SEMINAR OULINE FOR THE O.S.

A. Overview of the O.S.
   1. Elements of the O.S.
      a. ROM-Based Character Set.
      b. System Data Base
      c. A set of Vectors to System Routines
      d. I/O Subsystem Structure
         1. I/O Control Blocks
         2. I/O System Routines
      e. Interrupt Handlers
         1. Non-Maskable Interrupts (NMI's)
         2. Maskable Interrupts (IRQ's)
      f. Monitor
      g. Timers
      h. I/O Hardware Registers
      i. Program RAM
      j. Floating Point Package
      k. Cartridges
B. I/O Subsystem Structure
   1. I/O Control Blocks
      a. IOCB's
      b. ZIOCB
      c. DCB
   2. I/O System Routines
      a. CIO
         1. Calling Convention
         2. Handler Address Table(HATABS)
         3. Handler Entry Point Tables
      b. Device Handlers
         1. Resident Handlers
            a). Display Editor (E:)
            b). Screen Handler (S:)
            c). Keyboard (K:)
            d). Printer (P:)
            e). Cassette (C:)
         2. Resident Disk Handler
         3. Non-Resident Handlers
            a). DOS
            b). RS-232 Handler (850)
            c). User-added Handlers
      c. SIO
         1. Calling Convention
C. Monitor
   1. Called
      a. Power-up (Coldstart)
         1. Power-cycled
         2. Coldstart Vector (E477)
         3. SYSTEM RESET If COLDST<>0
      b. SYSTEM RESET (Warmstart)
         1. SYSTEM RESET Key
         2. Warmstart Vector (E474)
   2. Points of Interest
      a. A Warmstart Changes
         1. MEMLO

2. Handler Address Table (HATABS)
3. IRQ Vector Table

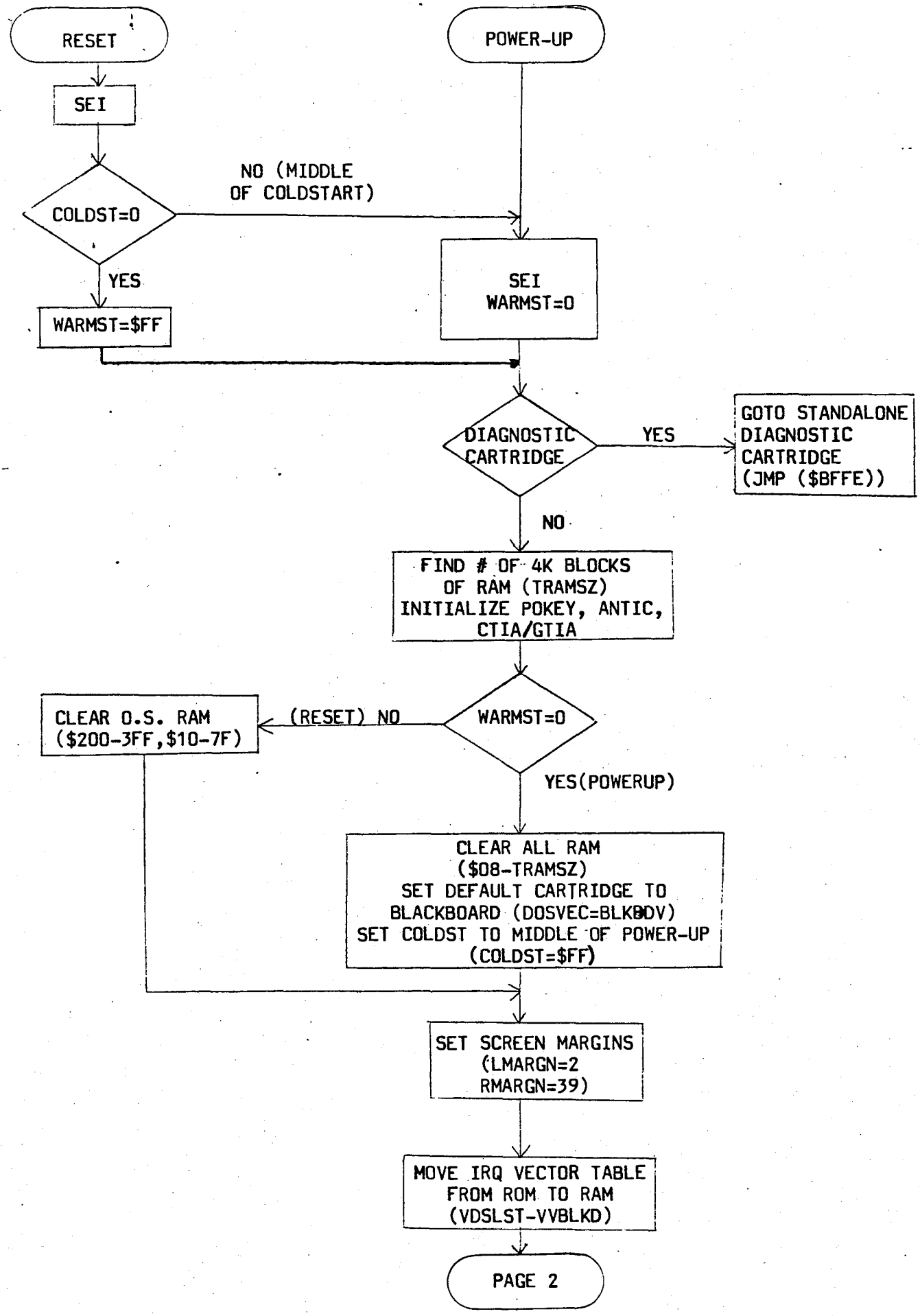
b. JSR (CASINI) if BOOT? set
c. JSR (DOSINI) if BOOT? set
e. Using DOSINI to fix a above

