



```

;
; PREFORM I/O ON THE DATA BYTE(S).
;
1518      PIOD      EQU      S      ;WRITE DATA
1518 0C      INR      C      ;BUMP UP THE DATA COUNT IN CASE C=0
1519 219616   LXI      H,IOT    ;GET THE BASE ADDR OF THE I/O ROUTINE
; JUMP TABLE.
151C 78      MOV      A,B      ;GET THE STANDARD FLAG
151D 07      RLC      ;COMPUTE THE JUMP TABLE OFFSET
151E CD4815   CALL     CJADR    ;COMPUTE THE ADDRESS OF THE SELECTED ROUTINE.
1521 79      MOV      A,C      ;PUT THE DATA LENGTH IN A.
1522 D1      POP      D      ;GET THE DATA ADDRESS.
1523 1B      DCX      D      ;SET UP ADDR FOR FIRST PASS OF LOOP.
1524 C5      PUSH     B      ;SAVE INTERNAL STATUS.
1525 012815   LXI      B,IORET  ;GET COMMON RETURN ADDRESS.
1528      IORET    EQU      S      ;COMMON I/O ROUTINE RETURN.
1528 13      INX      D      ;POINT TO NEXT BYTE TO BE OUTPUT
1529 3D      DCR      A      ;REDUCE THE LENGTH OF DATA LEFT TO BE TRANSMITTED OR RECEIVED.
152A CA2F15   JZ       IOCLUP   ;IS THERE ANY DATA TO BE TRANSMITTED OR RECEIVED?? - IF NOT
; THEN EXIT THE I/O LOOP.
152D C5      PUSH     B      ;PLACE A FRESH RETURN ADDR ON THE STACK.
152E E9      PCHL     ;TRANSFER CONTROL TO SELECTED CHARACTER I/O
; ROUTINE. ALL THE ROUTINES BRANCHED FROM THIS POINT
; RETURN TO IORET.
152F      IOCLUP  EQU      S      ;I/O ROUTINES CLEANUP
152F C1      POP      B      ;RESTORE INTERNAL STATUS.
;
; PROCESS BLOCK TRAILER IF REQUIRED.
;
1530      PTFLR   EQU      S      ;PROCESS TRAILER.
1530 97      SUE     A      ;CLEAR AC & CY.
1531 80      ADD     B      ;GET STANDARDS FLAGS.
1532 07      RLC      ;TEST FOR BLOCKED RECORDS & COMPUTE OFFSET.
1533 E24515   JNC     DONE    ;IF RECS NOT BLOCKED THEN FALL THRU
; TO COMPLETION.
1536 219E16   LXI      H,TRLT  ;GET THE BASE ADDR OF THE BLOCK TRAILER ROUTINE
; JUMP TABLE.
1539 D5      PUSH     D      ;HANG ON TO DATA ADDR
153A CD4815   CALL     CJADR    ;COMPUTE THE ADDRESS OF THE SELECTED ROUTINE.
153D D1      POP      D      ;GET DATA ADDR.
153E C5      PUSH     B      ;SAVE INTERNAL STATUS.
153F 014415   LXI      B,TRET    ;GET COMMON RETURN ADDRESS
1542 C5      PUSH     B      ;FAKE A CALL
1543 E9      PCHL     ;TRANSFER CONTROL TO THE SELECTED BLOCK TRAILER
;CONTROL WILL RETURN TO THE NEXT STATEMENT.
;TRAILER COMMON RETURN POINT.
1544      TRET    EQU      S      POPB
1544 C1      POP     B      ;HERE AT END OF OUTPUT
1545      DONE   EQU      S      ;RESTORE A AND FLAGS IF NO ERROR.
1545 AF      XRA     A      ;ENTER ON ERROR RETURN
1546      EDONE  EQU      S      ;
1546 D1      POP     D      ;
1547 C9      RET     ;RETURN TO USER TASK.

```

```

;
;
; SUBROUTINES .
