Atari 6502 System Equations and Macros#

The origin of this file is not clear. The filename and file-path naming suggest it comes from a crosscompiler (maybe PDP-11 or VAX). It might be the system equates used by Atari internally on their crosscompilers.

Besides Atari 8bit Equates it also contains Equates for KIM-1 and some kind of a virtual machine interpreter (maybe Forth, maybe a 6502 CPU Emulator)

```
;<HESS.ATARI>SYSMAC.SML.27  8-Mar-82 08:39:38, Edit by HESS

;6502 SYSTEM -*-MACRO-*- DEFINITIONS

; ***** ATARI SYSTEM DEFS *****

.MACRO   ATARI

;   VECTOR TABLE

EDITRV   =$E400         ;EDITOR
SCRENV   =$E410         ;TELEVISION SCREEN
KEYBDV   =$E420         ;KEYBOARD
PRINTV   =$E430         ;PRINTER
CASETV   =$E440         ;CASSETTE

;   JUMP VECTOR TABLE

DISKIV   =$E450         ;DISK INITIALIZATION
DSKINV   =$E453         ;DISK INTERFACE
CIOV   =$E456         ;CIO ROUTINE
SIOV   =$E459         ;SIO ROUTINE
SETVBV   =$E45C         ;SET VERTICAL BLANK VECTORS
SYSVBV   =$E45F         ;SYSTEM VERTICAL BLANK ROUTINE
XITVBV   =$E462         ;EXIT VERTICAL BLANK ROUTINE
SIOINV   =$E465         ;SIO INIT
SENDEV   =$E468         ;SEND ENABLE ROUTINE
INTINV   =$E46B         ;INTERRUPT HANDLER INIT
CIOINV   =$E46E         ;CIO INIT
BLKBDV   =$E471         ;BLACKBOARD MODE
WARMSV   =$E474         ;WARM START ENTRY POINT
COLDSV   =$E477         ;COLD START ENTRY POINT
RBLOKV   =$E47D         ;CASSETTE READ BLOCK VECTOR
DSOPIV   =$E480         ;CASSETTE OPEN FOR INPUT VECTOR

;   SOME USEFUL INTERNAL ROUTINES

KGETCH   =$F6E2         ;GET CHAR FROM KEYBOARD
EOUTCH   =$F6A4         ;OUTPUT CHAR TO SCREEN
PUTLIN   =$F385         ;OUTPUT LINE TO IOCB#0

;   COMMAND CODES FOR IOCB

OPEN   =$03         ;OPEN FOR INPUT/OUTPUT
GETREC   =$05         ;GET RECORD (TEXT)
GETCHR   =$07         ;GET CHARACTER(S)
PUTREC   =$09         ;PUT RECORD (TEXT)
PUTCHR   =$0B         ;PUT CHARACTER(S)
CLOSE   =$0C         ;CLOSE DEVICE
```

```
STATIS   =$0D         ;STATUS REQUEST
SPECIL   =$0E         ;SPECIAL ENTRY COMMANDS

;   SPECIAL ENTRY COMMANDS

DRAWLN   =$11         ;DRAW LINE
FILLIN   =$12         ;DRAW LINE WITH RIGHT FILL
RENAME   =$20         ;RENAME DISK FILE
DELETE   =$21         ;DELETE DISK FILE
FORMAT   =$22         ;FORMAT DISK
LOCKFL   =$23         ;LOCK FILE (READ ONLY)
UNLOCK   =$24         ;UNLOCK FILE
POINT   =$25         ;POINT SECTOR
NOTE   =$26         ;NOTE SECTOR

CCIO   =$28         ;CONCURRENT I/O MODE

IOCFRE   =$FF         ;IOCB "FREE"

;   AUX1 VALUES FOR OPEN

APPEND   =$01         ;OPEN FOR APPEND
DIRECT   =$02         ;OPEN FOR DIRECTORY ACCESS
OPNIN   =$04         ;OPEN FOR INPUT
OPNOT   =$08         ;OPEN FOR OUTPUT
OPNINO   =OPNIN!OPNOT      ;OPEN FOR INPUT/OUTPUT
MXDMOD   =$10         ;OPEN FOR MIXED MODE
INSCLR   =$20         ;OPEN WITHOUT CLEARING SCREEN

; OS STATUS CODES

SUCCES   =$01         ;SUCCESSFUL OPERATION
BRKABT   =$80         ;(128) BREAK KEY ABORT
PRVOPN   =$81         ;(129) IOCB ALREADY OPEN
NONDEV   =$82         ;(130) NON-EX DEVICE
WRONLY   =$83         ;(131) IOCB OPENED FOR WRITE ONLY
NVALID   =$84         ;(132) INVALID COMMAND
NOTOPN   =$85         ;(133) DEVICE OR FILE NOT OPEN
BADIOC   =$86         ;(134) INVALID IOCB NUMBER
