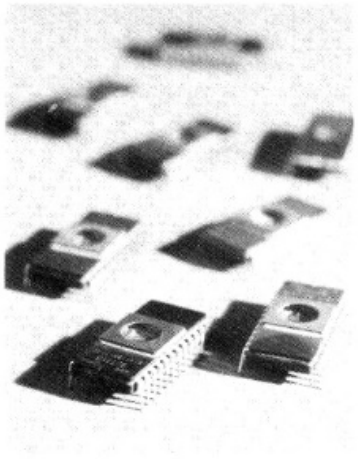
software cruncher and puncher

disassemble Junior Computer software and program 2716 EPROMs

Whereas developing one's own software is often like taking a leap in the dark, analysing other people's programs can sometimes be quite a revelation. In either case a disassembler is called for, such as the one described here. In addition, it is a useful aid towards 'BASIC' conversion. And, as the software cruncher is stored in 2716 EPROM, why not include an EPROM programming program, (to use up the remaining EPROM space), together with the EPROM hardware published in January?

Junior Computer owners regularly send us programs that they have written for 'their' machine and Elektor's editorial staff dutifully try them all out and 'unravel' them with the disassembler. Unfortunately, the task is not always a rewarding one and occasionally the work resembles that of a pathology lab. Nevertheless, it is gratifying to receive so many sparks of initiative!
Any attempt in this direction automatically calls for a disassembler, but that is by no means the only reason for having a 'software cruncher'. Used in combination with the editor and assembler the disassembler enables operators both to write their own programs and decode those gleaned from friends or magazines.



The details

The software cruncher is stored in 2716 EPROM. The software occupies the address range $F800 . . . $FFFF. The EPROM may either be mounted on a RAM/EPROM card or on the mini EPROM card published in the April issue. Locations $F800 . . . $FDD9 store the actual disassembler.
$FDDA . . . $FFF9 contain the 'EPROM PROGRAMMING UTILITIES' (which are described later on in the article) and $FFFA . . . $FFFF include the vector data with which JC owners are already familiar.
The disassembler section of the software 'cruncher' is shown in Table 1. After initiation (enter the start address $FC4E through PM!) the computer reports back by defining the relevant function keys. The D key is operated to enter two addresses which 'cordon off' the memory range that is to be disassembled (ending in CR). In the example given in Table 1 this comprises $0200 . . . $022F. Note that the end address must be entered and that the 'end address + 1' rule does not apply here.
This is followed by the message 'L, P, SP?'. By depressing the L key the operator can disassemble the entire memory range 'in one go'. The P key, on the other hand, does this in blocks of 15 instructions (a full TV screen, the top line being the last one to be printed *before* P was operated) and the space bar SP allows each instruction to be disassembled in turn and is therefore the slowest method.
The 'crunched' program in Table 1 gives an idea of the type of information that is printed. Table 2 shows the Hex dump of the disassembler. First of all, the address and the op code of the instruction are displayed followed by the byte(s) contained in the instruction. Then the mnemonics (the instruction 'shorthand') are printed preceded by several spaces. Wherever relevant, the line ends with the operand data. The displacements involved in conditional jump instructions are 'translated', so to speak, as the 'jump address'.
Data that is not acknowledged to be the op code of an instruction has the mnemonic consisting of three American AT symbols assigned to it (see address 021E, for example). Such data is one byte long. Note that FF is not acknowledged as a label op code.
Then R is operated and the program returns to PM. What could be easier?

