

# APPENDIX F

## LISTING OF THE SYSTEM MACRO FILE (SYSMAC.SML)

```
.MACRO .PARAM
R0=%A00
R1=%A01
R2=%A02
R3=%A03
R4=%A04
R5=%A05
R6=%A06
R7=%A07
SP=%A06
PC=%A07
PSW=%A0177776
SWR=%A0177570
    .ENDM
    .MACRO .INIT .LBLCK
    .MCALL .AMODE
    .AMODE .LBLCK
    EMT <A06>
    .ENDM

    .MACRO .RLSE .LBLCK
    .MCALL .AMODE
    .AMODE .LBLCK
    EMT <A07>
    .ENDM

    .MACRO .CLOSE .LBLCK
    .MCALL .AMODE
    .AMODE .LBLCK
    EMT <A017>
    .ENDM

    .MACRO .READ .LBLCK, .LBUFF
    .MCALL .AMODE
    .AMODE .LBUFF
    .AMODE .LBLCK
    EMT <A04>
    .ENDM

    .MACRO .WRITE .LBLCK, .LBUFF
    .MCALL .AMODE
    .AMODE .LBUFF
    .AMODE .LBLCK
    EMT <A02>
    .ENDM

    .MACRO .OPENO .LBLCK, .FBLCK
    .MCALL .CODE, .OPEN
    .CODE .FBLCK, <A02>
    .OPEN .LBLCK, .FBLCK
    .ENDM
```

```
.MACRO .OPENI .LBLCK,.FBLCK
.MCALL .CODE,.OPEN
.CODE .FBLCK,<^04>
.OPEN .LBLCK,.FBLCK
.ENDM
```

```
.MACRO .OPENU .LBLCK,.FBLCK
.MCALL .CODE,.OPEN
.CODE .FBLCK,<^01>
.OPEN .LBLCK,.FBLCK
.ENDM
```

```
.MACRO .OPENC .LBLCK,.FBLCK
.MCALL .CODE,.OPEN
.CODE .FBLCK,<^013>
.OPEN .LBLCK,.FBLCK
.ENDM
```

```
.MACRO .OPENE .LBLCK,.FBLCK
.MCALL .CODE,.OPEN
.CODE .FBLCK,<^03>
.OPEN .LBLCK,.FBLCK
.ENDM
```

```
.MACRO .OPEN .LBLCK,.FBLCK
.MCALL .AMODE
.AMODE .FBLCK
.AMODE .LBLCK
EMT <^016>
.ENDM
```

```
.MACRO .WAIT .LBLCK
.MCALL .AMODE
.AMODE .LBLCK
EMT <^01>
.ENDM
```

```
.MACRO .WAITR .LBLCK,.ADDR
.MCALL .AMODE
.AMODE .ADDR
.AMODE .LBLCK
EMT <^00>
.ENDM
```

```
.MACRO .BLOCK .LBLCK,.BBLCK
.MCALL .AMODE
.AMODE .BBLCK
.AMODE .LBLCK
EMT <^011>
.ENDM
```

```
.MACRO .TRAN .LBLCK,.TBLCK
.MCALL .AMODE
.AMODE .TBLCK
.AMODE .LBLCK
EMT <^010>
.ENDM
```

```
.MACRO .SPEC .LBLCK, .SARG
.MCALL .AMODE
.AMODE .SARG
.AMODE .LBLCK
EMT <^012>
.ENDM
```

```
.MACRO .STAT .LBLCK
.MCALL .AMODE
.AMODE .LBLCK
EMT <^013>
.ENDM
```

```
.MACRO .ALLOC .LBLCK, .FBLOCK, .N
.MCALL .AMODE
.AMODE .N
.AMODE .FBLOCK
.AMODE .LBLCK
EMT <^015>
.ENDM
```

```
.MACRO .DELET .LBLCK, .FBLOCK
.MCALL .AMODE
.AMODE .FBLOCK
.AMODE .LBLCK
EMT <^021>
.ENDM
```