;
1548 CJADR EQU $ ;COMPUTE JUMP ADDRESS SUBROUTINE.
1548 E6FE ANI CE10 ;CLEAR BIT 0 OF THE ACCUMULATOR.
154A 1600 MVI D,0 ;CLEAR HIGH ORDER BYTE FOR UPCOMMING ADD.
154C 5F MOV E,A ;GET THE OFFSET INTO THE TABLE.
154D 19 DAD D ;COMPUTE THE ADDRESS OF THE JUMP TABLE ENTRY.
154E 5E MOV E,M ;GET THE LOW ORDER BYTE OF THE SUBROUTINES ADDRESS.
154F 23 INX H ;POINT TO THE HIGH ORDER BYTE OF THE ADDR IN THE
; JUMP TABLE.
1550 56 MOV D,M ;GET THE HIGH ORDER BYTE OF SUBROUTINE ADDR.
1551 EB XCHG ;PUT THE COMPLETD ADDR IN HL.
1552 C9 RET ;RETURN TO CALLER.
;
;
1553 HBHW EQU $ ;WRITE BLOCK HEADER FOR THE HIT STANDARD.
1553 D5 PUSH D ;SAVE
1554 E5 PUSH H ; SYSTEM
1555 C5 PUSH B ; STATUS.
1556 11A616 LXI D,SYN ;PICK UP THE ADDR OF THE SYN CHARACTER.
1559 061F MVI B,NSYN ;GET THE NUMBER OF SYN CHARS TO BE PRINTED.
155B WSYNL EQU $ ;WRITE SYN CHARACTER LOOP.
155B CD0816 CALL HWRT ;WRITE OUT THE CHARACTER TO TAPE.
155E 05 DCR B ;REDUCE NUMBER OF CHARS TO BE TRANSMITTED.
155F C25B15 JNZ WSYNL ;ARE THERE ANY CHARACTERS LEFT? IF SO THEN LOOP BACK,
; OTHERWISE FALL THRU.
1562 13 INX D ;HERE ONLY WHEN ALL SYN CHARS HAVE BEEN TRANSMITTED.
; BUMP THE CHARACTER POINTER TO POINT TO THE STX CHAR.
1563 CD0816 CALL HWRT ;WRITE OUT THE STX CHAR TO TAPE.
1566 210000 LXI H,0 ;CLEAR H,L
1569 6C MOV L,H ; REGISTER PAIR.
156A 39 DAD SP ;GET THE STACK POINTER WHICH CURRENTLY POINTS TO THE
; C-REG WHICH HOLDS THE NUMBER OF DATA BYTES TO BE
; OUTPUTED.
156B EB XCHG ;PUT THE ADDRES OF THE CONTENTS OF C IN DE FOR THE WRITE
; CHARACTER ROUTINE.
156C CD0816 CALL HWRT ;WRITE OUT THE DATA BLOCK LENGTH TO TAPE.
156F C1 POP B ;RESTORE
1570 E1 POP H ; SYSTEM
1571 D1 POP D ; STATUS.
1572 C9 RET ;RETURN TO CALLER.

```

```

;
;
1573      HBHR      EQU      $      ;HIT BLOCK HEADER READ SUBROUTINE.
;INPUTS - DE-ADDR OF DATA BUFFER.
;OUTPUTS - C-LENGTH OF DATA IN BLOCK.
;          A-ERROR CODE IF ANY
;SAVE SYSTEM STATUS

1573 C5      PUSH      B
1574 F5      PUSH      PSW
1575 D5      PUSH      D
1576 E5      PUSH      H
1577 0EFF     MVI      C,NONSA ;SET COUNTER OF # OF NON SYNC CHARS ALLOWED.
1579      S4SYN     EQU      $      ;SCAN FOR SYNC'S.
1579 0D      DCR      C      ;BUMP THE COUNT OF NON SYNC CHARS
157A C28415  JNZ      FNDSC ;IF COUNT >0 THEN KEEP
157D 3E01     MVI      A,ERR1 ;HERE ON ERROR, SET ERROR CODE.
