Harald Krake

Ein außergewöhnliches Speichertestprogramm

Die Ursachen des von vielen Anwendern so gefürchteten „Hangup" liegen nicht immer in einer fehlerhaften Software. Oft spielen mangelnde Zuverlässigkeit der Hardware, hervorgerufen durch ein schlechtes System-Design oder auch durch sogenannte „Dreckeffekte", eine große Rolle. Das Verhalten des Speichers (RAM) ist ein guter Indikator für die Beurteilung der Zuverlässigkeit eines Systems. Mit dem hier beschriebenen Programm lassen sich auch noch sehr kleine Fehlerwahrscheinlichkeiten nachweisen.

Speichertestprogramme gibt es inzwischen fast so viele wie Computersysteme. Doch untersuchen die meisten dieser Programme den Speicher nur nach statischen, d. h. logischen Kriterien. Damit lassen sich allenfalls grobe Hardware-Fehler wie defekte Chips oder fehlerhaftes Schaltungsdesign aufspüren. Das Gros der hardwarebedingten Systemabstürze geht jedoch auf das Konto statistischer, zufälliger Fehler, die plötzlich auftreten können und sich dann auch unter festem Vorsatz nicht mehr reproduzieren lassen.
Die Ursachen können in einer schlechten Stromversorgung, aber auch in den Speicherchips selbst liegen. Die Palette der Möglichkeiten ist hier so groß, daß bei der Fehlersuche eigentlich nur noch der Computer selbst helfen kann. So kann es z. B. sein, daß bestimmte Datenmuster mit hoher Wahrscheinlichkeit Fehler erzeugen und andere, nur geringfügig abweichende, auch hochgradig unzuverlässige Speicher als vollkommen „gesund" erscheinen lassen. Dieser Sachverhalt zeigt schon, daß es den Supertest gar nicht geben kann, und das empfindlichste Verfahren von Fall zu Fall variiert werden muß. Das hier beschriebene Programm trägt dieser Forderung durch die Möglichkeit mehrerer Testvarianten Rechnung, so daß man davon ausgehen kann, daß ein System, das auf keinen der insgesamt sieben verschiedenen Tests anspricht, zumindest unter Vorbehalt als „sicher" bezeichnet werden darf. In der Praxis haben sich die hier verwendeten Verfahren bereits durch mehrere Versionen hindurch bewährt, weshalb diese auch im folgenden etwas näher beschrieben werden sollen.

Der Buffer- oder Chip-Test

Dies ist ein sehr „weicher" Test, der vor allem dazu dient, grobe Hardwarefehler zu lokalisieren. Mit ihm lassen sich defekte Chips und Fehler in der Daten- bzw. Adreßpufferung ermitteln. Der Datenfluß ist hierbei bewußt in hohem Maße redundant gehalten, damit ein eventueller Fehler leicht eingekreist werden kann. Man gewinnt mit dem B-Test in erster Linie nur Aussagen über das logische Verhalten eines Speichers.

Der Peak-Test

Hierbei werden schon recht hohe Anforderungen an den Speicher gestellt. Es wird dazu eine logische „1" durch ein bestimmtes Datenmuster hindurchgeschoben und dieser Vorgang in Nicht-Zweierpotenzen (d. h. in diesem Fall modulo 9) mehrmals wiederholt. Das Hindurchschieben (shift) geschieht dabei bildlich sowohl in der X- und der Y-Richtung der Speichermatrix. Dieser Test eignet sich hervorragend zur Erfassung von statistischen Fehlern, die nicht selten auch durch einfaches Übersprechen zwischen den einzelnen Bitzellen entstehen können. Besonders bewährt hat sich dieses Verfahren bei byte- bzw. nibbleorganisierten Speichern (z. B. 2114 oder 6116, etc.). Vorsicht ist allerdings bei der Auswertung des später noch beschriebenen Fehlerprotokolls geboten. Es läßt sich zwar der „fehlerhafte" Chip genau bestimmen, doch kann der Fehler (z. B. bei schlecht geblockten Versorgungsleitungen) in einem ganz anderen Chip entstanden sein. Außerdem sollte man genauer untersuchen, ob der Fehler mehr statistischer oder logischer Art ist (s. Auswertung).

Der Valley-Test

Das bei diesem Test verwendete Verfahren ist mit dem Peak-Test identisch. Es wird lediglich das Datenmuster komplementiert und eine „0" anstatt einer „1" geshiftet. Es hat sich oft gezeigt, daß ein Speicher auf den Peak-Test anspricht und den Valley-Test ohne Fehler passiert bzw. umgekehrt, obwohl die Verfahren prinzipiell identisch sind. Dies ist ein guter Beweis dafür, daß man bei der Fehlersuche nichts unversucht lassen soll!

Der Worst-Case-Test

Auf diesen Test reagieren u. a. bitorganisierte Speicher (z. B. 2102, 4116 oder 4164), aber auch andere müssen sich hier oft geschlagen geben. Er gehört zu den härtesten Tests, deshalb auch sein Name. Das Verfahren ist im Grunde recht einfach: Ein Datenbyte (hier 01010101 sowie 10101010) wird durch sein Komplement hindurchgeschoben. Dabei werden immer ganze Blöcke kopiert (mit dem Z80-LDIR-Befehl) und dem Fehlerteufel auf diese Weise etwas nachgeholfen. Die erhaltenen Ergebnisse haben daher in jedem Fall nur statistischen Charakter und sind eher ein qualitatives Maß für die Zuverlässigkeit. Der Worstcase-Test läßt sich wieder in zwei Komplemente aufteilen, den L- und den H-Test.