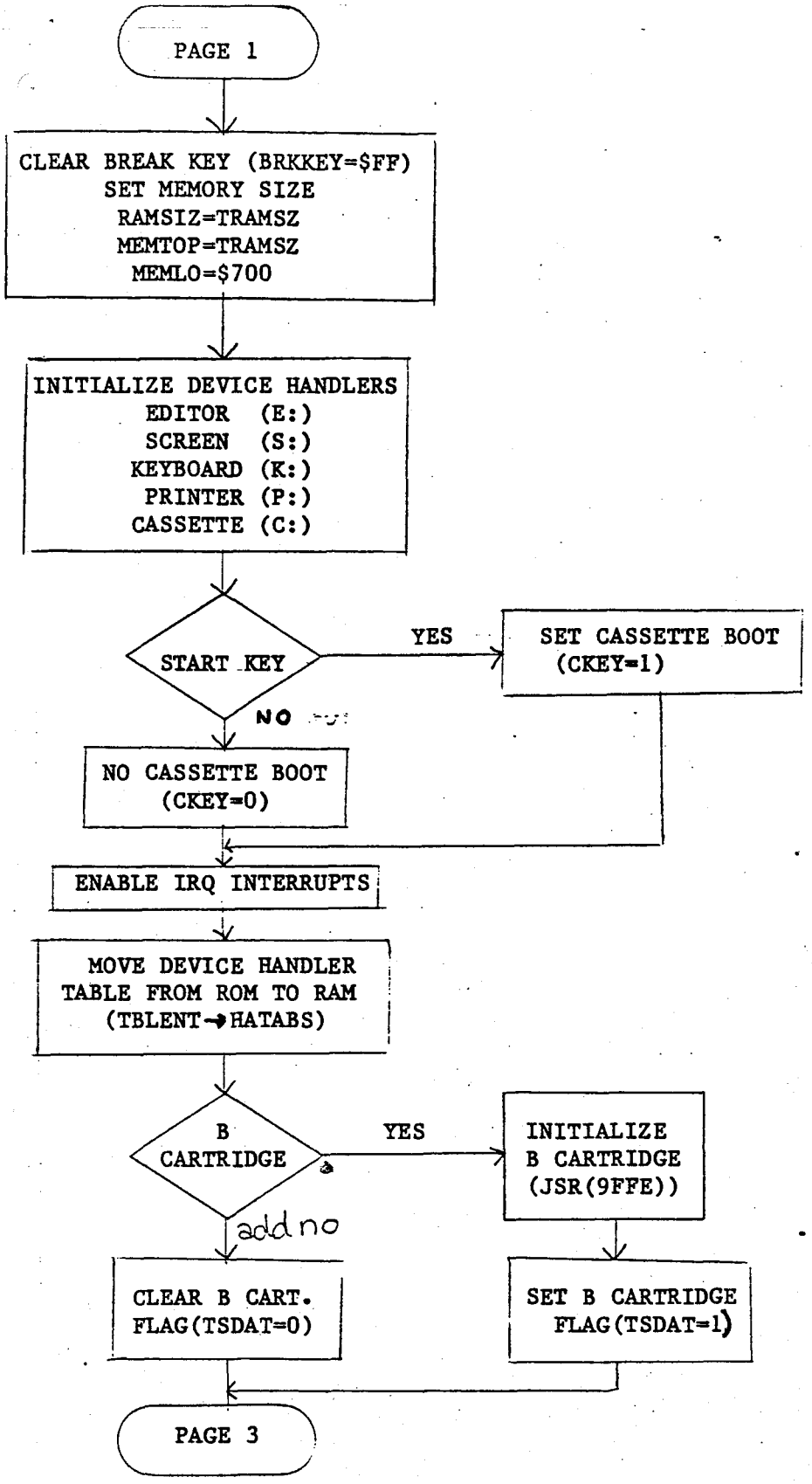
D. Program RAM
1. Memory Map
a. MEMLO
b. MEMTOP
c. APPMHI
d. RAMTOP

