test.

The Multibug Monitor

The Monitor is a peripheral interface program, resident in non-volatile ROM or EPROM, which provides the user with a basic set of operating commands. This program is resident at address '0000'H, and is executed by a CPU 'reset'.

The ETI-685 can be operated with any monitor program which commences at address '0000'H. In most cases one of three monitors will be used. Firstly, there is the Signetics' PIPBUG monitor (transferred into EPROM) for serial VDUs, then there is the range of BINBUG monitors (produced by MicroByte) for S100 memory-mapped VDUs and thirdly, the SBCBUG monitor.

The SBCBUG is a 2716 2K single-rail EPROM containing two 1K monitor programs. The EPROM has been written to support either serial communications or memory-mapped video terminals by selecting the appropriate 1K monitor. The EPROM-type link field is set up for a 2758 single-rail 1K EPROM, and the A10 address pin is wired to either 0 V or +5 V to select the required 1K monitor (see later).

Both monitor programs have been styled on the BINBUG monitor and their commands are compatible to the BINBUG and PIPBUG monitors. The original PIPBUG monitor supported seven basic commands, each selected by a single alpha character, and these have been retained. The SERIAL MONITOR communicates with the VDU at 300 Baud via the 2650 Flag and Sense pins, and contains a few new subroutines to erase the VDU screen and print a sign-on message. The MEMORY-MAPPED MONITOR differs only from BINBUG in the keyboard-in subroutine, which now utilises the on-board 8-bit parallel port.

Monitor commands

Following are the commands and their respective functions:

A — Examine and **A**lter memory contents.
B — Set a program **B**reakpoint.
C — **C**lear a set breakpoint.
D — **D**ump a block of memory to tape (300 Baud binary).
G — Execute a program at a specified **'Go'** address.
L — **L**oad a tape file into memory.
S — Examine (**S**ee) and modify the CPU registers.

Now let us look in detail at each command and what they do.

Examine and alter memory: This command provides the user with a means of *displaying* the contents of a specified memory location and *altering* ▶

MONITOR SUBROUTINE SUMMARY

The monitor is included in the microcomputer system to provide the user with a basic set of operating peripheral interface commands. Many of the program subroutines contained in the monitor can be incorporated into user programs, and their use will greatly simplify interface programming requirements.

The following subroutine descriptions have been compiled to give the programmer a brief explanation of the function of each subroutine, details of the CPU registers affected and the maximum level of subroutine nesting achieved by each subroutine. The subroutines are listed in 'name' alphabetical order. The subroutines are only available in either the Serial or Memory-Mapped VDU Monitors.

Name	Address	Nest	Description
AGAP	027D	3	Outputs 'the number in Register 3' spaces (H'20').
BIN	0224	3	Inputs two hexadecimal characters from the keyboard and forms as one 8-bit byte in register R1. Serial monitor only.
BOUT	0269	3	The byte in R1 is output in binary as two hexidecimal characters.
CHIN	0286	3	An ASCII character is input to R0 from the keyboard.
COUT	02B4	2	The byte in R0 is output as an ASCII character.
CRLF	008A	3	Outputs a carriage return and line feed to VDU.
DLAY	039B	1	Produces a 1-bit delay at 300 Baud (approx. 3.3 ms).
DLY	039F	1	Produces a half-bit delay at 300 Baud (approx. 1.6 ms).
FORM	027B	3	Outputs three spaces (H'20') to the VDU.
GNUM	02DB	2	Places the next entry in the line buffer into R1 and R2. It ignores leading zeros and correctly interprets a 1, 2, 3 or 4-character entry.
INCT	00AB	1	Adds the two-byte number stored at TEMP and TEMP + 1 to R1 and R2 (with carry) and stores the two-byte result back in TEMP and TEMP + 1.
LINE	005B	3	Inputs up to 20 characters from the keyboard into the Line buffer. 'Delete' is used for entry corrections and CR or LF terminates the routine.
LKUP	028C	1	Converts an ASCII character in R0 into a hex value in R3. Generates an error message if a character is not hexadecimal.
STRT	00A4	1	Stores the number in R1 and R2 in TEMP and TEMP + 1.

the current contents if necessary. It contains an automatic address increment facility and may be used to load a program into consecutive memory locations.

