

M I C R O C A L C U L A T O R

**“ ELEKTRONIKA
MK 61 ”**

FOR SCIENTIFIC,
ENGINEERING
AND STATISTICS
CALCULATIONS

OPERATING INSTRUCTIONS

Table of Contents

Foreword	3
1. Notice to Owners	4
2. Package Contents	5
3. Safety Instructions	6
4. Micro Calculator Specifications	7
5. Keyboard and Switches Description	8
Latin keyboard layout transcription	8
Original Cyrillic keyboard layout	9
5.1. Primary functions on the key caps	10
5.2. Additional symbols for shifted keys	12
5.3. Additional register indirect commands	14
6. Entering Data and Displaying Results	15
7. Modes of Operation and Features	16
“RUN” Mode	16
The Register Stack	17
BCD system and Boolean Operations	19
“Programming” Mode	21
Debugging	22
Indirect addressing revisited	23
Loop control	23
8. Calculation functions accuracy	24
9. Technical data	25
Average computation times	25
Power Supply	26
Environment considerations	26
10. Invalid Operations and Overflow	27
11. Battery Replacement	28
12. Storage and Handling	30
13. Warranty	31
14. Literature References	32

15.	Appendix 1 - Test checklist	34
16.	Appendix 2 - Command Codes.....	35
17.	Appendix 3 - Program Examples	36
	Calculating the area of a circle.....	36
	Using unconditional indirect branching GTO (i) (K, БП)	37
	Using conditional indirect branching $x < 0$ (i) (K, $x < 0$)	38
	Looping calculations using commands such as DSNZ0 (F, L0)	40
	Looping calculations using indirect load RCL (i) (K, $\Pi \rightarrow X$).....	42
18.	Appendix 4 - Warranty Card	43
19.	Appendix 5 - Tear-off Vouchers	45

Foreword

This book is an English translation of the original Russian Elektronika MK-61 Operating Instructions guide, circa 1996. There are other original versions of the Russian Elektronika MK-61 Operating Instructions guide, showing different organization and contents. This version was chosen because it was the most recent version found.

The translation process was initiated by scanning the original paperback guide at 300dpi grey scale, and later converted to Cyrillic text using the ABBY Fine Reader tool. From that point forward, the translation was a manual time consuming process, using Google Translator as the main tool, and occasional assistance from Russian and Ukraine acquaintances.

Enough care has been taken to be faithful to the original Russian text where possible. This means that some expressions have retained the Russian tone. For example, a calculator is designated as *micro calculator* and sometimes it is referred by its nick name as *MK*.

Whenever possible, the original Cyrillic keyboard symbols are retained thru this book. They are written between parentheses just after the equivalent Latin symbols.

Concerning the decimal separator symbol, despite the *comma* being used as decimal separator in the original Russian guide, the MK calculator uses the *dot* symbol printed on the decimal separator key. So in this guide the *dot* symbol was chosen in order to be consistent with the MK keyboard.

Interestingly, the RPN term never appears in the original text, despite this MK-61 being a machine that operates as such. Footnotes were added stating that this is indeed a RPN calculator, though.

To complement the original guide's information, additional detail was included, based on Internet literature searches and testing on the physical MK-61 calculator. These remarks are presented as footnotes references.

Many thanks to all contributors to this small project, namely the good people from the [MoHPC](#), and special thanks to [Sergei Frolov](#) for giving permission to include parts of his work in this book.

1. Notice to Owners

- 1.1. When you purchase a micro calculator (MK) please check for all of its delivery contents, namely the MK quality control test, the presence of the warranty card in this manual, the two tear-off maintenance vouchers in compliance with the MK serial number and warranty card, the integrity of the seals on the device, and one power supply unit (PSU).
- 1.2. Make sure you have a warranty and a detachable coupon signature (stamp) of the seller, date of sale and shop stamp.
- 1.3. If the MK needs a repair during the warranty period, use the MK maintenance vouchers and register the number and date of issue.
- 1.4. Vouchers for repair are taken by the representative maintenance organization only after the remedial work intervention.
- 1.5. Before using the MK, please read this instruction manual.
- 1.6. If MK was transported in winter conditions or stored in a cold room, please keep it for 4 hours at room temperature before turning it on.
- 1.7. A self-adhering protective film is attached to the MK display color filter, but it can be removed if necessary.
- 1.8. After turning off the MK, allow a minimum of 30 seconds before switching it on again.
- 1.9. For a more detailed acquaintance with the construction and operation of the MK, use the reference literature listed at the end of the manual.
- 1.10. The factory is constantly improving the MK, so the product specification may change and it will not be reported in the manual, without prejudice to the buyer.

2. Package Contents

- 2.1. Micro Calculator "Elektronika MK 61"
- 2.2. Instruction Manual
- 2.3. Power Supply D2-10M
- 2.4. Packaging bundle with cardboard inserts, plastic bag, and pouch products.

3. Safety Instructions

- 3.1. The power supply calculator can only be opened and repaired by qualified personnel for such procedures.
- 3.2. Plugging the power supply to the 220VAC power grid without a calculator connected to the adapter is prohibited¹.
- 3.3. The use of rechargeable batteries is prohibited.

¹ See Chapters 9 and 11 for details .

4. Micro Calculator Specifications

4.1. Program memory of 105 steps.

4.2. Featured functions:

- Arithmetic (+, −, \times , \div);
- Direct and inverse trigonometric (sin, cos, tg, \sin^{-1} , \cos^{-1} , tg^{-1});
- Logarithmic (lg, ln);
- Calculations (x^y , e^x , \sqrt{x} , x^2 , $1/x$)
- Constants (π)
- Isolation of the decimal (fractional) part of a number, determination of their absolute magnitude and sign, maximum value between two numbers, inverse of a number;
- Pseudo-random numbers generation between 0 and 1;
- Convert angular (time) values, expressed in degrees (hours), minutes, and fractions of a minute, in values expressed in degrees (hours) and fractions of degrees (hour), and vice-versa;
- Logical bitwise operations for addition, multiplication, xor, and negation.

4.3. Store data in the 15 memory registers RG0-RG9, RGa-RGe and recall them to read stored data into the stack registers; Stack registers X, Y, Z, T data rotation; exchange of data between the registers X and Y; clearing the X register and retrieve the previous value of register X after an operation execution, thru the register X1.

4.4. Perform operations requiring a single value in register X (ln, sin, among others), double values operations with numbers in registers X and Y (arithmetic, etc.), and chain operations.

4.5. Creating, editing and correction programs, commands for direct and indirect jumps, call to subroutines with handling of up to 5 levels of depth, commands for direct and indirect unconditional jumps, conditional jumps, looping commands, indirect entries in registers, indirect branching on the contents of registers, program counter reset command, start and stop execution in the Run mode, commands to step through a program, display up to three program code steps per line with indication of the current address of the program counter.

5. Keyboard and Switches Description

Latin keyboard layout transcription










Original Cyrillic keyboard layout



5.1. Primary functions on the key caps

The exterior view of the original Russian MK-61 is shown in Fig.1 below.

 to ,  - Numeric keys for data entry into the register X, using digits 0 through 9, plus the decimal point.

, , ,  - Keys to do arithmetic operations² (addition, subtraction, multiplication and division).

