MICROKIT ASSEMBLER -- VER 2.2

```
                ***     ITT VIDEO- UND LEHRSYSTEME
                ***     SEL AG, PFORZHEIM

                *       ITT MP-LEHRSYSTEM
                *       BETRIEBSPROGRAMM: "MON MF"

                *       RESISTENT IN MASKENROM TYP
                *       AM 9208 EXP01 (1 K X 8 BIT)
                *       ODER INTEL 8308 EXP01

                *                       RP/KTEW, UL, 1.10.76

                *       KONSTANTEN- UND NAMENS-TABELLE:
0002            ASCH    EQU  2          *A-SCHALTER
0001            BSCH    EQU  1          *B-SCHALTER
0004            CSCH    EQU  4          *C-SCHALTER
                *                       =FUNKTIONSSCHALTER
0002            LLAMP   EQU  2          *LINKE LED'S
0001            RLAMP   EQU  1          *RECHTE LED'S
0018            RHBP    EQU  X'18'      *BIT 3 UND 4 ALS
                *                       POSITIONSKENNUNG
                *                       DER SCHALTER "RUN"
                *                       UND "HLT AT BP"
04FF            RMTP    EQU  X'04FF'    *HOECHSTE RAM-ADR
0408            RMRST1  EQU  X'0408'    *SPRUNGZIEL IM RAM
                *                       BEI RST 1-BEFEHL
0400            RMBS    EQU  X'0400'    *UNTERSTE RAM-ADR
0410            RAMINT  EQU  X'0410'    *SPRUNGZIEL IM RAM
                *                       BEI RST 7-BEFEHL
                *                       ODER BEI INTERUPT
0002            EXMS    EQU  2          *BIT 1 FUER EXAMIN
0002            EXJA    EQU  2          *BIT 1=1, WENN
                *                       EXAMIN
0001            LDMS    EQU  1          *BIT 0 FUER LOAD-
                *                       ADR-SCHALTER
0001            LDJA    EQU  1          *BIT 0=1, WENN
                *                       LOAD-ADR
0004            DPMS    EQU  4          *BIT 2 FUER DEPOS
0004            DPJA    EQU  4          *BIT 2=1, WENN
                *                       DEPOSIT
0006            EDMS    EQU  6          *BIT 1 UND 2 FUER
                *                       EXAM OD. DEPOS
0008            RNMS    EQU  8          *BIT 3 FUER RUN
0008            RNJA    EQU  8          *BIT 3=1, WENN RUN
0000            RNNO    EQU  0          *BIT 3=0, WENN
                *                       NICHT RUN
00E0            SYMS    EQU  X'E0'      *BITS FUER SYSTEM
                *                       SCHALTER
0020            S6JA    EQU  X'20'      *BITS, WENN SYST.6
0000            S7JA    EQU  0          *BITS, WENN SYST.7
0800            SYST7   EQU  X'0800'    *STARTADR FUER
                *                       SYSTEM 7 (ERWEI-
                *                       TERUNG

0000                    ORG  X'0000'    *SYSTEMSTARTADR

0000 31FE04 MONMP   LXI  SP,RMTP-1 *STACK-POINTER
```

MICROKIT ASSEMBLER -- VER 2.2

```
                *                       INITIALISIEREN
0003 DB04               IN    CSCH      *FU.SCH. EINLESEN
0005 C39D00             JMP   MAIN      *SPRUNG ZUM HAUPT
                *                       TEIL, UM DIE NACH
                *                       FOLGENDEN START-
                *                       ADRESSEN FUER DIE
                *                       RST-BEFEHLE FREI
                *                       ZU BEKOMMEN.
0008 C30804     RAMGO   JMP   RMRST1    *NEUES ZIEL FUER
                *                       RST 1-BEFEHL
000B 00                 NOP
000C 210004     SYST6   LXI   H,RMBS    *STARTADR FUER
                *                       "RUN" LADEN
000F E5                 PUSH  H         *FUER "RUN" NACH
                *                       RUNSUB AUF STACK
0010 F5         BREAK   PUSH  PSW       *EINSPRUNG IN MON
0011 C5                 PUSH  B         *MIT RST 2 FUER
0012 D5                 PUSH  D         *"RUN" MIT "HLT
                *                        AT BP", GESAMT-
                *                       STATUS FUER DIS-
                *                       PLAY AUF DEN STACK
                *                      (PUSH H FOLGT NOCH)
0013 0618               MVI   B,RHBP    *MASKE FUER "RUN
                *                       HLT AT BP" LADEN
                *                       (FLANKENERKENNUNG)
0015 C39903             JMP   BREAK2    *ZUR FORTSETZUNG

                *       SUBROUTINE "SRCKJP"
                *PRUEFT, OB DIE DURCH DIE "MASKE" FEST-
                *GELEGTEN BITS DES INH. DES B-REG. MIT
                *DER ZAHL "VERGL" UEBEREINSTIMMEN.
                *WENN JA, ERFOLGT EIN SPRUNG ZU "SPRADR"
                *WENN NEIN, GEHT ES IM PROGR. WEITER.
                *       AUFRUF MIT:     RST   3,SRCKJP
                *                       DC    MASKE
                *                       DC    VERGL
                *                       DC    B(SPRADR)
0018 E3         SRCKJP  XTHL
0019 F5                 PUSH  PSW
001A 78                 MOV   A,B
001B A6         FLJPRT  ANA   M
001C 23                 INX   H
001D C36300             JMP   CKJP2     *ZUR FORTS.

                *       SUBROUTINE "SCKCK1"
                *PRUEFT, OB DIE DURCH "MASKE" FESTGE-
                *LEGTEN BITS DES IHN. DES B-REG. MIT DER
                *ZAHL "VERGL" UEBEREINSTIMMEN.
                *WENN JA, WERDEN DIE 3 AUF DIESE SR
                *FOLGENDEN BYTES BEARBEITET,
                *WENN NEIN UEBERSPRUNGEN.