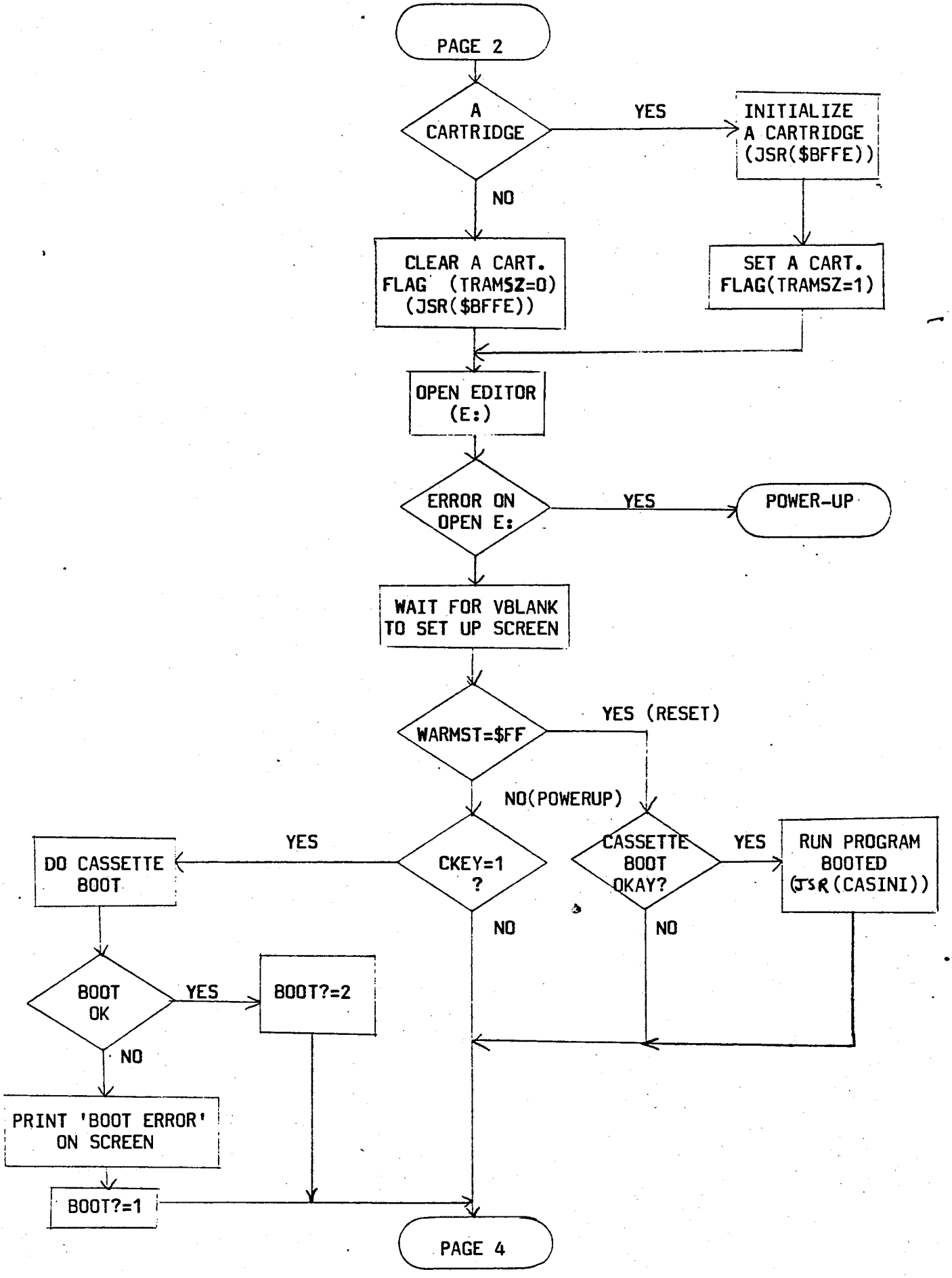
E. Interrupt Handlers
1. NMI's
a. SYSTEM RESET - Non-Maskable
b. DLI
1. VDSLST - Display List Vector
c. VBLANK
1. Immediate (Stage 1)
a. VVBLKI - Vector
b. Critical Sections
1). SEI
2). CRITIC
c. Stage 2
1). Shadows
2. Deffered
a. VVBLKD - normally points to RTI

