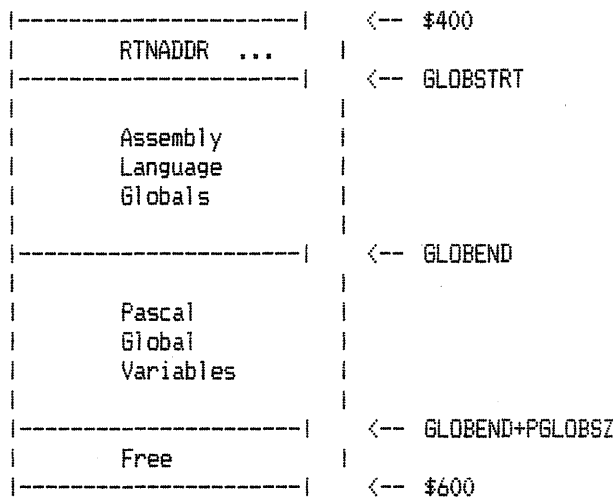
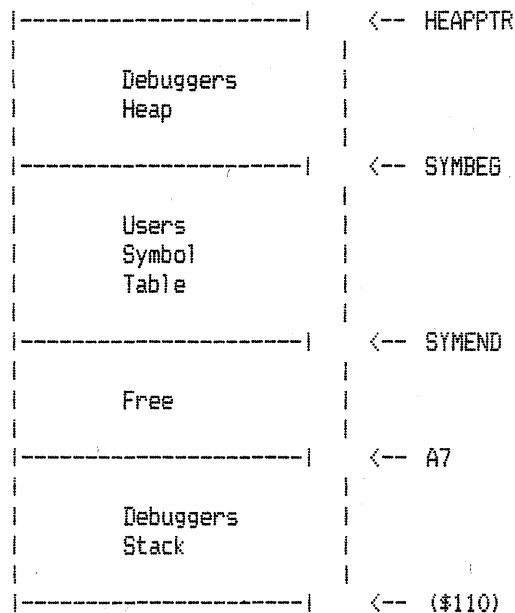


```
FILENAME: DBUG EQU
```

```
Low Memory (ie $400 -- $600) is as follows:
```



```
Debuggers data area below the screen
```



```
DATASIZ EQU 4096 ; size of debuggers data area below the screen
```

```
; Assembly Language Globals Initialized by STARTUP
```

```

RTNADDR EQU $0400 ; return address for talk
TRAPSR EQU RTNADDR+4 ; value to set into SR on next trap
SYMBBEG EQU TRAPSR+2 ; ptr to symbol table begin
SYMBEND EQU SYMBBEG+4 ; ptr to symbol table end

```

```
; Assembly Language Globals Zeroed by STARTUP, Initialized by INITDEBUG
```

```

GLOBSTRT EQU SYMBEND+4 ; global area start
RUNFLAG EQU GLOBSTRT ; running the user boolean