                *       AUFRUF MIT:     RST   4,SCKCK1
                *                       DC    MASKE
                *                       DC    VERGL
0020 E3         SCKCK1  XTHL
0021 F5                 PUSH  PSW
0022 78                 MOV   A,B
```

```
0023 A6             ANA  M
0024 23             INX  H
0025 C35000         JMP  CKSK1      *ZUR FORTS.

                    *AUF DER ZIELADRESSE DES RST 5-BEFEHLS
                    *BEGINNT EINE FUER DIE SYSTEM-SIMULATION
                    *BENOETIGTE SR
0028                SIMSR  DS   8

                    *       SUBROUTINE "SRCKSK"
                    *PRUEFT, OB DIE DURCH "MASKE" FESTGE-
                    *LEGTEN BITS DES INH. DES B-REG. MIT DER
                    *ZAHL "VERGL" UEBEREINSTIMMEN,
                    *WENN JA, WERDEN DIE AUF DIESE SR FOL-
                    *GENDEN 9 BYTES BEARBEITET,
                    *WENN NEIN, WERDEN SIE UEBERSPRUNGEN.
                    *       AUFRUF MIT:     RST  6,SRCKSK
                    *                       DC   MASKE
                    *                       DC   VERGL
0030 E3             SRCKSK XTHL
0031 F5                    PUSH PSW
0032 78                    MOV  A,B
0033 A6                    ANA  M
0034 23                    INX  H
0035 C35800                JMP  CKSK2      *ZUR FORTS.
0038 C31004         INTERU JMP  RAMINT     *NEUES ZIEL FUER
                    *                       RST 7-BEFEHL, DER
                    *                       U.A. AUCH DURCH
                    *                       EINEN HARDWARE-
                    *                       INERUPT VON DER
                    *                       HARDWARE ERZEUGT
                    *                       WIRD.

                    *       SUBROUTINE "SFLANK"
                    *PRUEFT AUF DIE L > H-FLANKE DES MIT
                    *"MASKE" FESTGELEGTEM BITS ZWISCHEN DEM
                    *ALTEN ZUSTAND (IM D-REG.) UND NEUEM
                    *ZUSTAND (IM B-REG.).
                    *WENN L > H-FLANKE, GEHT ES IM PROGRAMM
                    *WEITER,
                    *WENN  K E I N E  FLANKE, SPRUNG ZU
                    *"SPRADR"
                    *       AUFRUF MIT:     CALL SFLANK
                    *                       DC   MASKE
                    *                       DC   B(SPRADR)
003B E3             SFLANK XTHL
003C F5                    PUSH PSW
003D 7A                    MOV  A,D
003E 2F                    CMA
003F A0                    ANA  B
0040 A6             NFLEND ANA  M
0041 23                    INX  H
0042 CA6800                JZ   CNFUND
0045 C35E00                JMP  CKJPND

                    *       SUBROUTINE "NFLANK"
                    *PRUEFT AUF H > L-FLANKE DER MIT "MASKE"
```

```
                *FESTGELEGTEN BITS ZWISCHEN DEM ALTEN
                *ZUSTAND (IM D-REG.) UND NEUEM ZUSTAND
                *(IM B-REG.)
                *WENN H > L-FLANKE, GEHT ES IM PROGRAMM
                *WEITER,
                *WENN  K E I N E  FLANKE, SPRUNG ZUR
                *"SPRADR".
                *       AUFRUF MIT:     CALL NFLANK
                *                       DC   MASKE
                *                       DC   B(SPRADR)
0048 E3         NFLANK XTHL
0049 F5                PUSH PSW
004A 78                MOV  A,B
004B 2F                CMA
004C A2                ANA  D
004D C34000            JMP  NFLEND

                *       FORTS. DER SR'S
0050 BE         CKSK1  CMP  M
0051 23                INX  H
0052 CA6000            JZ   CKSEND
0055 C35F00            JMP  CKSCON

0058 BE         CKSK2  CMP  M
0059 23                INX  H
005A CA6000            JZ   CKSEND
005D 23                INX  H
005E 23         CKJPND INX  H
005F 23         CKSCON INX  H
0060 F1         CKSEND POP  PSW
0061 E3                XTHL
0062 C9                RET

0063 BE         CKJP2  CMP  M
0064 23                INX  H
0065 C25E00            JNZ  CKJPND
0068 7E         CNFUND MOV  A,M
0069 23                INX  H
006A 66                MOV  H,M
006B 6F                MOV  L,A
006C F1         CNEND  POP  PSW
006D E3                XTHL
006E C9                RET

                *       SUBROUTINE "RUNSUB"
                *PRUEFT DIE STELLUNG DER FUNKTIONS-
                *SCHALTER. ES ERFOLGT EINE VERZWEIGUNG
                *ZU "EXAM" ODER ZU "DEPOS" WENN DIESE
                *SCHALTER BETAETIGT WURDEN, ODER ZUR
                *STARTADR "RUN", WENN RUN BETAETIGT,
                *ODER ZURUECK IN DEN MONITOR ZU "DISLOP"
                *       AUFRUF MIT:     CALL RUNSUB
                *                       DC   B(EXAM)
                *                       DC   B(DEPOS)
                *                       DC   B(DISLOP)
006F E3         RUNSUB XTHL
0070 F5                PUSH PSW
```

```
0071 CD9600          CALL FLNKJP
0074 02              DC   EXMS
0075 02              DC   EXJA
0076 6800            DC   B(CNFUND)
0078 23              INX  H
0079 23              INX  H
007A CD9600          CALL FLNKJP
007D 04              DC   DPMS
007E 04              DC   DPJA
007F 6800            DC   B(CNFUND)
0081 23              INX  H
0082 23              INX  H
0083 DF              RST  3,SRCKJP
0084 08              DC   RNMS
0085 00              DC   RNNO
0086 6800            DC   B(CNFUND)
0088 DF              RST  3,SRCKJP
0089 18              DC   RHBP
008A 08              DC   RNJA
008B 5E00            DC   B(CKJPND)
008D CD3B00          CALL SFLANK
0090 08              DC   RNMS
0091 6800            DC   B(CNFUND)
0093 C35E00          JMP  CKJPND

                *      SUBROUTINE "FLNKJP"
                *PRUEFT AUF FLANKE DER DURCH "MASKE"
                *FESTGELEGTEN BITS ZWISCHEN DEM ALTEN
                *ZUSTAND (IM D-REG.) UND DEM NEUEM (IM
                *B-REG.).
