```
f7cf  85 54     sta  $54
f7d1  c8        iny
f7d2  b1 2e     lda  ($2e),y
f7d4  85 55     sta  $55
f7d6  c8        iny
;     und ggf. codieren
f7d7  d0 e1     bne  $f7ba
;     Prüfsumme Datenblock
f7d9  a5 3a     lda  $3a
;     ebenfalls codieren und
;     in Puffer schreiben
f7db  85 53     sta  $53
f7dd  a9 00     lda  #$00
f7df  85 54     sta  $54
;     Leerbytes zum Auffüllen
f7e1  85 55     sta  $55
;     ebenfalls codieren
f7e3  4c d0 f6  jmp  $f6d0
```

Umwandlung von 5 GCR-Bytes in 4 Daten-Bytes

```
;     Pufferzeiger holen
f7e6  a4 34     ldy  $34
;     Byte aus Puffer holen
f7e8  b1 30     lda  ($30),y
f7ea  29 f8     and  #$f8
f7ec  4a        lsr
f7ed  4a        lsr
f7ee  4a        lsr
f7ef  85 56     sta  $56
;     und decodieren;
;     zwischen-
;     speichern für späteres
f7f1  b1 30     lda  ($30),y
f7f3  29 07     and  #$07
f7f5  0a        asl
f7f6  0a        asl
;     zusammenschieben
f7f7  85 57     sta  $57
f7f9  c8        iny
f7fa  d0 06     bne  $f802
f7fc  a5 4e     lda  $4e
f7fe  85 31     sta  $31
f800  a4 4f     ldy  $4f
;     nächstes Byte holen
f802  b1 30     lda  ($30),y
f804  29 c0     and  #$c0
f806  2a        rol
f807  2a        rol
f808  2a        rol
f809  05 57     ora  $57
f80b  85 57     sta  $57
;     codieren und
f80d  b1 30     lda  ($30),y
f80f  29 3e     and  #$3e
f811  4a        lsr
;     zwischenspeichern
f812  85 58     sta  $58
f814  b1 30     lda  ($30),y
f816  29 01     and  #$01
f818  0a        asl
f819  0a        asl
f81a  0a        asl
f81b  0a        asl
f81c  85 59     sta  $59
f81e  c8        iny
f81f  b1 30     lda  ($30),y
f821  29 f0     and  #$f0
f823  4a        lsr
f824  4a        lsr
f825  4a        lsr
f826  4a        lsr
f827  05 59     ora  $59
f829  85 59     sta  $59
f82b  b1 30     lda  ($30),y
f82d  29 0f     and  #$0f
f82f  0a        asl
f830  85 5a     sta  $5a
f832  c8        iny
f833  b1 30     lda  ($30),y
f835  29 80     and  #$80
f837  18        clc
f838  2a        rol
f839  2a        rol
f83a  29 01     and  #$01
f83c  05 5a     ora  $5a
f83e  85 5a     sta  $5a
f840  b1 30     lda  ($30),y
f842  29 7c     and  #$7c
f844  4a        lsr
f845  4a        lsr
f846  85 5b     sta  $5b
f848  b1 30     lda  ($30),y
f84a  29 03     and  #$03
f84c  0a        asl
f84d  0a        asl
f84e  0a        asl
f84f  85 5c     sta  $5c
f851  c8        iny
f852  d0 06     bne  $f85a
f854  a5 4e     lda  $4e
f856  85 31     sta  $31
f858  a4 4f     ldy  $4f
f85a  b1 30     lda  ($30),y
f85c  29 e0     and  #$e0
f85e  2a        rol
f85f  2a        rol
f860  2a        rol
f861  2a        rol
f862  05 5c     ora  $5c
f864  85 5c     sta  $5c
f866  b1 30     lda  ($30),y
f868  29 1f     and  #$1f
f86a  85 5d     sta  $5d
f86c  c8        iny
f86d  84 34     sty  $34
;     Bytes aus
;     Zwischenspeicher
f86f  a6 56     ldx  $56
;     holen, gemäß Tabelle
;     übersetzen und ab $52
;     bis $55
f871  bd a0 f8  lda  $f8a0,x
f874  a6 57     ldx  $57
f876  1d c0 f8  ora  $f8c0,x
;     speichern für Übergabe
;     1. Byte
f879  85 52     sta  $52
f87b  a6 58     ldx  $58
f87d  bd a0 f8  lda  $f8a0,x
f880  a6 59     ldx  $59
f882  1d c0 f8  ora  $f8c0,x
;     2. Byte
f885  85 53     sta  $53
f887  a6 5a     ldx  $5a
f889  bd a0 f8  lda  $f8a0,x
f88c  a6 5b     ldx  $5b
f88e  1d c0 f8  ora  $f8c0,x
;     3. Byte
f891  85 54     sta  $54
f893  a6 5c     ldx  $5c
f895  bd a0 f8  lda  $f8a0,x
f898  a6 5d     ldx  $5d
f89a  1d c0 f8  ora  $f8c0,x
;     4. Byte
f89d  85 55     sta  $55
f89f  60        rts
;     Tabelle dient der
;     Decodierung der GCR-
;     Bytes in die entspr.
;     Hex-Werte.
;     Die $FF-Bytes geben
;     die Positionen der
;     illegalen Bitkombina-
;     tionen an. Sie veran-
;     lassen ggf. einen
;     »24, READ ERROR«.
f8a0  ff ff ff ff ff ff ff ff
f8a8  ff 80 00 10 ff c0 40 50
f8b0  ff ff 20 30 ff f0 60 70
f8b8  ff 90 a0 b0 ff d0 e0 ff
f8c0  ff ff ff ff ff ff ff ff
f8c8  ff 08 00 01 ff 0c 04 05
f8d0  ff ff 02 03 ff 0f 06 07
f8d8  ff 09 0a 0b ff 0d 0e ff
```

