```
0001    * 0078-001-H102 (G$DR)        DRAWING NO. 41285543-001-03
0002    *
0003    *
0004    *
0005    * COMPUTER:  H316, H516
0006    *
0007    *
0008    * PROGRAM CATEGORY:  INPUT-OUTPUT
0009    *
0010    *
0011    * PROGRAM TITLE:  G$DR
0012    *                 GENERIC DUMMY ROUTINE
0013    *
0014    *
0015    *
0016    *
0017    *
0018    *
0019    *
0020    *
0021    *
0022    *
0023    *
0024    *                APPROVAL                    DATE
0025    *
0026    *
0027    *
0028    *                PROG-------------------
0029    *
0030    *
0031    *                SUPR-N.G. Depledge-------
0032    *
0033    *
0034    *                QUAL-H. A. Dimmond-------
0035    *
0036    *                                NO. OF PAGES
0037    *
```

PROG 9th Mar 71

SUPR 9-MAR 71

QUAL 9 MAR 71

NO. OF PAGES 11


```
0038    *
0039    *
0040    * AUTHOR
0041    *
0042    *     HONEYWELL INFORMATION SYSTEMS LTD.
0043    *
0044    *
0045    * PURPOSE
0046    *
0047    *  1     TO DETERMINE THE ADDRESS AND SIZE OF THE FREE CORE AREA
0048    *        AVAILABLE TO THE CALLING PROGRAM FOR WORKSPACE, AND ENABLE
0049    *        EXTENDED ADDRESSING IF APPROPRIATE.
0050    *
0051    *  2     TO SATISFY ANY OUTSTANDING CALLS ON INPUT-OUTPUT LIBRARY
0052    *        ROUTINES, AND IGNORE OR ABORT ANY ATTEMPT TO USE THEM AT RUN
0053    *        TIME.
0054    *
0055    *
0056    * RESTRICTIONS
0057    *
0058    *     THIS PROGRAM WILL WORK ON ANY H316 OR H516 COMPUTER, WITH UP TO
0059    *     32 768 WORDS OF CORE STORE.  IT IS WRITTEN IN UNIVERSAL MODE,
0060    *     I.E. IT WILL WORK EQUALLY WELL WHETHER EXTENDED ADDRESSING IS
0061    *     ENABLED OR NOT (SEE 'USE' BELOW).
0062    *
0063    *     IF G$DR IS REQUIRED FOR PURPOSE 1 ABOVE, IT MUST BE LOADED AFTER
0064    *     ALL OTHER SUBROUTINES.  IN THIS CASE, CARE MUST BE TAKEN THAT
0065    *     NEITHER WORD '42 NOR WORD '43 OF G$DR LIES IN THE FIRST LOCATION
0066    *     OF A 4K MEMORY MODULE (I.E. IN LOCATION '10000, '20000, ETC.),
0067    *     AS THIS WILL CAUSE A HALT ON THE FIRST G$DR CALL.  ALSO, FOR THIS
0068    *     PURPOSE G$DR MUST NOT LIE IN SECTOR 0 OR SECTOR 1.
0069    *
0070    *     IT SHOULD ALSO BE NOTED THAT G$DR ALTERS THE CURRENT BASE SECTOR.
0071    *
0072    *
0073    * STORAGE
0074    *
```

```
*    56     (OCTAL)
*    46     (DECIMAL)
*
*    ALL BUT THE FIRST '26 (22) LOCATIONS ARE FREED WHEN G$DR IS FIRST
*    CALLED. IF THE PROGRAM IS NOT REQUIRED FOR PURPOSE 1 ABOVE, THE
*    LAST '30 (24) LOCATIONS MAY BE OVERWRITTEN AT LOAD-TIME.
*
*
* TIMING
*
*    G$DR:  11 CORE STORE CYCLES (EXCEPT THE FIRST TIME IT IS CALLED)
*    TABS ENTRIES:  16 CYCLES
*    SKP ENTRIES:   8 CYCLES
*    NOP ENTRIES:   5 CYCLES
*    ERNA ENTRIES:  NOT APPLICABLE
*    INCLUDING THE CALL, ASSUMING NO DESECTORING TO BE NECESSARY.