                *WENN FLANKE, SPRUNG ZU "SPRADR",
                *SONST WARTET DIE SR AUF DIE FLANKE.
0096 E3      FLNKJP XTHL
0097 F5             PUSH PSW
0098 78             MOV  A,B
0099 B2             ORA  D
009A C31B00         JMP  FLJPRT


009D 50      MAIN   MOV  D,B        *ALTER ZUSTAND DES
009E 47             MOV  B,A        *B-REG. NACH D
                *                   UM AUF FLANKEN
                *                   PRUEFEN ZU KOENNEN

                *      HIER WERDEN DIE SYSTEME 1 BIS 5
                *      ENTSPRECHEND WIE FOLGT DIE SYS-
                *      TEME 6 UND 7 VON DEM SYSTEM-
                *      SCHALTER ABGEFRAGT. ES ERFOLGT
                *      EINE SPRUNGVERZWEIGUNG ZUM ENT-
                *      SPRECHENDEM SIMULATIONSPROGRAMM.
009F         SY1TO5 DS   25

00B8 DF      SY6    RST  3,SRCKJP
00B9 E0             DC   SYMS
00BA 20             DC   S6JA
00BB 0C00           DC   B(SYST6)
00BD DF      SY7    RST  3,SRCKJP
```

MICROKIT ASSEMBLER -- VER 2.2

```
00BE E0             DC    SYMS
00BF 00             DC    S7JA
00C0 0008           DC    B(SYST7)
                    *     SONST GEHT DAS PROGRAMM WEITER
                    *     ZU SYSTEM 0.
                    *     NACHFOLGENDER SPEICHERRAUM IST
                    *     MIT DEN SIMULATIONSROUTINEN DER
                    *     SYSTEME 0 BIS 5 BELEGT.
00C2                SIMPR DS    727

0399 E5       BREAK2 PUSH H           *FORTS. DER RET-
                    *                 TUNG DES STATUS
039A 210000         LXI   H,0         *CLEAR HL
039D 39             DAD   SP          *HL MIT SP LADEN
039E 22FE04         SHLD  RMTP-1      *SP IN RAMTOP-1
                    *                 ABSPEICHERN
03A1 DB02     DISLOP IN   ASCH        *STACK-OFFSET
                    *                 EINLESEN
03A3 5D             MOV   E,L         *EINGANGS-ADR
                    *                  RETTEN
03A4 21FE04         LXI   H,RMTP-1    *HL MIT SP LADEN
03A7 86             ADD   M           *SP ZUM STACK-
                    *                 OFFSET ADIEREN,
                    *                 =ECHTE ADRESSE
03A8 6F             MOV   L,A         *DIESE NACH L ALS
                    *                 LOW-ADR
03A9 7E             MOV   A,M         *INH. DIESER ADR
                    *                 (=REGISTER AUF
                    *                 STACK) LADEN
03AA D302           OUT   LLAMP       *UND ANZEIGEN
03AC DB01           IN    BSCH        *GEWAEHLTE LOW-ADR
                    *                 EINLESEN
03AE 6F             MOV   L,A         *LOW-ADR NACH L
03AF 7E             MOV   A,M         *INH. DER RAM-ADR
                    *                 LADEN
03B0 D301           OUT   RLAMP       *UND ANZEIGEN
03B2 6B             MOV   L,E         *GERETTETE EIN-
                    *                  GANGS-ADR ZURUECK
03B3 DB04     BRKLOP IN   CSCH        *FUNKTIONSSCH.
                    *                 EINLESEN
03B5 50             MOV   D,B         *ALTEN INHALT DER
                    *                 FU.SCH. NACH D
03B6 47             MOV   B,A         *NEUEN INH. DER
                    *                 FU.SCH. NACH B
                    *                 ZUR ERKENNUNG VON
                    *                 AENDERUNGEN
03B7 DB02           IN    ASCH        *ASCH FUER L-ADR
                    *                 ODER DATEN LESEN
03B9 DF             RST   3,SRCKJP    *ABFRAGE OB "LOAD
03BA 01             DC    LDMS        *ADRESS"?
03BB 01             DC    LDJA        *WENN JA,
03BC CC03           DC    B(LOADAD)   *DANN ZU LOADAD
03BE CD6F00         CALL  RUNSUB      *ABFRAGE OB
03C1 D703           DC    B(EXAMIN)   *EXAMIN, JA >>>
03C3 D003           DC    B(DEPOS)    *DEPOS, JA >>>
03C5 A103           DC    B(DISLOP)   *NICHT-RUN, JA >>>
03C7 E1       CONTIN POP  H           *WENN "RUN", DANN
```

MICROKIT ASSEMBLER -- VER 2.2

```
03C8 D1            POP  D          *GESAMTSTATUS ZU-
03C9 C1            POP  B          *RUECK UND RUECK-
03CA F1            POP  PSW        *KEHR ZUM AUFRU-
03CB C9            RET             *FENDEN ANWENDER-
                   *               PROGRAMM (RST 2 !)
03CC 6F      LOADAD MOV  L,A       *ASCH (=L-ADR)
                   *               NACH L
03CD C3D703        JMP  EXAMIN     *ZUR ANZEIGE VON
                   *               L-ADR UND DEREN
                   *               INHALTS
03D0 CD3B00  DEPOS CALL SFLANK     *H>L-FLANKE VON
03D3 04            DC   DPMS       *DEPOS-SCH SUCHEN
03D4 D703          DC   B(EXAMIN)  *WENN KEINE FLANKE
                   *               DANN ZU EXAMIN
03D6 77            MOV  M,A        *INH. ASCH (=DATA)
                   *               IN GEWAEHLTER ADR
                   *               ABSPEICHERN
03D7 7E      EXAMIN MOV  A,M       *INH. VON GEWAEHL
                   *               TER ADR HOLEN
03D8 D301          OUT  RLAMP      *INH. ANZEIGEN
03DA 7D            MOV  A,L        *L-ADR HOLEN
03DB D302          OUT  LLAMP      *L-ADR ANZEIGEN
03DD CD4800        CALL NFLANK     *H>L-FLANKE VON
03E0 06            DC   EDMS       *EXAM. OD. DEPOS
03E1 B303          DC   B(BRKLOP)  *SUCHEN, WENN
                   *               KEINE FLANKE, DANN
                   *               ZUR BRKLOP ZURUECK
03E3 2C            INR  L          *L  AUF NAECHSTE
                   *               ADRESSE (AUTOIN-
                   *               CREMENT)
03E4 C3B303        JMP  BRKLOP     *ZURUECK ZUR AB-
                   *               FRAGE WEITERER
                   *               FUNKTIONEN

                   *      IM FOLGENDEM SPEICHERRAUM LIEGT
                   *DIE MULTIPLIKATIONS-SUBROUTINE DES
                   *SYSTEMS 5. DIESE WIRD VOM SYSTEM 5
                   *IN DIE UNTEREN RAM-ADRESSEN KOPIERT UND
                   *VOM SYSTEM 5 WIE EIN ROM BENUTZT, D.H.