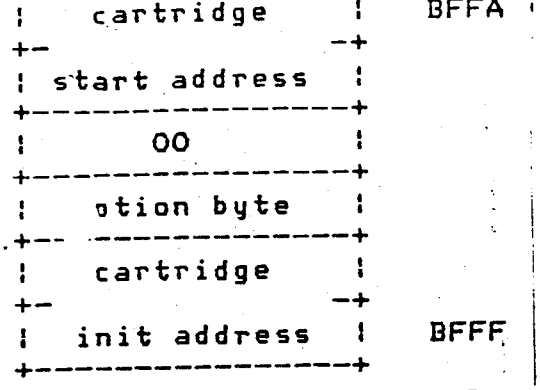
2. IRQ's
a. One Bit Mask (SEI/CLI)
b. IRQEN and POKMSK
c. The IRQ's Vectors and Their uses
1. VMIRQ - system IRQ Vector
2. VBREAK - Software BRK instr.
3. VKEYBD - Key board interrupt
4. VSERIN - Serial Bus Input Ready
5. VSEROR - Serial Bus Output Ready
6. VSEROC - Serial Bus Complate
7. VTIMR1 - Pokey Timer 1
8. VTIMR2 - Pokey Timer 2
9. VTIMR4 - Pokey Timer 4
10. CDTMA1 - System Timer 1
11. CDTMA2 - System Timer