```

```

GO2DBUG EQU RUNFLAG+2 ; indicates we trapped in user
NOTALK EQU GO2DBUG+2 ; indicates no talk program (since $120 was 0)
DBGMODE EQU NOTALK+1 ; (normal,tracing,ixtracing,assembling,heapfault)
USERREGS EQU DBGMODE+1 ; save area for user registers
TRAPNUMB EQU USERREGS+80 ; the short address passed along to the user
FCODE EQU TRAPNUMB+2 ; the function code
ACCADDR EQU FCODE+2 ; the access address
INSTR EQU ACCADDR+4 ; the instruction
USERA5 EQU INSTR+2 ; User A5 register recorded by the Loader
NEWCPU EQU USERA5+4 ; Feb82 CPU board
NEWIO EQU NEWCPU+1 ; Feb82 I/O board
MEMOFS EQU NEWIO+1 ; offset to start of memory
VIA2PTR EQU MEMOFS+4 ; contains @VIA2, note: @vial is at $134
IXAREA EQU VIA2PTR+4 ; 5 word save area for IX command
IXPTR EQU IXAREA+10 ; pointer to Pc-10 for IX command
SETVIDL EQU IXPTR+4 ; boolean, video page latch needs to be set
OLDSETT EQU SETVIDL+1 ; old screen setting, 0=primary, 1=alternate
MSBASE EQU OLDSETT+1 ; four bytes of screen control
UNUSED EQU MSBASE+4
JUMP3PTR EQU UNUSED+4 ; ptr to jump table for Lisabug3.obj
JUMP2PTR EQU JUMP3PTR+4 ; ptr to jump table for Lisabug2.obj
JUMP1PTR EQU JUMP2PTR+4 ; ptr to jump table for Lisabug.obj
MAINPC EQU JUMP1PTR+4 ; saved pc for procedure main
MAINSP EQU MAINPC+4 ; saved sp for procedure main
MAINA6 EQU MAINSP+4 ; saved A6 for procedure main
COPSFLLAG EQU MAINA6+4 ; Internally maintained flags
COPSBYTE EQU COPSFLLAG+1 ; Last byte read from COPS
TWGDONE EQU COPSFLLAG+2 ; Twiggy Read, Write or Format is complete
BPTINIT EQU TWGDONE+2 ; address of bpt initialize
BPTCREAT EQU BPTINIT+4 ; address of bpt create
BPTSET EQU BPTCREAT+4 ; address of bpt set
BPTCLEAR EQU BPTSET+4 ; address of bpt clear
ADDRDCT EQU BPTCLEAR+4 ; @array [0..3] of record p:longint; r:int end
ADDRBPT EQU ADDRDCT+4 ; @ANYBPT
HEAPPTR EQU ADDRBPT+4 ; address of heap base
TENTRY EQU HEAPPTR+4 ; time at entry to debugger
TEXTIT EQU TENTRY+4 ; time at exit from debugger
PRBASE EQU TEXTIT+4 ; printer base address
OLDATRAP EQU PRBASE+4 ; address of MacTrap dispatcher
ATRACEON EQU OLDATRAP+4 ; is MacTrap tracing enabled
ATRACESE EQU ATRACEON+2 ; old dispatch address has been saved
BLKMOVE EQU ATRACESE+2 ; address of Mac BlockMove routine
GLOBEND EQU BLKMOVE+4 ; global area end
;
BPTCOUNT EQU 0 ; number of break points set
BPTTABLE EQU BPTCOUNT+2 ; array [1..16] of record proc,addr,save end
PPTCOUNT EQU BPTTABLE+160 ; number of patch points set
PPTTABLE EQU PPTCOUNT+2 ; array [1..4] of record ins,save,dst end
RTNCOUNT EQU PPTTABLE+56 ; number of rtn addr in the return stack
RTNSTACK EQU RTNCOUNT+2 ; array [1..4] of record rtn addr end
TIMING EQU RTNSTACK+32 ; timing globals 14+10*20+20*1
HEAPEND EQU TIMING+234 ; heap area end
;
TOPBUCKT EQU TIMING+2 ; offset of topbuckt in timing record
;
CRTROW EQU 0 ;BYTE, ROW FOR SIMULATED CRT
CRTCOL EQU CRTROW+1 ;BYTE, COLUMN FOR SIMULATED CRT
CRTSTATE EQU CRTCOL+1 ;BYTE, STATE OF CRT SIMULATOR
CTLSFLAG EQU CRTSTATE+1 ;BYTE, COUNT OF THROWN AWAY CMND-PERIODS
;
SEGRES EQU $FCE008 ; base for set & reset of SEG1 & SEG2 bits
VIDLATCH EQU $FCEB00 ; address of the video latch

