



---

# **Od GAM'a do ZAM'a Niewykorzystana szansa polskiej informatyki? Część II - Architektura**

---

**Opracowanie: Krzysztof Bytnerowicz**

Sydney, maj 2020



---

## Zeszyty Historyczne PTI

Zeszyt nr 5 - Od GAM-a do ZAM-a. Zaprzepaszczona szansa polskiej Informatyki?

CZĘŚĆ II - Architektura

Copyright © 2020 Polskie Towarzystwo Informatyczne

Copyright © 2020 Krzysztof Bytnerowicz

ISBN 978-83-65750-05-1 (kolekcja, części I-III; wydania elektroniczne)

ISBN 978-83-65750-14-3 (część II - Architektura; wydanie elektroniczne)



Opracowanie:  
Krzysztof Bytnerowicz

Redakcja techniczna:  
Krzysztof Bytnerowicz, Jerzy S. Nowak

Projekty graficzne:  
Krzysztof Bytnerowicz

Wydawca:  
Polskie Towarzystwo Informatyczne  
Zarząd Główny  
ul. Solec 38/103, 00-394 Warszawa

Wydanie I: maj 2020 r.

# Zeszyty Historyczne PTI\_ (N° 5)

## Od GAM-a do ZAM-a

Zaprzepaszczona szansa polskiej informatyki?

### CZĘŚĆ II – ARCHITEKTURA

red. Krzysztof Bytnerowicz



Sydney, Australia  
2020



## Spis Treści

EMAL.....	5
EMAL 2.....	8
P-1.....	10
GAM-1.....	11
XYZ.....	12
Budowa:.....	13
Dane techniczne.....	13
Języki programowania.....	14
Zespół.....	14
Krótki opis programowanej maszyny cyfrowej XYZ – I.....	15
Ogólna charakterystyka maszyny XYZ.....	15
Słowa, liczby i rozkazy.....	17
Adresowanie słów.....	18
Części składowe maszyny.....	20
Przesyłanie informacji w maszynie.....	23
Cykl pracy maszyny.....	23
Zestawienie rozkazów XYZ.....	25
Opis rozkazów XYZ.....	26
Bliższe dane dotyczące wejścia i wyjścia.....	28
Stolik operatora.....	30
ZAM-2.....	32
Zmiany w stosunku do XYZ.....	32
Dane techniczne.....	32
Języki programowania.....	34
Produkcja.....	34
Eksploatacja.....	34
Opis maszyny ZAM-2.....	35
Pamięć, słowo, liczba, rozkaz.....	36
Budowa liczby.....	38
Budowa rozkazu.....	38
Modyfikacja rozkazów.....	39
Arytmometr i Sterowanie.....	40
Akumulator, symbol A.....	40
Mnożnik, symbol M.....	41
Rejestr nadmiaru, symbol N.....	43
Rejestry sterowania.....	43
Rejestr modyfikacji, symbol B.....	43
Rejestr rozkazów, symbol R.....	44
Licznik rozkazów, symbol LR.....	44
Rejestr pomocniczy, symbol RP.....	44
Arytmetyka Maszyny.....	45
Dodawanie, odejmowanie, odejmowanie bezwzględnych wartości.....	45
Mnożenie.....	46
Dzielenie.....	46
Przesunięcia arytmetyczne.....	47
Przesunięcia cykliczne.....	47
Zaokrąglenie.....	47
Koniunkcja.....	48

Alternatywa.....	48
Właściwości zera.....	48
Arytmetyka rejestru B.....	49
Bęben Magnetyczny.....	50
Urządzenia Wejścia i Wyjścia.....	52
Lista Rozkazów Maszyny Zam-2.....	54
Rozkazy Dotyczące Arytmometru.....	55
Grupa A.....	55
Rozkazy Dotyczące Rejestru B.....	62
Grupa B.....	62
Rozkazy Sterujące.....	64
Grupa S.....	64
Rozkazy Dotyczące Urządzeń Zewnętrznych.....	68
Rozkazy Specjalne.....	72
Konkretne problemy rozwiązane na maszynie ZAM-2.....	73
ZAM-3.....	74
RODZINA ZAM 11,21,31,41,52.....	75
Wstęp.....	79
Zastosowania maszyn rodziny ZAM.....	80
Typy maszyn ZAM.....	81
Modułowa konstrukcja maszyn rodziny ZAM.....	82
Jednolitość programowania w rodzinie maszyn ZAM.....	86
Rozwój ośrodka obliczeniowego w ramach jednego przedsiębiorstwa.....	87
Współpraca ośrodków obliczeniowych jednej organizacji.....	87
Część centralna maszyn rodziny ZAM.....	89
System Wejścia i Wyjścia.....	91
Standardowe Urządzenia Wejścia i Wyjścia.....	92
Podział czasu i wieloprogramowość w maszynach ZAM.....	93
Systemy programowania w maszynach ZAM.....	97
SAS - System Adresów Symbolicznych.....	98
MAKRO-SAS.....	98
ALGOL — standardowy język algorytmiczny dla problemów numerycznych.....	98
SAKO - polski język algorytmiczny dla problemów numerycznych.....	99
COBOL - międzynarodowy język dla przetwarzania danych.....	99
SO — System Operacyjny Maszyny.....	100
Zarys organizacji ogólnej maszyn ZAM.....	101
Rejestry i Wskaźniki.....	102
Obliczanie efektywnego argumentu rozkazu.....	104
Przerywanie Programu.....	105
Rozkazy Nielegalne.....	105
Rozkazy Programowane.....	106
Lista Rozkazów Maszyn ZAM.....	107
Rozkazy Sterujące.....	107
Rozkazy Programowane.....	107
Rozkazy rejestru B.....	107
Przesłania A oraz M.....	107
Działania Logiczne.....	107
Działania Stałoprzecinkowe A oraz M.....	107
Działania Zmiennoprzecinkowe.....	108
Działania na słowach w Pamięci.....	108
Zmiana zawartości rejestrów.....	108
Operacje Wejścia i Wyjścia.....	109

Opis techniczny maszyn matematycznych rodziny ZAM.....	110
Wstęp.....	111
Ogólna charakterystyka maszyn rodziny ZAM.....	111
Technika maszyn rodziny ZAM.....	113
Konstrukcja maszyn rodziny ZAM.....	115
Moduły maszyn rodziny ZAM.....	116
Pamięć ferrytowa PA05.....	116
Pamięć bębnowa PB5.....	117
Pamięć na taśmie magnetycznej PT2.....	117
Stolik operatora SO2.....	118
Moduł czytnika taśmy CP1.....	119
Moduł dziurkarki taśmy papierowej PD1.....	119
Moduł drukarki wierszowej DW1.....	119
Moduł czytnika kart CK2.....	120
Moduł monitora MD2.....	120
Moduł dziurkarki kart DK1.....	121
Moduł kanału automatyki KA1.....	121
Moduł kanału transmisji danych XT1.....	121
System zasilania maszyn rodziny ZAM.....	123
Zestawienie typowych wariantów maszyn rodziny ZAM.....	124
Warunki eksploatacji technicznej maszyn rodziny ZAM.....	127
Niezawodność maszyn.....	127
Obsługa maszyny.....	127
Wymagania mechaniczne i klimatyczne.....	128
Pomieszczenie dla maszyny.....	128
Zasilanie maszyny.....	129
Uwagi końcowe.....	129



# EMAL

(EMAL to skrót nazwy Elektronowa Maszyna Automatycznie Licząca)

Pierwszym polskim projektem był EMAL (lata 1953 - 1955). Maszyna EMAL była szeregową, dwójkową, jedno-adresową. Zbudowana była na logice opartej na starej technologii lampowej i wcześniej opracowanej pamięci rtęciowej o pojemności 512 słów 40-bitowych umieszczonych w 32 rurach z rtęcią pracującej na częstotliwości 750 kHz. Wpływ temperatury był kompensowany przez automatyczną zmianę częstotliwości. EMAL miał stałoprzecinkową arytmetykę, znak plus absolutną wartość. Format danych i rozkazów pokazano w tab. I.

Na Rys.1 pokazano schemat blokowy EMALa. Składa się on z pamięci P, jednostki arytmetyczno - logicznej AR, sterowania S, sprzęgu do pamięci TL i sprzęgów WY i WM dla wejścia/wyjścia. EMAL miał dynamiczne i statyczne rejestry. Statyczne rejestry oznaczono białą czarnymi trójkątami.

Maszyna miała dwa niezależnie adresowalne dostępy do pamięci. Jeden dostęp był dla danych drugi dla rozkazów lecz miał tylko jeden dostęp dla zapisu danych. Mechanizmy te pozwalały na niezależny odczyt danych i rozkazów.

TABLICA I - ROZKAZY MASZINY EMAL

1. Dodaj	$(A) + (P_n) \rightarrow A$
2. Odejmij	$(A) - (P_n) \rightarrow A$
3. Pomnóż i dodaj	$(A) + (M) * (P_n) \rightarrow (A)$
4. Pomnóż i odejmij	$(A) - (M) * (P_n) \rightarrow (A)$
5. Podziel i dodaj	$(A) + (M) * (P_n) \rightarrow (A)$
e. Podziel i odejmij	$(A) - (M) * (P_n) \rightarrow (A)$
7. Pomnóż przez (n)	$(A) * 2^n \rightarrow (A)$
8. Pomnóż przez (-n)	$(A) * 2^{-n} \rightarrow (A)$
9. Przesuń w lewo	$\leftarrow (A)$
10. Przesuń w prawo	$(A) \rightarrow$
11. Wartość absolutna	$  (A)   \rightarrow (A)$

## TRYB 1

12. Suma logiczna	$(M) \cup (P_n) + (A) \rightarrow (A)$
13. Iloczyn logiczny	$(M) \cap (P_n) + (A) \rightarrow (A)$
14. Załaduj	$M_r(P_n) \rightarrow M$
15. Pamiętaj i zeruj A	$(A) \rightarrow P_n, 0 \text{ w } A$

16. Zapamiętaj A (A) → Pn
17. Skok 1 warunkowy n → LR if (A) > 0
18. Skok 2 warunkowy n → LR if (A) < 0
19. Skok bezwarunkowy n → LR
20. Drukuj jeden znak
21. Czytaj jeden znak
22. STOP

## **TRYB 2**

23. Czytaj taśmę

Maszyna EMAL pracowała w dwóch trybach pracy:

- Tryb 1 praca normalna,
- Tryb 2 automatyczne czytanie informacji.

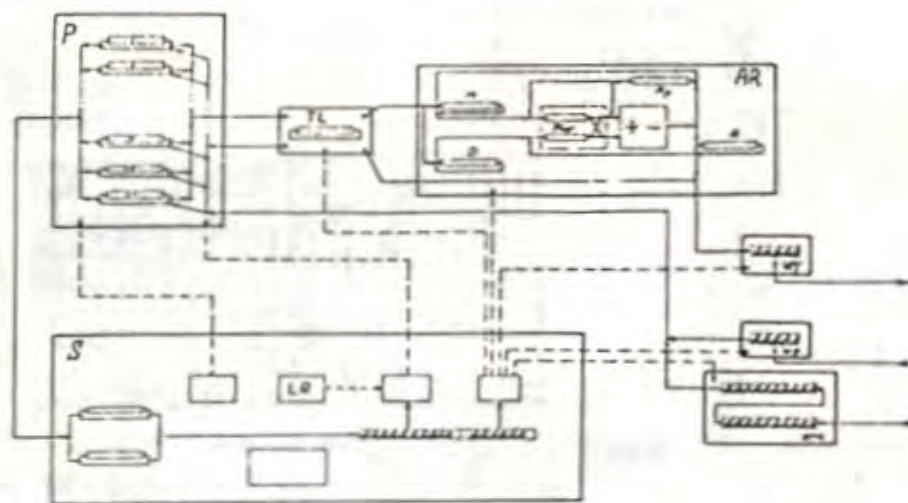
W trybie 2 przez wejście WM przesyła się dane do kolejnych komórek pamięci począwszy od komórki m do końca taśmy z informacjami.

Maszyna ta była wyposażona w dodatkową pamięć dla rozkazów w układzie sterującym w postaci dwóch rejestrów dla przechowywania dwóch rozkazów. Ta dodatkowa pamięć i mechanizm podwójnego dostępu do pamięci powodował, że szybkość pracy maszyny była większa o około 60% niż szybkość bez tych mechanizmów. Dodatkowa pamięć o niezależnym dostępie dla rozkazów bez skoków podnosi szybkość do około 100%, zaś skoki zmniejszyły jej szybkość o 40%. Dlatego, gdy skoki nie były wykonywane nie malała jej szybkość. Pojemność tej dodatkowej pamięci wynikała stąd, że dla pobrania jednego rozkazu potrzeba było jednego dużego cyklu pamięci.

Problem przyspieszania pracy uwzględniając skoki wymagał użycia dodatkowych mechanizmów architekturno-sprzętowych, które po raz pierwszy zapoczątkowano w EMAL'u. Nie jest on jednak do dziś całkowicie rozwiązany. Prowadzone są ciągle badania w tym kierunku na świecie. Niestety maszyna ta w pełni nigdy nie pracowała. Spowodowane to było złą jakością elementów dostępnych wtedy w Polsce (lampy, elementy, łączówki, itp), które powodowały problemy bardzo trudne do pokonania w realizacji tak dużej maszyny pracującej na statycznych przerzutnikach. Trudności te zostały pokonane w następnych trzech polskich maszynach używających lepszych i nowocześniejszych elementów oraz wykorzystujących technikę dynamiczną w układach logicznych.

Jest oczywiste, że tzw. raport von Neumanna i maszyna Wilkesa miały wpływ na projekt EMAL'a, tym niemniej w EMAL'u, jak wynika z dotąd opublikowanych prac, po raz pierwszy na świecie zastosowano specjalne mechanizmy dla przyspieszenia pobierania, a więc i wykonywania rozkazów.

Programy dla komputera EMAL przygotowali Adam Empacher i Andrzej Wakulicz.



Rys.1 Blokowy Schemat Emala.

P - Pamięć, S - Jednostka sterująca, AR - Jednostka arytmetyczna,  
M, D, A, Ap - Rejestry, TL Transformator liczb, WY, WD, WM, - Rejestry  
Wejściowe/Nyjsciowe.

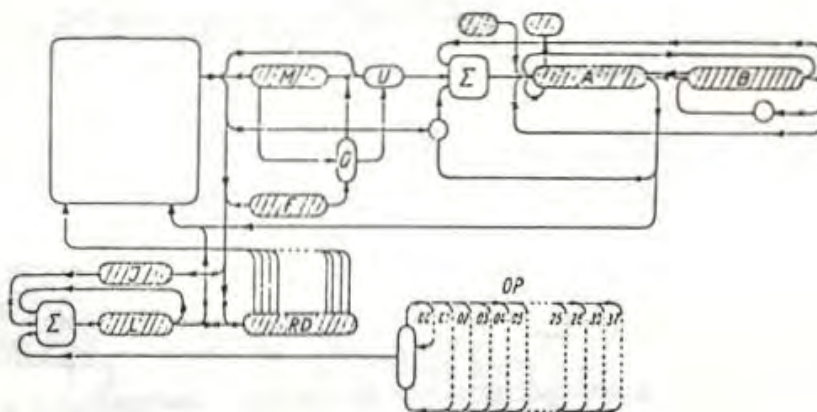
## EMAL 2

(KB - aczkolwiek skonstruowana poza ZAM, maszyna cyfrowa EMAL 2 jest logiczną kontynuacją uprzednio zbudowanej w przez ZAM maszyny EMAL. Wydaje się więc celowym umieszczenie jej opisu w tym opracowaniu).

Maszyna była projektowana i budowana wraz z Kazimierzem Bałakierem, Lesławem Niemczyckim i Andrzejem Harlandem oraz technikami Henrykiem Furmanem, Gustawem Śliwickim, Stefanem Kostrzewą i Zbigniewem Grzywaczem w latach 1957-58.

EMAL 2 miał pamięć bębnową o pojemności 1024 słów rozmieszczonych na 32 ścieżkach. Bęben mógł pracować do 12000 obrotów/m. Ścieżka zerowa była pewnego rodzaju pamięcią ROM i zawierała prosty program wprowadzający. EMAL 2 nie miał dynamicznych rejestrów. Cała logika i rejestry były zbudowane na magnetycznych elementach. Była to maszyna szeregowo o słowie długim 34-bitowym i krótkim 17-bitowym. Pracowała w arytmetyce uzupełnienia do 2.

Dzięki użyciu specjalnie sterowanych generatorów dla zasilania elementów magnetycznych maszyna pobierała stosunkowo mało energii. Pobór energii w dużym zakresie zależał od ilości aktualnie czynnych bloków funkcjonalnych. Równie z tego powodu obniżono czterokrotnie częstotliwość zegara bez obniżenia szybkości pracy. Dlatego w tym celu w pamięci zapisane były na jednej ścieżce szeregowo-równoległe. Obniżyło to częstotliwość zegara ze 108 kHz do 27 kHz, bez zmniejszenia szybkości pracy. Średnia szybkość maszyny wynosiła ponad 150 operacji/sek. Szybkość ta wynikała z szybkości pamięci bębnowej, przy liczbie obrotów bębna 6000 obrotów/m. Rysunek 2 przedstawia schemat blokowy maszyny EMAL 2, zaś tablica II jej rozkazy.



Rys. 2 Schemat blokowy Emala - 2.

TABLICA II Rozkazy EMALa 2

Kod	Rozkaz	Funkcja	Generacja Warunku W	Generacja Nadmiaru N
00	Pobierz rozkaz o adresie n	$(n) \rightarrow RD$	--	0
01	Skocz do adresu zawartego w n	$(n) \rightarrow L$	--	0
02	Skocz do n	$n \rightarrow L$	--	0
03	Skocz warunkowo przy 1	if $W=1$ , $L \rightarrow n$	--	0
04	Skok ze śladem Stop	$L \rightarrow n$ ; $n+1 \rightarrow L$	--	0
05	Stop	$(n) \rightarrow LE$	--	0
06	Skocz warunkowo przy 0	if $W=0$ , $L \rightarrow n$	--	0
07	Skok ze śladem, gdy $N=1$	$(L) \rightarrow n$ ; $n+1 \rightarrow L$	--	0
10	Dodawanie	$(A) + (n) \rightarrow A$	$(A) < 0$	1
11	Odejmowanie	$(A) - (n) \rightarrow A$	$(A) < 0$	1
12	Przesłanie do Akumulatora	$(n) \rightarrow A$	$(A) = 0$	0
13	Przesłanie do Akumulatora -	$(-n) \rightarrow A$	$(A) = 0$	0
14	Przesłanie do pamięci	$(A) \rightarrow P_n$	$(A) = 0$	0
15	Zaokrąglenie	$(A) +/ - 2^{-34}$	$(A) < 0$	0
16	Mnożenie	$(M) * (n) \rightarrow AB$	$(A) < 0$	0
17	Dzielenie nr $\rightarrow B$	$(A) : (n) \rightarrow A$	$(A) < 0$	0
20	Potęgowanie dodatnie	$(AB) * 2^n$	$(A) < 0$	0
21	Potęgowanie ujemne	$E(AB) * 2^{-2p}$	$(A) < 0$	0
22	Przesyłanie cykliczne w prawo	$(AB) \rightarrow n$	$(A) < 0$	0
23	Przesuwanie cykliczne w lewo	$(AB) \leftarrow n$	$(AB) < 0$	0
24	Czytanie Znaku	$(T) 5 \rightarrow A$	$V = 1$	0
25	Drukowanie Znaku	$(A) 5 \rightarrow D$	$Z = 1$	0
26	Bezwzględna wartość	$  (A)   \rightarrow A$	$A = 0$	0
27	Przesłanie do rejestru M	$(n) \rightarrow M$	$(M) < 0$	0
30	Iloczyn logiczny	$(A) \cap (n) \rightarrow A$	$A = 0$	0

Maszyna EMAL 2 miała hierarchiczną strukturę sterowania bardzo podobną do sieci Petriego. Była to maszyna bardzo zwarta i z tego względu łatwa do transportu.

Jedną z istotnych cech maszyny była bardzo wysoka jak na owe czasy, niezawodność logicznych układów ferrytowych. Niezawodność ograniczały elektronowe lampy mocy stosowane do zasilania impulsowego. Niezłym przykładem jej niezawodności było liczenie tablic funkcji Lagrange'a, które trwały 3 miesiące non stop (całodobowo bez awarii).

**Maszyna ta stała się załącznikiem Centrum Obliczeniowego Polskiej Akademii Nauk, które następnie zostało przekształcone w Instytut Podstaw Informatyki Polskiej Akademii Nauk.**

## **P-1**

(Jerzy Mysior)

Wyspecjalizowane oprogramowanie cyklu podstawowego komputera P1 pracującego w czasie rzeczywistym i programy kontrolno diagnostyczne dla P1.

(Józef Winkowski, Michał Wiweger, Jerzy Mysior, Tadeusz Englert, Leopold Łabanowski, Stanisław Różycki, Zbigniew Juszczyk, Jerzy Jagielski).

## GAM-1

([pl.wikipedia.org/wiki/GAM-1](http://pl.wikipedia.org/wiki/GAM-1))

GAM-1 – doświadczalny komputer zerowej generacji opracowany przez Zdzisława Pawlaka. Pierwszy komputer powstały w Grupie Aparatów Matematycznych (GAM) Państwowego Instytutu Matematycznego w Warszawie w 1950 r. Nie był jednak nigdy stosowany do obliczeń praktycznych, służył wyłącznie do celów dydaktycznych. Został rozebrany na części w 1955 r[1].

Podobny, przekąźnikowy komputer dydaktyczny powstał w 1959 r. w Wojskowej Akademii Technicznej.

- arytmetyka binarna
- dwubitowe słowo maszynowe
- zakres liczb: 0-3 (binarnie: 00, 01, 10 i 11)
- rozmiar szyny adresowej: 4 bity
- pamięć operacyjna: 16 bitów (8 komórek pamięci)
- urządzenia we-wy: we - taśma dziurkowana, wy - dwie lampki
- szybkość: 1 rozkaz/s
- rozkazy: dodawania (kod 00), dopełnienia (01), porównania (10) i selekcji (11).
- technologia: przekąźniki.

## XYZ

([pl.wikipedia.org/wiki/XYZ](http://pl.wikipedia.org/wiki/XYZ))



XYZ (1958) – pierwsza Uniwersalna Maszyna Cyfrowa należąca do rodziny wczesnych komputerów, która została zbudowana i uruchomiona w Polsce. Wyprzedziła o kilka miesięcy EMAL-2, natomiast wcześniejszy EMAL nie został w pełni uruchomiony.

Poprzedziły ją: lampowy komputer analogowy ARR oraz nieelektroniczne komputery zerowej generacji: dydaktyczny GAM-1 i użytkowy PARK. Komputer ten zbudowany i uruchomiony został w Warszawie przy ul. Śniadeckich 8, w lokalu Biura Obliczeń i Programów Zakładu Aparatów Matematycznych PAN (późniejszego Instytutu Maszyn Matematycznych)

Zespołem twórców kierował prof. dr inż. Leon Łukaszewicz (wówczas docent). XYZ był modelem laboratoryjnym maszyny użytkowej; po późniejszych zmianach projektowych na bazie tego komputera stworzono serię maszyn ZAM-2.

Organizacja logiczna była wzorowana na uproszczonej IBM 701, ale elektronika na dynamicznych przerzutnikach maszyny M-20 wymagających



dwa razy mniej lamp. Konstrukcja przerzutników i bramek wywodziła się z EMAL, ale diody próżniowe zastąpiono germanowymi. Z maszyny EMAL po doskonaleniu pochodziła także pamięć operacyjna.

Był dynamicznym komputerem szeregowym liczącym w arytmetyce binarnej. Podstawowym układem logicznym był dynamiczny przerzutnik na jednej triodzie (typowo połowce lampy elektronowej) oraz diodowo-ferrytowe bramki OR oraz AND, składające się z transformatora impulsowego i ostrzowych diod germanowych.

### **Budowa:**

Część rejestrów procesora była wykonana na krótkich, rtęciowych liniach opóźniających podobnych do zastosowanych w pamięci operacyjnej, ale mieszczących po jednym słowie.

Maszyna początkowo nie miała stałej pamięci, tylko RAM oparty konstrukcyjnie na opóźnieniu ultradźwięku w rurze wypełnionej rtęcią. W roku 1960(?) została rozbudowana o pamięć bębnową, układ wejścia-wyjścia realizowany poprzez prymitywną konsolę sterującą i reproducer kart (później czytnik/perforator taśmy).

Podstawowymi polami użytkowania XYZ były obliczenia matematyczne, jak też przeliczniki artyleryjskie na potrzeby wojska.

Mimo laboratoryjno-użytkowego charakteru komputera, programista Bogdan Miś napisał dla niego w roku 1958 lub 1960 również program rozrywkowy – grę w kółko i krzyżyk, wykorzystując oscyloskop do prezentowania toku rozgrywki. XYZ doprowadzał do własnej wygranej lub remisu, gdyż w programie gry zawarto wszystkie strategie jej prowadzenia. Kolejnym nienaukowym, a raczej demonstracyjnym, programem napisanym dla XYZ była animacja pieska obsikującego drzewo (również pokazana na ekranie oscyloskopu) stworzona na potrzeby wizyty ekipy filmowej, tworzącej materiał o pierwszym polskim komputerze.

### **Dane techniczne**

Jednoadresowy, dynamiczny komputer szeregowy o sterowaniu układowym  
Arytmetyka binarna, zapis liczb znak-moduł  
Długość słowa: 36 bity  
Technologia: 400 lamp elektronowych i 2000 diod.

### **Szybkość:**

650–4500 dodawań na sekundę (średnio około 1000)  
250–500 mnożeń na sekundę (średnio 350)

**Zegar:** ok. 680 kHz

### **Pamięć:**

- operacyjna pamięć rtęciowa:

- pojemność: 2.25 KiB - 512 słów (32 rury po 576 bitów)
- średni czas dostępu: 0,4 ms
- bębnowa: dodana w 1960 r.
  - głowice stałe
  - pojemność: 36 KiB (64 ścieżki po 128 słów)
  - średni czas dostępu: 20 ms

### **Urządzenia zewnętrzne:**

czytnik i perforator kart

### **Języki programowania**

- język wewnętrzny maszyny
- prosty asembler PROBIN
- makroassembler SAS
- w 1962 polski język algorytmiczny SAKO

### **Zespół**

**Kierownik:** Leon Łukaszewicz

**Projekt logiczny i elektronika:** Antoni Mazurkiewicz, Zdzisław Pawlak, Jerzy Fiett, Zygmunt Sawicki, Jerzy Dańda

**Oprogramowanie:** Antoni Mazurkiewicz, Jan Borowiec, Jerzy Mysior, Krzysztof Moszyński, Jerzy Swianiewicz, Andrzej Wiśniewski.

## Krótki opis programowanej maszyny cyfrowej XYZ – I

Zakład Aparatów Matematycznych Polskiej Akademii Nauk

### Ogólna charakterystyka maszyny XYZ

Zasadnicze dane maszyny XYZ podane są w następującym zestawieniu.

- Rodzaj pracy maszyny : szeregowy
- Rozkazy jedno-adresowe o 18 bitach
- Kod rozkazowy składa się z 24 rozkazów

Liczby w systemie binarnym mogą być 18 i 36-bitowe. Każda liczba przedstawiona jest przez znak i wartość bezwzględną / 1 + 17 lub 1 + 35 bitów /.

Arytmometr w układzie stałoprzecinkowym z dzieleniem. Średni czas wykonywania dodawania lub odejmowania – ok.1 ms, a mnożenia lub dzielenia 2,7 ms.

Pamięć rtęciowa o pojemności 512 słów długich /1024 krótkie/. Przewidywane w przyszłości dołączenie pamięci o 8000-16000 słów długich.

Wejście i wyjście na kartach dziurkowanych z reproducera "Bull".

Stolik umożliwia ręczne wprowadzenie danych, kontrolę optyczną stanu rejestrów oraz ręczne sterowanie niektórymi mikrooperacjami.

Budowa maszyny oparta jest na elementach typowych o technice dynamicznej /impulsowej/. Częstotliwość podstawowa ok. 700 kc/s.

Ilość użytych lamp ok. 400, diod ok. 2000. Pobór mocy ok. 3kW.

Przy opracowywaniu projektu maszyny kierowano się następującymi przesłankami:

- Pełny charakter użytkowy maszyny do średnio obszernych problemów wymagających  $10^6 - 10^7$  operacji arytmetycznych.
- Wykorzystanie opracowanej pamięci rtęciowej i dynamicznych elementów logicznych pracujących z częstotliwością ok. 700 kc/s.

- Prostota logiczna i układowa oraz zapewnienie niezawodności działania maszyny nawet kosztem zwiększenia ilości użytego sprzętu.

Kod rozkazowy maszyny XYZ obejmuje wszystkie podstawowe działania arytmetyczne łącznie z różnicą wartości bezwzględnych i dzieleniem.

Operacje te są wykonywane zgodnie z prawami liczenia na liczbach zakresu  $-1 < x < +1$  tj. w zakresie wartości bezwzględnych mniejszych od jedności.

Przy działaniach na liczbach o większej wartości bezwzględnej stosujemy odpowiednie przesunięcia. Przy mnożeniu otrzymujemy iloczyn podwójnej długości 70 bitów, a przy dzieleniu dzielną zawiera 70 bitów. Pozwala to na stosowanie programów o podwójnej długości słowa.