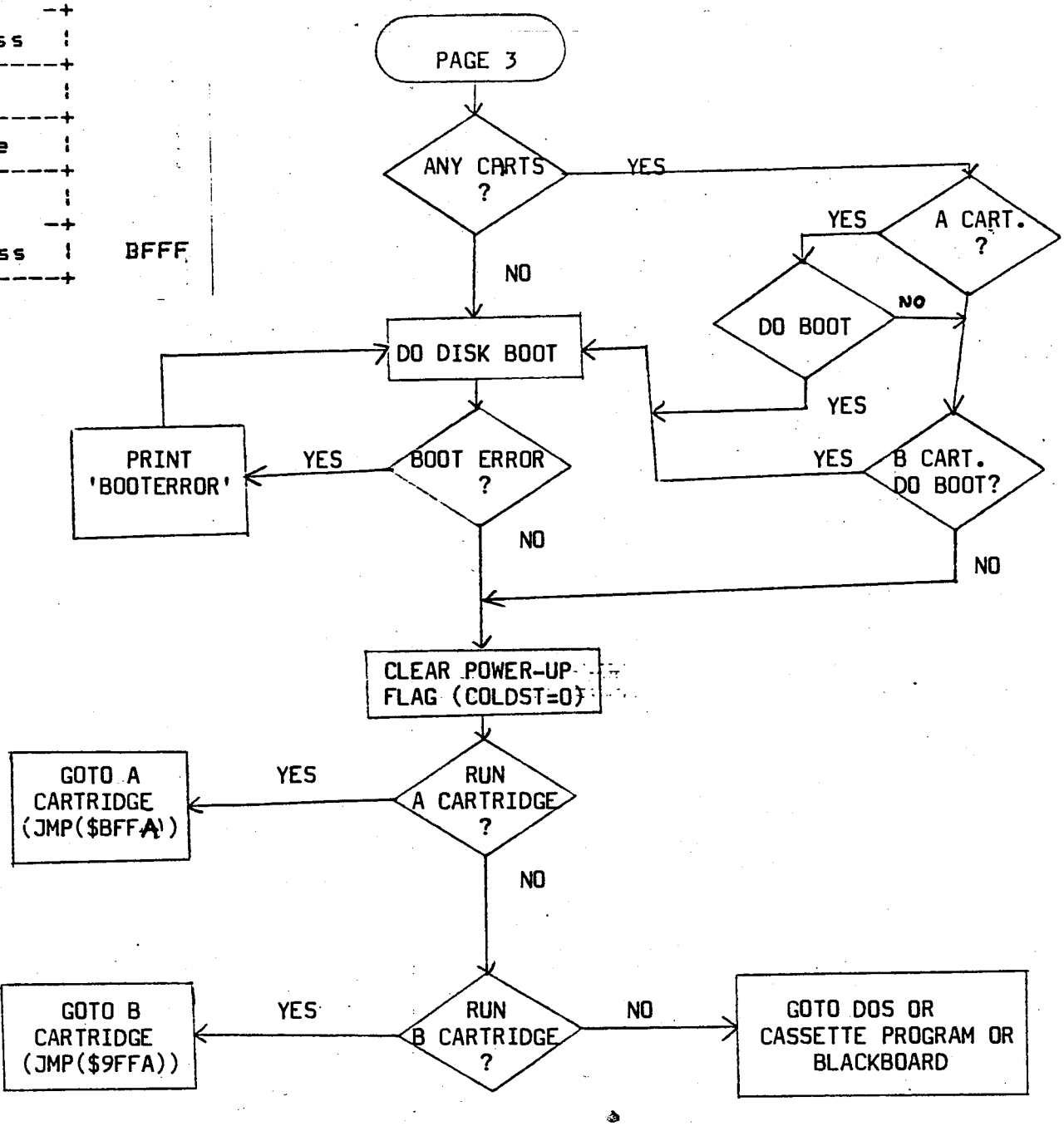
F. Timers
1. Real Time Clock (RTCLOK)
a. 3- byte Frame Counter
2. System Timers
a. CDTMV1 - CDTMA1
b. CDTMV2 - CDTMA2
c. CDTMV3 - CDTMF3
d. CDTMV4 - CDTMF4
e. CDTMV5 - CDTMA5


RESET
POWER-UP
SEI
NO (MIDDLE OF COLDSTART)
COLDST=0
YES
WARMST=$FF
SEI
WARMST=0
DIAGNOSTIC CARTRIDGE
YES
GOTO STANDALONE DIAGNOSTIC CARTRIDGE (JMP ($BFFE))
NO
FIND # OF 4K BLOCKS OF RAM (TRAMSZ)
INITIALIZE POKEY, ANTIC, CTIA/GTIA
WARMST=0
(RESET) NO
CLEAR O.S. RAM ($200-3FF,$10-7F)
YES(POWERUP)
CLEAR ALL RAM ($08-TRAMSZ)
SET DEFAULT CARTRIDGE TO BLACKBOARD (DOSVEC=BLKBDV)
SET COLDST TO MIDDLE OF POWER-UP (COLDST=$FF)
SET SCREEN MARGINS (LMARGN=2 RMARGN=39)
MOVE IRQ VECTOR TABLE FROM ROM TO RAM (VDSLST-VVBLKD)
PAGE 2