Der Refresh-Test

Bei dynamischen Speichern treten oft Probleme mit dem Refresh auf. Dieser Test ähnelt dem Peak/Valley-Test, ver-

```
0100   C3 E8 07 C3   00 00 C3 00   00 C3 00 00   C3 00 00 C3  Ch.C..C..C..C..C
0110   00 00 18 50   00 01 D2 0F   00 00 00 00   FF 20 20 20  ...P..R.....
0120   20 20 20 20   20 20 20 20   20 20 20 20   20 20 20 20
0130   20 FF 00 00   00 00 00 00   00 00 00 00   00 00 00 00   ................
0140   00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00  ................
0150   00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00  ................
0160   00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00  ................
0170   00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00  ................
0180   00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00  ................
0190   00 00 00 00   00 00 00 00   00 00 00 00   00 00 00 00  ................
01A0   3E 0D 18 56   82 3E 0A 18   51 82 21 D1   0E 0E 49 1E  >..V.>..Q.!Q..I.
01B0   D5 F5 1A F5   CD 03 01 F1   13 CB 7F 28   F5 F1 D1 C9  Uu.uM..q.K.(uqQI
01C0   E3 F5 7E CB   BF CD 03 01   7E 23 CB 7F   28 F4 F1 E3  cu~K?M..~#K.(tqc
01D0   C9 F5 D5 CD   06 01 CB BF   FE 08 20 10   1B 1A FE FF  IuUM..K?~. ...~.
01E0   13 28 F0 1B   CD C0 01 08   20 88 18 E7   FE 7F 30 E3  .(p.M@.. ..g~.0c
01F0   FE 20 30 04   FE 0D 20 DB   F5 CD 03 01   F1 FE 0D 28  ~ 0.~. [uM..q~.(
0200   0D C5 4F 1A   FE FF 79 C1   28 C9 12 13   18 C5 1B 1A  .EO.~.yA(I...E..
0210   CB FF 12 3E   0A CD 03 01   D1 F1 C9 11   1D 01 18 B1  K..>.M..QqI....1
0220   CD B0 01 CD   C0 01 20 28   59 2F 4E 29   20 3F A0 CD  M0.M@. (Y/N) ? M
0230   06 01 F5 CD   03 01 CD C0   01 0D 8A F1   FE 4E 28 07  ..uM..M@...q~N(.
0240   FE 6E 28 03   3E FF C9 AF   C9 21 00 00   D5 CD D1 01  ~n(.>.I/I!..UMQ.
0250   1A FE FF 28   26 CB BF FE   30 13 38 F4   1B FE 41 30  .~.(&K?~0.8t.~A0
0260   04 D6 30 18   08 D6 37 FE   10 38 02 D6   20 29 29 29  .V0..V7~.8.V )))
0270   29 85 6F 1A   CB 7F 20 03   13 18 D5 D1   C9 11 1D 01  ).o.K. ...UQI...
0280   18 C7 F5 1F   1F 1F 1F CD   8B 02 F1 F5   E6 0F F6 30  .Gu....M..quf.v0
0290   FE 3A 38 02   C6 07 CD 03   01 F1 C9 F5   7C CD 82 02  ~:8.F.M..qIu|M..
02A0   7D CD 82 02   F1 C9 CD 0F   01 CD C0 01   2A 2A 2A 20  }M..qIM..M@.***
02B0   53 59 53 54   45 4D 2D 43   52 41 53 48   20 2A 2A AA  SYSTEM-CRASH ***
02C0   76 0D 0A 4F   6B 61 F9 42   75 66 66 65   72 2F 43 68  v..OkayBuffer/Ch
02D0   69 70 2D 54   65 73 F4 50   65 61 6B 2D   54 65 73 74  ip-TestPeak-Test
02E0   20 20 20 20   20 20 A0 56   61 6C 6C 65   79 2D 54 65         Valley-Te
02F0   73 74 20 20   20 20 A0 53   70 69 6B 65   2D 54 65 73  st      Spike-Tes
0300   74 20 20 20   20 20 A0 57   6F 72 73 74   63 61 73 65  t       Worstcase
0310   2D 52 2F 57   20 20 A0 48   2D 57 6F 72   73 74 63 61  -R/W    H-Worstca
0320   73 65 20 20   20 20 A0 4C   2D 57 6F 72   73 74 63 61  se      L-Worstca
0330   73 65 20 20   20 20 A0 52   65 66 72 65   73 68 2D 54  se      Refresh-T
0340   65 73 74 20   20 20 A0 52   65 66 72 65   73 68 2D 44  est     Refresh-D
0350   65 6C 61 79   20 20 A0 4D   31 2D 54 65   73 74 20 20  elay    M1-Test
0360   20 20 20 20   20 20 A0 CD   0F 01 CD C0   01 4D 65 6D           M..M@.Mem
0370   6F 72 79 2D   54 65 73 74   65 72 20 52   41 4D 42 55  ory-Tester RAMBU
0380   47 20 56 65   72 73 69 6F   6E 20 32 2E   38 34 0D 0A  G Version 2.84..
0390   28 43 29 31   39 38 32 2C   48 2E 4B 72   61 6B 65 2C  (C)1982,H.Krake,
03A0   48 6F 66 6C   65 68 65 6E   20 39 2C 37   37 34 20 54  Hoflehen 9,774 T
03B0   72 69 62 65   72 67 0D 0A   0A 8A ED 5B   14 01 21 00  riberg....m[..!.
03C0   00 7E 2F 77   BE 2F 77 23   20 F7 2B 22   18 01 ED 5B  .~/w>/w# w+"..m[
03D0   16 01 21 FF   FF 7E 2F 77   BE 2F 77 2B   20 F7 23 22  ..!..~/w>/w+ w#"
03E0   1A 01 E5 CD   C0 01 52 61   6E 64 6F 6D   20 61 63 63  ..eM@.Random acc
03F0   65 73 73 20   6D 65 6D 6F   72 79 3A A0   2A 18 01 CD  ess memory: *..M
0400   9B 02 CD C0   01 AD E1 CD   9B 02 CD C0   01 0D 0A 52  ..M@.-aM..M@...R
0410   65 73 65 72   76 65 64 20   66 6F 72 20   52 41 4D 42  eserved for RAMB
0420   55 47 20 3A   A0 2A 14 01   CD 9B 02 CD   C0 01 AD 2A  UG : *..M..M@.-*
0430   16 01 CD 9B   02 CD C0 01   0D 0A 0A 53   74 61 72 74  ..M..M@....Start
0440   69 6E 67 20   74 65 73 74   20 61 74 20   28 68 65 78  ing test at (hex
0450   29 BA CD 7D   02 22 32 01   E5 CD C0 01   53 74 6F 70  ):M}."2.eM@.Stop
0460   20 74 65 73   74 69 6E 67   20 61 74 20   20 28 68 65   testing at  (he
0470   78 29 BA CD   7D 02 22 34   01 D1 AF ED   52 23 22 36  x):M}."4.Q/mR#"6
0480   01 E5 CD C0   01 2D 2D 3E   4C 65 6E 67   74 68 20 6F  .eM@.-->Length o
0490   66 20 52 41   4D 20 28 68   65 78 29 BA   CD 9B 02 E5  f RAM (hex):M..e
04A0   2A 16 01 ED   5B 14 01 ED   52 23 29 29   29 E5 2A 1A  *..m[..mR#)))e*.
04B0   01 ED 5B 18   01 23 ED 52   D1 ED 52 D1   ED 52 30 1E  .m[..#mRQmRQmR0.
04C0   CD C0 01 0D   0A 0A 2A 2A   2A 20 6E 6F   20 77 6F 72  M@....*** no wor
04D0   6B 73 70 61   63 65 20 2A   2A AA E1 C3   35 04 CD C0  kspace ***aC5.M@
04E0   01 0D 0A 42   6C 6F 63 6B   2D 73 69 7A   65 20 20 20  ...Block-size
04F0   20 20 20 20   28 68 65 78   29 BA CD 7D   02 22 38 01       (hex):M}."8.
-
```

