APPENDIX F

LISTING OF THE SYSTEM MACRO FILE (SYSMAC.SML)

```
        .MACRO  .PARAM
R0=%^00
R1=%^01
R2=%^02
R3=%^03
R4=%^04
R5=%^05
R6=%^06
R7=%^07
SP=%^06
PC=%^07
PSW=^0177776
SWR=^0177570
        .ENDM
        .MACRO  .INIT   .LBLCK
        .MCALL  .AMODE
        .AMODE  .LBLCK
        EMT <^06>
        .ENDM

        .MACRO  .RLSE   .LBLCK
        .MCALL  .AMODE
        .AMODE  .LBLCK
        EMT <^07>
        .ENDM

        .MACRO  .CLOSE  .LBLCK
        .MCALL  .AMODE
        .AMODE  .LBLCK
        EMT <^017>
        .ENDM

        .MACRO  .READ   .LBLCK,.LBUFF
        .MCALL  .AMODE
        .AMODE  .LBUFF
        .AMODE  .LBLCK
        EMT <^04>
        .ENDM

        .MACRO  .WRITE  .LBLCK,.LBUFF
        .MCALL  .AMODE
        .AMODE  .LBUFF
        .AMODE  .LBLCK
        EMT <^02>
        .ENDM

        .MACRO  .OPENO  .LBLCK,.FBLCK
        .MCALL  .CODE,.OPEN
        .CODE   .FBLCK,<^02>
        .OPEN   .LBLCK,.FBLCK
        .ENDM
```

```
.MACRO  .OPENI  .LBLCK,.FBLCK
.MCALL  .CODE,.OPEN
.CODE   .FBLCK,<^O04>
.OPEN   .LBLCK,.FBLCK
.ENDM

.MACRO  .OPENU  .LBLCK,.FBLCK
.MCALL  .CODE,.OPEN
.CODE   .FBLCK,<^O01>
.OPEN   .LBLCK,.FBLCK
.ENDM


.MACRO  .OPENC  .LBLCK,.FBLCK
.MCALL  .CODE,.OPEN
.CODE   .FBLCK,<^O13>
.OPEN   .LBLCK,.FBLCK
.ENDM

.MACRO  .OPENE  .LBLCK,.FBLCK
.MCALL  .CODE,.OPEN
.CODE   .FBLCK,<^O03>
.OPEN   .LBLCK,.FBLCK
.ENDM

.MACRO  .OPEN   .LBLCK,.FBLCK
.MCALL  .AMODE
.AMODE  .FBLCK
.AMODE  .LBLCK
EMT <^O16>
.ENDM

.MACRO  .WAIT   .LBLCK
.MCALL  .AMODE
.AMODE  .LBLCK
EMT <^O1>
.ENDM

.MACRO  .WAITR  .LBLCK,.ADDR
.MCALL  .AMODE
.AMODE  .ADDR
.AMODE  .LBLCK
EMT <^O0>
.ENDM

.MACRO  .BLOCK  .LBLCK,.BBLCK
.MCALL  .AMODE
.AMODE  .BBLCK
.AMODE  .LBLCK
EMT <^O11>
.ENDM

.MACRO  .TRAN   .LBLCK,.TBLCK
.MCALL  .AMODE
.AMODE  .TBLCK
.AMODE  .LBLCK
EMT <^O10>
.ENDM
```

```
.MACRO  .SPEC   .LBLCK,.SARG
.MCALL  .AMODE
.AMODE  .SARG
.AMODE  .LBLCK
EMT <^O12>
.ENDM

.MACRO  .STAT   .LBLCK
.MCALL  .AMODE
.AMODE  .LBLCK
EMT <^O13>
.ENDM

.MACRO  .ALLOC  .LBLCK,.FBLCK,.N
.MCALL  .AMODE
.AMODE  .N
.AMODE  .FBLCK
.AMODE  .LBLCK
EMT <^O15>
.ENDM

.MACRO  .DELET  .LBLCK,.FBLCK
.MCALL  .AMODE
.AMODE  .FBLCK
.AMODE  .LBLCK
EMT <^O21>
.ENDM

.MACRO  .RENAM  .LBLCK,.OFB,.NFB
.MCALL  .AMODE
.AMODE  .NFB
.AMODE  .OFB
.AMODE  .LBLCK
EMT <^O20>
.ENDM

.MACRO  .APPND  .LBLCK,.1FB,.2FB
.MCALL  .AMODE
.AMODE  .2FB
.AMODE  .1FB
.AMODE  .LBLCK
EMT <^O22>