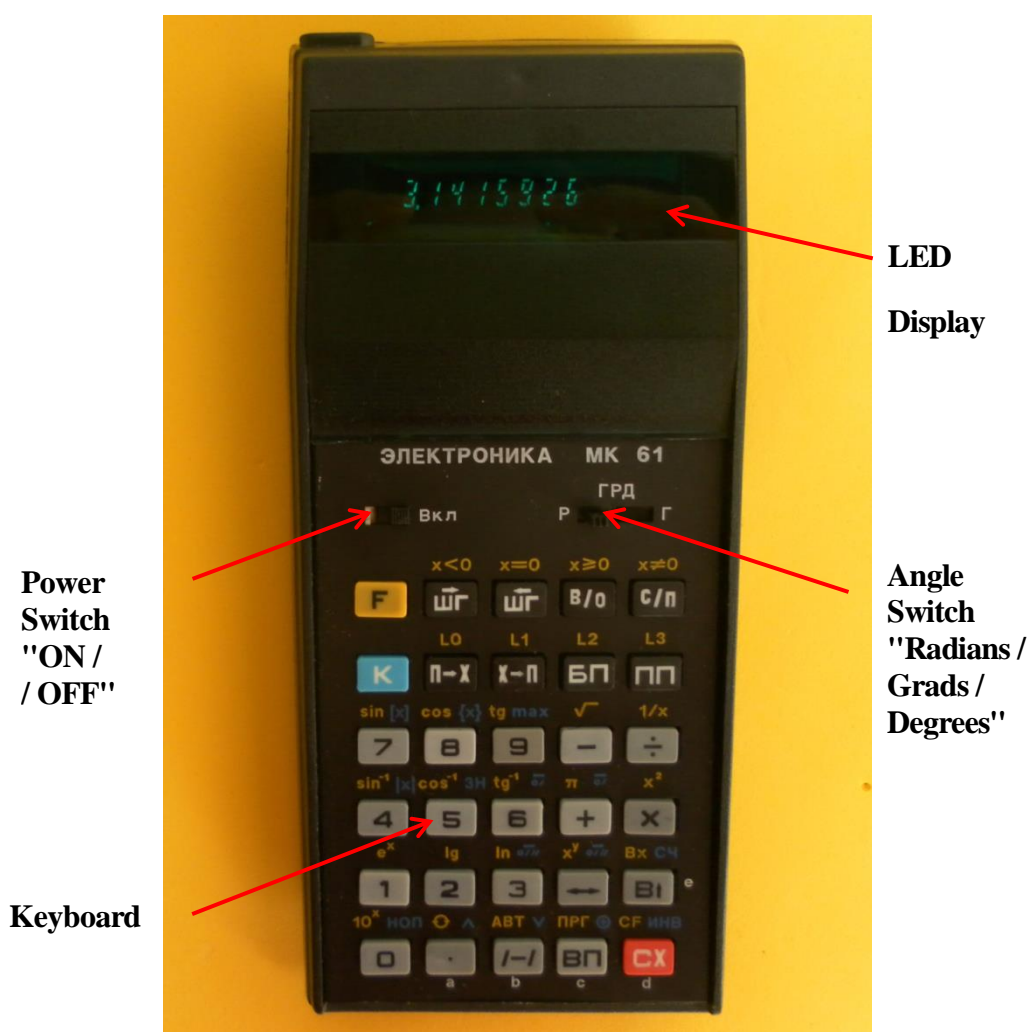


Fig.1. Exterior view of the micro calculator (MK)

² This is a RPN calculator, therefore it uses postfix notation where the parentheses and the equal keys are not needed. To do a math operation, first enter the operand(s), then chose the operator.



- From left to right:

The **ENTER** (**B↑**) key is used for data entry into stack register X;

The **EEX** (**BΠ**) key is used to input the exponent of a number;

The **CHS** (**/-/**) key change the sign of mantissa or exponent;

The **x↔y** (**↔**) key exchange data contents between the registers of X and Y;

The **CX** key clears the contents of register X.



- The **SST** (**IIIΓ→**) and **BST** (**IIIΓ←**) keys are used to single-step through a program.



- The **STO** (**X→II**) key stores the X register content in one of the addressable memory registers. The **RCL** (**II→X**) key loads the register X with one of the addressable memory register's contents³.



- In "PROGRAM" mode, the **GTO** (**BΠ**) key specifies an unconditional jump, while the **GSB** (**III**) key indicates a call to a subroutine. In "RUN" mode these keys step through a program.



- In "PROGRAM" mode, the **RTN/0** (**B/0**) key returns from a subroutine, while it goes to address zero when in the "RUN" mode.



- In "PROGRAM" mode, the **R/S** (**C/II**) key is used to stop a program execution and display the register X contents. In "RUN" mode, it starts or stop a program or a loop cycle.






- The **F** and **K** keys select the shift functions, whose symbols are printed above the other MK's keys in yellow (**F**) or blue (**K**) color.





³ The MK-61 has got 15 addressable memory registers (RG0 to RG9, and "RGa" to "RGe") for additional data storage and to support a number programming features like looping control and indirect branching.






The calculator operations uses a primary memory area: The stack with 4 registers (X, Y, Z, T), plus one additional register X1 used to automatically save the previous register X value when executing a command that modifies the register X contents.

5.2. Additional symbols for shifted keys

  - The **F, CF** combination cancels the shift key selection.

  - The **F, LST x (F, Bx)** combination retrieve the previous value of X after an operation execution.



  ;   - The **F, RUN (F, ABT)** combination selects the “RUN” mode, while **F, PRG (A, PIPT)** enters into the "PROGRAM" mode.

  ;   ;   ;  

These four key combinations **x<0**, **x=0**, **x≥0**, **x≠0** gives direct access to the four corresponding logic conditions.

  ;   ;   ;  

These four key combinations **F, DSNZ0 (F, L0)**, **F, DSNZ1 (F, L1)**, **F, DSNZ2 (F, L2)**, **F, DSNZ3 (F, L3)** allows loop control by using conditional direct branch depending on the contents of register 0, 1, 2 or 3, respectively.

  - The **F, R↓** combination rotates data thru the 4-stack registers.

  ;   ;   ;   ;  

- From left to right, these five key combinations allow to determine:

The integer part: **K, INT (K, [x])**

The fractional part of a number: **K, FRAC (K, {x})**

The absolute magnitude: **K, |x|**

The sign: **K, sign (K, 3H)**

Find the maximum of two numbers: **K, max**

 ,  ;  ,  - From left to right:









The combination **K, H \leftrightarrow H.M.S** converts from sexagesimal degrees (hours), minutes, seconds and fractions of seconds, into decimal angular (time) values.

The combination **K, H \rightarrow H.M.S** converts from decimal angular (time) values into sexagesimal degrees (hours), minutes, seconds and fractions of seconds.

 ,  ;  ,  - From left to right:

The combination **H \leftrightarrow H.M** converts from sexagesimal degrees (hours), minutes, and fractions of a minute, into decimal angular (time) values.

The combination **H \rightarrow H.M** converts between decimal angular (time) values into sexagesimal degrees (hours), minutes, and fractions of a minute.

 ,  ;  ,  ;  ,  ;  , 

These four key combinations are used for bitwise logical operations. From left to right:

Multiplication: **K, AND**

Addition: **K, OR**



Exclusive or: **K, XOR**



Negation (one's complement): **K, NOT (K, ИHB)**

 ,  - This key combination **K, RND (K, CЧ)** generates pseudo random numbers.




 ,  - This key combination **K, NOP (K, НОП)** invokes a "No Operation".



5.3. Additional register indirect commands



  - This key combination **K + GTO(i) (K, BII)** does an unconditional indirect branch by jumping to an address stored in one of the registers RG0 to RG9 and RGe to RGe.



  - This key combination **K, X<0 (i)** does a conditional indirect jump to the address stored in one of the registers RG0 to RG9 and RGe to RGe.

Similar operations are carried out under the conditions for indirect branching for the remaining three commands:

  ;   ;  

  - This key combination **K, GSB(i) (K, IIII)** does an indirect call to a subroutine according to the address stored in one of the registers RG0 to RG9 and RGe to RGe.

  - This key combination **K, RCL(i) (K, II→X)**⁴ does an indirect load of register X with the contents of one of the 15 registers whose number is specified in one of the registers RG0 to RG9 and RGe to RGe.

  - And of course the indirect store operation is available as well. This key combination **K, STO(i) (K, X→II)** takes the value of register X and does an indirect store into one memory register referenced by one of the registers RG0 to RG9 and RGe to RGe.

⁴ Suppose RG7 contents = 123; we want to load stack Reg. X with the RG7 contents by using indirect addressing, and we chose RG0 as the pointer. What is the value to assign to RG0?

In this case, store the value 8 into the RG0 in order to retrieve the RG7 contents at the first try. Why? Because when the instruction executes, first RG0 is **modified and decremented by one**, before looking into its indirect pointer value.