```
.MACRO .RENAM .LBLCK, .OFB, .NFB
.MCALL .AMODE
.AMODE .NFB
.AMODE .OFB
.AMODE .LBLCK
EMT <^020>
.ENDM
```

```
.MACRO .APPND .LBLCK, .1FB, .2FB
.MCALL .AMODE
.AMODE .2FB
.AMODE .1FB
.AMODE .LBLCK
EMT <^022>
.ENDM
```

```
.MACRO .LOOK .LBLCK, .FBLOCK, .OP
.MCALL .AMODE
.AMODE .FBLOCK
.IIF NB, .OP, CLR -(SP)
.AMODE .LBLCK
EMT <^014>
.ENDM
```

```
.MACRO .KEEP .LBLCK, .FBLOCK
.MCALL .AMODE
.AMODE .FBLOCK
.AMODE .LBLCK
EMT <^024>
.ENDM
```

```
.MACRO .EXIT
EMT <^060>
.ENDM
```

```
.MACRO .TRAP .STUS,.ADDR
.MCALL .AMODE
.AMODE .ADDR
.AMODE .STUS
MOV #A01,-(SP)
EMT <A041>
.ENDM
```

```
.MACRO .STFPU .STUS,.ADDR
.MCALL .AMODE
.AMODE .ADDR
.AMODE .STUS
MOV #A03,-(SP)
EMT <A041>
.ENDM
```

```
.MACRO .RECRD .LBLCK,.RBLCK
.MCALL .AMODE
.AMODE .RBLCK
.AMODE .LBLCK
EMT <A025>
.ENDM
```

```
.MACRO .DUMP .LOW,.HIGH,.CDE
.MCALL .AMODE
.AMODE .LOW
.AMODE .HIGH
.AMODE .CDE
EMT <A064>
.ENDM
```

```
.MACRO .RSTRT .ADDR
.MCALL .AMODE
.AMODE .ADDR
MOV #A02,-(SP)
```

```
EMT <A041>
.ENDM
```

```
.MACRO .CORE
MOV #A0100,-(SP)
EMT <A041>
.ENDM
```

```
.MACRO .MONR
MOV #A0101,-(SP)
EMT <A041>
.ENDM
```

```
.MACRO .MONF
MOV #A0102,-(SP)
EMT <A041>
.ENDM
```

```
.MACRO .DATE
MOV #A0103,-(SP)
EMT <A041>
.ENDM
```

```
.MACRO .TIME
MOV #A0104,-(SP)
EMT <A041>
.ENDM
```

```
.MACRO .GTUIC
MOV #A0105,-(SP)
EMT <A041>
.ENDM
```

```
.MACRO .SYSDV
MOV #A0106,-(SP)
EMT <A041>
.ENDM
```

```
.MACRO .RADPK .ADDR
.MCALL .AMODE
.AMODE .ADDR
CLR -(SP)
EMT <A042>
.ENDM
```

```
.MACRO .RADUP .ADDR,.WRD
.MCALL .AMODE
.AMODE .WRD
.AMODE .ADDR
MOV #A01,-(SP)
EMT <A042>
.ENDM
```

```
.MACRO .D2BIN .ADDR
.MCALL .AMODE
.AMODE .ADDR
MOV #A02,-(SP)
EMT <A042>
.ENDM
```

```
.MACRO .BIN2D .ADDR,.WRD
.MCALL .AMODE
.AMODE .WRD
.AMODE .ADDR
MOV #A03,-(SP)
EMT <A042>
.ENDM
```

```
.MACRO .O2BIN .ADDR
.MCALL .AMODE
.AMODE .ADDR
MOV #A04,-(SP)
EMT <A042>
.ENDM
```

```
.MACRO .BIN2O .ADDR,.WRD
.MCALL .AMODE
.AMODE .WRD
.AMODE .ADDR
MOV #A05,-(SP)
EMT <A042>
.ENDM
```

```

.MACRO .CSI1 .CMDBF
.MCALL .AMODE
.AMODE .CMDBF
EMT <A056>
.ENDM