PAGE 1
CLEAR BREAK KEY (BRKKEY=$FF)
SET MEMORY SIZE
RAMSIZ=TRAMSZ
MEMTOP=TRAMSZ
MEMLO=$700
INITIALIZE DEVICE HANDLERS
EDITOR (E:)
SCREEN (S:)
KEYBOARD (K:)
PRINTER (P:)
CASSETTE (C:)
START KEY
YES
SET CASSETTE BOOT
(CKEY=1)
NO
NO CASSETTE BOOT
(CKEY=0)
ENABLE IRQ INTERRUPTS
MOVE DEVICE HANDLER
TABLE FROM ROM TO RAM
(TBLENT→HATABS)
B
CARTRIDGE
YES
INITIALIZE
B CARTRIDGE
(JSR(9FFE))
add no
CLEAR B CART.
FLAG(TSDAT=0)
SET B CARTRIDGE
FLAG(TSDAT=1)
PAGE 3


PAGE 2
A CARTRIDGE
YES
INITIALIZE A CARTRIDGE (JSR($BFFE))
NO
CLEAR A CART. FLAG (TRAMSZ=0) (JSR($BFFE))
SET A CART. FLAG(TRAMSZ=1)
OPEN EDITOR (E:)
ERROR ON OPEN E:
YES
POWER-UP
WAIT FOR VBLANK TO SET UP SCREEN
WARMST=$FF
YES (RESET)
NO(POWERUP)
CKEY=1 ?
YES
DO CASSETTE BOOT
CASSETTE BOOT OKAY?
YES
RUN PROGRAM BOOTED (JSR(CASINI))
NO
NO
BOOT OK
YES
BOOT?=2
NO
PRINT 'BOOT ERROR' ON SCREEN
BOOT?=1
PAGE 4





The byte of "00" is used to allow the OS to determine when a cartridge is inserted; locations BFFC and 9FFC will not read zero when there is neither RAM at those locations nor a cartridge inserted. RAM is differentiated from a cartridge by its ability to be altered.

The option byte has the following option bits:

Bit-0 = 0, then do not boot the disk.
1, then boot the disk.

Bit-2 = 0, then init but do not start the cartridge.
1, then init and start the cartridge.

Bit-7 = 0, then cartridge is not a diagnostic cartridge.
1, then cartridge is a diagnostic cartridge & control will be given to the cartridge before any of the OS is initialized (JMP (BFFE)).

IOCB CHART

CALL	ICHID	ICDNO	ICCOM	ICSTA	ICBAL	ICBAH	ICPTL	ICPTH	ICBLL	ICBLH	ICAX1	ICAX2
OPEN FILE-READ	X	X	3	note 1	$80	06	X	X	X	X	4	0
OPEN FILE-WRITE	X	X	3	note 1	$80	06	X	X	X	X	8	note 2
GET BYTES	X	X	7	note 1	00	06	X	X	$80	00	X	X
PUT BYTES	X	X	11	"	00	06	X	X	$80	00	X	X
GET RECORD	X	X	5	"	00	06	X	X	$80	00	X	X
PUT RECORD	X	X	9	"	00	06	X	X	$80	00	X	X
CLOSE FILE	X	X	$C	"	X	X	X	X	X	X	X	X
STATUS	X	X	$D	"	X	X	X	X	X	X	X	X

NOTE 1 - The status of the I/O command is stored here and in the Y REG. on return from CIO

NOTE 2 - The AUXilary bytes of the IOCB's are used by some handlers to indicate special modes

GENERAL NOTE: THE ABOVE IOCB definitions assume

```
*=$600
IOBUFF  .RES   80
FILE    .BYTE  'D:MYPROG.BAS'
```