157F 1607     MVI      D,HBHRP ;SET NUMBER OF POPS TO RETURN ADDR.
1581 C38716  JMP      EHDLP ;TRANSFER CONTROL TO THE ERROR HANDLER.
1584      FNDSC     EQU      $      ;FIND SYNC CHAR.
1584 CDBB15  CALL     HFS      ;FIND SYNC CHAR
1587 IA      LDAX     D      ;PUT IT IN A
1588 21A616  LXI      H,SYN ;POINT TO A SYNC CHAR.
158B BE      CMP      M      ;IS THE CURRENT CHAR A SYNC?
158C C27915  JNZ      S4SYN ;NO - KEEP WAITING FOR ONE.
;HERE ONLY WHEN FIRST SYNC OF THE HEADER HAS BEEN FOUND.
158F 0E20     MVI      C,NSYN+1
1591      CSYN      EQU      $      ;COUNT THE SYNC'S.
1591 CDC515  CALL     HRD      ;GET NEXT CHAR OFF THE TAPE.
1594 IA      LDAX     D      ;PUT IT IN A.
1595 BE      CMP      M      ;IS IT A SYNC?
1596 C2A415  JNZ      CHNC ;NO - GO FIND OUT WHAT IT IS.
1599 0D      DCR      C      ;YES - HAVE WE COUNTED MORE THAN 32 SYNC'S IN THE HEADER?
159A F29115  JP       CSYN ;NO - KEEP READING IN THOSE SYNC'S.
159D 3E02     MVI      A,ERR2 ;YES - FATAL ERROR PREPARE TO ABORT. LOAD ERROR CODE.
159F 1607     MVI      D,HBHRP ;LOAD NUMBER OF POPS NEEDED TO GET TO THE RETURN ADDR.
15A1 C38716  JMP      EHDLP ;TRANSFER CONTROL TO THE ERROR HANDLER.
15A4      CHNC      EQU      $      ;CHECK NEXT CHARACTER.
15A4 23      INX      H      ;POINT TO AN START -OF-TEXT CHAR.
15A5 EE      CMP      M      ;IS THE CURRENT CHAR AN STX?
15A6 CAB015  JZ       RDLNG ;YES - THEN CONTINUE PROCESSING THE HEADER.
15A9 3E03     MVI      A,ERR3 ;NO - FATAL ERROR. SET ERROR CODE.
15AB 1607     MVI      D,HBHRP ;SET # OF POPS NEEDED TO GET TO THE RETURN ADDR.
15AD C38716  JMP      EHDLP ;TRANSFER CONTROL TO THE ERROR HANDLER.
15B0      RDLNG     EQU      $      ;READ LENGTH OF DATA FIELD.
15B0 CDC515  CALL     HRD      ;READ IN LENGTH
15B3 IA      LDAX     D      ;PUT LENGTH I A .
15B4 4F      MOV      C,A ;SET UP C FOR THE UPCOMING READ OF THE DATA BLOCK.
15B5 E1      POP      H      ;RESTORE STATUS.
15B6 D1      POP      D
15B7 F1      POP      PSW
15B8 C1      POP      B
15E9 C9      RET

;
;
15BA      BBHW      EQU      $      ;BYTE STANDARD BLOCK HEADER WRITE
15BA      BBHR      EQU      $      ;BYTE STANDARD BLOCK HEADER READ.
15BA C9      RET

```

```

;
;
15BB      HFS      EQU      $      ;HIT STD FIND SYNC ROUTINE
15BB F5      PUSH   PSW      ;SAVE SYSTEM STATUS
15BC C5      PUSH   B
15BD E5      PUSH   H
15BE 0600    MVI    B,0      ;SCAN UP TO 256 BITS FOR SYNC
15C0 2E00    MVI    L,0      ;SET SYNC SEARCH FLAG
15C2 C3CC15  JMP    HRDX

;
;
15C5      HRD      EQU      $      ;HIT STANDARD READ ROUTINE.
15C5 F5      PUSH   PSW      ;SAVE
15C6 C5      PUSH   B      ; SYSTEM
15C7 E5      PUSH   H      ;STATUS.