Bild 1. Das komplette Hex-Listing von RAMBUG wird ohne Änderung systemunabhängig eingegeben. Es ist allerdings nur auf Z80-, nicht auf 8080-Systemen lauffähig

```
0500   0E 00 EB E1  AF 0C 28 56  ED 52 28 23  30 F7 CD C0   ..kà/.(VmR(#0wM@
0510   01 0D 0A 0A  2A 2A 2A 20  6D 69 73 6D  61 74 63 68   ....*** mismatch
0520   69 6E 67 20  64 61 74 61  20 2A 2A AA  C3 35 04 79   ing data ***C5.y
0530   32 3A 01 CD  C0 01 2D 2D  3E 20 4E 75  6D 62 65 72   2:.M@.--> Number
0540   20 6F 66 20  42 6C 6F 63  6B 73 20 20  BA 47 AF 3C    of Blocks  :G/<
0550   27 10 FC CD  82 02 3A 12  01 D6 03 91  30 20 CD C0   '.|M..:..V..0 M@
0560   01 0D 0A 0A  2A 2A 2A 20  74 6F 6F 20  6D 61 6E 79   ....*** too many
0570   20 62 6C 6F  63 6B 73 20  2A 2A AA C3  35 04 CD C0    blocks ***C5.M@
0580   01 0D 0A 0A  57 68 69 63  68 20 74 65  73 74 73 20   ....Which tests
0590   64 6F 20 79  6F 75 20 77  61 6E 74 20  3F 0D 0A 8A   do you want ?...
05A0   3A 13 01 FE  29 F5 38 09  11 C7 02 CD  20 02 32 3B   :..~)u8..G.M .2;
05B0   01 F1 FE 29  F5 30 0B 11  F7 02 CD 20  02 32 3E 01   .q~)u0..w.M .2>.
05C0   18 12 11 D7  02 CD 20 02  32 3C 01 11  E7 02 CD 20   ...W.M .2<..g.M
05D0   02 32 3D 01  F1 FE 41 F5  38 14 11 17  03 CD 20 02   .2=.q~Au8....M .
05E0   32 40 01 11  27 03 CD 20  02 32 41 01  18 09 11 07   2@..'.M .2A.....
05F0   03 CD 20 02  32 3F 01 11  37 03 CD 20  02 32 42 01   .M .2?..7.M .2B.
0600   F1 FE 29 38  09 11 57 03  CD 20 02 32  43 01 11 C1   q~)8..W.M .2C..A
0610   02 CD 20 02  CA 67 03 C9  52 57 53 4D  52 57 56 50   .M .Jg.IRWSMRWVP
0620   42 4D 52 4C  48 56 50 42  CD 0F 01 CD  C0 01 41 64   BMRLHVPBM..M@.Ad
0630   64 72 2E A0  06 08 3A 13  01 FE 29 38  05 CD C0 01   dr. ..:..~)8.M@.
0640   20 A0 78 3D  CD 82 02 3A  13 01 FE 29  30 04 0E 02     x=M..:..~)0...
0650   18 0A FE 41  30 04 0E 05  18 02 0E 06  CD C0 01 A0   ..~A0.......M@.
0660   0D 20 F9 10  DD CD C0 01  0D 8A 3A 13  01 06 45 FE   . y.]M@...:...E~
0670   50 30 08 06  3D FE 40 30  02 06 25 CD  C0 01 AD 10   P0..=~@0..%M@.-.
0680   FA 2A 32 01  ED 5B 38 01  3A 3A 01 47  CD C0 01 0D   z*2.m[8.::.GM@..
0690   8A CD 9B 02  CD C0 01 20  A0 C5 06 08  3A 13 01 0E   .M..M@.  E..:...
06A0   07 FE 50 30  08 0E 06 FE  40 30 02 0E  03 CD C0 01   .~P0...~@0...M@.
06B0   AE 0D 20 F9  CD C0 01 A0  10 E2 C1 19  10 CE C9 F5   .. yM@. .bA..NIu
06C0   C5 D5 E5 2A  32 01 2B DD  2B ED 5B 38  01 06 00 19   EUe*2.+]+m[8....
06D0   04 EB E5 DD  E5 E1 ED 52  E1 EB 30 F3  3F ED 52 78   .keleamRak0s?mRx
06E0   3C FD E5 C1  5F 78 06 08  16 05 07 30  37 D5 C5 F5   <)eA_x.....07UEu
06F0   21 21 06 3A  13 01 06 07  FE 50 30 0E  21 1B 06 06   !!.:....~P0.!...
0700   06 FE 40 30  05 21 18 06  06 03 79 BE  28 06 23 10   .~@0.!....y>(.#.
0710   F9 C3 A6 02  78 82 57 CD  0C 01 79 CD  03 01 CD 0C   yC&.x.WM..yM..M.
0720   01 F1 C1 D1  F5 C5 3A 13  01 06 08 FE  50 30 08 06   .qAQuE:....~P0..
0730   07 FE 40 30  02 06 04 78  C1 82 57 F1  CB 27 10 AB   .~@0...xA.WqK'.+
0740   E1 D1 C1 F1  DD 23 C9 50  72 65 73 73  20 61 6E 79   aQAq]#IPress any
0750   20 6B 65 79  20 6F 72 20  72 65 73 65  74 20 73 79    key or reset sy
0760   73 74 65 6D  AE 4C 65 61  76 65 A0 70  61 72 61 6D   stem.Leave param
0770   65 74 65 72  73 20 75 6E  63 68 61 6E  67 65 E4 2A   eters unchanged*
0780   2A 20 4D 31  3A 20 6E 6F  74 20 65 6E  6F 75 67 68   * M1: not enough
0790   20 52 41 4D  2F 62 6C 6F  63 6B 20 2A  2A 0D 8A 00    RAM/block **...
07A0   88 01 7F 00  88 00 04 88  00 08 00 08  00 10 02 0E   ................
07B0   08 04 88 2D  88 00 08 80  00 00 D5 57  3A A8 01 5F   ...-.......UW:(._
07C0   D5 FD E1 E5  DD E1 CD BF  06 D1 C9 F5  D5 16 00 3A   U)aelaM?.QIuU..:
07D0   3A 01 C6 02  5F 3A 12 01  93 DA A6 02  CD 0C 01 CD   :.F._:...Z&.M..M
07E0   73 0C D1 CD  B0 01 F1 C9  31 A8 01 CD  67 03 CD 44   s.QM0.qI1(.Mg.MD
07F0   0B 31 A8 01  CD 28 06 3A  13 01 FE 29  38 37 3A 3B   .1(.M(.:..~)87:;
0800   01 FE FF 20  38 3E 42 32  A8 01 11 C7  02 CD CB 07   .~. 8>B2(..G.MK.
0810   06 08 2A 32  01 ED 5B 36  01 3E 55 77  AE C4 BA 07   ..*2.m[6.>Uw.D:.
0820   CD 09 01 C2  21 0B 3E AA  77 AE C4 BA  07 1B 23 7A   M..B!.>*w.D:..#z
0830   B3 20 E6 10  DD 3A 13 01  FE 29 DA 09  09 3A 3C 01   3 f.]:..~)Z..:<.
0840   FE FF 20 5D  3E 50 32 A8  01 11 D7 02  CD CB 07 CD   ~. ]>P2(..W.MK.M
0850   5A 08 CD 09  01 C2 21 0B  18 47 AF 2A  32 01 77 06   Z.M..B!..G/*2.w.
0860   09 2A 32 01  ED 5B 36 01  7E A7 20 01  37 17 77 08   .*2.m[6.~' .7.w.
0870   23 1B CD 09  01 C0 7A B3  28 03 08 18  F0 2A 32 01   #.M..@z3(...p*2.
0880   ED 5B 36 01  7E A7 20 01  37 4F 08 79  AE C4 BA 07   m[6.~' .7O.y.D:.
0890   23 1B CD 09  01 C0 7A B3  28 04 08 17  18 EB 10 C1   #.M..@z3(....k.A
08A0   C9 3A 13 01  FE 29 DA A6  02 3A 3D 01  FE FF 20 71   I:..~)Z&.:=.~. q
08B0   3E 56 32 A8  01 11 E7 02  CD CB 07 CD  C6 08 CD 09   >V2(..g.MK.MF.M.
08C0   01 C2 21 0B  18 5B 2A 32  01 36 FF 06  09 2A 32 01   .B!..[*2.6...*2.
08D0   ED 5B 36 01  7E FE FF 17  77 08 23 1B  CD 09 01 C0   m[6.~~..w.#.M..@
08E0   7A B3 28 03  08 18 F0 2A  32 01 ED 5B  36 01 7E FE   z3(...p*2.m[6.~~
08F0   FF 4F 08 79  AE C4 BA 07  23 1B CD 09  01 C0 7A B3   .O.y.D:.#.M..@z3
-
```

```