*
*
* USE
*
*    LOADING PROCEDURE
*
*          EXACT DETAILS CANNOT BE GIVEN HERE, BUT THE FOLLOWING
*          OBJECT MODULES SHOULD BE LOADED, IN ORDER, USING LDR-APM OR A
*          SIMILAR LOADER, FOR A TYPICAL CASE:
*
*          1     MAIN PROGRAM (USUALLY STARTING AT '1000)
*
*          2     APPROPRIATE INPUT-OUTPUT SUPERVISOR
*
*          3     AS MANY INDIVIDUAL INPUT-OUTPUT LIBRARY ROUTINES AS
*                REQUIRED, DEPENDING ON THE HARDWARE OPTIONS AVAILABLE
*
*          4     G$DR
*
*          G$DR MUST BE LOADED LAST, AS IT ASSUMES THAT THE REST OF THE
*          CORE STORE (STARTING AT LOCATION G$FF (FIRST FREE LOCATION))
```



```
*          CAN BE USED BY THE CALLING PROGRAM AS WORKSPACE AT RUN TIME.
*
*          CARE SHOULD BE TAKEN TO SELECT THE CORRECT DESECTORING MODE.
*          IF THE PROGRAM IS TO WORK ON COMPUTERS HAVING NOT MORE THAN
*          16 384 WORDS OF CORE STORE, OR IF ANY ONE OR MORE OF THE
*          CONSTITUENT MODULES IS RESTRICTED TO 16K WORKING BY, FOR
*          EXAMPLE, SIMULTANEOUS USE OF INDEXING AND INDIRECT, THEN ALL
*          LOADING SHOULD BE CARRIED OUT IN LXD MODE, E.G. BY ENTERING
*          LDR-APM AT 'XX000.   CONVERSELY, IF THE PROGRAM WILL ONLY BE
*          USED ON COMPUTERS WITH MORE THAN 16 384 WORDS OF CORE STORE,
*          AND ALL MODULES ARE WRITTEN IN UNIVERSAL MODE, THE LOADING
*          MAY BE CARRIED OUT IN EXD MODE, E.G. BY ENTERING LDR-APM AT
*          'XX006.
*
*          IF DESECTORING IN THE CURRENT BASE SECTOR IS TO BE AVOIDED,
*          DO NOT LOAD G$DR ACROSS A SECTOR BOUNDARY.
*
*
*    GENERIC DUMMY ROUTINE
*
*          THIS ENTRY MAY BE USED BY AN INPUT-OUTPUT SUPERVISOR PROGRAM
*          TO DETERMINE THE ADDRESS AND SIZE OF THE FREE CORE AREA
*          AVAILABLE FOR USE AS WORKSPACE, SUCH AS A COMPILER SYMBOL
*          TABLE.   THE CALLING SEQUENCE IS:
*
*                CALL  G$DR              GENERIC DUMMY ROUTINE
*                                        NORMAL RETURN, WITH THE A-
*                                        REGISTER CONTAINING THE
*                                        ADDRESS OF THE FIRST FREE
*                                        LOCATION, AND THE X-REGISTER
*                                        CONTAINING MINUS THE NUMBER OF
*                                        FREE LOCATIONS UP TO THE TOP
*                                        OF MEMORY ACTUALLY AVAILABLE
*                                        IN THE OBJECT COMPUTER.
*                                        EXTENDED ADDRESSING WILL HAVE
*                                        BEEN ENABLED OR DISABLED.
*
```

```
*
*    DEFAULT INPUT-OUTPUT ROUTINES
*
*          THE FOUR ROUTINES TABS, SKP, NOP AND ERNA SATISFY ANY
*          OUTSTANDING CALLS FOR ANY OF THE 85 INPUT/OUTPUT LIBRARY
*          ROUTINES LISTED IN THE 'SUBR' SECTION AT THE BEGINNING OF
*          THE CODING BELOW.   THE ROUTINES TABS, SKP AND NOP CAUSE THE
*          RELEVANT CALLS TO BE IGNORED AT RUN TIME, TOGETHER WITH ANY
*          ASSOCIATED PARAMETERS.   HOWEVER, ALL CALLS ASSOCIATED WITH
*          THE ROUTINE ERNA INDICATE AN ERROR CONDITION (SEE NEXT
*          PARAGRAPH).