Możliwe jest również przeprowadzenie obliczeń na liczbach krótkich /18-bitowych/, co przy mniejszej dokładności daje znaczne zwiększenie ilości miejsc pamięci.

Programowana, warunkowa zmiana obliczeń może być uzależniona od znaku, lub zerowej wartości akumulatora, jak również od stanu wskaźnika nadmiar, sygnalizującego przekroczenie wartości 1 przez moduł liczby w akumulatorze.

Schematy logiczne maszyny starano się rozwiązać możliwie prosto. Bardziej złożone czynności maszyny sprowadzono do wykonywania kilkunastu czynności prostszych, tzw. mikrooperacji. Obwody szeregu mikrooperacji zostały wydzielone i stanowią niezależne całości.

Ponadto dla celów kontrolnych sterowanie wielu mikrooperacji zostało wyprowadzone do stolika i może być uruchomione indywidualnie niezależnie, od układu głównego sterowania. Dane mogą być też wprowadzane do maszyny ręcznie poprzez rejestr P, pośredniczący między pamięcią i innymi częściami maszyny. Obserwacje podstawowych przebiegów mogą być dokonywane na wmontowanym oscyloskopie.

## **Słowa, liczby i rozkazy**

W dalszym ciągu przez termin bit rozumieć będziemy dowolny ze znaków "1" lub "0". Znaki te w XYZ są reprezentowane przez impulsy elektryczne. Pojawienie się impulsu w danym miejscu maszyny i w określonej chwili oznacza bit "1", a jego brak – bit "0". Ciągi po 18 lub 36 bitów nazywać będziemy słowami.

Numeracja bitów w słowie następuje kolejno w czasie zgodnie z numeracją impulsów wytwarzanych przez urządzenie zwane zegarem. Zegar w maszynie XYZ wysyła impulsy periodycznie co okres wynoszący  $1,4\mu s$ . Impulsy te synchronizują pracę całej maszyny tzn. wyznaczają chwile ewentualnego występowania impulsów w określonym miejscu maszyny. Następujące po sobie grupy kolejnych 36-ciu impulsów, począwszy od impulsu mającego numer 0, a skończywszy na 35-tym tworzą tzw. okresy duże T maszyny, odpowiadające słowom długim /36-bitowym/. Numeracja bitów w słowie długim rozpoczyna się zgodnie z pojawieniem się ich w czasie od bitu "Zerowego" a kończy na 35-tym.

W każdym słowie długim wyróżnić możemy dwa słowa krótkie z których "prawe" zajmuje w słowie długim 18 bitów ponumerowanych od 0 do 17, a "lewe" 18 bitów ponumerowanych od 18 do 35.

Przez k-ty bit słowa krótkiego rozumieć będziemy: k-ty bit słowa długiego w przypadku słowa prawego i k+18 bit słowa długiego w przypadku słowa lewego. Słowa stanowią zarówno liczby, jak i rozkazy, przy czym rozkazy są zawsze słowami krótkimi.

Bitów używamy do zapisu liczb w układzie binarnym.

Liczby wyrażamy w maszynie XYZ przy pomocy znaku i modułu /wartości bezwzględnej/. Następujące kolejno w czasie bity od zerowego do przedostatniego oznaczają cyfry w układzie binarnym, począwszy od najmniej znaczącej. Ostatni bit oznacza znak /zero oznacza +, jedynek oznacza minus/

Liczba w XYZ może zajmować całe długie słowo, wówczas jej znak jest wyrażony przez bit 35-ty a moduł przez pozostałe bity. Liczbę taką nazywamy długą. Liczba może również zajmować tylko słowo krótkie, wówczas jej znak wyrażony jest przez bit 17 w wypadku słowa prawego, lub przez bit 35 w wypadku słowa lewego.

Położenia przecinka w liczbach nie wyrażamy żadnymi dodatkowymi bitami w słowie. Arytmometr XYZ wykonuje działania arytmetyczne tak, jak gdyby przecinek umieszczony był bezpośrednio na lewo od skrajnej lewej cyfry modułu liczby, tzn. wszystkie liczby traktuje jako ułamki właściwe. Jednakże przy pomocy pewnych rozkazów /rozkazów przesunięcia/ możemy działanie

arytmetyczne modyfikować tak, jak gdyby przecinek umieszczony był np. w środku liczby.

Rozkaz zajmuje zawsze słowo krótkie lewe albo prawe, składa się z części operacyjnej i części adresowej. Część operacyjna określa rodzaj operacji. Część adresowa oznacza adres miejsca w pamięci dla operacji związanych z pamięcią. Dla operacji niezwiązanych z pamięcią część adresowa ma inne znaczenie.

Rozmieszczenie rozkazów i liczb w słowie i ich budowę przedstawia rys.1.

Arytmometr i urządzenia wejścia – wyjścia operują tylko słowami długimi. Sterowanie pobiera rozkazy z pamięci tylko w postaci słów krótkich. Pamięć natomiast przechowuje słowa w sposób taki, że mogą być traktowane jako długie, albo jako krótkie.

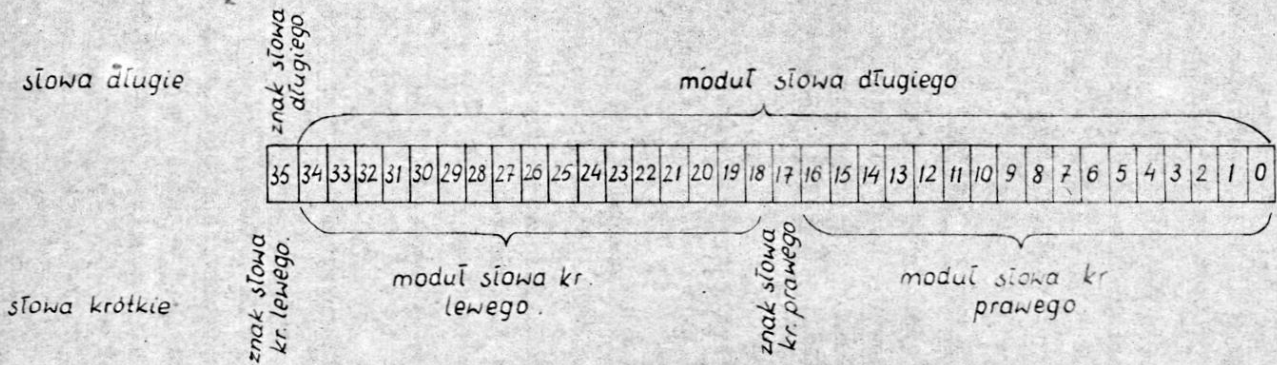
Gdy słowo, prawe lub lewe, jest przesyłane z pamięci do arytmmometru, to po drodze zostaje przekształcone w słowo długie, którego 18 lewych bitów stanowi przesyłane słowo krótkie, a pozostałe 18 prawych bitów są zerami. Podobnie lewe 18 bitów słowa długiego, zapisanego w arytmmetrze może być przesłane do pamięci jako pół-słowo lewe lub prawe. Na to, czy przesłane ma być słowo długie czy krótkie wskazuje bit 11 adresu zawartego w rozkazie.

### **Adresowanie słów**

W pamięci wewnętrznej XYZ mieści się 512 słów długich albo 1024 słowa krótkie. Słowa krótkie numerowane są kolejno od 0 do 1023 i numery te umieszczone są w rozkazach jako adresy. Dla zaznaczenia, że adres odnosi się do słowa krótkiego bit 11 adresu /tzw "znak adresu"/ czynimy równy zero " + ". Słowa długie adresujemy liczbami parzystymi od 0 do 1022, a dla zaznaczenia, że, adres odnosi się do słowa długiego "znak adresu" /bit 11/ czynimy równy jedności "-" Np. adresy + 1022 i + 103 odnosi się do słów krótkich, zaś adresy - 102 i - 1022 odnoszą się do słów długich. Należy podkreślić, że tzw "znak adresu" nie ma własności znaku liczby. Jest on po prostu cyfrą binarną. Tak więc dla adresów mamy np.: - 1020 + 2 = - 1022.

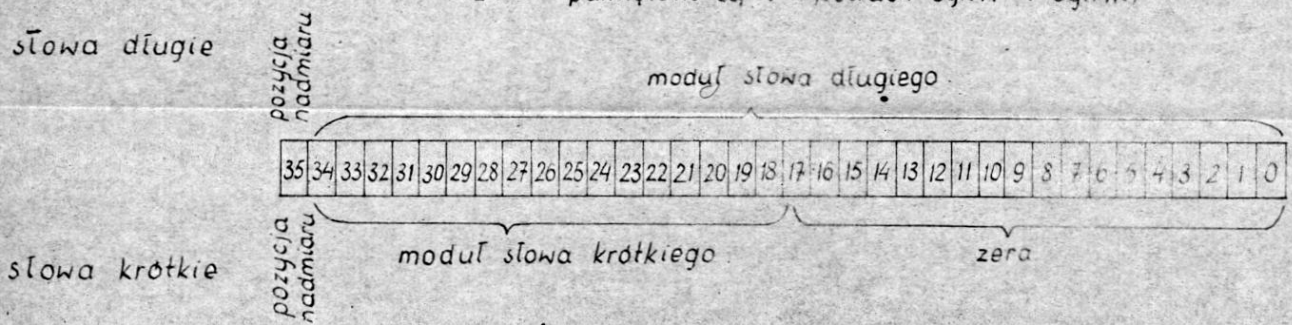
Rys. 1.

Rozmieszczenie słowa w pamięci.



Rozmieszczenie słowa w rejestrach A i M.

(znaki pamiętane są w rejestrach Sgn A i Sgn M)



Budowa rozkazu.



pozycje wykreślone nie wchodzą do rejestru R.

## Części składowe maszyny

Schemat blokowy XYZ przedstawiono na rys.2. Wyróżniamy w nim następujące części składowe:

1. Pamięć – składająca się z 32 rur rtęciowych o pojemności 16 słów 36-bitowych każda. Słowa te obiegają rurę w w ok. 850 $\mu$ s. Położenie słów w rurach określone jest przez licznik czasu LT. Wprowadzenie względnie pobieranie liczb z pamięci odbywa się wg adresu, zapisanego w aR.

2. Rejestr przerzutnikowy P. pracujący jako:

- Rejestr pośredniczący między pamięcią a innymi układami maszyny cyfrowej. Wszystkie słowa są zapisywane lub pobierane z pamięci poprzez rejestr P., który również w razie potrzeby przekształca słowa długie w krótkie lub odwrotnie.
- Rejestr Wejścia i Wyjścia. Karty w reproducerze są odczytywane lub dziurkowane za pośrednictwem rejestru P.
- Rejestr sterowania. Pobieranie rozkazów odbywa się za pośrednictwem rejestru P. Rejestr pomocniczy arytmometru, zawierający jeden ze składników przy dodawaniu i odejmowaniu, mnożną przy mnożeniu i dzielnik przy dzieleniu.

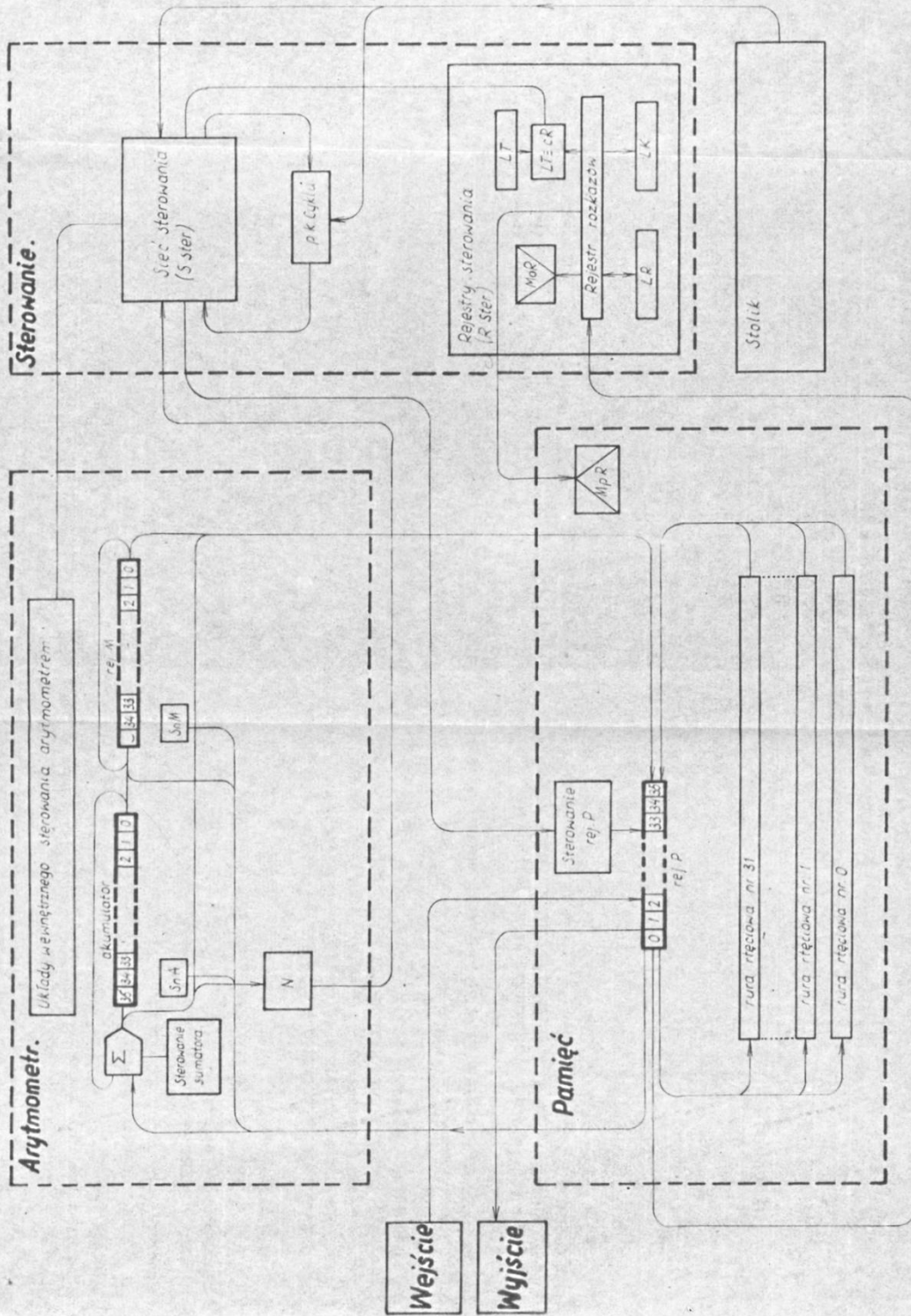
Rejestr P może być ładowany ręcznie ze stolika bit po bicie.

3. Arytmometr składający się z:

- a) Akumulatora /rejestr A/ zawierającego 36 pozycji. Oprócz 35 pozycji modułu akumulator zawiera dodatkową pozycję modułu, o wartości liczbowej równej  $2^0 = 1$ . Pozycja ta nie ma odpowiednika w pozostałych składnikach maszyny, między innymi nie może przyjmować informacji z pamięci ani przesyłać swojej zawartości do pamięci. Poza tym pozycja ta ma te same własności przesuwania, dodawania, itd. jak pozostałe pozycje 0-34 akumulatora. Celem wprowadzenia tej pozycji jest notowanie bitów powstałych z ew. przekroczenia zakresu np. gdy w wyniku dodania liczby do akumulatora otrzymujemy wynik większy od jedności,
- b) Rejestru M, zawierającego mnożnik przy mnożeniu i iloraz powstały w wyniku dzielenia, składającego się z 35 pozycji modułu,
- c) Sumatora umożliwiającego operacje dodawania, odejmowania, mnożenia i dzielenia,
- d) Rejestru SnA. zawierającego znak liczby, której moduł jest w akumulatorze,
- e) Rejestru SnM. zawierającego znak liczby, której moduł jest w rejestrze M.



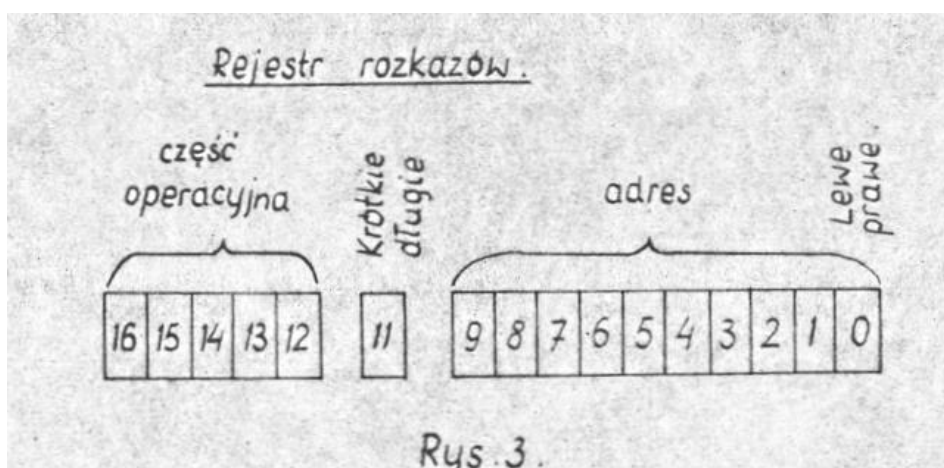
# Schemat blokowy maszyny XYZ.



4. Urządzenia Wejścia i Wyjścia, których podstawą jest reproducer przekazujący maszynie informacje wydziurkowane na kartach, lub dziurkujący karty wg informacji uzyskanych z maszyny. Przekazywanie tych informacji odbywa się zawsze za pośrednictwem rejestru P.

5. Sterowanie składające się z:

- a) Rejestru rozkazów R, dzielącego się na pięcio-pozycyjną część operacyjną oR i dwunasto-pozycyjną część adresową aR. Zawartość oR określa rodzaj wykonywanej operacji, zaś zawartość aR adres składnika lub parametr operacji. Rozmieszczenie poszczególnych części rozkazu w rejestrze R pokazuje rys.3.
- b) Rejestru nadmiaru N sygnalizującego przekroczenie zakresu akumulatora. Wyzerowanie rejestru N zachodzi jedynie w wyniku specjalnego rozkazu, a nie jest związane ze zniknięciem jedynki z pozycji nadmiarowej akumulatora.
- c) Licznika rozkazów LR, zawierającego adresy kolejnych rozkazów. Wartość jego po wykonaniu jakiegokolwiek rozkazu zwiększa się o jedność, albo też przy rozkazach zmieniających sekwencje rozkazów staje się równa zawartości aR.
- d) Licznika kroków, określającego czas trwania operacji długich /mnożenie, dzielenie, przesuwanie w lewo i prawo/.
- e) Licznika czasu, określającego pozycje określonego przez aR słowa w rurze.
- f) Sieci sterującej, rządzącej przekazywaniem i przekształcaniem informacji w całej maszynie.
- g) Stolika operatora, umożliwiającego kierowanie pracą maszyny i jej kontrolę przez operatora. Bliższe informacje dotyczące stolika podane zostaną w rozdziale 9.



## **Przesyłanie informacji w maszynie**

Przesyłanie informacji w XYZ odbywa się dwoma sposobami, szeregowo i równoległe.

Przesyłanie szeregowe polega na przesyłaniu słowa z jednego miejsca maszyny w drugie bit po bicie, przy czym czas przesłania jednego bitu wynosi  $\tau$  a całego słowa  $T = 36\tau$ . Szeregowe przesyłanie jest stosowane przy przekazywaniu informacji:

- z pamięci do rejestru P i odwrotnie,
- z rejestru P do rejestru R
- z rejestru P do rejestru do rejestru A i odwrotnie
- z rejestru P do rejestru M i odwrotnie

Przesyłanie równoległe polega na przesyłaniu wszystkich bitów słowa jednocześnie,

Równoległe przesyła się:

- Zawartość wiersza karty z reproducera do rejestru P
- Zawartość rejestru P do reproducera celem wydziurkowania,
- z części aR rejestru R do LR i odwrotnie
- z części aR rejestru R do LK
- z części aR do przerzutników "Inf."

## **Cykl pracy maszyny**

Maszyna pracuje, wykonując kolejno cykle pracy. Cykl taki składa się zawsze z dwóch czynności: pobranie rozkazu i wykonanie pobranego rozkazu. Cykl pracy maszyny przedstawia schematycznie rys. 4.

W prostokątach zaznaczone są symbolicznie czynności maszyny, a linie opatrzone strzałkami wskazują na ich kolejność.

Pierwsza czynność cyklu:  $/n/\rightarrow P$  następuje albo po naciśnięciu guzika "Start", albo po zakończeniu poprzedniego cyklu. Symbol  $/n/\rightarrow P$  oznacza "Prześlij zawartość n-tego miejsca w pamięci do rejestru P". W tym przypadku zawartość ta przedstawia rozkaz, który ma być wykonany w cyklu pracy. Wybieranie tego rozkazu dokonuje się zgodnie ze stanem aR.

Następna czynność  $P\rightarrow R$  powoduje przeniesienie pobranego rozkazu z rejestru P do R.

W wyniku więc dwóch pierwszych czynności został pobrany z pamięci i umieszczony w R odpowiedni rozkaz.

Następna czynność: Op polega na wykonaniu operacji określonej pobranym rozkazem /tj. zawartością, R/.

Jeżeli w rejestrze R znajdował się np. rozkaz „Zawartość n-tego miejsca w pamięci dodaj do zawartości akumulatora” to przebieg czynności maszyny jest mniej więcej następujący. Najpierw zawartość n-tego miejsca w pamięci zostaje przesłana do rejestru P

podobnie jak w przypadku pierwszego kroku cyklu. Pierwszych pięć bitów zawartych w aR określa numer rury, w której znajduje się szukane słowo, a ostatnie pięć bitów pozycję tego słowa w rurze. Pozycja ciągu słów w rurze wskazywana jest przez LT, po zrównaniu się jego stanu z ostatnimi pięcioma bitami aR następuje pobranie słowa. Dodatkowo bit 11 wskazuje czy mamy pobrać słowo długie, czy krótkie. Gdy liczba jest umieszczona w P następuje dodanie jej do zawartości A przy pomocy sumatora.

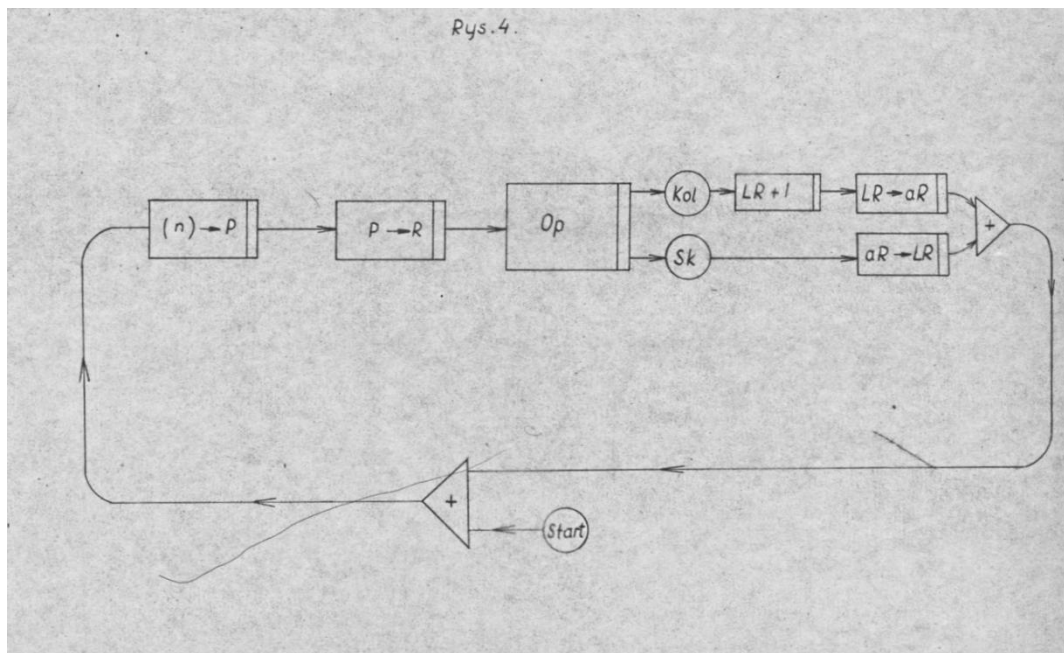
Przy rozkazie "przesuń zawartość A i M w lewo" stan aR zostaje przesłany do LK i rozpoczyna się przesuwanie. Po każdym przesunięciu o jedno miejsce, stan LK zmniejsza się o jedność, a gdy osiągnie 0, wykonanie rozkazu zostaje zakończone.

Po wykonaniu rozkazu np. arytmetycznego, dalsze czynności przebiegają po linii zaznaczonej na rys. 4, zapisem "Kol" /kolejny/, tzn. zawartość licznika rozkazów zostaje zwiększona o 1 /LR + 1/, a następnie przesłana do aR /LR → aR /.

W ten sposób w następnym cyklu pracy zostanie pobrany kolejny co do numeru rozkaz. LR spełni więc tutaj rolę rejestru pamiętającego adres wykonywanego rozkazu.

Przy rozkazach skocz, stop oraz przy spełnieniu warunków dla rozkazów warunkowych dalsza czynność przebiega po linii zaznaczonej na rys. 4 napisem "sk" /skok/, tj. zawartość aR zostaje przesłana do LR. W ten sposób dalsza sekwencja pobieranych rozkazów rozpocznie się od adresu zawartego w rozkazie warunkowym.

Po wymienionych wyżej czynnościach, jednych lub drugich, rozpoczyna się następny cykl pracy maszyny.



## Zestawienie rozkazów XYZ

W kolumnie "Kod" podany został dziesiętny odpowiednik kodu operacji zawartej w rozkazie, który w rzeczywistości wyrażony jest przy pomocy pięcio-cyfrowej liczby w układzie binarnym. Dla oznaczenia rozkazu można używać zarówno jego symbolu, jak i skrót, które podane zostały w kolumnie drugiej i trzeciej tablicy.

Dwie następne kolumny zawierają czas minimalny i średni wykonywania danego rozkazu. Czas działania jest czasem wykonania rozkazu odliczając czas oczekiwania i pobrania rozkazu i ew. czas oczekiwania na odpowiednie słowo w pamięci /np. przy dodawaniu/. Czas średni obliczony jest przy założeniu, że wszystkie czasy oczekiwania równe są 7,5T. W ostatniej kolumnie podana jest pełna nazwa rozkazu.

Tablica I - Zestawienie rozkazów XYZ

Kod	Symbol	Skrót	Czas dział.	Czas średni	Grupa Rozk.	Pełna nazwa
0	Stop n	Stop	1 T	10,5 T	b	Stop i skocz
1	n→R	Roz	2 T*	9,5 T*		Wykonaj podany rozkaz
2	Sk n	Sk	1 T	10,5 T	b	Skocz
3	Sk + n	Sk +	1 T	10,5 T	b	" przy plusie
4	SkO n	SkO	1 T	10,5 T	b	" zerze
5	SkN n	SkN	1 T	10,5 T	b	" nadmiarze
6	A +(n)	Do	2 T	19 T	a	Dodaj
7	A -(n)	Od	2 T	19 T	a	Odejmij
8	A -(n)	Od	2 T	19 T	a	Odejmij Absolutne Wartości
9	M · (n)	Mn	38 T	55 T	a	Pomnóż
10	AM : (n)	Dz	38 T	55 T	a	Podziel
11	n→A	do A	2 T	19 T	a	Zeruj i dodaj
12	n→M	do M	2 T	19 T	a	Naładuj M
13	A→n	zA	2 T	19 T	a	Zapamiętaj A
14	M→n	zM	2 T	19 T	a	" "
15	Okr. n	Okr.	1 T	10,5 T	b	zaokrąglj
16	Am. 2 <sup>n</sup>	Lw.	n+1 T	(n+10,5)T	b	Przesun Am w lewo
17	AM: 2 <sup>n</sup>	Pr.	n+1 T	(n+10,5)T	b	" " w prawo
18	PgCz	PgCz	1 T	10,5 T	b	Przygotuj czytanie
19	PgPsn	PgPsn	1 T	10,5 T	b	Przygotuj pisanie
20	Cz n	Cz			a	Czytaj
21	Psn n	Psn			a	Pisz
22	Inf.	Inf.	1 T	10,5 T	b	Informuj
23	Pkp	Pkp	2 T	10,5 T	b	Pisz koniec pliku

Dokładny czas:

- Dla rozkazów typu a Czas całkowity = czas oczekiwania na rozkaz + 2T + czas działania + oczekiwania na liczbę.
- Dla rozkazów typu b czas całkowity = czas oczekiwania na rozkaz + 2T + czas działania.
- Dla rozkazu "Roz" Czas całkowity = czas oczekiwania na rozkaz + czas całkowity dla rozkazu określonego adresem. Należy do tego czasu dodać czas wykonania rozkazu określonego adresem rozkazu "Roz". W wypadku czasu średniego dodajemy czas średni.