RDONLY   =$87         ;(135) IOCB OPENED FOR READ ONLY
EOFERR   =$88         ;(136) END OF FILE
TRNRCD   =$89         ;(137) TRUNCATED RECORD
TIMOUT   =$8A         ;(138) DEVICE TIMEOUT
DNACK   =$8B         ;(139) DEVICE DOES NOT ACK COMMAND
FRMERR   =$8C         ;(140) SERIAL BUS FRAMING ERROR
CRSROR   =$8D         ;(141) CURSOR OUT OF RANGE
OVRRUN   =$8E         ;(142) SERIAL BUS DATA OVERRUN
CHKERR   =$8F         ;(143) SERIAL BUS CHECKSUM ERROR
DERROR   =$90         ;(144) DEVICE ERROR (OPERATION INCOMPLETE)
BADMOD   =$91         ;(145) BAD SCREEN MODE NUMBER
FNCNOT   =$92         ;(146) FUNCTION NOT IN HANDLER
SCRMEM   =$93         ;(147) INSUFFICIENT MEMORY FOR SCREEN MODE

;   PAGE 0 LOCATIONS

LINZBS   =$00         ;LINBUG STORAGE

;  THESE LOCS ARE NOT CLEARED
```

```
CASINI   =$02         ;CASSETTE INIT LOC
RAMLO   =$04         ;RAM POINTER FOR MEM TEST
TRAMSZ   =$06         ;TEMP LOC FOR RAM SIZE
TSTDAT   =$07         ;RAM TEST DATA LOC

;  CLEARED ON COLDSTART ONLY

WARMST   =$08         ;WARM START FLAG
BOOTQ   =$09         ;SUCCESSFUL BOOT FLAG
DOSVEC   =$0A         ;DOS START VECTOR
DOSINI   =$0C         ;DOS INIT ADDRESS
APPMHI   =$0E         ;APPLICATION MEM HI LIMIT

;  CLEARED ON COLD OR WARM START

INTZBS   =$10         ; START OF OS RAM CLEAR LOC => $7F
POKMSK   =$10   \^Y   ;SYSTEM MASK FOR POKEY IRQ ENABLE
BRKKEY   =$11         ;BREAK KEY FLAG
RTCLOK   =$12         ;REAL TIME CLOCK (60HZ OR 16.66666 MS)
BUFADR   =$15         ;INDIRECT BUFFER ADDRESS REG
ICCOMT   =$17         ;COMMAND FOR VECTOR HANDLER
DSKFMS   =$18         ;DISK FILE MANAGER POINTER
DSKUTL   =$1A         ;DISK UTILITIES POINTER
PTIMOT   =$1C         ;PRINTER TIME OUT REGISTER
PBPNT   =$1D         ;PRINT BUFFER POINTER
PBUFSZ   =$1E         ;PRINT BUFFER SIZE
PTEMP   =$1F         ;TEMP REG\^]

ZIOCB   =$20         ;PAGE 0 I/O CONTROL BLOCK
IOCBSZ   =16         ;NUMBER OF BYTES / IOCB
MAXIOC   =8*IOCBSZ      ;LENGTH OF IOCB AREA
IOCBAS   =ZIOCB

ICHIDZ   =$20         ;HANDLER INDEX NUMBER ($FF := IOCB FREE)
ICDNOZ   =$21         ;DEVICE NUMBER (DRIVE NUMBER)
ICCOMZ   =$22         ;COMMAND CODE
ICSTAZ   =$23         ;STATUS OF LAST IOCB ACTION
ICBALZ   =$24         ;BUFFER ADDRESS (LOW)
ICBAHZ   =$25         ;  "      "   (HIGH)
ICPTLZ   =$26         ;PUT BYTE ROUTINE ADDRESS - 1
ICPTHZ   =$27
ICBLLZ   =$28         ;BUFFER LENGTH (LOW)
ICBLHZ   =$29         ;  "      "   (HIGH)
ICAX1Z   =$2A         ;AUX INFO
ICAX2Z   =$2B
ICSPRZ   =$2C         ;SPARE BYTES (CIO LOCAL USE)
ICIDNO   =ICSPRZ+2      ;IOCB LUMBER * 16
CIOCHR   =ICSPRZ+3      ;CHARACTER BYTE FOR CURRENT OPERATION

STATUS   =$30         ;INTERNAL STATUS STORAGE
CHKSUM   =$31         ;CHECKSUM (SINGLE BYTE SUM WITH CARRY)
BUNRLO   =$32         ;POINTER TO DATA BUFFER (LO BYTE)
BUFRHI   =$33         ;POINTER TO DATA BUFFER (HI BYTE)
BFENLO   =$34         ;NEXT BYTE PAST END OF BUFFER (LO BYTE)
BNENHI   =$35         ;NEXT BYTE PAST END OF BUFFER (HI BYTE)
CRETRY   =$36         ;NUMBER OF COMMAND FRAM RETRIES
DRETRY   =$37\^Y      ;NUMBER OF DEVICE RETRIES
BUFRFL   =$38         ;DATA BUFFER FULL FLAG
RECVDN   =$39         ;RECEIVE DONE FLAG
```

```
XMTDON   =$3A         ;XMIT DONE FLAG
CHKSNT   =$3B         ;CHECKSUM SENT FLAG