0900   28 04 08 17   18 EB 10 C5   C9 3A 3E 01   FE FF 20 11   (....k.EI:).~. .
0910   3E 53 32 A8   01 11 F7 02   CD CB 07 CD   5A 08 CD C6   >S2(..w.MK.MZ.MF
0920   08 3A 13 01   FE 50 38 41   3A 40 01 FE   FF 20 1D 3E   .:...~P8A:@.~. .>
0930   48 32 A8 01   11 17 03 CD   CB 07 3E 55   32 A9 01 2F   H2(....MK.>U2)./
0940   32 AA 01 CD   9C 09 CD 09   01 C2 21 0B   3A 13 01 FE   2*.M..M..B!.:..~
0950   50 DA A6 02   3A 41 01 FE   FF C2 FC 09   3E 4C 32 A8   PZ&.:A.~.B!.>L2(
0960   01 11 27 03   CD CB 07 18   1F 3A 3F 01   FE FF C2 FC   ..'.MK...:?.~.B!
0970   09 3E 57 32   A8 01 11 07   03 CD CB 07   3E 55 32 A9   .>W2(....MK.>U2)
0980   01 2F 32 AA   01 CD 9C 09   3E AA 32 A9   01 2F 32 AA   ./2*.M..>*2)./2*
0990   01 CD 9C 09   CD 09 01 C2   21 0B 18 60   2A 32 01 ED   .M..M..B!..`*2.m
09A0   5B 36 01 3A   AA 01 77 06   11 3A A9 01   4F 18 01 71   [6.:*.w..:).O..q
09B0   23 1B CD 09   01 C0 7A B3   28 04 10 F3   18 E5 06 10   #.M..@z3(..s.e..
09C0   C5 ED 5B 34   01 D5 E1 2B   ED 4B 36 01   0B ED B8 2A   Em[4.Ua+mK6..m8*
09D0   32 01 3A A9   01 77 C1 CD   09 01 C0 10   E3 ED 5B 36   2.:).wAM..@.cm[6
09E0   01 06 10 3A   A9 01 AE C4   BA 07 23 1B   CD 09 01 C0   ...:)..D:.#.M..@
09F0   7A B3 C8 10   EE 3A AA 01   06 11 18 EA   3A 42 01 FE   z3H.n:*./...j:B.~
0A00   FF 20 6B 3E   52 32 A8 01   11 37 03 CD   CB 07 0E 8E   . k>R2(..7.MK...
0A10   2A 32 01 71   06 08 2A 32   01 ED 5B 36   01 4E CB 09   *2.q..*2.m[6.NK.
0A20   71 23 1B CD   09 01 C2 21   0B 7A B3 20   F1 11 47 03   q#.M..B!.z3 q.G.
0A30   CD CB 07 21   0A 00 11 FF   FF 1B 7A B3   20 FB 2B CD   MK.!......z3 {+M
0A40   09 01 C2 21   0B 7C B5 20   ED 11 37 03   CD CB 07 2A   ..B!.|5 m.7.MK.*
0A50   32 01 ED 5B   36 01 4E 79   AE C4 BA 07   23 1B CD 09   2.m[6.Ny.D:.#.M.
0A60   01 C2 21 0B   7A B3 28 04   CB 09 18 EB   10 A8 3A 13   .B!.z3(.K..k.(:.
0A70   01 FE 29 DA   21 0B 3A 43   01 FE FF C2   21 0B 3E 4D   .~)Z!.:C.~.B!.>M
0A80   32 A8 01 11   57 03 CD CB   07 ED 5B 38   01 21 20 00   2(..W.MK.m[8.! .
0A90   ED 52 38 12   11 7F 07 CD   CB 07 11 47   07 CD B0 01   mR8....MK..G.M0.
0AA0   CD 06 01 C3   E8 07 3A 3A   01 47 2A 32   01 C5 D5 E5   M..Ch.::.G*2.EUe
0AB0   EB 21 9F 07   01 1B 00 ED   B0 E1 C1 09   C5 E5 ED 52   k!.....m0aA.EemR
0AC0   3E C9 12 13   2B 7C B5 20   F7 E1 D1 C1   10 DF D9 01   >I..+|5 waQA._Y.
0AD0   00 04 D9 ED   5B 32 01 CD   09 01 20 45   3A 3A 01 47   ..Ym[2.M.. E::.G
0AE0   D5 C5 21 F1   0A E5 D5 01   00 00 21 80   00 3E FF 08   UE!q.eU...!..>..
0AF0   C9 30 24 A7   20 21 78 FE   01 20 1C 7D   FE 7F 20 17   I0$' !x~. .)~. .
0B00   79 FE 08 20   12 C1 D1 2A   38 01 19 EB   10 D2 D9 0B   y~. .AQ*8..k.RY.
0B10   78 B1 D9 20   BE 18 0A C1   E1 3E FF CD   BA 07 EB 18   x1Y >..Aa>.M:.k.
0B20   E6 11 47 07   CD CB 07 CD   09 01 28 03   CD 06 01 CD   f.G.MK.M..(.M..M
0B30   06 01 11 65   07 CD CB 07   11 6B 07 CD   20 02 CA E8   ...e.MK..k.M .Jh
0B40   07 C3 F1 07   ED 5B 16 01   2A 32 01 AF   ED 52 D0 ED   .Cq.m[..*2./mRPm
0B50   5B 34 01 13   2A 14 01 ED   52 D0 2A 16   01 ED 5B 14   [4..*..mRP*..m[.
0B60   01 AF ED 52   22 AB 01 2A   32 01 ED 4B   18 01 AF ED   ./mR"+.*2.mK../m
0B70   42 EB 2A AB   01 29 29 E5   ED 52 E1 38   14 E5 ED 4B   Bk*+.))emRa8.emK
0B80   34 01 03 2A   1A 01 ED 42   EB E1 ED 52   38 03 C3 A6   4..*..mBkamR8.C&
0B90   02 2A 16 01   AF ED 42 38   22 D5 EB 09   E5 ED 5B 14   .*../mB8"Uk.em[.
0BA0   01 ED 52 E1   D1 38 14 E5   2A 14 01 AF   ED 42 38 0F   .mRaQ8.e*../mB8.
0BB0   D5 ED 5B AB   01 ED 52 D1   38 05 E1 C5   D1 18 16 E1   Um[+.mRQ8.aEQ..a
0BC0   ED 5B 16 01   D5 AF ED 52   DA A6 02 ED   5B AB 01 ED   m[..U/mRZ&.m[+.m
0BD0   52 DA A6 02   D1 CD C0 01   0D 0A 2D 2D   3E 52 65 6C   RZ&.QM@...--->Rel
0BE0   6F 63 61 74   69 6E 67 20   52 41 4D 42   55 47 20 66   ocating RAMBUG f
0BF0   72 6F 6D A0   2A 14 01 CD   9B 02 CD C0   01 20 74 6F   rom *..M..M@. to
0C00   A0 EB CD 9B   02 EB D5 11   C1 02 CD 20   02 D1 CA E8    kM..kU.A.M .QJh
0C10   07 ED 4B AB   01 E5 D5 ED   B0 AF D1 E1   ED 52 E5 C1   .mK+.eUm0/QamReA
0C20   DD 21 D3 0C   DD 6E 00 DD   23 DD 66 00   DD 23 7C B5   ]!S.]n.]#]f.]#|5
0C30   28 18 2B 19   D5 5E 23 56   EB ED 42 EB   72 7E BA 20   (.+.U^#VkmBkr~: 
0C40   0D 2B 73 7E   BB 20 07 D1   18 DA E1 ED   42 E9 CD C0   .+s~; .Q.ZamBiM@
0C50   01 0D 0A 2A   2A 2A 20 62   61 64 20 6D   65 6D 6F 72   ...*** bad memor
0C60   79 20 2A 2A   2A 0D 8A 11   47 07 CD B0   01 CD 06 01   y ***...G.M0.M..
0C70   C3 E8 07 F5   C5 D5 E5 FD   21 00 00 2A   14 01 01 B0   Ch.uEUe)!..*...0
0C80   00 09 E5 C1   2A 14 01 11   76 0E 19 EB   DD 21 E3 0C   ..eA*...v..k]!c.
0C90   DD 6E 00 DD   23 DD 66 00   DD 23 2B D5   ED 5B 14 01   ]n.]#]f.]#+Um[..
0CA0   19 D1 AF ED   42 28 15 D5   EB AF ED 42   38 12 EB 0A   .Q/mB(.Uk/mB8.k.
0CB0   5F 16 00 FD   19 D1 03 2B   7C B5 20 EB   03 03 18 D0   _..).Q.+|5 k...P
0CC0   D1 ED 4B AD   01 FD E5 E1   AF ED 42 C2   A6 02 E1 D1   QmK-.)ea/mBB&.aQ
0CD0   C1 F1 C9 02   00 05 00 08   00 0B 00 0E   00 11 00 15   AqI.............
0CE0   00 17 00 B6   00 C7 00 D5   00 E6 00 FB   00 17 01 1D   ...6.G.U.f.{.,...
0CF0   01 22 01 25   01 31 01 35   01 38 01 4F   01 7F 01 89   ."."%.1.5.8.O....
-
```