## Opis rozkazów XYZ

Kod	Symbol	Objaśnienie
0	<b>Stop n</b>	Zatrzymaj maszynę. Po naciśnięciu guzika "start" jako pierwszy rozkaz weź zawartość n-tego miejsca w pamięci. [Przerwanie cykli pracy i $aR \rightarrow LR$ ].
1	<b>N → R</b>	Wykonaj rozkaz zawarty w n-tym miejscu pamięci. Weź kolejny rozkaz z niezmienionej sekwencji rozkazów. [wprowadzenie z pamięci n rozkazu do R i wykonanie jego. Poprzednia zawartość LR zwiększa się o 1].
2	<b>Sk n</b>	Weź następny rozkaz z n-tej pamięci [ $aR \rightarrow LR$ ].
3	<b>Sk+n</b>	Jeżeli znak A równa się plus (0), następny rozkaz weź z n-tej pamięci. W przeciwnym przypadku weź rozkaz kolejny. [Jeżeli $S_n A = 0$ to $aR \rightarrow LR$ ].
4	<b>Sk0 n</b>	Jeżeli zawartość A równa jest liczbie +0 lub -0, następny rozkaz weź z n-tej pamięci. W przeciwnym przypadku weź rozkaz kolejny. Zawartość pozycji nadmiaru w rejestrze A nie wpływa na przebieg operacji. [Jeżeli $A = 0$ to $aR \rightarrow LR$ ].
5	<b>SkN n</b>	Jeżeli w wyniku jakiegokolwiek poprzedniej operacji licząc od ostatniego rozkazu $SkN$ /lub uruchomienia maszyny/ nastąpiło przekroczenie zakresu pozycji 0-34 akumulatora, następny rozkaz weź z n-tej pamięci. [Wyzeruj rejestr N. Przekroczenie zakresu powoduje $N=1$ i wówczas rozkaz $SkN$ powoduje $aR \rightarrow LR$ i powrót do $N=0$ ]
6	<b>A+ (n) *</b>	Do zawartości akumulatora dodaj zawartość n-tego miejsca w pamięci i sumę umieść w akumulatorze [ $n \rightarrow P$ $P=A \rightarrow A$ ]
		* Jeśli wyniku tych operacji, otrzymujemy zero, to znak akumulatora jest odwrotny do znaku sprzed operacji.
7	<b>A- (n) *</b>	Jak w p.6, tylko zamiast dodawania mamy odejmowanie, ewentualna sygnalizacja nadmiaru.
8	<b>A- (n) **</b>	Jak p.6, tylko odejmowanie dotyczy bezwzględnych wartości zawartości akumulatora i n-tego miejsca w pamięci.
9	<b>M . (n)</b>	skasuj poprzednią zawartość A. Pomnóż zawartość M przez zawartość n-tego miejsca w pamięci. Z 70 cyfr pełnego iloczynu, 35 lewych umieść w pozycjach 0-34 akumulatora, a 35 prawych w pozycjach 0-34 rejestru M. Znaki A i M uczynić równe znakowi iloczynu [ $n \rightarrow P$ , $P.M \rightarrow AM$ ]
10	<b>AM:n</b>	Podziel łączną zawartość rejestrów A i M traktując ją jako jedną liczbę o znaku równym $S_n A$ i module złożonym z pozycji 0-34 rejestru A i pozycji 0-34 rejestru M, przez zawartość n-tego miejsca w pamięci, 35 cyfr iloczynu umieść w M., a 35 cyfr reszty w A. [ $n \rightarrow P$ , $A:P \rightarrow M$ znak $M =$ znakowi ilorazu. Ewentualna sygnalizacja nadmiaru.]
11	<b>(n )→ A</b>	Skasuj poprzednią zawartość A. Zawartość n-tego miejsca w pamięci prześlij do akumulatora [ $n \rightarrow P$ , $P \rightarrow A$ ]
12	<b>(n) → M</b>	Skasuj poprzednią zawartość M. Zawartość n-tego miejsca w pamięci prześlij do rejestru M.

		[ $n \rightarrow P$ , $P \rightarrow M$ ]
13	<b>A → (n)</b>	Zawartość akumulatora prześlij do n-tego miejsca w pamięci [ $A \rightarrow P$ , $P \rightarrow n$ ].
14	<b>M → (n)</b>	Zawartość rejestru M prześlij do n-tego miejsca w pamięci [ $M \rightarrow P$ , $\rightarrow n$ ].
15	<b>Okr.</b>	W ostatniej najmniej ważnej pozycji rejestru M zapisz 1. Jeżeli w pierwszej najważniejszej pozycji rejestru M jest cyfra 1 dodaj jedność do ostatniej pozycji akumulatora. Ewentualna sygnalizacja nadmiaru].
		*/ Jeżeli w wyniku tych operacji, otrzymujemy zero to znak akumulatora jest odwrotny do znaku z przed operacji. **/ Jeżeli w wyniku operacji 8 otrzymujemy zero znak jego jest minus.
16	<b>AM*2n</b>	Łączną zawartość rejestrów A i M przesun o n miejsc w lewo. Znak A uczyn równy znakowi M. [ $aR \rightarrow LK$ . Ewentualna sygnalizacja nadmiaru].
17	<b>AM:2n</b>	Jak wyżej, lecz w prawo. Bez sygnalizacji nadmiaru].
18	<b>PgCz</b>	Włącz mechanizm podający karty. Za każdym razem, gdy kolejny wiersz karty jest gotów do czytania, podaj maszynie impuls <b>Syn</b> /synchronizujący/. Gdy jakikolwiek wiersz nie zostanie odczytany przez rozkaz Cz, wstrzymaj podawanie impulsów synchronizujących, jak również zatrzymaj podanie następnyc kart.
19	<b>PgPsn</b>	Jak wyżej, tylko czytanie kart zastąpione jest dziurkowaniem i rozkaz Cz rozkazem Psn.
20	<b>Cz</b>	Na impuls synchronizujący z urządzenia czytającego karty /por. rozkaz PgCz/ przeczytaj treść bieżącego wiersza i umieść w n-tym miejscu pamięci. [Zawartość bieżącego wiersza $\rightarrow$ ., $P \rightarrow n$ .] Przy znaku "Koniec pliku" przeskocz przez cztery rozkazy i skasuj " PgCz" dla następnej karty.
21	<b>Psn</b>	Na impuls synchronizujący z urządzenia drukującego karty /por. PgPsn/, zawartość n-tego miejsca w pamięci wydrukuj w bieżącym wierszu Karty .W wypadku niepodania rozkazu Psn dla jakiegoś wiersza Karty skasuj stan Pg Psn dla następnej Karty. [ $n \rightarrow P$ , $P \rightarrow$ bieżący wiersz Karty].
22	<b>Inf.</b>	Na stoliku operatora zapal kombinację lampek Inf. Wskazaną przez ostatnie sześć bitów adresu. Bitom "1" odpowiada lampka zapalona, bitom "0" - zgaszona.
23	<b>Pkp</b>	W bieżącym wierszu karty wypisz symbol "Koniec pliku". /Rozkaz ten nie może następować po rozkazie Psn w czasie dłuższym niż 15 ms. Jeżeli w czasie $\leq 15$ ms. po poprzednim Psn, a za Pkp pojawi się znów rozkaz Psn, zawartość rejestru P zostanie wypisana w tym samym wierszu co "koniec pliku". Brak rozkazu Psn, po rozkazie Pkp powoduje skasowanie stanu PgPsn dla następnej karty.

## **Blizsze dane dotyczące wejścia i wyjścia**

W XYZ przewidziano zastosowanie kart dziurkowanych systemu 80-kolumnowego. Słowa zapisywane są w 12 wierszach karty; w każdym wierszu zapisujemy jedno słowo 36-bitowe, w kolumnach 1-36 licząc od lewej. Razem więc na karcie możemy zapisać 12 słów długich lub 24 słowa krótkie.

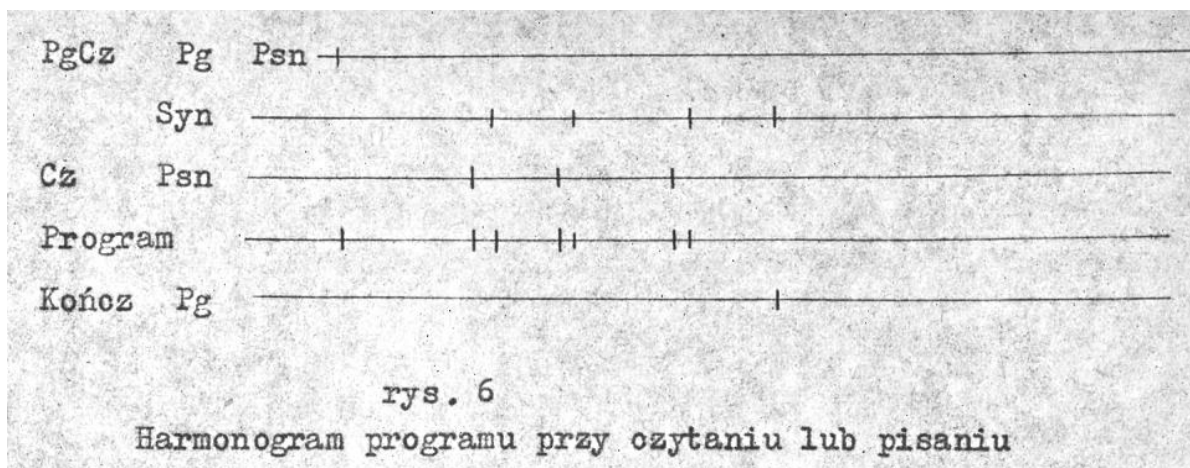
Kolumna 37 może zawierać otwory, zwane znakami końca pliku. Znak taki powoduje, że:

- maszyna przeskoczy przez 4 rozkazy,
- stan przygotuj zostanie skasowany tzn. impulsy synchronizujące przestaną być wydawane i reproducer nie poda następnej karty.

Znaki końca pliku mają na celu załatwienie zakończenia czytania określonego pliku kart.

Jeśli w pewnych wierszach zapisane są liczby, np. w systemie dziesiętnym, to przy pomocy odpowiedniego programu maszyna pobiera te liczby, tworzy ich "obraz" w pamięci i przekształca np. na zapis dwójkowy. Odwrotnie, dla zapisania na karcie np. liczb w układzie dziesiętnym, maszyna przy pomocy programu tworzy najpierw "obraz karty" w pamięci, który następnie drukuje na karcie.

Harmonogram programu przy czytaniu lub pisaniu przedstawiony jest na rys. 6.



Program ten rozpoczyna się rozkazem PgCz /lub Pg Psn/, który powoduje rozpoczęcie podawania kart w reproducerze. Po tym rozkazie maszyna wykonuje dalsze, aż do momentu wydania rozkazu Cz /lub Psn/. Wówczas to maszyna wstrzymuje pracę, aż do otrzymania impulsu Syn, wskazującego gotowość do przeczytania /wydziurkowania/ odpowiedniego wiersza. Impuls Syn powoduje wydawanie rozkazu Cz /Psn / i przejście do następnych rozkazów aż do nowego rozkazu Cz /Psn/, który znów wstrzymuje pracę maszyny



itd. Dla przeczytania /wydziurkowania/ danego wiersza konieczne jest poprzedzenie impulsu Syn przez rozkaz Cz /Psn/. Gdy pojawi się impuls Syn nie poprzedzony takim rozkazem, stan wywołany rozkazem Pg zostaje skasowany, tzn. nie zostaną już wydane, dalsze impulsy Syn i podanie następnej karty zostanie wstrzymane.

Pojawienie się w rej. R rozkazu PgCz powoduje uruchomienie urządzenia podającego karty. Od tej chwili, aż do skasowania stanu "przygotuj czytanie". Karty będą podawane z szybkością 1 karty na każde 0,5 sek. Od chwili pojawienia się rozkazu PgCz w rej. R do chwili w której należy wydać rozkaz Cz dla pierwszego czytanego wiersza karty upływa ok. 140 ms. Czas ten może być wykorzystany przez maszynę na wykonanie programu.

Czas do wykorzystania między kolejnymi rozkazami "Cz" dla wierszy jednej karty wynosi do 23 ms. Między rozkazem "Cz" dla ostatniego wiersza karty a rozkazem "Cz" dla pierwszego wiersza karty następnej można wykorzystać na wykonanie programu do 180 ms.

Jak podano wyżej niepodanie w odpowiednim czasie rozkazu "Cz" /"Psn"/ powoduje skasowanie stanu "Pg Cz" /"Pg Psn"/. Aby spowodować przeczytanie /napisanie/ następnego wiersza, należy rozkaz "Cz" /"Psn"/ poprzedzić rozkazem "Pg Cz" /"Pg Psn" n/.

Pojawienie się w rejestrze R rozkazu Pg Psn powoduje uruchomienie urządzenia podającego karty, które będzie podawało karty z szybkością 1 karty na sekundę, aż do chwili skasowania stanu Pg Psn.

Od chwili wydania rozkazu Pg Psn do chwili w której powinien pojawić się rozkaz "Psn" dla pierwszego wiersza karty upływa ok. 720 ms. Czas ten może być wykorzystany przez wykonywanie programu.

Czas do wykorzystania między rozkazami "Psn" dla kolejnych wierszy jednej karty nie powinien przekraczać 15 ms.

Między rozkazem "Psn" dla ostatniego wiersza karty a rozkazem "Psn" dla pierwszego wiersza następnej karty można wykorzystać do 650 ms.

Uwaga: Podanie rozkazu "Cz" /"Psn"/ niepoprzedzony w odpowiednim czasie rozkazem "PgCz" /"Pg Psn"/ spowoduje zatrzymanie maszyny w połowie cyklu. Ponowne uruchomienie musi być w tym wypadku poprzedzone załadowaniem rej. R adresem pod którym znajduje się następny rozkaz.

Rozkaz "Pkp" powoduje wydziurkowanie symbolu "koniec pliku" w wierszu karty który się najprędzej pojawi. Rozkaz ten jest niezależny od rozkazów "Pg Psn" i "Psn".

## Stolik operatora

Operator kieruje i kontroluje pracę maszyny posługując się stolikiem operatora. Stolik ten zawiera następujące urządzenia:

1. **Klucz** yalowski którego przekręcenie uruchamia stolik, przy kluczu nieprzekręconym stolik jest nieczynny. Klucz ten zabezpiecza przed manipulowaniem na stoliku przez osoby nieuprawnione.
2. Guzik **Start** - uruchamiający cykle pracy maszyny.
3. Guzik **Stop** - zatrzymujący cykle pracy maszyny.
4. Guzik **Kas.** - kasujący zawartość wszystkich rejestrów /realizacja - wstrzymanie impulsów katodowych wszystkich przerzutników/.
5. Guzik **1 Cykl** - uruchamiający pojedynczy cykl pracy maszyny.
6. Guzik **n Cykli** - powodujący /podczas przyciskania/ wykonywanie kolejnych cykli pracy z częstotliwością około 5 cykli pracy na sekundę.
7. Guzik **Op** - powodujący wykonanie danej operacji określonej rozkazem znajdującym się w rejestrze rozkazów.
8. Guzik **n→P** - powodujący przesłanie zawartości n-tego miejsca pamięci /określonego zawartością części adresowej rejestru rozkazów/ do rejestru P.
9. Guzik **P→n** - powodujący przesłanie zawartości rejestru P do n-tego miejsca pamięci.
10. Guzik **P→R** - powodujący przesłanie zawartości rejestru P do rejestru rozkazów R.
11. Guzik **aR→LR** - powodujący przesłanie zawartości części adresowej rejestru R do licznika rozkazów LR.
12. **12** guzików powodujących wykonanie jednej z 12 operacji oznaczonych w liście rozkazów numerami od 6 do 17. Operacje te nie obejmują jednak przesyłania informacji na drodze Pamięć - rejestr P, a jedynie wykonanie działań na arytmometrze i przesyłania informacji na drodze Arytmometr - rejestr P.
13. **17** przełączników i 1 guzik pozwalający ręcznie nastawić rozkaz i załadować nim rejestr R.
14. **36** przełączników i jeden guzik pozwalający ręcznie załadować rejestr P dowolną zawartością.
15. Guzik **Zer oR** - powodujący wyzerowanie części operacyjnej rejestru R.
16. Guzik **Zer aR** - powodujący wyzerowanie rejestru R.
17. Guzik **Zer. P** - powodujący wyzerowanie rejestru P.
18. Przełącznik **Wł.-Wył.** - powodujący uruchomienie i zatrzymanie rejestru P.
19. Guzik **Lektura** - powodujący uruchomienie drogi lektury.
20. Guzik **Pisanie** - powodujący uruchomienie drogi pisania.

21. Guzik **Przepisz kartę** - powodujący przepisanie karty przez urządzenie zewnętrzne.
22. Oscyloskop **A M P** wskazujący stan rejestrów A M i P.
23. Oscyloskop **Rura n** wskazujący zawartość rury której numer nastawiony jest pięcioma przełącznikami /32 kombinacje/.
24. Lampki **R** - 17 neonówek wskazujących zawartość rejestru R.
25. Lampki **LR** - 11 neonówek wskazujących zawartość rejestru LR.
26. Lampki **Inf.** - 6 neonówek informujących o aktualnym stanie maszyny w czasie wykonywania programu.
27. Lampka **SnA** - wskazująca znak zawartości rejestru A.
28. Lampka **SnM** - wskazująca znak zawartości rejestru M.
29. Lampka **N** - wskazująca powstanie nadmiaru w rej. A.

Posługując się stolikiem operator może wprowadzić dane do maszyny, zmieniać przebieg programu, kontrolować poprawność działania maszyny itp.

## ZAM-2

([pl.wikipedia.org/wiki/zam-2](http://pl.wikipedia.org/wiki/zam-2))

ZAM-2 – polski komputer pierwszej generacji zbudowany na lampach, wersja produkcyjna polskiego komputera XYZ. Przeznaczony przede wszystkim do obliczeń numerycznych był także używany do przetwarzania danych np. w Towarzystwie Ubezpieczeń i Reasekuracji WARTA, a w NRD współpracował z maszynami analitycznymi. Wyprodukowano 12 szt. w wersjach: Alfa, Beta i Gamma. Twórcy ZAM-2 otrzymali w 1964 r. nagrodę państwową II stopnia.

### Zmiany w stosunku do XYZ

- Dodanie rejestru indeksowego i uzupełnienie listy rozkazów.
- Zastąpienie pamięci rtęciowej magnetostrykcyjną.
- Podwojenie pojemności pamięci bębnowej.
- Użycie zamkniętych szaf zamiast stojaków centrali telefonicznej.
- Od wersji gamma zastąpienie przekaźnikowej macierzy przełączającej głowice pamięci bębnowej tranzystorową (zastosowana w pozostałych przy naprawach).

### Dane techniczne

**Rodzina:** ZAM

### Organizacja:

Jedno-adresowy, dynamiczny komputer szeregowy o sterowaniu układowym  
Arytmetyka binarna, zapis liczb znak-moduł stało-pozycyjny, słowo maszynowe:  
36 bitów (długie słowo) lub 18 bitów (krótkie słowo):

- 18-bitowe liczby całkowite
- 36-bitowe liczby ułamkowe
- 18-bitowe słowo rozkazu zawierające pola:
  - 1-bitowy modyfikator rozkazu np. zmieniający długość operandu
  - 5-bitowy kod rozkazu
  - 12-bitowy adres
- 32 rozkazy arytmetyczne, logiczne i sterownia; brak rozkazów zmiennoprzecinkowych i działań na znakach

### Szybkość:

- Taktowanie: 405 kc (kilocykli, 405 kHz)
- Ok. 1000 dodawań na sekundę
- Ok. 300 mnożeń na sekundę

### Pamięć:

### Operacyjna:

- Pamięć magnetostrykcyjna na liniach opóźniających
- 1024 krótkich słów (2.25 KiB)
- Średni czas dostępu: 0,36 ms

### Bębnowa:

- Głowice stałe
- Pojemność 32 768 słów krótkich (128 ścieżek po 256 słów krótkich) - 72 KiB
- Średni czas dostępu zależny od wykonania macierzy przełączającej głowice:
  - przekaźnikowa: 40 ms
  - tranzystorowa: 20 ms

### Urządzenia zewnętrzne:

- Dalekopis o szybkości 7 znaków na sekundę
- Fotelektryczny czytnik pięciokanałowej taśmy o szybkości 300 znaków na sekundę
- Perforator pięciokanałowej taśmy o szybkości 30 znaków na sekundę
- W NRD czytnik kart ELLIOTT-B 42 wymieniony następnie na reproducer kart

**Rozkazy:** 32 rozkazy 18-bitowe

### Rejestry:

- Arytmometru: 36 bitowe na magnetostrykcyjnych liniach opóźniających
- Akumulator – 36 bity i bit nadmiaru
- Mnożnik – 36 bitów
- Rejestr pośredniczący
- 1 bitowe:
  - bit nadmiaru akumulatora
  - bit znaku akumulatora
  - bit znaku mnożnika
  - Układu sterowania:
    - licznik rozkazów
    - rejestr rozkazów
    - licznik powtórzeń dla rozkazów mnożenia, dzielenia i przesunięć
    - bit modyfikacji rozkazów

### Technologia:

- Ok. 850 lamp elektronowych, 6000 ostrzowych diod germanowych i 500 tranzystorów w pamięci bębnowej
- Pakiety montowane na płytkach tekstolitowych, montaż przestrzenny, lutowane połączenia między łączówkami.

### Budowa (moduły):

- Dwie szafy o wymiarach 2300 × 2000 × 400 mm (arytmometr, pamięć wewnętrzna i układ sterowania)
- Stolik operatora
- Pamięć bębnowa
- Stolik urządzenia wejściowego
- Stolik urządzenia wyjściowego
- Zasilacz
- Zajmowana przestrzeń: około 60 m<sup>2</sup>
- Całkowita waga: około 1.8 ton

**Zasilanie:** 3 fazy 380/220 V, 50 Hz, pobór mocy: około 12,5 KW

### **Chłodzenie:**

Szafy i inne urządzenia posiadały wentylatory, a komputer nie wymagał klimatyzacji, jedynie szafa z pamięcią wewnętrzną posiadała wentylatory sterowane przez termostat (pamięć wewnętrzna wymagała stałej temperatury).

### **Języki programowania**

SAS - makroassembler

SAKO - autokod zwany "Polskim Fortranem".

### **Produkcja**

Alfa — dwie sztuki w latach 1960-61 z pamięcią rtęciową wymienioną w czasie uruchamiania na magnetostrykcyjną. Sprzedane Oficerskiej Szkole Łączności w Zegrzu i Biurze Projektów Przemysłu Syntezy Chemicznej w Gliwicach.

Beta — dwie sztuki w latach 1961-1962 z pamięcią magnetostrykcyjną.

Gamma — osiem sztuk w latach 1962-1965, z czego 2 sztuki wyeksportowano do NRD. W pamięci bębnowej zastosowano tranzystory.

### **Eksploatacja**

Opłata za 1 godz. pracy komputera w 1966 wynosiła 800 zł (ok. 433 zł w 2017 po inflacji)

# Opis maszyny ZAM-2

Polska Akademia Nauk  
ZAKŁAD APARATÓW MATEMATYCZNYCH

prace

zakładu aparatów matematycznych p a n

Praca C 3



M A S Z Y N A ZAM-2  
O p i s m a s z y n y  
K O M P E N D I U M P R O G R A M O W A N I A W J Ę Z Y K U S A L

Maszyna ZAM 2 składa się z czterech podstawowych części:

- A) PAMIĘĆ
- B) ARYTMOMETR
- C) STEROWANIE
- D) URZĄDZENIA WEJŚCIA I WYJŚCIA

### **Pamięć, słowo, liczba, rozkaz**

Pamięć składa się z komórek zwanych miejscami pamięci. Każde miejsce ma swój numer, który jest adresem tego miejsca pamięci. Komórki pamięci mogą zawierać liczby lub rozkazy zapisane w systemie binarnym /w postaci ciągu zer 1 jedynek/.

Pamięć maszyny ZAM-2 składa się z:

- Pamięci wewnętrznej, mającej bezpośrednie połączenie z pozostałymi częściami maszyny,
- Pamięci pomocniczej, służącej do przechowywania informacji, które nie mogą się pomieścić w pamięci wewnętrznej. Pamięć pomocnicza posiada bezpośrednie połączenie jedynie z pamięcią wewnętrzną.

W maszynie ZAM-2 rolę pamięci pomocniczej spełnia bęben magnetyczny /pamięć bębnowa/.

Pamięć wewnętrzna maszyny składa się z:

- 512 długich miejsc pamięci w każdym z nich może być zapamiętany ciąg 36 cyfr binarnych /**bitów**/
- lub
- 1024 krótkich miejsc pamięci w każdym z nich może być zapamiętany ciąg 18 cyfr binarnych /**bitów**/.

Każde długie miejsce pamięci jest sumą dwóch krótkich miejsc pamięci. Długie miejsca pamięci mają numery parzyste: 0, 2, 4, 1020, 1022. Krótkie miejsca pamięci ponumerowane są kolejno od 0 do 1023.

Długie miejsce pamięci o numerze  $2n$  składa się z dwóch krótkich miejsc pamięci o numerach  $2n$  i  $2n+1$ .



Pamięć wewnętrzna podzielona jest na 32 części zwane rurami, z których każda zawiera 32 miejsca, krótkie lub 16 miejsc długich.

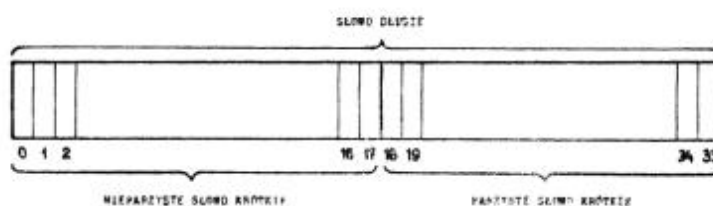
Rury ponumerowane są od 0 do 31, przy czym:

- rura 0 obejmuje krótkie miejsca pamięci 0 - 31
- rura 1 obejmuje krótkie miejsca pamięci 32 - 63
- rura 31 obejmuje krótkie miejsca pamięci 992 - 1023.

Ciągi 36 bitów zapamiętane w długich miejscach pamięci nazywają się słowami długimi maszyny.

Ciągi 18 bitów zapamiętane w krótkich miejscach pamięci nazywają się słowami krótkimi maszyny.

Słowo długie zawarte w komórce pamięci o numerze  $2n$  składa się ze słów krótkich zawartych w komórkach pamięci  $2n$  i  $2n+1$ .

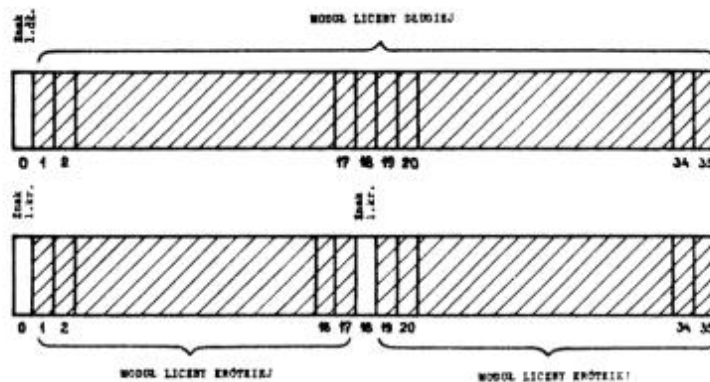


Rys. 1. Słowo długie

Ciągi bitów zawarte w komórkach pamięci mogą przedstawiać liczby lub rozkazy.

## Budowa liczby

Słowo /długie lub krótkie/ traktowane jako liczba składa się ze znaku liczby i modułu.



Rys. 2. Budowa liczby

Znak liczby zapisany jest w zerowym bicie słowa /jedynek oznacza minus, a zero plus/. Pozostałe bity są cyframi binarnym' modułu liczby; najwyższą pozycją modułu jest bit 1, a najniższą bit 35 /w przypadku liczby długiej/ lub bit 17 /w przypadku liczby krótkiej/.

Jeżeli w jednym słowie długim umieszczone są dwie liczby krótkie, to znak liczby krótkiej, umieszczonej w miejscu pamięci o adresie parzystym, jest 18-tym bitem słowa długiego o tym samym adresie, a sama liczba pokrywa się z niższymi pozycjami tego słowa długiego.

## Budowa rozkazu

Rozkaz zajmuje jedno słowo krótkie. Znaczenie poszczególnych pozycji binarnych rozkazu jest następujące:

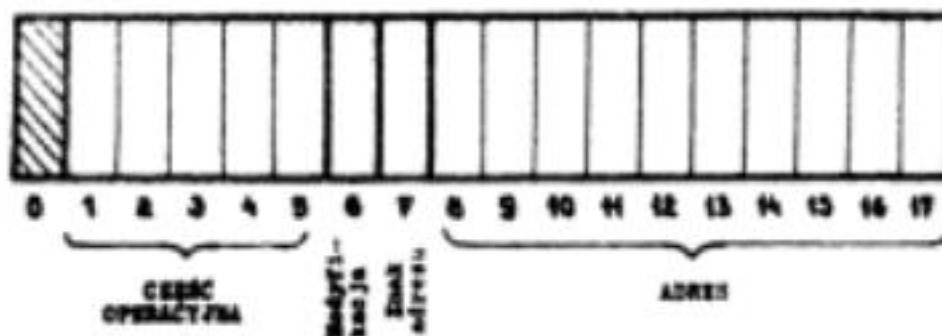
**BITY 1 - 5** - część operacyjna określa operację, która ma być wykonana.

