RELATIVE ADDRESS CALCULATOR PROGRAM

By: Russel Yost

This Relative Address Calculator Program may be used to calculate relative addresses for branch instructions. This is especially useful when calculating long branches where you are more likely to make an error if you do it by hand. This program lets the computer do the work for you.

To use the program, type in the machine code listing on the next page. The entire program fits inside the scratchpad RAM used by Mikbug®. Be sure to save the program on tape if you have a tape unit connected to your computer. After loading the program type a G for "Go to User Program". The computer will home the cursor and erase the screen on those systems using the CT-1024 Terminal System with the CT-CA option. It will then print out a BA which stands for "branch address". To this you should respond with the address of the branch instruction and not the address following it. The program will then output a T which stands for "TO". Now you type the destination address of the branch instruction. The program outputs a = followed by the relative address. If branching forward, the outputted address will be OOXX and you must be sure not to have XX greater than 7F. If branching backwards, the outputted address will be FFYY and you must be sure to have YY greater than 7F. Only the last two digits of the outputted address are used for the relative address.

If any non-hex character is input at either address, the program jumps to Mikbug® and outputs a *. Upon entering Mikbug®, typing a G will restart the Relative Address Calculator Program. After calculating each relative address, the program prepares itself for new data. When using the CT-1024 Terminal System, the program will home and erase the terminal's screen after each calculation.

Mikbug® is a registered trademark of Motorola, Inc.

```
                        NAM    RELADR
                * MIKBUG LOCATIONS
 E07E           PDATA1  EQU    $E07E
 E047           BADDR   EQU    $E047
 E0C8           OUT4HS  EQU    $E0C8

 A000                   ORG    $A000
 A000           RAMST   RMB    2
 A002           BRADR   RMB    2
 A004           DEST    RMB    2

 A014                   ORG    $A014
 A014 8E A0 47  BEGIN   LDS    #$A047    Saves BEGIN in  A048,49
 A017 CE A0 6F          LDX    #MSETUP   Clears screen
 A01A 8D 4F             BSR    PDATSR    See subroutine below.
 A01C BD E0 47  NEXT    JSR    BADDR     Gets 4 hex & store in X
 A01F FF A0 02          STX    BRADR     Stores branch addr
 A022 CE A0 77          LDX    #MT       Outputs " T "
 A025 8D 44             BSR    PDATSR
 A027 BD E0 47          JSR    BADDR
 A02A FF A0 04          STX    DEST      Stores dest'n addr.
 A02D CE A0 00          LDX    #RAMST    Prepare for indexed
 A030 0C                CLC              Addr. mode.
 A031 A6 04             LDA A  4,X       DEST H
 A033 E6 05             LDA B  5,X       DEST L
 A035 20 13             BRA    CONTN

 A048                   ORG    $A048
 A048 A0 14             FDB    BEGIN

 A04A C0 02     CONTN   SUB B  #02       Subtract 0002 from
 A04C 82 00             SBC A  #00       Destination Addr.
 A04E 0C                CLC
 A04F E0 03             SUB B  3,X       BRA L Subtract Br addr
 A051 A2 02             SBC A  2,X       BRA H (Destn. - 2)
 A053 A7 69             STA A  $69,X     REL H Store at REL
 A055 E7 6A             STA B  $6A,X     REL L
 A057 CE A0 7B          LDX    #MEQ      Outputs " = "
 A05A 8D 0F             BSR    PDATSR
 A05C CE A0 69          LDX    #REL
 A05F BD E0 C8          JSR    OUT4HS    Outputs 4 hex's + Sp.
 A062 CE A0 71          LDX    #MBA      Outputs CR, LF, "BA "
 A065 8D 04             BSR    PDATSR
 A067 20 B3             BRA    NEXT
 A069           REL     RMB    2
 A06B BD E0 7E  PDATSR  JSR    PDATA1    Outputs string
 A06E 39                RTS

 A06F 10        MSETUP  FCB    $10,$16
 A070 16
 A071 0D        MBA     FCB    $0D,$0A,$42,$41,$20,$04
 A072 0A 42
 A074 41 20
 A076 04
 A077 20        MT      FCB    $20,$54,$20,$04
 A078 54 20
 A07A 04
 A07B 20        MEQ     FCB    $20,$3D,$20,$04
 A07C 3D 20
 A07E 04
                END

NO ERROR(S) DETECTED

SYMBOL TABLE:
BADDR  E047   BEGIN  A014   BRADR  A002   CONTN  A04A   DEST   A004
MBA    A071   MEQ    A07B   MSETUP A06F   MT     A077   NEXT   A01C
OUT4HS E0C8   PDATA1 E07E   PDATSR A06B   RAMST  A000   REL    A069
```