```

```

.MACRO .CSI2 .CSBLK
.MCALL .AMODE
.AMODE .CSBLK
EMT <A057>
.ENDM

```

```

.MACRO .DTCVT .ADDR
.MCALL .CVTDT
.CVTDT #A00, .ADDR
.ENDM

```

```

.MACRO .TMCVT .ADDR
.MCALL .CVTDT
.CVTDT #A01, .ADDR
.ENDM

```

```

.MACRO .CVTDT .CDE, .ADDR, .VAL1, .VAL2
.MCALL .AMODE
.IF NB, .VAL2
.AMODE .VAL2
.ENDC
.IF NB, .VAL1
.AMODE .VAL1
.ENDC
.AMODE .ADDR
.AMODE .CDE
EMT <A066>
.ENDM

```

```

.MACRO .GTPLA
CLR -(SP)
MOV #A05, -(SP)
EMT <A041>
.ENDM

```

```

.MACRO .STPLA .ADDR
.MCALL .AMODE
.AMODE .ADDR
MOV #A05, -(SP)
EMT <A041>
.ENDM

```

```

.MACRO .GTCIL
MOV #A0111, -(SP)
EMT <A041>
.ENDM

```

```

.MACRO .GTSTK
CLR -(SP)
MOV #A04, -(SP)
EMT <A041>
.ENDM

```

```

.MACRO .STSTK .ADDR
.MCALL .AMODE
.AMODE .ADDR

```

```

MOV     #A04,-(SP)
EMT <A041>
.ENDM

```

```

.MACRO  .RUN      .RNBLK
.MCALL  .AMODE
.AMODE  .RNBLK
EMT <A065>
.ENDM

```

```

.MACRO  .FLUSH   .CDE
.MCALL  .AMODE
.AMODE  .CDE
EMT <A067>
.ENDM

```

```

; THE MACRO .AMODE ACCEPTS ONE ARGUMENT AND
; AS A FUNCTION OF THE ADDRESSING MODE OF
; THE ARGUMENT GENERATES THE APPROPRIATE
; MOV TO -(SP).
; ADDRESS MODES THAT ARE TROUBLESOME (E.G.
; X(SP)) OR UNLIKELY (E.G. SP) WILL RESULT
; IN A .ERROR TO CMO INCLUDING THE
; VALUE OF THE ADDRESS MODE (E.G. X(SP)
; IS REPRESENTED AS 000066), THE ARGUMENT ITSELF
; AND THE TEXT "ADDRESSING MODE ILLEGAL AS SYSTEM
; MACRO ARGUMENT".
;

```

```

.MACRO  .AMODE  .ARG
SP=X^A06
.NTYPE  .SYM,.ARG      ;.SYM=ADDRESS MODE.

.IF LE,.SYM=A05
MOV     .ARG,-(SP)      ;R0 TO R5
.MEXIT
.ENDC

.IF EQ,.SYM&A070=A010
.IF LE,.SYM&A07=A06
MOV     .ARG,-(SP)      ;R0 TO R6
.MEXIT
.ENDC
.ENDC

.IF EQ,.SYM&A060=A020
MOV     .ARG,-(SP)      ;[R0]+ TO [R7]+
.MEXIT
.ENDC
; #N, #ADDR

.IF EQ,.SYM&A040=A040
.IF LE,.SYM&A07=A05
MOV     .ARG,-(SP)      ;[R0]-(R5) TO [R5]-(R5)
.MEXIT
.ENDC
; [R0]X(R0) TO [R5]X(R5)
.ENDC

.IF EQ,.SYM&A067=A067
MOV     .ARG,-(SP)      ;ADDR AND #ADDR
.MEXIT
.ENDC