                   *SIE IST DORT VOM ANWENDER NICHT VER-
                   *AENDERBAR ODER ZERSTOERBAR.
03E7               MULT5 DS   25

                   ***    NO ERRORS      D.ULRICH PR/KTEW
0400               END  MONMP
```

MICROKIT ASSEMBLER -- VER 2.2

```
                ***     ITT VIDEO- UND LEHRSYSTEME
                ***     SEL AG, PFORZHEIM

                *       ITT MP-LEHRSYSTEM, EXTENSION-BOX
                *       BETRIEBSPROGRAMM: "MON 7"

                *       RESISTENT IN MASKENROM TYP
                *       AM 9214 EXP02 (512 X 8 BIT)

                *                       RP/KTEW;UL;17.1.78

                *       KONSTANTEN- UND NAMENS-TABELLE:
0002            ASCH    EQU  2          *A-SCHALTER (MP-S)
0001            BSCH    EQU  1          *B-SCHALTER (MP-S)
0004            CSCH    EQU  4          *C-SCHALTER (MP-S)
                *                       =FUNKTIONSSCHALTER
0008            HSCH    EQU  8          *SCHALTER DER EX-
                *                       BOX + CAS. INPUT
0008            HLAMP   EQU  8          *LED'S FUER H-ADR
                *                       DER EX.-BOX +
                *                       CAS. OUTPUT
0002            LLAMP   EQU  2          *LINKE LED'S(MP-S)
0001            RLAMP   EQU  1          *RECHTE LED'S (MP)
0C00            USROM   EQU  X'0C00'    *ANF. ADR DES USER
                *                       -REPROM (2708)
0064            USCODE  EQU  X'64'      *CODE AUF 1. PLATZ
                *                       DES USER-ROM, WENN
                *                       DIESES AUTOMATISCH
                *                       IN BETRIEB GEHEN
                *                       SOLL
0018            RHBP    EQU  X'18'      *BIT 3 UND 4 ALS
                *                       POSITIONSKENNUNG
                *                       DER SCHALTER "RUN"
                *                       UND "HLT AT BP"
04FF            RMTP    EQU  X'04FF'    *RAMTOP DER RAM-
                *                       PAGE 0 MIT DEM
                *                       SYSTEM-STACK
0001            LDMS    EQU  1          *BIT 0 FUER LOAD-
                *                       ADR-SCHALTER
0001            LDJA    EQU  1          *BIT 0=1, WENN
                *                       LOAD-ADR
0010            SHMS    EQU  X'10'      *BIT 4 FUER SHIFT
0010            SHJA    EQU  X'10'      *BIT 4=1, WENN
                *                       SHIFT
0040            CSMS    EQU  X'40'      *BIT 6 FUER CAS.
0040            CSJA    EQU  X'40'      *BIT 6=1, WENN CAS
006F            RUNSUB  EQU  X'006F'    *ADR DER SR IM MP
003B            SFLANK  EQU  X'003B'    *ADR DER SR IM MP
0004            DPMS    EQU  4          *BIT 2 FUER DEPOS
0048            NFLANK  EQU  X'0048'    *ADR DER SR IM MP
0006            EDMS    EQU  6          *BIT 1 UND 2 FUER
                *                       DEPOS OD. EXAMIN
0006            PGMS    EQU  6          *BIT 1 UND 2 FUER
                *                       PAGE-SCH. (EX-BOX)
0018            UPMS    EQU  X'18'      *BIT 3 UND 4 FUER
                *                       SHIFT-UP (EX-BOX)
0018            UPJA    EQU  X'18'      *BIT 3=1, BIT 4=1,
```

MICROKIT ASSEMBLER -- VER 2.2

```
                *                               WENN SHIFT-UP
0060            RDMS    EQU   X'60'            *BIT 5 UND 6 FUER
                *                               READ-CAS
0040            RDJA    EQU   X'40'            *BIT 5=0, BIT 6=1
                *                               WENN READ-CAS
0050            TSMS    EQU   X'50'            *BIT 4 UND 6 FUER
                *                               CASRUN UND SHIFT
0000            TSNO    EQU   X'00'            *BIT 4=0, BIT 6=0
                *                               WENN WEDER CASRUN,
                *                               NOCH SHIFT
0016            SYN     EQU   X'16'            *"SYN"-ZEICHEN DES
                *                               ASCII-CODES
0002            STX     EQU   X'02'            *"STX"-ZEICHEN DES
                *                               ASCII-CODES
0003            ETX     EQU   X'03'            *"ETX"-ZEICHEN DES
                *                               ASCII-CODES
FFF0            SLANG   EQU   -16              *HALBWELLENANZAHL
                *                               FUER BIT=1
FFF9            SKURZ   EQU   -7               *HALBWELLENANZAHL
                *                               FUER BIT=0
000C            SWAI    EQU   12               *ZAHL FUER HALB-
                *                               WELLENZEIT CA. 250
                *                               MICROSEC, D.H. CA.