**BIT 6** - bit modyfikacji wskazuje, czy rozkaz ma być przed wykonaniem modyfikowany /jedynek/, czy nie /zero/.

**BIT 7** - znak adresu wskazuje, czy adres rozkazu dotyczy słowa długiego /jedynek/, czy krótkiego /zero/.

**BITY 8 - 17** - adres rozkazu wskazuje numer miejsca pamieci, w której znajduje się, albo do której ma być przesłany argument rozkazu, bądź określa parametr rozkazu. Argumentem rozkazu może być dowolne słowo.

**BIT 0** - znak rozkazu, nie ma wpływu na treść rozkazu.



**Rys. 3. Rozkaz**

### **Modyfikacja rozkazów**

W maszynie ZAM-2 możliwa jest tzw. modyfikacja rozkazów. Sterowanie posiada specjalny rejestr modyfikujący, tzw. REJESTR B, którego zawartość jest dodawana do każdego modyfikowanego rozkazu przed jego wykonaniem. Dokładniej, proces wykonywania każdego rozkazu w maszynie ZAM -2 przebiega w sposób następujący:

Rozkaz zostaje pobrany do urządzenia sterującego, które w pierwszym rzędzie bada bit modyfikacji rozkazu /bit 6/.

- Jeżeli bit ten jest zerem, to rozkaz od razu zostaje wykonany.
- Jeżeli bit ten jest jedynką, to najpierw zostaje wykonana operacja modyfikacji tego rozkazu, po czym dopiero zmodyfikowany rozkaz zostaje wykonany. Operacja modyfikacji rozkazu polega na dodaniu do rozkazu zawartości rejestru B. Zarówno zawartość rejestru B, jak i rozkaz /niezależnie od jego znaku/ są przy tym traktowane jako liczby całkowite dodatnie, a dodawanie wykonuje się modulo  $2^{17}$ . Operacja modyfikacji nie zmienia zawartości rejestru B.

W pamięci maszyny rozkazy niczym nie różnią się od liczb. O tym, jak jest traktowane poszczególne słowo, decyduje fakt, dokąd to słowo zostaje pobrane. Jeżeli słowo krótkie zostaje pobrane do urządzenia sterującego, to jest ono traktowane jako rozkaz. Jeżeli dowolne słowo jest pobierane do arytmometru, to jest ono

traktowane jak liczba. W komunikacji między pamięcią wewnętrzną a urządzeniami wejścia i wyjścia oraz bębniem słowa traktowane są jako układy bitów.

Przy pobieraniu informacji z pamięci maszyny zawartość odpowiedniej komórki pamięci pozostaje niezmienną. Natomiast przy przesyłaniu informacji do pamięci poprzednia zawartość odpowiedniej komórki zostaje skasowana i na jej miejsce wstawiona nowa. Jest to ogólna właściwość pamięci, zarówno wewnętrznej, jak i bębnowej.

## **Arytmometr i Sterowanie**

Arytmometr i sterowanie stanowią część zasadniczą maszyny cyfrowej. Wykonuje się w nich całość obliczeń i interpretowane są rozkazy zawarte w programie. Praca maszyny polega na pobieraniu informacji z pamięci wewnętrznej i przetwarzaniu ich. W związku z tym arytmmometr i sterowanie muszą zawierać rejestry, do których te informacje są pobierane.

Poniżej podany jest opis podstawowych rejestrów arytmmometru i sterowania.

### **Rejestry arytmmometru**

Operacje arytmetyczne i logiczne wykonywane są na liczbach w części maszyny zwanej arytmmometrem. Zasadniczymi elementami arytmmometru są : rejestry /akumulator i mnożnik/ oraz układy wykonujące działania.

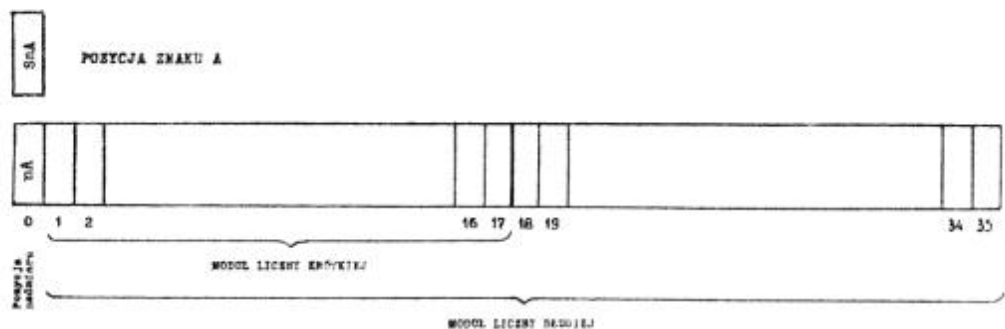
### **Akumulator, symbol A**

Akumulator jest to główny rejestr arytmmometru. Posiada on właściwości addytywne, tzn. można do jego zawartości dodawać lub odejmować od niej liczby z pamięci. W przypadku mnożenia umieszczana jest tu bardziej znacząca część iloczynu, a w przypadku dzielenia - reszta z dzielenia.

Rejestr akumulatora składa się z następujących pozycji:

pozycja znaku akumulatora - SnA  
pozycja nadmiaru akumulatora - nA  
35 pozycji modułu akumulatora - p1A, ..., P35A

Moduł akumulatora jako całość oznaczać będziemy symbolem mA.



Rys. 4. Akumulator

Przy przesyłaniu liczby do akumulatora poprzednia, zawartość akumulatora zostaje wyzerowana, po czym znak liczby zostaje przekazany do pozycji znaku akumulatora, a moduł wchodzi na pozycje 1-35 modułu akumulatora.

Do pozycji nadmiaru akumulatora zostaje wpisana cyfra zero. W pozycji nadmiaru może pojawić się jedynka w trakcie wykonywania pewnych działań arytmetycznych, gdy liczba będąca wynikiem któregoś z działań przekroczyła zakres maszyny. Przy wykonywaniu operacji arytmetycznych w akumulatorze pozycja nA traktowana jest jako przedłużenie /pozycja 0/ modułu A. Gdy mówimy o liczbie zapisanej w akumulatorze, to rozumiemy przez to liczbę, której znak umieszczony jest w SnA, a moduł w nA i mA.

Przy przesyłaniu liczby z akumulatora do pamięci cyfra z pozycji nadmiaru nie wchodzi do pamięci.

Przy przesyłaniu do akumulatora liczby krótkiej moduł tej liczby zajmuje pozycje 1-17 modułu akumulatora, a do pozostałych pozycji mA zostają wpisane zera. Ogólnie można powiedzieć, że liczba krótka w arytmmetrze jest traktowana tak jak liczba długa, której bity 18-35 są zerami.

## **Mnożnik, symbol M**

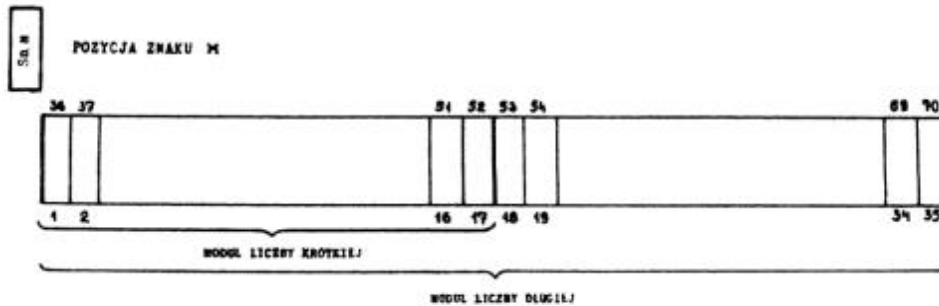
Mnożnik jest to drugi podstawowy rejestr arytmmetru.

Umieszcza się w nim jeden z czynników w przypadku wykonywania operacji mnożenia. Po wykonaniu mnożenia w mnożniku znajduje się mniej znacząca część iloczynu. W przypadku dzielenia przed wykonaniem operacji w mnożniku znajduje się mniej znacząca część dzielnej /bardziej znacząca jest w akumulatorze/, a po wykonaniu dzielenia zostaje tu umieszczony iloraz.

Rejestr mnożnika składa się z następujących pozycji:

- pozycja znaku mnożnika - SnM
- 35 pozycji modułu mnożnika - p1M, ..., p35M

Moduł mnożnika jako całość oznaczać będziemy symbolem mM.



Przy przesyłaniu liczby do mnożnika poprzednia jego zawartość zostaje wyzerowana, po czym znak liczby zostaje przesłany do SnM, a moduł wchodzi na pozycje 1-35 modułu mnożnika. Przy przesyłaniu do mnożnika liczby krótkiej moduł tej liczby zajmuje pozycje 1-17 modułu M, a do pozostałych pozycji mM zostają wpisane zera.

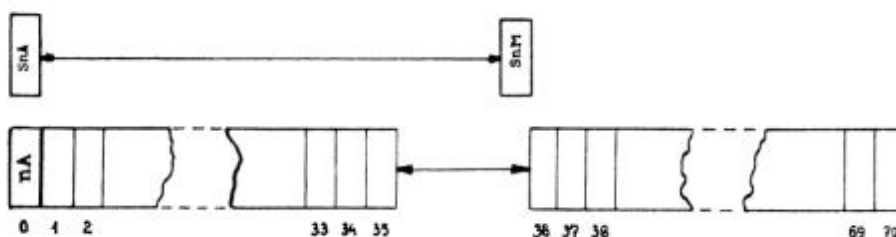
Jak wynika z powiedzianego wyżej, moduł mnożnika często jest traktowany jako przedłużenie akumulatora. Tak jest, gdy w rejestrach tych umieszczony jest iloczyn dwóch liczb lub dzielna /przed dzieleniem/. Również w przypadku operacji przesuwania arytmetycznego zawartości A i M rejestry te traktowane są jako jedna całość.

W powyższych przypadkach wygodnie jest posługiwać się pojęciem rejestru podwójnej długości AM.

Przez rejestr AM rozumiemy rejestr złożony z:

- pozycji znaku akumulatora                      - SnA
- pozycji modułu akumulatora                    - mA
- pozycji modułu mnożnika                      - mM.

Przez liczbę zapisaną w rejestrze AM rozumiemy liczbę, której znak zapisany jest w SnA, a moduł w nA, mA i mM, przy czym bardziej znacząca część modułu zapisana jest w mA, a mniej znacząca - w mM. Gdy mówimy o liczbie zapisanej w AM, to pozycje mM traktujemy jako pozycje 36-70, a pozycję nA jako pozycję 0 modułu liczby zapisanej w AM.



Rys. 6. Akumulator i mnożnik jako jedna całość

## **Rejestr nadmiaru, symbol N**

Rejestr nadmiaru jest to rejestr jednopoziomowy. Jego zawartość może więc być jedyneką lub zerem. Jedynka w rejestrze N wskazuje na to, że w trakcie wykonywania poprzednich obliczeń wynik co najmniej jednej operacji przekroczy/ zakres maszyny.

Zero w rejestrze N wskazuje, że przekroczenia zakresu maszyny nie było. Przekroczenie zakresu maszyny możliwe jest przy wykonywaniu w arytmometrze operacji dodawania, odejmowania, dzielenia i przesuwania w lewo arytmetycznie. Właściwość zerowania rejestru N posiada wyłącznie rozkaz 'Skocz przy Nadmiarze' /patrz niżej opis rozkazów maszyny ZAM-2/.

## **Rejestry sterowania**

Funkcją sterowania jest pobieranie z pamięci wewnętrznej maszyny rozkazów, ewentualne dokonywanie ich modyfikacji i zgodnie z ich treścią wysyłanie odpowiednich impulsów sterujących do pozostałych części maszyny.

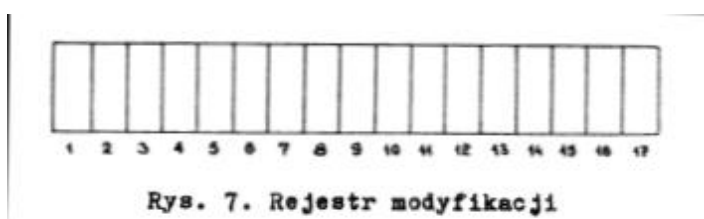
Sterowanie kieruje więc pracą całej maszyny zgodnie z informacjami zawartymi w rozkazach programu. Podstawowymi rejestrami sterowania są:

- rejestr modyfikacji /B/
- rejestr rozkazów /R/
- licznik rozkazów /LR/
- rejestr pomocniczy /RP/.

## **Rejestr modyfikacji, symbol B**

Do sterowania zalicza się rejestr B ze względu na jego funkcje przy modyfikacji rozkazów.

Rejestr B jest rejestrem 17-bitowym. Przy przesyłaniu słowa krótkiego do rejestru B pozycja znaku zostaje odcięta i do rejestru B wchodzi moduł słowa na pozycje 1-17. Przy przesyłaniu zawartości rejestru B do pamięci na miejsce znaku słowa wpisuje się w pamięci bit 0. Przy operacji modyfikacji zawartość rejestru B zostaje dodana do rozkazu, który ma być wykonany. Rejestr B posiada właściwości addytywne: można w nim dodawać i odejmować liczby krótkie, dzięki czemu można go traktować jako pomocniczy akumulator.



**Rys. 7. Rejestr modyfikacji**

## **Rejestr rozkazów, symbol R**

Rejestr rozkazów jest to rejestr 17-bitowy, w którym w czasie pracy maszyny umieszczony jest aktualnie wykonywany rozkaz. Znak rozkazu nie ma wpływu na jego treść i nie wchodzi do rejestru R.

## **Licznik rozkazów, symbol LR**

Licznik rozkazów jest to rejestr 10-bitowy. Jego zawartość wskazuje numer komórki pamięci, z której pobiera się rozkaz do wykonania. Po wykonaniu dowolnego rozkazu zawartość rejestru LR zmienia się tak, aby wskazać położenie w pamięci wewnętrznej rozkazu, który ma być wykonany jako następny. Najczęściej zmiana ta polega na automatycznym dodaniu do zawartości rejestru LR jedynek lub - w przypadku rozkazów sterujących na przesyłaniu do rejestru LR części adresowej rozkazu poprzedniego. Istnieją poza tym pewne specjalne rozkazy, które powodują inne zmiany zawartości rejestru LR.

W każdym jednak przypadku o zmianie zawartości LR decyduje ostatni wykonywany rozkaz.

## **Rejestr pomocniczy, symbol RP**

Rejestr pomocniczy, jest to rejestr 6-bitowy. Od jego zawartości zależy treść rozkazów zewnętrznych maszyny, tzn. rozkazów służących do przesyłania informacji między urządzeniami zewnętrznymi<sup>\*)</sup> a pamięcią wewnętrzną maszyny. Stan tego rejestru decyduje, do którego urządzenia zewnętrznego odnosić się będzie odpowiedni rozkaz zewnętrzny i jakiego typu przesyłanie będzie wykonane. Specjalny rozkaz służy do ładowania rejestru pomocniczego.



## **Arytmetyka Maszyny**

Maszyna ZAM-2 jest maszyną stałoprzecinkową.

Rozumie się przez to, że liczby, na których arytmometr wykonuje działania, nie zawierają informacji o umiejscowieniu przecinka.

Umiejscowienie przecinka w liczbie jest umowne. Przez skalę binarną liczby umieszczonej w rejestrze lub komórce pamięci będziemy rozumieli numer pozycji, po której występuje umowny przecinek.

Tak więc liczba w skali binarnej  $n$  jest to liczba, w której pozycja jedności umieszczona jest w  $n$ -tej pozycji modułu rejestru /ew. komórki pamięci/, zawierającego daną liczbę. Skala może być również liczbą ujemną lub większą od numeru ostatniej pozycji modułu liczby. Oznacza wówczas numer umownej pozycji, jaka powstałaby przy rozszerzeniu modułu liczby w odpowiednią stronę.

Ta sama liczba może być zapisana w postaci stałoprzecinkowej w różnej skali. Zmniejszanie skali powoduje zwiększanie dokładności obliczeń /o ile nie wystąpi przekroczenie zakresu maszyny/. Cechą charakterystyczną operacji na liczbach stałoprzecinkowych jest konieczność przyjęcia z góry pewnej ustalonej skali dla każdej liczby. Konieczna jest zatem analiza zakresu wielkości, występujących w czasie obliczeń jeszcze przed napisaniem odpowiedniego programu.

W arytmometrze działania wykonują się na liczbach 35-bitowych. Przy programowaniu obliczeń na liczbach w dowolnej skali należy mieć na uwadze zasady podane poniżej przy opisie poszczególnych operacji arytmetycznych.

Arytmometr maszyny ZAM-2 wykonuje następujące działania arytmetyczne i logiczne.

### ***Dodawanie, odejmowanie, odejmowanie bezwzględnych wartości***

Działania te wykonują się w akumulatorze. Do liczby zapisanej w akumulatorze można dodać /od liczby zapisanej w akumulatorze można odjąć/ dowolną liczbę zapisaną w pamięci wewnętrznej maszyny oraz od bezwzględnej wartości liczby zapisanej w akumulatorze można odjąć bezwzględną wartość liczby z dowolnej komórki pamięci wewnętrznej. Wynik każdego z tych działań zostaje umieszczony w akumulatorze w skali, równej skali składników. Działania powyższe wykonywane są z pełną dokładnością, o ile nie wystąpi przekroczenie zakresu maszyny. Przekroczenie zakresu maszyny /które jest sygnalizowane za pośrednictwem wskaźnika nadmiaru/ może wystąpić przy operacji dodawania lub odejmowania, natomiast

nie może wystąpić przy odejmowaniu bezwzględnych wartości. Powyższe operacje mają sens, jeżeli liczby, na których są wykonywane, zapisane są w jednakowej skali.

## **Mnożenie**

Mnożenie polega na tym, że liczba zapisana w mnożniku zostaje pomnożona przez liczbę z pamięci wewnętrznej, a wynik /podwójnej długości/ zostaje umieszczony w rejestrach A i M, które traktowane są w tym przypadku jako jeden rejestr AM, w skali równej sumie skal czynników. Znak iloczynu zostaje wpisany jednocześnie do pozycji  $S_nA$  i  $S_nM$ . A zatem w szczególnym przypadku:

- Jeżeli pomnożymy dwie liczby w skali 0, to w akumulatorze otrzymamy starszą część iloczynu również w skali 0.
- Jeżeli pomnożymy dwie liczby w skali 35 /liczby całkowite/, to w iloczynie pozycja jedności wypadnie na 35 pozycji mnożnika. Tak więc, jeżeli iloczyn nie przekroczy zakresu maszyny, będzie on umieszczony w rejestrze M jako liczba w skali 35.

Przy wykonywaniu operacji mnożenia nie może wystąpić sygnalizacja nadmiaru.

## **Dzielenie**

Przy operacji dzielenia dzielna jest liczba podwójnej długości, zapisana w rejestrze AM /znakiem tej liczby jest znak A/, a dzielnikiem - liczba zapisana w komórce pamięci. Jeżeli traktujemy dzielną jako zapisaną w skali  $S_1$ , a dzielnik - w skali  $S_2$ , to iloraz będzie obliczony z dokładnością do  $35 - S_1 + S_2$  znaków binarnych za przecinkiem i zapisany w rejestrze M w skali równej różnicy skal dzielnej i dzielnik  $S_1 - S_2$ . Reszta zostaje zapisana w akumulatorze w skali  $S_1 - 35$ . Znak reszty równa się znakowi dzielnej. Iloraz maszynowy jest zawsze przybliżeniem z niedomiarem odpowiedniego ilorazu arytmetycznego. W szczególnym przypadku:

- Jeżeli dzielimy liczby w skali 0, to wynik otrzymujemy w mnożniku w skali 0.
- Jeżeli chcemy podzielić dwie liczby całkowite w skali 35, to dzielną należy umieścić w mnożniku, a w akumulatorze - liczbę zero ze znakiem dzielnej /w ten sposób dzielna będzie miała skalę 70/. Iloraz otrzymamy w mnożniku jako liczbę całkowitą w skali 35.

Dzielenie wykonuje się prawidłowo w przypadku, gdy moduł zawartości akumulatora jest mniejszy od modułu dzielnika, traktując obie te liczby jako liczby w jednakowej skali. W przeciwnym przypadku dzielenie nie wykonuje się i sygnalizowane jest przekroczenie zakresu maszyny poprzez wskaźnik nadmiaru;

zawartości rejestru A i modułu M nie ulegają przy tym zmianie, zaś do pozycji SnM zostaje wpisane zero.

## **Przesunięcia arytmetyczne**

W arytmometrze mogą być wykonywane operacje przesuwania zawartości rejestrów A i M, które w tym przypadku traktowane są jako jeden rejestr AM. Arytmetycznie operacje te odpowiadają mnożeniu liczby podwójnej długości zapisanej w AM przez całkowitą potęgę dwójki. W przypadku przesuwania w prawo /mnożenie przez  $2^{-p}$ / znak mnożnika zostaje przyrównany do znaku akumulatora, znak A nie ulega zmianie. Przy przesuwaniu w lewo /mnożenie przez  $2^p$ / znak akumulatora zostaje przyrównany do znaku mnożnika, znak M nie ulega zmianie.

Przesunięcia arytmetyczne można traktować jako operacje zmieniające skalę liczb. Przy przesuwaniu o p pozycji w lewo skala liczby zmniejsza się o p, przy przesuwaniu w prawo - skala liczby analogicznie zwiększa się.

Przy przesuwaniu w lewo arytmetycznie o p pozycji sygnalizowany jest nadmiar, gdy którakolwiek z pozycji 1, ..., p modułu liczby przesuwanej jest jedyneką.

## **Przesunięcia cykliczne**

Istnieje możliwość przesuwania zawartości akumulatora w lewo cyklicznie. W takim przypadku znak akumulatora traktowany jest jako przedłużenie akumulatora /jego pozycja zerowa/. Pozycja nadmiaru akumulatora - nA nie bierze udziału w przesunięciu cyklicznym.

Przesuwanie cykliczne w lewo polega na tym, że z pozycji znaku bity przechodzą na ostatnią pozycję akumulatora, a z pozycji 1 akumulatora do znaku. Tak więc przy przesunięciu cyklicznym w lewo o p, p najstarszych bitów akumulatora wchodzi na p najmniej znaczących pozycji. Pierwotny bit znaku pojawia się zatem na pozycji 36 - p. W przypadku przesunięcia cyklicznego o 0 pozycji zawartość akumulatora nie ulega zmianie.

## **Zaokrąglenie**

Operacja zaokrąglania powoduje dodanie jedynki na 35-tej pozycji do modułu liczby zapisanej w akumulatorze, jeżeli w 1-szej pozycji mnożnika była jedynka, oraz wpisanie jedynki do 35-tej pozycji mnożnika. Jednocześnie liczba zaokrąglona zostaje przesłana do pamięci wewnętrznej W miejsce wskazane adresem rozkazu ZAOKRAGLIJ.

## **Koniunkcja**

Operacja koniunkcji na dwóch bitach daje wynik 1, Jeżeli oba, te bity są jedynkami, wynik 0 w przeciwnym przypadku.

$$0 \wedge 0 = 0$$

$$1 \wedge 0 = 0$$

$$0 \wedge 1 = 0$$

$$1 \wedge 1 = 1$$

W maszynie ZAM-2 operacja koniunkcji polega na wykonaniu koniunkcji na odpowiadających sobie bitach mnożnika i słowa z pamięci wewnętrznej. Wynik umieszczony zostaje w akumulatorze, zawartość mnożnika nie ulega zmianie. Operacji koniunkcji podlegają również znaki.

## **Alternatywa**

Operacja alternatywy na dwóch bitach daje wynik 1, jeżeli co najmniej jeden z nich jest jedynką. Jeżeli oba bity są zerami - wynik jest zerem.

$$0 \vee 0 = 0$$

$$1 \vee 0 = 1$$

$$0 \vee 1 = 1$$

$$1 \vee 1 = 1$$

W maszynie ZAM-2 operacja alternatywy polega na wykonaniu alternatywy na odpowiadających sobie bitach akumulatora i słowa z pamięci wewnętrznej. Wynik umieszczony zostaje w akumulatorze. Operacji alternatywy podlegają również znaki. Pozycja nadmiaru akumulatora nie ulega zmianie.

## **Właściwości zera**

W maszynie ZAM-2 liczba zero może występować w dwóch postaciach: ze znakiem plus lub ze znakiem minus. Jeżeli w wyniku działań dodawania lub odejmowania otrzymuje się w akumulatorze wynik zero, to znak akumulatora nie ulega zmianie /wynik zerowy otrzymuje znak składnika umieszczonego w akumulatorze/.

Jeżeli zero otrzymuje się w wyniku odejmowania bezwzględnych wartości, to jego znakiem będzie zawsze plus.

## Arytmetyka rejestru B

Rejestr B można traktować pod względem wykonywanych w nim operacji w sposób dwojaki:

1. Jako rejestr pozwalający na wykonywanie operacji dodawania i odejmowania na 17-bitowych liczbach całkowitych dodatnich. Wtedy pozycja 1 rejestru B jest traktowana jako najstarsza pozycja liczby. Przy takiej interpretacji wynikiem odejmowania liczby  $b$  od liczby  $a$ , przy założeniu, że  $b > a$ , będzie liczba:

$$2^{17} - (b-a)$$

2. Jako rejestr pozwalający na wykonywanie operacji dodawania i odejmowania na liczbach 16-bitowych ze znakiem, w arytmetyce uzupełnieniowej. W takim przypadku pozycja 1 rejestru B jest traktowana jako pozycja znaku /0 - plus, 1 - minus/. 16-bitowy moduł zajmuje pozycje 2-17, przy czym moduły liczb ujemnych zapisane są w postaci:

$$2^{16} - a$$

gdzie  $a > 0$  - moduł liczby ujemnej.

A więc np. liczba:

00000000000000010

w obu interpretacjach oznacza liczbę +2 natomiast, natomiast liczba:

11111111111111110

w pierwszej interpretacji oznacza liczbę  $2^{17}-2$ , a w drugiej interpretacji oznacza liczbę -2.

## Bęben Magnetyczny

Pamięć pomocniczą maszyny ZAM-2 stanowi bęben magnetyczny. Jego pojemność wynosi 16384 słów długich lub 32768 słów krótkich. Komórki pamięci bębnowej, z których każda może zawierać jedno słowo krótkie, ponumerowane są kolejno od 0 do 32767. Numery długich miejsc pamięci są parzyste. Długie miejsce pamięci bębnowej o numerze  $2n$  składa się z dwóch krótkich miejsc o numerach  $2n$  i  $2n + 1$ .

Numer komórki, w której znajduje się pewne słowo na bębnie, nazywa się adresem bębnowym tego słowa. Pamięć bębnowa składa się ze 128 ścieżek ponumerowanych od 0 do 127. Adresy bębnowe początków poszczególnych ścieżek są następujące:

0, 256, 512, ...,  $n \cdot 256$ , ..., 127256,

gdzie  $n$  - numer ścieżki.

Można przesyłać z bębna do pamięci wewnętrznej - i na odwrót - dowolne słowa długie oraz słowa krótkie jedynie z /lub do/ nieparzystych miejsc pamięci bębnowej. Przy przesyłaniu słowa krótkiego z pamięci wewnętrznej do miejsca  $2n + 1$  na bębnie miejsce  $2n$  na bębnie zostaje wyzerowane. Przesyłanie słów z pamięci wewnętrznej na bęben nazywa się pisanem na bęben, zaś przesyłanie w przeciwnym kierunku nazywa się czytaniem z bębna.

Pamięć bębnowa zaopatrzona jest w rejestr bębnowy symbol RB; zawartość jego wskazuje adres bębnowy skąd będzie czytane lub dokąd będzie zapisywane słowo.

Tak więc, jeśli chcemy zapisać coś na bębnie lub przeczytać jakieś słowo z bębna, należy najpierw do rejestru bębnowego przesłać odpowiedni adres bębnowy, a następnie dać rozkazy czytania względnie pisania.

Zawartość rejestru bębnowego po przeczytaniu jednego słowa z bębna lub po zapisaniu jednego słowa na bębnie automatycznie zwiększa się o 2. Jeżeli zatem chcemy przeczytać z bębna grupę słów umieszczonych w kolejnych długich komórkach pamięci, wystarczy raz załadować rejestr bębnowy adresem początkowym, a następnie wykonać odpowiednią ilość rozkazów czytania z bębna. Rejestr RB może zawierać jedynie adresy parzyste. Tym niemniej przy przesyłaniu liczb krótkich, gdy rejestr RB zawiera adres  $2n$ , słowo przesyłane będzie pobrane względnie wpisane do miejsca  $2n+1$  na bębnie.

Zarówno czytanie, jak i pisanie na bęben dokonuje się przy pomocy tego samego rozkazu PRZEPISZ. Rozkaz ten oznacza czytanie z bębna, jeżeli w rejestrze RP jest umieszczona liczba 16, i pisanie na bęben, jeżeli w rejestrze RP jest liczba 32.

Przesyłanie adresu do rejestru RB dokonuje się przy pomocy specjalnego rozkazu /Patrz - lista rozkazów/.

Wszelkie przesyłania z bębna do pamięci wewnętrznej podlegają tzw. kontroli parzystości. Kontrola ta polega, na tym, że przy zapisywaniu słów długich na bębnie każde słowo zostaje zaopatrzone w dodatkowy bit w

ten sposób, żeby suma bitów każdego słowa na bębnie była parzysta. Przy przesyłaniu z bębna do pamięci wewnętrznej specjalny układ kontroluje parzystość sumy bitów każdego słowa przesyłanego. W razie wykrycia nieparzystości maszyna staje, zapalając odpowiedni wskaźnik na stoliku operatora.