Pufferinhalt decodieren

```
;     Pufferzeiger auf Null
f8e0  a9 00     lda  #$00
;     setzen
f8e2  85 34     sta  $34
f8e4  85 2e     sta  $2e
;     2. Pufferzeiger
f8e6  85 36     sta  $36
;     Adresse für
;     Ausweichpuffer
f8e8  a9 01     lda  #$01
f8ea  85 4e     sta  $4e
;     nach $4e/4f
f8ec  a9 ba     lda  #$ba
f8ee  85 4f     sta  $4f
;     Pufferadresse Hi retten
f8f0  a5 31     lda  $31
f8f2  85 2f     sta  $2f
;     Bytes decodieren
f8f4  20 e6 f7  jsr  $f7e6
;     1. Byte als Kennzeichen
f8f7  a5 52     lda  $52
;     für Datenblock
f8f9  85 38     sta  $38
f8fb  a4 36     ldy  $36
;     restliche Bytes in
;     Puffer
f8fd  a5 53     lda  $53
f8ff  91 2e     sta  ($2e),y
f901  c8        iny
f902  a5 54     lda  $54
f904  91 2e     sta  ($2e),y
f906  c8        iny
f907  a5 55     lda  $55
f909  91 2e     sta  ($2e),y
f90b  c8        iny
f90c  84 36     sty  $36
;     nächste Kolonne
;     codieren
f90e  20 e6 f7  jsr  $f7e6
f911  a4 36     ldy  $36
f913  a5 52     lda  $52
;     Bytes in Puffer
;     schreiben
f915  91 2e     sta  ($2e),y
f917  c8        iny
f918  f0 11     beq  $f92b
f91a  a5 53     lda  $53
f91c  91 2e     sta  ($2e),y
f91e  c8        iny
f91f  a5 54     lda  $54
f921  91 2e     sta  ($2e),y
f923  c8        iny
f924  a5 55     lda  $55
f926  91 2e     sta  ($2e),y
f928  c8        iny
;     restliche Bytes ggf.
;     codieren
f929  d0 e1     bne  $f90c
;     Prüfsumme Datenblock
;     übernehmen
f92b  a5 53     lda  $53
f92d  85 3a     sta  $3a
;     Pufferadresse Hi
;     zurückholen
f92f  a5 2f     lda  $2f
f931  85 31     sta  $31
f933  60        rts
```