NOCKSM   =$3C         ;NO CHECKSUM FOLLOWS DATA FLAG

BPTR   =$3D         ;BUFFER POINTER (CASSETTE)
FTYPE   =$3E         ;FILE TYPE (SHORT IRG/LONG IRG)
FEOF   =$3F         ;END OF FILE FLAG (CASSETTE)
FREQ   =$40         ;FREQ COUNTER FOR CONSOLE SPEAKER
SOUNDR   =$41         ;NOISY I/O FLAG. (ZERO IS QUIET)
CRITIC   =$42         ;CRITICAL CODE IF NON-ZERO)

FMSZPG   =$43         ;DISK FILE MANAGER SYSTEM STORAGE (7 BYTES)

CKEY   =$4A         ;SET WHEN GAME START PRESSED
CASSBT   =$4B         ;CASSETTE BOOT FLAG
DSTAT   =$4C         ;DISPLAY STATUS
ATRACT   =$4D         ;ATTRACT MODE FLAG
DRKMSK   =$4E         ;DARK ATTRACT MASK
COLRSH   =$4F         ;ATTRACT COLOR SHIFTER (XOR'D WITH PLAYFIELD)

TMPCHR   =$50         ;TEMP CHAR STORAGE (DISPLAY HANDLER)
HOLD1   =$51         ;TEMP STG (DISPLAY HANDLER)
LMARGN   =$52         ;LEFT MARGIN
RMARGN   =$53         ;RIGHT MARGIN
ROWCRS   =$54         ;CURSOR COUNTERS
COLCRS   =$55
DINDEX   =$57         ;DISPLAY INDEX (VARIOUS QUANTS)
SAVMSC   =$58
OLDROW   =$5A         ;PREVIOUS ROW/COL
OLDCOL   =$5B
OLDCHR   =$5D         ;DATA UNDER CURSOR
OLDADR   =$5E
NEWROW   =$60         ;POINT DRAWS TO HERE
NEWCOL   =$61
LOGCOL   =$63         ;POINTS AT COLUMN IN LOGICAL LINE
ADRESS   =$64         ;INDIRECT POINTER
MLTTMP   =$66         ;MULTIPLY TEMP
OPNTMP   =MLTTMP         ;FIRST BYTE IS USED IN OPEN AS TEMP
SAVADR   =$68
RAMTOP   =$6A         ;RAM SIZE DEFINED BY POWER ON LOGIC
BUFCNT   =$6B         ;BUFFER COUNT
BUFSTR   =$6C         ;EDITOR GETCH POINTER
BITMSK   =$6E         ;BIT MASK
SHFAMT   =$6F         ;OUTCHR SHIFT

ROWAC   =$70         ;USED BY "DRAW"
COLAC   =$72
ENDPT   =$74
DELTAR   =$76
DELTAC   =$77
ROWINC   =$79
COLINC   =$7A
SWPFLG   =$7B         ;NON-0 IF TXT AND RAM SWAPPED
HOLDCH   =$7C         ;CH BEFORE CNTL & SHFT PROCESSING IN KGETCH
INSDAT   =$7D         ;INSERT CHAR SAVE
COUNTR   =$7E         ;DRAW COUNTER

;;;   $80 TO $FF ARE RESERVED FOR USER APPLICATIONS
```

```
;   PAGE 2 LOCATIONS

INTABS   =$200         ;INTERRUPT TABLE
VDSLST   =$200         ;DISPLAY LIST NMI VECTOR
VPRCED   =$202         ;PROCEED LINE IRQ VECTOR
VINTER   =$204         ;INTERRUPT LINE IRQ VECTOR
VBREAK   =$206         ;"BRK" VECTOR
VKEYBD   =$208         ;POKEY KEYBOARD IRQ VECTOR
VSERIN   =$20A         ;POKEY SERIAL INPUT READY
VSEROR   =$20C         ;POKEY SERIAL OUTPUT READY
VSEROC   =$20E         ;POKEY SERIAL OUTPUT DONE
VTIMR1   =$210         ;POKEY TIMER 1 IRQ
VTIMR2   =$212         ;POKEY TIMER 2 IRQ
VTIMR4   =$214         ;POKEY TIMER 4 IRQ (DO NOT USE)
VIMIRQ   =$216         ;IMMEDIATE IRQ VECTOR
CDTMV1   =$218         ;COUNT DOWN TIMER 1
CDTMV1   =$21A         ;COUNT DOWN TIMER 2
CDTMV1   =$21C         ;COUNT DOWN TIMER 3
CDTMV1   =$21E         ;COUNT DOWN TIMER 4
CDTMV1   =$220         ;COUNT DOWN TIMER 5
VVBLKI   =$222         ;IMMEDIATE VERTICAL BLANK NMI VECTOR
VVBLKD   =$224         ;DEFERRED VERTICAL BLANK NMI VECTOR
CDTMA1   =$226         ;COUNT DOWN TIMER 1 JSR ADDRESS
CDTMA2   =$228         ;COUNT DOWN TIMER 2 JSR ADDRESS
CDTMF3   =$22A         ;COUNT DOWN TIMER 3 FLAG
SRTIMR   =$22B         ;SOFTWARE REPEAT TIMER
CDTMF4   =$22C         ;COUNT DOWN TIMER 4 FLAG
INTEMP   =$22D         ;IAN'S TEMP (???)