Bęben magnetyczny obraca się z szybkością ok. 1500 obrotów na minutę. W związku z tym średni czas dostępu do pamięci bębnowej wynosi ok. 20 ms.

Miejsca pamięci na ścieżkach ponumerowane są w kierunku przeciwnym do ruchu bębna, przy czym kolejne adresy mają co trzynaste słowa. Pozwala to, przy zastosowaniu odpowiedniej pętli programowej (\*) na przesyłanie informacji pomiędzy bębniem i pamięcią wewnętrzną maszyny z szybkością ok. 240 słów na sekundę.

## Urządzenia Wejścia i Wyjścia

Urządzenia wejścia i wyjścia pośredniczą we wprowadzaniu i wyprowadzaniu informacji do /z/ pamięci wewnętrznej maszyny.

### Taśma dziurkowana

W maszynie ZAM-2 urządzenia wejścia i wyjścia wykorzystują dziurkowaną taśmę papierową jako środek przekazywania informacji. Informacje na taśmie zapisywane są w postaci układów do 5-ciu dziurek zwanych rządkiem taśmy. Rządek taśmy przedstawia jeden z dopuszczalnych znaków dalekopisowych, których spis podany jest w Uzupełnieniach.

Znaki dalekopisowe podzielone są na dwie grupy: cyfry i litery. O tym, czy dany rządek na taśmie przedstawia literę, czy cyfrę, decyduje fakt, czy wśród poprzedzających go rzędków występował ostatnio znak CYFRY, czy znak LITERY.

Znak CYFRY powoduje, że wszystkie rządki występujące po nim aż do znaku LITERY przedstawiają znaki cyfrowe. Podobnie znak LITERY powoduje, że wszystkie rządki aż do znaku CYFRY przedstawiają znaki literowe.

Do przygotowywania taśmy dziurkowanej używa się specjalnych urządzeń dalekopisowych firmy CREED, przy pomocy których można dziurkować taśmę, jednocześnie otrzymując wydrukowany tekst tzw. tabulogram. Te same urządzenia służą do drukowania informacji wydziurkowanych na taśmie przez urządzenie wyjściowe maszyny.

### Wejście

Informacje wydziurkowane na taśmie są przekazywane do pamięci wewnętrznej maszyny za pośrednictwem czytnika firmy FERRANTI. Urządzenie to pozwala na czytanie taśmy z szybkością 300 rzędków na sekundę.

W maszynie ZAM-2 możliwe są dwa sposoby wprowadzania informacji z taśmy papierowej: tzw. czytanie rządkiem i czytanie binarne taśmy.

### Czytanie rządkiem

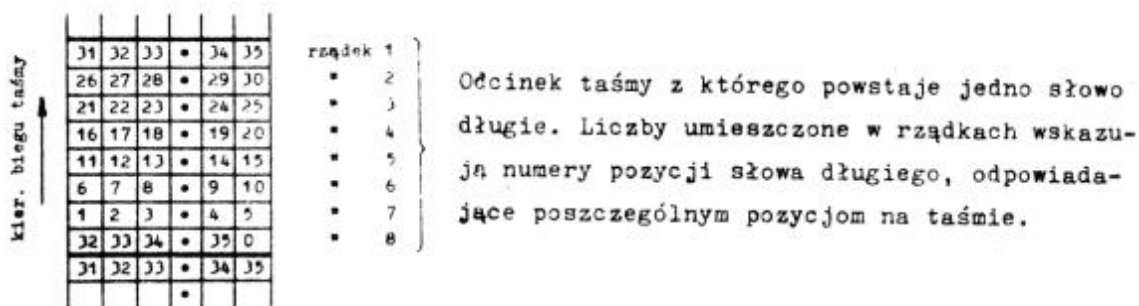
Czytanie rządkiem polega na przepisywaniu kolejnych rzędków taśmy do słów krótkich w pamięci wewnętrznej. Specjalny rozkaz czytania taśmy powoduje przepisanie rządka taśmy do pięciu ostatnich pozycji słowa krótkiego /pozycje 13-17/, wpisując do 6-tej od końca /12-tej/ pozycji 0, jeżeli rządek przedstawia cyfrę, i 1, jeżeli rządek przedstawia literę. Pozycje 0-11 zostają przy tym wyzerowane. Wiersz taśmy jest to ciąg rzędków, zawartych między dwoma kolejnymi znakami CR /Powrót Karetki/ z ostatnim znakiem CR włącznie.

Urządzenie czytające taśmę jest tak skonstruowane, że przy czytaniu rządkiem wiersz taśmy musi być czytany z pełną szybkością 300 rzędków na sekundę. Przerwanie czytania taśmy może nastąpić tylko bezpośrednio po przeczytaniu rządka CR, kończącego wiersz taśmy. Wynika stąd konieczność posługiwania się specjalną krótką pętlą programową przy wczytywaniu informacji do maszyny.



## Czytanie binarne

Podczas czytania binarnego z kolejnych 8 rzędków taśmy składane jest słowo długie, przesyłane następnie do pamięci wewnętrznej. Ilość słów długich, jaka zostanie wczytana do pamięci w efekcie wykonania specjalnego rozkazu czytania binarnego /rozkaz BT/, określona jest przez zawartość rejestru B. Mianowicie, po wczytaniu każdego słowa długiego zawartość rejestru B zmniejsza się o 2 i czytanie jest kontynuowane aż do wyzerowania B-rejestru. Kolejne słowa wczytywane umieszczane są w kolejnych długich miejscach pamięci, poczynając od adresu wskazanego przez rozkaz BT. Sposób składania słowa długiego z 8 rzędków taśmy przedstawia poniższy rysunek.



Słowo długie

Na pozycje 32, 33, 34, 35 słowa długiego zostaje wpisana suma logiczna 4-ech prawych pozycji rzędka 1 i 4-ech lewych pozycji rzędka 8 taśmy. W związku z tym 4 lewe pozycje 8 rzędka taśmy są zwykle zerami.

## Wyjście

Urządzeniem wyjściowym maszyny ZAM-2 jest REPERFORATOR CREEDA. Dziurkuje on taśmę z maksymalną szybkością 25 rzędków na sekundę. Tak wydziurkowana taśma wyjściowa przechodzi do urządzenia drukującego, które działa z szybkością 5-6 znaków na sekundę. Drukowanie informacji zawartych na taśmie jest niezależne od maszyny i może być wykonywane w innym miejscu i czasie.

Istnieją dwa rozkazy dziurkowania taśmy:

- Rozkaz dziurkowania rzędka - powoduje wydziurkowanie 5-ciu najmniej znaczących pozycji słowa krótkiego w rzędku taśmy.
- Rozkaz dziurkowania binarnego - powoduje dziurkowanie partii po 8 rzędków taśmy dla każdego słowa długiego. Sposób dzielenia słowa długiego na osiem rzędków jest odwrotnością opisanego powyżej składania słów długich w przypadku czytania binarnego. W przypadku binarnego dziurkowania taśmy najniższa pozycja rzędka 8 odpowiada pozycji 0 słowa długiego, a pozostałe pozycje tego rzędka są zerami /brak dziurek/.

## Lista Rozkazów Maszyny Zam-2

### OZNACZENIA

- A - rejestr akumulatora /lub krócej - akumulator/  
SnA - pozycja znaku akumulatora lub rejestru AM /patrz rej. AM/  
mA - pozycje modułu akumulatora od 1 do 35  
nA - pozycja nadmiaru akumulatora
- M - rejestr mnożnika /lub krócej - mnożnik/  
SnM - pozycja znaku mnożnika  
mM - pozycje modułu mnożnika od 1 do 35
- AM - rejestr podwójnej długości złożony z SnA, nA, mA i mM  
/pozycja znaku i 71 pozycji modułu/
- N - rejestr nadmiaru  
B- rejestr B złożony z 17 pozycji  
R - rejestr rozkazów złożony z 17 pozycji  
LR - rejestr licznika rozkazów /lub krócej - licznik rozkazów/  
złożony z 10 pozycji  
RB - rejestr bębnowy złożony z 14 pozycji  
RP - rejestr pomocniczy złożony z 6 pozycji  
WP - rejestr wskaźnika parzystości; 1 pozycyjowy  
KL - rejestr KLUCZY na stoliku  
W - rejestr wskaźników /informuj/ na stoliku  
n - numer miejsca pamięci wewnętrznej  
b - numer miejsca pamięci bębnowej  
t - rząddek taśmy  
p - parametr rozkazu /podawany w części adresowej/
- (X), (n); (b), itp. - zawartość rejestru lub odpowiedniej komórki pamięci  
P<sub>n</sub>X - n-ta pozycja rejestru X  
1<sub>n</sub> - słowo utworzone z samych zer i jedynek na n-tej pozycji

**Znak równości '='** - we wzorach opisujących funkcje rozkazów oznacza operację podstawiania: zawartość rejestru występującego z lewej strony znaku równości ma być przyrównana do wartości prawej strony

**Kropka '.'** - przed numerem miejsca pamięci w części adresowej rozkazu oznacza, że adres dotyczy słowa długiego \*)

**Numer operacji** - liczba odpowiadająca binarnemu kodowi części operacyjnej rozkazu.

-----

\*) Adres poprzedzony kropką powinien być parzysty; w przeciwnym wypadku kropka zostanie zignorowana - maszyna potraktuje adres rozkazu jako dotyczący słowa krótkiego.

Rozkazy maszyny zostały podzielone na grupy w zależności od ich treści.

Grupa A - Rozkazy dotyczące Arytmometru

Grupa B - Rozkazy B-Rejestru

Grupa S - Rozkazy Sterujące

Grupa Z - Rozkazy dotyczące Urządzeń zewnętrznych

Grupa SP - Rozkazy Specjalne

## **Rozkazy Dotyczące Arytmometru**

### **Grupa A**

Rozkazy dotyczące arytmometru zostały opisane w przypadku gdy adres dotyczy liczb długich. W przypadku liczb krótkich, przy pobieraniu z pamięci moduł liczby jest uzupełniany zerami na końcu do liczby długiej; natomiast przy przesyłaniu do pamięci moduł liczby krótkiej pobierany jest z pozycji 1 - 17 odpowiedniego rejestru.

#### **1. Umieść w Akumulatorze**

Zapis:                **UA n**

Funkcja:              $A = (n); \quad LR = LR + 1$

Numer operacji: 10

Przykłady:

UA 1000 : Umieść w A liczbę krótką z pamięci 1000

UA.302 : Umieść w A liczbę długą z pamięci 302

Treść:

Przepisz zawartość n-tego miejsca pamięci wewnętrznej do akumulatora w ten sposób, żeby zawartość zerowej pozycji n-tego miejsca pamięci wewnętrznej została przepisana do  $S_nA$ , a pozycje 1 - 35 n-tego miejsca pamięci wewnętrznej - kolejno do pozycji 1 - 35  $mA$ . Wpisz zero do  $nA$ .

M i N - bez zmian.

#### **2. Umieść w Mnożniku**

Zapis:                **UM n**

Funkcja:              $M = (n) ; \quad IKR = LR + 1$

Numer operacji: 11

Przykłady:

UM.12 : Umieść w M liczbę długą z pamięci 12  
UM 101 : Umieść w M liczbę krótką z pamięci 101

Treść:

Przepisz zawartość n-tego miejsca pamięci wewnętrznej do mnożnika w ten sposób, aby zawartość zerowej pozycji n-tego miejsca pamięci wewnętrznej została przepisana do  $S_nM$ , a pozycje 1 - 35 n-tego miejsca pamięci wewnętrznej - kolejno do pozycji 1 - 35 mM.  
A i N bez zmian.

### 3. Pamiętaj Akumulator

Zapis: **PA n**

Funkcja:  $(n) = A ; LR = LR + 1$

Numer operacji: 17

Przykłady:

PA 133 : Pamiętaj A w pamięci krótkiej 133  
PA.8 : Pamiętaj A w pamięci długiej 8

Treść :

Przepisz zawartość akumulatora do n-tego miejsca pamięci wewnętrznej w ten sposób, aby zawartość  $S_nA$  została przepisana do pozycji zerowej n-tego miejsca pamięci wewnętrznej, a pozycje 1 - 35 mA - kolejna do pozycji 1 - 35 n-tego miejsca pamięci wewnętrznej.  
A, M, i N - bez zmian.

### 4. Pamiętaj Mnożnik

Zapis: **PM n**

Funkcja:  $(n) = M ; LR = LR + 1$

Numer operacji: 18

Przykłady:

PM 1023 : Pamiętaj M w pamięci krótkiej 1 023  
PM.262 : Pamiętaj M w pamięci długiej 262

Treść:

Przepisz zawartość mnożnika do n-tego miejsca pamięci wewnętrznej w ten sposób, aby zawartość  $S_nM$  została przepisana do pozycji zerowej n-tego miejsca pamięci wewnętrznej, a pozycje 1 - 35 mM kolejno do pozycji 1 - 35 n-tego miejsca pamięci wewnętrznej.  
A, M i N - bez zmian.

## 5. Dodaj do akumulatora

Zapis:            **DO n**

Funkcja:            $A = A + (n) ; LR = LR + 1$

Ewentualna sygnalizacja nadmiaru.

Numer operacji: 24

Przykłady:

DO.256 : Dodaj do A liczbę długą z pamięci 256  
DO 1017 : Doda: do A liczbę krótką z pamięci 1017

Treść:

Dodaj do liczby umieszczonej w akumulatorze liczbę umieszczoną w n-tym miejscu pamięci wewnętrznej. Moduł sumy wpisz do mA i nA w skali składników sumy. Znak sumy wpisz do SnA. Wpisz do N jedynek, jeżeli w wyniku działania nastąpiła zmiana stanu nA. W przeciwnym razie - N bez zmian.  
M - bez zmian.

## 6. Odejmij od Akumulatora

Zapis:            **OD n**

Funkcja:            $A = A - (n) ; LR = LR + 1$

Ewentualna sygnalizacja nadmiaru.

Numer operacji: 25

Przykłady:

OD.350 : Odejmij od A liczbę długą z pamięci 350  
OD 121 : Odejmij od A liczbę krótką z pamięci 121

Treść:

Odejmij od liczby umieszczonej w akumulatorze liczbę umieszczoną w n-tym miejscu pamięci wewnętrznej. Moduł różnicy wpisz do mA i nA w skali odjemnej i odjemnika. Znak różnicy wpisz do SnA. Wpisz do N jedynek, jeżeli w wyniku działania nastąpi/a zmiana stanu nA. W przeciwnym razie - N bez zmian.  
M - bez zmian.

## 7. Odejmij Bezwzględnie

Zapis: **OB n**

Funkcja:  $A = |A| - |(n)|$  ;  $LR = LR + 1$

Przykłady:

OB 1 002 : Odejmij bezwzględnie liczbę krótką z 1 002  
OB.354 : Odejmij bezwzględnie liczbę długą z 354

Treść:

Odejmij od modułu liczby umieszczonej w akumulatorze moduł liczby umieszczonej w n-tym miejscu pamięci wewnętrznej. Moduł otrzymanej różnicy wpisz do mA i nA w skali odjemnej i odjemnika. Znak różnicy wpisz do SnA.  
M i N - bez zmian.

## 8. Mnóż

Zapis: **MN n**

Funkcja:  $AM = M \cdot (n)$  ;  $LR = LR + 1$

Nadmiar nie może wystąpić.

Numer operacji: 27

Przykłady:

MN 263 : Pomnóż M przez liczbę krótką z pamięci 263  
MN.400 : Pomnóż M przez liczbę długą z pamięci 400

Treść:

Pomnóż liczbę umieszczoną w mnożniku przez liczbę umieszczoną w n-tym miejscu pamięci wewnętrznej. Iloczyn wpisz do AM, to znaczy moduł iloczynu, w skali równej sumie skal czynników, wpisz do mA i mM /część bardziej znacząca do mA/, a znak iloczynu do SnA i SnM. Wpisz 0 do nA. N - bez zmian.

## 9. Dziel

Zapis: **DZ n**

Funkcja:  $M = AM : (n)$  ;  $A = \text{Reszta}$  ;  $LR = LR + 1$

Nadmiar jest sygnalizowany, gdy zawartość A przed dzieleniem jest  $\geq (n)$ .

Numer operacji: 28

Przykłady:

DZ.108 : Dziel przez liczbę długą z pamięci 108  
DZ 15 : Dziel przez liczbę krótką z pamięci 15

Treść:

Jeżeli moduł liczby umieszczonej w akumulatorze jest mniejszy od modułu liczby umieszczonej w n-tym miejscu pamięci wewnętrznej - traktując obie liczby jako zapisane w jednakowej skali - podziel liczbę umieszczoną w akumulatorze przez liczbę umieszczoną w n-tym miejscu pamięci wewnętrznej. Moduł ilorazu, w skali równej różnicy skal dzielnej i dzielnika, wpisz do mM, znak ilorazu - do SnM. Moduł reszty w skali s - 35 /gdzie s - skala dzielnej/ wpisz do mA. Zawartość SnA nie ulega zmianie i stanowi znak reszty.  
N - bez zmian.

W przeciwnym przypadku, gdy moduł liczby umieszczonej w akumulatorze jest nie większy od modułu liczby umieszczonej w n-tym miejscu pamięci wewnętrznej, wpisz jedynekę do N i zero do SnM i przejdź do wykonania następnego rozkazu.  
A i mM bez zmian - dzielenie nie wykona się.

## 10. Zaokrąglij

Zapis: **OK n**

Funkcja:  $(n) = A = A + SnA \cdot 1_{35} \cdot p_{1M} ; p_{35M} = 1 ; LR = LR + 1$

Ewentualna sygnalizacja nadmiaru.

Numer operacji: 20

Treść:

Wpisz jedynekę do 35 pozycji M. Jeżeli pierwsza pozycja mM zawiera jedynekę, dodaj do modułu liczby zawartej w akumulatorze jedynekę na 35 pozycji i wynik umieść, w nA i mA, pozostawiając SnA bez zmian. W przeciwnym wypadku - A bez zmian.  
Następnie przepisz liczbę zawartą w akumulatorze do n-tego miejsca pamięci wewnętrznej /sposób przepisania - patrz rozkaz PA/. Jeżeli zaokrąglenie powoduje zmianę stanu pozycji nA, wpisz jedynekę do N. W przeciwnym razie - N bez zmian.  
SnA i SnM - bez zmian.

## 11. Przesuń w lewo

Zapis: **LW p**

Funkcja:  $AM = AM \cdot 2^p ; SnA = SnM ; LR = LR +$

Ewentualna sygnalizacja nadmiaru

Numer operacji: 21

Przykłady:

LW 10 : Łączną zawartość A i M przesun o 10 w lewo  
LW 53 : Łączną zawartość A i M przesun o 53 w lewo  
LW 0 : Znak akumulatora uczyń równym znakowi mnożnika

Treść:

Łączną zawartość nA, mA i mM przesun o p miejsc w lewo arytmetycznie. Zawartość SnM przepisz do SnA. Jeżeli. którąkolwiek z pozycji 1, 2, ..p modułu AM przed wykonaniem rozkazu zawierała jedynekę, wpisz jedynekę do N. W przeciwnym przypadku - N bez zmian.

Wpisz zera do pozycji 70 - p + 1, 70 - p + 2, ..., 70 modułu AM.

Znak adresu nie wpływa na treść rozkazu. Rozkaz LW O powoduje przepisanie zawartości SnM do SnA. nA, mA i M bez zmian.

## 12. Przesun w prawo

Zapis: **PW p**

Funkcja:  $AM = AM \cdot 2^{-p}$  ;  $SnM = SnA$  ;  $LR = LR + 1$

Nadmiar nie może się pojawić.

Numer operacji: 22

Przykłady:

PW 17 : A i M przesun o 17 miejsc w prawo

PW 35 : A i M przesun o 35 miejsc w prawo

PW 0 : Znak mnożnika uczyn równym znakowi akumulatora

Treść:

Łączną zawartość nA, mA i mM przesun o p miejsc w prawo arytmetycznie. Zawartość SnA przepisz do SnM. Wpisz zera do pozycji nA oraz 1, 2, .. (p - 1) modułu AM. N - bez zmian.

Rozkaz PW O powoduje przepisanie zawartości SnA do SnM. A i mM bez zmian. Znak adresu nie ma wpływu na treść rozkazu.

## 13. Przesun, w lewo cyklicznie

Zapis: **LC p**

Funkcja:  $A = A * 2^p$  ;  $LR = LR + 1$

Nadmiar nie może się pojawić.

Operacja  $A * 2^p$  oznacza przesunięcie cykliczne akumulatora o p pozycji.

Numer operacji: 23

Przykłady:

LC 15 : Przesun A w lewo cyklicznie o 15

LC 0 : Nic nie rób



Treść:

Przesuń liczbę umieszczoną w akumulatorze w lewe cyklicznie u p pozycji, przy czym zawartość SnA je traktowana jako najwyższa pozycja mA. Parametr p powinien być mniejszy od 127.

M, nA i N - bez zmian.

Gdy parametr p jest większy od 127 nastąpi przesunięcie o wielkość P zawartą między 72 a 127.

Znak adresu nie ma wpływu na treść rozkazu.

#### 14. Koniunkcja

Zapis: **KO n**

Funkcja:  $A = M \cap (n)$  ;  $SnA = SnM \cap Sn n$  ;  $LR = LR + 1$

Nadmiar nie może się pojawić.

Numer operacji: 29

Przykłady:

KO.400 : Dokonaj koniunkcji mnożnika ze słowem długim 400

KO 993 : Dokonaj koniunkcji mnożnika ze słowem krótkim 993

Treść:

Wynik koniunkcji zawartości mM i zawartości pozycji 1 - 35 n-tego miejsca pamięci wewnętrznej wpisz do mA, a wynik koniunkcji zawartości Sr M i zawartości pozycji zerowej n-tego miejsca pamięci wewnętrznej wpisz do SnA. Wpisz 0 do nA.

N i M - bez zmian.

#### 15. Alternatywa

Zapis: **AL n**

Funkcja:  $A = A \cup (n)$  ;  $SnA = SnA \cup Sn n$  ;  $LR = LR + 1$

Nadmiar nie może się pojawić.

Numer operacji: 30

Przykłady:

AL.800 : Dokonaj alternatywy akumulatora ze słowem długim 800

AL 15 : Dokonaj alternatywy akumulatora ze słowem krótkim 15

Treść:

Wynik alternatywy zawartości mA i zawartości pozycji 1 - 35 n-tego miejsca pamięci wewnętrznej wpisz do mA, a wynik alternatywy zawartości SnA i zawartości zerowej pozycji n-tego miejsca pamięci wpisz do SnA.

M, N i nA - bez zmian.

## **Rozkazy Dotyczące Rejestru B**

### **Grupa B**

#### **1. Umieść w rejestrze B**

Zapis: **UB n**

Funkcja:  $B = (n) ; LR = LR + 1$

Numer operacji: 12

Przykłady:

UB 235 : Umieść w P moduł liczby krótkiej z pamięci 235

UB 15 : Umieść w B moduł liczby krótkiej z pamięci 15

Treść:

Przepisz do B moduł liczby umieszczonej w n-tym krótkim miejscu pamięci wewnętrznej. Adres rozkazu winien dotyczyć miejsca krótkiego.

Gdy adres dotyczy miejsca długiego, do B wpisuje się liczba umieszczona w nieparzystym miejscu krótkim wchodzącym w skład danego miejsca długiego, np. UB.14 zostanie wykonane jak UB 15.

#### **2. Pamiętaj rejestr B**

Zapis: **PB n**

Funkcja:  $(n) = B ; LR = LR + 1$

Przykłady:

PB 117 : Pamiętaj B w pamięci krótkiej 117

PB.80 : Pamiętaj B w pamięci kr. 81 i pamięci kr. 80

Treść:

Przepisz do pozycji 1 - 17 n-tego krótkiego miejsca pamięci wewnętrznej liczbę umieszczoną w B, wpisując 0 do pozycji zerowego miejsca pamięci. Adres rozkazu winien dotyczyć miejsca krótkiego. Gdy adres dotyczy miejsca długiego pamięci wewnętrznej, zawartość B zostanie przepisana do obu miejsc krótkich, składających się na miejsce długie.

### 3. Dodaj do rejestru B

Zapis: **DB n**

Funkcja:  $B = B + (n) ; LR = LR + 1$

Numer operacji: 13

Przykłady:

DB 13 : Dodaj do B moduł liczby krótkiej z pamięci 13

DB 88 : Dodaj do B moduł liczby krótkiej z pamięci 88

Treść:

Do liczby umieszczonej w B dodaj moduł liczby umieszczonej w n-tym krótkim miejscu pamięci wewnętrznej i sumę wpisz do B. Adres rozkazu winien dotyczyć miejsca krótkiego.

Gdy adres tego rozkazu dotyczy miejsca długiego, do liczby umieszczonej w rejestrze B zostaną dodane kolejno moduły liczb umieszczonych w dwu miejscach krótkich, z których składa się miejsce długie. Np. rozkaz DB.88 jest równoważny parze rozkazów

DB 88

DB 89

### 4. Odejmij od rejestru B

Zapis: **BB n**

Funkcja:  $B = B - (n) ; LR = LR + 1$

Numer operacji: 14

Przykłady:

BB 14 : Odejmij od B moduł słowa krótkiego z pamięci 14

BB 90 : Odejmij od B moduł słowa krótkiego z pamięci 90

Treść:

Od liczby umieszczonej w B odejmij moduł liczby umieszczonej w n-tym krótkim miejscu pamięci wewnętrznej. Różnicę wpisz do B. Gdy adres tego rozkazu dotyczy miejsca długiego, od modułu liczby umieszczonej w B zostaną odjęte kolejno moduły liczb umieszczonych w dwu miejscach krótkich, z których składa się miejsce długie. Np. rozkaz BB.90 jest równoważny parze rozkazów

BB 90

BB 91

## **Rozkazy Sterujące**

### **Grupa S**

#### **1. Stop i Skocz**

Zapis: **SS n**

Funkcja: STOP ; LR- n

Numer operacji: 0

Przykłady:

SS 992

SS 0

Treść:

Zatrzymaj maszynę. Po naciśnięciu klucza START przejdź do wykonywania rozkazu z n-tej komórki pamięci wewnętrznej.

#### **2. Wykonaj Rozkaz**

Zapis: **WR n**

Funkcja: R = (n)

Numer operacji: 1

Przykłady:

WR 15 : Wykonaj rozkaz z 15-ej komórki pamięci

WR 1010 : Wykonaj rozkaz z 1010-ej komórki pamięci

Treść:

Wykonaj rozkaz zawarty w n-tej komórce pamięci wewnętrznej.  
Po wykonaniu tego rozkazu przejdź do wykonywania rozkazu wskazanego przez zawartość rejestru LR.

#### **3. Skocz**

Zapis: **SK n**

Funkcja: LR = n

Przykłady:

SK 300

SK 45

Treść:

Przejdź do wykonania rozkazu zawartego w n-tej komórce pamięci wewnętrznej.

#### 4. Skocz przy Zerze

Zapis: **SZ n**

Funkcja:           Jeżeli  $A = +/- 0$  to  $LR = n$   
                  Jeżeli  $A \neq +/- 0$  to  $LR = LR$

Numer operacji: 4

Przykłady:

SZ 15  
SZ 83

Treść:

Jeżeli mA zawiera zero, przejdź do wykonywania rozkazu umieszczonego w n-tym miejscu pamięci wewnętrznej. W przeciwnym przypadku przejdź do wykonywania rozkazu umieszczonego w kolejnym miejscu pamięci. Zawartości nA i SnA nie są brane pod uwagę przy wykonywaniu rozkazu SZ.

#### 5. Skocz przy Plusie

Zapis: **SP n**

Funkcja:

Jeżeli  $SnA = 0$  to  $LR = n$   
Jeżeli  $SnA = 1$  to  $LR = LR + 1$

Numer operacji: 5

Przykłady:

SP 18  
SP 127

Treść:

Jeżeli SnA zawiera 1, przejdź do wykonywania rozkazu następnego. W przeciwnym przypadku przejdź do wykonywania rozkazu zawartego w n-tym miejscu pamięci wewnętrznej.

## 6. Skocz przy Nadmiarze

Zapis:           **SN n**

Funkcja:

Jeżeli  $N = 1$  to  $LR = n$  i  $N = 0$

Jeżeli  $N = 0$  to  $LR = LR + 1$

Numer operacji: 6

Przykłady:

SN 100

SN 415

Treść:

Jeżeli zawartość N jest jedyneką, umieść zero w N i przejdź do wykonywania rozkazu umieszczonego w n-tym miejscu pamięci wewnętrznej. W przeciwnym przypadku przejdź do wykonywania rozkazu umieszczonego w kolejnym miejscu pamięci.

## 7. Skocz przy Wskaźniku Parzystości

Zapis:           **SW n**

Funkcja:

Jeżeli  $WP = 1$  to  $LR = n$

Jeżeli  $WP = 0$  to  $LR = LR + 1$

Numer operacji: 7

Przykłady:

SW 5

SW 348

Treść:

Jeżeli rejestr WP zawiera jedynekę, umieść zero w rejestrze WP i przejdź do wykonywania rozkazu umieszczonego w n-tym miejscu pamięci. W przeciwnym przypadku przejdź do wykonywania rozkazu umieszczonego w kolejnym miejscu pamięci.

