



Soviet-era science, translated into English

CYBERNETICS AND CONTROL THEORY

G. M. Adelson-Velsky, A. L. Brudno, A. S. Kronrod,

1964

SovietRxiv

View the original and related papers at <https://sovietrxiv.org/items/ru-196401.44522>

Source: Math-Net.Ru and CyberLeninka. Machine translation. Verify with the original.

Abstract

Full Text

CYBERNETICS AND CONTROL THEORY

G. M. Adelson-Velsky, A. L. Brudno, A. S. Kronrod,
P. T. Reznikovskiy

ON AN INSTRUCTION SYSTEM FOR A THREE-ADDRESS MACHINE WITHOUT AN ADDRESS REGISTER

(Presented by Academician M. V. Keldysh on 27 VII 1963)

1°. The exposition is for a machine with 4096 memory cells containing 43-bit words. Machine-distinguished cells are: S , the standard end Ω , and the unforgettable zero in the zero cell.

2°. Bits in words are numbered from right to left; A_k denotes the k -th bit of the word A . Words may be interpreted:

- a) as instructions ($A_{43} \div A_{37}$ —code, $A_{36} \div A_{25}$ —address I, $A_{24} \div A_{13}$ —address II, $A_{12} \div A_1$ —address III).
- b) as numbers in normal form (A_{43} —sign, $A_{42} \div A_{37}$ —order $+100_8$, further —mantissa).
- c) as positive integer 43-bit numbers.

3°. Everywhere the following notation is used: KA is the code of the word A ; AdA is the address part of A (from A_1 to A_{36}). The I-III addresses of the word A are denoted $A1, A2, A3$. In the case of possible ambiguity, the 12-bit number constituting As is denoted by “ As .” The address of the cell containing the word A is denoted by $A0$. Everywhere $\alpha, \beta, \gamma, \delta$, and ε denote, respectively, 1, 2, 3, 6, and 7 bits taken consecutively. The cell containing the instruction under consideration is denoted by \cdot , and the one following it by $\cdot + 1$.

4°. The instruction system is given in the table. In the text, explanations are given only for instructions that are not clear from the table. Work with external devices and block memory is not considered.

5°. Explanations to the code table

The **immediate-execution instruction** A, B, C (001) executes the instruction from the cell $A3 + \langle B \rangle$. Before this, “ C ” enters the address register of the instruction.

$Prf[A], B = C$ (002) serves to form the addresses of the cell C by algebraic addition of the addresses (and the code) B . Let

$$A = \alpha_0 \beta_0 \alpha_1 \beta_1 \alpha_2 \beta_2 \alpha_3 \beta_3.$$

Then “ Bs ” $\times (-1)^{\alpha_s}$ is an addend for $C\beta_s$ ($s = 1, 2, 3$). If $\beta_s = 0$, the number “ Bs ” is not used. $KC = KB$ when $\alpha_0 = \beta_0 = 0$, and $KC = 0$ in all other cases. If $\beta_0 \neq 0$, the quantity $KB \times (-1)^{\alpha_0}$ is an addend for $C\beta_0$.

The instruction $\Phi(A, B, C)$ (003), executed immediately after Inf ABC (103), makes it possible to refer in one instruction to a function of one variable with specification of the addresses of the argument and of the result.

The **address-formation instruction** (004). Let

$$\begin{aligned} & : \Phi A, \quad A, \quad B, \quad C, \\ +1 & : \quad K, \quad P, \quad Q, \quad R. \end{aligned}$$

The instruction $(K, P + \langle A3 \rangle, Q + \langle B3 \rangle, R + \langle C3 \rangle)$ will be executed, with the address register of the instruction containing not γ , but $\gamma + 1$.

The **instruction-formation instruction** (005). Let

$$\begin{aligned} & : \Phi K \quad \alpha'_1 \alpha'_2 \alpha'_3 \alpha_1 \beta_1 \alpha_2 \beta_2 \alpha_3 \beta_3 \quad M \quad T, \\ +1 & : \quad K \quad L_1 \quad L_2 \quad L_3. \end{aligned}$$

If $T \neq 0$, the instruction

$$KT = K, \quad Ts = Ms \cdot \alpha'_s + (-1)^{\alpha_s} \cdot Ls \gamma_s$$

will enter the cell T .