wendet jedoch ein anderes Datenmuster und liegt in der Härte zwischen dem B- und dem W-Test. Er läßt sich deshalb auch gut bei statischen Speichern einsetzen. Zwischen den 8 Testphasen wird der Prozessor für einige Sekunden in einer Warteschleife aufgehalten, bevor der Speicherinhalt neu verglichen wird. Es ist natürlich darauf zu achten, daß RAMBUG nicht (!) auch nur zum Teil in den zu testenden Chips läuft, denn bei jedem Opcode-Zyklus wird wieder mindestens eine Matrix-Reihe aufgefrischt (s. Relocator). In der Praxis hat sich der R-Test auch als guter Bus-Test bewährt, denn der R-Test verwendet als einziger Test einen „rekursiven" Datenfluß, was soviel heißt, daß sich die jeweils folgenden Durchläufe an den im Speicher vorgefundenen Daten orientieren. Man läßt dazu den R-Test in einem nicht adressierten Speicherbereich laufen. Treten hierbei Fehler auf, dann sollte man die Terminierung der Busleitungen überprüfen.

Der M1-Test

Der Z80-Prozessor verwendet für seine Speicher-Schreib/Leseoperationen (memory read/write-cycle) und die Befehlsholphase (M1-cycle) unterschiedlich lange Zugriffszeiten, d. h. die Speicher müssen beim M1-Zyklus etwas schneller sein als bei einer einfachen Lese- bzw. Schreiboperation. Dieser Test läßt den Prozessor in dem zu testenden Speicherbereich bestimmte Testprogramme abarbeiten, die so beschaffen sind, daß Einzelbit-Fehler aufgefangen werden und RAMBUG niemals die Kontrolle verliert. Aus diesem Grund ist der M1-Test auch nicht sehr hart, liefert aber bei Timing-Problemen recht brauchbare Anhaltspunkte. Der beste M1-Test ist natürlich, RAMBUG selbst in dem zu testenden Bereich laufen zu lassen. Dies wird ermöglicht durch zwei Eigenschaften, die dieses Programm von anderen unterscheiden:

Eingebauter Relocator und Selbsttest

Es ist eigentlich nicht einzusehen, warum das, was in der Minirechner-Technik selbstverständlich ist, nicht auch auf

```
0D00   01 98 01 9F  01 A3 01 A8  01 AB 01 69  02 6C 02 BD  .....#.(.+.i.l.=
0D10   02 CD 02 D1  02 E1 02 E5  02 FE 02 01  03 04 03 09  .M.Q.a.e.~......
0D20   03 0C 03 27  03 2A 03 2D  03 31 03 34  03 37 03 54  ...'.*.-.1.4.7.T
0D30   03 57 03 5B  03 75 03 78  03 80 03 84  03 9E 03 A2  .W.[.u.x........"
0D40   03 A6 03 B0  03 B4 03 C2  03 DD 03 E0  03 FC 03 FF  .&.0.4.B.].`.|..
0D50   03 10 04 2E  04 32 04 35  04 55 04 58  04 60 04 7D  .....2.5.U.X.`.}
0D60   04 80 04 A2  04 AA 04 AD  04 B0 04 B9  04 BC 04 BF  ...".*.-.0.9.<.?
0D70   04 C4 04 C7  04 CA 04 CD  04 D0 04 D3  04 DC 04 DF  .D.G.J.M.P.S.\._
0D80   04 E2 04 E5  04 E8 04 EB  04 F0 04 F3  04 F6 04 F9  .b.e.h.k.p.s.v.y
0D90   04 FC 04 FF  04 07 05 0A  05 0D 05 10  05 13 05 16  .|..............
0DA0   05 2A 05 2D  05 38 05 3F  05 46 05 49  05 5E 05 67  .*.-.8.?.F.I.^.g
0DB0   05 6C 05 7D  05 83 05 87  05 8A 05 8E  05 93 05 96  .l.}............
0DC0   05 9E 05 AF  05 B6 05 C5  05 CC 05 F2  05 F5 05 FE  .../.6.E.L.r.u.~
0DD0   05 07 06 13  06 19 06 1D  06 20 06 28  06 BE 06 C8  ......... .(.>.H
0DE0   06 D1 06 D7  06 DB 06 DE  06 E1 06 E5  06 EA 06 ED  .Q.W.[.^.a.e.j.m
0DF0   06 F0 06 F3  06 F6 06 F9  06 00 07 09  07 0C 07 0F  .p.s.v.y........
0E00   07 14 07 18  07 1F 07 22  07 25 07 2C  07 37 07 3C  .......".%.,.7.<
0E10   07 3F 07 48  07 4B 07 4E  07 51 07 54  07 57 07 5D  .?.H.K.N.Q.T.W.]
0E20   07 63 07 67  07 74 07 7F  07 83 07 8F  07 94 07 A3  .c.g.t.........#
0E30   07 A8 07 AB  07 B4 07 B7  07 BA 07 BD  07 C0 07 C3  .(.+.4.7.:.=.@.C
0E40   07 C8 07 CF  07 D3 07 DE  07 E9 07 ED  07 F7 07 FC  .H.O.S.^.i.m.w.|
0E50   07 0B 08 14  08 17 08 1A  08 1D 08 20  08 23 08 2A  ........... .#.*
0E60   08 33 08 36  08 39 08 3E  08 42 08 45  08 48 08 4B  .3.6.9.>.B.E.H.K
0E70   08 4E 08 53  08 56 08 5B  08 60 08 63  08 66 08 6B  .N.S.V.[.`.c.f.k
0E80   08 70 08 75  08 78 08 7B  08 80 08 84  08 87 08 8C  .p.u.x.{........
0E90   08 90 08 93  08 96 08 99  08 9E 08 A2  08 A5 08 AB  ............".%.+
0EA0   08 B4 08 C4  08 CB 08 D1  08 D4 08 D9  08 E0 08 E5  .4.D.K.Q.T.Y.`.e
0EB0   08 E9 08 EE  08 F7 08 FE  08 07 09 0A  09 0D 09 12  .i.n.w.~........
0EC0   09 18 09 1C  09 25 09 28  09 2F 09 32  09 41 09 44  .....%.(./.2.A.D
0ED0   09 4B 09 4E  09 51 09 55  09 5B 09 60  09 63 09 70  .K.N.Q.U.[.`.c.p
0EE0   09 75 09 78  09 7D 09 82  09 85 09 88  09 8C 09 96  .u.x.}..........
0EF0   09 99 09 9C  09 9F 09 A2  09 A5 09 A8  09 AC 09 B3  ........".%.(.,.3
0F00   09 D6 09 D9  09 DE 09 E4  09 09 0A 1D  0A 23 0A 26  .V.Y.^.d.....#.&
0F10   0A 29 0A 2E  0A 31 0A 34  0A 37 0A 3A  0A 3D 0A 40  .)...1.4.7.:.=.@
0F20   0A 43 0A 47  0A 4A 0A 52  0A 56 0A 5C  0A 60 0A 66  .C.G.J.R.V.\.`.f
0F30   0A 69 0A 6D  0A 74 0A 81  0A 85 0A 90  0A 93 0A A0  .i.m.t..........
0F40   0A AA 0A B4  0A C3 0A CA  0A CE 0A D3  0A D7 0A F6  .*.4.C.J.N.S.W.v
0F50   0A F9 0A FC  0A 04 0B 09  0B 0C 0B 10  0B 14 0B 23  .y.|...........#
0F60   0B 50 0B 69  0B 6C 0B 6F  0B 72 0B 7D  0B 86 0B 8F  .P.i.l.o.r.}....
0F70   0B 9F 0B C4  0B CD 0B 00  00 00 00 00  00 00 00 00  ...D.M..........
0F80   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ................
0F90   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ................
0FA0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ................
0FB0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ................
0FC0   00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  ................
0FD0   00 00  ..
-
```