CDTMF5   =$22E         ;COUNT DOWN TIMER 5 FLAG
SDMCTL   =$22F         ;SAVE DMACTL REGISTER
SDLSTL   =$230         ;SAVE DISPLAY LIST (LOW)
SDLSTH   =$231         ;SAVE DISPLAY LIST (HIGH)
SSKCTL   =$232         ;SKCTL REGISTER RAM

LPENH   =$234         ;LIGHT PEN HORIZ VALUE
LPENV   =$235         ;LIGHT PEN VERT VALUE
           ; ($236 - $239 SPARE)
CDEVIC   =$23A         ;COMMAND FRAME BUFFER - DEVICE
CCOMND   =$23B         ;COMMAND
CAUX1   =$23C         ;COMMAND AUX BYTE 1
CAUX2   =$23D         ;COMMAND AUX BYTE 2
TEMP   =$23E         ;YES
ERRFLG   =$23F         ;ERROR FLAG - ANY DEVICE ERROR EXCEPT TIMEOUT

DFLAGS   =$240         ;DISK FLAGS FROM SECTOR ONE
DBSECT   =$241         ;NUMBER OF DISK BOOT SECTORS
BOOTAD   =$242         ;ADDRESS FOR DISK BOOT LOADER
COLDST   =$244         ;COLDSTART FLAG (1 = DOING COLDSTART)
           ;($245 SPARE)
DSKTIM   =$246         ;DISK TIME OUT REG
LINBUF   =$247         ;CHAR LINE BUFFER (40 BYTES)

GPRIOR   =$26F         ;GLOBAL PRIORITY CELL
PADDL0   =$270         ;POT 0 SHADOW
PADDL1   =$271         ;POT 1 SHADOW
PADDL2   =$272         ;POT 2 SHADOW
PADDL3   =$273         ;POT 3 SHADOW
PADDL4   =$274         ;POT 4 SHADOW
PADDL5   =$275         ;POT 5 SHADOW
```

```
PADDL6   =$276         ;POT 6 SHADOW
PADDL7   =$277         ;POT 7 SHADOW
STICK0   =$278         ;JOYSTICK 0 SHADOW
STICK1   =$279         ;JOYSTICK 1 SHADOW
STICK2   =$27A         ;JOYSTICK 2 SHADOW
STICK3   =$27B         ;JOYSTICK 3 SHADOW
PTRIG0   =$27C         ;PADDLE 0 TRIGGER
PTRIG1   =$27D         ;PADDLE 1 TRIGGER
PTRIG2   =$27E         ;PADDLE 2 TRIGGER
PTRIG3   =$27F         ;PADDLE 3 TRIGGER
PTRIG4   =$280         ;PADDLE 4 TRIGGER
PTRIG5   =$281         ;PADDLE 5 TRIGGER
PTRIG6   =$282         ;PADDLE 6 TRIGGER
PTRIG7   =$283         ;PADDLE 7 TRIGGER
STRIG0   =$284         ;JOYSTICK 0 TRIGGER
STRIG1   =$285         ;JOYSTICK 1 TRIGGER
STRIG2   =$286         ;JOYSTICK 2 TRIGGER
STRIG3   =$287         ;JOYSTICK 3 TRIGGER

CSTAT   =$288         ;(UNUSED)
WMODE   =$289         ;R/W FLAG FOR CASSETTE
BLIM   =$28A         ;BUFFER LIMIT (CASSETTE)
           ;($28B - $28F SPARE)
TXTROW   =$290         ;TEXT ROWCRS
TXTCOL   =$291         ;TEXT ROWCOL
TINDEX   =$293         ;TEXT INDEX
TXTMSC   =$294         ;FOOLS CONVRT INTO NEW MSC
TXTOLD   =$296         ;OLDROW & OLDCOL FOR TEXT (AND THEN SOME)
TMPX1   =$29C
HOLD3   =$29D
SUBTMP   =$29E
HOLD2   =$29F
DMASK   =$2A0
TMPLBT   =$2A1
ESCFLG   =$2A2         ;ESCAPE FLAG
TABMAP   =$2A3         ;TAB BUFFER
LOGMAP   =$2B2         ;LOGICAL LINE START BIT MAP
INVFLG   =$2B6         ;INVERSE VIDEO FLAG (ATARI KEY)
FILFLG   =$2B7         ;RIGHT FILL FLAG FOR DRAW
TMPROW   =$2B8
TMPCOL   =$2B9
SCRFLG   =$2BB         ;SET IF SCROLL OCCURS
HOLD4   =$2BC         ;MORE DRAW TEMPS
HOLD5   =$2BD
SHFLOK   =$2BE         ;SHIFT LOCK KEY