This “decrement by one” behavior is only valid for register pointers RG0 to RG3, and applies to all instructions using them.

However, this “decrement by one” sometimes takes place **before** the lookup, for example for instructions like the **RCL(i)**. And for other instructions, like **GTO(i)**, it happens **after** the lookup.

As for the registers RG4 to RG6, they are first **incremented by one** before the lookup operation.

However, registers RG7-RG9 and RGe-RGe are **modified but the value is not changed** during execution.

Registers RG0-RG9, RGe-RGe correspond to **modified codes** from 0000000 to 0000014 (using leading 0's)


6. Entering Data and Displaying Results


6.1. MK numeric system - Decimal. The information is displayed on the 12 digits luminous discharge type display, sporting a green glow; it has 8-digits for mantissa, 2-digits for exponent, 1-digit for mantissa sign and another digit for exponent sign.


6.2. The MK calculation range is: $1 \times 10^{-99} \leq x \leq 9.9999999 \times 10^{99}$

The results of calculations in the range of $1 \leq x \leq 99999999$ are displayed in the natural form, and outside this range are displayed as floating-point numbers.

6.3. The data is introduced in the MK by pressing the appropriate keys. To introduce fractional numbers, first type the integer part, then the decimal point, and then the fractional part.

6.4. When you enter a number with exponent, the mantissas is introduced first and then press the key  **EEX (BII)** to enter the exponent value.

To change mantissa or exponent signs, press the key  **/-/** after typing the values.

6.5. Pressing the key  **ENTER (B↑)** normalizes the inputted number in stack Reg. X, and a copy of the register X will go into register Y as well⁵.



6.6. After you enter an eight-digit mantissa number, further pressing additional number keys does not change the display reading.


⁵ The 4-register stack lift operation is automatic on the ENTER key execution: The Reg. Y content is also copied to Reg. T, and the previous Reg. T contents is lost.

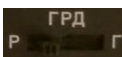
7. Modes of Operation and Features

7.1. The MK works in one of two modes: "RUN" or "PROGRAMMING".

"RUN" Mode




7.2. The "Run" mode is set immediately after switching ON the MK, or after clicking the key combination   **F, RUN (F, ABT)**

7.3. When performing single number operations, first type the numeric argument on the keyboard, then press the key  **F** and then press the corresponding function.


7.4. Before performing the direct or inverse trigonometric functions, set the switch  **RAD / GRD / DEG (P / ГРД / Г)** to the desired angles units.



7.5. When converting time (angle) values, the hours (degrees) value is separated from the value for minutes, seconds and fractions of second by the decimal point. For example, the following time value:

142.24314 – Represents 142 hours, 24 minutes, and 31.4 seconds⁶.


7.6. To call a pseudo-random number to be displayed, type in a seed number and press the keys    **ENTER, K, RND.**

When the MK memory is reset, the generated number will be equal to 0.404067.

7.7. When performing double number operations, start by typing the first number on the keypad, next introduce it with the key  **ENTER**, then type the second number, and press the button for the desired function. The display will show the result of the operation.

⁶ To get this time result, type the decimal value 142.40874 and then choose   **H→H.M.S**

The Register Stack

7.8. The MK has got a memory stack with the registers X, Y, Z, T. After switching on the MK, there is no data in these registers. Entering information into the stack always occurs through the register X. Pressing the key  **ENTER** will copy the number in register X into register Y, and Y into Z, and Z into T. In this way, the content of the register X is preserved⁷.



7.9. Pressing the key combination ,  **F, R↓**, data rotates through the stack:

$$X \rightarrow T \rightarrow Z \rightarrow Y \rightarrow X$$

7.10. When performing single number operations, the MK operates with a number from register X, and the contents of registers Y, Z, T are not modified.

The operation result is stored back in register X, but a copy of its previous content is copied into the register X1 – This register X1 saves the previous register X content from being lost as a result of another operation execution.

In this way, the result of one single number operation remains in register X and its value is shown in the MK display.

7.11. To check the content register X1 press key the combination ,  **F, LST x (F, Bx)**.

It is cleared (or cancel any other operation command) by using the key  **CX**.






⁷ RPN calculators are not created equal. Each different RPN machine vendor and model behaves in its specific and unique way. RPN was adopted by HP since the beginning of the slide rule obsolescence with their HP-35 calculator invention. And since then, HP calculators did evolve with the passing time, and there are at least three different behaviors handling the stack operations, depending on calculator models.


What about the MK-61 RPN register stack operations behavior?



Well, running the stack lift test (**5, ENTER, CHS, 2, +**) mentioned in the **HP Solve #27 “HP RPN Evolves” article by Richard J. Nelson**, the result is **7**.

This places the MK-61 in the same league as the HP Classics, like the famous HP-25 or HP-65 machines.

7.12. The four stack registers and the previous X value in register X1 facilitate complex calculations⁸.

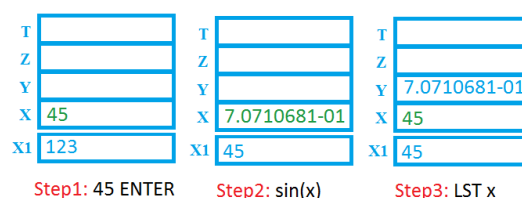
7.13. To store the register X contents into one of the additional 15 memory registers, use the key  **STO** ($X \rightarrow \Pi$) and choose the destination register number with one of the keys  to  and  to .

To load register X with a number from a memory register contents, press key  **RCL** ($\Pi \rightarrow X$) followed by one of the source register numbers.

7.14. To clear a memory register by writing zero into it, use the key  **CX** followed by the key  **STO** ($X \rightarrow \Pi$) and choose the destination register address number.

To clear all MK memory registers, simply turn off the calculator⁹.

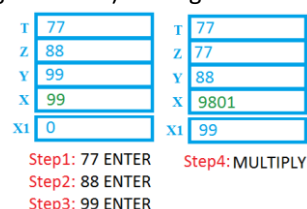
⁸ MK-61 RPN stack manipulation details are beyond the scope here, but let's see a couple of examples. The stack Register X is linked to the Display, so its content is always visible. When typing a number, it is being entered into Register X. There is no need to do an ENTER to execute single number operations.



On the above example, when executing $\sin(45)$, it first copies reg. X contents into Reg. X1, then calculates the $\sin(45)$ function, and the result is put back into Reg. X.

If we retrieve the previous Reg. X value, it will invoke an automatic stack lift up (X goes into Y, Y into Z, Z into T, and previous T content is lost), to free up Reg. X to receive the LST x operator result.

On the next example, we pushed 3 numbers into the Y, Z, T stack registers, leaving $X=Y$. A Multiply consumes X and Y as operands, leaving the answer in X. During the execution, an automatic stack lift down was executed (T goes into Z, Z goes into Y) leaving T and Z with identical contents.



⁹ After switching off the calculator, please allow at least 30 seconds before turning it back on. See Chapter 1. Notice to Owners.

BCD system and Boolean Operations

7.15. Numbers in the MK are implemented in a BCD system, so Boolean operations can be used on them. The binary values for each number and respective display character are shown in Table 1.

















Information in registers		Symbol on the display
In the decimal system	In the BCD system	
0	0000	
1	0001	
2	0010	
3	0011	
4	0100	
5	0101	
6	0110	
7	0111	
8	1000	
9	1001	
10	1010	
11	1011	
12	1100	
13	1101	
14	1110	
15	1111	 (blank)

Table 1 – BCD system symbol display

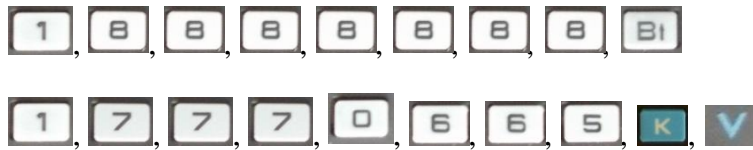
7.16. The output from bitwise logical operations execution is signaled by the presence of the digit “8” on the leftmost part of the displayed number.