Mikroprozessoren implementiert werden kann. Gemeint sind voll verschiebliche Programme, die ohne Umweg über einen sogenannten Binder (Linker) in jedem Adreßbereich lauffähig sind. Man fügt dazu dem Programm einen sogenannten „Header" an, in dem sämtliche für eine Adreßumrechnung notwendigen Informationen enthalten sind. Das Betriebssystem kann damit das Programm für jede Speicheradresse blitzschnell umschreiben (relokatieren). Auf Mikrocomputern ist dieses Verfahren naturgemäß zwar etwas aufwendiger (die Programme belegen ca. 20...30 % mehr Speicherplatz), aber es ist durchführbar, wie dieses Programm beweist.
Da RAMBUG als Speichertestprogramm natürlich ohne Betriebssystem auskommen muß, ist der Relocator bereits fester Bestandteil. Es wird damit möglich, auch jenen Speicherbereich zu testen, in dem sich RAMBUG gerade selbst befindet. Der Speicher ist für den Anwender vollkommen transparent, denn der Relocator arbeitet automatisch und verhindert damit auch jede Fehlbedienung.
Dieses Verfahren bietet mehrere Vorteile:

1. Es wird verhindert, daß sich RAMBUG selbst zerstört.
2. Jeder Speicherbereich kann getestet werden.
3. Durch Verlagerung in einen anderen Arbeitsbereich können bestimmte Fehlerursachen ausgeschlossen bzw. eingekreist werden, z. B. statistischer oder logischer Fehler; Übersprechen; Simulation eines harten M1-Tests, indem man RAMBUG in den zu testenden Bereich verschiebt usw.
4. Einfache Implementierung.

Neben dem eingebauten Relocator besitzt RAMBUG noch eine Reihe weiterer nützlicher Eigenschaften:
Ein automatischer Selbsttest verhindert einen sonst vielleicht nicht bemerkten „hang-up", denn die Einzeltests dauern oft recht lange und es könnte immerhin sein, daß der Speicher, in dem RAMBUG gerade läuft, selbst nicht sehr zuverlässig ist. Das Programm ist vollkommen systemunabhängig und wird über eine Vektor-Tabelle mit den notwendigen systemabhängigen I/O-Routinen verbunden. Diese Routinen sollten nicht das Betriebssystem verwenden, denn es ist ja in der Regel gerade der Bereich der Systemprogramme von besonderem Interesse. Die I/O-Module können derart in RAMBUG integriert werden, daß sie auch voll-relokatibel sind und entsprechend mitverschoben werden.
Um die Aufteilung des Speicherbereichs braucht sich der Anwender nicht zu kümmern. Sie wird automatisch erkannt und von RAMBUG (*Bild 1*) berücksichtigt.
Die Implementierung vereinfacht sich auf ein Minimum durch die Verwendung eines Generatorprogramms (*Bild 2*). Das in Bild 1 abgedruckte Listing kann dafür direkt ohne Änderung eingegeben werden. Die sonst immer recht mühsame Adreßumrechnung macht der Computer (...wozu hat man ihn denn?).

Die Programmstruktur: interaktiv

RAMBUG ist ein interaktives Programm. In einem Menü wird der Anwender straff geführt, weshalb die einzelnen Bedienungsschritte auch nicht explizit beschrieben werden sollen. Zu der hier veröffentlichten Version ist allerdings noch einiges zu sagen.
Die Bildschirmausgabe ist immer ein kritischer Punkt bei systemunabhängigen Programmen, die interaktiv arbeiten. Damit dieses Programm aber auch auf wirklich allen Systemen mit den unterschiedlichsten Bildschirmformaten läuft, wurde hier ein Weg beschritten, der zwar etwas aufwendig erscheinen mag, dem Anwender aber eine Menge Arbeit ersparen kann.
In der Vektortabelle sind zwei Bytes vorgesehen, in denen das Bildschirmformat eingetragen wird, d. h. Anzahl der Spalten und Zeilen. RAMBUG richtet sich nach diesen Parametern und paßt die Bildschirmausgabe entsprechend an.
Auf diese Weise kommen Besitzer eines 24 × 80-Terminals in den Genuß einer ausführlichen Speichermatrix, und Hobbycomputer-Anwender, die vielleicht nur eine Zeilenbreite von 40 Zeichen zur Verfügung haben, müssen auf RAMBUG nicht verzichten. Für sie erscheint die Matrix in einem komprimierten Format, daß zumindest die wichtigsten Informationen enthält. Für die unterschiedlichen Formate müssen allerdings folgende Einschränkungen beachtet werden:
Bei Systemen mit 80 und mehr Zeichen pro Zeile können alle 7 Tests gefahren werden. Bei Systemen mit 64 bis 79 Zeichen werden der H- und L-Test zu einem gemeinsamen W-Test zusammengefaßt. Für Systeme mit weniger als 64 Zeichen/Zeile mußte leider ein Kompromiß gefunden werden. Der B- und M1-Test werden für solche Anwender leider auf ewig im Verborgenen bleiben. Der P- und V-Test sind zu einem einzigen S

```
5000   21 00 01 11  00 40 AF ED  52 E5 C1 11  00 20 DD 21  !....@/mReA.. ]!
5010   D3 0B DD 19  DD 6E 00 DD  23 DD 66 00  DD 23 7C B5  S.].]n.]#]f.]#|5
5020   28 10 2B 19  D5 5E 23 56  EB ED 42 EB  72 2B 73 D1  (.+.U^#VkmBkr+sQ
5030   18 E2 C9  .bI
-
```

Bild 2. Dieses Generatorprogramm wird vom CP/M-Anwendern nicht benötigt. Die beiden Parameter sind unterstrichen

```
Addr.    07      06      05      04      03      02      01      00
-----------------------------------------------------------------------
4000  ....... ....... ....... ....... ....... ....... ....... .......
4400  ....... ....... ....... ....... ....... ....... ....... .......
4800  .P..... ..VHL.. ....LR. ....... ....... ....... ....... .......
4C00  ....... ....... ....... ....... ....... ....... ....... .......
5000  ......M ......M ......M ......M ......M ......M ......M ......M
5400  ..V.... .P.H... ....... ....... ....... ....... ....... .......
5800  .....R. B...... .....R. ....... ....... ....... ..V.... .......
5C00  .P..LR. ....... ....... BP.H... ....... ...HLR. ....... B.V....
Press any key or reset system.
```

