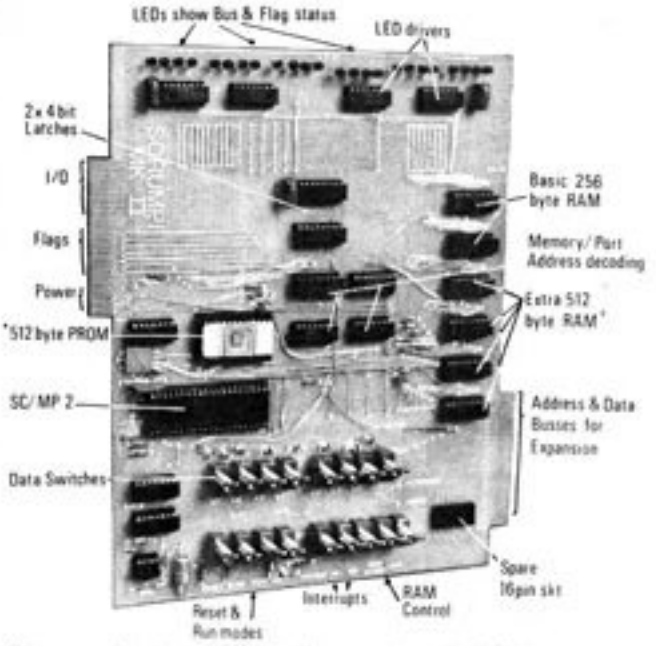
SCRUMPI is good for you!

and Education, Training, Development, In-circuit testing,



All component & sockets included in Basic kit except those marked * (Full kit only).

SC/MP MPU Chips by NATIONAL

PRICES:	£
SCRUMPI 1	46.30
SCRUMPI 2B	55.56
SCRUMPI 2F	74.07

8% VAT Excluded.

We also stock:

E.T.I. System 68 MPU kits, VDU kits, Case kit. Comprehensive selection of Hardware & Software support. Digital clock chips, kits, displays. Please send SAE for our catalogue.

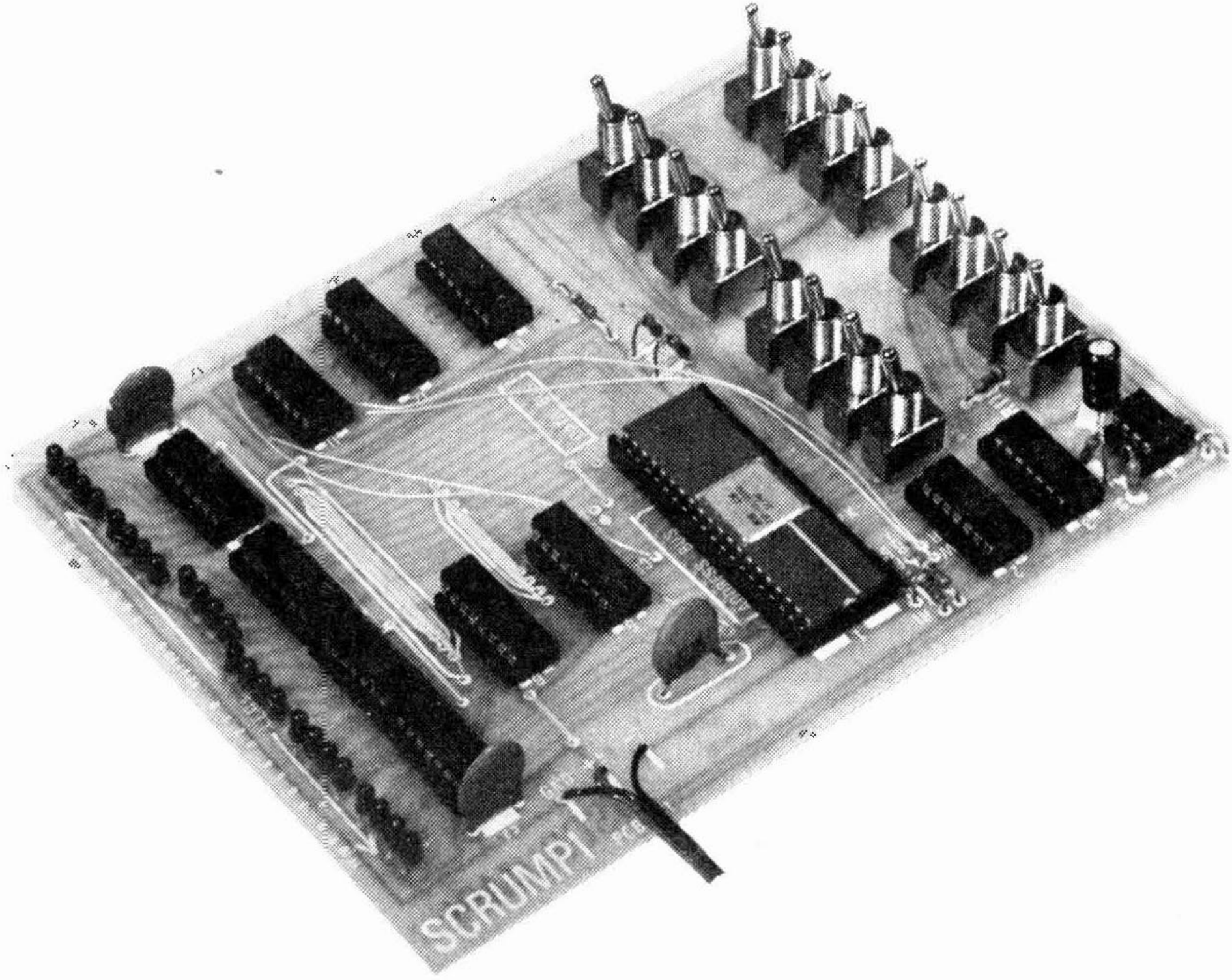
Products from:

Fairchild, General Instruments, Liton, Litronix, Mostek, Motorola, National Semi, S.M.C., Vero.

BYWOOD ELECTRONICS

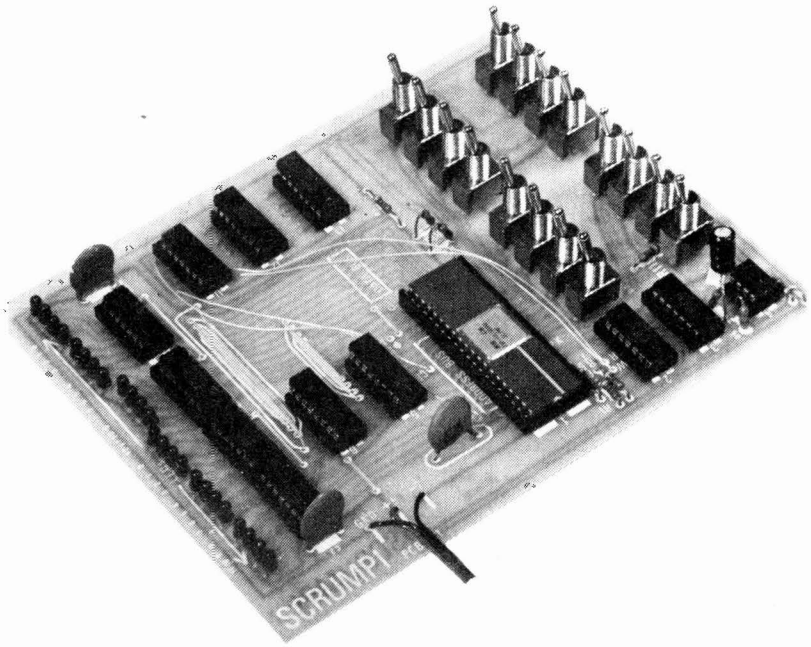
68 Ebberns Road, Hemel Hempstead, HP3 9QRP.

Tel. 0442-62757


SCRUMPI

SCRUMPI KIT REVIEWED

By D.B.JOHNSON-DAVIES



WITH all the articles on microprocessors that have been appearing recently there must be a number of readers who feel that they will never really understand micros until they have actually used and programmed one, and who are therefore wondering how they can get their hands on a system as cheaply as possible. Bywood's "Scrumpi" kit may be the answer as it provides a self-contained development system using the minimum of parts, and at £55·56 costs less than most other solutions.

DESIGN CENTRE

It is designed around National Semiconductor's SC/MP, an 8-bit low cost micro which has a simplified instruction set and architecture in aid of economy. Its lack of sophistication means that most programs require more steps to achieve the same as a micro with a greater variety of instructions, registers, and addressing modes. On the other hand the chip provides a good selection of control inputs and outputs eliminating the need for I/O devices in simple applications: three outputs, flags F0 -1, and -2, are controlled by bits in the status register and two inputs, SENSE-A and -B, set bits in the status register. In addition, SENSE-A can optionally cause an interrupt. Serial I/O can be performed via the SIN and SOUT pins using the extension register.

In "Scrumpi" the states of the twelve address lines and the eight data lines are displayed in binary form on l.e.d.s driven by CMOS buffers. The data lines can be taken to ground by eight programming switches. The memory consists of two 256 × 4 bit memory chips, providing 256 words of read/write memory. Two four-bit latches act as an eight-bit I/O port in which each set of four can be wired as either inputs or outputs. They are enabled by the highest address line, All, so that all addresses in the range X'800 to X'FFF (where the X' signifies hexadecimal notation) are mapped on to the one I/O port.