                *                               F= 2 KHZ
0007            LZEIT   EQU   7                *KONSTANTE FUER
                *                               BIT-PRUEFZEIT

0000                    ORG   X'0800'

0800 3A000C     MON7    LDA   USROM            *1. PLATZ AUS USROM
0803 FE64               CPI   USCODE           *LADEN UND AUF INH
0805 CA000C             JZ    USROM            *USROM VERGL.,
                *                               WENN JA, DANN ZU
                *                               USROM
0808 2E00       SYS7    MVI   L,0              *L-ADR=0 FUER RUN
                *                               LADEN
080A CD8108             CALL  IND              *PAGE-SCHALTER
                *                               EINLESEN UND H MIT
                *                               HIGH-ADR FUER "RUN
                *                               LADEN
080D E5                 PUSH  H                *START-ADR "RUN"
                *                               AUF STACK
080E F5         BREAK7  PUSH  PSW              *EINSPRUNG IN MON7
080F C5                 PUSH  B                *MIT RST 1 FUER
0810 D5                 PUSH  D                *"RUN" MIT "HLT AT
0811 E5                 PUSH  H                *AT BP", GESAMT-
                *                               STATUS FUER DIS-
                *                               PLAY AUF DEN STACK
0812 0618               MVI   B,RHBP           *MASKE FUER "RUN
                *                               HLT AT BP" LADEN
0814 210000             LXI   H,0              *CLEAR HL
0817 39                 DAD   SP               *HL MIT SP LADEN
0818 22FE04             SHLD  RMTP-1           *SP IN RAMTOP-1
                *                               ABSPEICHERN
081B DB02       DISLOP  IN    ASCH             *STACK-OFFSET
                *                               EINLESEN
081D 5D                 MOV   E,L              *EINGANGS-ADR
```

MICROKIT ASSEMBLER -- VER 2.2

```
081E 4C              MOV  C,H        *RETTEN
081F 21FE04          LXI  H,RMTP-1   *HL MIT SP LADEN
0822 86              ADD  M          *SP ZUM STACK-
               *                     OFFSET ADIEREN,
               *                     =ECHTE ADRESSE
0823 6F              MOV  L,A        *DIESE NACH L ALS
               *                     LOW-ADR
0824 7E              MOV  A,M        *INH. DIESER ADR
               *                     (=REGISTER AUF
               *                     STACK) LADEN
0825 D302            OUT  LLAMP      *UND ANZEIGEN
0827 CD8108          CALL IND        *PAGE-SCH. EINLE-
               *                     SEN UND HIGH-ADR
               *                     DES GEWAEHLTEN RAM
               *                     NACH H LADEN
082A 07              RLC             *BITKORREKTUR
082B D308            OUT  HLAMP      *HI-ADR AUF EX-BOX
               *                     ANZEIGEN
082D DB01            IN   BSCH       *GEWAEHLTE LOW-ADR
               *                     EINLESEN
082F 6F              MOV  L,A        *LOW-ADR NACH L
0830 7E              MOV  A,M        *INH. DER RAM-ADR
               *                     LADEN
0831 D301            OUT  RLAMP      *UND ANZEIGEN
0833 6B              MOV  L,E        *GERETTETE EIN-
0834 61              MOV  H,C        *GANGS-ADR ZURUECK
0835 DB04    BRKLOP  IN   CSCH       *FUNKTIONSSCH.
               *                     EINLESEN
0837 50              MOV  D,B        *ALTEN INHALT DER
               *                     FU.SCH. NACH D
0838 47              MOV  B,A        *NEUEN INH. DER
               *                     FU.SCH. NACH B
               *                     ZUR ERKENNUNG VON
               *                     AENDERUNGEN
0839 DB08            IN   HSCH       *HSCH ZUR ERKENN.
               *                     DER FUNKTIONEN
               *                     PAGE, SHIFT UND
               *                     CASSETTE EINLESEN
083B 4F              MOV  C,A        *NACH C KOPIEREN
083C DB02            IN   ASCH       *ASCH FUER L-ADR
               *                     ODER DATEN LESEN
083E DF              RST  3,SRCKJP   *ABFRAGE OB "LOAD
083F 01              DC   LDMS       *ADRESS"?
0840 01              DC   LDJA       *WENN JA,
0841 5E08            DC   B(LOADAD)  *DANN ZU LOADAD
0843 58              MOV  E,B        *FU.SCH. NACH E
0844 41              MOV  B,C        *HSCH NACH B FUER
               *                     ABFRAGE
0845 DF              RST  3,SRCKJP   *ABFRAGE OB
0846 10              DC   SHMS       *"SHIFT"?
0847 10              DC   SHJA       *WENN JA,
0848 8A08            DC   B(SHIFT)   *DANN ZU SHIFT
084A DF              RST  3,SRCKJP   *ABFRAGE OB
084B 40              DC   CSMS       *"CASSETTE"?
084C 40              DC   CSJA       *WENN JA,
084D C408            DC   B(CASRUN)  *DANN ZU CASRUN
084F 43              MOV  B,E        *FU.SCH. NACH B
```

```
                *                               ZURUECK
0850 CD6F00             CALL  RUNSUB    *ABFRAGE OB
0853 6D08               DC    B(EXAMIN) *EXAMIN, JA >>>
0855 6608               DC    B(DEPOS)  *DEPOS, JA >>>
0857 1B08               DC    B(DISLOP) *NICHT-RUN, JA >>>
0859 E1                 POP   H         *WENN "RUN", DANN
085A D1                 POP   D         *GESAMTSTATUS ZU-
085B C1                 POP   B         *RUECK UND RUECK-
085C F1                 POP   PSW       *KEHR ZUM AUFRU-
085D C9                 RET             *FENDEN ANWENDER-
                *                       PROGRAMM (RST 1 !)