When you enter values in register X and Y for Boolean operations, it is also necessary to type in a digit other than 0 in the most significant digit of the inputted numbers.

7.17. Example for a bitwise logical addition:

8888888 **OR** 7770665

Keystrokes:



Please note an extra digit, in this case with the value 1, was typed in to fill the leftmost position of each number to be evaluated by a Boolean operator.

The result on the vacuum fluorescent display will show a value of¹⁰:



Note the digit 8 in the leftmost position of the result – This is the sign of a Boolean logical operation.



¹⁰ These are two numbers having 7-digit with 4-bit each, to be added with the Boolean bitwise OR operator.

As for the result, we get 7-digits with 4-bit each with the following values: **15, 15, 15, 8, 14, 14, 13**

According to the “Table 1 – BCD system symbol display”, we expect to see these symbols on display:

- Three blank characters;
- An “8”
- Two “E”’s
- A “Γ”

“Programming” Mode

7.18. To go into "PROGRAM" mode, press ,  **F, PROG (F, ΠΓ)**.

In this mode, the codes for numbers, operations and commands¹¹ are written into the MK 105 memory cells, from addresses 00 to 104¹².

7.19. The MK has an internal Program Counter (PC) that changes its content with the specified jump addresses or recorded in the registers RG0-RG9, RGa-RGe.


7.20. The MK shows the current address of the PC at the rightmost position of the display¹³.

7.21. To create complex programs with subroutines, there are commands for direct and indirect branching, indirect subroutine calls and data storing, and loop controls. When typing these commands in the program one should specify the branch address (except for indirect branching commands, which already contains the address).

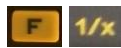
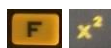
¹¹ See Appendix 2 - Command Codes

¹² In reality step 00 is not available to store any instructions. The first memory address available to store a command is 01. See the MK-61 behavior below for details.

¹³ Up to three of the previously entered commands are displayed on the next positions from right (at a lower PC address) to left (at a higher PC address). The command at the current PC address is located at the leftmost position of the display. Each command takes 2 digits on the display.




Example to code $1/\pi^2$ (a complete program needs to be terminated by a  R/S command, though).

Keystrokes





Result on the display



7.22. Addresses branching from 00 to 99 are typed by pressing the corresponding number keys. Addresses branching from 100 to 104 are used only for direct branching commands and are typed using the key  and one of the keys from  to .

The addresses from 00 to 99 are displayed using the appropriate numbers, but for the addresses from 100 to 104 the two digits used at hundreds and tens positions are displayed with a "minus" sign. For example, the address 102 is displayed as -2.


7.23. To begin a program at address zero, select the "RUN" mode and reset the Program Counter to zero by pressing  **RTN/0 (B/0)**, and then switch to "PROGRAM" mode. The display will show the address 00.

7.24. If you want to start a program at any specific address, select the "RUN" mode and press the key  **GTO (BΠ)** followed by the required address, and then turn into the "PROGRAM" mode.

Debugging

7.25. Debugging programs is done in "RUN" mode by pressing key  **SST (ΠΠ→)**.

Typically the required raw data needs to be entered in the appropriate registers, and then set the start address of the program.

Finally execute the program in single step mode by using the key  **SST (ΠΠ→)**.



Alternatively run the program in automatic mode by using the key  **R/S (C/Π)**.

In the latter case, the display blinks during the program execution.

WARNING!


When you turn off the MK, programs and data are lost.

Indirect addressing revisited



7.26. When executing commands for indirect unconditional jump   **GTO (i)** followed by the address register), modification occurs in the contents of the address register after the execution of the command, pointing to a new address.

If the command uses the address registers: ¹⁴





- RG0-RG3 - their content is decremented by 1;
- RG4-RG5 - their content is increased by 1;
- RG7-RG9 and RGe-RGe – Their content does not change.

7.27. When using the indirect store   **STO(i)** (**K**, **X**→**Π**), modification also occurs on the contents of addressable register, and writes the contents of register X into the address that is pointed to by the modified code.

Registers RG0-RG9, RGe-RGe correspond to modified codes ¹⁵ from 0000000 to 0000014.

7.28. When using the indirect load command   **RCL(i)** (**K**, **Π**→**X**), modification occurs on the contents of addressable register, and loads register X with the contents of the address that is pointed to by the modified code.

Loop control

7.29. Computing cycles can be implemented by using the commands   **DSNZ0** to   **DSNZ03**, as referenced in the register RG0 to RG3, respectively.

For each loop, the content of the address register is subtracted by 1 and test if its content is equals to 0.

If address register equals 0 then execution continues after the next instruction, otherwise execution jumps to the specified address.

¹⁴ As far as we can tell, the behavior for branching instructions is:

- Indirect addresses for RG0-3, the value is decremented **after** evaluating the branch address;
- Indirect addresses for RG4-6, the value is incremented **before** evaluating the branch address;
- Indirect addresses for RG7-e, the value is maintained but the branch address is incremented by 1.
- Conditional and unconditional direct branch, the address value is incremented by one.

¹⁵ See Chapter 5.3 – Additional register indirect commands

8. Calculation functions accuracy

Table 2

Function	Permissible argument values	Maximum relative error
$\sin x$	$10^{-99} \leq x \leq 10^{10}$	$3 \cdot 10^{-7}$
$\cos x$	$10^{-99} \leq x \leq 10^{10}$	$3 \cdot 10^{-7}$
$\operatorname{tg} x$	$10^{-99} \leq x \leq 10^{10}$	$3 \cdot 10^{-7}$
$\arcsin x$	$ x \leq 1$	$3 \cdot 10^{-7}$
$\arccos x$	$ x \leq 1$	$3 \cdot 10^{-7}$
$\operatorname{arc} \operatorname{tg} x$	$ x \leq 9.9999999 \cdot 10^{99}$ $ x \geq 1 \cdot 10^{-99}$	
\max	$y \neq 0; x \neq 0$	
x^y	$0 < x$	$1 \cdot 10^{-6}$
e^x	$ x < 100 \ln 10$	$1 \cdot 10^{-7}$
x^2	$ x < 10^{50}$	$1 \cdot 10^{-7}$
10^x	$ x \leq 99.999999$	$4 \cdot 10^{-7}$
$1/x$	$x \neq 0$	$1 \cdot 10^{-7}$
\sqrt{x}	$0 \leq x$	$1 \cdot 10^{-7}$
$\ln x$	$0 < x$	$4 \cdot 10^{-7}$
$\lg x$	$0 < x$	$4 \cdot 10^{-7}$
$H \rightarrow H.M$	$-1 \cdot 10^{-99} \leq x < -0.06$	$1 \cdot 10^{-7}$
$H \leftarrow H.M$	$-0.06 < x \leq 9.9999999 \cdot 10^{99}$	$1 \cdot 10^{-7}$
$H \rightarrow H.M.S$	$-1 \cdot 10^{-99} \leq x < -0.06$	$1 \cdot 10^{-7}$
$H \leftarrow H.M.S$	$-0.06 < x \leq 9.9999999 \cdot 10^{99}$	$1 \cdot 10^{-7}$

9. Technical data

Average computation times

9.1. Computation¹⁶ time for arithmetic, $1/x$, \sqrt{x} , and x^2 operations: No more than 0.5 seconds.

Average computation time for logarithmic calculations and direct and inverse trigonometric functions: No more than 2 seconds.

Time calculation for function x^y : No more than 3.5 seconds.

¹⁶ Elektronika MK-61 hardware details:

New MK-61's are supplied with operating instructions including the schematic diagram, where we can identify the main active electronic components.