15C8 2EFF    MVI    L,OFFH    ;CLEAR SYNC SEARCH FLAG
15CA 0608    MVI    B,DBPC     ;LOAD LOOP COUNTER WITH THE #
; OF BITS PER CHARACTER
15CC      HRDX     EQU      $      ;COMMON POINT FOR HRD & HFS
15CC AF      XRA    A      ;CLEAR A& CY.
15CD 67      MOV    H,A      ;CLEAR THE BYTE HOLDING REG.
;MAKE SURE WE ARE ON A TRAILING PULSE TO INSURE PROPER BIT FRAME ALIGNMENT
15CE      W4TP     EQU      $      ;WAIT FOR TRAILING PULSE.
15CE DB10    IN     UCRIP    ;SAMPLE THE INPUT SIGNAL
15D0 1F      RAR    ;TEST IT.
15D1 DACE15  JC     W4TP     ;ARE WE CURRENTLY ON A TRAILING PULSE?
; IF SO THEN FALL THRU, OTHERWISE KEEP LOOPING
;PROPERLY ALIGNED, WAIT FOR THE FIRST LEADING PULSE OF THE DATA.
15D4      GNBL     EQU      $      ;GET NEXT BIT LOOP.
15D4 0E00    MVI    C,ZEROE    ;CLEAR THE PULSE WIDTH COUNTER.
15D6      W4LPL    EQU      $      ;WAIT FOR LEADING PULSE LOOP.
15D6 DB10    IN     UCRIP    ;SAMPLE THE SIGNAL
15D8 1F      RAR    ;TEST IT.
15D9 D2D615 JNC    W4LPL    ;HAS THE LEADING EDGE OF THE NEXT PULSE ARRIVED?
; IF SO THEN FALL THRU, ELSE KEEP WAITING.
;THE BEGINING OF A SIGNAL HAS BEEN CAPTURED! NOW IS THE TIME TO START COUNTING.
15DC      WOLPL    EQU      $      ;WAIT ON LEADING PULSE LOOP.
15DC 0C      INR    C      ;BUMP UP THE PULSE LENGTH COUNTER.
15DD DB10    IN     UCRIP    ;SAMPLE THE SIGNAL.
15DF 1F      RAR    ;TEST IT.
15E0 DADC15  JC     WOLPL    ;IS THE SIGNAL PRESENT?
; IF SO THEN KEEP COUNTING, ELSE FALL THRU.
15E3      DBSL     EQU      $      ;DETERMINE BIT STATE LOOP.
15E3 0D      DCR    C      ;BUMP DOWN THE PULSE LENGTH COUNTER.
15E4 CAED15  JZ     BSD      ;HAVE WE COUNTED THE PULSE ALL THE WAY DOWN?
; IF SO THEN THE PULSE WIDTH IS SHORTER THAN
; 1/2 THE BIT FRAME SIZE AND THE DATA BIT IS
; A "0"
15E7 DB10    IN     UCRIP    ;SAMPLE THE SIGNAL.
15E9 1F      RAR    ;TEST IT
15EA D2E315 JNC    DBSL     ;HAS THE NEXT LEADING PULSE ARRIVED?
; IF SO THEN THE PULSE IS > 1/2 THE
; BIT FRAME SIZE AND THE DATA BIT IS A "1",
; ELSE KEEP COUNTING DOWN.

```

```

15ED      BSD      EQU      $      ;BIT STATE DETERMINED.
15ED 7C      MOV      A,H      ;GET THE BYTE UNDER CONSTRUCTION.
15EE 1F      RAR      ;STUFF IN THE NEWLY DECODED BIT.
15EF 67      MOV      H,A      ;PUT IT BACK.
15F0 3AA616  LDA      SYN      ;GET SYN CHAR
15F3 94      SUB      H      ;SUBTRACT OUT CURRENT CHAR
15F4 B5      ORA      L      ;OR IN SYNC SEARCH FLAG TO CALCULATE
                        ;(FOUND SYNC).AND.(LOOKING FOR SYNC)
15F5 CAFD15  JZ      W4SBL  ;IF TRUE, GO GET STOP BIT
15F8 05      DCR      B      ;BUMP DOWN THE BITS PER BYTE COUNTER.