085E 6F         LOADAD  MOV   L,A       *ASCH (=L-ADR)
                *                       NACH L
085F 79                 MOV   A,C       *HSCH (PAGE)
                *                       NACH AC
0860 CD8308             CALL  HIAD      *H-ADR BILDEN UND
                *                       NACH H
0863 C36D08             JMP   EXAMIN    *ZUR ANZEIGE VON
                *                       L- UND H-ADR UND
                *                       DEREN INHALTS
0866 CD3B00     DEPOS   CALL  SFLANK    *H>L-FLANKE VON
0869 04                 DC    DFMS      *DEPOS-SCH SUCHEN
086A 6D08               DC    B(EXAMIN) *WENN KEINE FLANKE
                *                       DANN ZU EXAMIN
086C 77                 MOV   M,A       *INH. ASCH (=DATA)
                *                       IN GEWAEHLTER ADR
                *                       ABSPEICHERN
086D 7E         EXAMIN  MOV   A,M       *INH. VON GEWAEHL
                *                       TER ADR HOLEN
086E D301               OUT   RLAMP     *INH. ANZEIGEN
0870 7D                 MOV   A,L       *L-ADR HOLEN
0871 D302               OUT   LLAMP     *L-ADR ANZEIGEN
0873 7C                 MOV   A,H       *H-ADR HOLEN
0874 07                 RLC             *BITKORREKTUR
0875 D308               OUT   HLAMP     *H-ADR ANZEIGEN
0877 CD4800             CALL  NFLANK    *H>L-FLANKE VON
087A 06                 DC    EDMS      *EXAM. OD. DEPOS
087B 3508               DC    B(BRKLOP) *SUCHEN, WENN
                *                       KEINE FLANKE, DANN
                *                       ZUR BRKLOP ZURUECK
087D 23                 INX   H         *HL AUF NAECHSTE
                *                       ADRESSE (AUTOIN-
                *                       CREMENT)
087E C33508             JMP   BRKLOP    *ZURUECK ZUR AB-
                *                       FRAGE WEITERER
                *                       FUNKTIONEN

0881 DB08       IND     IN    HSCH      *H-SCH FUER H-ADR
0883 E606       HIAD    ANI   PGMS      *(PAGE) EINLESEN,
0885 0F                 RRC             *PAGE ISOLIEREN,
0886 C604               ADI   4         *ZUR H-ADR KORRI-
0888 67                 MOV   H,A       *GIEREN UND NACH H
0889 C9                 RET             * ZURUECK VON IND
                *                       BEZW. HIAD

088A DB01       SHIFT   IN    BSCH      *SCHRITTZAHL EIN-
088C 4F                 MOV   C,A       *LESEN UND NACH C
```

MICROKIT ASSEMBLER -- VER 2.2

```
0880 DB02          IN   ASCH      *BLOCKLAENGE EIN-
                   *               LESEN
088F DF            RST  3,SRCKJP  *ABFRAGE OB
0890 18            DC   UPMS      *"UP"(VORWAERTS)?
0891 18            DC   UPJA      *WENN JA,
0892 AC08          DC   B(FORW)   *DANN ZU FORW
0894 B7      REW   ORA  A         *SETZE C-FLAG=0
0895 47            MOV  B,A       *BLOCKZAEHLER LAD.
0896 54            MOV  D,H       *ALTE ANF.ADR NACH
0897 5D            MOV  E,L       *NACH DE
0898 7D            MOV  A,L       *SUBTRAKTION DER
0899 91            SUB  C         *SCHRITTZAHL ER-
089A 6F            MOV  L,A       *GIBT NEUE ANF.ADR
089B D29F08        JNC  RSH       *WENN ANDERE H-ADR
089E 25            DCR  H         *DANN KORREKTUR
089F 1A      RSH   LDAX D         *INH. AUS ALTER
                   *               ADR LADEN
08A0 77            MOV  M,A       *IN NEUE BRINGEN
08A1 97            SUB  A         *AC LOESCHEN (=0)
08A2 12            STAX D         *"0" IN ALTE ADR
08A3 13            INX  D         *BEIDE ADR UM 1
08A4 23            INX  H         *ERHOEHEN
08A5 05            DCR  B         *BLOCKZAEHLER -1
08A6 C29F08        JNZ  RSH       *ZUR NAECHSTEN ADR
                   *               WENN BLOCK NOCH
                   *               NICHT ZU ENDE
08A9 C3E909        JMP  SHEND     *NACH ENDE DES
                   *               BLOCKES ZU SHEND
08AC 5F      FORW  MOV  E,A       *BLOCKLAENGE > E
08AD 1D            DCR  E         *KORREKTUR
08AE 1600          MVI  D,0       *D LOESCHEN (=0)
08B0 42            MOV  B,D       *B LOESCHEN
08B1 19            DAD  D         *ALTE ANF.ADR +
                   *               BLOCKL. = ALTE END
                   *               ADR
08B2 54            MOV  D,H       *ALTE END-ADR NACH
08B3 5D            MOV  E,L       *DE
08B4 09            DAD  B         *ALTE END-ADR +
                   *               SCHRITTZAHL =
                   *               NEUE END-ADR
08B5 4F            MOV  C,A       *BLOCKZAEHLER LAD.
08B6 1A      FSH   LDAX D         *INH. AUS ALTER
08B7 77            MOV  M,A       *ADR IN NEUE
08B8 97            SUB  A         *AC MIT "0" LADEN
08B9 12            STAX D         *"0" IN ALTE ADR
08BA 1B            DCX  D         *BEIDE ADR ER-
08BB 2B            DCX  H         *NIEDRIGEN
08BC 0D            DCR  C         *BLOCKZAEHLER -1
08BD C2B608        JNZ  FSH       *ZUR NAECHSTEN ADR
                   *               WENN BLOCK NOCH
                   *               NICHT ZU ENDE
08C0 C3E909        JMP  SHEND     *WENN ZU ENDE DANN
                   *               ZU SHEND
08C3 00            NOP
08C4 DF      CASRUN RST 3,SRCKJP  *ABFRAGE OB LESEN?
08C5 60            DC   RDMS      *WENN JA,
08C6 40            DC   RDJA      *DANN ZU
```

MICROKIT ASSEMBLER -- VER 2.2

```
08C7 3909          DC   B(LBLK)   *LESEN (LBLK)
08C9 78            MOV  A,B       *HSCH HOLEN
08CA CD8308        CALL HIAD      *H-ADR BILDEN > H
08CD 2E00          MVI  L,0       *L-ADR= 0
08CF 1EFF          MVI  E,255     *BLOCKLAENGE =255
08D1 1620    SBLK  MVI  D,32      *ANZAHL DER SYN-
             *                    ZEICHEN LADEN
08D3 0E16    SYNLP MVI  C,SYN     *SYN-ZEICHEN LAD.