(- indicates ignore but do not change the current value

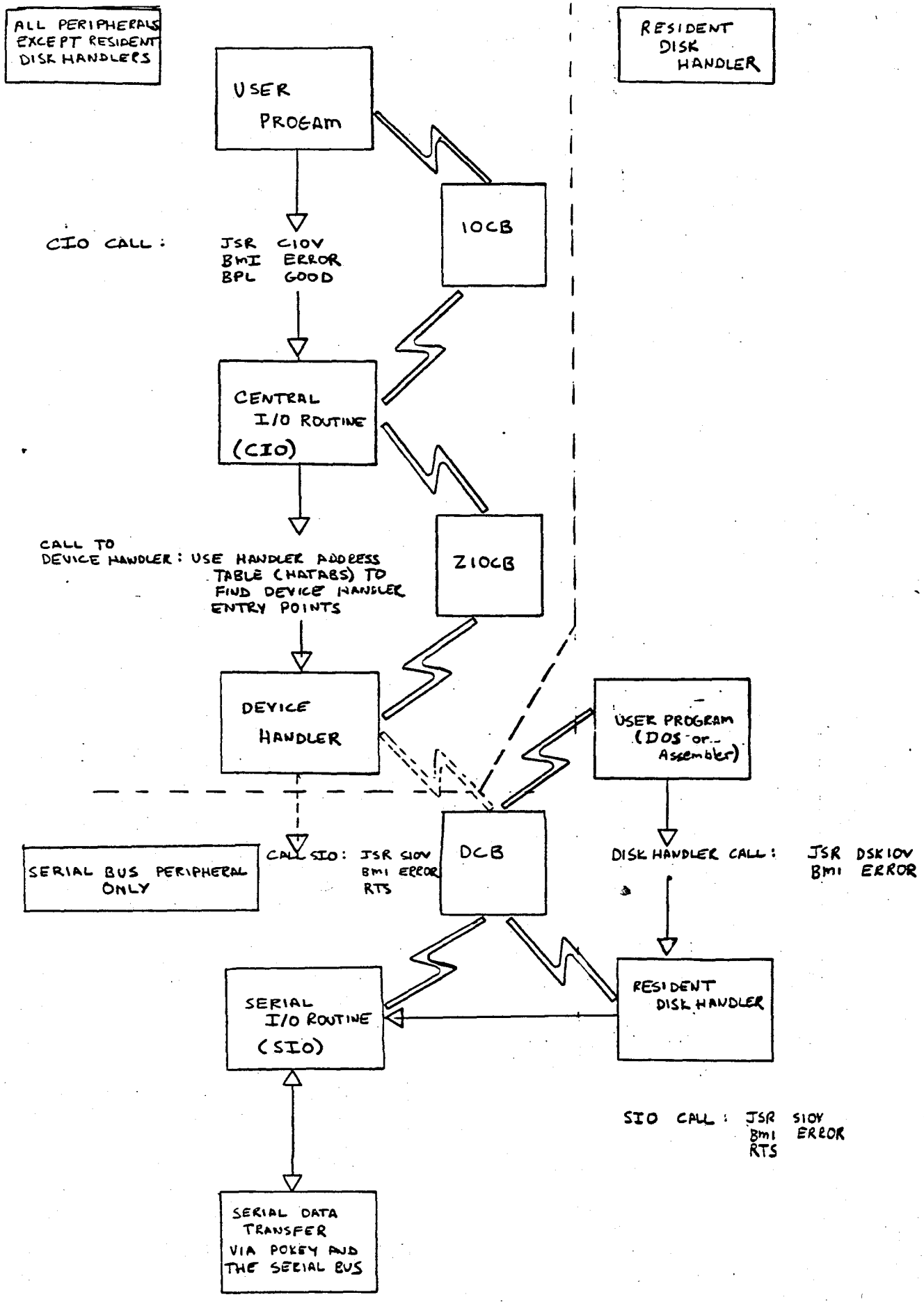
IOCB CHART

HANDLER ADDRESS TABLE

```
E430            PRINTV  =       $E430
E440            CASETV  =       $E440
E400            EDITRV  =       $E400
E410            SCRENV  =       $E410
E420            KEYBDV  =       $E420
                ;
0000                    *=      $031A
                ;
                HATABS
031A 50                 .BYTE   'P      '
031B 30E4               .WORD   PRINTV
031D 43                 .BYTE   'C      '
031E 40E4               .WORD   CASETV
0320 45                 .BYTE   'E      '
0321 00E4               .WORD   EDITRV
0323 53                 .BYTE   'S      '
0324 10E4               .WORD   SCRENV
0326 4B                 .BYTE   'K      '
0327 20E4               .WORD   KEYBDV
0329 00                 .BYTE   0
032A 00                 .BYTE   0,0
032B 00
032C 00                 .BYTE   0
032D 00                 .BYTE   0,0
032E 00
032F 00                 .BYTE   0
0330 00                 .BYTE   0,0
0331 00
0332 00                 .BYTE   0
0333 00                 .BYTE   0,0
0334 00
0335 00                 .BYTE   0
0336 00                 .BYTE   0,0
0337 00
0338 00                 .BYTE   0
0339 00                 .BYTE   0,0
033A 00
033B 00                 .BYTE   0
033C 00                 .BYTE   0,0
033D 00
```

```
                ;
                ;       PRINTER HANDLER ENTRY POINTS
                ;
                ;
                ;
                ;
                *=$E430
                ;
E430  9E EE             .WORD   PHOPEN-1        PRINTER HANDLER OPEN
E432  DB EE             .WORD   PHCLOS-1        PH CLOSE
E434  9D EE             .WORD   BADST-1     PH READ
E436  A6 EE             .WORD   PHWRIT-1        PH WRITE
E438  80 EE             .WORD   PHSTAT-1        PH STATUS
E43A  9D EE             .WORD   BADST-1     PH SPECIAL
E43C  4C 78 EE          JMP     PHINIT          PH INIT.
E43F  00                .BYTE   0               ROM FILLER
```

I/O Subsystem Structure



```
                10 ;WRITTEN BY...MICHAEL EKBERG
                20 ;
600             30 START    =    $600
00(             40 DOSINI   =    $0C
2E7             50 MEMLO    =    $2E7