Table 1.

```
FC4E
FC4E A9 R
VALID COMMANDS: A D H L P R SP

D
DISASSEMBLE: 200, 22F
L, P, SP ?
L
0200 A9  00     LDA # $00
0202 AD  01 02  LDA $0201
0205 A5  03     LDA $03
0207 A1  04     LDA ($04,X)
0209 B1  05     LDA ($05),Y
020B B5  06     LDA $06,X
020D BD  07 08  LDA $0807,X
0210 B9  09 0A  LDA $0A09,Y
0213 B6  0B     LDX $0B,Y
0215 20  0C 0D  JSR $0D0C
0218 4C  0E 0F  JMP $0F0E
021B 6C  10 11  JMP ($1110)
021E 77         @@@
021F FF         @@@
0220 00         BRK
0221 00         BRK
0222 CA         DEX
0223 C8         INY
0224 E8         INX
0225 0A         ASL A
0226 F0  12     BEQ $023A
0228 D0  FE     BNE $0228
022A B0  34     BCS $0260
022C 90  EE     BCC $021C
022E FA         @@@
022F 00         BRK
```

1

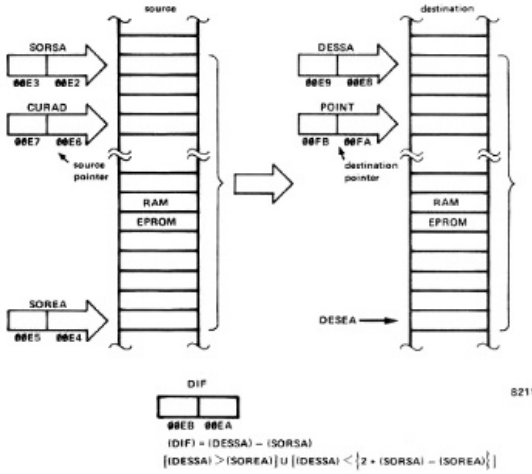


Figure 1. The memory ranges involved in the EPRUTL routine. Operating the M, R, V or F key causes the CURAD and POINT pointers to move through every location from SORSA and DESSA to SOREA and DESEA, respectively.

As for the H and A keys, depressing H is equivalent to operating M during PM and A represents 'ASCII dump'. Thus, a hex dump is printed after two address entries followed by CR (see table 2). The A key causes a hex dump to be printed showing the ASCII code of any alphanumeric character within the $20 . . . $7E range. In the case of data outside this range, a space appears. This feature allows data, such as computer messages that need printing, to be located swiftly. Once readers manage to crunch the disassembler they will see that this is riddled with such messages.

Not only, but also . . .

The printing operation of the dump or listing may be interrupted by depressing the BRK key. The BRK jump vector leads the 6502 μP to a central point in the program where it waits for a (new) key to be operated.

When two addresses are entered for the purpose of defining a listing or dump, the second address must be higher than the first. Otherwise, the two addresses will have to be re-entered, only this time in the right order please!

As well as storing data in much used memory locations in pages 00 and 1A, the software cruncher must dispose of $0010 . . . $0027. $0028 must now also be added to accommodate the extra software. Operators must be careful not to use these memory locations for the program they wish to 'sort out'.

Software puncher

As mentioned earlier, now that we have the necessary hardware (Elektor January 1982), a start can be made on loading RAM or EPROM software into 2716s. The program is started by way of PM at address $FDDA. After initialisation, the name of the program is printed along with a list of valid keys. Then the *parameter key, P,* should be depressed so as to define the address range by entering three addresses, as shown in figure 1. First of all, the 'FIRST, LAST SOURCE ADDRESS' must be specified, in other words, the SORSA and SOREA addresses at either end of the data block that is to be stored or relocated. Make sure SOREA has a higher number than SORSA, as otherwise the entry procedure (first address – comma – last address – CR) will have to be repeated. Next, enter the 'FIRST DESTINATION ADDRESS'. This is known as DESSA and determines the location of the first address belonging to the data being programmed or moved. (Enter the first address followed by CR.)

The following key functions are valid:

The M (MOVE) key ensures that the SORSA . . . SOREA data block is stored or relocated (provided the EPROM programmer is connected and prepared for programming – more about this later) into the destination block DESSA . . . DESEA. For reasons involving the V key, the two blocks may not overlap. The three address pointers must be set according to the parameters indicated in figure 1. At the end of the program 'DATA MOVED' appears on the screen/is printed.