.ENDM

.MACRO  .LOOK   .LBLCK,.FBLCK,.OP
.MCALL  .AMODE
.AMODE  .FBLCK
.IIF NB,.OP,CLR -(SP)
.AMODE  .LBLCK
EMT <^O14>
.ENDM

.MACRO  .KEEP   .LBLCK,.FBLCK
.MCALL  .AMODE
.AMODE  .FBLCK
.AMODE  .LBLCK
EMT <^O24>
.ENDM

.MACRO  .EXIT
EMT <^O60>
.ENDM
```

```
.MACRO  .TRAP   .STUS,.ADDR
.MCALL  .AMODE
.AMODE  .ADDR
.AMODE  .STUS
MOV     #^O1,-(SP)
EMT <^O41>
.ENDM

.MACRO  .STFPU  .STUS,.ADDR
.MCALL  .AMODE
.AMODE  .ADDR
.AMODE  .STUS
MOV     #^O3,-(SP)
EMT <^O41>
.ENDM

.MACRO  .RECRD  .LBLCK,.RBLCK
.MCALL  .AMODE
.AMODE  .RBLCK
.AMODE  .LBLCK
EMT <^O25>
.ENDM

.MACRO  .DUMP   .LOW,.HIGH,.CDE
.MCALL  .AMODE
.AMODE  .LOW
.AMODE  .HIGH
.AMODE  .CDE
EMT <^O64>
.ENDM

.MACRO  .RSTRT  .ADDR
.MCALL  .AMODE
.AMODE  .ADDR
MOV     #^O2,-(SP)



EMT <^O41>
.ENDM

.MACRO  .CORE
MOV     #^O100,-(SP)
EMT <^O41>
.ENDM

.MACRO  .MONR
MOV     #^O101,-(SP)
EMT <^O41>
.ENDM

.MACRO  .MONF
MOV     #^O102,-(SP)
EMT <^O41>
.ENDM

.MACRO  .DATE
MOV     #^O103,-(SP)
EMT <^O41>
.ENDM
```

```
.MACRO  .TIME
MOV     #^O104,-(SP)
EMT <^O41>
.ENDM

.MACRO  .GTUIC
MOV     #^O105,-(SP)
EMT <^O41>
.ENDM

.MACRO  .SYSDV
MOV     #^O106,-(SP)
EMT <^O41>
.ENDM

.MACRO  .RADPK  .ADDR
.MCALL  .AMODE
.AMODE  .ADDR
CLR     -(SP)
EMT <^O42>
.ENDM


.MACRO  .RADUP  .ADDR,.WRD
.MCALL  .AMODE
.AMODE  .WRD
.AMODE  .ADDR
MOV     #^O1,-(SP)
EMT <^O42>
.ENDM

.MACRO  .D2BIN  .ADDR
.MCALL  .AMODE
.AMODE  .ADDR
MOV     #^O2,-(SP)
EMT <^O42>
.ENDM

.MACRO  .BIN2D  .ADDR,.WRD
.MCALL  .AMODE
.AMODE  .WRD
.AMODE  .ADDR
MOV     #^O3,-(SP)
EMT <^O42>
.ENDM

.MACRO  .O2BIN  .ADDR
.MCALL  .AMODE
.AMODE  .ADDR
MOV     #^O4,-(SP)
EMT <^O42>
.ENDM

.MACRO  .BIN2O  .ADDR,.WRD
.MCALL  .AMODE
.AMODE  .WRD
.AMODE  .ADDR
MOV     #^O5,-(SP)
EMT <^O42>
.ENDM
```

```
.MACRO  .CSI1   .CMDBF
.MCALL  .AMODE
.AMODE  .CMDBF
EMT <^O56>
.ENDM

.MACRO  .CSI2   .CSBLK
.MCALL  .AMODE
.AMODE  .CSBLK
EMT <^O57>
.ENDM

.MACRO  .DTCVT  .ADDR
.MCALL  .CVTDT
.CVTDT  #^O0,.ADDR
.ENDM

.MACRO  .TMCVT  .ADDR
.MCALL  .CVTDT
.CVTDT  #^O1,.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 <^O66>
.ENDM

.MACRO  .GTPLA
CLR     -(SP)
MOV     #^O5,-(SP)
EMT <^O41>
.ENDM

.MACRO  .STPLA  .ADDR
.MCALL  .AMODE
.AMODE  .ADDR
MOV     #^O5,-(SP)
EMT <^O41>
.ENDM

.MACRO  .GTCIL
MOV     #^O111,-(SP)
EMT <^O41>
.ENDM

.MACRO  .GTSTK
CLR     -(SP)
MOV     #^O4,-(SP)
EMT <^O41>
.ENDM

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