## 8. Porównaj

Zapis: **SB n**

Funkcja:

Jeżeli  $B = (n)$  to  $LR = LR + 3$

Jeżeli  $B \neq (n)$  to  $LR = LR + 1$

Numer operacji: 8

Przykłady:

SB 300

SB 415

Treść:

Jeżeli zawartość rejestru B równa się modułowi zawartości n-tego krótkiego miejsca pamięci, przeskocz dwa rozkazy. W przeciwnym przypadku przejdź do wykonywania rozkazu umieszczonego w kolejnym miejscu pamięci. Adres tego rozkazu winien dotyczyć krótkiego miejsca pamięci wewnętrznej; w przeciwnym przypadku nastąpi porównanie zawartości rejestru B z pozycjami 1 - 17 odpowiedniego słowa długiego. Efekt porównania - jak wyżej.

## Rozkazy Dotyczące Urządzeń Zewnętrznych

Grupa Z

### 1. Przygotuj

Zapis: **PG**

Funkcja:  $RP = p$  ;  $LR = LR + 1$

Numer operacji: 2

Przykłady:

PG 1 : Przygotuj czytanie taśmy  
PG 16 : Przygotuj czytanie z bębna

Treść:

Do rejestru RP przepisz zawartość sześciu najniższych pozycji części adresowej rozkazu PG. Powoduje to przygotowanie maszyny do wykonania jednej z czynności wykonywanych przez rozkaz PP lub BT /patrz niżej/.

### 2. Ustaw Bęben

Zapis: **RB n**

Funkcja:  $RBa(n)$  ;  $LR = LR +$

Numer operacji: 15

Przykłady:

RB 20 : Umieść w RB zawartość pamięci krótkiej 20  
RB 81 : Umieść w RB zawartość pamięci krótkiej 81

Treść:

Przepisz do RB zawartość pozycji 3 m 16 n-tego krótkiego miejsca pamięci wewnętrznej. Adres winien dotyczyć miejsca krótkiego.

### 3. Przepisz

Zapis: **PP n**

Funkcja:

Jeżeli  $RP = 1$  i  $t \neq CR, FS, LS, ER$  to  $(n) = t$ ,  $B = B + 1$  i  $LR = LR + 1$   
Jeżeli  $RP = 1$  i  $t = FS, LS, ER$  to  $(n) = t$  i  $LR = LR + 1$



Jeżeli  $RP = 1$  i  $t = CR$  to  $(n) = t$ ,  $B = B + 1$  i  $LR = LR + 3$   
Jeżeli  $RP = 2$  to  $t = (n)$  i  $LR = LR + 1$   
Jeżeli  $RP = 4$  to  $(n) = KL$  i  $LR = LR + 1$   
Jeżeli  $RP = 8$  to  $W n$  i  $LR = LR + 1$   
Jeżeli  $RP = 16$  to  $(n) = (b)$ ,  $RB = RB + 2$ ,  $B = B + 2$  i  $LR = LR + 1$   
gdzie  $b = (RB)$   
Jeżeli  $RP = 32$  to  $(b) = (n)$ ,  $RB = RB + 2$ ,  $B = B + 2$  i  $LR = LR + 1$

Numer operacji: 16

Przykłady :

PP 385  
PP 400

Treść:

Treść tego rozkazu zależy od stanu rejestru RP. I tak:

- Jeżeli zawartością RP jest 1 -

czytaj rządęk taśmy, tzn.

uruchom urządzenie czytające i najbliższy rządęk taśmy przepisz do pięciu ostatnich pozycji n-tego słowa krótkiego. Na szóstą od końca pozycję wstaw zawartość wskaźnika Z. Wpisz zera do pozostałych pozycji tego słowa krótkiego. Po przeczytaniu znaku CYFRY (FS) wpisz 0 do wskaźnika Z a po przeczytaniu znaku LITERY (LS) wpisz 1 do wskaźnika Z. Przy pozostałych znakach wskaźnik Z nie ulega zmianie.

Jeżeli wczytany rządęk nie jest jednym ze znaków CYFRY (FS), LITERY (LS), BUD - zwiększ zawartość rejestru B o 1.

W przeciwnym przypadku rejestr B pozostaw bez zmian.

Jeżeli wczytany rządęk nie jest znakiem CR przejdź do wykonania kolejnego rozkazu ( $LR = LR + 1$ ). Po wczytaniu znaku CR zatrzymaj taśmę i przeskocz 2 rozkazy ( $LR = LR + 3$ )\*.

Po wczytaniu każdego rządęka o parzystej ilości dziurek /poza rządękiem CR/ zostaje wpisana Jedyńska do rejestru parzystości WP.

- Jeżeli zawartością RP jest 2 -

dziurkuj rządęk taśmy, tzn.

zawartość 5 najmniej znaczących pozycji n-tego słowa krótkiego z pamięci wewnętrznej wydziurkuj na taśmie. Przesuń taśmę w reperforatorze o 1 rządęk. Jeżeli adres rozkazu PP dotyczy miejsca długiego - wydziurkowanych zostanie 5 najmniej znaczących pozycji nieparzystej części odpowiedniego słowa długiego.

-----

\*)Czytanie rządęków taśmy występujących pomiędzy kolejnymi znakami CR musi się odbywać z „pełną szybkością taśmy”. W związku z tym do czytania taśmy należy używać specjalnej pętli programowej.

- Jeżeli zawartością RP jest 4 -

czytaj klucze, tzn.

prześlij zawartość rejestru kluczy /klucze  $k_1$ / na stoliku do n-tego miejsca pamięci wewnętrznej. /Klucze po lewej stronie stanowią pozycje bardziej znaczące/.

Jeżeli część adresowa rozkazu PP wskazuje pamięć krótką, to informacje pobierane są z lewej części rejestru kluczy.

- Jeżeli zawartością RP jest 8 -

informuj, tzn.

zapal układ lampek 'informuj' na stoliku zgodnie ze stanem pięciu najmniej znaczących pozycji części adresowej rozkazu PP.

- Jeżeli zawartością RP jest 16 -

czytaj z bębna, tzn.

zawartość b-tego miejsca pamięci bębnowej, gdzie b jest zawartością RB, prześlij do n-tego miejsca pamięci wewnętrznej. Następnie zawartość rejestru RB zwiększ o 2 i dodaj 2 do rejestru B.

Zawartość rejestru RB jest zawsze parzysta /patrz rozkaz RB/.

Jeżeli część adresowa rozkazu PP wskazuje słowo długie, to z bębna pobrane będzie słowo długie z b-tego długiego miejsca pamięci bębnowej. Jeżeli zaś rozkaz PP dotyczy słowa krótkiego, słowo to jest pobierane z (b+1)-szego krótkiego miejsca na bębnie.

- Jeżeli zawartością RP jest 32 -

pisz na bębnie, tzn.

zawartość n-tego miejsca pamięci wewnętrznej prześlij do b-tego miejsca pamięci bębnowej, gdzie b jest zawartością rejestru RB.

Następnie zawartość rejestru RB zwiększ o 2 i dodaj 2 do rejestru B.

Jeżeli adres rozkazu PP dotyczy słowa krótkiego, to słowo to zostaje przesłane do (b+1)-szego krótkiego miejsca na bębnie, zaś b-te krótkie miejsce pamięci bębnowej zostaje wyzerowane.

#### 4. Działaj na Binarnej Taśmie

Zapis: **BT n**

Numer operacji: 9

Treść:

Treść tego rozkazu zależy od stanu rejestru RP. I tak:

- Jeżeli zawartością RP jest 1 -

czytaj binarną taśmę, tzn.

przeczytaj  $((B) + 2)/2$  grup po 8 rzędów taśmy, tworząc z każdej grupy słowo długie zgodnie z opisanymi zasadami. Kolejne słowa długie zapamiętaj w kolejnych długich miejscach pamięci wewnętrznej poczynając od n. Po wczytaniu taśmy wpisz 0 do rejestru B i skocz do 1-szego miejsca pamięci.

Przykłady:

BT.24, przy  $B = 0$  : przeczytaj 8 najbliższych rzędów taśmy tworząc z nich słowo długie, zapamiętaj je w 24-tym długim miejscu pamięci.

BT.88, przy  $B = 4$  : przeczytaj 3 razy po 8 rzędów taśmy tworząc z nich 3 słowa długie, które zapamiętaj kolejno w 88, 90 i 92 długich miejscach pamięci.

- Jeżeli zawartością RP jest 2 -

dziurkuj binarnie taśmę, tzn.

wydziurkuj binarnie  $(B)/2$  słów długich umieszczonych w miejscach pamięci wewnętrznej:  $n, n+2, \dots, n+(B)-2$ . Każde słowo zostaje wydziurkowane w 8 rzędkach taśmy w sposób opisany poprzednio. Po wydziurkowaniu taśmy wpisz 0 do rejestru B i skocz do 1-szego miejsca pamięci wewnętrznej.

Przykłady:

BT.24, przy  $B = 2$  : wydziurkuj w 8-iu najbliższych rzędkach taśmy słowo długie spod adresu 24

BT.88, przy  $B = 4$  : wydziurkuj dwie grupy po 8 rzędów taśmy odpowiadające kolejno słowom długim spod adresów 88 i 90 pamięci wewnętrznej.

#### Uwaga:

W obu przypadkach /RP - 1,2/ adres rozkazu BT musi dotyczyć słowa długiego oraz zawartość rejestru B przed wykonaniem tego rozkazu musi być parzysta.

## **Rozkazy Specjalne**

### **1. Nic nie rób**

Zapis: **NI**

Funkcja:  $LR = LR + 1$

Numer operacji: 31

Przykład:

NI : Część adresowa nie ma wpływu na treść rozkazu.

Treść:

Przejdź do wykonania rozkazu umieszczonego w kolejnym miejscu pamięci.

Część adresowa rozkazu NI nie ma wpływu na jego działanie.

## Konkretne problemy rozwiązane na maszynie ZAM-2

W ciągu próbnej pracy w IMM PAN rozwiązanych zostało na maszynie ZAM-2 wiele problemów naukowych, technicznych i ekonomicznych. Poczyniono również pewne próby przystosowania maszyny ZAM-2 do zastosowań administracyjnych. Zastosowania ekonomiczne głównie sprowadzają się do rozwiązań zadań z programowania liniowego. Maszyna ZAM-2 wykonuje tego typu obliczenia głównie dla Ministerstwa Handlu Wewnętrznego i dla potrzeb budownictwa.

Klasycznym i typowym przykładem zadania z dziedziny programowania liniowego jest obliczenie najekonomiczniejszego przewozu piasku z  $n$  źródeł na  $m$  budów.

Zastosowania naukowe to obliczenia związane z rozpraszaniem nukleonów. Doświadczenia z tej dziedziny prowadzi się w najbardziej zaawansowanych badaniach jądra atomowego. Na maszynie ZAM-2 dokonano również wielu obliczeń typu geodezyjnego, jak np. rozwiązywanie układów równań liniowych wyrównujących sieć geodezyjną metodą najmniejszych kwadratów. Inny problem, to wyznaczenie najbardziej ekonomicznego rozkładu obciążeń dla poszczególnych elektrowni pracujących w jednym systemie mocy.

Typowym zastosowaniem maszyny cyfrowej jest statystyczne opracowywanie wyników eksperymentów. Przedmiotem badań w tym wypadku może być np. rozrzut parametrów tranzystorów albo określenie standardowych odchyłek próbek zbóż lub nasion. Tego typu problemy były wielokrotnie opracowywane na maszynie ZAM-2.

Technicznym zastosowaniem maszyny było badanie charakteru drgań występujących w różnorodnych układach elektrycznych lub mechanicznych, np. w generatorach mocy, lub w kratownicach. Na maszynie ZAM-2 wyznaczane były również optymalne kształty łopatek turbin.

Z obliczeń prowadzonych na ZAM-2 dla potrzeb przemysłu warto wspomnieć o badaniach procesów tematyczna analiza tego zagadnienia bez użycia elektronicznej maszyny cyfrowej stanowi nadzwyczaj żmudny i pracochłonny problem. Na ZAM-2 obliczono również najkorzystniejsze zestawy soczewek optycznych w obiektywach. Dla potrzeb przemysłu chemicznego określono parametry mieszanin gazowych w różnych warunkach ciśnienia i temperatury.

Ostatnio przystąpiono do prac nad przystosowaniem maszyny ZAM-2 do zastosowań administracyjnych. Ponieważ zastosowania te charakteryzują się mniejszą różnorodnością obliczeń, natomiast większą ilością danych wejściowych i wyników, maszyna ZAM-2 przystosowana do zastosowań administracyjnych jest wyposażona w dwa bębny pamięci magnetycznej, dwa czytniki oraz dwie drukarki. Pierwszym ukończonym już problemem z tej dziedziny jest prowadzenie ewidencji materiałowej magazynu przez maszynę ZAM-2. Również maszynę ZAM-2 zastosowano do prób opracowania wyciągów ewidencji ludności dla celów Stołecznej Rady Narodowej. Wszystkie te zastosowania podane są ilustracyjnie i nie wyczerpują możliwości maszyny, która jako maszyna uniwersalna może być stosowana do obliczania szerokiego wachlarza problemów.

## ZAM-3

([pl.wikipedia.org/wiki/ZAM-3](http://pl.wikipedia.org/wiki/ZAM-3))

**Typ:** komputer równoległy do przetwarzania danych

ZAM-3 (ZAM-3M) – doświadczalny komputer IMM budowany w latach 1961–64 (wstępne uruchomienie - 1963, publiczne - sobota 4 styczeń 1964); rozwinięcie technologii EMAL-2.

Zarzucony ze względu na małą niezawodność i skomplikowaną budowę.

**Rodzina:** ZAM

Bezlampowy komputer z pogranicza I. i II. generacji, zbudowany na ferrytowo-diodowych układach logicznych

**Organizacja:**

arytmetyka binarna, zapis liczb znak-moduł  
słowo maszynowe długości 24-bitów

**Prędkość:**

ponad 10 000 (14 000) dodawań na sekundę  
cykl odczytu pamięci operacyjnej: 10  $\mu$ s

**Pamięć operacyjna:**

ferrytowa na rdzeniach o średnicy 2 mm, 24-bitowa + bit parzystości, 8 Kilosłów

**Technologia:**

ponad 100 000 ostrzowych diod germanowych  
pakiety montowane na jednostronnych płytkach drukowanych.  
8 szaf  
wyprodukowano: 1 szt.

**Języki programowania:**

PJP (assembler przeznaczony dla twórców oprogramowania)  
COBOL.

**Urządzenia wejścia-wyjścia:**

monitor – dalekopis  
czytnik taśmy pięciokanałowej  
perforator taśmy pięciokanałowej  
czytnik kart dziurkowanych firmy ELLIOTT  
pamięć taśmowa PT-1 (taśma 1/2 cala, 8 ścieżek, gęstość zapisu 4 b/mm): 1 szt.  
pamięć bębnowa PB-3 o pojemności 32 Ksłów: 2 szt.

## **RODZINA ZAM 11,21,31,41,52**

([pl.wikipedia.org/wiki/ZAM-21](http://pl.wikipedia.org/wiki/ZAM-21))

ZAM-21 (Alfa) – doświadczalny komputer IMM rozwijany i budowany równolegle z ZAM-3, a produkowany w Elwro w latach 1961–1965.

### **Rodzina:** ZAM

Typ: tranzystorowy komputer drugiej generacji zbudowany na selekcjonowanych germanowych tranzystorach stopowych produkowanych w Tewie (pakiety typu S-400).

### **Organizacja:**

arytmetyka binarna, zapis liczb znak-moduł  
słowo maszynowe długości 24 bitów

### **Prędkość:**

ponad 30 000 dodawań na sekundę  
cykl odczytu pamięci operacyjnej: 10  $\mu$ s

### **Pamięć operacyjna:**

ferrytowa na rdzeniach o średnicy 2 mm, 24-bitowa + bit parzystości 4-8 kilostów.  
Szafy jednostki centralnej i pamięci bębnowej o wymiarach 1700x680x680 mm

**Wyprodukowano:** 3 szt.

### **Urządzenia we-wy:**

- monitor - dalekopis
- czytnik taśmy pięciokanałowej
- perforator taśmy pięciokanałowej
- czytnik kart dziurkowanych firmy ELLIOTT
- drukarka wierszowa DW-1, a później DW-2

### **Pamięć masowa:**

- pamięć taśmowa PT-2
- pamięć bębnowa PB-5 o pojemności 32 kstów

### **Zachowany:**

Muzeum Techniki w Warszawie.

([pl.wikipedia.org/wiki/ZAM-41](http://pl.wikipedia.org/wiki/ZAM-41))

ZAM 41 – pierwszy polski komputer do przetwarzania danych zaprojektowany w Instytucie Maszyn Matematycznych (IMM) w Warszawie, a produkowane przez Zakład Doświadczalny Instytutu.

Był jedynym produkowanym seryjnie modelem rodziny komputerów do przetwarzania danych opracowanym przez IMM na polecenie rządu z roku 1961. Prace nad pozostałymi:

minikomputerem ZAM-11 i największym ZAM-51 nie zostały ukończone. Wykonano jedynie 2 lub 3 maszyny prototypowe ZAM-21.

Komputer często umieszczano w dwóch albo trzech pomieszczeniach rozdzielonych przeszklonymi ściankami. Wydzielano pomieszczenie dla wymagających stałej temperatury i wyższej czystości pamięci taśmowych. W oddzielnym pomieszczeniu umieszczona była bardzo hałaśliwa drukarka wierszowa.

### **Rodzina:** ZAM

Typ: komputer II generacji zbudowany na selekcyonowanych germanowych tranzystorach stopowych TG1 produkowanych w Tewie (pakiety typu S-400)

### **Organizacja:**

arytmetyka binarna, zapis liczb znak-moduł

słowo maszynowe długości 24 bitów podzielone na 4 znaki po 6 bitów i 3 znaki po 8 bitów.

### **Pamięć operacyjna:**

ferrytowa na rdzeniach o średnicy 2 mm, 24-bitowa + bit parzystości od 8 do 256 Ksłów (w modułach po 8 Ksłów)

32 k adresowane bezpośrednio, a pozostałe pośrednio lub z zastosowaniem B-modyfikacji (rejestr indeksowego)

Szafy jednostki centralnej i pamięci bębnowej o wymiarach 1700x680x680 mm[5] wyprodukowano łącznie: 16 szt.

### **Liczby:**

- krótkie – 24 bity
- długie – 48 bitów
- zmiennoprzecinkowe – 48 bitów
- rozkazy jednoadresowe z 15-bitowym adresem
- zestaw znaków nie zawierał małych liter
- programowy zmienny przecinek realizowany przez ekstrakody (brak rozkazów
- zmiennoprzecinkowych realizowanych sprzętowo)

### **Prędkość:**

- ponad 30 000 rozkazów stałoprzecinkowych na sekundę
- cykl odczytu pamięci operacyjnej: 6 lub 10  $\mu$ s[4]
- czas dostępu: 3,5 lub 6  $\mu$ s

### **Urządzenia we-wy:**

- monitor – dalekopis
- czytnik taśmy pięciokanałowej
- perforator taśmy pięciokanałowej
- czytnik kart dziurkowanych



- drukarka wierszowa.

### **Pamięć masowa:**

- pamięć taśmowa – jednostki pamięci PT-2 z jednostką sterującą
- od 1 do 4 modułów z pamięci bębnowej PB-5 o pojemności 32 Ksłów każdy (przy
- pojemności 64 Ksłów ok. 42 Ksłów przeznaczonych dla programu)

Z jednym komputerem był używany prototyp pamięci bębnowej PB-6 o pojemności 20 mln bitów (ok. 800 Ksłów).

### **Wieloprogramowość:**

2 programy w trybie wsadowym; rzadko wykorzystywana.  
(KB) Teoretyczna 4 programy

### **Oprogramowanie**

#### **Systemy operacyjne:**

- SO (system operacyjny): SO 41, SO 141 (wsadowy), SO 241
- TRAN – czasu rzeczywistego.

#### **Języki programowania:**

- PJES – Podstawowy Język Etykiet Symbolicznych
- PJEG (assembler przeznaczony dla twórców oprogramowania)
- JOM – Język Operacyjny Maszyny
- SAS 41 (makroassembler)

#### Algorytmiczne:

- SAKO
- Algol

#### Do przetwarzania danych:

- EOL – język do przetwarzania informacji tekstowej
- COBOL

#### Symulacyjne:

- CEMMA – język do symulacji procesów ciągłych
- ZAM-GPSS – język do symulacji procesów dyskretnych
- Astek – język opisu i obróbki statystycznej

#### Przewidziane w projekcie lecz niezrealizowane:

FORTTRAN IV.

14056/02

187256

II

RODZINA MASZYN

MATEMATYCZNYCH

ZAM

---

LEON LUKASZEWICZ

informacje wstępne o rodzinie  
maszyn matematycznych zam

---

INSTYTUT MASZYN MATEMATYCZNYCH

# Wstęp

Rodzina Maszyn ZAM składa się z pięciu typów maszyn o różnym przeznaczeniu, wielkości i cenie lecz o jednolitym systemie programowania i jednolitej bazie podzespołów i modułów technicznych.

Maszyny rodziny ZAM odznaczają się prostą logiczną budową, dużą sprawnością działania oraz wysoką niezawodnością pracy. Każda z nich może wykonywać jednocześnie kilka niezależnych programów.

Maszyny rodziny ZAM mogą być wydajnie i ekonomicznie stosowane do:

- Przetwarzania Danych,
- Obliczeń Naukowych i Technicznych,
- Sterowania Procesami

Modułowa budowa maszyn ZAM pozwala na elastyczne dobieranie różnych zestawów maszyny oraz łatwą ich rozbudowę.

Opracowanie elektronicznych maszyn matematycznych i ich produkcja jest zadaniem trudnym i kosztownym. Wymaga ono wielkiego wkładu pracy wysoko kwalifikowanej kadry specjalistów oraz złożonej aparatury niezbędnej do prowadzenia badań naukowych, oraz do oprzyrządowania produkcji. Opracowanie programów dla wyprodukowanych maszyn wymaga również znacznego wysiłku zarówno od konstruktorów, jak i użytkowników maszyn.

Aby problem ten rozwiązać możliwie ekonomicznie, opracowano w Instytucie Maszyn Matematycznych koncepcję rodziny maszyn ZAM, która składa się z pięciu typów maszyn o różnym przeznaczeniu, lecz opartych na tych samych standardowych podzespołach technicznych oraz posiadających jednolity system programowania. W ten sposób starano się zrealizować ideę wspólnych nakładów dla rozwiązania kilku zadań. Pramodelem rodziny ZAM jest mała maszyna lampowa ZAM-2, wyprodukowana w ilości dwunastu egzemplarzy przez Instytut Maszyn Matematycznych. Maszyna ta zaopatrzona jest w system automatycznego kodowania SAKO. Produkcja, instalowanie u użytkowników i eksploatacja tych maszyn przyniosła Instytutowi bogate doświadczenie. W międzyczasie opracowano w Instytucie wiele nowych rozwiązań technicznych jak nowoczesna technika tranzystorowa, pamięci ferrytowe, bębnowe i taśmowe, elektronika urządzeń wejścia i wyjścia oraz wiele innych.

Rozpoczęte zostały pierwsze w kraju prace w dziedzinie przetwarzania danych, na razie jeszcze przy wykorzystaniu maszyn ZAM-2, które jednak przyniosły szereg ważnych doświadczeń. Opracowane zostały również projekty nowych języków automatycznego programowania, przeznaczonych zarówno do obliczeń numerycznych, jak i do przetwarzania danych. Dzięki połączeniu doświadczeń i wysiłków kadry naukowej, technicznej i wykonawczej Instytutu stało się obecnie możliwe zaprojektowanie całej rodziny maszyn ZAM, spełniającej, jak się wydaje, najważniejsze postulaty wymogów ekonomicznych i rozwoju perspektywnego, jakie można postawić maszynom krajowym.

Przy opracowaniu maszyn ZAM wykorzystano też szereg rozwiązań konstrukcyjnych Wrocławskich Zakładów Elektronicznych „Elwro”, które przewidziane są jako podstawowy producent tych maszyn.

## Zastosowania maszyn rodziny ZAM

Maszyny matematyczne rodziny ZAM odznaczają się bardzo dużą uniwersalnością zastosowań. Przykłady takich zastosowań są następujące:

Przetwarzanie Danych:

- Planowanie produkcji zmierzające do możliwie efektywnego wykorzystania parku maszynowego, operatywna kontrola i korekta tego planu, rejestracja kosztów z rozbiciem na poszczególne zamówienia.
- Planowanie zaopatrzenia, ewidencja materiałowa, automatyczne drukowanie zamówień.
- Sporządzanie list wypłat.
- Księgowość bankowa, ubezpieczeniowa, budżetowa; analizy i statystyka ekonomiczna.
- Prowadzenie dużych i złożonych ewidencji.

Obliczenia Naukowe i Techniczne:

- Odwracanie macierzy wysokiego rzędu.
- Całkowanie równań różniczkowych zwyczajnych i cząstkowych.
- Modelowanie systemów przy użyciu metod statystycznych.
- Obliczenia geodezyjne.

Sterowanie Procesami w czasie realnym:

- Centralna rejestracja i redukcja danych.
- Sterowanie procesami chemicznymi lub metalurgicznymi.
- Automatyczna dyspozycja mocy w systemach energetycznych.
- Automatyczna nawigacja.

Maszyny rodziny ZAM mogą współpracować wzajemnie ze sobą, ewentualnie za pośrednictwem torów transmisji danych.

## Typy maszyn ZAM

Rozróżniamy pięć następujących modeli maszyn rodziny ZAM:

**ZAM 51** Najpełniej rozbudowany model maszyny o następujących właściwościach technicznych:

- Automatyczne przesyłanie bloku słów pomiędzy pamięcią ferrytową a urządzeniami wejścia i wyjścia lub pamięciami pomocniczymi, na przykład pamięcią na taśmach magnetycznych. W czasie tego przesyłania centralna część maszyny może jednocześnie wykonywać obliczenia.
- Operacje arytmetyczne zmiennoprzecinkowe wykonywane są automatycznie za pomocą rozkazów wbudowanych o bardzo dużej szybkości działania.

Maszyny ZAM 51 są to maszyny dużych rozmiarów o wysokiej sprawności działania zarówno w obliczeniach naukowych i technicznych, jak i w przetwarzaniu masowych danych, występujących na przykład przy automatyzacji czynności zarządzania. Maszyny ZAM 51 są szczególnie przydatne w centralnych ośrodkach obliczeniowych, obsługujących jednocześnie instytuty naukowo-badawcze oraz wszelkiego typu instytucje gospodarcze.

**ZAM 41** Model o następujących właściwościach technicznych:

- Automatyczne przesyłanie bloku słów pomiędzy pamięcią ferrytową a urządzeniami wejścia i wyjścia lub pamięciami pomocniczymi, podobnie jak w maszynie ZAM 51.
- Operacje arytmetyczne zmiennoprzecinkowe wykonywane są automatycznie przy pomocy rozkazów programowanych, które są powolniejsze od rozkazów wbudowanych lecz za to pozwalają na znacznie oszczędniejsze rozwiązanie arytmetru maszyny.

Maszyny ZAM 41 są to maszyny średnich rozmiarów, przeznaczone w szczególności dla ośrodków obliczeniowych, nastawionych na przetwarzanie masowych danych.

**ZAM 31** Model o następujących właściwościach technicznych:

- Programowe przesyłanie bloku słów pomiędzy pamięcią ferrytową a urządzeniami wejścia i wyjścia. System ten jest powolniejszy od przesyłania automatycznego, lecz pozwala na znaczne oszczędności środków technicznych.
- Operacje arytmetyczne zmiennoprzecinkowe wykonywane są automatycznie za pomocą rozkazów wbudowanych.

Maszyny ZAM 31 są to maszyny średnich rozmiarów przeznaczone w szczególności dla ośrodków obliczeniowych, nastawionych na obliczenia naukowe i techniczne.

**ZAM 21** Model o następujących właściwościach:

- Programowe przesyłanie bloku słów pomiędzy pamięcią ferrytową a urządzeniami wejścia i wyjścia.
- Operacje zmiennoprzecinkowe wykonywane są automatycznie za pomocą rozkazów programowanych.

Maszyny ZAM21 są to maszyny małych rozmiarów lecz o uniwersalnym przeznaczeniu. Dość duże możliwości ZAM 21 w zakresie obliczeń naukowych i technicznych mogą zaspokoić potrzeby wielu instytutów i biur konstrukcyjnych. Bardzo duża szybkość działania na słowach 24-bitowych czyni je szczególnie przydatnymi do zagadnień sterowania, wymagających dużej szybkości działania.

**ZAM 11** Model o najdalej idącej ekonomii wykonania.

Większość operacji w tych maszynach wykonywana jest jako rozkazy programowane. Maszyny ZAM 11 mogą być stosowane do obliczeń naukowych i technicznych oraz do zagadnień automatyki, na przykład do sterowania procesami technologicznymi.

## **Modułowa konstrukcja maszyn rodziny ZAM**

Wszystkie maszyny rodziny ZAM mają wysoce elastyczną strukturę, pozwalającą na dobieranie różnych zestawów modułów w zależności od jej przeznaczenia.