FUNCTIONS

The various functions of the kit are controlled by a flip/flop, a 555 timer and some NAND gates, and are selected by a further eight toggle switches. These are: RESET, SLOW, STEP, RUN/HALT, PROTECT, SENSE-A, SENSE-B, and LOAD. The circuit of the kit is shown in simplified form in Fig. 1.

All the components are mounted on the single-sided fibreglass printed-circuit board; twenty wire links are needed to complete the connections. A double-sided board would add little to the cost and it is difficult to see why one was not used. Apart from this inconvenience construction was straightforward. All the parts were supplied and sockets were provided for all the i.c.s. The switches are soldered to the board by their terminals, but the whole board could be mounted behind a suitably drilled panel to make a more robust unit. The circuit needs a power supply of +5V and −7V and these can be derived from a single 12V supply with a 5V Zener diode.

HOW IT WORKS

"Scrumpi" gets away without the need for any monitor program in ROM by making cunning use of the control signals provided by the MPU. The memory is programmed by a primitive form of DMA (direct memory access) by automatically stopping the MPU during each instruction cycle. All the instructions consist of at least one read cycle—the "instruction fetch" which gets the op-code from memory. For example, SR (shift right) has only one cycle. For the two-word instructions there is a second read cycle to fetch the displacement or data; for example, LDI (load immediate) has a second read cycle to get the data from the next location. Store instructions obviously have an additional write cycle, and the two longest instructions ILD (increment and load) and DLD (decrement and load) consist of three read cycles and one write cycle.

The MPU is stopped by taking the NHOLD input low during the input or output cycle, and this extends the cycle indefinitely until NHOLD is

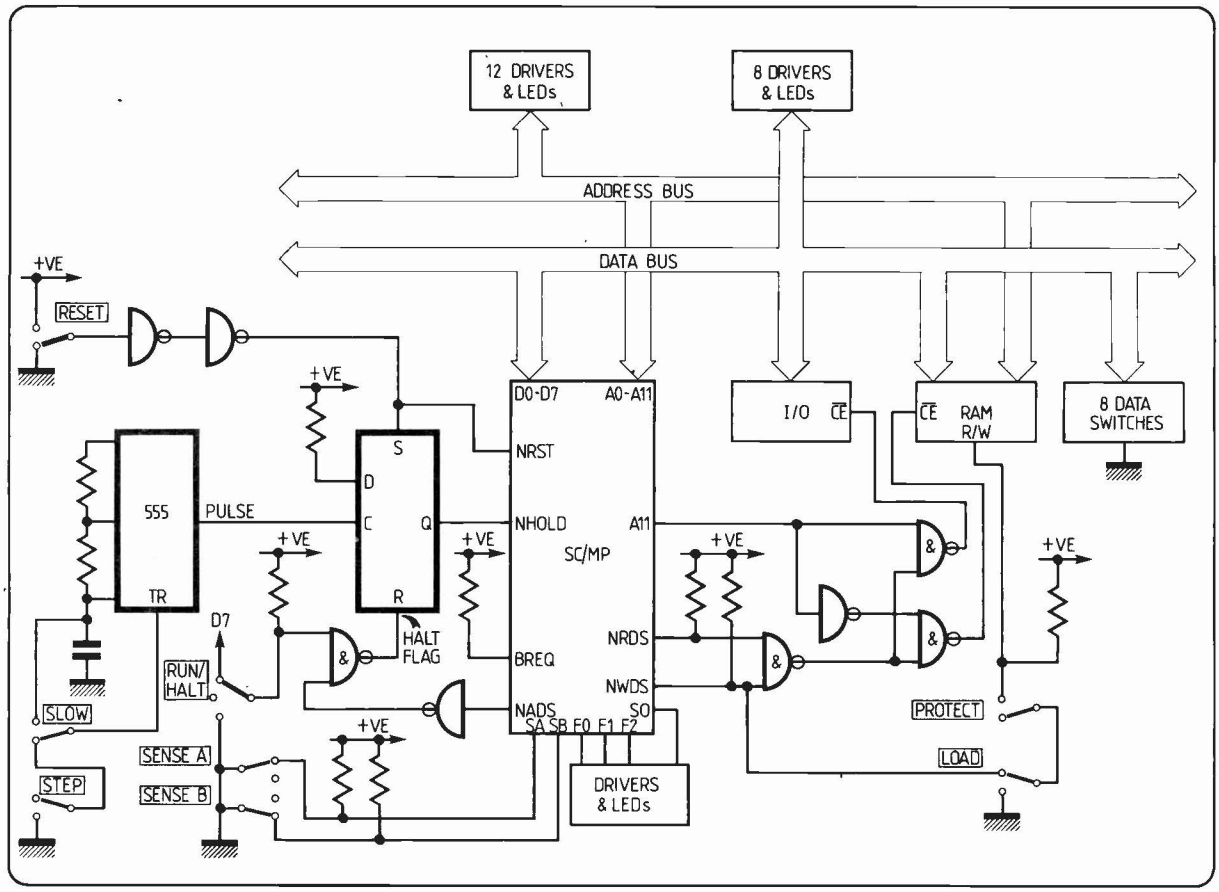


Fig. 1. Simplified circuit diagram of "Scrumpi". The eight function switches control the various modes of operation of the kit