```
        MOV     #^O4,-(SP)
        EMT <^O41>
        .ENDM

        .MACRO  .RUN    .RNBLK
        .MCALL  .AMODE
        .AMODE  .RNBLK
        EMT <^O65>
        .ENDM

        .MACRO  .FLUSH  .CDE
        .MCALL  .AMODE
        .AMODE  .CDE
        EMT <^O67>
        .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=%^O6
        .NTYPE  .SYM,.ARG       ;.SYM=ADDRESS MODE.

        .IF LE,.SYM-^O5
        MOV     .ARG,-(SP)      ;R0 TO R5
        .MEXIT
        .ENDC

        .IF EQ,.SYM&^O70-^O10
        .IF LE,.SYM&^O7-^O6
        MOV     .ARG,-(SP)      ;@R0 TO @R6
        .MEXIT
        .ENDC
        .ENDC

        .IF EQ,.SYM&^O60-^O20
        MOV     .ARG,-(SP)      ;[@](R0)+ TO [@](R7)+
        .MEXIT                  ; #N,@#ADDR
        .ENDC

        .IF EQ,.SYM&^O40-^O40
        .IF LE,.SYM&^O7-^O5
        MOV     .ARG,-(SP)      ;[@]-(R0) TO [@]-(R5)
        .MEXIT                  ;[@]X(R0) TO [@]X(R5)
        .ENDC
        .ENDC

        .IF EQ,.SYM&^O67-^O67
        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,-^O2(.FBLK) ;R0 TO R5
        .MEXIT
        .ENDC

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


        .MACRO  .F4DEF N
        .IF NB  N
.F4SEQ  =N
        .ENDC
.MCALL  .MVMRB,.MVMRL,.MVMRI,.MVMRJ,.MVMRR,.MVMRD,.MVMRC
.MCALL  .MVSRB,.MVSRL,.MVSRI,.MVSRJ,.MVSRR,.MVSRD,.MVSRC
.MCALL  .MVRSB,.MVRSL,.MVRSI,.MVRSJ,.MVRSR,.MVRSD,.MVRSC
.MCALL  .MVRMB,.MVRML,.MVRMI,.MVRMJ,.MVRMR,.MVRMD,.MVRMC
        .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     %5
        .ENDM
        .MCALL  .F4P1,.F4P2,F4VAL2
        .MACRO  F4CALL  SUB,ARGS,?LABEL
$$L     =0
        .IRP    X,<ARGS>
        .F4P1   LABEL,X
        .ENDM
        JSR     %5,SUB
LABEL:  BR      .+2+$$L
        .IRP    X,<ARGS>
        .F4P2   X
        .ENDM
        .ENDM
        .ENDM
```