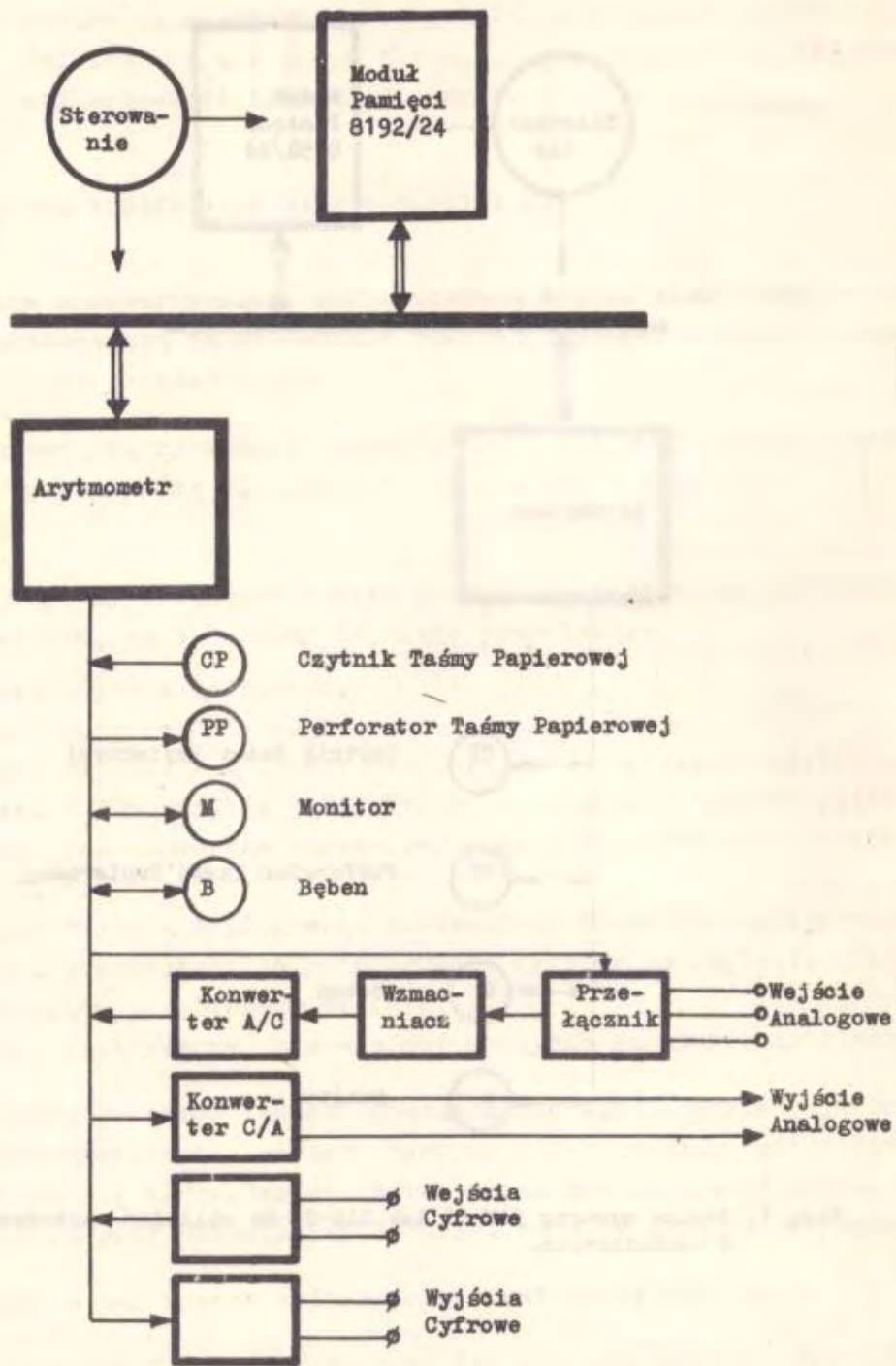
Przykładowo, na rysunku 1 przedstawiony jest mały zestaw maszyny ZAM 21 przeznaczony do obliczeń naukowo-technicznych.

Jeśli powyższy zestaw ma służyć ponadto do sterowania obiektem przemysłowym, to dołączamy do niego przykładowo:

- Kanał Wejścia Cyfrowego.
- Kanał Wyjścia Cyfrowego.
- Kanał Wejścia Analogowego zawierający przełącznik wybierający jeden z 200 punktów pomiarowych, wzmacniacz sygnałów wejściowych oraz konwerter napięciowy analogowo-cyfrowy o dokładności 0,1%.
- Kanał Wyjścia Analogowego zawierający konwerter cyfrowo-analogowy, przekształcający informacje cyfrowe na napięcie stałe, sterujące przebiegiem procesu.
- Zegar elektroniczny, pozwalający maszynie na odmierzanie czasu.

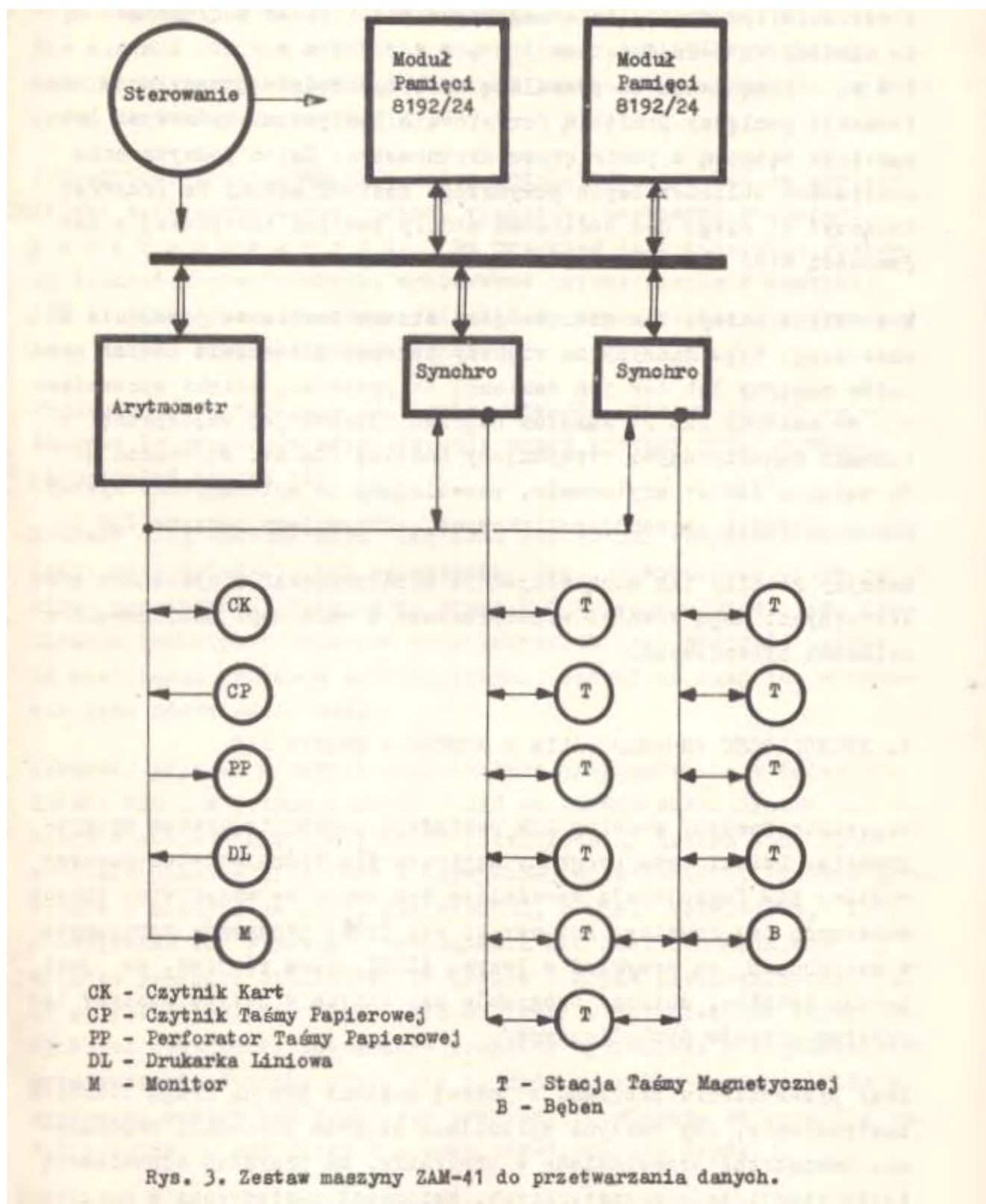
W przypadku procesów technologicznych niewymagających zbyt dużej szybkości sterowania, zamiast maszyny ZAM 21 możemy zastosować ZAM 11, co się sprowadza do zastosowania bardziej ekonomicznego arytmetru przy pozostawieniu innych modułów maszyny bez zmiany.

Otrzymany w ten sposób zestaw przedstawiono na rysunku 2.



Rys. 2. Zestaw maszyny ZAM-11 lub ZAM-21 do sterowania procesem przemysłowym.

Średniej wielkości zestaw maszyny ZAM 41, przeznaczony do przetwarzanie danych przedstawiony jest na rysunku 3.





W zestawie tym stacje taśm magnetycznych i bęben podłączone są do maszyny za pośrednictwem specjalnych kanałów. Urządzenia te pozwalają na bezpośrednie przesyłanie informacji pomiędzy pamięcią ferrytową a pamięciami taśmowymi lub pamięcią bębnową z pominięciem arytmometru. Celem podwyższenia możliwości obliczeniowych powyższego zestawu możemy na przykład dołączyć do niego dwa dodatkowe moduły pamięci ferrytowej o pojemności 8192 słowa każdy.

W rodzinie maszyn ZAM możliwe jest stosunkowo łatwe przejście z mniejszego typu maszyny na większy poprzez dołączenie nowych zespołów maszyny lub też ich zamianę. Na przykład, dzięki wprowadzeniu do maszyny ZAM 21 kanałów usprawniających jej współpracę z taśmami magnetycznymi otrzymujemy maszynę ZAM 41. Wprowadzając do maszyny ZAM 41 arytmometr, pozwalający na automatyczne wykonywanie operacji zmiennoprzecinkowych, otrzymujemy maszynę ZAM 51.

Maszyny rodziny ZAM mogą wzajemnie współpracować w systemach wielokrotnych. Mogą również współpracować z maszynami analogowymi w układach hybrydowych.

## Jednolitość programowania w rodzinie maszyn ZAM

Wszystkie maszyny rodziny ZAM posiadają jednolity system programowania. Dzięki temu programy napisane dla którejkolwiek maszyny rodziny ZAM funkcjonują zasadniczo bez zmian we wszystkich innych maszynach tej rodziny. Dotyczy to nie tylko programów napisanych w autokodach, na przykład w języku ALGOL, lecz również, co jest bardzo istotne, dotyczy programów napisanych w języku maszyny SAS (System Adresów Symbolicznych).

Przy przenoszeniu programu z jednej maszyny ZAM na drugą istnieje zastrzeżenie, aby maszyna wykonująca program posiadała wyposażenie zewnętrzne przewidziane w programie, na przykład odpowiednią ilość stacji taśm magnetycznych. Natomiast zastosowana w maszynach ZAM automatyczna segmentacja programów sprawia, że program napisany dla maszyny z bardzo dużą pamięcią wewnętrzną może być często bez żadnych zmian dostosowany do maszyny o pamięci wewnętrznej mniejszych rozmiarów. W tym ostatnim przypadku czas wykonania programu jest oczywiście dłuższy.

Jednolitość programowania w całej rodzinie maszyn ZAM osiągnięto dzięki zastosowaniu specjalnych rozkazów, nazwanych rozkazami programowanymi. Na przykład dość kosztowne rozkazy działań arytmetycznych, wykonywane automatycznie w maszynie ZAM 31, zastąpione są w maszynie ZAM 21 przez rozkazy programowane o ściśle równoważnym działaniu.

Podprogramy określające działanie rozkazów programowanych są dołączane do programów automatycznie przez systemy programowania, na przykład system SAS.

Rozkazy programowane mają taką samą postać jak wszystkie inne rozkazy, lecz działanie ich ograniczone jest do wywołania odpowiedniego podprogramu, wskazanego przez kod rozkazu. Dzięki temu realizacja techniczna rozkazów programowanych jest znacznie tańsza od realizacji rozkazów automatycznych, natomiast czas ich wykonania jest odpowiednio dłuższy.

Korzyści płynące z jednolitego systemu programowania w całej rodzinie ZAM i w języku maszyny – SAS są bardzo duże. System ten zawiera w sobie wszystkie rozkazy całej rodziny maszyn ZAM, co umożliwia pisanie programów o najwyższej efektywności. Pisanie programów w autokodach jest, jak wiadomo, o wiele łatwiejsze, lecz efektywność ich w pewnych przypadkach nie jest zadowalająca. Z tych to przyczyn bibliotekę programów o dużej powtarzalności, jak na przykład programy odwracania macierzy w obliczeniach numerycznych lub codziennie powtarzane programy wydawnicze w zagadnieniach przetwarzania danych, pisze się na ogół w języku maszyny. Otóż w przypadku maszyn ZAM tego typu programy obowiązują od razu dla całej rodziny, a nie tylko dla poszczególnych jej typów.

W języku maszyny pisanych jest również wiele Programów Wyspecjalizowanych, translatorów dla autokodów oraz programów dla Systemów Operacyjnych. Są to często programy bardzo trudne do opracowania, zawierające po kilkanaście lub kilkadziesiąt tysięcy rozkazów. I tu również jednolitość programowania w systemie SAS daje możliwość wykorzystania raz napisanych programów jednocześnie dla całej rodziny ZAM.

Jednolitość programowania całej rodziny różnego typu maszyn posiada w wielu konkretnych sytuacjach ogromne znaczenie. Rozpatrzmy dwa przykłady.

## **Rozwój ośrodka obliczeniowego w ramach jednego przedsiębiorstwa.**

Założmy, że pewne przedsiębiorstwo organizuje ośrodek obliczeniowy, dla którego zakupuje stosunkowo niedrogą maszynę ZAM 21. Po pewnym okresie eksploatacji i po wyszkoleniu personelu, kierownictwo przedsiębiorstwa może uznać, że wydajność tej maszyny jest już niewystarczająca. Zamiana maszyny ZAM 21 na maszynę o innym systemie programowania spowodowałaby potrzebę tworzenia nowej biblioteki programów w języku maszyny oraz odpowiedniego przekwalifikowania programistów pracujących przy maszynie.

Natomiast przy przejściu z maszyny ZAM 21 do maszyny ZAM 41 lub ZAM 51, wszystkie napisane dotychczas programy mogą być praktycznie bez żadnych zmian używane na nowej maszynie. Jednolitość rozwiązań technicznych wszystkich maszyn rodziny ZAM powoduje, że również personel konserwacyjno-techniczny maszyny musi być tylko w nieznacznym stopniu doszkolony.

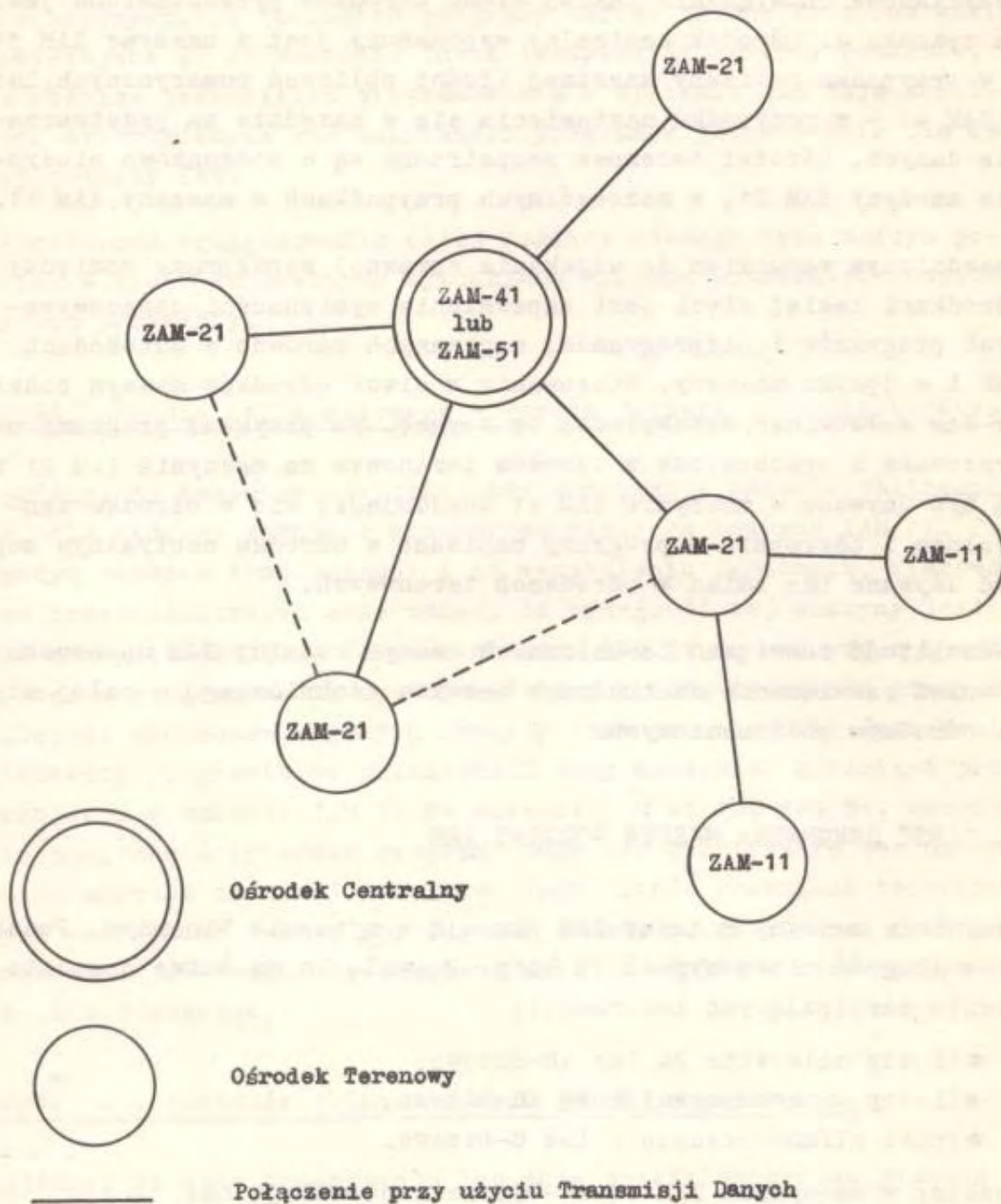
## **Współpraca ośrodków obliczeniowych jednej organizacji.**

Założmy, że duża organizacja lub duże przedsiębiorstwo planuje sieć ośrodków, z których jeden przewidziany jest jako centralny i powinien być zaopatrzonej w dużą maszynę o wysokiej wydajności, natomiast inne ośrodki mogą być zaopatrzone w maszyny mniejsze, lecz za to stosunkowo niedrogie. Zakłada się przy tym stałą współpracę pomiędzy tymi ośrodkami ewentualnie przy zastosowaniu automatycznej transmisji danych.

Przykładowe rozwiązanie takiej sieci ośrodków przedstawione jest na rysunku 4. Ośrodek centralny wyposażony jest w maszynę ZAM 51, w przypadku potrzeby znacznej ilości obliczeń numerycznych, lub w ZAM 41, w przypadku nastawienia się w zasadzie na przetwarzanie danych. Ośrodki terenowe zaopatrzone są w stosunkowo niedrogie maszyny ZAM 21, w szczególnych przypadkach w maszyny ZAM 11.

Zasadniczym warunkiem do uzyskania sprawnej współpracy pomiędzy ośrodkami takiej sieci jest zapewnienie wymienności opracowywanych programów i podprogramów, napisanych zarówno w autokodach, jak i w języku maszyny. Stosowanie w sieci ośrodków maszyn rodziny ZAM całkowicie zabezpiecza tę sprawę. Na przykład programy opracowane i uruchomione w ośrodku terenowym na maszynie ZAM 21 mogą być używane w maszynie ZAM 41 znajdującej się w ośrodku centralnym i odwrotnie – programy napisane w ośrodku centralnym mogą być używane bez zmian w ośrodkach terenowych.

Jednolitość rozwiązań technicznych maszyn rodziny ZAM upraszcza również rozwiązanie centralnego serwisu technicznego w całej sieci ośrodków obliczeniowych.



Rys. 4. Przykład sieci ośrodków obliczeniowych należących do jednej organizacji.

## Część centralna maszyn rodziny ZAM

Wszystkie maszyny rodziny ZAM pracują w systemie binarnym. Podstawowa długość słowa wynosi 24 bity. Pozwala to na łatwe przedstawienie następujących informacji:

- Liczby całkowite 24 lub 48-bitowe,
- Liczby zmiennoprzecinkowe 48-bitowe,
- Znaki alfanumeryczne 6 lub 8-bitowe.

Rozkazy w maszynach ZAM są 24-bitowe i pozwalają na:

- Wyróżnienie przeszło 80 różnych rozkazów.
- Tworzenie adresów efektywnych przez:
- Pośrednie adresowanie (bit P)
- Modyfikację przy pomocy rejestrów indeksów (bit B)
- Bezpośrednie adresowanie do 32768 słów
- Efektywne adresowanie do 262144 słów

Postać powyższych informacji w maszynach ZAM przedstawiona jest na rys. 5.

Pamięć ferrytowa w maszynach ZAM składana jest z bloków standardowych zawierających 4096 lub 8192 słów. Maksymalna pojemność pamięci ferrytowej przy normalnym wykonaniu maszyn ZAM wynosi 32768 słów, a przy wykonaniu specjalnym – 262144 słów.

Maszyny ZAM 21 do ZAM 51 posiadają specjalne układy ułatwiające wykonywanie kilku niezależnych programów jednocześnie. Układy te zapewniają również pełne zabezpieczenie przed wzajemną interferencją wykonywanych jednocześnie programów.

W każdej maszynie rodziny ZAM znajduje się 16 tak zwanych rozkazów programowanych, których znaczenie może być określone przez dowolny podprogram. Liczba ta nie obejmuje rozkazów programowanych zastępujących rozkazy wbudowane, na przykład rozkazy zmiennoprzecinkowe.

Rozkazy programowane pozwalają programiście na swobodne rozszerzenie listy rozkazów maszyny w zależności od jego potrzeb.

### Szybkość Operacji Wewnętrznych

Szybkość wykonywania typowych rozkazów wewnętrznych w różnych maszynach rodziny ZAM jest następująca:

Działania stałoprzecinkowe na słowach 24-bitowych (w mikrosekundach):

ZAM	11	21	31	41	51
Dodaj	200	20	20 130	20	20
Mnóż	750	130		130	130

Działania zmiennoprzecinkowe przy wykładniku 9 bitów i ułamku 39 bitów (w mikrosekundach):

ZAM	11	21	31	41	51
Dodaj	2000	450	80	450	80
Mnóż	3000	850	240	850	240

W maszynach ZAM 11, 21 i 41 celowe jest często stosowanie liczb zmiennoprzecinkowych, w których wykładnik posiada 9 bitów a ułamek 24 bity. Czas wykonania działań na tego rodzaju liczbach wynosi (w mikrosekundach):

ZAM	11	21	41
Dodaj	800	350	350
Mnóż	1200	500	500

Czas potrzebny na rozwiązanie układu 30 równań algebraicznych liniowych o 30 niewiadomych dla poszczególnych maszyn rodziny ZAM wynosi (w sekundach):

ZAM	11	21	31	41	51
Ułamek 24 bity	90	20	--	20	—
Ułamek 39 bitów	150	30	8	30	8

W maszynach ZAM możliwe jest również stosowanie rozkazów programowanych na liczbach zmiennoprzecinkowych, w których wykładnik wynosi 9 bitów, a ułamek — 63 bity.

Przy rozwiązywaniu problemów przetwarzania danych większość wykonywanych rozkazów trwa niewiele ponad 20 mikrosekund. W wypadku tym szybkość pracy centralnej części maszyny, począwszy od ZAM 21, wynosi około 40000 operacji na sekundę.

## System Wejścia i Wyjścia

Wszystkie maszyny ZAM posiadają jednolity i wysoce uniwersalny system współpracy z urządzeniami wejścia i wyjścia. Dowolne z tych urządzeń może być podłączone do Arytmometru lub dowolnego Kanału za pośrednictwem takiej samej standardowej Szyny Wejścia-Wyjścia współpracującej z arytmometrem lub jednym z kanałów. Szyna ta zawiera ustaloną ilość przewodów służących do przesyłania danych, informacji sterujących oraz sygnałów przerwania programu. Wszystkie urządzenia wejścia i wyjścia są więc wykonywane standardowo, niezależnie od typu maszyny z jaką będą współpracować oraz sposobu ich współpracy z maszyną.

System współpracy z urządzeniami wejścia i wyjścia oparty jest w maszynach ZAM na zasadzie podziału czasu jednostki centralnej. Ponadto, dzięki dużej szybkości operacji wewnętrznych, wiele funkcji wykonywanych w innych maszynach przez urządzenia elektroniczne spełniają w maszynach ZAM odpowiednie podprogramy. Przyjęty system zapewnia w szczególności:

- Jednoczesność pracy wielu urządzeń wejścia i wyjścia w ramach jednego programu.
- Łatwość dołączenia do maszyny urządzeń wejścia lub wyjścia dowolnego typu.
- Dowolność przyjętego kodu zapisu informacji na nośnikach zewnętrznych jak na przykład kartach perforowanych lub taśmie papierowej.

Każde urządzenie wejścia i wyjścia buforowane jest przez odpowiednie pole, znajdujące się w centralnej pamięci ferrytowej. Wielkość i położenie tego pola ustalone są programowo. Przesyłanie bloku informacji pomiędzy polem buforowym a urządzeniem wejścia i wyjścia może przebiegać w następujący sposób:

- Programowe przesyłanie bloku słów za pośrednictwem arytmometru. System ten używany jest przy powolniejszych urządzeniach wejścia i wyjścia, na przykład czytniku taśmy papierowej. Maksymalna szybkość przesyłania informacji w tym systemie wynosi 12000 znaków na sekundę w maszynach ZAM 11 i 30000 znaków na sekundę w pozostałych maszynach.
- Automatyczne przesyłanie bloku słów za pośrednictwem kanałów, pozwalające na bardzo szybką współpracę maszyny z pamięciami masowymi, na przykład taśmami magnetycznymi. Maksymalna szybkość przesyłania informacji w tym systemie wynosi dla wszystkich maszyn ZAM 200000 znaków na sekundę. Do każdej maszyny ZAM można dołączyć do sześciu kanałów.

W każdej maszynie ZAM zawarty jest system wielopriorytetowego przerywania programu o następujących możliwościach:

- Praktycznie dowolna ilość możliwych przyczyn przerwania programu,
- Oddzielny priorytet dla każdej przyczyny,
- Programowe włączanie lub blokada przerwania programu.

Do każdej maszyny rodziny ZAM dołączony może być zegar, pozwalający na sterowanie urządzeniami zewnętrznymi maszyny według zadanego z góry harmonogramu czasowego.

## Standardowe Urządzenia Wejścia i Wyjścia

Do każdej maszyny rodziny ZAM może być dołączona praktycznie nieograniczona ilość urządzeń wejścia lub wyjścia dowolnego typu.

W najbliższym okresie przewiduje się wyposażenie maszyn ZAM w następujące urządzenia:

- Czytnik Taśmy Papierowej 5, 7 lub 8 ścieżkowej o szybkości czytania 300, lub 1000 znaków na sekundę.
- Perforator Taśmy Papierowej 5, 7 lub 8 ścieżkowej o szybkości dziurkowania 150 znaków na sekundę.
- Elektryczna Maszyna do Pisania o szybkości 10 znaków na sekundę
- Drukarka Wierszowa 120-kolumnowa o szybkości drukowania 600 linii na minutę.
- Czytnik Kart z odczytem kolumnowym o szybkości czytania 400-600 lub 900 kart na minutę.
- Czytnik Kart z odczytem wierszowym o szybkości czytania do 900 kart na minutę.
- Perforator Kart o szybkości perforowania 100 kart na minutę.
- Bęben Magnetyczny o następujących danych:
  - Pojemność jednego bębna 131 072 lub 524 288 znaków.
  - Kontrola poprawności zapisu przez badanie parzystości.
  - Szybkość obrotów bębna 1500/sek
  - Do jednego kanału lub synchronizatora może być dołączonych osiem bębnow.
- Stacje Taśm Magnetycznych o następujących właściwościach:
  - Zgodność z proponowanym standardem ISO:
  - Szerokość taśmy 1/2 cala
  - Osiem ścieżek informacyjnych oraz jedna kontrolna.
  - Poprzeczna i podłużna kontrola parzystości.
  - Szybkość pisania i czytania 24 000 znaków 8-bitowych na sekundę.
  - Kontrola poprawności zapisu przez podwójny układ głowic.
- Kanały Automatyki łączące maszynę z obiektem sterowanym.



## Podział czasu i wieloprogramowość w maszynach ZAM

Maszyny rodziny ZAM, podobnie jak wiele innych współczesnych maszyn matematycznych, składają się z jednej bardzo szybkiej elektronicznej części centralnej (arytmometr, sterowanie i pamięć ferrytowa) oraz z wielu stosunkowo powolnych mechanicznych urządzeń wejścia i wyjścia, takich jak czytniki, perforatory, drukarki lub taśmy magnetyczne. A więc w przypadku, kiedy maszyna współpracuje z jednym tylko takim urządzeniem, możliwości części centralnej są wykorzystane w niewielkim tylko stopniu.