This command is also used to *examine* the contents of memory following a program execution or breakpoint. The automatic address increment feature can be used to display the contents of consecutive memory locations.

Command format: **Axxxx<**

xxxx is the hexadecimal address of the memory location for display. Address leading zeros may be omitted.

Following the execution of this command by the entry of the 'carriage return' key (shown thus '<'), the memory address and its contents will be displayed in the following format:

xxxx . . . **zz** . . . []

zz is the current contents and [] represents the cursor location. To *alter* the memory contents, key in (in hex.) the required data. Leading zeros may be omitted. If no data is entered before a carriage return (CR) or line feed (LF) the contents will remain unaltered.

To *examine* the next memory location (auto address increment) enter a line feed. To exit from this command enter a carriage return.

Set breakpoint: A program *breakpoint* is primarily used during *program fault finding (debugging)* to terminate the execution of a program at a predetermined location. When the breakpoint is encountered, control is returned to the user, who is then able to use the other monitor commands to examine the microprocessor's internal registers or the program's memory locations.

Only one program breakpoint can be set at a time.

Command format: **Bxxxx<**

xxxx is the hexadecimal address of the first byte of the program instruction at which the program 'break' is required. Leading zeros may be omitted.

The breakpoint program operates by altering the contents of the program memory and cannot be used on programs which reside in ROM or EPROM. Two bytes of program data are replaced with '9B'H and the previous data is saved in reserved locations in the monitor's scratchpad RAM memory. When the breakpoint is encountered the original data is returned (auto-clear) to the program and the contents of the microprocessor's internal registers are saved in the monitor's scratchpad RAM.

Clear breakpoint: This command is used to erase a pending program breakpoint. The previous program data is returned and the breakpoint flag reset. If no breakpoint exists the monitor's error message is displayed.

Command format: **C<**

The user should note that the monitor's RAM memory is cleared following a processor reset and any program data stored there due to a pending breakpoint will be lost.

Dump to tape: The *DUMP* command provides the user with a means of saving programs on audio-quality magnetic tape. The SBC monitor outputs binary data at 300 Baud in the same format as the popular BINBUG monitor. This format is approximately six times faster than the original Signetics PIPBUG routine and represents the best compromise of speed and reliability. A suitable frequency shift keyed cassette tape interface must be connected to the serial port on-board.

Command format: **Dssss-ffff-eeee<**

ssss is the start address of the block of data to be saved.

ffff is the finish address of the block of data.

eeee is the optional auto-start program entry address.

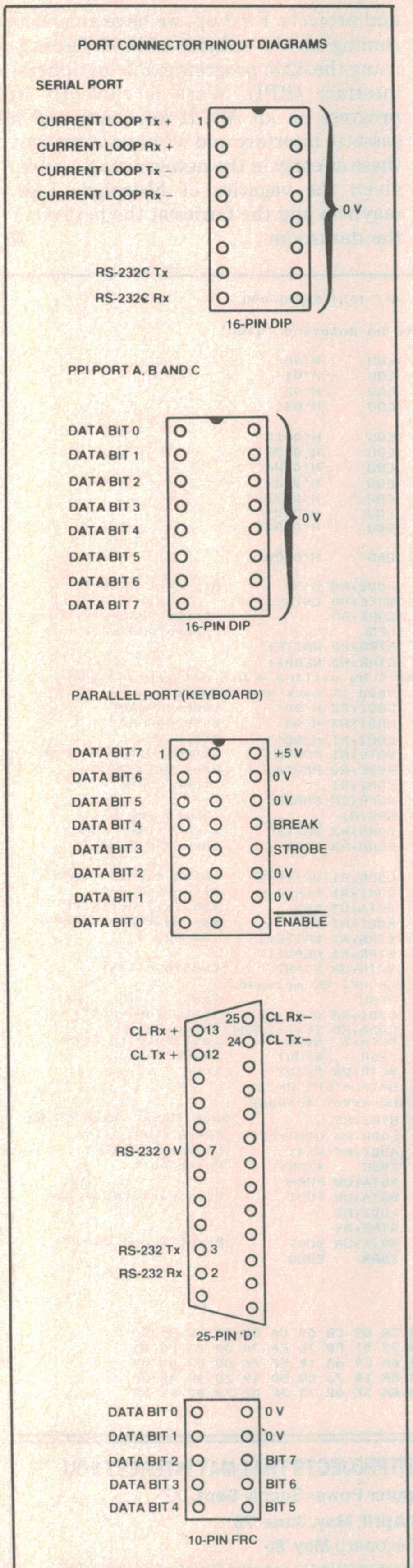
The output format consists of a leader of 32 nulls, a ':' header, a four-byte start address, a two-byte block length, a two-