If $T = 0$, then the instruction formed in the indicated way will not be written into memory, but will be executed immediately, as one located in $\gamma + 1$.

Example.

$$\begin{aligned} & : \Phi K \ 101 \ 13 \ 02 \ 00 \ S \ T \\ +1 & : \quad \times \quad A \quad B \ C \end{aligned}$$

Group designation	0	...1	...2	...3	...4	...5	...6	...7
00. Interpret- ing and service com- mands	2- address com- mands -un- condi- tional trans- fer of con- trol	NI A, B, C (see $(0, 0, C)$ text)	[[unclear: com- mand name]] $(A)B \rightarrow$ Areforming (see text)	$\Phi(A, B, C)$ $A(B - S2; S2) -$ C	of ad- resses text)	of operands text) com- mands	of opera- tion condi- tional	Work (see exter- nal de- vices
10. Un- condi- tional trans- fer of con- trol C	Stop A, B, C and B punched tape	Command write work- ing with long mem- ory	Multiple com- mand	MNP $(A, B, C) \rightarrow$ $S[(0, 0, 0; q^{n+1}) -$ $1] \rightarrow$ Ω	$A \rightarrow$ $B + \epsilon$	$A \rightarrow$ $B :$ $\Omega[(0, 0, 0; q^{n+1}) -$ $1] \rightarrow$ Ω	$A^* \rightarrow$ $B, {}^3C(0, 0, 0, \dots, A^*) \rightarrow$ $-B(0, 0, 0, q^{n+1}) \rightarrow$ Ω	
01. Addi- tion action	$A +$ $B =$ C	$A -$ $B =$ C	$A \times$ $B =$ C	$A :$ $B =$ C	$A +$ ${}_n B^* =$ C	$A -$ ${}_n B^* =$ C	$A \times$ ${}_n B^* =$ C	$A :$ ${}_n B^* =$ C
11. Addi- tion ac- tion, result in S ; condi- tional trans- fer of con- trol C	$A +$ $B :$ C $S < 0$	$A -$ $B :$ C $S < 0$	$A \times$ $B :$ C over- flow	$A :$ $B :$ C over- flow	$A +$ ${}_n B^* :$ C $S < 0$	$A -$ ${}_n B^* :$ C $S < 0$	$A \times$ ${}_n B^* :$ C over- flow	$ A -$ $ B :$ C $S < 0$

Group designation	0	...1	...2	...3	...4	...5	...6	...7
02. Complex tabular action	$(A \times B) + C = C$	$(A \times B) + C = C$	$(A \times B) + C = C$	$(A : B) + C = C$	$(A : B) + C = C$	$(A : B) + C = C$	$(A \times B) \times C = C$	$(A : B) \times C = C$
12. Complex tabular action, result entered in S	$(A \times B) + S$	$(A \times B) + S$	$(A \times B) + S$	$(A : B) + S$	$(A : B) + S$	$(A : B) + S$	$(A \times B) \times S$	$(A : B) \times S$
03. Actions with whole cycles	$A + B = C$	$A - B = C$	$A \times B = C_{\text{tail}}$ in S C_{head} in S	$A(\text{mod } B) = C_{\text{quotient}}$ in S	$A + {}_n B^* = C$	$A - {}_n B^* = C$	$A^* \cdot B = C_{\text{tail}}$ in S C_{head} in S	$A(\text{mod } B^*) = C_{\text{quotient}}$ in S

Group designation	0	...1	...2	...3	...4	...5	...6	...7	
13. Ac-tions with whole cycles, result entered in S ; conditional transfer of control C	$A +$ $B :$ C_{on} over-flow	$A -$ $B :$ C_{if} $S < 0$	$A \times$ $B :$ C_{tail} in S_{on} over-flow	$A(\text{mod } B) :$ C_{if} $S = 0$	$A +$ $_n B^* :$ C_{on} over-flow	$A -$ $_n B^* :$ C_{if} $S < 0$	$A \times$ $_n B^* :$ C_{tail} in S_{on} over-flow	$A(\text{mod } B^*) :$ C_{if} $S = 0$	