15F9 C2D415  JNZ      GNLB   ;HAVE WE ASSEMBLED A COMPLETE DATA BYTE?
                        ; IF SO THEN FALL THRU , ELSE GO GET NEXT BIT.
15FC      W4SBL  EQU      $      ;WAIT FOR STOP BIT LOOP.
15FC DB10    IN      UCRIP  ;SAMPLE THE SIGNAL.
15FE 1F      RAR      ;TEST IT.
15FF D2FC15  JNC      W4SBL  ;HAS THE BEGINING OF THE STOP BIT PULSE ARRIVED?
                        ; IF SO THEN FALL THRU , ELSE WAIT FOR IT.
;HERE ON COMPLETION OF BYTE ASSEMBLY
1602 7C      MOV      A,H      ;GET THE COMPLETE BYTE
1603 12      STAX   D      ;PUT IN THE BUFFER.
1604 E1      POP     H      ;RESTORE
1605 C1      POP     B      ;      SYSTEM
1606 F1      POP     PSW   ;      STATUS.
1607 C9      RET     ;BYE BYE
;
;
1608      HWRT   EQU      $      ;WRITE CHARACTER ACCORDING TO THE HIT STANDARD.
1608 F5      PUSH   PSW   ;SAVE
1609 C5      PUSH   B      ;      CURRENT
160A D5      PUSH   D      ;      SYSTEM
160B E5      PUSH   H      ;      STATUS.
160C 1A      LDAX   D      ;PUT THE DATA BYTE IN THE ACCUMULATOR.
160D 0609    MVI    B,BPC  ;LOAD COUNTER TO LOOP THRU EACH BIT
160F A7      ANA    A      ;CLEAR THE CARRY, TO SET THE STOP BIT=0.
1610      BSLP   EQU      $      ;BIT STRIPPING LOOP.
1610 0E05    MVI    C,ZPL  ;SET THE TRANSMISSION TIMER TO IT'S DEFAULT
                        ; VALUE OF A ZERO PULSE.
1612 1F      RAR      ;PUT THE NEXT BIT TO BE TESTED IN THE CARRY.
1613 D21816  JNC    ZEROB  ;IS THE CURRENT BIT A ONE? - IF
1616 0E0F    MVI    C,OPL  ;YES - OVERRIDE THE DEFAULT TRANSMISSION TIME
1618      ZEROB  EQU      $      ;BRANCH HERE DIRECTLY ON DATA BIT = 0.
1618 A7      ANA    A      ;CLEAR CARRY FOR I/O TOGGLING.
1619 1618    MVI    D,HCPBF ;INITIALIZE THE BIT FRAME COUNTER TO THE NUMBER
                        ; OF HALF CYCLES IN THE BIT FRAME.
;THE FOLLOWING LOOP HCLP (HALF CYCLE LOOP) IS CRITICALLY TIMED.
;EACH PASS THRU IT SHOULD TAKE ONE HALF CYCLE OF THE TRANSMISSION FREQUENCY.
;THIS PROGRAM ASSUMES A TRANSMISSION FREQUENCY OF 2000HZ, YEILD A CYCLE
; TIME OF 500US. THEREFORE THE HALF CYLE TIME IS 250US OR 500THETA1 CYLES WHERE
; THETA1 IS ASSUMED TO BE 500NS IE THIS LOOP SHOULD CONSUME 500 INSTRUCTION CYCLES
;EACH COMMENT IS PREFACE WITH A PAIR OF NUMBERS. THE FIRST NUMBER IS THE
; NUMBER OF CYCLES THAT INSTRUCTION TAKES AND THE SECOND IS THE NUMBER OF WAIT
; CYCLES INCLUDED IN THE TOTAL. IT WAS ASSUMED THAT EACH MEMORY ACCESS TOOK
; ONE WAIT CYCLE.