2650 MEMORY ADDRESS ASSIGNMENT TABLE

Memory Sector (K)	Starting Address	Ending Address
1	0000	03FF
2	0400	07FF
3	0800	0BFF
4	0C00	0FFF
5	1000	13FF
6	1400	17FF
7	1800	1BFF
8	1C00	1FFF
- - - -	- - - -	- - - -
9	2000	23FF
10	2400	27FF
11	2800	2BFF
12	2C00	2FFF
13	3000	33FF
14	3400	37FF
15	3800	3BFF
16	3C00	3FFF
- - - -	- - - -	- - - -
17	4000	43FF
18	4400	47FF
19	4800	4BFF
20	4C00	4FFF
21	5000	53FF
22	5400	57FF
23	5800	5BFF
24	5C00	5FFF
- - - -	- - - -	- - - -
25	6000	63FF
26	6400	67FF
27	6800	6BFF
28	6C00	6FFF
29	7000	73FF
30	7400	77FF
31	7800	7BFF
32	7C00	7FFF
- - - -	- - - -	- - - -

NOTES:

1. Caution should be exercised when attempting program flow across page boundaries (shown thus '– – – –'). Refer to the Signetics 2650 Microprocessor Data Manual.
2. The 2650 Monitor resides in the first 1K sector (0000–03FF) and uses the next 64 bytes of RAM (0400–043F). User RAM commences at 0400.
3. The ETI-640 VDU resides in the 2K address sector 7800–7FFF.
4. The 2650 Disk Operating System (DOS) resides in the 2K address range 6800–6FFF and uses 2K of RAM at 7000–77FF.



PORT CONNECTOR PINOUT DIAGRAMS

byte SOT checksum, the data block and the block checksum. As the data block has a maximum length of 256 characters, the above process is repeated as often as necessary until the end address is reached.

GOTO (and execute): The *GOTO* command instructs the processor to execute the program at a specified hexadecimal address.

Command format: **Gaaaa<**

aaaa is the hexadecimal program execution address. Leading zeros can be omitted.

This command utilises the monitor 'line' subroutine to input a line of up to 20 characters into the 'line buffer'. As only five characters are used by the command, a further 14 characters may be entered (following a delimiting 'space') to pass additional parameters to the executing program.

Load from tape: The *LOAD* command is used to read back a binary data file from tape which has been recorded using the *DUMP* command, or an identical output format. The program extracts the start address from the data file and performs CRC checking. The tape load will be aborted and the monitor's error message displayed if a CRC error is detected.

Command format: **L<**

At the completion of an error-free load the program checks the end of the data file for an auto-start address. If an address is found the program will direct the processor to execute the program at that address. If no address is found the monitor will respond with the '*' prompt message.

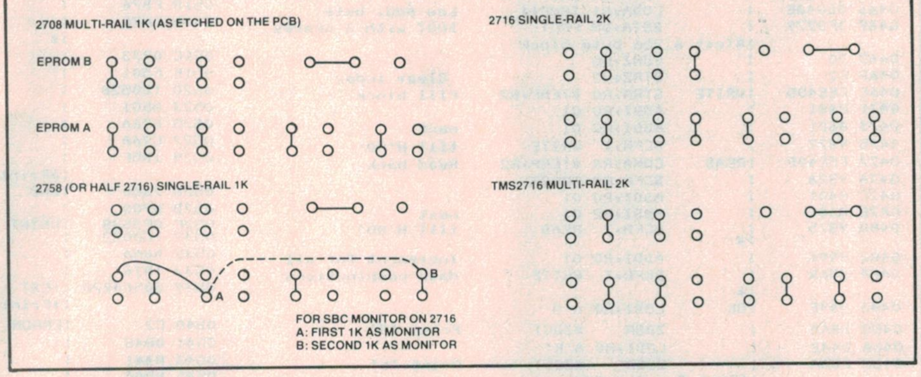
Examine (See and alter) CPU registers: This command is primarily used in conjunction with program breakpoints during program fault finding. Program breakpoints can be used to obtain a 'snapshot' of the program and the microprocessor's status immediately prior to the execution of the instruction at the breakpoint address.