	Group name	.0	.1	2	.3	.4	.5	.6	.7
04	Logical operations	$A \vee$ $B =$ $C_{logical}$ addition	$A \wedge$ $B =$ $C_{logical}$ multiplication	$A \wedge$ $B =$ $C_{logical}$ multiplication	$A *$ $B =$ $C_{top}(C_i$ 1, if $A_i \neq \text{ext}$) $B_i)$	$[\delta] \wedge$ $B =$ $C_{tracking}(\text{word}$ shift by address(text))	$[f] \rightarrow$ $B =$ C_{word} shift by address(text)	$f \rightarrow$ $B =$ C_{shift} shift by address(text)	$f \rightarrow$ $B =$ C_{shift} of address part by address(text)

	Group name	.0	.1	2	.3	.4	.5	.6	.7
14	Logical operations; result sign in S , condition for storing result in C	$A \vee B :$ C provided $S = 0$	$A \wedge B :$ C provided $S = 0$	$A \wedge B :$ C provided $S = 0$	$A * B :$ C provided $S \neq 0$	$[\delta] \wedge B :$ C provided $S = 0$	$[f] \rightarrow B :$ C provided (see text) $S = 0$	$f_3 \rightarrow B :$ C provided $S = 0$	$f_3 \rightarrow B :$ C provided $S = 0$
05	Forward shift (see text)	$f_{-\log B} C$	$f_{-1, B''} C$	$f_{-2, B''} C$	$f_{-3, B''} C$	$f_{-13} B C$	$f_{-23} B C$	$f_{-33} B C$	$f_{-\text{adr} B} C =$
15	Forward shift; result sign in S , condition for storing result in C (see text)	$f_{-\log B} C$	$f_{-1, B''} C$	$f_{-2, B''} C$	$f_{-3, B''} C$	$f_{-13} B C$	$f_{-23} B C$	$f_{-33} B C$	$f_{-\text{adr} B} C :$ $\log S < S1 < S2 < S3 < \text{adr} S <$ $0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0$
06	Backward shift (see text)	$f_{+\log B} C$	$f_{+1, B''} C$	$f_{+2, B''} C$	$f_{+3, B''} C$	$f_{+13} B C$	$f_{+23} B C$	$f_{+33} B C$	$f_{+\text{adr} B} C =$

Group name	.0	.1	2	.3	.4	.5	.6	.7	
16	Conditional transfer of control to C with changing of the index	$f_3^+ \Rightarrow B'' : f_3^* \geq f_3''$	$f_3^+ \leq B'' : f_3^* < f_3''$	$f_3^- \Rightarrow B'' : f_3^* > f_3''$	$f_3^- < B'' : f_3^* \leq f_3''$	$f_3^+ \Rightarrow B_3'' : f_3^* \geq f_3''$	$f_3^+ < B_3'' : f_3^* < f_3''$	$f_3^- \Rightarrow B_3'' : f_3^* > f_3''$	$f_3^- < B_3'' : f_3^* \leq f_3''$
07	Arithmetic miscellaneous cellaneous	$ A + B = C$	$\text{sign } \sqrt{A+B}$	$\{A : B\} = C$	$A \oplus B = C$	$A \ominus B = C$	$A'' \dots B'$	command result for operation with cyclic transfer through point one digit	
17	Boolean miscellaneous cellaneous	$C[[A]]B$	$P3F[[A]]B$	$B^2 + f \leftarrow B^* + C3''$	Set the special function digit (see text)	$\text{sign } f[(\delta S) \times (A/B)]$	$\max A B = C$	$\min A, C$	

the command $(S1 - A3) \times B2 = (S3 + "C")$ will enter T .

Group operation (006). Let

$$\begin{array}{l}
 : O \quad \alpha_0 \beta_0 \alpha_1 \beta_1 \alpha_2 \beta_2 \alpha_3 \beta_3 \quad B \quad C \\
 + 1 : \quad \dots \quad \dots \quad \dots \quad \dots
 \end{array}$$

Then the command located in +1 is executed in a cycle "C" times if $\beta_0 = 0$, and $C\beta_0$ times if $\beta_0 \neq 0$. To the s-th execution address of +1, at each step there is added $(-1)^s \cdot "B\beta_s"$ if $\alpha_0 = 1, \beta_s \neq 0$, and $(-1)^s \cdot "B"$ if $\alpha_0 = 0, \beta_s \neq 0$. If $\beta_s = 0$, the s-th execution address of +1 is not changed. At the end of the cycle, control passes to +2, unless in the cycle it has been transferred to another command. The command +1 itself remains unchanged.