BOTSCR   =$2BF         ;BOTTOM OF SCREEN (24 NORM, 4 SPLIT)

PCOLR0   =$2C0         ;P0 COLOR
PCOLR1   =$2C1         ;P1 COLOR
PCOLR2   =$2C2         ;P2 COLOR
PCOLR3   =$2C3         ;P3 COLOR
COLOR0   =$2C4         ;COLOR 0
COLOR1   =$2C5
COLOR2   =$2C6
COLOR3   =$2C7
COLOR4   =$2C8         ;BACKGROUND
           ;($2C9 - $2DF SPARE)
GLBABS   =$2E0         ;GLOBAL VARIABLES
           ;($2E0 - $2E3 SPARE)
```

```
RAMSIZ   =$2E4         ;RAM SIZE (HI BYTE ONLY)
MEMTOP   =$2E5         ;TOP OF AVAILABLE MEMORY
MEMLO   =$2E7         ;BOTTOM OF AVAILABLE MEMORY
           ;($2E9 SPARE)
DVSTAT   =$2EA         ;STATUS BUFFER
CBAUDL   =$2EE         ;CASSETTE BAUD RATE (LO BYTE)
CBAUDH   =$2EF         ;   "      "    "   (HI BYTE)
CRSINH   =$2F0         ;CURSOR INHIBIT (00 = CURSOR ON)
KEYDEL   =$2F1         ;KEY DELAY
CH1   =$2F2
CHACT   =$2F3         ;CHACTL REGISTER (SHADOW)
CHBAS   =$2F4         ;CHBAS REGISTER (SHADOW)
           ;($2F5 - $2F9 SPARE)
CHAR   =$2FA
ATACHR   =$2FB         ;ATASCII CHARACTER
CH   =$2FC         ;GLOBAL VARIABLE FOR KEYBOARD
FILDAT   =$2FD         ;RIGHT FILL DATA (DRAW)
DSPFLG   =$2FE         ;DISPLAY FLAG: DISP CONTROLS IF NON-ZERO
SSFLAG   =$2FF         ;START/STOP FLAG (CNTL-1) FOR PAGING

;   PAGE 3 LOCATIONS

DCB   =$300         ;DEVICE CONTROL BLOCK
DDEVIC   =$300         ;BUS I.D. NUMBER
DUNIT   =$301         ;UNIT NUMBER
DCOMND   =$302         ;BUS COMMAND
DSTATS   =$303         ;COMMAND TYPE/STATUS RETURN
DBUFLO   =$304         ;DATA BUFFER POINTER
DBUFHI   =$305         ; ...
DTIMLO   =$306         ;DEVICE TIME OUT IN 1 SEC. UNITS
DUNUSE   =$307         ;UNUSED
DBYTLO   =$308         ;BYTE COUNT
DBYTHI   =$309         ; ...
DAUX1   =$30A         ;COMMAND AUXILLARY BYTES
DAUX2   =$30B         ; ...

TIMER1   =$30C         ;INITIAL TIMER VALUE
ADDCOR   =$30E         ;ADDITION CORRECTION
CASFLG   =$30F         ;CASSETTE MODE WHEN SET
TIMER2   =$310         ;FINAL TIME VALUE (USED TO COMPUTE BAUD RATE)
TEMP1   =$312         ;TEMP LOCATIONS
TEMP2   =$314         ; ...
TEMP3   =$315         ; ...
SAVIO   =$316         ;SAVE SERIAL IN DATA PORT
TIMFLG   =$317         ;TIME OUT FLAG FOR BAUD RATE CORRECTION
STACKP   =$318         ;SIO STACK POINTER SAVE LOC
TSTAT   =$319         ;TEMP STATUS LOC

HATABS   =$31A         ;HANDLER ADDRESS TABLE
MAXDEV   =$21         ;MAXIMUM HANDLER ADDRESS INDEX

;   IOCB OFFSETS

IOCB   =$340         ;I/O CONTROL BLOCKS
ICHID   =$340         ;HANDLER INDEX ($FF = FREE)
ICDNO   =$341         ;DEVICE NUMBER (DRIVE NUMBER)
ICCOM   =$342         ;COMMAND CODE
ICSTA   =$343         ;STATUS
ICBAL   =$344         ;BUFFER ADDRESS
```

```
ICBAH   =$345         ; ...
ICPTL   =$346         ;PUT BYTE ROUTINE ADDRESS - 1
ICPTH   =$347         ; ...
ICBLL   =$348         ;BUFFER LENGTH
ICBLH   =$349         ; ...
ICAX1   =$34A         ;AUXILLARY INFO
ICAX2   =$34B         ; ...