This command is used to display the contents of any of the CPU's seven internal registers and two program status words following the execution of a program breakpoint. It also permits the user to alter the contents of any of these registers and resume program execution using the G command.

Command format: **Sn<**

The number 'n' (valid in the range 0 to 8) is used to select the particular register for display. ▶

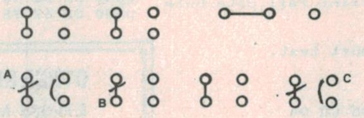
EPROM TYPE SELECTION FIELD (W6)



LINKING FOR THE MULTIBUG MONITOR ROM (V1.1)

The Multibug monitor EPROM is a single +5 V EPROM containing two 1K monitor programs which have been developed for the ETI-685 from the popular BINBUG monitor. This EPROM contains in the first 1K a monitor to interface with 300 Baud serial video terminals and in the second 1K a monitor to interface with S100 buss memory-mapped VDUs (such as the ETI-640). The required monitor program is selected by setting the EPROM's A10 address line to either 0 V (for the serial monitor) or +5 V (for the S100 monitor).

To use the 2716 EPROM the **W6** link field must be rewired from its 'etched' linking, which has been preset for multi-rail 2708 EPROMs. Three links must be broken and the adjacent links closed; all other links can be left 'as etched'. The 2716 EPROM must only be inserted at IC16.



Break the links marked thus

Close the links marked thus

Link A to B for the serial monitor
Link B to C for the S100 monitor

THE S100 MONITOR FOR ETI-640 TYPE VDUs.
The S100 Monitor uses the X5 parallel-in-port connection to interface to a standard parallel keyboard which produces a positive strobe. The port 'enable' line can be tied direct to 0 V and the keyboard should not draw more than 50 mA from the +5 V supply. A 'Break' function using the CPU Sense line can be used if the top half of the W2 link set is rewired (break 1-2 and close 2-3).

0 = register 0
1 = register 1 bank 0
2 = register 2 bank 0
3 = register 3 bank 0
4 = register 1 bank 1
5 = register 2 bank 1
6 = register 3 bank 1
7 = Program Status Word Upper
8 = Program Status Word Lower

The user may alter the displayed register's contents by entering a two-character hexidecimal number before entering a CR or LF. To display the next register (auto increment) enter a LF. To exit from this command, enter a CR.