Cut-out $[\delta_1\delta_2] \wedge B = C$ (044). If $\delta_1 \leq \delta_2$, then $C_i = B_i$ for $\delta_1 \leq i \leq \delta_2$ and $C_i = 0$ for $i < \delta_1$ or $i > \delta_2$. If $\delta_1 > \delta_2$, then $C_i = B_i$ for $i \geq \delta_1$ or $i \leq \delta_2$, and $C_i = 0$ for $\delta_2 < i < \delta_1$.

Complex shift $[A] \Rightarrow B = C$ (045). Let $A = \alpha_{-1}\alpha_0\alpha_1\alpha_2\alpha_3\varepsilon$. Let $\tau = \varepsilon - 100_8$.

The shift is by $|\tau|$ digits to the left for $\tau \geq 0$ and to the right for $\tau < 0$. Shifted are: the code B if $\alpha_0 \neq 0$, and the addresses Bs for which $\alpha_s \neq 0$. Let M be the set of numbers of digits subject to shifting. Let $T_i = 1$ for $i \in M$ and $T_i = 0$ for $i \notin M$. Into C goes Q , the shifted part of B , intersected with T . If $\alpha_{-1} = 1$, then the unshifted part of B is also added to Q . Command 145 is executed analogously. Control is transferred to C when $Q = 0$.

Shifts by address (046, 146, 047, and 147). Let $\tau = "A3" - 100_8$. The shift is by $|\tau|$ digits to the left for $\tau \geq 0$ and to the right for $\tau < 0$. The commands $A_3 \rightarrow B = C$ and $A_3 \rightarrow B; C$ (046 and 146) shift the whole word B , while $A_3 \Rightarrow B = C$ and $A_3 \Rightarrow B; C$ (047 and 147) shift only the address part of B , with $KC = KB$.

Commands $A \pm_i "B"$ (051–053; 151–153 and 061–063) are executed as follows: $Ai \pm "B" = Ci$. The remaining addresses and the code of C coincide with the code and the corresponding addresses of A .

Commands $A \pm_{i3} B$ (054–056, 154–156 and 064–066) add (subtract) "B3" to Ai ; the remaining addresses and the code are preserved.

Assembly $[A], B = C$ (170) and **disassembly** $[A], B = C$ (171). Let $i_1 < i_2 < \dots < i_k$ be the numbers of all digits of A for which $A_{i_s} \neq 0$. Then in assembly $C_s = B_{i_s}$ for $s \leq k$ and $C_s = 0$ for $s > k$, while in disassembly $C_{i_s} = B_s$ ($1 \leq s \leq k$), and the remaining digits of C are equal to 0. **Selection of the most significant one.** St. ed. A, B, C (174). Let $\lambda = \max i$ ($A_i = 1$). Then $B_\lambda = 1$; $B_s = 0$ ($s \neq \lambda$), $KC = C1 = C2 = 0$ and $C3 = \lambda$.

Count $[\delta_1\delta_2] \wedge B = C$ (175). If $\delta_1 \leq \delta_2$, the number of digits $B_i = 1$ for $\delta_1 \leq i \leq \delta_2$ is placed in $C3$, and if $\delta_1 \geq \delta_2$, the number of digits $B_i = 1$ for $i \geq \delta_1$ and $i \leq \delta_2$. In both cases $KC = C1 = C2 = 0$.

6°. In addition to the designated cells 0, S , and Ω , it is useful to have also a "stopwatch" T —a counter cell in which the right 10–12 digits are electronic. To this counter, upon execution of each command (or, still better, of a standard cycle), 1 is added.

7°. It is important that numbers of cells be assigned to the command register, to several RAM units, and to the register containing in address III the address of the preceding command.

8°. When operations with normalized numbers are performed, an emergency stop must occur upon an attempt to operate on the number -0 (zero in the result of a floating-point operation must be issued as $+0$). This same number, as a command, is a stop. Provision should be made for memory layout -0 from the console.

Institute of Theoretical and Experimental Physics
of the State Committee for the Utilization of Atomic Energy
Institute of Electronic Control Machines
Central Scientific-Research Institute of Complex Automation

Received
8 VII 1963

Note: Figure translations are in progress. See original paper for figures.

Source: Math-Net.Ru and CyberLeninka. Machine translation. Verify with the original.