returned high. In "Scrumpi" the NHOLD input is controlled by a D-type flip/flop, which is reset by the pulse on the NADS (address strobe) output occurring at the start of each input/output cycle; see Fig. 2. This puts NHOLD low extending the cycle until the flip/flop is clocked by a pulse from the output of the 555 timer; the MPU is then released to run until the next NADS pulse at the start of the next cycle.

In a read cycle the MPU is stopped with NRDS (read strobe) low which is used to enable the memory in write mode, causing the data in memory at the location addressed by the MPU to be put on to the data bus. In a write cycle the MPU is stopped with NWDS (write strobe) low which enables the memory to read data put on to the data bus by the MPU.

"Scrumpi" is programmed by stepping or running the MPU to the required address, putting the eight data switches to the required eight-bit binary value, and then operating the LOAD switch. This switch puts the memory chips into read mode and so loads the value on the data bus into the memory location. Programming can only be done during a read cycle since in a write cycle the MPU is putting data on to the data bus too.

ENTERING A PROGRAM

To make this operation clear consider how one would enter the following program which uses the instruction ILD (increment and load) to increment the contents of location X'032 using program-counter-relative addressing. The program follows:

Address:	Data:	
001	A8	ILD
002	30	disp.
003	next instruction	
.	.	.
.	.	.
032	04	data

First all memory locations are set to X'00 by running the MPU while loading with the data switches set to 0. Operating RESET now starts the MPU at address X'001 (Fig. 2 (a)). The required value, X'A8, is set on the data switches (as 10101000) and LOAD operated to store this to memory. The STEP switch will now cause the 555 timer to deliver a pulse, releasing the MPU from hold state to execute the instruction. Since the ILD instruction is four cycles long (see Fig. 2) a further three operations of the STEP switch are needed to complete execution of it.

The second cycle fetches the displacement (X'30); see Fig. 2 (b). This added to the program counter gives the effective address of the data: X'032. The third cycle fetches the data from this location. Finally the fourth cycle writes the incremented value back to the same location (Fig. 2 (d)).

The kit could thus be said to provide a hardware trace facility by making use of the SC/MP's control signals; "Scrumpi" makes an asset out of economy and provides a graphic demonstration of how each instruction behaves in action.

BREAKPOINTS

"Scrumpi" also provides a hardware breakpoint facility. The code X'00 is interpreted by the MPU as a HALT instruction; in fact execution of it will pulse the H-flag which is put out on line D7 when NADS is

