

TN-85-45

A Simulation of a Versatile
Scientific Calculator

By

A.A. Patel
and
D.R. Kulkarni

June 1985

PHYSICAL RESEARCH LABORATORY
AHMEDABAD

A Simulation of a Versatile Scientific Calculator

A.A. Patel and D.R. Kulkarni

ABSTRACT

A software package DECALC which simulates a versatile, non-programmable scientific calculator has been developed on a DEC system 1091. The package enables any person to use VT 100 (or VT 100 compatible) terminal connected to DEC system directly as a scientific calculator. The characteristic feature of this calculator is that it allows you to enter the mathematical expressions as they appear in your problem in a very flexible way. The expressions are also evaluated with an accuracy which is as high as 15 to 18 significant digits. In addition it has 10 memory registers which greatly enhance its power and ease of operation compared to conventional pocket calculators. It has also four independent memory registers, two for cumulative additions and two for cumulative subtractions. DECALC offers number of mathematical, statistical and other useful functions built into it. For statistical functions it uses a built-in buffer of 99 words available in it. It has number of short and crisp commands to simulate most of the operations of the pocket calculators. DECALC has an attractive stationary display which indicates the final result along with the current expression (or command)

entered and its value (or the remarks). The display also shows if the arguments of the trigonometric functions are in radian mode or in degree mode. DECALC can repeatedly execute the same command/expression without entering it again and again. The count of repetition is explicitly shown in the display.

A Simulation of a Versatile Scientific Calculator

I. INTRODUCTION

In scientific research the importance of calculations need not be overemphasized. One may not make use of modern computer, but he/she may not, perhaps, avoid using a simple pocket scientific calculator which is so handy and easy to operate and needs no knowledge of programming. Even one who uses modern computers, invariably needs a pocket calculator for testing his program thoroughly. Though various versatile models of pocket calculators having different features are available in the market, it may not be possible to get a single pocket calculator having most of the features one may like to have. Further the accuracy of calculations in these calculators is generally limited to only 8 to 10 significant digits. It has also been observed that due to factors such as a single panel display, small number (generally 2 to 3) of memory registers etc. they do not offer enough ease in operation especially for large scientific calculations.

We have, therefore, developed a software package which simulates a versatile, non-programmable scientific calculator having number of desirable features chosen from various commercial pocket calculators. We have also incorporated some additional features which enhance considerably its power and ease of operation. Hereafter we refer to this

calculator simulated on a system DEC 1091 as DECALC. Using DECALC a VT 100 (or VT 100 compatible) terminal connected to DEC 1091 readily presents itself as a scientific calculator which can, then, be easily used by any person who may not know programming at all.

DECALC accepts as input a string of maximum 58 characters which may contain embedded blanks. The alphabetic letters in the string may be in upper case, lower case or mixed case. The string may be a command or an expression to be executed by DECALC. DECALC first checks if the string is a command and then executes it. If it is not a command, the string is interpreted as an expression. The successful execution of command or expression (as well as) invalid command or expression will be indicated to the user through meaningful messages on the display. We describe below some of the salient features of DECALC.

1. You can enter the mathematical expression as they appear in your problem except that the arithmetic operations are to be indicated explicitly by the corresponding operators.
2. The expression may or may not contain parentheses. The parentheses may also be indicated by curly or square brackets available on VT 100 terminal keyboard. In any case the expression may contain maximum 15 pairs of parentheses.
3. The expression may contain number of operators appearing consecutively. (For details see section II).

4. DECALC has large number of mathematical, statistical and other useful functions built into it. (See appendix II).
5. DECALC has high accuracy of 15 to 18 significant digits.
6. Unlike conventional pocket calculators, DECALC has 10 general-purpose memory registers for storing intermediate results. The registers are denoted by M0, M1, M2, M3, M4, M5, M6, M7, M8 and M9. They greatly increase the ease of operation of DECALC.
7. In addition DECALC has four special-purpose memory registers denoted by MA1, MA2, MS1 and MS2. The registers MA1 and MA2 are exclusively used for cumulative addition and registers MS1 and MS2 exclusively for cumulative subtraction.
8. For evaluation of statistical functions, DECALC has a built-in buffer of 99 words where the terms are stored before the statistical functions are invoked.
9. DECALC has six short and crisp commands which simulate most of the operations of the pocket calculators.
10. The unique feature is, however, its display system. Its stationary non-scrolling display appears on the terminal screen as soon as you start working with DECALC. The display has 3 panels in reverse video to attract the immediate attention of the user. The execution of every command/expression changes the display followed by a bell to signal the user to enter new command/expression if desired.

The top panel contains the final result in an especially enlarged digits. The final result may be simply the value of the expression just entered. However, sometimes, the final result can be the value obtained by using the previous result and the current expression connected by an arithmetic operators. (For details see section II).

The middle panel contains the expression or command entered by the user. Thus you can see your expression or command on the display even after the execution is over. This gives you one more opportunity to ensure if you have entered your expression/command correctly or not.