The Integrated Circuits:

D1: K745IK1302-2 (K745IK1302-2) - Control Unit for Processor, Display and Keyboard;
 D2: K745IK1303-2 (K745IK1303-2) – One-bit Arithmetic Processor; clock frequency around 100KHz;
 D3: K745IP2-2 (K745IR2-2) - Memory Registers;
 D4: K745IP2-2 (K745IR2-2) - Memory Registers;
 D5: K745ГФ3-2 (K745GF3-2) - Clock Generator;
 D6: K745IK1306-2 (K745IK1306-2) - Additional MK-61 features not present on the B3-34 series;

The Vacuum Fluorescent Display:

VL1: ИЛЦ2-12/8Л (ILC2-12/8L or ILZ2-12/8) - Vacuum Fluorescent Display; Filament voltage: 2.15 - 2.9 V at 20 to 24mA; Brightness: 700 cd/m²; Number of digits: 12 (8 mantissa + 2 exponent + 2 sign digits); Size of the digits: 2.4 x 4.3 mm; Color: Green; Lifetime: more than 10000 hours;

The Power Supply Transistors:

VT1: кТ361Г (KT361G) - PNP Si 250MHz, hfe>50, 35V, 0.05A, 0.15W
 VT2: кТ315Б (KT315B) - NPN Si 250MHz, hfe>50, 20V, 0.1A, 0.15W (schematics wrongly specify it as PNP);
 VT3: кТ814Г (KT814G) - PNP Si 3MHz, hfe>60, 50V, 3A, 10W
 VT5: кТ315Б (KT315G) - NPN Si 250MHz, hfe>50, 35V, 0.1A, 0.15W

Power Supply

9.2. The MK power supply¹⁷ takes three battery cells¹⁸ model A-316 "Quantum" or "Prima", and it can use an external power adapter model D2-10M¹⁹ (or similar) to be connected to an 220VAC (50 ±1Hz) power outlet.

9.3. Power consumption from the three battery cells A316 is less than 0.6 Watt²⁰.

Environment considerations

9.4. Operating temperature from 10 to 35°C at a relative humidity of 50 to 80% and an atmospheric pressure from 86 to 106 kPa.

9.5. The MK weight (without batteries and power adapter) is less than 250 g.

9.6. The MK physical dimensions are not exceeding 170 x 80 x 38 mm.

9.7. Content of precious metals in the MK: Gold - 0.06455 g, Silver - 0.04274 g

¹⁷ The MK-61 internal power supply includes a DC-DC converter circuit, taking the 4.5V battery supply to generate a number of negative output supply voltages for the electronics, consuming between 100mA and 130mA depending on how many digits are on. The 4.5V battery Positive terminal is used as common GND (ground).

MK-61 DC-DC converter pin out

Common connection pin:

+BATT/GND (**pin9**): Connected to Gnd/+5V to IC's.

Input 4.5V battery supply voltage:

-V1 -BATT (**pin3**): Connected to the Negative terminal of the 4.5V battery.

Output negative supply voltages:

-V2 -15V (**pin4**): Connected to IC's logic circuits supply

-V3 -27V (**pin7**): Connected to the VFD display

~V4: 2.4VAC (**pin 1 and 2**): Connected to the VFD display filament

¹⁸ Western 1.5V type AA LR6 Alkaline or similar battery cells can safely be used as well.

¹⁹ Output under normal calculator load of 100mA: 5.2VDC. Output voltage in open circuit: 8.5VDC.

²⁰ Calculator current consumption varies from about 100mA (one digit turned on) to 130mA (12 digits turned on).

10. Invalid Operations and Overflow

10.1. Invalid operations include division by 0, elevating 0 or a negative number to any power, square of a negative number, logarithms of 0 or a negative number, calculation of direct and inverse trigonometric functions outside its domain, and transfer time (angular) variables for values in minutes (seconds) larger than 50, results in an error²¹ message presented in the vacuum fluorescent display:



10.2. The ERROR signal is also displayed when the calculation result exceeds the value $+9.9999999 \times 10^{99}$.

If the result of the calculation is less than 10^{-99} , the register X is set to zero.


10.3. After the appearance of an error condition, you can continue to enter numbers and perform calculations on the MK.

10.4. Using the results of logical operations as an argument in cases where these are outside the domain of the calculated functions, can lead to incorrect operation and erratic MK behavior (crashes, infinite loops, incorrect results, etc.).


²¹ This error message makes sense in English assuming Latin characters. But this is a Russian calculator where the error message can be wrongly read as Cyrillic text and then it translates to “EGGOG”.

Like MK-52, the MK-61 inherits from the Elektronika B3-34 series. All are famous, and one of the reasons is related to this “EGGOG” message, as it allows hacking and exploring the “EGGOG” state to great extent, taking advantage of undocumented “features”. It happens that not every “EGGOG” state is a real error. One exploit is to manipulate big numbers (larger than 9.9999999×10^{99}).



From number exponents between 100 and 200, the display shows  but the numbers can be manipulated in the registers. They are too large to be displayed, hence the error message.



From exponents above 200 the display can show a . Now we have a “ZGGOG”!

One way to find out where the displayed error is genuine or is a big number, is to do a **K, NOP**. If the error message goes away it is a genuine error (like divide by zero), else it will stay on display.


Experts in this calculator series have shared some information on the subject, for instance here:


- How to calculate on the MK B3-34, by Sergei Frolov <http://www.leningrad.su/b334/b334doce.htm>
- “MK B3-34 Soviet Calculator – Riddle” by Sergei Frolov <http://www.leningrad.su/b334/b334e.htm>
- “Pseudo-writing mode”, by Gregory Escov <http://gregescov.tripod.com/psch.htm>
- “The journey of the big numbers”, by Gregory Escov <http://gregescov.tripod.com/bignums.htm>

11. Battery Replacement

11.1. When replacing batteries, the MK must be turned off and the external power adapter disconnected from the outlet.

11.2. Pay attention to the polarity of the batteries to match with the label signs inside the MK battery compartment (Figure 2).

11.3. Remove the battery compartment cover by sliding it out and insert the three fresh batteries. Put back the cover and turn on the MK power switch  **ON (BKJ)**.

The leftmost display digit should light up to show .

11.4. If the MK display shows up decimal points, it means that the battery level is low and it should be replaced.

WARNING

Do not leave dead batteries in the MK in order to avoid damage arising from the electrolyte leaking – The MK cannot be repaired²² in these conditions.

11.5. When using an external power adapter, first connect it to the MK²³, and then plug the adapter to the 220 VAC power outlet.

To stop using the power adapter, first unplug it from the 220 VAC outlet and then disconnect it from the MK.

²² Troubleshooting procedures for the most common problems:

- Check the obvious point of failures – Battery leaking, oxidized or corroded battery terminals, broken power supply wiring, and power switch – Clean it and make sure good continuity is restored.
- Check for normal current consumption – It varies from about 100mA (one digit turned on) to 130mA (12 digits turned on). If the value is outside this threshold, investigate the DC-DC converter.
- Check the DC-DC power converter output voltages. The IC's requires -15VDC, and the VFD requires two voltages (-27VDC and 2.4VAC) - see details on Chapter 9 – Technical Data – Power Supply. Defective electrolytic capacitors are a common cause for erratic or total operation failure.

²³ The reason for this procedure is related to the fact that the Power Adapter model "Elektronika D2-10M" output voltage is not regulated. In open circuit it will present about 8.5VDC, while the calculator nominal working voltage is only 4.5VDC.

Connecting the Power Adapter to the calculator with 8.5VDC will stress the internal DC-DC converter a bit, with eventual hardware failures arising from this transitory overvoltage.

However the Power Adapter will present around 5.2VDC for workloads of about 110mA and 4.9VDC at 145mA, meaning that if you follow the above manufacturer recommendation, the calculator will never see more than 5.2VDC and it will run safely.



Use your thumb to gently press the protruding rib portion of the MK battery compartment cover and carefully slide it up.



Put fresh A-316 batteries into the battery compartment according to the label on the bottom of the compartment, and then insert the cover into the slots and slide it into place.

Fig. 2 Install the batteries A-316 ²⁴

²⁴ Western 1.5V type AA LR6 Alkaline or similar battery cells can safely be used as well.

12. Storage and Handling

12.1. Use the supplied transport packaging to protect the MK from mechanical damage, dust, moisture, and excessive climatic influences.

12.2 The MK should be stored in a dry heated room at a temperature between 5 and 35°C, and a relative humidity of less than 85%.

The MK should not be exposed to aggressive environments where substances like acids, alkalis, and others, are present.