Follow up

Plans are well advanced to follow up this project with a number of related articles and projects. First up, we have an article coming on the subject of **interfacing**, using the 8255 programmable peripheral interface (PPI). Work is currently in progress on an ASCII keyboard and a cassette interface and we hope to present these as early in the new year as possible, given the vagaries of Murphy's Law, mayhem and the fairies at the bottom of the darkroom . . . ●

```
                  :* MEMORY TEST PROGRAMME
                  :*
                  :* Execute by keying G440_SSSS_EEEE
                  :*
                  :EBUG     EQU     H'001D'
                  :MBUG     EQU     H'0022'
                  :CRLF     EQU     H'0025'
                  :ZOUT     EQU     H'002A'
                  :GAST     EQU     H'00A1'
                  :BOUT     EQU     H'0269'
                  :FOUT     EQU     H'0279'
                  :GNUM     EQU     H'02DB'
                  :*
                  :TEMP     EQU     H'040D'       Start Address pointer
                  :TEMQ     EQU     H'040F'       End Address pointer
                  :*
                  :         ORG     H'0440'
                  :*
0440 0402         :INIT     LODI,R0 02            Arith. compare
0442 93           :         LPSL                   Clear Carry, RS1
0443 7640         :         PPSU    FLAG          Set flag
0445 3F00A1       :         BSTA,UN GAST          Fetch Start Add.
0448 E504         :         COMI,R1 04
044A 9805         :         BCFR,EQ CONT          Not 0400-4FF block
044C 0505         :         LODI,R1 05            then start at 0500
044E CD040D       :         STRA,R1 TEMP          Start Add. in TEMP
0451 3F02DB       :CONT     BSTA,UN GNUM          Fetch End Add.
0454 CD040F       :         STRA,R1 TEMQ          End Add. in TEMQ
0457 20           :         EORZ,R0                Hex '00'
0458 CC040E       :         STRA,R0 TEMP+1        Only 256 byte blocks
045B CC0410       :         STRA,R0 TEMQ+1
                  :*Print Address block to test
045E BBA5         :BLOCK    ZBSR    *CRLF         New line
0460 0D040D       :         LODA,R1 TEMP          High Add. byte
0463 3F0269       :         BSTA,UN BOUT
0466 0D040E       :         LODA,R1 TEMP+1        Low Ádd. byte
0469 3F0279       :         BSTA,UN FOUT          BOUT with 3 spaces
                  :*Test a 256 byte block
046C 20           :         EORZ,R0
046D C2           :         STRZ,R2                Clear index
046E CEE40D       :WRITE    STRA,R0 *TEMP,R2      Fill block
0471 8401         :         ADDI,R0 01
0473 8601         :         ADDI,R2 01            next
0475 9877         :         BCFR,Z  WRITE         till H'00'
0477 FEE40D       :READ     COMA,R0 *TEMP,R2      Read back
047A 9826         :         BCFR,EQ ERROR
047C 8401         :         ADDI,R0 01
047E 8601         :         ADDI,R2 01            next
0480 9875         :         BCFR,Z  READ          till H'00'
                  :*
0482 8401         :         ADDI,R0 01            Increment for all
0484 9868         :         BCFR,Z  WRITE         data combinations
                  :*
0486 044F         :OK       LODI,R0 A'O'
0488 BBAA         :         ZBSR    *ZOUT         Print "O"
048A 044B         :         LODI,R0 A'K'
048C BBAA         :         ZBSR    *ZOUT         Print "K"
                  :*Test for End Address
048E 0D040D       :         LODA,R1 TEMP
0491 ED040F       :         COMA,R1 TEMQ
0494 1C0022       :         BCTA,EQ MBUG          End of test
0497 1D001D       :         BCTA,GT EBUG          Address wrong
049A 8501         :         ADDI,R1 01            Next block
049C CD040D       :         STRA,R1 TEMP
049F 1F045E       :         BCTA,UN BLOCK
                  :*Report error and Address
04A2 C3           :ERROR    STRZ,R3                Save fail Data byte
04A3 0D040D       :         LODA,R1 TEMP          High Add. byte
04A6 3F0269       :IBOUT    BSTA,UN BOUT
04A9 02           :         LODZ,R2                Low Add. byte
04AA C1           :         STRZ,R1
04AB 3F0279       :         BSTA,UN FOUT
04AE 03           :         LODZ,R3                Print fail data byte
04AF C1           :         STRZ,R1
04B0 3BF5         :         BSTR,UN *IBOUT+1
04B2 9B22         :         ZBRR    MBUG          Abort test.