```

```

        .ERROR .SYM          ;.ARG ADDRESSING MODE ILLEGAL
        .PRINT          ;AS SYSTEM MACRO ARGUMENT.
        .ENDM

; THE MACRO .CODE SETS UP THE FILEBLOCK
; WITH THE HOW OPEN CODE.
; THE ADDRESS OF THE FILEBLOCK MUST
; BE IN A REGISTER (R0 TO R5)

        .MACRO .CODE .FBLK,.N
        .NTYPE .SYM,.FBLK

        .IF LE,.SYM=^O5
        MOVB #.N,^A02(.FBLK) ;R0 TO R5
        .MEXIT
        .ENDC

        .ERROR .SYM          ;.FBLK ADDRESSING MODE ILLEGAL
        .PRINT          ;FOR .OPEN FILE BLOCK
        .ENDM

        .MACRO .F4DEF N
        .IF NB N
        .N
        .ENDC
.F4SEQ
.MCALL .MVMRB,.MVMRL,.MVMRI,.MVMRJ,.MVMRR,.MVMRD,.MVMRC
.MCALL .MVSRB,.MVSRL,.MVSRI,.MVS RJ,.MVSRR,.MVS RD,.MVSRC
.MCALL .MVR SB,.MVR SL,.MVR SI,.MVR SJ,.MVR SR,.MVR SD,.MVR SC
.MCALL .MVR MB,.MVR ML,.MVR MI,.MVR MJ,.MVR MR,.MVR MD,.MVR MC
        .MCALL F4RTN
        .IF EQ .F4SEQ
        .MCALL .F4OLD
        .F4OLD
        .MEXIT
        .ENDC
        .IF EQ <.F4SEQ-1>* <.F4SEQ-2>
        .MCALL .F4NEW
        .F4NEW
        .MEXIT
        .ENDC
        .ERROR N;FORTRAN CALL SEQUENCE DEFINITIONAL ERROR
        .ENDM

;MACRO TO DEFINE OLD FORM
        .MACRO .F4OLD
        .MACRO .F4RTS
        RTS X5
        .ENDM
        .MCALL .F4P1,.F4P2,F4VAL2
        .MACRO F4CALL SUB,ARGS,?LABEL
        .0
        .IRP X,<ARGS>
        .F4P1 LABEL,X
        .ENDM
        .JSR X5,SUB
        .BR .+2+SSL
        .IRP X,<ARGS>
        .F4P2 X
        .ENDM
        .ENDM
        .ENDM
        .ENDM

```

;MACRO TO DEFINE THE NEW FORM

```
.MACRO .F4NEW
.MACRO .F4RTS
RTS X7
.ENDM
.MCALL .F4P1,.F4P2,F4VAL2
.MACRO F4CALL SUB,ARGS,?LABEL
SSL =0
.IRP X,<ARGS>
.F4P1 LABEL,X
.ENDM
.IF NE .F4SEQ-2
MOV X5,-(X6)
.ENDC
MOV #LABEL,X5
JSR X7,SUB
BR .+4+SSL
LABEL: .BYTE SSL/2,0
.IRP X,<ARGS>
.F4P2 X
.ENDM
.IF NE .F4SEQ-2
MOV (X6)+,X5
.ENDC
.ENDM
.ENDM
```

;MACRO TO BUILD THE DYNAMIC PART OF ARG LIST

```
.MACRO .F4P1 L,X,Y
.NARG $SN
.IF EQ $SN-3
MOV X,L+2+SSL
.IF NE Y
ADD #Y,L+2+SSL
.ENDC
SSL =SSL+2
.MEXIT
.ENDC
.IF NE $SN-2
.ERROR ,BAD F4CALL ARG
.MEXIT
.ENDC
.IF B X
SSL =SSL+2
.MEXIT
.ENDC
.NTYPE $SAM,X
.IF NE $SAM-^067
MOV X,L+2+SSL
.ENDC
SSL =SSL+2
.ENDM
```