```

```

;
161B      HCLP   EQU    $      ;HALF CYCLE LOOP.
161B 1E0B   MVI    E,WAITC ;9-2 SET THE WAIT TIMER TO ITS DEFAULT VALUE .
161D 0D     DCR    C        ;6-1 REDUCE THE NUMBER OF TONE PULES LEFT TO BE SENT.
161E FA3716 JM     PTS      ;13-3 ARE THERE ANY PULSES LEFT TO BE TRANSMITTED?
1621 3F     CMC    ;5-1 YES TOGGLE THE BIT.
1622 17     RAL    ;5-1 MOVE IT TO DO FOR TRANSMISSION.
1623 67     MOV    H,A     ;SAVE A
1624 E601   ANI    01     ;IS THE BIT SET
1626 C22E16 JNZ   SBITS    ;YES SO SET THEM BOTH
1629 3E00   MVI    A,0     ;NO TURN BOTH OFF
162B C33016 JMP   SBOUT    ;GO OUTPUT THEM
162E       EQU    $      ;SET BITS
162E 3E03   MVI    A,3     ;TURN ON THE 2 BITS
1630       SBOUT  EQU    $      ;SEND BITS OUT TO CRI
1630 D310   OUT   UCRIP  ;11-1 TRANSMIT THE BIT TO THE UCRI BOARD.
1632 7C     MOV    A,H     ;RESTORE A
1633 1F     RAR    ;5-1 PUT THE BIT BACK.
1634 C33E16 JMP   WAIT     ;13-3 GO WAIT FOR THE NEXT HALF CYCLE TO ROLL AROUND.
1637       PTS   EQU    $      ;PULSE TRANSMISSION SKIPPED.
1637 67     MOV    H,A     ;SAVE A
1638 3E02   MVI    A,2     ;SET THE BITS IN OPPOSITION.
163A D310   OUT   UCRIP  ;SEND THEM OUT TO THE OLD CRI BORED.
163C 7C     MOV    A,H     ;RESTORE A
163D 1C     INR    E        ;6-1 INCREASE THE WAIT TIME TO REFLECT THE TIME A PULSE
;          ;          TRANSMISSION WOULD HAVE TAKEN.
163E       WAIT  EQU    $      ;WAIT LOOP - EACH PASS THROUGH THE LOOP IS EQUAL TO THE
;          ;          DIFFERENCE IN TIME BETWEEN TRANSMITTING THE PULSE
;          ;          AND NOT TRANSMITTING IT.
163E 00     NOP                ;5-1 WASTE
163F 66     MOV    H,M     ;9-2      14 MACHINE CYCLES.
1640 1D     DCR    E        ;6-1 IS OUR TIME UP?
1641 C23E16 JNZ   WAIT     ;13-3 IF NOT THEN KEEP WAITTING.
1644 5B     MOV    E,E     ;6-1 WASTE
1645 5B     MOV    E,E     ;6-1      18
1646 5B     MOV    E,E     ;6-1 MACHINE CYCLES.
1647 15     DCR    D        ;6-1 BUMP DOWN 1/2 CYCLE COUNTER.
1648 C21B16 JNZ   HCLP    ;13-3 HAVE ALL THE 1/2 CYCLES OF THE CURRENT FRAME BEEN PROCESSED.
;          ;          IF NOT THEN KEEP LOOPING.
164B 05     DCR    B        ;REDUCE THE THE DATA BIT COUNT
164C C21016 JNZ   BSLP    ;IF ALL THE DATA BITS HAVE BEEN PROCESSED THEN FALL
;          ;          THRU OTHERWISE KEEP LOOPING.
;HERE ONLY WHEN ENTIRE CHARACTER HAS BEEN PROCESSED.