08D5 CDFB08        CALL SBYTE     *"SYN" SCHREIBEN
08D8 15            DCR  D         *ANZAHL -1
08D9 C2D308        JNZ  SYNLP     *NAECHSTES "SYN"
             *                    BIS ALLE RAUS
08DC 0E02          MVI  C,STX     *STX-ZEICHEN LAD.
08DE CDFB08        CALL SBYTE     *"STX" SCHREIBEN
08E1 4C            MOV  C,H       *H-ADR LADEN
08E2 CDFB08        CALL SBYTE     *H-ADR SCHREIBEN
08E5 4E            MOV  C,M       *1.DAT-BYTE LADEN
08E6 23            INX  H         *ADR AUF 2.DAT-BYT
08E7 CDFB08        CALL SBYTE     *1.DAT-BYTE SCHR.
08EA 4E      SDAT  MOV  C,M       *DATA-BYTE LADEN
08EB 23            INX  H         *ADR AUF NAECHSTES
             *                    DATA-BYTE
08EC CDFB08        CALL SBYTE     *DATA-BYTE SCHR.
08EF 1D            DCR  E         *BLOCKLAENGE -1
08F0 C2EA08        JNZ  SDAT      *NAECHSTES BYTE
             *                    SCHREIBEN BIS
             *                    BLOCK ZU ENDE
08F3 0E03          MVI  C,ETX     *ETX-ZEICHEN LADEN
08F5 CDFB08        CALL SBYTE     *"ETX" SCHREIBEN
08F8 C3E909        JMP  SHEND     *RUECKSPRUNG, WENN
             *                    FERTIG

             *     SUBROUTINES ZUM SCHREIBEN
08FB 79      SBYTE MOV  A,C       *BYTE VON C HOLEN
08FC 37            STC            *SETZE C-FLAG=1
08FD 1F      SCMX  RAR            *LSB NACH C-FLAG
08FE 4F            MOV  C,A       *REST NACH C RETT.
08FF 3EF0          MVI  A,SLANG   *CODE FUER BIT=1
             *                    LADEN
0901 DA0609        JC   STAKT     *WENN LSB=C-FLAG=
             *                    1, DANN ZU STAKT
0904 3EF9          MVI  A,SKURZ   *CODE FUER BIT=0
             *                    LADEN
0906 47      STAKT MOV  B,A       *BIT-CODE NACH B
             *                    (HALBWELLENZAEHL.)
0907 CD2E09        CALL SZEIT     *POS. HALBW. DES
             *                    TAKTES SCHREIBEN
090A CA1A09        JZ   SEND      *WENN CODE-ZAEHLER
             *                    AUF 0, DANN > SEND
090D CD2E09        CALL SZEIT     *NEG.HALBW. SCHR.
0910 C20709        JNZ  STAKT+1   *WENN CODE-Z. #0,
             *                    DANN POS.HALBW.
0913 D308          OUT  HLAMP     *AUSGABE EINER
             *                    HALBWELLE
0915 3EF9          MVI  A,SKURZ   *BITCODE 0 LADEN
0917 C31E09        JMP  SVERZ-1   *CODEZAEHLER LAD.
091A D308    SEND  OUT HLAMP      *ENDE EINER HALBW.
```

MICROKIT ASSEMBLER -- VER 2.2

```
091C 3EF0          MVI  A,SLANG   *BITCODE 1 LADEN
091E 47            MOV  B,A       *CODEZ. LADEN
091F CD3109 SVERZ  CALL ZSCH      *ZEITSCHLEIFE,
             *                     SCHREIBE "0" FUER
             *                     DEN REST DER BIT-
             *                     ZEIT AUS
0922 220008        SHLD MON7      *BEFEHL OHNE WIR-
             *                     KUNG; VERZOEGERUNG
0925 C21F09        JNZ  SVERZ     *VERZOEGERE BIS
             *                     CODEZ =0
0928 79            MOV  A,C       *BYTE-REST HOLEN
0929 B7            ORA  A         *WENN AC=0, DANN
092A C8            RZ             *RETURN
092B C3FD08        JMP  SCMX      *SONST SCHREIBE
             *                     DAS NAECHSTE BIT
092E 78      SZEIT MOV  A,B       *CODE NACH AC
092F D308          OUT  HLAMP     *SCHREIBE HALBW.
0931 3E0C    ZSCH  MVI  A,SWAI    *HALBWELLENZEIT
             *                     LADEN
0933 3D            DCR  A         *ZAEHLE ZEIT AB
0934 C23309        JNZ  ZSCH+2    *ZEITSCHLEIFE
0937 04            INR B          *CODEWORT -1
0938 C9            RET            *RUECKSPRUNG