The F (FF check) key enables the operator to check whether locations DESSA . . . DESSA + n – 1 contain FF (n represents the number of memory locations in the data block being programmed). If so, data may be stored in that particular range. The address and contents of any memory locations that do not contain FF are printed. Once all 'n' locations have been run through, 'DATA COMPARED' appears.


```
F800: A0 00 B1 10 85 12 29 0F 85 13 A5 12 4A 4A 4A 4A
F810: 85 14 4A 90 1E A9 01 85 15 A2 04 A5 13 D0 0D FB
F820: F0 15 CA 10 F8 A2 19 A5 12 D0 05 FB F0 0F CA 10
F830: F8 50 2A A9 00 F0 ED A5 12 C9 A2 F0 DE A2 01 86
F840: 20 86 16 A2 08 20 12 FC 4C 81 FB A2 02 86 16 A2
F850: 07 86 21 A2 30 86 22 20 FF FB 4C 0B F9 A2 0F E4
F860: 13 D0 43 E8 86 16 20 ED FB A5 15 F0 22 20 F2 FA
F870: 28 24 03 A0 01 B1 10 20 DB FB 20 F2 FA 29 2C 59
F880: 03 18 A5 10 65 16 85 10 A5 11 69 00 85 11 60 20
F890: F2 FA 28 24 03 A0 01 B1 10 20 DB FB 20 F2 FA 2C
F8A0: 58 29 03 4C 81 FB A2 05 E4 13 D0 2F A2 02 86 16
F8B0: 20 ED FB A5 15 F0 15 20 F2 FA 24 03 A0 01 B1 10
F8C0: 20 DB FB 20 F2 FA 2C 58 03 4C 81 FB 20 F2 FA 24
F8D0: 03 A0 01 B1 10 20 DB FB 4C 81 FB A2 09 E4 13 D0
F8E0: 5A A5 15 F0 1F A2 05 86 16 20 ED FB 20 F2 FA 24
F8F0: 03 A0 02 B1 10 20 DB FB 88 D0 FB 20 F2 FA 2C 59
F900: 03 4C 81 FB A2 02 86 16 20 ED FB 20 F2 FA 23 24
F910: 03 A0 01 B1 10 20 DB FB 4C 81 FB A2 0B E4 13 D0
F920: 35 A2 03 86 16 20 ED FB A5 15 F0 18 20 F2 FA 24
F930: 03 A0 02 B1 10 20 DB FB 88 D0 FB 20 F2 FA 2C 58
F940: 03 4C 81 FB 20 F2 FA 24 03 A0 02 B1 10 20 DB FB
F950: 88 D0 FB 4C 81 FB A2 06 E4 13 D0 32 A2 02 86 16
F960: 20 FF FB A2 09 E4 14 F0 10 A2 0B E4 14 F0 0A A5
F970: 15 F0 03 4C B7 F8 4C CC F8 20 F2 FA 24 03 A0 01
F980: B1 10 20 DB FB 20 F2 FA 2C 59 03 4C 81 FB A2 0E
F990: E4 13 D0 1A A2 03 86 16 20 FF FB A2 0B E4 14 F0
F9A0: 0A A5 15 F0 03 4C 2C F9 4C 44 F9 4C EC F8 A2 00
F9B0: E4 13 D0 7A A5 12 C9 00 F0 26 C9 40 F0 22 C9 60
F9C0: F0 1E C9 20 F0 10 29 1F C9 10 F0 1E A2 02 86 16
F9D0: 20 10 FC 4C DB F9 A2 03 86 16 20 10 FC 4C 44 F9
F9E0: A2 01 86 16 20 10 FC 4C 81 FB A2 02 86 16 20 10
F9F0: FC A0 01 B1 10 10 17 20 13 FB 49 FF 85 19 38 A5
FA00: 12 E5 19 85 17 A5 13 E9 00 85 18 4C 1C FA 20 13
FA10: FB 38 65 17 85 17 A9 00 65 18 85 18 20 F2 FA 24
FA20: 03 A2 01 B5 17 20 DB FB CA 10 F8 4C 81 FB A2 04
FA30: E4 13 D0 11 A2 02 86 16 20 1F FC A5 15 F0 03 4C
FA40: B7 F8 4C CC F8 A2 0C E4 13 D0 21 A2 03 86 16 20
FA50: 1F FC A5 12 C9 6C F0 0A C9 8C F0 03 4C 44 F9 4C