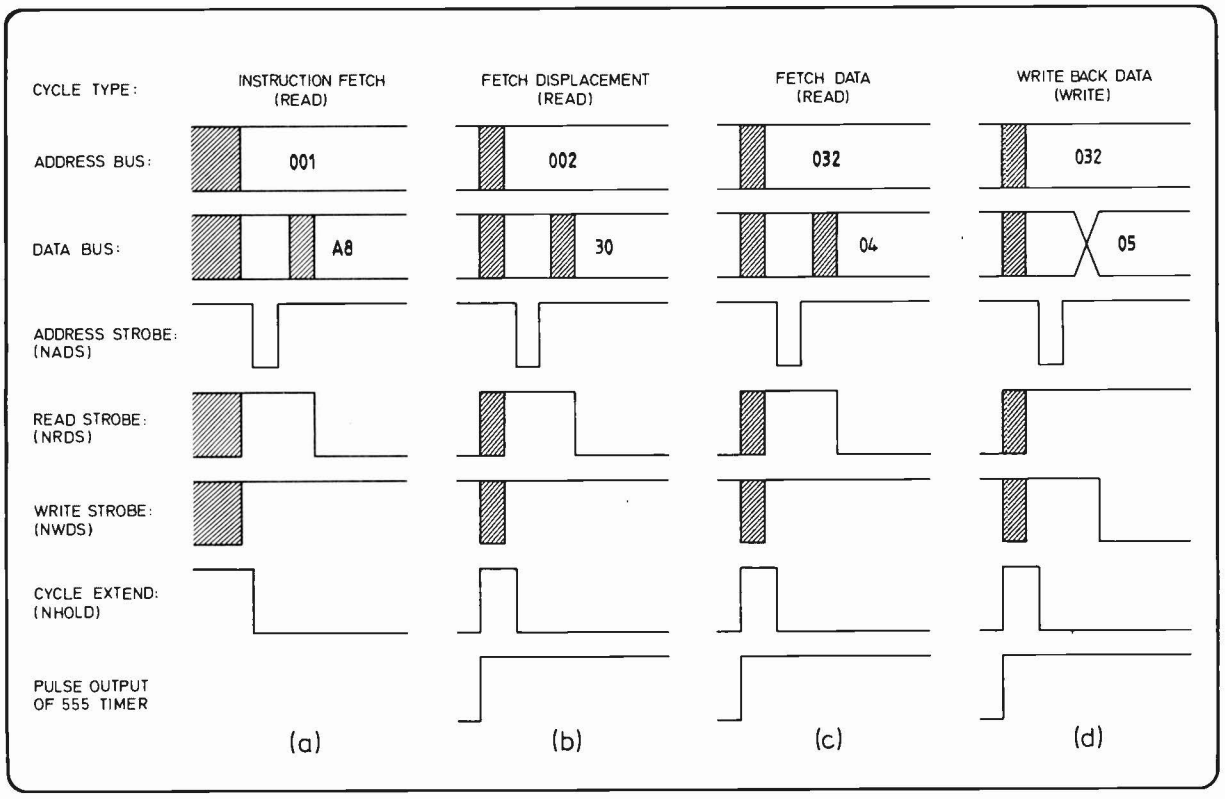


Fig. 2. Timing diagram showing how the control input NHOLD is used to stop the MPU after each of the read/write cycles of the four-cycle instruction ILD (increment and load). The shading indicates that the outputs concerned are in high-impedance state

low. With the RUN/HALT switch in the correct position D7 is taken to the flip/flop, gated by NADS. A HALT instruction placed anywhere in a program will then act as a breakpoint; executing it will rest the flip/flop and put the MPU into hold state.

It should be obvious from the foregoing description that programming is a tedious business; the data switches must be set for each instruction to be entered and although the conversion from hexadecimal to binary becomes automatic after a time, the process is error-prone and slow which discourages attempts at large programs; added to which is the knowledge that the program will evaporate on switching off the power.

JUMP TO SUBROUTINE

A fair amount of ingenuity is needed to get some programs into memory, especially if they contain conditional jumps, as the only access to a location is by executing instructions which lead to it. It might therefore be prudent to leave the first seven locations free so they can be loaded with the following "jump to subroutine":

Address:	Data:		
001	C4	LDI	
002	01	X'01	load P3 with X'0123
003	37	XPAH P3	
004	C4	LDI	
005	23	X'23	
006	33	XPAL P3	
007	3F	XPPC P3	

Execution of this will cause a jump to X'0124. Any location can be reached by loading the correct address in X'002 and X'005.