**;MACROS TO MOVE FROM MEMORY TO REGISTERS**

**;BYTE**

```
.MACRO .MVMRB A
MOV    A,%0
.ENDM
```

**;LOGICAL**

```
.MACRO .MVMRL A
MOV    A,%0
.ENDM
```

**;INTEGER**

```
.MACRO .MVMRI A
MOV    A,%0
.ENDM
```

**;DOUBLE INTEGER**

```
.MACRO .MVMRJ A
MOV    A,%0
MOV    A+2,%1
.ENDM
```

**;REAL**

```
.MACRO .MVMRR A
MOV    A,%0
MOV    2+A,%1
.ENDM
```

**;DOUBLE REAL**

```
.MACRO .MVMRD A
MOV    A,%0
MOV    2+A,%1
MOV    4+A,%2
MOV    6+A,%3
.ENDM
```

**;COMPLEX**

```
.MACRO .MVMRC A
.MVMRD A
.ENDM
```

**;MACROS TO MOVE FROM STACK TO REGISTERS**

**;BYTE**

```
.MACRO .MVS RB A
MOV    A,%0
.ENDM
```

**;LOGICAL**

```
.MACRO .MVSRL A
MOV    A,%0
.ENDM
```

**;INTEGER**

```
.MACRO .MVSRI A
MOV    A,%0
.ENDM
```

**;DOUBLE INTEGER**

```
.MACRO .MVS RJ A
MOV    A,%0
MOV    A,%1
.ENDM
```

**;REAL**

```
.MACRO .MVS RR A
MOV    A,%0
MOV    A,%1
.ENDM
```

```

;DOUBLE REAL
  .MACRO      .MVSRD  A
    MOV      A,%0
    MOV      A,%1
    MOV      A,%2
    MOV      A,%3
  .ENDM

```

```

;COMPLEX
  .MACRO      .MVSRC  A
    .MVSRD  A
  .ENDM

```

```

;MACROS TO MOVE FROM REGISTERS TO STACK
;BYTE

```

```

  .MACRO      .MVRSB  A
    MOV      %0,A
  .ENDM

```

```

;LOGICAL
  .MACRO      .MVRSL  A
    MOV      %0,A
  .ENDM

```

```

;INTEGER
  .MACRO      .MVRSI  A
    MOV      %0,A
  .ENDM

```

```

;DOUBLE INTEGER
  .MACRO      .MVRSJ  A
    MOV      %1,A
    MOV      %0,A
  .ENDM

```

```

;REAL
  .MACRO      .MVRSR  A
    MOV      %1,A
    MOV      %0,A
  .ENDM

```

```

;DOUBLE REAL
  .MACRO      .MVRSD  A
    MOV      %3,A
    MOV      %2,A
    MOV      %1,A
    MOV      %0,A
  .ENDM

```

```

;COMPLEX
  .MACRO      .MVRSC  A
    .MVRSD  A
  .ENDM

```

;MACROS TO MOVE FROM REGISTERS TO MEMORY

```
;BYTE
      .MACRO  .MVRMB  A
      MOV     %0,A
      .ENDM
```

```
;LOGICAL
      .MACRO  .MVRML  A
      MOV     %0,A
      .ENDM
```

```
;INTEGER
      .MACRO  .MVRMI  A
      MOV     %0,A
      .ENDM
```

```
;DOUBLE INTEGER
      .MACRO  .MVRMJ  A
      MOV     %0,A
      MOV     %1,A+2
      .ENDM
```

```
;REAL
      .MACRO  .MVRMR  A
      MOV     %0,A
      MOV     %1,2+A
      .ENDM
```

```
;DOUBLE REAL
      .MACRO  .MVRMD  A
      MOV     %0,A
      MOV     %1,2+A
      MOV     %2,4+A
      MOV     %3,6+A
      .ENDM
```

```
;COMPLEX
      .MACRO  .MVRMC  A
      .MVRMD  A
      .ENDM
```

```
      .MACRO  .GTRDV
      CLR     -(SP)
      CLR     -(SP)
      MOV     #112,-(SP)
      EMT     41
      .ENDM
```

```
      .MACRO  .GTCLK
      MOV     #113,-(SP)
      EMT     41
      .ENDM
```

```
      .MACRO  .GTOVF
      MOV     #114,-(SP)
      EMT     41
      .ENDM  .GTOVF
```

