IDENTIFICATION

DIVIDE BY ZERO LOCATER - MU DIVZR
704 SAP Subroutine
Terry W. Edwards - October 3, 1964
Midwestern Universities Research Association, Stoughton, Wisconsin

PURPOSE

To locate division by zero within the users program. (Division by zero is usually a program error and is normally difficult to pinpoint in extensive programs.)

METHOD

MU DIVZR replaces every divide and proceed instruction (DVP and/or FDP as compiled by FORTRAN) with its divide and halt counterpart (DVH and/or FDH) found within that section of the memory designated by the user. During subsequent execution of the calling program, if division by zero is attempted, a program stop occurs so that the memory location of the offending divide instruction may be noted.

USAGE

MU DIVZR may be utilized in two different ways, either completely automatically or with manual control.

In the automatic mode, an appropriate calling statement from within a FORTRAN program or subprogram is:

CALL DIVZR (+6HENDPRG).

The appearance of an argument whose value is nonzero results in replacement of DVP and FDP instructions by DVH and FDH instructions in that portion of the memory between the location of the calling TSX instruction and the location of the argument. (It should be noted that an alphanumeric argument will be assigned an address at the end of the program.) This is intended to enable handling of individual programs or subprograms. It is recommended that the location of the argument not be in COMMON storage unless extreme care is exercised, since replacement of data words whose leading half is identical in form to the FDP and DVP instructions can also take place. Furthermore, replacement of these instructions in the input-output routines (IOH)I, (SPH), etc. has been found fatal to some subsequent input-output operations.

In the manual mode, the value of the argument must be zero. This mode permits manual entry of that portion of memory to be searched and modified. At the program stop HPR 77700777, enter (first word address in the decrement, last word address in the address) into the MQ the desired memory section to be processed, and push START.

Note: In both automatic and manual modes of operation, MU DIVZR replaces the calling TSX instruction in the calling program by a bypass, so that only one entry is made into MU DIVZR from that point.

CODING INFORMATION

MU DIVZR requires 57 storage locations and is available in binary card form.

```
DIVZR REM DIVIDE BY ZERO LOCATER SUBROUTINE - EDWARDS
      REM
      REM          LINKAGE FROM FORTRAN PROGRAMS
      REM
      REM               CALL DIVZR (+6HENDPRG)
      REM
      REM THIS SUBROUTINE WILL REPLACE EVERY DVP BY A DVH INSTRUCTION
      REM AND EVERY FDP BY A FDH IN THAT PORTION OF THE CALLING PROGRAM
      REM (OR SUBROUTINE) FROM THE LOCATION OF THE CALL STATEMENT TO
      REM THE STORAGE LOCATION OF THE ARGUMENT.  ALTERNATIVELY, IF THE
      REM VALUE OF THE ARGUMENT IS ZERO, A PAUSE OCCURS DURING WHICH A
      REM PARTICULAR REGION OF THE MEMORY MAY BE ENTERED INTO THE MQ
      REM (FWA IN DECREMENT, LWA IN ADDRESS) FOR PROCESSING.
      REM
      REM ONLY ONE ENTRY IS MADE INTO THIS ROUTINE, SINCE THE CALLING
      REM TSX INSTRUCTION IS REPLACED BY A TRA*+2 INSTRUCTION.
      REM
      REM PROGRAM CARD
      REM
      FUL
PCARD MZE 0,0,4              9L
      PZE 0                  9R
      PZE (END)              8L
      PZE 0                  8R
      BCD 1DIVZR             7L
      PZE DIVZR              7R
      REM
      REL
      ORG 0
      REM
DIVZR SXD X4,4               SAVE INDEX REGISTER 4.
      REM
OVRWR PXD 0,4                PREPARE TO OVERWRITE TSX.
      ARS 18
      COM
      ADD ONE
      STA PLANT              STORE ADDRESS OF TSX INSTRUCTION.
      ADD TWO
      STA TRAP               STORE ADDRESS OF TRA*+2 INSTRUCTION.
      CLA TRAP
PLANT STO **                 OVERWRITE TSX BY TRA*+2.
      REM
      CLA 1,4                PICK UP AND TEST ARGUMENT.
      STA *+1
      CLA **                 BRANCH. IF ZERO - MANUAL ENTRY.
      TNZ AUTOM              IF NON-ZERO - AUTOMATIC ENTRY.
      REM
MANUL PXD 0,0                ENTRY FOR MANUAL SPECIFICATION.
      LRS 35
      HPR 511,7,63           PAUSE  777C0777
      LGL 18
      COM                    COMPUTE TWO S COMPLEMENT
      ADD ONE                OF FIRST WORD ADDRESS
      PAX 0,4                AND LOAD IT INTO INDEX 4.
      PXD 0,0
      LGL 18
      COM
      ADD ONE                COMPUTE TWO S COMPLEMENT
      ALS 18                 OF LAST WORD ADDRESS
      STD TEST               AND STORE IT IN TEST.
```

```
X4      TXI LOOK,0,**
        REM
AUTOM   CLA 1,4             ENTRY FOR AUTOMATIC SPECIFICATION.
        COM
        ADD ONE             COMPUTE TWO S COMPLEMENT
        ALS 18              OF LAST WORD ADDRESS
        STD TEST            AND STORE IT IN TEST.
        REM
LOOK    CAL 0,4             PICK UP AN INSTRUCTION.
        TZE MODX4
        TMI MODX4           REJECT IF ZERO OR NEGATIVE.
        ARS 18              SHIFT TO PREVENT PROCESSING OF
        SUB DVP             THIS PROGRAM CONSTANTS.
        TZE REPLC
        ADD DVP             REJECT IF NOT DVP OR FDP.
        SUB FDP
        TNZ MODX4
        REM
REPLC   CLA 0,4             IF DVP OR FDP INSTRUCTION
        ARS 24              CONVERT TO DVH OR FDH.
        SUB ONE
        ALS 24
        STD 0,4             STORE DECREMENT AND PREFIX
        STP 0,4             IN DIVIDE INSTRUCTION.
        REM
MODX4   TXI TEST,4,-1       MODIFY INDEX REGISTER AND
TEST    TXH LOOK,4,**       REPEAT IF NOT FINISHED.
        REM
RETN    LXD X4,4            RELOAD INDEX REGISTER 4 AND
        TRA 2,4             RETURN TO CALLING PROGRAM.
        REM
ONE     OCT 000001          USEFUL CONSTANTS, ETC.
TWO     OCT 000002
TRAP    TRA 0
DVP     OCT 022100
FDP     OCT 024100
        REM
(END)   SYN *
        END 0
```