FA60: 2C F9 20 F2 FA 28 24 03 A0 02 B1 10 20 DB FB B8
FA70: D0 F8 20 F2 FA 29 03 4C 81 FB A2 0A E4 13 D0 1C
FA80: A2 01 86 16 A6 14 E0 07 90 06 20 30 FC 4C 81 FB
FA90: 20 30 FC 20 F2 FA 20 41 03 4C 81 FB A2 01 86 16
FAA0: 20 1F FC 4C 81 FB 20 DB FB CA 00 FA 60 20 DB FB
FAB0: 4C DB FB 20 CF FB A5 11 20 DB FB A5 10 20 AD FA
FAC0: A0 00 A2 0F B1 10 20 AD FA CA C4 CA C8 C4 16 D0
FAD0: F3 20 A6 FA 60 20 B3 FA A2 03 A0 05 A9 00 06 22
FAE0: 26 21 2A 88 D0 F8 09 40 20 92 FB CA 00 EC 20 DB
FAF0: FB 60 68 85 23 68 85 24 E6 23 D0 02 E6 24 A0 00
FB00: B1 23 C9 03 F0 06 20 D2 FB 4C FB FA A5 24 4B A5
FB10: 23 48 60 A6 10 A4 11 E8 D0 01 C8 86 17 84 18 60
FB20: 7C 0B 26 09 9B 61 1B 9B 82 BB E4 06 02 02 60 86
FB30: DC 93 64 93 9B 61 21 4B 8B D8 E4 E4 30 30 46 86
FB40: 14 14 54 13 95 15 95 15 00 10 61 10 1C 13 1C 11
FB50: 96 18 E4 52 12 86 26 A6 00 C6 32 E6 32 B4 30 02
FB60: FF 12 53 53 9B 61 1C 1C FE 68 60 60 32 32 32 30
FB70: DC FF 93 FF B4 FF 93 FF A6 A6 A6 A4 FF FF 73 FF
FB80: D8 FE D8 FE E4 FE E4 FE 02 26 70 F0 70 FE E0 FE
FB90: 82 1A 83 99 82 1B 83 99 21 A6 A0 1B 4B 1B 4B 99
FBA0: 20 06 20 46 02 12 02 52 72 42 72 2C B2 D8 B0 48
FBB0: 02 03 07 DB 0F B0 04 14 34 44 54 64 74 84 F4 89
FBC0: 1A 3A 5A 7A 8A FA 0C 1C 3C 5C 7C 9C DC FC 9E 4C
FBD0: E8 FF 4C 34 15 4C AE 12 4C BF 12 4C F3 11 4C B7
FBE0: 13 4C 68 12 4C 13 12 4C 90 12 4C 5F 10 A5 14 4A
FBF0: AA B9 20 FB 85 21 B0 28 FB 85 22 20 05 FA 60 A5
FC00: 14 4A AA B0 30 FB 85 21 B0 38 FB 85 22 4C FB FB
FC10: A6 14 B0 40 FB 85 21 B0 50 FB 85 22 4C FB FB A5
FC20: 14 4A AA B0 60 FB 85 21 B0 68 FB 85 22 4C FB FB
FC30: A6 14 B0 70 FB 85 21 B0 80 FB 85 22 4C FB FB A6
FC40: 14 B0 90 FB 85 21 B0 A0 FB 85 22 4C FB FB A9 7E
FC50: A2 FC B9 7C 1A 8E 7D 1A 20 F2 FA D0 0A 56 41 4C
FC60: 49 46 20 43 4F 4D 4D 41 4E 44 53 3A 20 41 20 44
FC70: 20 48 20 4C 20 50 20 52 20 53 50 0D 0A 03 20 CF
FC80: FB 20 85 FB C9 44 B0 5B 20 F2 FA D0 0A 44 49 53
FC90: 41 53 53 45 4D 42 4C 45 3A 20 03 20 B7 FB 20 B8
FCA0: FB 30 FB 20 C5 FD 90 B6 A0 63 7A AE 04 7A B5 7D
FCB0: B6 11 20 F2 FA D0 0A 4C 2C 50 2C 53 50 20 3F 03
FCC0: 4C 7E FC C9 50 90 20 A9 0F 85 25 85 26 38 A0 65
FCD0: 1A E5 7D A0 66 1A E5 11 90 A7 20 00 FB A5 25 F0
FCE0: EC C6 26 B0 E8 F0 9A C9 4C D0 06 A9 00 85 25 F0
FCF0: DC C9 20 D0 D8 A9 07 85 25 85 26 D0 D0 C9 48 B0
FD00: A9 07 85 27 20 F2 FA D0 04 48 45 58 20 44 53 53
FD10: 4D 50 3A 20 03 20 E7 FB 20 D6 FB 70 D6 4C 7E FC
FD20: 4C A5 F9 20 C5 FD 90 F5 20 CF FB 20 CF FB A2 06