13. Warranty

- 13.1. The manufacturer guarantees the MK to the technical specifications subject to the operating rules. Warranty period - 24 months from the date of sale.
- 13.2. In the absence of guarantee vouchers and coupons sales date and shop stamp, the MK warranty period is calculated from the date of manufacture of the MK manufacturer.
- 13.3. During the warranty period, for the first and second repairs use the detachable coupons. Subsequent repairs are carried out under the act.
- 13.4. A MK without warranty card or detachable coupons, as well as broken seals, is not covered under the warranty service.
- 13.5. MK exchange through sales network help repair facilities and the warranty card under the rules of the exchange of manufactured goods.
- 13.6. Address repair facility can be found in the shop where the MK was sold, as well as at the plant – Manufacturer phone: 2-28-54.

14. Literature References

Astanin L. Yu, Daren Yu. - "Using programmable calculators for scientific and engineering calculations" - Ed.: Energoatomizdat, 1986 - 157 pages (*Астанин Л. Ю. ,Дарений Ю.Д. Применение программируемых микрокалькуляторов для научных и инженерных расчетов*)

Bloch AS, Pavlovsky AI, VV Penkrat – "Programming calculators" - Minsk, 1981 - 192 pages (*Блох А.Ш., Павловский А.И., Пенкрат В.В. Программирование на микрокалькуляторах. - Минск: Выш. шк.*)

Boyko A.B. – "Games with the calculator" – Ed.: Znanie, 1987 - 46 pages (*Бойко А.Б. Игры с микрокалькулятором -М.: Знание.1987 -46 с.*)

Danilov, ID – "Secrets of the programmable calculator" – Ed.: Nauka, 1986 -160 pages (*Данилов И.Д. - Секреты программируемого микрокалькулятора -М.:Наука,1986 - 160 с.*)

Dyakonov V.P. – "Calculation of non-linear and switching devices on programmable calculators" – Ed. Radio and Communication, 1964, 170 pages (*Дьяконов В.П. Расчет нелинейных и импульсных устройств на программируемых микрокалькуляторах. - М., Радио и связь,1964 г.-170 с.*)

Dyakonov V.P. – "Handbook of calculations on calculators" –Ed.: Science, 1985 - 224 pages (*Дьяконов Б.П. Справочник по расчетам на микрокалькуляторах - М.:Наука,1985 - 224 с.*)

Krol G. – "What can my calculator do?" –Ed.: Mir, 1981 -132 pages (*Кроль Г. Что умеет мой микрокалькулятор? -М.:МИР,1981 -132 с.*)

"Calculators: Specifications and design characteristics" - EJ Kuznetsov, V.V.Ostetsov, Minikin YN, Yu.N.Egorova -Ed.: Radio and communication, 1984 128 pages - (Ser. inzhenera.Vyp.40 Public Library) (*Микрокалькуляторы: Технические и конструктивные характеристики. Е.Ю. Кузнецов, В.В.Остецов, Миникин Ю.Н., Ю.Н.Егорова - М.:Радио и связь,1984 - 128 с. - (сер. Массовая библиотека инженера.Вып.40))*

V.F.Ochkov, V.N.Hmelyuk – "From a calculator to a personal computer" -Ed.: MEI 1990 - 224 pages (*В.Ф.Очков, В.Х.Хмелюк. От микрокалькулятора к персональному компьютеру. Изд. МЭИ, 1990 - 224 с.*)

Pukhnachov Y., Danilov N.D. – “Calculators for all” –Ed. Knowledge, 1986 - 192 pages (*Пухначев Ю.В., Данилов Н.Д. Микрокалькуляторы для всех. - М.,Знание,1986 - 192 с.*)

Trohimenko Y.K. - “Programming the MK-61 and MK-52” –Ed.: Radio and communication, 1987 - 209 pages (*Трохименко Я.К. Программирование микрокалькуляторов МК-61 и МК-52. -М.:Радио и связь, 1987 - 209 с.*)

Trohimenko Y.K., Lubich F.D. – “Engineering analysis on calculators” - Kiev, Technology, 1980 - 394 pages (*Трохименко Я. К. ,Любич Ф.Д. Инженерные расчеты на микрокалькуляторах - Киев,Техника, 1980. - 394 с.*)

























































Trohimenko Y.K., Lubich F.D. – “Engineering analysis on programmable calculators” - Kiev: Technology. 1985 - 328 pages (*Трохменко Я.К., Любич Ф.Д., Инженерные расчет на программируемых микрокалькуляторах - Киев: Техника. 1985 -328 с.*)



Trohimenko Y.K., Lubich F.D. – “Radio engineering analysis on calculators” - Ed.: Radio and communication, 1983 - 206 pages (*Трохименко Я. К., Любич Ф.Д. Радиотехнические расчеты на микрокалькуляторах . М.: Радио и связь, 1983 - 206 с.*)

Trohimenko Y.K., Lubich F.D. – “Programming the calculators "Elektronika MK 52" and Elektronika MK 61""- Kiev: Technology. 1987 – 208 pages (*Трохименко Я.К., Любич Ф.Д., Программирование микрокалькуляторов "Электроника МК 52" и Электроника МК 61" - Киев: Техника.1987 - 208 с*)

Chupka A.M. – “Calculation – Engineering operations on programmable calculators” - Ed.: Machinery, 1968 - 143 pages (*Чапка А.М. Расчетно - проектировочные работы на программируемых микрокалькуляторах - М.: Машиностроение,1968 - 143 с.*)

15. Appendix 1 - Test checklist

Slide the Switches (ON) "BKJ", (RAD/GRAD/DRG) "P/ГРД/Г", tap the keys	Digits Lit on Display											
	1	2	3	4	5	6	7	8	9	10	11	12
1. "BKJ" (Power ON)		0										
2.      		123456										
3.     		78BΠ9/-										
4.    		B 0.9										
5. 		x										
6. "Г" DEG   (sin)		sin										
7. "ГРД" GRAD   (cos)		cos										
8. "P" RAD     (tg, STO 1)		tg I-Π 1										
9.   (π)		π										
10.     (lg, STO 2)		lg I-Π 2										
11.   (RCL 1)		Π-X 1										
12.   (RCL 2)		Π-X 2										
13.   (enter PROG mode)		ΠΓ										
14.   (max)		max										
15.   (abs)		x										
16.   (STO 3)		I-Π 3										
17.   (x^y)		x^y										
18.   (OR)		V										
19.   (DSNZ3)		L3										
20.  		04										
21.  (R/S)		C/n										
22.    (exit RUN mode, RTN/0)		ABT B/O										
23.  (R/S) Test executes in less than 5sec.		C/n										
24.  (CHS)		/-										

NOTE! To perform the shifted functions printed in yellow (or blue) color, you need to press the keys  (or ) to access those operations.