Bild 3. So sieht die von RAMBUG ausgegebene Speicher-Matrix aus. Hier wird ein Speicher mit 1-KBit-Chips getestet. Bei einem „gesunden" Speicher muß die Matrix allerdings leer bleiben

```
          NAME    SYSIO
          TITLE   System-I/O-Routinen fuer RAMBUG V1.XX and V2.XX
          REM     Beispiel: fuer serielle Schnittstelle Intel 8251
          REM     in Verbindung mit DEC-VT52-Terminal.

          ENTRY   OUTCH,INCH,TEST,POS,CLR
          ;       Vektoren fuer den Jumpertable.

;         Systemparameter:

CONOUT    EQU     01        ;Console-Outputport (INTEL 8251A)
CONIN     EQU     01        ;Console-Inputport
CONSTAT   EQU     00        ;Console-Statusport

TRM       EQU     00000001B         ;Transmitter-Ready-Mask
RRM       EQU     00000010B         ;Receiver-Ready-Mask

;         Terminalparameter:

ESC       EQU     1BH       ;ESC-Sequenz identifiziert Steuercodes.
XOFF      EQU     20H       ;X-Offsetkomponente,
YOFF      EQU     20H       ;Y-Offsetkomponente fuer direkte Cursoradressierung.

;-------------------------------------------------------------------------------

          ORG     0FD2H     ;I/O-Routinen werden RAMBUG angehaengt.

OUTCH:    ;Druckt ein ASCII im Akku. Keines der Register (ausser A)
          ;darf veraendert werden. (sonst PUSH & POP einbauen!)

          PUSH    AF                ;ASCII retten
WAIT:     IN      A,CONSTAT         ;Consol-Status in A
          AND     TRM               ;Transmitter ready ?
          JR      Z,WAIT            ;Z=1, falls nicht.
          POP     AF                ;ASCII zurueck
          RES     7,A               ;Parity-Bit abschneiden
          OUT     CONOUT,A          ;Zeichen senden
          RET                       ;return


INCH:     ;Liest ein Zeichen von der Console in den Akku.
          ;Die Routine loopt solange, bis eine Taste gedrueckt wird!
          ;Kein Register (ausser A) darf veraendert werden.

          CALL    TEST              ;Warte bis Taste gedrueckt...
          JR      Z,INCH
          IN      A,CONIN           ;ASCII in A
          RES     7,A               ;Parity abschneiden
          RET                       ;return


TEST:     ;Testet, ob eine Taste gedrueckt wurde.
          ;Return: Z=0, falls ja, Z=1, falls nicht.
          ;Keines der Register (ausser A) darf veraendert werden.

          IN      A,CONSTAT         ;Status in A
          AND     RRM               ;Set/Reset Z-Flag
          RET

POS:      ;Bewegt den Cursor auf die durch DE bestimmte Position auf dem
          ;Bildschirm. D = X-Komponente (Spalte), E = Y-Komp. (Zeile).
          ;Die Nummerierung beginnt jeweils mit 0.
          ;Der Akku darf veraendert werden.
          ;Bei Systemen, die diese Moeglichkeit nicht besitzen, kann die
          ;POS-Funktion durch Ausgabe von "Cursor Home", "Cursor rechts"
          ;und "Line-Feed" simuliert werden.
```

Bild 4. Musterlösung für die I/O-Routinen für RAMBUG. In der Reihenfolge von oben nach unten: Assembler-Source, Hex-Code ab 0FD2 H, Eintragungen im Header ab 0F77 H, Eintragungen in der Vektortabelle

(= Spike)-Test zusammengefaßt, so daß insgesamt nur drei verschiedene Kriterien (S, W, R) verwendet werden können.

Auswertung komfortabel durch Speichermatrix

Während die einzelnen Tests laufen, erscheint auf dem Bildschirm eine Matrix wie in *Bild 3*. Unterhalb der Matrix wird der momentane Test-Status angezeigt (z. B. „Peak-Test"). Je nach Bildschirmbreite (s. o.) teilt sich jede Zeile in 8 Gruppen (pro Bit eine Gruppe) mit je 3, 6 oder 7 Punkten auf. Jede Zeile wird einem bestimmten Speicherbereich zugeordnet. Es ist am zweckmäßigsten, die Blöcke so einzuteilen, daß sie den Adressenbereichen der Speicherchips entsprechen. Auf diese Weise kann von der Matrix die Position eines fehlerhaften Chips direkt abgelesen werden. Jeder Punkt steht stellvertretend für ein mögliches Fehlerkriterium. Stellt RAMBUG einen Fehler fest, so wird an den entsprechenden Koordinaten der korrespondierende Fehlertyp eingetragen (z. B. P für Peak-Error). Der Cursor bleibt immer auf dem zuletzt angezeigten Fehler stehen, so daß bei längeren Tests die Orientierung gewahrt bleibt. Auf diese Weise erhält man einen schnellen Überblick und keine (wie oft noch üblich) mitunter seitenlangen Fehlerlistings. Trotz des hohen Komforts ist bei der Auswertung Vorsicht geboten.

Es sind nicht unbedingt die Chips defekt, für die ein Fehler angezeigt wird.

Beim W-Test z. B. kann der Fehler beim Kopieren aus einem anderen Adressenbereich entstanden sein. Der M1-Test kann nur grob den Block angeben, in dem der Fehler aufgetreten ist. Wenn der Zufall eine große Rolle spielt, dann sollte man auf keinen Fall mit dem Auswechseln von Chips den Speicher kurieren wollen. In der Mehrzahl liegen die Ursachen im Schaltungsdesign oder unzureichend störsicherem Layout. Gerade bei 4-MHz-Systemen können oft Kleinigkeiten ausschlaggebend sein.
In der Praxis haben sich bei bestimmten Fehlern folgende Verfahren bewährt:

- Versorgungsleitungen zusätzlich mit Kondensatoren abblocken (z. B. 10 nF). Gerade bei dynamischen Speichern werden hohe Anforderungen an die Stromversorgung gestellt [2].
- Treten nur P-, V- und W-Fehler auf, dann sind in den meisten Fällen die Puffer verdächtig. Patentrezept: Je zwei Treiberchips im Huckepack übereinanderlöten (keine Angst! Es sollen natürlich zwei gleiche sein.). Das Fanout vergrößert sich, und störende Leitungskapazitäten/Induktivitäten können nicht mehr so stark zur Geltung kommen.
- Oft ist die Masseführung unsauber. Man kann durch geschicktes Anlegen von sogenannten Masseschleifen versuchen, die Störsignale durch ihr eigenes „Komplement" auszubooten. Das ist ohne Oszilloskop jedoch eine Sisyphusarbeit, und man kann damit auch genau das Gegenteil erreichen.
- Treten nur M1-Fehler auf, dann sind entweder die Puffer zu langsam (Huckepackmethode versuchen) oder schnellere Speicherchips notwendig. Letzteres sollte man aber nur als letzten Ausweg probieren.

```
LD      A,ESC
CALL    OUTCH           ;ESC-Char. leitet Steuersequenz ein.
LD      A,'Y'           ;ESC Y bedeutet direkte Cursoradressierung
CALL    OUTCH
LD      A,E             ;Zeilennummer in A
ADD     YOFF            ;Zeilen-Offset
CALL    OUTCH
LD      A,D             ;Spaltennummer in A
ADD     XOFF            ;Spalten-Offset
JP      OUTCH           ;return via OUTCH

CLR:    ;Stellt den Cursor home und loescht den Bildschirm
        ;A darf veraendert werden.

        LD      A,ESC           ;ESC-Steuercode
        CALL    OUTCH
        LD      A,'H'           ;ESC H bedeutet "Cursor Home"
        CALL    OUTCH
        LD      A,ESC
        CALL    OUTCH
        LD      A,'J'           ;ESC J bedeutet "Rest des Schirms loeschen"
        JP      OUTCH           ;return...

        END
```

```
0FD2   F5 DB 00 E6  01 28 FA F1  CB BF D3 01  C9 CD E9 0F  u[.f.(zqK?S.IMi.
0FE2   28 FB DB 01  CB BF C9 DB  00 E6 02 C9  3E 1B CD D2  ((I.K?II.f.I).MR
0FF2   0F 3E 59 CD  D2 0F 7B C6  20 CD D2 0F  7A C6 20 C3  .>YMR.{F MR.zF C
1002   D2 0F 3E 1B  CD D2 0F 3E  48 CD D2 0F  3E 1B CD D2  R.>.MR.>HMR.>.MR
1012   0F 3E 4A C3  D2 0F  .>JCR.
-

0F77   E1 0E F2 0E  F7 0E FD 0E  03 0F 08 0F  0D 0F 12 0F  a.r.w.}.........
0F87   17 0F  ..
-

0100   C3 E8 07 C3  D2 0F C3 DF  0F C3 E9 0F  C3 EE 0F C3  Ch.CR.C_.Ci.Cn.C
0110   04 10 18 50  00 01 18 10   ...P....
-
```