FD30: 20 A6 FA A0 00 98 20 E7 FB A2 02 20 A6 FA C8 C0
FD40: 10 B0 F2 A9 65 1A 85 FA A9 64 1A 85 FB 20 CF FB
FD50: 20 CF FB A2 10 B6 26 A5 FB 20 04 FB A5 FA 20 DB
FD60: FB 20 F2 FA 3A 20 03 A0 65 1A 38 E5 FA A9 66 1A
FD70: E5 FB B0 03 4C 7E FC A0 00 B1 FA A6 27 F0 0F 20
FD80: D8 FB 20 DB FB 20 E4 FB C6 26 D0 DB F0 C2 C9 20
FD90: 90 0F C9 7F B0 0B 20 D2 FB A2 01 20 A6 FA 4C B2
FDA0: F0 A2 02 B0 F6 C9 41 D0 2A A9 00 85 27 20 F2 FA
FDB0: D0 A4 53 43 41 4E 20 44 55 4D 50 3A 20 03 4C
FDC0: 15 F9 4C 7E FC A0 65 1A 38 E0 63 1A A0 66 7A E0
FDD0: 64 7A 60 C9 52 D0 EB 4C EA FB A9 0F A0 FF 85 7A
FDE0: 1A 8C 7B 1A 20 F2 FA D0 0A 45 50 52 4F 46 20 50
FDF0: 52 4F 47 52 41 4D 40 69 4E 47 20 55 54 49 4C 69
FE00: 54 49 45 53 D0 0A 56 41 4C 49 44 20 43 4F 4D 4D
FE10: 41 4E 44 53 3A 20 50 2E 4D 2C 42 2C 56 2C 46 2C
FE20: 52 03 20 CF FB 20 95 FB C9 50 00 7A 20 F2 FA 09
FE30: 0A 46 49 52 53 54 2C 4C 41 53 54 20 53 4F 55 52
FE40: 43 45 20 41 44 44 52 45 53 53 3A 20 20 03 20 E1
FE50: FB 20 8E FB 30 96 20 C5 FB 90 01 A0 63 7A AC 64
FE60: 7A 85 E2 84 E3 A0 65 7A AC 66 7A 85 E4 84 E5 20
FE70: 4C FA D0 0A 46 49 52 53 54 20 44 45 53 54 49 4E
FE80: 41 54 49 4F 4E 20 47 44 44 52 45 53 53 3A 20 20
FE90: 03 20 E1 FB 20 A2 13 30 06 A0 65 7A AC 66 7A 85
FEA0: E8 84 E9 4C 22 FE C9 40 00 2B 20 03 7E 20 8C FF
FEB0: A0 00 B1 E6 91 FA 20 13 12 A9 01 85 F6 20 A3 FF
FEC0: B0 EE 20 F2 FA D0 0A 44 41 54 41 20 48 4F 56 45
FED0: 44 03 4C 22 FE C9 42 D0 03 4C EA FB C9 56 D0 38
FEE0: A9 00 85 28 20 03 7E 20 8C FF A0 00 A5 28 D0 35
FEF0: B1 E6 01 FA F0 03 20 F8 11 20 13 12 A9 01 85 F6
FF00: 20 A3 FF B0 E5 20 F2 FA D0 0A 44 41 54 41 20 43
FF10: 4F 4D 50 41 52 45 44 03 4C 22 FE C9 46 D0 D0 A9
FF20: 01 85 28 B0 BF B1 FA C9 FF 4C FA FB C9 52 D0 59
FF30: 20 03 7E 20 8C FF 20 95 FF 20 5C 7E C0 03 D0 29
FF40: B8 B8 B1 E6 38 E5 E2 C8 B1 E6 E5 E3 90 1F B8 A5
FF50: E4 F1 E6 C8 A5 E5 F1 E6 90 13 18 B8 B1 E6 65 EA
FF60: 91 E6 91 FA C8 B1 E6 65 E8 91 E6 91 FA A4 F6 20
FF70: 13 12 88 D0 FA 20 A3 FF B0 BF 20 F2 FA D0 0A 52
FF80: 45 4C 4F 41 54 45 44 03 4C 22 FE A5 E8 AA E9
FF90: B5 FA 84 FB 60 38 A5 E8 E5 E2 B5 EA A5 E9 E5 E3
FFA0: B5 EB 60 18 A5 E6 65 F6 B5 E6 A5 E7 69 00 B5 E7
FFB0: B0 0A 38 A5 E4 E5 E6 A5 E5 E5 E7 60 18 90 FC 68
FFC0: 68 68 20 CF FB A5 E3 20 D8 FB A5 E2 20 D8 FB 20
FFD0: F2 FA 3C 3D 47 44 3D 3C 03 A5 E5 20 D8 FB A5 E4
FFE0: 20 D8 FB 20 F2 FA 20 54 4F 20 3E 3D 03 A5 E9 20
FFF0: D8 FB A5 E8 20 D8 FB 4C 22 FE 2F 1F 70 1C 32 1F
```