16. Appendix 2 - Command Codes

1 st digit code	2 nd digit code														
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E
0	0	1	2	3	4	5	6	7	8	9	.	/-	EEX	CX	ENTER↑
1	+	-	*	/	$x \leftrightarrow y$	$F 10^x$	$F e^x$	$F \lg$	$F \ln$	$F \sin^{-1}$	$F \cos^{-1}$	$F \operatorname{tg}^{-1}$	$F \sin$	$F \cos$	$F \operatorname{tg}$
2	$F \square$	$F \sqrt{}$	$F x^2$	$F 1/x$	$F x^y$	$F R \downarrow$	K H←H.M				K H→H.M.S				
3	K H←H.M.S	K x	K sign	K H→H.M	K INT	K FRAC	K max	K AND	K OR	K XOR	K NOT	K RND			
4	STO 0	STO 1	STO 2	STO 3	STO 4	STO 5	STO 6	STO 7	STO 8	STO 9	STO a	STO b	STO c	STO d	STO e
5	R/S	GTO	RTN/0	GSB	K NOP			$F x \neq 0$	F DSNZ2	$F x \geq 0$	F DSNZ3	F DSNZ1	$F x < 0$	F DSNZ0	$F x = 0$
6	RCL 0	RCL 1	RCL 2	RCL 3	RCL 4	RCL 5	RCL 6	RCL 7	RCL 8	RCL 9	RCL a	RCL b	RCL c	RCL d	RCL e
7	K $x \neq 0$ (0)	K $x \neq 0$ (1)	K $x \neq 0$ (2)	K $x \neq 0$ (3)	K $x \neq 0$ (4)	K $x \neq 0$ (5)	K $x \neq 0$ (6)	K $x \neq 0$ (7)	K $x \neq 0$ (8)	K $x \neq 0$ (9)	K $x \neq 0$ (a)	K $x \neq 0$ (b)	K $x \neq 0$ (c)	K $x \neq 0$ (d)	K $x \neq 0$ (e)
8	K GTO(0)	K GTO(1)	K GTO(2)	K GTO(3)	K GTO(4)	K GTO(5)	K GTO(6)	K GTO(7)	K GTO(8)	K GTO(9)	K GTO(a)	K GTO(b)	K GTO(c)	K GTO(d)	K GTO(e)
9	K $x \geq 0$ (0)	K $x \geq 0$ (1)	K $x \geq 0$ (2)	K $x \geq 0$ (3)	K $x \geq 0$ (4)	K $x \geq 0$ (5)	K $x \geq 0$ (6)	K $x \geq 0$ (7)	K $x \geq 0$ (8)	K $x \geq 0$ (9)	K $x \geq 0$ (a)	K $x \geq 0$ (b)	K $x \geq 0$ (c)	K $x \geq 0$ (d)	K $x \geq 0$ (e)
A	K GTO(0)	K GTO(1)	K GTO(2)	K GTO(3)	K GTO(4)	K GTO(5)	K GTO(6)	K GTO(7)	K GTO(8)	K GTO(9)	K GTO(a)	K GTO(b)	K GTO(c)	K GTO(d)	K GTO(e)
B	K STO(0)	K STO(1)	K STO(2)	K STO(3)	K STO(4)	K STO(5)	K STO(6)	K STO(7)	K STO(8)	K STO(9)	K STO(a)	K STO(b)	K STO(c)	K STO(d)	K STO(e)
C	K $X < 0$ (0)	K $X < 0$ (1)	K $X < 0$ (2)	K $X < 0$ (3)	K $X < 0$ (4)	K $X < 0$ (5)	K $X < 0$ (6)	K $X < 0$ (7)	K $X < 0$ (8)	K $X < 0$ (9)	K $X < 0$ (a)	K $X < 0$ (b)	K $X < 0$ (c)	K $X < 0$ (d)	K $X < 0$ (e)
D	K RCL(0)	K RCL(1)	K RCL(2)	K RCL(3)	K RCL(4)	K RCL(5)	K RCL(6)	K RCL(7)	K RCL(8)	K RCL(9)	K RCL(a)	K RCL(b)	K RCL(c)	K RCL(d)	K RCL(e)
E	K $X = 0$ (0)	K $X = 0$ (1)	K $X = 0$ (2)	K $X = 0$ (3)	K $X = 0$ (4)	K $X = 0$ (5)	K $X = 0$ (6)	K $X = 0$ (7)	K $X = 0$ (8)	K $X = 0$ (9)	K $X = 0$ (a)	K $X = 0$ (b)	K $X = 0$ (c)	K $X = 0$ (d)	K $X = 0$ (e)

17. Appendix 3 - Program Examples

Calculating the area of a circle

$$S = \frac{\pi \times d^2}{4}$$

This program assumes that a given value of d is stored in Reg. RG2.

Address	Keys	Codes	Comments
01		62	Load stack register X with the d value from RG2
02		22	Calculation of d^2
03		0E	Copy d^2 into stack register Y
04		20	Load the π constant
05		12	Calculation of $\pi \times d^2$
06		04	Input number 4 into stack register X
07		13	Compute the area S
08		50	Stop the program

The command R/S is required to stop the program and display the result; otherwise the MK will execute all 105 steps of the program.

To calculate S for a particular value of $d = 4$, it is necessary to perform the following operations.

Specific procedures to this program:

- Store the number 4 in Reg. RG2: ;

Generic procedures to prepare any program to run:

- While in "RUN" mode, go to address 00 ;
- Enter "PROGRAM" mode and enter the program;
- Go back to "RUN" mode ;
- Reset the program counter to 0 by pressing RTN/0 (B/O) and
- Run the program with the R/S (C/N) key.

Expected result on display: 12.56637.

Using unconditional indirect branching GTO (i) (K, БП)

Evaluate the expression: $(4 + 2) \times 3$

Address	Keys	Codes
01	4	04
02	БП	0E
03	K БП 3	83
04	F √	21
05	2	02
06	+	10
07	K БП 4	84
08	+	10
09	3	03
10	×	12
11	K БП a	8-
12	-	11
13	C/П	50

Store the numbers 5, 7 and 12 in the registers RG3, RG4, e RGa, respectively. Enter the program; then go back to “RUN” mode and reset the program counter with RTN/0 (B/0).

During execution of the instruction GTO(3) (K, БП, 3) at address 03, because RG3 contains the value 5, the program jumps to that address 05. The Reg. RG3 is decremented by 1 *after evaluating the branch address* ²⁵.

For the next indirect branch, we have number 7 in Reg. RG4 which is increased by 1 *before evaluating the branch address*, so the program instruction GTO(4) (K, БП, 4) at address 07 will jump to the address 08.

At last, Reg. RGa will not change the value, so the program instruction GTO(a) (K, БП, a) at the address 11 will jump to address 13, *one address higher than specified in RGa*.

²⁵ This MK behavior is bizarre. This example works fine and it is presented here as written in the original Russian guide. However their explanation seems to be incorrect:

- 1) Firstly it lists the instruction addresses from 00, when it is obvious that it starts at 01.
- 2) And then it states that the pointer register RG3 is decremented before evaluating the jump address.

As far as we can tell, the behavior for branching instructions is:

- Indirect addresses for RG0-3, the value is decremented **after** evaluating the branch address;
- Indirect addresses for RG4-6, the value is incremented **before** evaluating the branch address;
- Indirect addresses for RG7-e, the value is maintained but the branch address is incremented by 1.
- Conditional and unconditional direct branch, the address value is incremented by one.

The above arithmetic expression should give a result of 18.

Using conditional indirect branching $x < 0$ (i) (K, $x < 0$)

Evaluate the expression: $y = -9x^2 + e^{2x}$

This program assumes that the x value is stored in RG1.

Depending on the expression result, if $y < 0$ then add the value of $\tan x$ to the result, else add the value of $\sin x$.

Address	Keys	Codes
01		01
02		09
03		4L
04		09
05		61
06		22
07		12
08		42
09		61
10		02
11		12
12		16
13		62
14		11
15		CL
16		61
17		1E
18		10
19		50
20		61
21		1C
22		10
23		50

Address 01 - 03: Store the branch address for the command ($x < 0$) (i).

Address 04 - 14: Evaluation of the expression $= -9x^2 + e^{2x}$.

Address 15: Evaluate the result to check for conditional branching.

Address 16 - 17: Calculation of $\tan x$.

Address 20 - 21: Calculation of $\sin x$.

If $y > 0$, then after the command at 15 the program jumps to address at 20, *one address higher than specified in the pointer register RGb*.²⁶

If < 0 , then after command at 15 it executes the next instruction at address 16.

For example, to calculate the value of y for a particular value of $x = 4$, it is necessary to perform the following steps:

Go to "PROGRAM" mode at address 00 and enter the program;

Go back to "RUN";

Set the angle switch  **RAD / GRD / DEG (P / ГРД / Г)** to **DEG (Г)**;

Store the x value = 4 to RG1;

Reset the Program Counter by pressing RTN/0 (**B/0**);

Expected result: 2836.2007.

²⁶ This example works fine and it is the original one as written in the original Russian guide. However their explanation seems to be incorrect:

- 1) Firstly it lists the instruction addresses from 00, when it is obvious that it starts at 01.
- 2) And then it states that the conditional indirect jump takes the pointer register RGb address as it is (19 in this case, when in reality it should be 20, as the first valid address is 01, not 00).