Wichtige Vektoren in RAMBUG

Die Zahlenangaben sind hexadezimal und beziehen sich auf Startadresse 100H.

Adresse	Wert	Beschreibung
100	C3 E8 07	Entrypoint. Um RAMBUG auszuführen, muß das Betriebssystem zu dieser Adresse springen.
103	C3 XX XX	Sprung zur Zeichenausgabe-Routine. Sie sollte kein Bestandteil des Betriebssystems sein, sondern autark arbeiten, damit man sie in RAMBUG integrieren kann. Dasselbe gilt für alle folgenden Systemroutinen. Näheres findet sich in Bild 4.
106	C3 XX XX	Sprung zur Zeicheneingabe-Routine.
109	C3 XX XX	Sprung zur Routine, die prüft, ob eine Taste gedrückt wurde. Diese Routine wird von RAMBUG während eines Tests ständig aufgerufen, um festzustellen, ob der Benutzer den laufenden Test vorzeitig abbrechen möchte. Man kann dazu jede beliebige Taste drücken.
10C	C3 XX XX	Sprung zur Cursoradressierungs-Routine.
10F	C3 XX XX	Sprung zur Clear-Home-Routine.
112	XX	Anzahl der Zeilen pro Bildschirm. Beispiel: Ein Terminal habe das Format 24 Zeilen zu je 80 Zeichen, dann müßte in 112 die Hexzahl 18 stehen.
113	XX	Anzahl der Zeichen pro Zeile. In unserem Beispiel stünde hier 50 hex.
114	XX XX	Startadresse von RAMBUG im Speicher. Sie gibt den Beginn des Maschinencodes an, der bei einer Adressenverschiebung schon zu RAMBUG gehört. In unserem Fall ist dies die Adresse 100 hex, denn die Vektortabelle muß natürlich ebenfalls umgerechnet werden.
116	XX XX	Endadresse von RAMBUG. Wenn einige I/O-Routinen hinzugefügt werden, dann enthält 116 deren Endadresse. Das Beispiel in Bild 4 würde die Endadresse von RAMBUG von 0FD2 nach 1018 verschieben.

- Treten B-Fehler statistisch auf, dann ist mit großer Wahrscheinlichkeit der Bus unsauber, denn eine Karte mit derart miserablem Layout wäre ein Kunstwerk. Es ist z. B. illusorisch, einen Z80A mit 4 MHz auf einem 20er ECB-Bus zu betreiben, ohne die Busleitungen terminiert zu haben. Wie man das macht, ist in [1] beschrieben.

Implementierung mit Programmgenerator

Vom dem Idealismus aus Zeiten des Z80-Texteditors gründlich geheilt, wird hier erstmalig ein Verfahren angewendet, das von größeren Anlagen her schon lange bekannt ist und im kleinen Maßstab auf diesen Anwendungsfall übertragen wurde. Ein Generatorprogramm (Bild 2) übernimmt die Hauptarbeit – die Adreßumrechnung. Das Generatorprogramm ist selbst natürlich vollkommen relokatibel und die notwendigen Parameter werden direkt in den Maschinencode eingetragen. Es ist damit systemunabhängig. Alles was zu tun bleibt, ist in den folgenden Schritten zusammengefaßt:

- Das in Bild 1 abgedruckte Listing (hier aufgelistet ab 100H) direkt und ohne Änderung in einen freien Speicherbereich, gleichgültig ab welcher Adresse, eingeben.
- Das in Bild 2 aufgelistete Generatorprogramm ebenso in einen noch freien Speicherbereich eingeben.
- In das Generatorprogramm (hier aufgelistet ab 5000H) zwei Parameter eintragen:
  5004H: Startadresse, ab der RAMBUG später tatsächlich laufen soll. 500CH: Startadresse, ab der RAMBUG momentan im Speicher steht. Die Startadresse ist jeweils die Adresse des ersten in Bild 1 abgedruckten Bytes (100H) und nicht der Einsprungvektor!
- Das Generatorprogramm starten (hier ab 5000H). In 5032H steht ein RET-Befehl. Dies ist der Ausgangspunkt des Generatorprogramms.
- Das so generierte RAMBUG-Programm steht zwar immer noch an derselben Stelle, die absoluten Adressen sind jetzt jedoch für die gewünschte Zieladresse umgerechnet. RAMBUG muß nur noch an die entsprechende Stelle im Speicher verschoben werden.
- Die systemabhängigen I/O-Routinen ab 0FD2H bzw. der korrespondierenden Adresse implementieren, damit diese bei einer automatischen Adreßumrechnung mit verschoben werden. Dazu müssen noch die Vektoren entsprechend geändert werden und im Header die Absolutadressen der I/O-Routinen ab 0F77H (bzw. entsprechend) eingetragen werden. In der *Tabelle* sind alle wichtigen Vektor- und Headeradressen angegeben. Eine Absolutadresse zeigt immer auf das höherwertige Byte eines Opcodes und wird relativ zur Startadresse von RAMBUG gezählt. In *Bild 4* ist eine Musterlösung als Assemblerlisting sowie den entsprechenden Vektoren und Header-Adressen abgedruckt.
- Auf ein sicheres Medium abspeichern, z. B. Kassette oder Floppy-Disk.

Einfacher geht's nicht mehr!

Wie schon erwähnt, führt RAMBUG in regelmäßigen Abständen einen automatischen Selbsttest durch, um zu überprüfen, ob nicht wichtige Programmteile durch defekte Speicher o. ä. zerstört wurden. Der Selbsttest überprüft jedoch nur die Programmbytes, die keine absoluten Adressen sind (denn diese können sich ja ändern). Ein eventueller Tippfehler bei der Programmeingabe wird, sofern er eine Absolutadresse betrifft, deshalb nicht bemerkt. Wenn RAMBUG die Meldung „System Crash" ausgibt, dann liegt mit absoluter Sicherheit ein Tippfehler im Programm vor.

Literatur

[1] *Krake, H.*: Bus-Terminierung für ECB-Systeme. Vorauss. mc 1983, Heft 1.
[2] *Oettle, F., Reichle, T.*: Dynamische Speicher. mc 1981, Hefte 3 und 4.

Spruch des Monats

„Schon heute bewältigen in den verschiedenen US-Forschungsinstituten und Entwicklungsanstalten 100 000 Computeranlagen einen Arbeitsanfall, der ohne sie, auf manuellem Weg, nur von etwa 400 Milliarden Menschen zu schaffen wäre. Das wären aber hundertmal mehr Menschen, als derzeit auf dem Globus leben."

Dr. Carl Hammer, Forschungsdirektor der Univac, 1974 auf einer Fachtagung in Moskau.