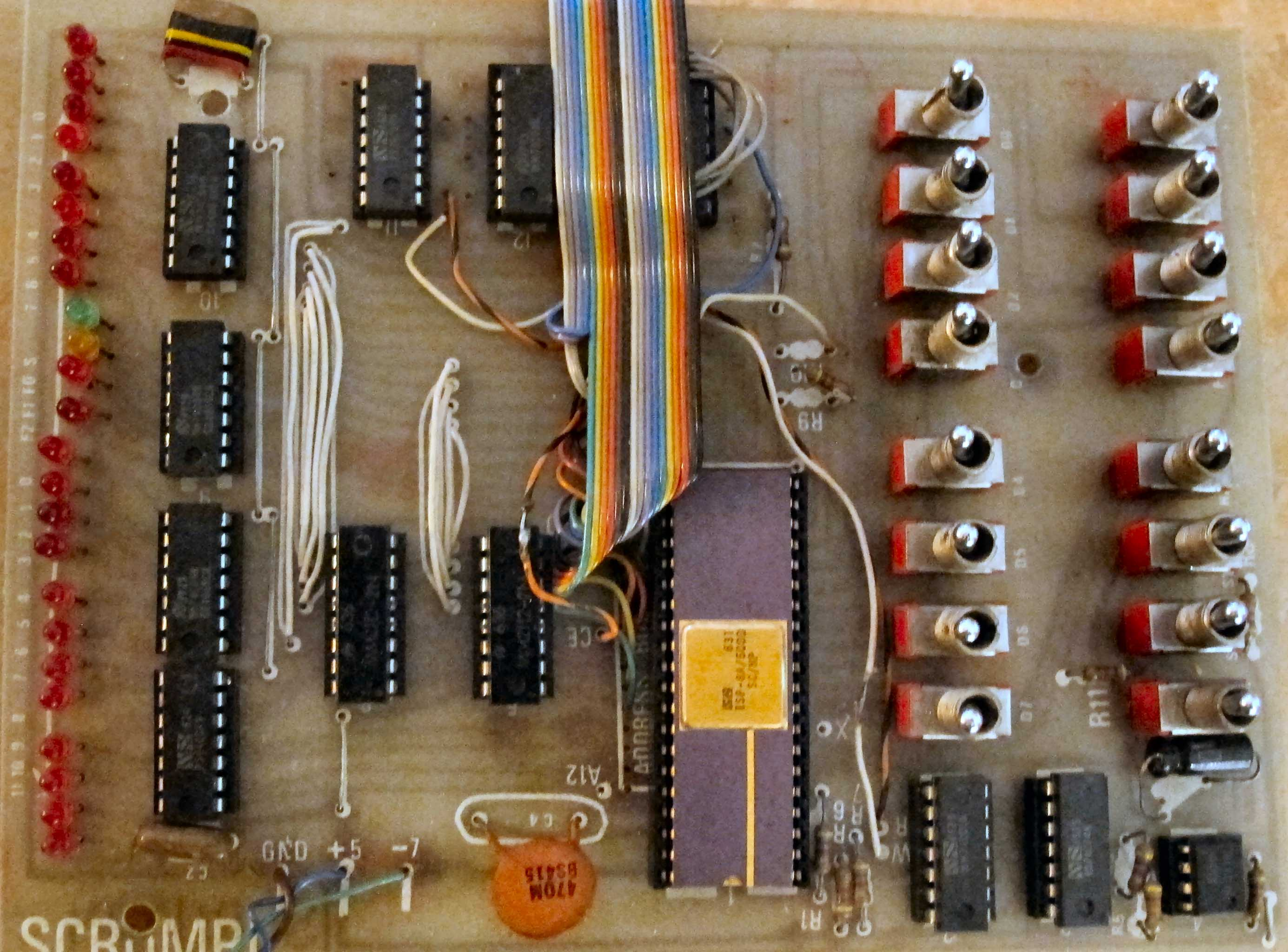
KIT DESIGN

One worrying aspect in the design of this kit is the way programming is achieved by using the data switches to ground the data lines linking the MPU and memory. Suppose that X'FF is to be altered to X'00 at a certain location. In this case all eight outputs from the memory devices are, until the LOAD switch is operated, driving into a short-circuit. The "on" resistance of the outputs is about 30 ohms so dissipation under these conditions could reach 3 watts; the maximum recommended dissipation is 1 watt. This is one reason for the instruction to load the memory with X'00 before programming.

Operation of the LOAD switch was also somewhat erratic; it is surprising that the spare flip/flop was not used to eliminate contact-bounce. Mr. Miller-Kirkpatrick of Bywood is currently involved in designing a new version of the kit which may overcome these problems.

CONCLUSION

Aspiring computer programmers who want to forget about the hardware the moment "Scrumpi" is working would be well advised not to spend their money on this kit; it is just not a practical proposition to write more than the simplest of programs on the system. To quote from the manual: "You will very soon realise that "Scrumpi" is very limited as it stands because it does no more than light up l.e.ds." The constructor who is more interested in hardware than software, however, could use "Scrumpi" to form the base from which to build a more extensive microprocessor system.