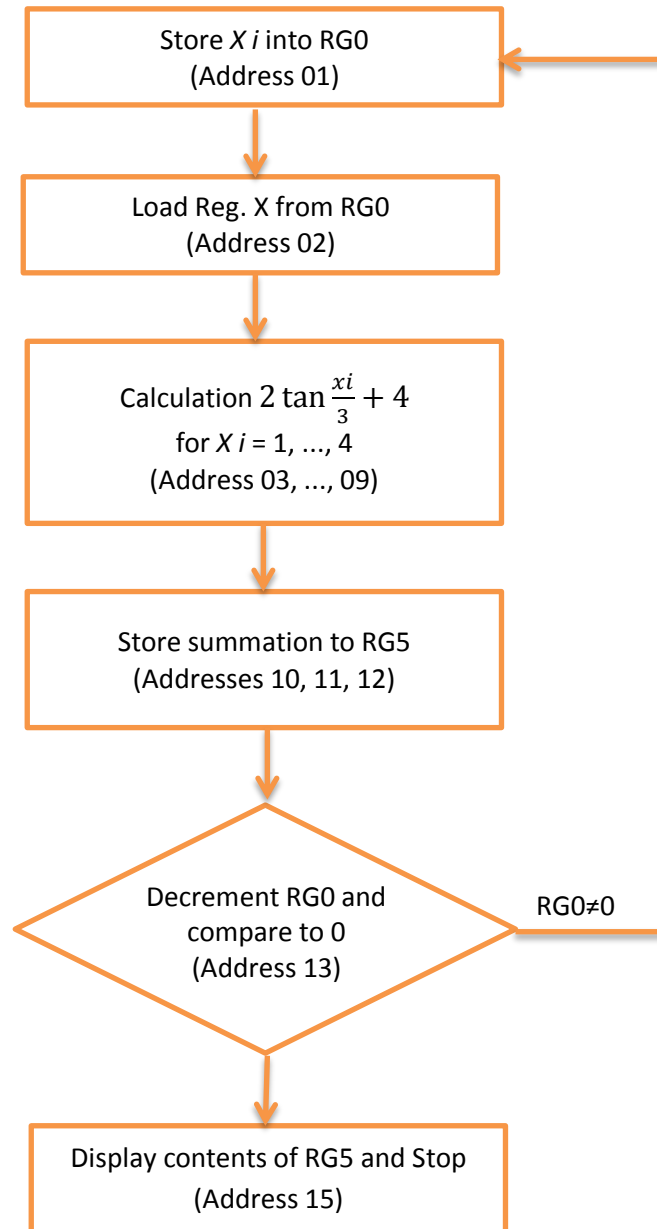
As far as we can tell, the behavior for branching instructions is:

- Indirect addresses for RG0-3, the value is decremented **after** evaluating the branch address;
- Indirect addresses for RG4-6, the value is incremented **before** evaluating the branch address;
- Indirect addresses for RG7-e, the value is maintained but the branch address is incremented by 1.
- Conditional and unconditional direct branch, the address value is incremented by one.























Looping calculations using commands such as DSNZ0 (F, L0)

Evaluate the expression: $\sum_{i=1}^4 (2 \tan \frac{xi}{3} + 4)$

Program flowchart:



Program:

Address	Keys	Codes
01	 	40
02	 	60
03		03
04		13
05	 	1E
06		02
07		12
08		04
09		10
10	  ²⁷	65
11		10
12	 	45
13	 	5Γ
14	  ²⁸	02
15		50

²⁷ Because the program does not initialize the Summation register, we need to do it manually (Store 0 into RG5), or else we get erroneous results when running this program more than once.































²⁸ The original Russian manual lists the addresses starting from 00, which is not correct, because the first usable address to store instruction codes is 01.

So this example will loop forever if entered as indicated in the original Russian guide (by specifying DSNZO 01). They mean the 2nd address when counting from address 00, so the correct address is 02 when counting from 01.

The correct branch address to use is DSNZO 02 as listed above.

Looping calculations using indirect load RCL (i) (K, Π→X)

Evaluate the expression: $\sum_{i=1}^4 (2 \tan \frac{xi}{3} + 4$

Address	Keys	Codes	
01	 	43	Store X i into RG0
02	  	Γ3	Indirect load X from a Register pointed to by RG3. The only purpose is to decrement the RG3.
03	 	63	Load X with next x i value to evaluate.
04	 	5E	Test for the end of the loop.
05	 	07	If X is not Zero, jump to 08 ²⁹
06	 	65	Else, show the Summation
07		50	And stop
08		03	Evaluating $2 \tan \frac{xi}{3} + 4$
09		13	...
10	 	1E	...
11		02	...
12		12	...
13		04	...
14		10	$2 \tan \frac{xi}{3} + 4$ is evaluated
15	 	65	Load X with previous Summation
16		10	Evaluate the new Summation
17	 	45	Store Summation in RG5
18		51	Jump to...
19	 	01	Address 02 ³⁰

Before running the program: Set angles to DEG (Γ); type $xi=4$ into X; Clear RG5³¹; Reset with RTN/0 (B/0).

²⁹ The Branch addresses are automatically incremented by one, meaning that we need to specify one address less than the Target one.

As far as we can tell, the behavior for branching instructions is:

- Indirect addresses for RG0-3, the value is decremented **after** evaluating the branch address;
- Indirect addresses for RG4-6, the value is incremented **before** evaluating the branch address;
- Indirect addresses for RG7-e, the value is maintained but the branch address is incremented by 1.
- Conditional and unconditional direct branch, the address value is incremented by one.

³⁰ See previous footnote above.

³¹ Because the program omits the Summation register initialization.

18. Appendix 4 - Warranty Card

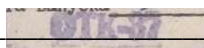


WARRANTY CARD

Micro Calculator "Elektronika MK-61"

Serial Number. _____ Date of Manufacture _____

Quality Department stamp _____



Address for claims:

283830, г.Збараж, Тернопольской обл.,з-д "Квантор"

ул.Галицкого,56

To be filled by the trade company

Date of sale _____ Seller _____
Format: d-m-y Signature or stamp

Store stamp _____

Included warranty service _____
Name

Repair shop, date, month, year

Warranty Number _____

19. Appendix 5 - Tear-off Vouchers



Expires by using it

(front side)

TEAR-OFF VOUCHER

For warranty repairs during the
warranty period

Micro Calculator "Elektronika MK-61"

Serial Number. _____ Date of Manufacture _____

Quality Department stamp _____



Address for claims:

283830, г.Збараж, Тернопольской обл.,з-д "Квантор"

ул.Галицкого,56

To be filled by the trade company

Date of sale _____ Seller _____

Format: d-m-y

Signature or stamp

Store stamp _____

(Back side)

To be filled the repair shop

Warranty Product number _____

Repair report. Name and reference number of the replaced part or assembly
in the schematic diagram. Location and nature of defects:

Repair date _____

Format: d-m-y

Signature of the person who performed the repairs _____

Signature of the owner, confirming the repair

Repair shop stamp _____

Specifying the city



Expires by using it

(front side)

TEAR-OFF VOUCHER

For warranty repairs during the
warranty period

Micro Calculator "Elektronika MK-61"

Serial Number. _____ Date of Manufacture _____

Quality Department stamp _____

ОТК-87

Address for claims:

283830, г.Збараж, Тернопольской обл.,з-д "Квантор"

ул.Галицкого,56

To be filled by the trade company

Date of sale _____ Seller _____

Format: d-m-y

Signature or stamp

Store stamp _____

(Back side)

To be filled the repair shop

Warranty Product number _____

Repair report. Name and reference number of the replaced part or assembly in the schematic diagram. Location and nature of defects:

Repair date _____

Format: d-m-y

Signature of the person who performed the repairs _____

Signature of the owner, confirming the repair

Repair shop stamp _____

Specifying the city

Circulation 30,000 Orders 2220

Printed by " UKRTIPPROEKT". Eugene Pottier Str., Kiev-57, 252057 Ukraine.