```
;MACRO TO DEFINE THE NEW FORM
        .MACRO  .F4NEW
        .MACRO  .F4RTS
        RTS     %7
        .ENDM
        .MCALL  .F4P1,.F4P2,F4VAL2
        .MACRO  F4CALL  SUB,ARGS,?LABEL
$$L     =0
        .IRP    X,<ARGS>
        .F4P1   LABEL,X
        .ENDM
        .IF NE  .F4SEQ-2
        MOV     %5,-(%6)
        .ENDC
        MOV     #LABEL,%5
        JSR     %7,SUB
        BR      .+4+$$L
LABEL:  .BYTE   $$L/2,0
        .IRP    X,<ARGS>
        .F4P2   X
        .ENDM
        .IF NE  .F4SEQ-2
        MOV     (%6)+,%5
        .ENDC
        .ENDM
        .ENDM


;MACRO TO BUILD THE DYNAMIC PART OF ARG LIST
        .MACRO  .F4P1   L,X,Y
        .NARG   $$N
        .IF EQ  $$N-3
        MOV     X,L+2+$$L
        .IF NE  Y
        ADD     #Y,L+2+$$L
        .ENDC
$$L     =$$L+2
        .MEXIT
        .ENDC
        .IF NE  $$N-2
        .ERROR  ;BAD F4CALL ARG
        .MEXIT
        .ENDC
        .IF B   X
$$L     =$$L+2
        .MEXIT
        .ENDC
        .NTYPE  $$AM,X
        .IF NE  $$AM-^O67
        MOV     X,L+2+$$L
        .ENDC
$$L     =$$L+2
        .ENDM
```

```
;MACRO TO DO STATIC PART OF ARG LIST
        .MACRO  .F4P2   X,Y
        .IF B   X
        .WORD   -1
        .MEXIT
        .ENDC
        .IF NB  Y
        .WORD   -1
        .MEXIT
        .ENDC
        .NTYPE  $$AM,X
        .IF EQ  $$AM-^O67
        .WORD   X
        .IFF
        .WORD   -1
        .ENDC
        .ENDM

;MACRO TO DO FORTRAN RETURNS
        .MACRO  F4RTN   T,L
        .IF NB  T
        .IF B   L
        .ERROR  ;MISSING ARGUMENT
        .ENDC
        .ENDC
        .IF NB  T
        .IF NB  L
        .NTYPE  $$AM,L
        .IF EQ  $$AM-↑O67
        .MVMR'T L
        .IFF
        .IF EQ  $$AM&↑O7Ø-↑O2Ø
        .IF EQ  $$AM-↑O27
        .ERROR  $$AM;BAD ADDRESS MODE
        .IFF
        .MVSR'T L
        .ENDC
        .IFF
        .ERROR  $$AM;BAD ADDRESS MODE
        .ENDC
        .ENDC
        .ENDC
        .ENDC
        .F4RTS
        .ENDM

;MACRO TO DO FORTRAN VALUE STORE
        .MACRO  F4VAL2  T,L
        .NTYPE  $$AM,L
        .IF EQ  $$AM&^O7Ø-^O4Ø
        .MVRS'T L
        .MEXIT
        .ENDC
        .IF EQ  $$AM-^O67
        .MVRM'T L
        .MEXIT
        .ENDC
        .ERROR  $$AM;BAD ADDRESS MODE
        .ENDM
```

```
;MACROS TO MOVE FROM MEMORY TO REGISTERS
;BYTE
        .MACRO  .MVMRB  A
        MOVB    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  .MVSRB  A
        MOVB    A,%0
        .ENDM
;LOGICAL
        .MACRO  .MVSRL  A
        MOV     A,%0
        .ENDM
;INTEGER
        .MACRO  .MVSRI  A
        MOV     A,%0
        .ENDM
;DOUBLE INTEGER
        .MACRO  .MVSRJ A
        MOV     A,%0
        MOV     A,%1
        .ENDM
;REAL
        .MACRO  .MVSRR  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
        MOVB    %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
        MOVB    %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
```