

E. g. for Vidicon kU_1 is usually about 0.7. The usual time lag in Vidicons amounts to about 40% on the completion of a single frame, corresponding to $1/\alpha \approx 1.45$.

The graph in Fig. 2 yields $I_s/I_w \ll 0.5$.

At the foregoing values of the parameters, the current flowing through the thin layer cannot exceed one half of that conveyed by the electron beam incident thereon.

The ratio of the current in the dark and that at illumination is obtained from Fig. 2, if the variations of RC at illumination are known.

The author expresses his indebtedness to Professor J. Groszkowski and Docent W. Barwicz for their discussions of the problems investigated.

INDUSTRIAL INSTITUTE OF ELECTRONICS, WARSAW
(PRZEMYSŁOWY INSTYTUT ELEKTRONIKI, WARSZAWA)

REFERENCES

- [1] A. Rose, *Maximum performance of photoconductors*, *Helv. Phys. Acta*, **30** (1957), 242.
- [2] R. W. Redington, *Maximum performance of high-resistivity photoconductors*, *J. Appl. Phys.*, **29** (1958), 189.
- [3] A. Rose, A. Lampert, *Gain-bandwidth product for photoconductors* *RCA Rev.*, **20** (1959), 57.
- [4] A. Fryszman, *Bull. Acad. Polon. Sci., Sér. sci. math. astr. et phys.*, **7** (1959), 733.
- [5] —, *Bull. Acad. Polon. Sci., Sér. sci. math., astr. et phys.*, **8** (1960), 99.

The Organization of a Digital Computer Based on the "—2" System

by

Z. PAWLAK

Presented by P. SZULKIN on March 8, 1960

This paper contains a description of the organization of the digital computer described in [1].

Simplifying the question it may be said that every digital computer is composed of *registers*, an *arithmometer* and the *connection* between them. The way of joint operation between the elements just mentioned determines the type of the organization of the computer. This organization is uniquely connected with the list of orders that can be performed by the machine. If we construct a computer we have in general a given list of orders and we seek for an organization realizing the list required. In the computer in question the procedure is just the inverse: a certain organization of the computer is assumed beforehand enabling the realization of a possibly high number of orders for the purpose of investigating the elementary relations between the list (realizable in a given computer) of orders and the structure of the computer*).

From the point of view of the organization the list of orders of the machine depends on three factors:

- a) the topological structure of the computer (the type of connections between the registers and the arithmometer),
- b) the decoding network,
- c) the operation cycle of the machine.

In the organization presented here, it is assumed that every connection in the computer is controlled independently by means of one position of the control register s . With a fixed topologic structure — this principle gives a maximum number of possible orders. Of course, not all of the orders thus obtained are interesting and, conversely, not all of the

*) It seems that in the first case we can speak of a *synthetic* method (a synthesis) in the second — of an *analytic* method (an analysis) of the organization of the computer.

interesting orders can be obtained in this way. Nevertheless, a systematic study of this problem would be of use. In the present paper I shall confine myself to the description of more interesting orders of the computer just described.

Notations. The layout of the computer is assumed to be that of [1] with the addition of the multiplier register m . The registers are denoted as follows:

- x — any memory register,
- a — accumulator register,
- r — order register (or multiplicand register),
- l — order-counter register,
- d — input-output register,
- m — multiplier register,
- s — control register.

The letters x, a, r, l, d, m, s denote also numbers in the corresponding registers.

Φ denotes one of the letters x, r, l, d, m or the number in the corresponding register.

Ψ denotes a subset of the set of letters $\{r, l, d, a, m\}$. The remaining notations are as in [1].

The first degree orders

An arbitrary order of the first degree will be denoted by P . Orders of the first degree will be defined thus

I. Basic orders $a \in P$, if a can be obtained from the scheme

$$(k \cdot \Phi + p \cdot a) \rightarrow \Psi,$$

where

$$k = 0, 1, -1, \\ p = 0, 1, -1, -1/2, -2,$$

(The case of $k = p = 0$ is excluded).

For example,

$$(x+a) \rightarrow a, \\ (-r+a) \rightarrow a, l, \\ (m-2a) \rightarrow r, \\ m \rightarrow r, l, \\ a \rightarrow m.$$

We shall omit, for brevity, the arrow. Then, the above orders take the form:

$$(x+a)a, \\ (-r+a)al, \\ (m-2a)r, \\ mrl, \\ am.$$

More interesting basic orders are collected in Table I.

II. The successor

$$a \in P \supset Na \in P.$$

For example,

$$Nra, \\ N(r+a)a, \\ N(m-a)rl.$$

*) The basic orders have been systematized by T. Kulikowski.