Table 2. Hex dump of the cruncher.

The R (RELOCATE) key. All the absolute addresses *within* the data block SORSA . . . SOREA are adapted to the new situation brought about by moving or programming a data block. The new address is determined by the contents of DIF (see figure 1). At the end of the procedure, 'RELOCATED' is printed. The R key function is not needed while relocatable software is being stored (without any internal JMPs and subroutines) or if the contents of one EPROM are being loaded into another. In order to copy RAM data into EPROM, depress R, then M. But to store EPROM data into RAM, depress M followed by R.

The V (VERIFY) key. This compares the original data block and its relocated version, byte by byte. Whenever an error crops up, the offending location is printed along with its address and contents. The operation signs off with the 'DATA COMPARED' message.

The B (BACK) key introduces a return to PM whenever the computer is ready or the operator wishes to disassemble a relocated/programmed data block (to verify the R key routine).

The ST key (ST/NMI on the main keyboard) allows a return to take place from PM to EPRUTL, like a warm start entry. Then 'XXXX < = AD = < YYYY TO > = ZZZZ' appears, where XXXX stands for the FIRST SOURCE ADDRESS, YYYY stands for the LAST SOURCE ADDRESS and ZZZZ stands for the FIRST DESTINATION ADDRESS.

By the way, ST may also be operated during EPRUTL to print the three address parameters and their interim status during an operation. This is extremely useful, as sometimes the parameters need to be temporarily altered. At the same time, the operator is reminded of what was entered three 'screenfuls' before.