GND
+5
-7
CE
A12
R9
R1
C4

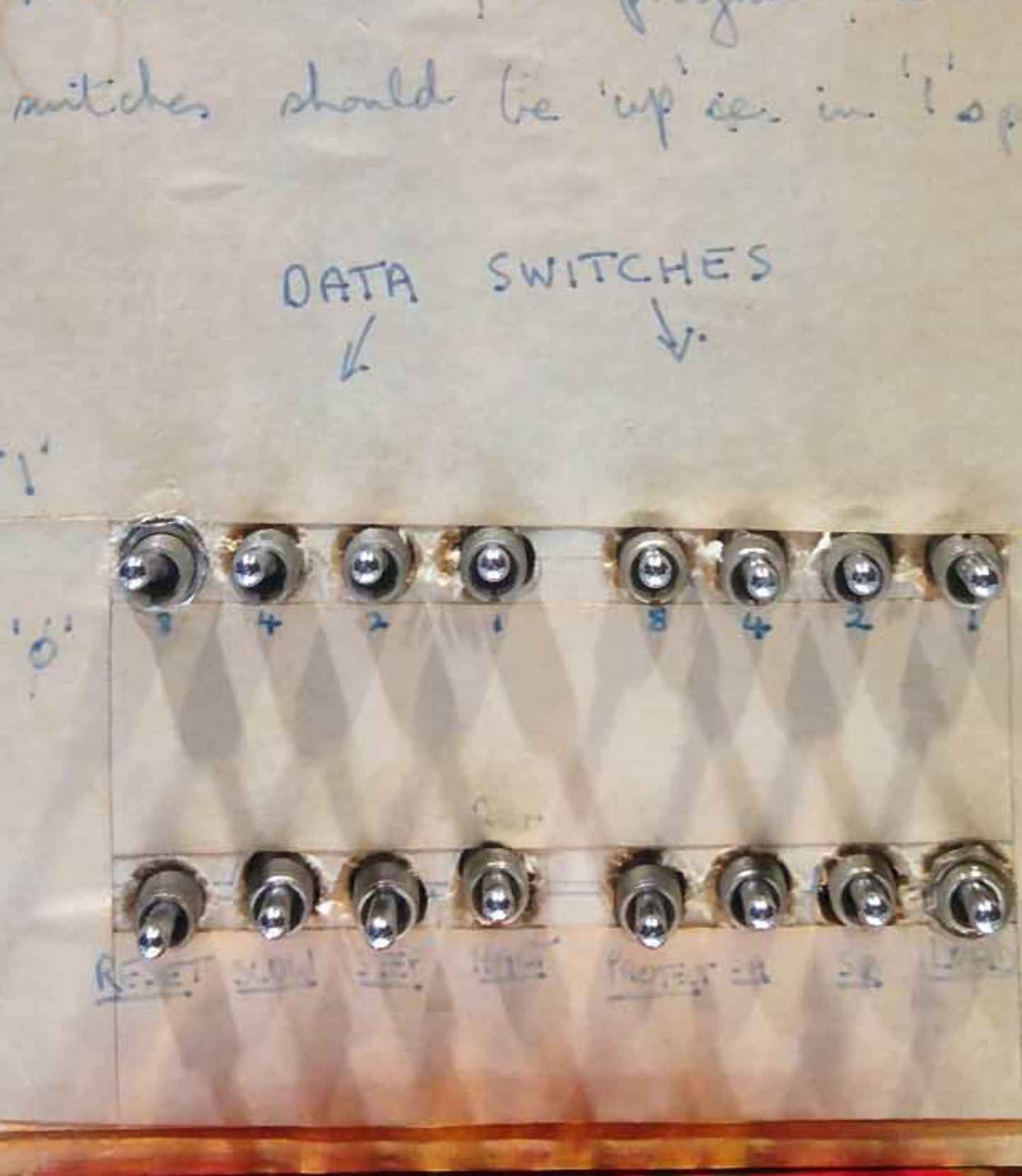
-ve (Black) +ve (Red)

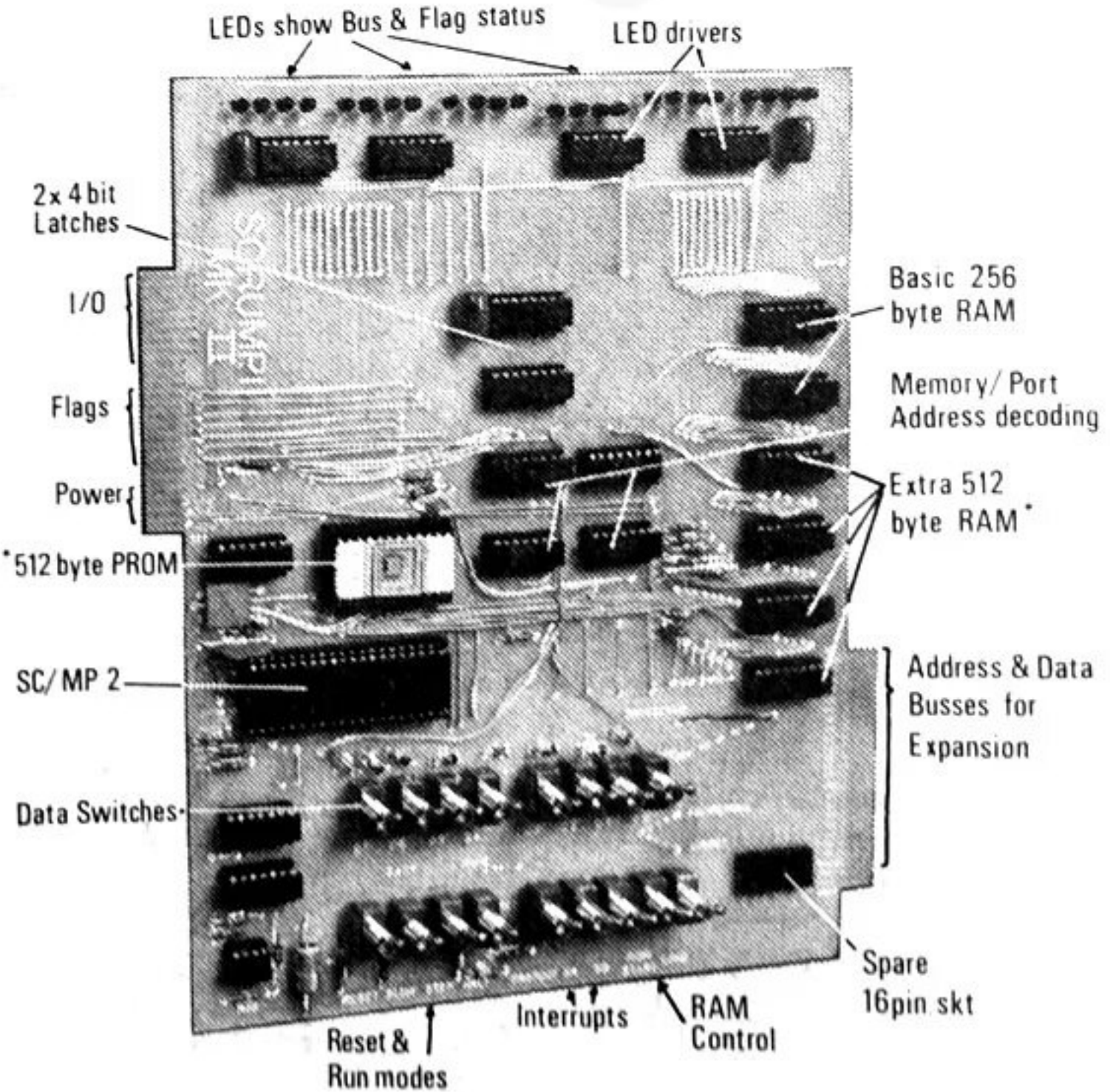
Before switching on check polarity!