III. Sign and overflow examination orders $a \in P \supset Sga \in P$ and $Nda \in P$.

For example,

$$Sg(x-a)a, \\ Sgxa, \\ Nd(x+a)a.$$

IV. Orders concerning the address part

$$a \in P \supset \#a \in P.$$

For example,

$$\#(x-2a)a, \\ \#Sg(x-a)a,$$

V. Conditional orders

$$a \in P \supset a? \in P.$$

For example,

$$Nra?, \\ N(x+r)l?.$$

VI. Stop orders

$$a \in P \supset a! \in P.$$

For example,

$$\#Nlr?!, \\ \#Sg(x+a)r!$$

VII. The transfer order

$$ax \in P.$$

VIII. The logical product

$$(\Phi \cap a)\Psi \in P.$$

For example,

$$(m \cap a)l, \\ (r \cap a)a.$$

Discussion of some orders

Jump orders. In the organization type assumed there is a considerable number of jump orders. Some are given below. The basic jump orders are shown in Table I, columns E, F, G. These orders will be called

E — indirect jumps, F — direct jumps, G — relative jumps.

TABLE I

	A	B	C	D	E	F	G	H
1.	0a	—	—	0m	0l	—	—	0r
2.	—	aL	aP	am	al	—	—	ar
3.	-aa	-aL	-aP	-am	-al	—	—	-ar
4.	xa	—	—	xm	xl	rl	—	xr
5.	-xa	—	—	-xm	-xl	-rl	-ll	-xr
6.	(x+a)a	(x+aL)a	(x+aP)a	(x+a)m	(x+a)l	(r+a)l	(l+a)l	(x+a)r
7.	(x-a)a	(x-aL)a	(x-aP)a	(x-a)m	(x-a)l	(r-a)l	(l-a)l	(x-a)r
8.	(-x+a)a	(-x+aL)a	(-x+aP)a	(-x+a)m	(-x+a)l	(-r+a)l	(-l+a)l	(-x+a)r
9.	(x∩a)a	(x∩aL)a	(x∩aP)a	(x∩a)m	(x∩a)l	(r∩a)l	(l∩a)l	(x∩a)r
	I	J	K	L	M	N	O	P
1.	—	—	0d	—	—	—	—	—
2.	—	—	ad	—	—	—	—	—
3.	—	—	-ad	—	—	—	—	—
4.	rr	ra	xd	rd	da	dr	dl	—
5.	-rr	-ra	-xd	-rd	-da	-dr	-dl	—
6.	(r+a)r	(r+a)a	(x+a)d	(r+a)d	(d+a)a	(d+a)r	(d+a)l	—
7.	(r-a)r	(r-a)a	(x-a)d	(r-a)d	(d-a)a	(d-a)r	(d-a)l	—
8.	(-r+a)r	(-r+a)a	(-x+a)d	(-r+a)d	(-d+a)a	(-d+a)r	(-d+a)l	—
9.	(r∩a)r	(r∩a)a	(x∩a)d	(r∩a)d	(d∩a)a	(d∩a)r	(d∩a)l	—

An indirect jump, e. g. x_l enables the transfer to the order, the address of which is given under that of x . A direct jump order, for instance $\#rl$ causes the transfer to the order, the address of which is given in r . A relative jump, e. g. $\#(l+a),l$ causes the omission of a orders in the program. Conditional jumps are obtained by adding to any jump order the interrogation mark?, for instance $\#rl?$

b) Tape orders. These orders are given in columns M and N (Table I). The orders of the group M cause a modification of the order introduced; the orders of the group N serve the purpose of interpreting the orders, if the interpretation technique is used.

The modification of the orders during their bringing into the computer proceeds as follows. Let ρ be an order punched on the input tape and let C be one of the 32 possible code letters punched on the tape directly after the order ρ .

The preliminary program causes introduction of the order ρ into the accumulator a and that of the code letter C — into the order register r . Since the input-output register d has only five positions, the operational part in the register r is equal to zero; this corresponds to the order xr (the taking of the order from the memory). The address of this order is the numerical value of the code letter C . Under address C there is the order $\#(x+a)a$. This order causes a modification of the address part of the order ρ contained in the accumulator. After modification the order ρ is sent to the memory.

The interpretation of an order proceeds similarly to the modification process except that the code letter, which is interpreted as a symbol of the operation of an external order — is sent not to r but to l . This causes the selection of the proper subroutine for interpreting the order introduced.

Second degree orders

Any order of the second degree will be denoted by the letter D .

I. Basic orders

The basic orders of the second degree are given in Table II.

II. Conditional orders,

$$a \in D \supset a? \in D.$$

For example,

$$(x+a)?.$$

In second degree orders the head of the result is always contained in the accumulator a , the tail being contained in the register m .