0939 0E80    LBLK  MVI  C,X'80'   *MSB=1 SETZEN
093B 1E16          MVI  E,SYN     *SYN-ZEICHEN LADEN
093D CD8709        CALL LBIT      *1 BIT EINLESEN
0940 79            MOV  A,C       *MSB=1 NACH AC
0941 BB            CMP  E         *VERGL., OB BISHER
             *                     EINGELESENE BITS
             *                     "SYN" ERGEBEN
0942 CA4D09        JZ   LSYN+2    *WENN JA, DANN
             *                     ZUM BYTE-LESEN
0945 F601    LSTB  ORI  1         *LSB=1 ALS STOP-
             *                     BIT SETZEN
0947 4F            MOV  C,A       *LSB=1 NACH C, UM
0948 C33B09        JMP  LBLK+2    *NAECHSTES BIT ZU
             *                     LESEN
094B 1602    LSYN  MVI  D,STX     *STX-ZEICHEN LADEN
094D CD7909        CALL LBYTE     *BYTE EINLESEN
0950 79            MOV  A,C       *BYTE NACH AC
0951 BB            CMP  E         *AUF "SYN" VERGL.
0952 CA4B09        JZ   LSYN      *WENN NOCH "SYN",
             *                     DANN WEITER AUF
             *                     "STX" WARTEN
0955 BA            CMP  D         *AUF "STX" VERGL.
0956 C24509        JNZ  LSTB      *WENN NICHT "STX",
             *                     DANN SYNCHR. VER-
             *                     LOREN, ZU LSTB
0959 CD7909        CALL LBYTE     *BYTE EINLESEN
             *                     (MUSS H-ADR SEIN)
095C 61            MOV  H,C       *H-ADR NACH H
095D 2E00          MVI  L,0       *L-ADR= 0 SETZEN
095F 16FF          MVI  D,255     *BLOCK LAENGE LADEN
0961 CD7909        CALL LBYTE     *1. DATA-BYTE LESEN
0964 71            MOV  M,C       *ABSPEICHERN
0965 23            INX  H         *ADR AUF NAECHSTEN
```

```
                *                       RAM-PLATZ
0966 CD7909 LDAT    CALL LBYTE          *DATA-BYTE LESEN
0969 71             MOV  M,C            *ABSPEICHERN
096A 23             INX  H              *ADR ERHOEHEN
096B 15             DCR  D              *BLOCKZ. -1
096C C26609         JNZ  LDAT           *NAECHSTES BYTE,
                *                       WENN BLOCK ZU ENDE
096F CD7909         CALL LBYTE          *DANN LIES BYTE
0972 79             MOV  A,C            *UND VERGLEICHE,
0973 D603           SUI  ETX            *OB "ETX", WENN JA
                *                       WIRD Z-FLAG=1 UND
0975 37             STC                 *SETZE C-FLAG=1,
                *                       D.H. BLOCK RICH-
                *                       TIG GELESEN
0976 C3E909         JMP  SHEND          *FERTIG

                *       SUBROUTINEN FUER LESEN
                *       DIESE SR'S SIND NICHT MEHR EIN-
                *       ZELN KOMMENTIERT, DA DAS SEHR
                *       SCHWER VERSTAENDLICH IST.
                *SIE LESEN DIE DEMODULIERTE RECHTECK-
                *SCHWINGUNG EIN, STELLEN DEREN LAENGE
                *DURCH AUSZAEHLEN FEST, D.H. DAMIT WIRD
                *ENTSCHIEDEN, OB DAS GELESENE BIT LANGE
                *ZEIT "1" UND KURZE ZEIT "0" (BIT=1)
                *ODER KURZE ZEIT "1" UND LANGE ZEIT "0"
                *(BIT=0) IST. DIE EINZELBITS WERDEN DANN
                *ZU BYTES ZUSAMMENGESETZT UND VOM LESE-
                *HAUPTPROGRAMM ENTSPRECHEND VERARBEITET.

0979 0E80   LBYTE   MVI  C,X'80'
097B DB08           IN   HSCH
097D 0F             RRC
097E DA7B09         JC   LBYTE+2
0981 CDDD09         CALL LWAR-2
0984 DA7B09         JC   LBYTE+2
0987 CDD009 LBIT    CALL LSFL
098A 3E01           MVI  A,1
098C 47             MOV  B,A
098D 80     LZAUF   ADD  B
098E 47             MOV  B,A
098F DAE709         JC   LSTR
0992 3E08           MVI  A,LZEIT+1
0994 CDDF09 LNXT    CALL LWAR
0997 3E01           MVI  A,1
0999 DA8D09         JC   LZAUF
099C 00             NOP
099D CDDD09         CALL LWAR-2
09A0 3E02           MVI  A,2
09A2 DA8D09         JC   LZAUF
09A5 3EFE           MVI  A,-2
09A7 80     LZAB    ADD  B
09A8 47             MOV  B,A
09A9 D2CA09         JNC  LNUL
09AC 3E08           MVI  A,LZEIT+1
09AE CDDF09         CALL LWAR
09B1 3EFF           MVI  A,-1
```

MICROKIT ASSEMBLER -- VER 2.2

```
09B3 D2A709          JNC  LZAB
09B6 00              NOP
09B7 CDDD09          CALL LWAR-2
09BA 3EFE            MVI  A,-2
09BC D2A709          JNC  LZAB
09BF 79              MOV  A,C
09C0 1F              RAR
09C1 4F              MOV  C,A
09C2 D8              RC
09C3 0602            MVI  B,2
09C5 3E07            MVI  A,LZEIT
09C7 C39409          JMP  LNXT

09CA 79      LNUL    MOV  A,C
09CB 1F              RAR
09CC 4F              MOV  C,A
09CD D28709          JNC  LBIT
09D0 DB08    LSFL    IN   HSCH
09D2 0F              RRC
09D3 D2D009          JNC  LSFL
09D6 CDDD09          CALL LWAR-2
09D9 D2D009          JNC  LSFL
09DC C9              RET

09DD 3E09            MVI  A,LZEIT+2
09DF 3D      LWAR    DCR  A
09E0 C2DF09          JNZ  LWAR
09E3 DB08            IN   HSCH
09E5 0F              RRC
09E6 C9              RET

09E7 AF      LSTR    XRA  A
09E8 C9              RET

09E9 F5      SHEND   PUSH PSW        *VERZOEGERUNGS-
09EA 110000          LXI  D,0        *SCHLEIFE FUER
09ED 7E      SHLOP   MOV  A,M        *3 SEC, FUER
09EE DB08            IN   HSCH       *BESSERE BEOBACH-
09F0 47              MOV  B,A        *TUNG DER FLAGS
09F1 1B              DCX  D          *UND AUF DEUT-
09F2 7A              MOV  A,D        *LICHES
09F3 B3              ORA  E          *ERKENNEN DER BE-
09F4 C2ED09          JNZ  SHLOP      *ENDIGUNG VON
09F7 F1              POP  PSW        *"SHIFT".
09F8 DF              RST  3,SRCKJP   *ABFRAGE, OB
09F9 50              DC   TSMS       *TASTEN "SHIFT"
09FA 00              DC   TSNO       *OD. "CAS" NOCH
09FB 0E08            DC   B(BREAK7)  *GEDRUECKT SIND,
             *                       WENN JA, DANN
             *                       WEITERE 3 SEC
             *                       VERZOEGERUNG,
09FD C3E909          JMP  SHEND      *SONST ZUM MONITOR

             ***     NO ERRORS       D.ULRICH RP/KTEW
0A00                 END  MON7
```