*
*
* ERRORS
*
*    ANY CALLS ON INPUT-OUTPUT ROUTINES ASSOCIATED WITH THE DEFAULT
*    ROUTINE ERNA CAUSE THE EXTERNAL ROUTINE G$NA (DEVICE-NOT-AVAILABLE
*    ERROR ROUTINE) TO BE CALLED.   NO RETURN IS POSSIBLE AFTER THIS
*    CALL.   G$NA MUST BE PROVIDED AS AN ENTRY (SUBR) TO THE INPUT-
*    OUTPUT SUPERVISOR, WHICH MAY, FOR EXAMPLE, PRINT AN ERROR MESSAGE.
*
*
* METHOD
*
*    THE FIRST TIME G$DR (GENERIC DUMMY ROUTINE) IS CALLED, THE
*    APPROPRIATE ADDRESSING MODE - DXA OR EXA - IS DETERMINED,
*    ACCORDING TO THE MODE IN WHICH THE PROGRAM WAS DESECTORED ON
*    LOADING.  THE ACTUAL CORE SIZE AVAILABLE IN THIS ADDRESSING
*    MODE IS DETERMINED, THE RESULTS ARE STORED FOR FUTURE
*    REFERENCE, AND THE LOCATIONS CONTAINING THE ALGORITHM ARE
*    RELEASED TO FORM PART OF THE FREE CORE AREA.
*
*    THE METHOD USED IS BASED ON THE FACT THAT WHEN ONE ATTEMPTS TO
*    WRITE A WORD INTO A LOCATION ONE ABOVE THE MAXIMUM CORE SIZE,
*    THE WORD IS EITHER LOST OR WRITTEN INTO LOCATION ZERO,
*    DEPENDING UPON THE ACTUAL CORE SIZE.  WHEN ONE ATTEMPTS TO READ
*    THIS WORD BACK, IN THE FORMER CASE A ZERO IS READ, AND IN THE
```



```
*    LATTER CASE THE CONTENT OF LOCATION ZERO (THE X-REGISTER) IS
*    READ - THIS OF COURSE CAN BE CLEARED BEFOREHAND.  IF THE PROGRAM
*    IS RUNNING IN DXA MODE IN A MACHINE WITH A CORE SIZE GREATER THAN
*    16K, THE CORE SIZE WILL STILL APPEAR TO BE 16K - WHICH IS CORRECT
*    FOR THE MODE OF USE.
*
*    G$DR CONTAINS ONE INSTRUCTION WHICH MUST BE DESECTORED, AND TO
*    ENABLE THE USER TO AVOID MAKING USE OF THE NORMAL BASE SECTOR
*    FOR CERTAIN SPECIALISED APPLICATIONS, THE BASE SECTOR IS RESET
*    BEFORE THIS INSTRUCTION.
*
*
*************************************************************************
*
*
     SUBR  G$DR                GENERIC DUMMY ROUTINE
*
     SUBR  I$AI,TABS           INITIALISE SOURCE ASR
     SUBR  I$AIE,TABS          INITIALISE SOURCE ASR
     SUBR  I$AA,ERNA           ASR INPUT SOURCE RECORD
     SUBR  I$AAE,ERNA          ASR INPUT SOURCE RECORD
     SUBR  I$AB,ERNA           ASR INPUT OBJECT LINK
     SUBR  I$AO,ERNA           ASR INPUT OBJECT
*
     SUBR  O$AI,TABS           INITIALISE SOURCE ASR
     SUBR  O$AIE,TABS          INITIALISE SOURCE ASR
     SUBR  O$AA,ERNA           ASR OUTPUT SOURCE RECORD
     SUBR  O$AAE,ERNA          ASR OUTPUT SOURCE RECORD
     SUBR  O$AS,NOP            ASR OUTPUT END OF TEXT
     SUBR  O$ASE,NOP           ASR OUTPUT SOURCE END OF TEXT
     SUBR  O$ASS,NOP           ASR OUTPUT SOURCE END OF TEXT
     SUBR  O$AC,NOP            ASR OUTPUT CARRIAGE RETURN
     SUBR  O$AF,NOP            ASR OUTPUT LINE FEED
     SUBR  O$ALDR,NOP          ASR PUNCH LEADER
     SUBR  O$AB,ERNA           ASR OUTPUT OBJECT LINK
     SUBR  O$AO,ERNA           ASR OUTPUT OBJECT
     SUBR  O$ASO,NOP           ASR OUTPUT OBJECT END OF TEXT
```

```
          SUBR  O$AH,SKP        ASR HEADING ROUTINE
          SUBR  O$AL,ERNA       ASR LISTING ROUTINE
          SUBR  O$HH,SKP        ASR LIST HEADING
          SUBR  O$LL,ERNA       ASR LIST
          SUBR  O$AE,NOP        ASR LIST EJECT PAGE
          SUBR  O$AP,ERNA       ASR PRINT ROUTINE