The meaning of the remaining orders is obvious. The linkage of sub-routines is conventional.

III. Stop orders

$$a \in D \supset a! \in D.$$

For example,

$$(x+a)!.$$

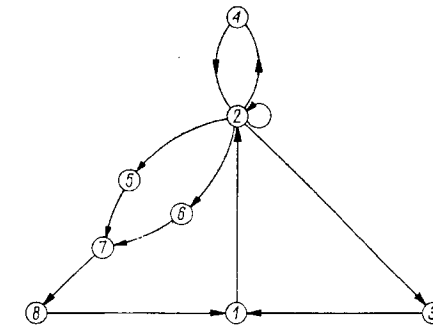
Besides the orders of the first and second degree the computer realizes also composite orders causing the performance of several (usually two) simultaneous orders of first degree. In view of their complexity these orders will be described in a subsequent paper.

TABLE II

	Multiplication	Square
1.	—	—
2.	—	a^2
3.	$-a \cdot a$	$(-a)^2$
4.	$x \cdot a$	x^2
5.	$-x \cdot a$	$(-x)^2$
6.	$(x+a) \cdot a$	$(x+a)^2$
7.	$(x-a) \cdot a$	$(x-a)^2$
8.	$(-x+a) \cdot a$	$(-x+a)^2$
9.	—	—

The operation cycle

A simplified scheme of the operation cycle of the computer is shown in Fig. 1. The circles denote successive states of the control register s . The arrows denote the sequence of the change of states. For facility, all the states are numbered from 1 to 8.



The meaning of the numbers is as follows:

- 1 — the order Nlr — the taking of the address of the order in the order register and the increase of the number in the orders counter by one;
- 2 — the order xr — the taking of the order to the order register;
- 3 — any non-control order of the first degree;
- 4 — any control order *;

* We call a control order every order causing an introduction of new contents into the order register r .

5 — the order xr — the introduction of the multiplicand from the memory into the multiplicand register (the register r plays the role of an order register and a multiplicand register),

6 — the order xam — the introduction of the multiplier and the multiplicand from the memory into the accumulator and a multiplicand register r ,

7 — the order am — the introduction of the multiplier from the accumulator a into the multiplier register m ,

8 — the multiplication of the number in the register m by the number in the register r .

State 1 is the initial state. From state 1 the register can pass only to state 2. Starting from state 2, the register s can, depending on the content of the memory, remain in state 2 (repeated taking of the order) or pass to one of the states 3, 4, 5, 6. If the register s is in state 3, the order performed is one of the non-control orders of the first degree such as $(x+a)a$. From state 3 s can pass only to state 1 (the taking of the address of the next order). In the state 4 any control order is taken, for instance $(x+a)r$. After state 4 anyone of the states 2, 3, 4, 5, 6 can follow, similarly to state 2 (For simplicity the figure shows only the passage to state 2). State 2 may be followed by 5 or 6. These states correspond to orders of the second degree; state 5 — to the multiplication, state 6 to the raising to a power. In both states the multiplicand is introduced into the register r . In state 7 the multiplier is introduced into the register m ; in state 8 the multiplication is performed.

Thus, any (non-control) order of the first degree corresponds to the cycle 1, 2, 3, 1; a multiplication — to 1, 2, 5, 7, 8, 1; a raising to a power — to 1, 2, 6, 7, 8, 1.

A conditional order is not performed, if the condition is satisfied. The stop order causes the stopping of the computer before the order is performed.

There exists the possibility of performing orders step-by-step, and controlling the computer from outside.

INSTITUTE OF MATHEMATICS, POLISH ACADEMY OF SCIENCES
(INSTYTUT MATEMATYCZNY, PAN)
DEPARTMENT OF TELE- AND RADIOFONIC-CONSTRUCTION, WARSAW TECHNICAL
UNIVERSITY
(ZAKŁAD KONSTRUKCJI TELE- I RADIOFONII POLITECHNIKI WARSZAWSKIEJ)

REFERENCES

- [1] Z. Pawlak, *An electronic digital computer based on the "—2" system*, Bull. Acad. Polon. Sci., Sér. sci. techn., 7 (1959), 713.

BULLETIN DE L'ACADÉMIE POLONAISE DES SCIENCES

Rédacteur en chef
K. KURATOWSKI

Rédacteur en chef suppléant
L. INFELD

SÉRIE DES SCIENCES TECHNIQUES

Rédacteur de la Série
J. GROSZKOWSKI

Comité de Rédaction de la Série
C. KANAFOJSKI, W. NOWACKI, W. OLSZAK, B. STEFANOWSKI,
P. SZULKIN, W. SZYMANOWSKI

VOLUME VIII
NUMÉRO 6



VARSOVIE 1960