How to prevent programs from going 'off the rails'

1. The EPROM programmer must be connected to the bus board. The card is addressed in the normal manner during programming. This means that a 'FIRST DESTINATION ADDRESS' ($2000 or higher) must be entered for reasons described in Book 3. But this does not imply that any EPROM data located below $2000 in the memory map, such as the main board monitor and the TM and PM software, is excluded. Details are provided in point 3.
2. Using the S3 . . . S6 switches, a 4K address block must be selected that does not coincide with any existing data blocks. Otherwise double addressing occurs. If necessary, remove one or two memory cards from the bus board for the time being. Remember that the first two 4K blocks are also out of bounds (see point 1).
3. The FIRST DESTINATION ADDRESS entered just before the start of the program must be located within the selected 4K block (see point 2). This address does not necessarily have to be the ultimate first address (it may be modified later). Right now we intend to load data into the EPROM on the programmer, byte by byte, with the aid of the M key. But take heed! If any absolute addresses need to be altered, start by entering the *real* FIRST DESTINATION ADDRESS using the P key. (Then depress R and P again, followed by the first address of the EPROM programmer.) Finally, operate M.
4. S2 on the EPROM programmer is not switched 'on' until just before the actual programming sequence (with the M key). During programming LED D9 lights and remains lit for the entire process. (About 20 bytes are loaded per second, so it takes quite a while). S2 should be switched off as soon as D9 has gone out and 'DATA MOVED' appears on the screen!
5. 2716 and 2732 EPROMs have one thing in common: they do not enjoy being exposed to the full brunt of the 25 V programming voltage without having the comforting protection of the 5 V supply voltage. The circuit in figure 2 is added to the EPROM programmer hardware 'to cushion the blow'.
6. To find out whether a 2716 IC is truly empty, access a 4K block on the EPROM programmer; select a FIRST DESTINATION ADDRESS that either corresponds to the first address in the range or to one 2048 locations further on, and enter any 2K data block. Now depress the F key.
7. Whenever EPROM software needs to be duplicated, store the 'master' version on a RAM/EPROM card, (unless it is a system EPROM). Insert the (presumably) empty EPROM on the programmer board. Then follow the instructions given in points 3 and 4. After a short while the data 'transfusion' should be complete.
8. Loading EPROM software into RAM is no problem and may come in handy whenever system programs are to be stored on cassette or the contents of an EPROM are to be changed. First copy the data (using the M key) and then relocate it (with the R key), if necessary. The V key allows the operator to check which locations have been altered as a result of the R key routine.
9. When using the R key, watch out for look-up tables and 'strings'! Data such as '20 41 54' is ambiguous, for it may either be the ASCII code for 'uAT', or stand for JSR-$5441! If 54 constitutes an ADH within the data block being programmed ($2000 . . . $5FFF on the dynamic RAM card) the chances are, operating the R key will cause the 54 to be deleted. That is why it is a good idea to check the location of such tables beforehand, and make sure they remain intact after R is depressed (before M is operated). The disassembler is of great help in these matters.
10. A special program, as described in the January article on EPROM software, would be needed to store data in the 'step' mode using the original monitor routine. Fortunately, this is no longer necessary, thanks to the PM routine. Just enter the EPROM location to be programmed (the EPROM programmer version! see point 3), depress the space bar, enter the data and press the '.' key. Make sure the EPROM programmer is ready for programming, as indicated in point 4.

Although very few keys are needed to program EPROMs, operators will discover that they offer a surprisingly versatile repertoire.

We are informed that a suitably programmed 2716 will be available from Technomatic Ltd, London.

2

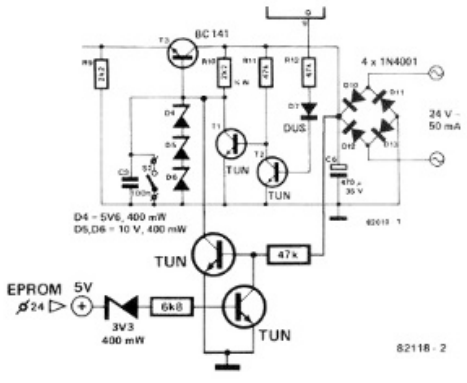


Figure 2. This circuit is added to the EPROM programmer to prevent the EPROM being loaded from becoming a fried chip if S2 is inadvertently switched on before the EPROM programmer power supply is connected up. A common mains switch does not provide a 100% guarantee!