ICSPR   =$34C         ;4 SPARE BYTES

PRNBUF   =$3C0         ;PRINTER BUFFER
           ;($3EA - $3FC SPARE)

;   PAGE 4 LOCATIONS

CASBUF   =$3FD         ;CASSETTE BUFFER

; USER AREA STARTS HERE AND GOES TO THE END OF PAGE 5

USAREA   =$480

;ATASCII CHARACTER DEFS

.ATCLR   =$7D         ;CLEAR SCREEN CHARACTER
.ATRUB   =$7E         ;BACK SPACE (RUBOUT)
.ATTAB   =$7F         ;TAB
.ATEOL   =$9B         ;END-OF-LINE
.ATBEL   =$FD         ;CONSOLE BELL
.ATURW   =$1C         ;UP-ARROW
.ATDRW   =$1D         ;DOWN-ARROW
.ATLRW   =$1E         ;LEFT-ARROW
.ATRRW   =$1F         ;RIGHT-ARROW

; USEFUL VALUES

LEDGE   =2         ;LMARGN'S INITIAL VALUE
REDGE   =39         ;RMARGN'S INITIAL VALUE

ZPC   =0      ;PC CODE FOR ZERO PAGE PC
P6PC   =1      ;PC CODE FOR PAGE 6
PPC   =2      ;PC CODE FOR PROGRAM MEMORY

;INIT PC VALUES

CURPC   =0
PC0   =0      ;PAGE ZERO
PC1   =$600      ;PAGE 6 PC
PC2   =$3800      ;PROGRAM PC

.MACRO   PCBRK
  .PRINT PC0   ;PAGE ZERO BREAK
  .PRINT PC1   ;PAGE 6 BREAK
  .PRINT PC2   ;PROGRAM BREAK
.ENDM

.ENDM      ;; ATARI
\^L; ***** KIM SYSTEM DEFS *****

.MACRO   KIMDEF
```

```
;LOCATIONS IN 6530-002 I/O

KSAD   =$1740      ;PORT A DATA
KPADD   =$1741      ;PORT A DATA DIRECTION
KSBD   =$1742      ;PORT B DATA
KSBDD   =$1743      ;PORT B DATA DIRECTION
KC1T   =$1744      ;CLOCK /1
KC8T   =$1745      ;CLOCK /8
KC64T   =$1746      ;CLOCK /64
KCKT   =$1747      ;CLOCK /1024

;LOCATIONS IN 6530-003 I/O

PAD   =$1700      ;PORT A DATA
PADD   =$1701      ;PORT A DATA DIRECTION
PBD   =$1702      ;PORT B DATA
PBDD   =$1703      ;PORT B DATA DIRECTION
CLK1T   =$1704      ;CLOCK /1
CLK8T   =$1705      ;CLOCK /8
CLK64T   =$1706      ;CLOCK /64
CLKKT   =$1707      ;CLOCK /1024
IC1T   =$170C      ;CLOCK /1 INTS ENABLED
IC8T   =$170D      ;CLOCK /8   "
IC64T   =$170E      ;CLOCK /64   "
ICKT   =$170F      ;CLOCK /1024   "

KRAM   =$1780      ;SCRATCH PAD RAM
KRAMX   =$17FF      ;KRAM END
\^L;PAGE ZERO VARIABLES USED BY KIM MONITOR

PCL   =$EF      ;PROGRAM COUNTER
PCH   =$F0
PS   =$F1      ;PROCESSOR STATUS REG
SP   =$F2      ;STACK POINTER
AC   =$F3      ;ACCUMULATOR
YREG   =$F4      ;Y INDEX
XREG   =$F5      ;X INDEX
CHKSUM   =$F6      ;CHECKSUM TEMP (2 BYTES)
INBUF   =$F8      ;INPUT BUFFER (2 BYTES)
POINT   =$FA      ;OPEN CELL ADDRS (2 BYTES)
TEMP   =$FC      ;TEMPORARY
TMPX   =$FD      ;TEMPORARY X SAVE
CHAR   =$FE      ;INPUT CHARACTER
MODE   =$FF      ;ADDRS/DATA FLAG FOR DPY

;PAGE 23 VARIABLES USED BY KIM MONITOR

CHKL   =$17E7      ;ANOTHER CHECKSUM
CHKH   =$17E8
SAVX   =$17E9      ;3 BYTE SCRATCH AREA
VEB   =$17EC      ;6 BYTE PROGRAM FOR CASETTE CODE
CNTL   =$17F2      ;TTY DELAY COUNT
CNTH   =$17F3
TIMH   =$17F4      ;TEMP FOR TTY TIMING
SAL   =$17F5      ;START ADDRS FOR CASETTE
SAH   =$17F6
EAL   =$17F7      ;END ADDRS FOR CASSETE
EAH   =$17F8
CID   =$17F9      ;FILE ID FOR CASETTE
```