164F E1     POP    H        ;RESTORE
1650 D1     POP    D        ;          SYSTEM
1651 C1     POP    B        ;          STATUS.
1652 F1     POP    PSW
1653 C9     RET                ;RETURN TO CALLER.
;
;
1654       BWRT  EQU    $      ;WRITE CHARACTER ACCORDING TO THE BYTE STANDARD.
1654       BRD   EQU    $      ;BYTE STANDARD READ CHARACTER SUBROUTINE.
1654 C9     RET

```

```

;
;
1655      HBTW   EQU    $      ;WRITE BLOCK TRAILER FOR THE HIT STANDARD.
1655 D5    PUSH   D          ;SAVE THE STATUS OF THE D-REG.
1656 11A816 LXI    D,ETX        ;GET THE ADDR OF THE END-OF-TEXT CHAR.
1659 CD0816 CALL   HWRT         ;WRITE OUT THE ETX TO TAPE
;HERE TO WRITE OUT THE ERROR CHECKING CODE IF ONE IS EMPLOYED. DEFAULT IS
; NO ERROR CHECKING. IF AN ERROR CHECKING CODE IS DESIRED THE USER SHOULD
; CUSTOM PROGRAM WHATEVER ALGORITHM HE SO DESIRES AND PLACE THE CODE TO WRITE
;PROVISION IS MADE FOR A 2-BYTE CHECK

165C 11A916      LXI    D,ZEROC ;PICKUP THE ADDRESS OF A BYTE OF 00H
165F CD0816      CALL   HWRT ;WRITE 2 BYTES
1662 CD0816      CALL   HWRT ;OF 00H.
1665 D1          POP    D          ;RESTORE D-REG.
1666 C9          RET     ;RETURN TO CALLER.
1667      HBTR   EQU    $      ;HIT BLOCK TRAILER READ ROUTINE.
;INPUTS - DE-DATA BUFFER ADDR
;OUTPUTS - HL-HAS ERROR CHECK CODE & RETURN CODES IN A
1667 E5          PUSH   H          ;SAVE STATUS.
1668 CDC515      CALL   HRD         ;GET CHAR
166B 21A816      LXI    H,ETX        ;POINT TO AN END-OF-TEXT CHAR.
166E 1A          LDAX  D          ;PUT CHAR IN A.
166F BE          CMP    M          ;IS IT AN ETX?
1670 CA7A16      JZ     ETXF        ;YES - CONTINUE PROCESSING THE TRAILER.
1673 3E04        MVI    A,ERR4      ;NO - FATAL ERROR. SET ERROR CODE.
1675 1602        MVI    D,HBTRP     ;SET # OF POPS NEEDED TO GET TO RETURN ADDR.
1677 C38716      JMP    EHDLR       ;TRANSFER CONTROL TO THE ERROR HANDLER.
167A      ETXF  EQU    $      ;ETX FOUND.
167A CDC515      CALL   HRD         ;GET CHARACTER
167D 1A          LDAX  D          ; PUT IN A
167E 67          MOV    H,A         ;PUT HIH ORDER BYTE OF ERROR CHECK IN H.
167F CDC515      CALL   HRD         ;GET LOW ORDER BYTE OF ERROR CHECK.
1682 1A          LDAX  D          ;PUT IT IN A.
1683 6F          MOV    L,A         ;PUT LOW ORDER BYTE IN L.
1684 E1          POP    H          ;RESTORE STATUS.
1685 C9          RET

;
;
1686      BBTW   EQU    $      ;WRITE BLOCK TRAILER FOR THE BYTE STANDARD RET
1686      BBTR   EQU    $      ;BYTE STANDARD BLOCK TRAILER READ.
1686 C9          RET

;
;
1687      EHDLR EQU    $      ;COMMON ERROR HANDLER.
; THIS LOOP POPS THE NUMBER OF ADDRS REQUESTED +1 TO
; PREVENT THE A FROM BEING RESET BY THE HIGHEST LEVEL RETURN.
1687 E1          POP    H          ;CLEAR THE STACK FOR THE UPCOMMING RETURN.
1688 15          DCR    D          ;BUMP DOWN THE POP COUNTER.
1689 C28716      JNZ   EHDLR       ;ARE THERE STILL ADDRS LEFT TO POPPED?
168C 37          STC     ;SET CY AS ERROR FLAG.