*
          SUBR  I$PI,TABS       INITIALISE SOURCE HSR
          SUBR  I$PIE,TABS      INITIALISE SOURCE HSR
          SUBR  I$PA,ERNA       HSR INPUT SOURCE RECORD
          SUBR  I$PAE,ERNA      HSR INPUT SOURCE RECORD
          SUBR  I$PB,ERNA       HSR INPUT OBJECT LINK
          SUBR  I$PO,ERNA       HSR INPUT OBJECT
*
          SUBR  O$PI,TABS       INITIALISE SOURCE HSP
          SUBR  O$PIE,TABS      INITIALISE SOURCE HSP
          SUBR  O$PA,ERNA       HSP OUTPUT SOURCE RECORD
          SUBR  O$PAE,ERNA      HSP OUTPUT SOURCE RECORD
          SUBR  O$PS,NOP        HSP OUTPUT END OF TEXT
          SUBR  O$PSE,NOP       HSP OUTPUT SOURCE END OF TEXT
          SUBR  O$PSS,NOP       HSP OUTPUT SOURCE END OF TEXT
          SUBR  O$PC,NOP        HSP OUTPUT CARRIAGE RETURN
          SUBR  O$PF,NOP        HSP OUTPUT LINE FEED
          SUBR  O$PLDR,NOP      HSP PUNCH LEADER
          SUBR  O$POFF,NOP      HSP OFF
          SUBR  O$PB,ERNA       HSP OUTPUT OBJECT LINK
          SUBR  O$PO,ERNA       HSP OUTPUT OBJECT
          SUBR  O$PSO,NOP       HSP OUTPUT OBJECT END OF TEXT
          SUBR  O$PH,SKP        HSP LIST HEADING
          SUBR  O$PL,ERNA       HSP LIST
          SUBR  O$PE,NOP        HSP LIST EJECT PAGE
          SUBR  O$PP,ERNA       HSP PRINT ROUTINE
*
          SUBR  I$CA,ERNA       CARD READER INPUT SOURCE RECORD
          SUBR  I$CABL,NOP      CARD READER SOURCE INPUT BUFFER LENGTH
          SUBR  I$CO,NOP        CARD READER OFFSET STACK CARD
          SUBR  I$CB,ERNA       CARD READER INPUT OBJECT LINK
```



```
          SUBR  I$CBO,ERNA      CARD READER INPUT OBJECT
*
          SUBR  O$CA,ERNA       CARD READER OUTPUT SOURCE RECORD
          SUBR  O$CABL,NOP      CARD PUNCH SOURCE OUTPUT BUFFER LENGTH
          SUBR  O$CO,NOP        CARD PUNCH OFFSET STACK CARD
          SUBR  O$CS,NOP        CARD PUNCH OUTPUT END OF TEXT
          SUBR  O$CSS,NOP       CARD PUNCH OUTPUT SOURCE END OF TEXT
          SUBR  O$CB,ERNA       CARD PUNCH OUTPUT OBJECT LINK
          SUBR  O$CBO,ERNA      CARD PUNCH OUTPUT OBJECT
*
          SUBR  O$LH,SKP        LINE PRINTER LIST HEADING
          SUBR  O$LA,ERNA       LINE PRINTER LIST LINE
          SUBR  O$LE,NOP        LINE PRINTER LIST EJECT PAGE
          SUBR  O$LF,NOP        LINE PRINTER ADVANCE ONE LINE
          SUBR  O$LO,ERNA       LINE PRINTER PRINT ONE LINE
          SUBR  O$LP,NOP        LINE PRINTER ADVANCE TO TOP OF FORM
*
          SUBR  I$MA,ERNA       READ MAGNETIC TAPE BCD
          SUBR  I$MB,ERNA       READ MAGNETIC TAPE BINARY, 2 CH/WORD
          SUBR  I$MC,ERNA       READ MAGNETIC TAPE BINARY, 3 CH/WORD
*
          SUBR  O$MA,ERNA       WRITE MAGNETIC TAPE BCD
          SUBR  O$MB,ERNA       WRITE MAGNETIC TAPE BINARY, 2 CH/WORD
          SUBR  O$MC,ERNA       WRITE MAGNETIC TAPE BINARY, 3 CH/WORD
          SUBR  O$ME,SKP        MAGNETIC TAPE WRITE FILE MARK
*
          SUBR  M$UNIT,NOP      MAGNETIC TAPE UNIT NUMBER ROUTINE
          SUBR  C$MR,SKP        MAGNETIC TAPE REWIND
          SUBR  C$FR,ERNA       MAGNETIC TAPE FORWARD ONE RECORD
          SUBR  C$BR,ERNA       MAGNETIC TAPE BACKSPACE ONE RECORD