The lowest panel is unique in that it echoes back the message to indicate if your expression/command has gone through successfully or not. In the case of valid expression it shows the value of the expression entered. As stated earlier, the value of the final result in the top panel need not always be same as that of the current expression and hence one can appreciate the importance of this panel.

11. DECALC can evaluate the trigonometric functions having arguments either in radian mode or in degree mode. The mode can be selected by the user using commands RAD and DEG. (See section III). The default mode is DEGREES. The display also shows explicitly in what mode DECALC is operating. Further DECALC allows the same expression/command to be executed

repeatedly if required without entering it again and again. The count of repetition is also indicated explicitly on the display.

The DECALC package has been developed in FORTRAN 10, version 7 available on DEC 1091. It has a main program along with 37 subprograms consisting of approximately 1400 lines of coding. The structure of the package is given appendix I.

II. EXECUTION OF EXPRESSION

The execution of expression is the most crucial task in the simulation of a scientific calculator. The task consists of two sub-tasks. The first one is to validate the expression and second one is to evaluate the expression to yield the numeric value. Before we discuss how these sub-tasks are performed, we look into the various components of the valid expression. The valid expression consists of parentheses, arithmetic operators, operands and function names. The parentheses in the expression, as stated in section I, can be circular, curly or square (rectangular) brackets.

1. Arithmetic Operators:

DECALC has five arithmetic operators represented by plus (addition), dash (subtraction), star (Multiplication), slash (division), and circumflex (exponentiation). The operator circumflex (^) has the highest priority among all operators.

The second priority is given to star (*) and slash (/). The last priority is given to plus (+) and dash (-). The functions in the expression have higher priority than the operators. The parentheses, of course, will override the priorities defined above.

2. Operands :

DECALC takes following items as valid operands.

- a) All the numbers with or without decimal point.
- b) The words R0, R1, R2, R3, R4, R5, R6, R7, R8, and R9 which represent the values retrieved from memory registers M0 to M9 respectively.
- c) The words RA1, RA2, RS1 and RS2 which represent the values retrieved from memory registers MA1, MA2, MS1 and MS2 respectively.
- d) The valid function names. (See appendix II).
- e) The word PI which is defined as 3.1415926535897932

Any other character or character combination is considered as invalid operand or function-name. The upper case and lower case characters are treated as equivalent.

3. Validation and Evaluation Of The Expression :

At the outset DECALC brackets the complete expression entered by the user for the convenience of the program logic. Before evaluating the expression DECALC ensures that the expression is valid. An invalid expression can not go for evaluation stage. Initially DECALC looks for unacceptable

characters, the wrong positioning and absurd sequences of operators, parentheses and decimal points that make the expression outright invalid. It also ensures that the starting and ending parentheses are balanced and the number of pairs does not exceed 15. It then separates out the expression for each pair of bracket. Later DECALC finds the number of operators and operands contained in each pair of bracket. The operands are subsequently converted into numeric values and the expression for each bracket is evaluated independently starting from the innermost bracket. Thus the complete expression is evaluated. The operands that are separated are first validated before they are converted. The invalid operand makes the expression invalid and further execution stops automatically. The expression is also not evaluated if the function has invalid argument. If the buffer is empty, the statistical functions are not evaluated. In all these cases, the appropriate message will appear in the display to guide the user properly.

4. Features of DECALC Expression :

We describe below some of the unique features of the DECALC expression.

a) Two consecutive operators are allowed if the combination is meaningful such as $*-$, $/+$, $/-$, $*+$ etc.

b) If the operators $+$ and $-$ appear in a combination containing two or more operators consecutively, the combination is then reduced to a single operator by simple algebraic rule.

- c) When two or more operators appear in a combination which is not meaningful, the right-most operator in the combination is taken to evaluate the expression.
- d) If the operators *, /, or ^ appear as the first character in the expression, it will be taken as a binary operator connecting the value in the top panel and the expression on the right-hand side of the operator.
- e) If the operators + or - appear as the first character in the expression, it is considered either as a binary operator or an unary operator depending on the expression. If the expression on its right-hand side contains only one operand or if it is completely in the bracket, the operator is treated as a binary operator connecting the value in the top panel and the expression. Otherwise the operator is treated as an unary operator.

III. EXECUTION OF COMMANDS

We describe below six commands available on DECALC. For each command, there exists two subroutines, one for decoding the command and other for implementing the command. An invalid command is not implemented and the message to that effect is displayed. The successful implementation of the command is also echoed back by appropriate message in the display.

1. ST command :

This command has following three forms.

a) ST M0 to M9

This command stores the value contained in the top panel (ie. final result) in the memory register appeared as an argument. The original contents of the memory register are replaced by the new value.

b) ST MA1 or MA2.

This command adds the value contained in the top panel to the original contents of the memory register MA1 or MA2.

c) ST MS1 or MS2.