```

```

CONLATCH EQU    $FCD01C      ; address of contrast latch
STATREG  EQU    $FCF800      ; address of status register
VERTRES  EQU    $FCE018      ; vertical retrace reset
;
; note: VIA1 is parallel port, VIA2 is cops port
;
COPSVIA  EQU    $FCDD81      ; Via 2 base for NEW & OLD I/O boards
COPSIFR  EQU    $1A          ; Interrupt flag register
PORTA    EQU    2            ; Offset to Port A
;
; PIA REGISTERS for the parallel port
;
ORB      EQU    0            ; output regs
ORA      EQU    8
IRA      EQU    ORA          ; input regs
IRB      EQU    ORB
DDRB     EQU    $10          ; data direction regs
DDRA     EQU    $18
ACR      EQU    $58          ; aux control
PCR      EQU    $60          ; peripheral control
IFR      EQU    $68          ; int flags
IER      EQU    $70          ; int enable
NHS      EQU    $78          ; reg A no hand shake
;
; Equates for user registers
;
RD0      EQU    0
RD1      EQU    4
RD2      EQU    8
RD3      EQU    12
RD4      EQU    16
RD5      EQU    20
RD6      EQU    24
RD7      EQU    28
RA0      EQU    32
RA1      EQU    36
RA2      EQU    40
RA3      EQU    44
RA4      EQU    48
RA5      EQU    52
RA6      EQU    56
RA7      EQU    60
RPC      EQU    64
RSR      EQU    68
RUS      EQU    72
RSS      EQU    76
;
REGA7    EQU    USERREGS+RA7
;
; SCC MACROS
;
.MACRO SCCWT      ; <register number>,<source operand>
NOP
MOVE.B    %1,(A0)      ; specify the register
NOP      ; we have to delay for latency timing limits
MOVE.B    %2,(A0)      ; write to that register
.ENDM

.MACRO SCCWTO      ; <register number>,<source operand = 0>
NOP
MOVE.B    %1,(A0)      ; specify the register
;undone NOP      *** ; we have to delay for latency timing limits

```

```

MOVE.L DO,-(SP) ;DONT CLOBBER TEMPORARY REGISTER
CLR.B DO
MOVE.B DO,(A0) ; write to that register
MOVE.L (SP)+,DO ;RESTORE TEMPORARY REGISTER
.ENDM

.MACRO SCCRD ; <register number>,<destination operand>
NOP ; this nop assures us for latency timing
MOVE.B %1,(A0) ; specify the register
NOP
MOVE.B (A0),%2 ; read from that register
.ENDM

;
; SCC EQUATES
;
SCCBASE .EQU $FCD204 ; would have been $FCD200 but SCC wired wrong
; requires subtract of 1 or 3 to get to A or B

SCCCB .EQU $FCD201
SCCCA .EQU $FCD203
SCCDATA .EQU 4
LITTLE0 .EQU $1200 ; SCC offset between LISA and Little machine

;
; EQUATES FOR THE UART
;
CR0 EQU 0
CR3 EQU 3 ;CONTROL REGISTER SELECTS
CR4 EQU 4
CR5 EQU 5
SRO EQU 0 ;STATUS REGISTER OFFSET
CTL EQU 4 ;CONTROL REG OFFSET
RST EQU $18 ;RESET CODE FOR SIDES A AND B
CR3DAT EQU $C1 ;ENABLE REC, 8 BIT WORDS
CR4DAT EQU $44 ;ASYNC, 2 STOP BITS, X16 CLK
CR5DAT EQU $EA ;ENABLE XMIT, 8 BIT WORDS
RDA EQU 0 ;RECEIVE STATUS BIT
TBE EQU 2 ;XMIT STATUS BIT
SYNCA EQU 4 ;SYNC/HUNT BIT (PIN 11)

;
; equates for the debug card
;
BDC EQU 0 ; bus data capture
LAC EQU $800 ; low address capture
HACC EQU $1000 ; high address & control capture
SMIG EQU $2000 ; select & map interrupt generation
MR EQU $1FFF ; mode register, byte, write only
IR EQU $1FFE ; index register, word, read only

;
Silence .EQU 48
NMIKey .EQU 114
SetNMIKey .EQU 116
SetKeyRoutine .EQU 120
KeysDown .EQU 52
NMISync .EQU 140
ScreenKeybd .EQU 142
SetScreenKeybd .EQU 144
AltKeyEvent .EQU 148
MicroTimer .EQU 154
Poll .EQU 158

;
; Macro TrapTo is used to call the hardware interface routines.
; Parameter %1 is the magic number associated with the routine begin called.
;

```

```
.MACRO TRAPTO ; <routine>
MOVE.L D7,-(SP) ; save register
MOVE.W #%1,D7 ; routine number
TRAP #5 ; trap
MOVE.L (SP)+,D7 ; restore register
.ENDM

;

;
; equates for Mac debugging.....
;
TargetHeap .EQU $2AA
BufPtr .EQU $10C
```