          SUBR  C$FF,ERNA       MAGNETIC TAPE FORWARD ONE FILE MARK
          SUBR  C$BF,ERNA       MAGNETIC TAPE BACKSPACE ONE FILE MARK
*
          SUBR  C$6TO8,ERNA     CONVERT FROM SIX-BIT CODE TO ISO CODE
          SUBR  C$8TO6,ERNA     CONVERT FROM ISO CODE TO SIX-BIT CODE
          SUBR  C$IS,ERNA       CONVERT FROM INVISIBLE TO SIX-BIT CODE
          SUBR  C$SI,ERNA       CONVERT FROM SIX-BIT CODE TO INVISIBLE CODE
```

```
                          SUBR  C$IN,TABS        INITIALISE SOURCE
                     *
                          SUBR  G$FF             FIRST FREE LOCATION
                     *
                     *
                          CF5                    H316 OR DDP-516
                          REL                    RELOCATABLE
                     *
                     *
                     * GENERIC DUMMY ROUTINE
                     *
00000   0 000000     G$DR DAC   **                GENERIC DUMMY ROUTINE
                     *
                     * THE NEXT INSTRUCTION IS CHANGED, ON FIRST ENTRY, TO EITHER DXA OR EXA
00001   0 01 00026   MODE JMP   DCS               DETERMINE CORE SIZE
00002   0 02 00024        LDA   AFFL              ADDRESS OF FIRST FREE LOCATION
00003   0 35 00025        LDX   MNFL              MINUS NUMBER OF FREE LOCATIONS
00004  -0 01 00000        JMP*  G$DR              EXIT
                     *
                     *
                     * DEFAULT INPUT-OUTPUT ROUTINES
                     *
                     * IGNORE BUFFER SIZE, NUMBER OF TABS, AND TAB TABLE
00005   0 000000     TABS DAC   **                IGNORE BUFFER SIZE AND TABS
00006   0 12 00005        IRS   TABS              TALLY RETURN ADDRESS (IGNORE BUFFER SIZE)
00007  -0 02 00005        LDA*  TABS              NUMBER OF TABS (N)
00010   0 06 00005        ADD   TABS              ADD (N+1) TO RETURN ADDRESS
00011   141206            AOA
00012   0 04 00005        STA   TABS              RETURN ADDRESS
00013  -0 01 00005        JMP*  TABS              EXIT
                     *
                     * IGNORE HEADING ADDRESS (OR LOGICAL UNIT NUMBER)
00014   0 000000     SKP  DAC   **                IGNORE HEADING ADDRESS
00015   0 12 00014        IRS   SKP               TALLY RETURN ADDRESS (IGNORE HEADING ADDR)
00016  -0 01 00014        JMP*  SKP               EXIT
                     *
                     * IGNORE CALL WITHOUT PARAMETERS
```



```
00017   0 000000     NOP  DAC   **                IGNORE CALL WITHOUT PARAMETERS
00020  -0 01 00017        JMP*  NOP               EXIT
                     *
                     * ERROR:  DEVICE NOT AVAILABLE
00021   0 000000     ERNA DAC   **                ERROR:  DEVICE NOT AVAILABLE
00022  -0 10 00023        JST*  G$NA              DEVICE NOT AVAILABLE ERROR ROUTINE
                     *
                     *
                     * CONSTANTS AND VARIABLES 1
                     *
00023   0 000000     G$NA XAC   G$NA              DEVICE NOT AVAILABLE ERROR ROUTINE
00024   0 000026     AFFL DAC   G$FF              ADDRESS OF FIRST FREE LOCATION
00025                MNFL BSS   1                 MINUS NUMBER OF FREE LOCATIONS
                     *