```
;INTERUPT VECTORS

NMIV   =$17FA      ;NMI VECTOR (STOP := $1C00)
RSTV   =$17FC      ;RESET VECTOR
IRQV   =$17FE      ;IRQ VECTOR (BRK := $1C00)
\^L;VARIOUS HANDY ROUTINE LOCATIONS IN KIM MONITOR

SAVE   =$1C00      ;KIM ENTRY TO SAVE WORLD FIRST
SAVER   =$1C05      ;KIM ENTRY VIA JSR (A LOST)
RESET   =$1C22      ;KIM RESET ENTRY
KIM   =$1C4F      ;KIM START ADDRS
GOEXEC   =$1DC8      ;RESTORE MACHINE AND RETURN
PRTPNT   =$1E1E      ;ROUTINE TO PRINT "POINT" (CALLS CHK)
CRLF   =$1E2F      ;PRINT CRLF
PRTBYT   =$1E3B      ;PRINT 1 HEX BYTE AS 2 ASCII CHARS
         ;A PRESERVED
HEXTA   =$1E4C      ;PRINT 1 ASCII HEX DIGIT (4 BITS)
GETCH   =$1E5A      ;GET CHARACTER (PRESERVES X)
INITS   =$1E88      ;INITIALIZATION
OUTSP   =$1E9E      ;PRINT A SPACE
OUTCH   =$1EA0      ;PRINT CHARACTER IN A
AK   =$1EFE      ;KEYBOARD ROUTINE
SCANDS   =$1F1F      ;DISPLAY F9-FB
INCPT   =$1F63      ;INCREMENT "POINT"
GETKEY   =$1F6A      ;GET KEY FROM KEYBOARD
CHK   =$1F91      ;CHECKSUM ROUTINE (COMPUTES "CHKSUM")
GETBYT   =$1F9D      ;GET 2 ASCII CHARS INTO HEX BYTE
         ;X PRESERVED
PACK   =$1FAC      ;PACK CHAR INTO INPUT BUFFER
         ;RETURNS A=0 IF HEX CHAR
OPEN   =$1FCC      ;COPIES INBUF TO POINT
DPYTAB   =$1FE7      ;HEX TO 7 SEGMENT TABLE

;ROUTINES IN CASSETTE DRIVER

CHKT   =$194C      ;COMPUTE CHKSUM FOR TAPE
INTVEB   =$1932      ;INIT VEB WITH SAL,SAH / CLEAR CHKSUM
INCVEB   =$19EA      ;INCREMENT VEB+1,2
RDBYT   =$19F3      ;READ BYTE FROM TAPE
PACKT   =$1A00      ;PACK ASCII INTO SAVX
RDCHT   =$1A24      ;GET 1 CHAR FROM TAPE
RDBIT   =$1A41      ;GET 1 BIT FROM TAPE IN SIGN OF A
DUMPT   =$1800      ;DUMP MEM TO TAPE
LOADT   =$1873      ;LOAD MEM FROM TAPE

ZPC   =0      ;PC CODE FOR ZERO PAGE PC
PPC   =1      ;PC CODE FOR PROGRAM PC
KPC   =2      ;PC CODE FOR KRAM PC
XPC   =3      ;PC CODE FOR LOW 1K

;INIT PC VALUES

CURPC   =0
PC0   =0      ;PAGE ZERO
PC1   =$200      ;PROGRAM PC
PC2   =KRAM      ;KRAM PC
PC3   =$200      ;PC FOR LOW 1K
```

```
.MACR   PCBRK
  .PRINT PC0   ;PAGE ZERO BREAK
  .PRINT PC1   ;PROGRAM BREAK
  .PRINT PC2   ;KRAM BREAK
  .PRINT PC3   ;LOW 1K BREAK
.ENDM

.ENDM   ;;KIMDEF


\^L;GENERAL 6502 DEFS

.MACRO   M6502

;ASCII CHARACTER DEFS

.CHNUL   =@00         ;NULL
.CHSOH   =@01         ;SOH
.CHSTX   =@02
.CHETX   =@03
.CHEOT   =@04
.CHENQ   =@05
.CHACK   =@06
.CHBEL   =@07
.CHBS   =@10
.CHTAB   =@11
.CHLF   =@12
.CHVT   =@13
.CHFF   =@14
.CHCR   =@15
.CHSO   =@16
.CHSI   =@17
.CHDLE   =@20
.CHDC1   =@21
.CHDC2   =@22
.CHDC3   =@23
.CHDC4   =@24
.CHNAK   =@25
.CHSYN   =@26
.CHETB   =@27
.CHCAN   =@30
.CHEM   =@31
.CHSUB   =@32
.CHESC   =@33
.CHFS   =@34
.CHGS   =@35
.CHRS   =@36
.CHUS   =@37
.CHSP   =@40

.CHRUB   =@177
\^L;HANDY MACROS

.EQUIV   SEI,PIOFF      ;TURN OFF IRQ INTS
.EQUIV    CLI,PION      ;ALLOW IRQ INTS
.EQUIV   JSR,CALL      ;SUBROUTINE CALL
.EQUIV   RTS,RET         ;SUBROUTINE RETURN