This command subtracts the value contained in the top panel from the original contents of the memory register MS1 or MS2.

2. CLR command :

This command has following arguments.

a) CLR M0, M1, M2, M3, M4, M5, M6, M7, M8, M9, MA1, MA2, MS1 or MS2.

This command clears the memory register by making its content zero.

b) CLR BUF : This command clears the buffer area. It is advisable to use this command before you use IN command.

c) CLR RES : This command clears the contents of the top panel by making it zero.

d) CLR EXP : This command clears the second panel by making the expression or command blank.

e) CLR ALL : This command clears everything described above. In addition it also makes the count of repetition in the display zero.

3. DIS command :

This command has following arguments.

a) DIS M0 to M9, MA1, MA2, MS1 or MS2.

This command displays the contents of memory register appearing in the command.

b) DIS NT : This command displays the number of terms available in the buffer area.

c) DIS t1, t2, t3, - - - - -, t97, t98, or t99.

This command displays the contents of the buffer word appeared in the command.

4. IN command :

This command has no argument and it stores the value contained in the top panel in the buffer word. It also keeps count of number of terms entered in the buffer. Using this command the terms are placed in the buffer before you invoke any statistical functions.

5. DEG command :

This command has no arguments and it allows the trigonometric functions to assume their arguments to be in degrees. This is a default mode.

6. RAD comand:

This command also does not have argument and it allows the trigonometric functions to assume their arguments to be in radians.

APPENDIX I

Structure of Package DECALC

I. Main Program :

The main program DECALC calls the following subroutines.

- 1) CLSCRN 2) PANEL 3) CURSOR 4) PACK 5) LOWTUP 6) VLDCOM
- 7) GETVAL 8) GETRES 9) DISPLY

The sub. PANEL calls subroutines CHRATR and CURSOR.

II. Subroutines for Evaluating Expression :

a) Subroutine GETVAL calls subroutines

- 1) VLDEXP 2) NEWEXP 3) DCDQ 4) EVLEXP

b) Sub. VLDEXP calls subroutine VALEX1.

c) Sub. NEWEXP calls subroutines

- 1) PUSHCH 2) POPCH 3) PACK

d) Sub. DCDQ calls subroutines

- 1) SEPOR 2) STAVAL

e) Sub. EVLEXP calls subroutines

- 1) FVAL 2) EVLQ

f) Sub. FVAL calls functions

- 1) FACT1 2) REC 3) PER

g) Sub. EVLQ calls subroutines

- 1) VALUE 2) PUSHDB 3) HIR 4) POPDB

The subroutine VALUE calls a FORTRAN system routine OVERFL.

III. Subroutines for Executing Commands :

a) Subroutines VLDCOM calls subroutines

- 1) DCDST 2) IMPST 3) DCDCLR 4) IMPCLR 5) DCDIN
- 6) IMPIN 7) DCDDIS 8) IMPDIS 9) DCDARG 10) IMPARG

IV. Subroutines for Exhibiting Display :

a) The following routines are required for display.

- 1) CLSCRN 2) PANEL 3) CURSOR 4) DISPLY

b) Sub. DISPLY calls subroutines

- 1) CHARATR 2) CURSOR.

APPENDIX II

The List of Function-names in DECALC

1. MATHEMATICAL FUNCTIONS :

There are 15 mathematical functions.

- | | | | |
|--------------|------------------------------|--------------|--------------------|
| 1) SQRT(X) | $X > 0$ | 2) ALOG(X) | $X > 0$ |
| 3) ALOG10(X) | $X > 0$ | 4) ASIN(X) | $-1 \leq X \leq 1$ |
| 5) ACOS(X) | $-1 \leq X \leq 1$ | 6) ATAN(X) | any X |
| 7) SIN(X) | any X | 8) CDS(X) | any X |
| 9) TAN(X) | any X | 10) COTAN(X) | any X |
| 11) ABS(X) | any X | 12) COSH(X) | |
| 13) SINH(X) | $1 \times 10^{88.722}$ | 14) TANH(X) | |
| 15) EXP(X) | $-89.415 \leq X \leq 88.029$ | | |

b) STATISTICAL FUNCTIONS :

There are 5 statistical functions.

- | | |
|------------------------------|------------------------------------|
| 1) SGMX (sum of all terms) | 2) SGMX2 (sum of squares of terms) |
| 3) MN (mean of the terms) | 4) VAR (Variance of the terms) |
| 5) SDV (Standard deviation). | |

These functions make use of terms already placed in the buffer.

c) OTHER FUNCTIONS :

There are other four functions.

- 1) SQR(X) square of X.
- 2) FACT(X) factorial of X.
- 3) REC(X) reciprocal of X.
- 4) PER(X) :

This function yields the value which is X% (X per cent) of the value contained in the top panel.

.....X.....

Acknowledgement

We thank Mr.G.G.Dholkia for providing us an assembly subroutine which controls the line-width and the tty page-size in our package.