                     *
                     * FIRST FREE LOCATION (FROM HERE UPWARDS MAY BE OVERWRITTEN)
                     *
00026                G$FF BSS   0                 FIRST FREE LOCATION
                     *
                     *
                     * DETERMINE CORE SIZE (THIS CODE IS EXECUTED ONLY ONCE)
                     *
00026   0 02 00055   DCS  LDA   IPZE              INDEXED PZE INSTRUCTION
00027   0414 76           LGL   2                  TAG BIT TO C
00030   0 02 00052        LDA   DXA                DXA INSTRUCTION
00031   100001            SRC                      CONVERT TO EXA INSTRUCTION....
00032   0 06 00035        ADD   TWO                ..IF PROGRAM DESECTORED IN EXD MODE
00033   0 04 00001        STA   MODE               SET ADDRESSING MODE INSTRUCTION
00034   0 04 00035        STA   *+1
00035   000002       TWO  DEC   2                  OVERWRITTEN BY MODE INSTRUCTION
                     *
                     * CALCULATE CORE SIZE
00036   140040            CRA
00037   0 06 00053   CLCS ADD   4K                 MEMORY MODULE SIZE
00040   0 04 00025        STA   CS                 UPDATE CORE SIZE
00041  -0 13 00025        IMA*  CS                 WRITE NON-ZERO WORD
00042   0 35 00054        LDX   ZERO               CLEAR INDEX IN CASE OF WRAP-AROUND WRITE
```

```
0371 00043  -0 13 00025      IMA*  CS          ATTEMPT TO RETRIEVE NON-ZERO WORD
0372 00044   100040          SZE               WORD LOST - CORE SIZE FOUND
0373 00045   0 01 00037      JMP   CLCS        WORD PRESERVED - TRY LARGER CORE SIZE
0374                     *
0375                     * CORE SIZE FOUND
0376 00046   0 02 00024      LDA   AFFL        ADDRESS OF FIRST FREE LOCATION
0377 00047   0 07 00025      SUB   CS          CORE SIZE
0378 00050   0 04 00025      STA   MNFL        MINUS NUMBER OF FREE LOCATIONS
0379 00051   0 01 00001      JMP   MODE        GO LOAD REGISTERS AND EXIT
0380                     *
0381                     *
0382                     * CONSTANTS AND VARIABLES 2 (MAY BE OVERWRITTEN)
0383                     *
0384 00052   000011      DXA  DXA              DISABLE EXTENDED ADDRESSING
0385 00053   010000      4K   OCT   10000      MEMORY MODULE SIZE
0386 00054   000000      ZERO OCT   0          ZERO
0387                     *
0388         000025      CS   EQU   MNFL       CORE SIZE
0389         000025      BASE EQU   MNFL       SPACE TO DESECTOR IPZE INSTRUCTION BELOW
0390                     *
0391                     * SET BASE SECTOR
0392                         SETB  BASE
0393 00055   1 00 01000 IPZE PZE   '1000,1     INDEXED PZE INSTRUCTION
0394                     *
0395                     *
0396                         END               END OF G$DR 04.MAR.71 G.STUARTSON LITTLE

4K      000053    AFFL    000024    BASE    000025    CLCS    000037
CS      000025    DCS     000026    DXA     000052    ERNA    000021
G$DR    000000    G$FF    000026    G$NA    000023    IPZE    000055
MNFL    000025    MODE    000001    NOP     000017    SKP     000014
TABS    000005    TWO     000035    ZERO    000054

0000 WARNING OR ERROR FLAGS
$DA 41285326-002-03 8.1.71 JDC
```