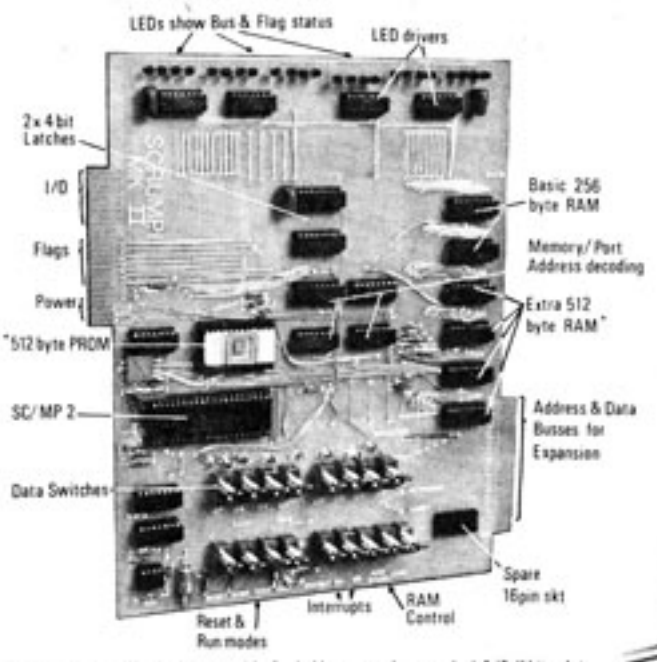
SC/MP MICROPROCESSOR

When running a program all data switches should be 'up' ie. in '1' position

DATA SWITCHES




LEDs show Bus & Flag status
LED drivers
2x 4 bit Latches
I/O
Flags
Power
*512 byte PROM
SC/MP 2
Data Switches
Basic 256 byte RAM
Memory/Port Address decoding
Extra 512 byte RAM*
Address & Data Busses for Expansion
Spare 16pin skt
RAM Control
Interrupts
Reset & Run modes



All component & sockets included in Basic kit except those marked * (Full kit only).

2+3=?

Both SCRUMPI 2 and SCRUMPI 3 are powerful MPU kits in their own rights . . . Together they make one of the most powerful MPU Hardware/Software development systems available. Please write for details of combined kit/upgrade offer.

SCRUMPI 2 is a single board MPU system based on the SC/MP2 microprocessor chip. Switches allow Single-step/Halt/Run modes with PROM or RAM bootstraps. RAM protect and interruption. Basic kit includes all IC sockets, all ancilliary components, SC/MP2, drivers, decoders, latches and 256 bytes of RAM. Full kit includes additional 512 byte PROM & 512 byte RAM.

SCRUMPI 2B £55.56+VAT
SCRUMPI 2F £74.07+VAT

SCRUMPI 3 is a single board MPU system based on the SC/MP2 microprocessor chip and including Keyboard, VDU interface, UART, two 8 bit parts, 128 byte RAM, 1K PROM and sockets for additional 1K PROM & 1K RAM.

SCRUMPI 3 Basic kit **£154.92,** with case & PSU **£189.75.**

Quantity discounts are available to OEM users, Distributors, Retailers and Training Establishments.