168D C9          RET     ;GO BACK TO THE USER.

```



```

;
;
;   J U M P   T A B L E S .
;
;
168E      HDRT      EQU      $           ;BLOCK HEADER ROUTINES
168E 5315      DW       HBHW          ;HIT STANDARD WRITE.
1690 7315      DW       HBHR          ;HIT STANDARD READ.
1692 BA15      DW       BBHW          ;BYTE STANDARD WRITE.
1694 BA15      DW       BBHR          ;BYTE STANDARD READ.
;
1696      IOT       EQU      $           ;DATA CHARACTER WRITTING ROUTINE
1696 0816      DW       HWRT          ;HIT STANDARD WRITE.
1698 C515      DW       HRD           ;HIT STANDARD READ.
169A 5416      DW       BWRT          ;BYTE STANDARD WRITE.
169C 5416      DW       BRD           ;BYTE STANDARD READ.
;
169E      TRLT      EQU      $           ;BLOCK TRAILER WRITTING ROUTINES
169E 5516      DW       HBTW          ;HIT STANDARD WRITE.
16A0 6716      DW       HBTR          ;HIT STANDARD READ.
16A2 8616      DW       BBTW          ;BYTE STANDARD.
16A4 8616      DW       BBTR          ;BYTE STANDARD READ.
;
;
;   C O N S T A N T S .
;
;
16A6      SYN       EQU      $           ;ASCII SYNC CHARACTER
16A6 16        DB       16H
16A7      STX       EQU      $           ;ASCII START-OF-TEXT CHARACTER.
16A7 02        DB       02H
16A8      ETX       EQU      $           ;ASCII END-OF-TEXT CHARACTER.
16A8 03        DB       03H
16A9      ZEROC    EQU      $           ;ZERO BYTE.
16A9 00        DB       00H
;
;
;   E Q U A T E S .
;
;
001F      NSYN     EQU      31D          ;THE # OF SYN CHARS TO BE PRINTED - 1.
0009      BPC      EQU      9           ;BITS PER CHARACTER.
0001      NDBPC    EQU      1           ;# OF NON-DATA BITS PER TRANSMITTED CHARACTER.
0008      DBPC     EQU      BPC-NDBPC   ;NUMBER OF DATA BITS PER CHARACTER.
0005      ZPL      EQU      5           ;ZERO PULSE LENGTH IN UNITS OF 1/2 CYCLES.
000F      OPL      EQU      15          ;ONE PULSE LENGTH IN UNITS OF 1/2 CYCLES.
0018      HCPBF    EQU      24          ;# OF 1/2 CYCLES PER BIT FRAME - 1.
0000      ZEROE    EQU      0           ;ZERO EQUATE.
000B      WAITC    EQU      11D         ;NUMBER OF ITERATIONS OF THE WAIT LOOP - 1.
0010      UCRIIP   EQU      10H         ;UCRI BOARD I/O PORT.
00FE      CBIT0    EQU      0FEH        ;CLEAR LAST BIT.
0007      HBHRP    EQU      07          ;NUMBER OF POPS NEEDED TO GET RETURN ADDR FROM INSIDE
; HBHR.
0002      HBTRP    EQU      02          ;# OF POPS NEEDED TO GET RETURN ADDR FROM INSIDE HBTR.
00FF      NONSA    EQU      OFFH        ;# OF NON SYNC'S THAT WILL BE READ BEFORE AN
; ERROR CONDTION IS RAISED DURING BLOCK HEADER READ IN
;ERROR FLAGS
0001      ERR1     EQU      1           ;NO INITIAL SYNC FOUND FROM HBHR.
0002      ERR2     EQU      2           ;MORE THAN 32 SYNC'S FOUND IN HEADER BLOCK. FROM HBHR
0003      ERR3     EQU      3           ;NO STX FOUND. FROM HBHR.
0004      ERR4     EQU      4           ;NO ETX FOUND. FROM HBTR.
0000      END

```