000             60 NEWMEM   =    $3000     ALTER THIS TO GET SIZE
                70 ;THIS ROUTINE RESERVES SPACE FOR
                80 ;  ASSEMBLY ROUTINES BY RESETTING
                90 ;  THE MEMLO POINTER. IT RUNS AS
               0100 ;  AN AUTORUN.SYS FILE. IT ALSO
               0110 ;  RESETS MEMLO ON [RESET]. MEMLO
               0120 ;  IS SET TO THE VALUE OF NEWMEM.
               0130 ;
               0140 ;THIS PART IS PERMANENT, IE. NEEDS
               0150 ;  TO BE RESIDENT. THE SYSTEM DOS INIT VECTOR
               0160 ;  HAS BEEN STOLEN, AND STORED IN
               0170 ;  THE LOCATION INITDOS+1&2.
               0180 ;  DOS IS INITIALIZED AND MEMLO  IS INITIALIZED
               0190 ;  INITDOS EXECUTES ON [RESET].
000            0200        *=   START
               0210 INITDOS
600 200D06     0220        JSR  ENDRTS    ;DO DOS INITLIST
603 A900       0230        LDA  #NEWMEM & 255
605 8DE702     0240        STA  MEMLO
608 A930       0250        LDA  #NEWMEM/256
60A 8DE802     0260        STA  MEMLO+1
               0270 ENDRTS
60D 60         0280        RTS
               0290 ; THIS PART IS EXECUTED AT POWER
               0300 ;   UP ONLY AND CAN BE DELETED
               0310 ;   AFTER POWER-UP.
               0320 ; THIS ROUTINE STORES THE
               0330 ;   CONTENTS OF DOSINI INTO A JSR
               0340 ;   AT LOCATION INITDOS+1. IT
               0350 ;   THEN REPLACES DOSINI WITH
               0360 ;   IT'S OWN VALUE, THE LOCATION
               0370 ;   INITDOS.
               0380 BEGIN
60E A50C       0390        LDA  DOSINI       SAVE DOSINI
610 8D0106     0400        STA  INITDOS+1
613 A50D       0410        LDA  DOSINI+1
615 8D0206     0420        STA  INITDOS+2
618 A900       0430        LDA  #INITDOS&255 SET DOSINI
61A 850C       0440        STA  DOSINI
61C A906       0450        LDA  #INITDOS/256
61E 850D       0460        STA  DOSINI+1
620 A900       0470        LDA  #NEWMEM&255  SET MEMLO
622 8DE702     0480        STA  MEMLO
625 A930       0490        LDA  #NEWMEM/256
627 8DE802     0500        STA  MEMLO+1
62A 60         0510        RTS



62B            0520        *=   $2E2
02E2 0E06      0530        .WORD BEGIN    SET RUN ADDRESS
 2E4           0540        .END
```