                  :*
00 ERRORS DETECTED

0440 04 02 93 76 40 3F 00 A1 E5 04 98 05 05 05 CD 04
0450 0D 3F 02 DB CD 04 0F 20 CC 04 0E CC 04 10 BB A5
0460 0D 04 0D 3F 02 69 0D 04 0E 3F 02 79 20 C2 CE E4
0470 0D 84 01 86 01 98 77 EE E4 0D 98 26 84 01 86 01
0480 98 75 84 01 98 68 04 4F BB AA 04 4B BB AA 0D 04
0490 0D ED 04 0F 1C 00 22 1D 00 1D 85 01 CD 04 0D.1F
04A0 04 5E C3 0D 04 0D 3F 02 69 02 C1 3F 02 79 03 C1
04B0 3B F5 9B 22
```

```
                  :* 8255 PPI TEST PROGRAMME
                  :*
                  :*Execute by entering "G500"
                  :*
                  :PORTA    EQU     H'00'
                  :PORTB    EQU     H'01'
                  :PORTC    EQU     H'02'
                  :CNTRL    EQU     H'03'
                  :*
                  :EBUG     EQU     H'001D'
                  :MBUG     EQU     H'0022'
                  :ZOUT     EQU     H'002A'
                  :BOUT     EQU     H'0269'
                  :FORM     EQU     H'027B'
                  :FOUT     EQU     H'0279'
                  :COUT     EQU     H'02B4'
                  :*
                  :         ORG     H'0500'
                  :*
0500 0480         :INIT     LODI,R0 H'80'       All port output
0502 D403         :         WRTE,R0 CNTRL
0504 20           :         EORZ,R0               Arith. compare, clear
0505 93           :         LPSL                  carry and RS1.
0506 C809         :         STRR,R0 WRITE+1
0508 C809         :         STRR,R0 READ+1
                  :*Test port by writing a bit pattern and then
                  :*        read it back and compare
050A 0600         :START    LODI,R2 H'00'       Loop counter
050C 0708         :TEST     LODI,R3 H'08'       Byte counter
050E 0580         :         LODI,R1 H'80'       Bit 8 set
0510 D500         :WRITE    WRTE,R1 PORTA       Output data pattern
0512 5400         :READ     REDE,R0 PORTA       Read data back
0514 E1           :         COMZ,R1              then compare
0515 9829         :         BCFR,EQ ERROR
0517 51           :         RRR,R1               Shift set bit
0518 FB76         :         BDRR,R3 WRITE       eight times then
051A FA70         :         BDRR,R2 TEST        loop 256 times
                  :*
051C 0973         :         LODR,R1 WRITE+1     Fetch current port
051E E501         :         COMI,R1 PORTC-1     At port C yet
0520 1D052B       :         BCTA,GT END         Yes, all ports OK
0523 8501         :         ADDI,R1 01          No, next port number
0525 C96A         :         STRR,R1 WRITE+1      loaded
0527 C96A         :         STRR,R1 READ+1
0529 1B5F         :         BCTR,UN START       Continue test
                  :*Print the PPI OK message
052B 7600         :END      PPSU                FLAG
052D 0700         :         LODI,R3 0           Init. loop cntr.
052F 0F2538       :PRINT    LODA,R0 TEXT-1,R3+  Fetch byte
0532 1C0022       :         BCTA,Z  MBUG        Last byte is zero
0535 BBAA         :         ZBSR    *ZOUT       Print byte
0537 1B76         :         BCTR,UN PRINT       Loop
0539 50504920     :TEXT     DATA A'PPI OK',0
                  :*Print the error message
0540 C2           :ERROR    STRZ,R2             Save found value in R2
0541 0848         :         LODR,R0 START+1     Fetch port value
0543 8441         :         ADDI,R0 H'41'       Convert to Alpha
0545 BBAA         :         ZBSR    *ZOUT       Print port
0547 3F027B       :         BSTA,UN FORM
054A 3F0279       :         BSTA,UN FOUT        Print written value
054D 02           :         LODZ,R2
054E C1           :         STRZ,R1
054F 3F0269       :         BSTA,UN BOUT        Print found value
0552 9B1D         :         ZBRR    EBUG
                  :*
00 ERRORS DETECTED

0500 04 80 D4 03 20 93 C8 09 C8 09 06 00 07 08 05 80
0510 D5 00 54 00 E1 98 29 51 FB 76 FA 70 09 73 E5 01
0520 1D 05 2B 85 01 C9 6A C9 6A 1B 5F 76 00 07 00 0F
0530 25 38 1C 00 22 BB AA 1B 76 50 50 49 20 4F 4B 00
0540 C2 08 48 84 41 BB AA 3F 02 7B 3F 02 79 02 C1 3F
0550 02 69 9B 1D
```

OTHER RELATED ETI PROJECTS THAT MAY INTEREST YOU

ETI-635 Microcomputer Power Supply **Sept. '77**
ETI-640 S100 VDU **April, May, June '78**
ETI-636 S100 Motherboard **May '80**
ETI-681 S100 Programmable Character Generator **June '80**
ETI-682 S100 PROM Board **March '81**