;DOUBLE BYTE HANDLING MACROS
```

```
.MACR   MOV2   FROM,TO,INDX   ;;COPY 2 BYTE ITEM
   LDA   FROM      ;;GET FIRST BYTE
   .IIF B,<INDX>,STA TO
   .IIF NB,<INDX>,STA TO,INDX
   LDA   FROM+1      ;;THEN LAST
   .IIF B,<INDX>,STA TO+1
   .IIF NB,<INDX>,STA TO+1,INDX
.ENDM

.MACR   INC2   LOC,?MEXIT   ;;INCREMENT LOCATION
   INC   LOC      ;;LOW BYTE FIRST
   BNE   MEXIT      ;;EXIT IF NO CARRY
   INC   LOC+1      ;;ELSE INCR HIGH BYTE
MEXIT:
.ENDM

.MACR   DEC2   LOC,?MEXIT   ;;DECREMENT LOCATION
   SEC
   LDA   LOC      ;;GET LOW BYTE
   SBC   #1      ;;DEC DOESNT AFFECT CARRY
   STA   LOC
   BCS   MEXIT      ;;EXIT IF NO BORROW
   DEC   LOC+1      ;;ELSE ADJUST HIGH BYTE
MEXIT:
.ENDM

.MACR   CMP2EQ   A,B,TARGET,?NOMAT
   LDA   A+1
   CMP   B+1      ;;CHECK FOR MATCH
   BNE   NOMAT      ;;QUICK FINISH
   LDA   A
   CMP   B
   BEQ   TARGET      ;;IF .EQ. JUMP TO TARGET
NOMAT:            ;;FALL THROUGH IF NO MATCH
.ENDM
\^L.MACR   CMP2NE   A,B,TARGET   ;;OPOSITE OF CMP2EQ
   LDA   A+1
   CMP   B+1
   BNE   TARGET      ;;NO MATCH
   LDA   A+1
   CMP   B
   BNE   TARGET      ;;NOT SAME
            ;;FALL THROUGH IF SAME
.ENDM

.MACR   SET2   VALUE,LOC,INDX   ;;SET 2 BYTE IMMEDIATE
   LDA   #<VALUE>&$FF   ;;LOW BYTE
   .IIF B,<INDX>,STA   LOC
   .IIF NB,<INDX>,STA   LOC,INDX
   LDA   #<VALUE>^   ;;HIGH BYTE
   .IIF B,<INDX>,STA   LOC+1
   .IIF NB,<INDX>,STA   LOC+1,INDX
.ENDM

.MACR   SETAX   VALUE      ;; LOAD A&X WITH 16-BIT VALUE
   LDA   #<VALUE>&$FF   ;; LOW BYTE
   LDX   #<VALUE>^
.ENDM
```

```
.MACR   SETXA   VALUE       ;; LOAD X&A WITH 16-BIT VALUE
   LDX   #<VALUE>&$FF   ;; LOW BYTE
   LDA   #<VALUE>^
.ENDM

.MACR   SETXY   VALUE       ;; LOAD X&Y WITH 16-BIT VALUE
   LDX   #<VALUE>&$FF   ;; LOW BYTE
   LDY   #<VALUE>^
.ENDM

.MACR   MOV2X   FROM,TO      ;;COPY 2 BYTE ITEM
   LDX   FROM      ;;GET FIRST BYTE
   STX   TO
   LDX   FROM+1      ;;THEN LAST
   STX   TO+1
.ENDM

.MACR   SET2X   VALUE,LOC   ;;SET 2 BYTE IMMEDIATE
   LDX   #<VALUE&$FF>   ;;LOW BYTE
   STX   LOC
   LDX   #<VALUE>^   ;;HIGH BYTE
   STX   LOC+1
.ENDM

.MACR   CLR LST         ;;CLEAR LOCATIONS
   LDA   #0
    .IRP LOC,<LST>
   STA   LOC
    .ENDR
.ENDM

.MACR   CLR2 LST      ;;CLEAR FOR 2 BYTE THINGS
   LDA   #0
    .IRP LOC,<LST>
   STA   LOC
   STA   LOC+1
    .ENDR
.ENDM
\^L;MACROS TO MANIPULATE PCS

.MACR   SETPC   NAM,VAL
   SAVPC \\CURPC
   .=VAL
   CURPC=NAM
.ENDM

.MACR   USEPC   NAM
   SWPPC   \\CURPC,\\NAM
   CURPC=NAM
.ENDM

.MACR   SAVPC   NAM
   PC'NAM=.
.ENDM

.MACR   SWPPC   OLD,NEW
   PC'OLD=.
   .=PC'NEW
```

```
.ENDM

;FANCY END MACRO

.MACR   END   ARG
   SAVPC   \\CURPC
    .IF P2
   PCBRK
    .ENDC
   .END   ARG
.ENDM
\^L;UTILITY MACROS

.MACR   TYPE   MSGAD
   LDX   #MSGAD&$FF
   LDY   #MSGAD^
   CALL   PUTLIN
.ENDM

.MOD.   =$FFFF         ;MODIFIED LOC (REMOVE FOR ROM)

.ENDM   ;; M6502
```