Listing 3. Die DOS-Routinen zur GCR-Codierung (Schluß)

Hier gibt's Clubs

Luxemburg

Commodore Club Luxemburg (C.C.L.), 1 Rue des Muguets, L 5970 Itzig/Luxemburg, Clubmagazin, Extrainfos, Musikteil, Kummertelefon;

Holland

Copy Tronics, Postbus 700, 7400 AS Deventer Nederland, Clubzeitschrift, Clubtreffen;

Österreich

CHIPSI-Computer-Club, Jasomirgottstr. 3, Mezzanin, 1010 Wien, Programm- und Literaturtausch, Programmierkurse, begünstigter Bezug von Computern, praktische Programmiertips, Clubzeitung, Clubtreffen;

Wiener Hard- u. Software Computerclub, René Fürst, Herbortgasse 22, 1110 Wien, Softwareerstellung und Tausch, Hardwareunterstützung, Beitrag 150 Schilling pro Quartal, Beitrittsgebühr 1x150 Schilling, Tel. (0222) 74391 54, (0222) 61 26 83

Bund Technischer Amateure, Stromstr. 36-38/29/1, 1200 Wien, Clubtreffen, Grundlagen, Tips und Tricks, PRG-Bibliothek, Softwaretausch;

COMUSCLUS, Computer User Club Salzburg, A-5033 Salzburg, Postfach 1 33, Clubtreffen, Clubstammtisch, monatliche Zeitschrift, Buch- und Softwarebibliothek, Arbeitskreise, Programmierkurse, auch für Nichtmitglieder, Sammelbestellungen;

Ruf-Commodore-User-Club/Salzburg, A-5020 Salzburg, Haunspergstraße 21, Tel. (0662) 75770, Herr Wiedermann

Ruf-Commodore-User-Club/Graz, Grazbachgasse 60, Tel. (03 16) 70 02 66, Herr Hiess

Ruf-Commodore-User-Club/Klagenfurth, A-9020 Klagenfurt, Sponheimer Straße 9-11, Tel. 04222) 55105, Herr Krassnitzer

Clubtreffen (wöchentlich), Softwarebibliothek, Hardwareangebot, Programmierkurse etc.

IKC, Internationaler Kommunikationsclub, Postfach 18, CH-6312 Steinhausen

Info gegen 3 DM, Clubzeitschrift, Beratertelefon, DFÜ, Sektionen, Kurse, Listingkorrekturservice, Meisterschaften, etc. Aktiv-, Passiv- und Ehremitglieder sowie Gönner.

Schweiz

COBEC Computer-Benutzer-Club Wohlen, Postfach 10, CH-5613 Hilfikon AG, Schweiz, Clubtreffen, Softwarebibliothek, diverse Hardware, Kurse;

C 64-Anwenderclub Zürich (ACZ), Postfach 194, 8029 Zürich, Clubtreffen, Softwarebibliothek, Hardware, Modem, Assembler- und Basic-Kurse, Jahresbeitrag 50 Franken;

C 64 Club, J. Reich, Am Glattbogen 170, 8050 Zürich
für Anfänger, Clubtreffen, Zeitschrift (geplant), Kurse, Seminare, Softwaretausch, vergünstigter Bezug von Soft- und Hardware, Beitrag 30 Franken

Syntax Error C 64, Postfach 253, CH-9004 St. Gallen, Clubtreffen, Mailbox, DFÜ, Softwarebibliothek, Hard- und Softwareentwicklung.