Organizacja maszyn rodziny ZAM pozwala na to, aby w ramach jednego programu mogło pracować kilka urządzeń wejścia i wyjścia jednocześnie. Uzyskuje się to przez podział czasu (Time Sharing) części centralnej maszyny, która jest dostatecznie szybka, aby obsłużyć kilka takich urządzeń, nawet w przypadku pełnej szybkości ich działania.

Rozpatrzmy przykładowo czynność czytania przez maszynę danych zapisanych na taśmie papierowej. Przy szybkości pracy czytnika równej 300 znaków na sekundę jeden znak pojawia się średnio co 3,3 milisekundy. Na przeczytanie jednego znaku i umieszczenie go w odpowiednim miejscu pamięci ferrytowej część centralna maszyny ZAM 41 zużywa 0,2 milisekundy. Wynika z tego, że czynność odczytania danych z taśmy papierowej i umieszczenie ich w pamięci ferrytowej zabiera części centralnej średnio tylko 6% czasu.

Założmy, że czynność kontroli i przeliczenia danych przeczytanych z taśmy papierowej zajmuje 20% czasu części centralnej maszyny, a czynność zapisania przeliczonych wyników na taśmie magnetycznej wymaga 4% jej czasu. Wobec tego cały program przepisania danych z taśmy papierowej na taśmę magnetyczną łącznie z przeliczeniem wymaga tylko 30% czasu części centralnej maszyny. Ponadto, tego rodzaju program zajmuje na ogół tylko niewielką część pamięci ferrytowej maszyny.

Jak wynika z powyższego przykładu, często jeden program nie jest w stanie wykorzystać pełnych możliwości maszyny. Dlatego też organizacja maszyn matematycznych rodziny ZAM została tak pomyślana, aby umożliwić wykonywanie kilku programów jednocześnie. Podnosi to na ogół bardzo znacznie wydajność całego zestawu maszyny dzięki lepszemu wykorzystaniu szybkości jej części centralnej oraz zmniejszeniu przestojów urządzeń wejścia i wyjścia.

Wieloprogramowość maszyn rodziny ZAM została osiągnięta przy użyciu niewielkich dodatkowych układów elektronicznych, dołączonych do układów sterowania maszyną. Większość funkcji związanych z wieloprogramowością spełnia specjalny program, zwany DYRYGENTEM, umieszczony na stałe w pamięci ferrytowej maszyny. Maksymalna ilość jednocześnie wykonywanych programów, zwanych często PROGRAMAMI NORMALNYMI, zależy jedynie od struktury DYRYGENTA i może być każdorazowo dostosowywana do przyjętego zastosowania maszyny. W praktyce ilość ta prawie nigdy nie przekracza pięciu.

Przykładowo, maszyna ZAM 41, pracująca w zestawie przedstawionym na rysunku 4, może jednocześnie wykonywać następujące programy:

A. Program czytania z kart perforowanych, przeliczania tych danych do postaci wewnętrznej w maszynie, następnie zapisywania ich na taśmie magnetycznej. Program ten wykorzystuje:

- Czytnik Kart o szybkości czytania 400 kart na minutę,
- Jedną Stację Taśmy Magnetycznej
- 1500 słów z Pamięci Ferrytowej.

Przyjmując, że na każdej karcie wydziurkowanych jest przeciętnie po 60 kolumn i czytnik kart pracuje z pełną szybkością, program ten zajmie około 30% czasu centralnej części maszyny.

B. Program czytania danych z taśmy magnetycznej, przetwarzania ich do odpowiedniej postaci wydawniczej, następnie pisanie na drukarce wierszowej. Program ten wykorzystuje:

- Jedną Stację Taśmy Magnetycznej,
- Drukarkę o szybkości pisanie 600 wierszy 120-kolumnowych na minutę
- 2000 słów Pamięci Ferrytowej.

Przyjmując, że w każdym wierszu zapisujemy przeciętnie 20 znaków alfanumerycznych i 30 cyfr dziesiętnych oraz że drukarka pracuje z pełną szybkością, program ten zajmuje około 30% czasu części centralnej maszyny.

C. Program aktualizacji ewidencji. Polega on na przejrzeniu Ewidencji Pierwotnej, zapisanej na taśmach magnetycznych i utworzeniu Ewidencji Aktualnej, którą zapisujemy na nowych taśmach. Ewidencja Aktualna powstaje z Ewidencji Pierwotnej przez wprowadzenie do niej zmian zgodnie z Wykazem Zaszłości, zapisanym na oddzielnych taśmach magnetycznych. Jednocześnie utworzona zostaje taśma Sprawozdań, zawierająca informacje związane z aktualizacją poszczególnych pozycji Ewidencji Pierwotnej. Przyjmujemy, że program ten zajmuje:

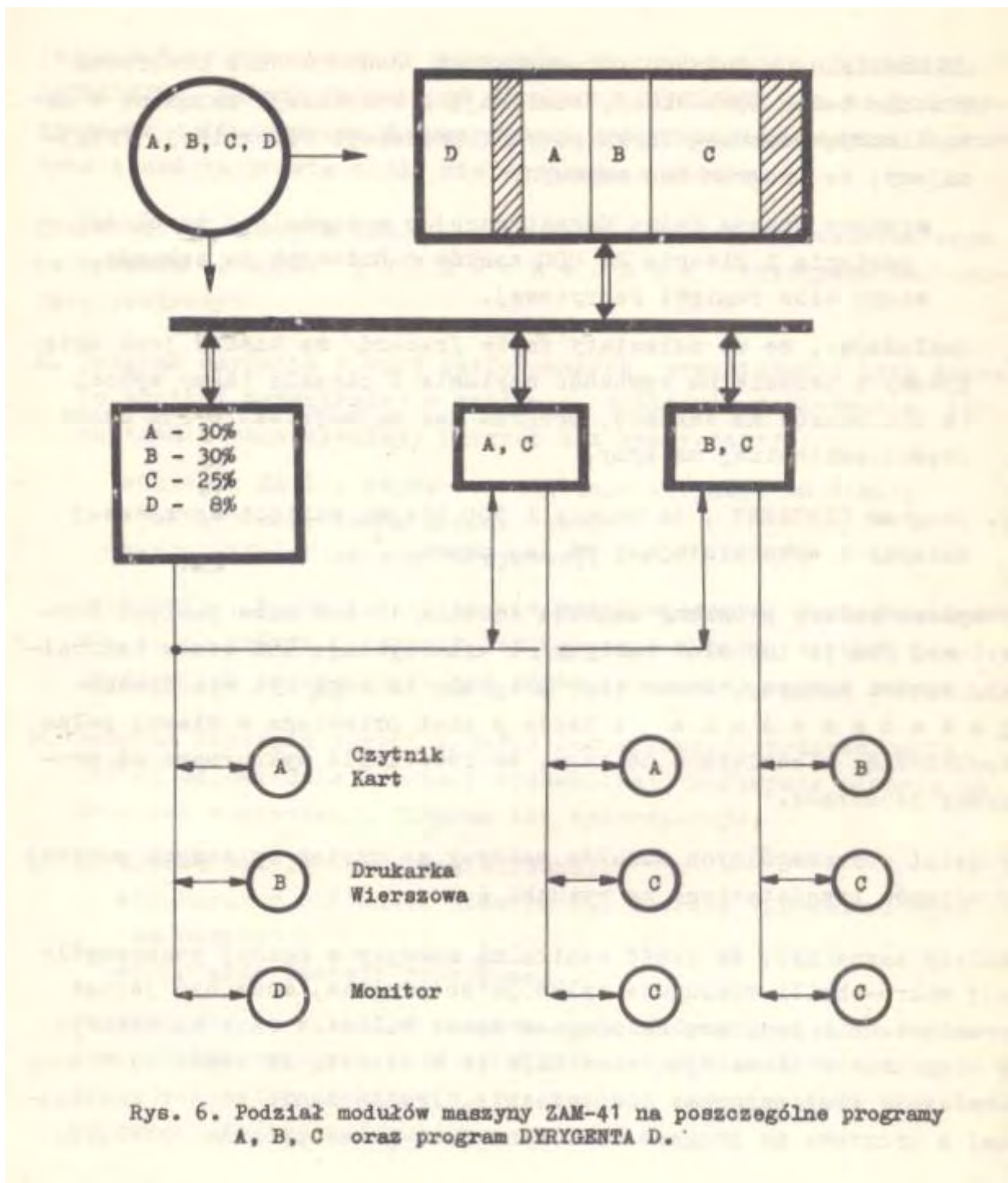
- Cztery Stacje Taśmy Magnetycznej o maksymalnej szybkości czytania i pisanie 24 000 znaków 8-bitowych na sekundę,
- 6000 słów Pamięci Ferrytowej.

Zakładając, że co dziesiąty zapis (record) na taśmie jest księgowany i przeciętna szybkość czytania i pisanie taśmy wynosi 16 000 znaków na sekundę, program ten zajmuje około 25% czasu części centralnej maszyny.

D. Program DRYGENT, zajmujący 1500 miejsc pamięci ferrytowej maszyny i wykorzystujący 8% jej czasu.

Powyższe cztery programy zajmują łącznie 11000 słów pamięci ferrytowej (na 16192 słów maszyny) i wykorzystują 93% czasu centralnej części maszyny. Wobec tego programy te mogą być wykonywane jednocześnie a każdy z nich przebiega z niemal pełną szybkością, niezależnie od tego, że równoległe wykonywane są programy pozostałe.

Podział poszczególnych modułów maszyny na użytek opisanych powyżej programów przedstawiono na rysunku 6.



Należy zaznaczyć, że część centralna maszyny w każdej poszczególnej mikro-chwili obsługuje tylko jeden program, może być jednak przełączana z programu na program nawet kilkaset razy na sekundę. W ciągu nieco dłuższego czasu daje to wrażenie, że część centralna obsługuje trzy programy jednocześnie. Przełączanie części centralnej z programu na program sterowane jest przez DYRYGENTA.

Wieloprogramowość maszyn ZAM pozwala również na osiągnięcie pełnej niezależności wykonywanych jednocześnie programów, zarówno przy ich kodowaniu, jak i wykonywaniu.

W szczególności jakakolwiek omyłka w jednym programie nie jest w stanie zakłócić prawidłowego przebiegu programów pozostałych. Wieloprogramowość maszyn ZAM jest na ogół bardzo opłacalna w zagadnieniach sterowania. Dzięki niej jedna maszyna może sterować wieloma obiektami w tym samym czasie i niezależnie jeden od drugiego. Z powyższych przykładów widoczne jest, że w wielu przypadkach jedna maszyna ZAM staje się równoważna wielu maszynom pracującym jedno-programowo.

## Systemy programowania w maszynach ZAM

Systemy programowania, przyjęte dla maszyn ZAM, pozwalają na stosunkowo łatwe opracowanie programów przez ich bezpośrednich użytkowników, jak na przykład konstruktorów lub ekonomistów. Systemy te rozszerzają krąg osób stosujących maszyny matematyczne, a ponadto samym użytkownikom przynoszą znaczne korzyści, gdyż umożliwiają im bezpośredni kontakt z maszyną bez pośrednictwa wysoko wyspecjalizowanych programistów.

Każdy system programowania składa się z języka, w którym użytkownik koduje swoje programy oraz programu translatora, który tłumaczy program zakodowany w języku systemu na program wynikowy zapisany w języku maszyny.

Przyjęte dla maszyn ZAM języki ALGOL i COBOL oparte są na międzynarodowych standardach, stosowanych już przez bardzo liczne ośrodki obliczeniowe. Język ALGOL posługuje się notacją, stosowaną na co dzień przez matematyków, a język COBOL stosuje pojęcia łatwo zrozumiałe przez ekonomistów. Dzięki temu opanowanie tych języków zabiera stosunkowo niewiele czasu. Ponadto pozwalają one na o wiele szybsze układanie programów niż w przypadku programowania w języku maszyny.

Programy - translatory, w przypadku maszyn ZAM, zawierają od kilku do kilkudziesięciu tysięcy rozkazów maszyny. Są one opracowywane przez odpowiednie zespoły specjalistów i tworzą standardowe wyposażenie wszystkich maszyn ZAM. Pozwalają one na uzyskiwanie sprawnych programów wynikowych, na ogół nie gorszych od programów pisanych przez doświadczonych programistów. Stało się to możliwe w znacznej mierze dzięki temu, że już od samego początku projektowania rodziny maszyn ZAM starano się uwzględnić wszystkie istotne potrzeby, wynikające z przewidywanych systemów programowania.

## **SAS - System Adresów Symbolicznych.**

W języku SAS zwanym też językiem maszyny korzystać możemy ze wszystkich rozkazów przewidzianych przez organizację maszyny. Pozwala to na najbardziej sprawne i wszechstronne wykorzystanie wszystkich możliwości maszyny. Niektóre zastosowania języka SAS opisane zostały w rozdziale dotyczącym jednolitości programowania rodziny maszyn ZAM.

W języku SAS adresy poszczególnych miejsc pamięci oznaczane są symbolicznie, co znacznie ułatwia zarówno napisanie, jak i korektę każdego programu. Symbole adresów w SAS mogą się składać z dowolnej ilości znaków. Język SAS daje też możliwość łatwego korzystania ze standardowych podprogramów związanych na przykład z operacjami wejścia i wyjścia lub obliczaniem funkcji trygonometrycznych. Wystarczy w tym celu zapisać symbol danej operacji, a odpowiadający jej podprogram zostaje automatycznie dołączony do programu w czasie jego tłumaczenia przez translator.

## **MAKRO-SAS**

Język MAKRO-SAS zawiera w sobie język SAS, a ponadto umożliwia definiowanie operacji symbolicznych, zwanych też często makro-rozkazami, których znaczenie jest określone przez programistę, na przykład za pomocą kilku rozkazów maszyny lub też odpowiedniego podprogramu. Przykładowo możliwe jest wprowadzenie makro-rozkazów określających działania arytmetyczne na wielkościach zespolonych. Programista, po określeniu znaczenia takich makro-rozkazów, może korzystać z nich zupełnie podobnie jak z rozkazów maszyny. W ten sposób programista może każdorazowo tworzyć nowy, własny język programowania najlepiej dostosowany do rozwiązywanego przezeń problemu. Korzystanie z makro-rozkazów znacznie ułatwia i przyspiesza pisanie wielu programów przy jednoczesnym zachowaniu ich dużej efektywności.

Korzystanie z języka MAKRO-SAS jest na ogół trudniejsze od korzystania z autokodów ALGOL lub COBOL. Pomimo to, język MAKRO-SAS jest tak prosty w użyciu, że może być stosunkowo łatwo przyswojony przez bardzo wielu użytkowników niebędących zawodowymi programistami.

## **ALGOL — standardowy język algorytmiczny dla problemów numerycznych.**

We wszystkich maszynach rodziny ZAM stosowany będzie język ALGOL, przyjęty już przez wiele organizacji zarówno jako standardowy język publikowania algorytmów oraz jako standardowy język programowania. W najbliższym czasie należy oczekiwać przyjęcia języka ALGOL jako oficjalnego standardu międzynarodowego. Do krajów, które przyjmują ALGOL jako język obowiązujący zaliczyć należy w szczególności większość państw socjalistycznych.

Realizacja języka ALGOL w maszynach ZAM pozwoli więc użytkownikom tych maszyn na bezpośrednie korzystanie z bibliotek programów i podprogramów opracowanych w bardzo wielu ośrodkach na całym świecie.

Wersja języka ALGOL przyjęta dla maszyn ZAM zawiera w szczególności standardowy podzbiór ECMA oraz podzbiór SUBSET ALGOL 60 - IFIP. Ponadto nie posiada ona ograniczeń w „podstawianiu przez nazwę” w procedurach oraz pozwala na stosowanie procedur rekursywnych.

## **SAKO - polski język algorytmiczny dla problemów numerycznych**

Język SAKO opracowany został dla maszyn ZAM 2 i został dość szybko rozpowszechniony w naszym kraju. Stał się on narzędziem codziennego użytku dla wielu specjalistów różnych dziedzin, a ośrodki obliczeniowe, posiadające maszyny ZAM 2, zaopatrzone zostały w liczne programy i podprogramy w języku SAKO.

Dla przedłużenia aktualności tego dorobku wszystkie maszyny rodziny ZAM będą zaopatrzone również w system SAKO, dzięki czemu każdy program opracowany w tym języku dla maszyny ZAM 2 będzie mógł pracować bez żadnych przeróbek na każdej nowej maszynie rodziny ZAM. Pojawia się tu nawet dodatkowe udogodnienia, na przykład rozszerzenie pamięci wewnętrznej maszyny usunie kłopoty związane z podziałem programu na rozdziały, a działania zmiennoprzecinkowe uczynią zbędne skalowanie liczb.

Należy zaznaczyć, że język SAKO odznacza się daleko idącą przejrzystością i prostotą, dzięki czemu jest on bardzo chętnie stosowany na przykład przez konstruktorów, którym zależy na szybkim opanowaniu programowania. Na tej podstawie można się spodziewać, że język SAKO utrzyma długo swój własny krąg zastosowań pomimo istnienia dla maszyn rodziny ZAM innych języków numerycznych, na przykład ALGOL-u.

## **COBOL - międzynarodowy język dla przetwarzania danych**

W maszynach rodziny ZAM stosowanych do przetwarzania danych proponowany jest jako międzynarodowy język standardowy dla problemów przetwarzania danych. Język COBOL będzie zrealizowany zarówno w oryginalnej wersji angielskiej, jak i polskiej. Możliwość stosowania terminów polskich, jako tak zwanych słów kluczowych, ułatwi jego zastosowanie przez użytkowników krajowych. Istnieje też możliwość względnie łatwego przetłumaczenia słów kluczowych języka COBOL na dowolny inny język, na przykład język niemiecki.

Przyjęta wersja języka COBOL obejmuje wszystkie zasadnicze idee pełnej wersji tego języka, pomija natomiast bardzo wiele rozwiązań szczegółowych, które powodują w znacznym stopniu skomplikowanie tego języka. Dzięki temu przyjęta wersja języka COBOL jest stosunkowo prosta, łatwa do nauczenia się i stosowania. Została ona szczególnie dobrze przystosowana do maszyn binarnych, jakimi są maszyny ZAM, przez co efektywność programów wynikowych tego języka jest na ogół dość znaczna.

## SO — System Operacyjny Maszyny

Obsługa maszyny matematycznej, zapewniająca ciągle i stosunkowo pełne jej wykorzystanie, jest zadaniem dość złożonym. Dlatego też w celu zasadniczego ułatwienia pracy operatorowi maszyny opracowany został dla maszyn ZAM specjalny program, zwany Systemem Operacyjnym Maszyny. Program ten składa się z szeregu podprogramów zapisanych na stałe w pamięci bębnowej i przekazywanych w miarę potrzeby do pamięci ferrytowej maszyny. Wykonuje on wiele funkcji, jakie normalnie musiałyby spełniać operator, w szczególności:

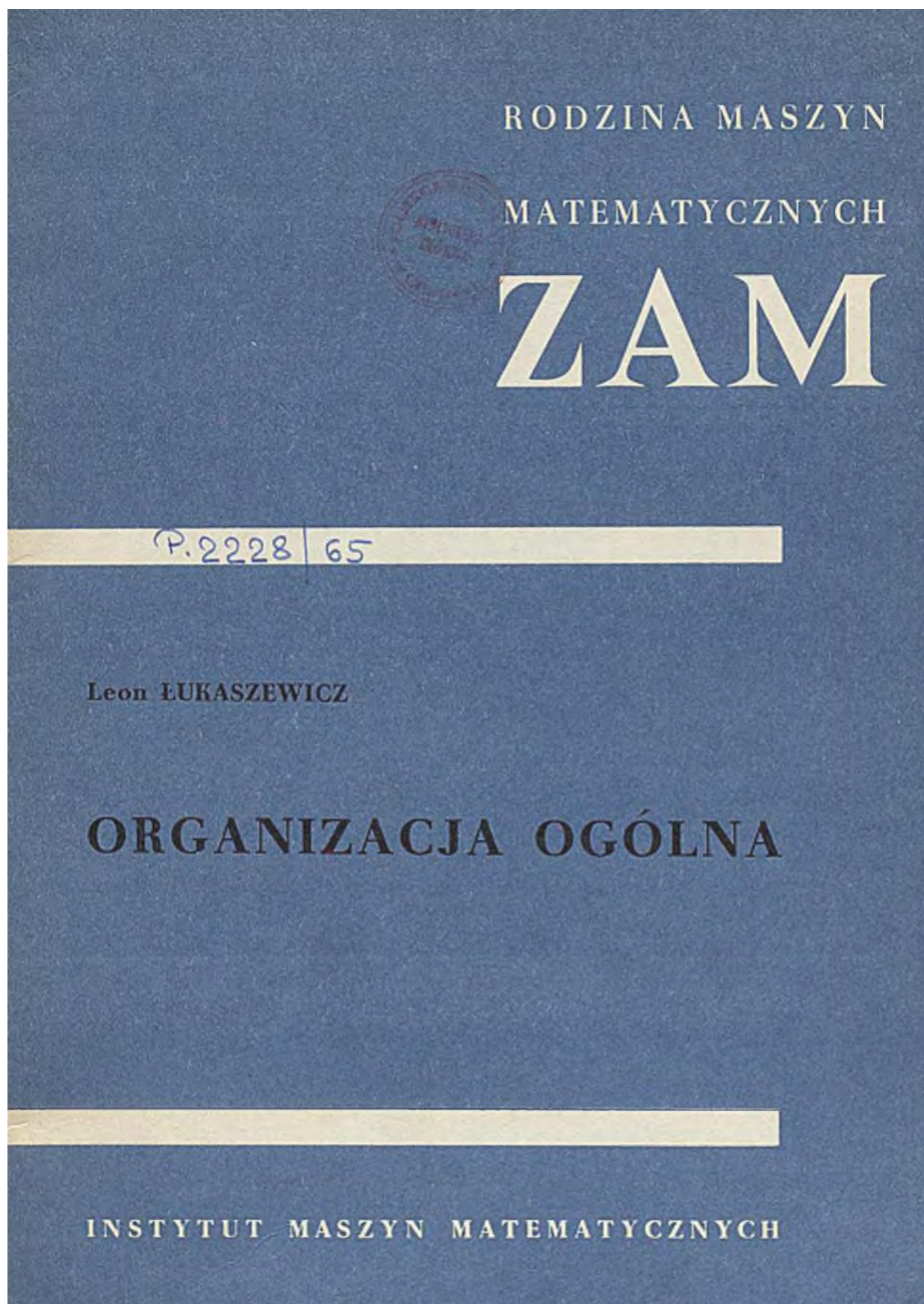
- Sporządza harmonogramy pracy maszyny zapewniające najlepsze jej wykorzystanie, co ma szczególne znaczenie w przypadku maszyn wieloprogramowych.
- Automatycznie 'ładuje' do pamięci ferrytowej programy przewidziane do kolejnego wykonania i przechowywane przejściowo na taśmie magnetycznej lub w pamięci bębnowej maszyny. W ten sposób unika się niepotrzebnej straty czasu pomiędzy zakończeniem jednego programu a rozpoczęciem następnego.
- Sprawuje ciągłą kontrolę nad bieżąco wykonywanymi programami i dostarcza operatorowi odpowiednich raportów na temat przebiegu wszystkich programów. Włącza automatycznie standardowe podprogramy korekcji lub lokalizacji wykrytych błędów, dzięki czemu konieczność interwencji samego operatora ograniczona jest do minimum.
- Zapewnia łatwą komunikację operatora z maszyną.

W wyniku zastosowania Systemu Operacyjnego Maszyny, obsługa maszyn ZAM staje się stosunkowo łatwa i nie wymaga wysokiej specjalizacji operatora.

Na zakończenie opisu systemów programowania w maszynach ZAM należy dodać, że zawierają one szereg dodatkowych środków kontrolnych, pozwalających na wykrycie nie tylko wielu błędów zawartych w samych programach, ale również wielu błędów maszyny, powstałych w czasie wykonywania tych programów.



Zarys organizacji ogólnej maszyn ZAM



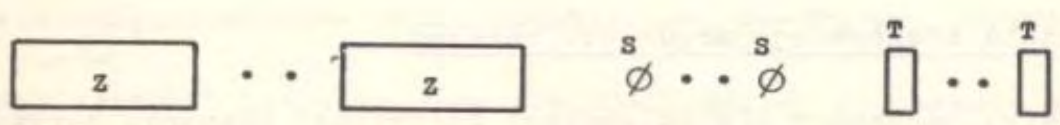
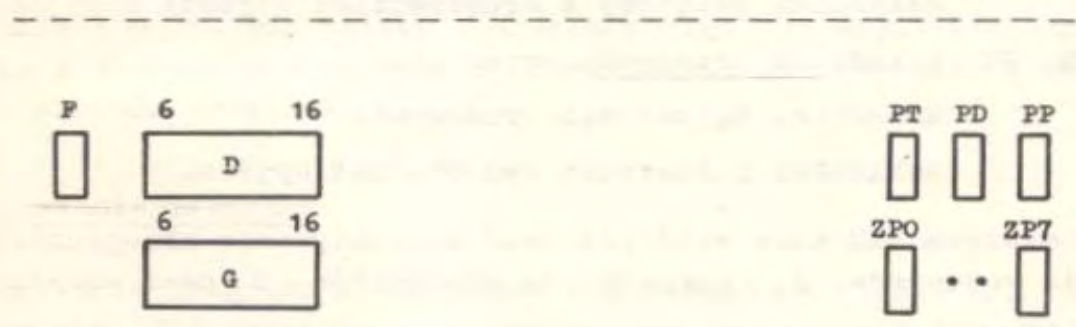
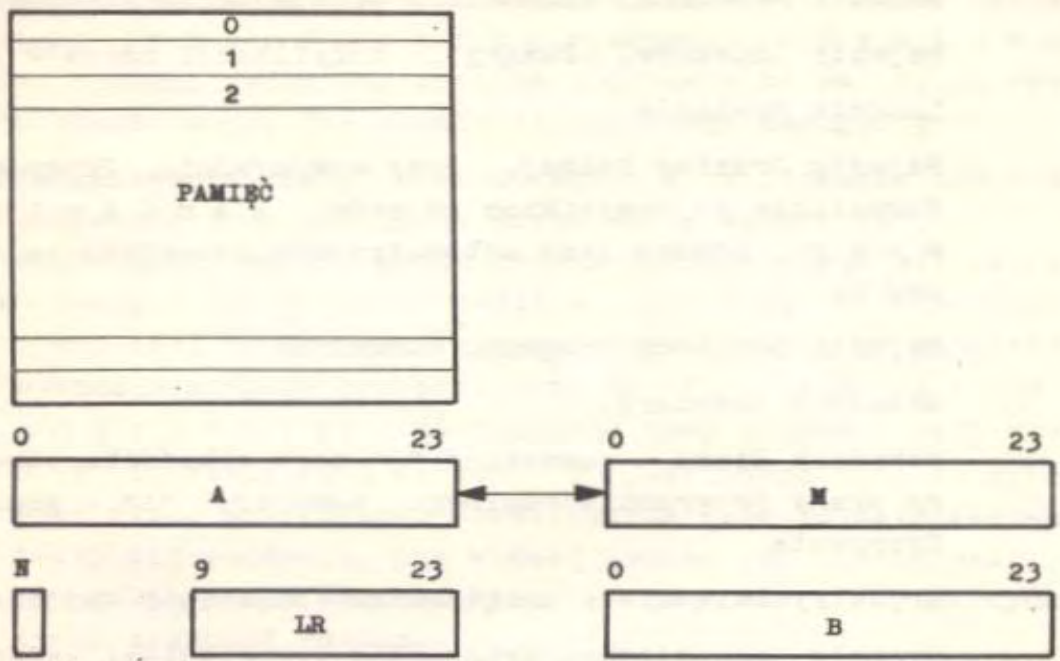
Przedstawiony poniżej zarys organizacji ogólnej odnosi się do wszystkich maszyn ZAM, aczkolwiek zrealizowany jest w całości tylko w maszynach ZAM 51. W maszynach ZAM 21 i ZAM 41 rozkazy zmiennoprzecinkowe ZAM-51 zastąpione są przez rozkazy programowane. W maszynie ZAM 11 większość rozkazów wbudowanych ZAM 51 zastąpiono rozkazami programowanymi. Dzięki rozkazom programowanym wszystkie rozkazy ZAM 51 mogą być łatwo odtworzone w pozostałych maszynach ZAM, co pozwala na opracowanie systemu programowania jednolitego dla całej rodziny maszyn ZAM.

## Rejestry i Wskaźniki

W maszynach ZAM dostępne są dla programisty rejestry i wskaźniki, przedstawione na rys. 7. Znaczenie ich jest następujące:

Symbol	Nazwa i Funkcja
A	Akumulator, podstawowy rejestr dla operacji arytmetycznych i logicznych. Może pośredniczyć pomiędzy rejestrami Z a pamięcią ferrytową maszyny.
M	Rejestr Mnożnika, stanowiący przedłużenie Akumulatora
B	Rejestr Indeksów, służący do modyfikacji adresów
LR	Licznik Rozkazów
D	Rejestr Granicy Dółnej. Przy wykonywaniu Programu Normalnego do wszystkich adresów pamięciowych dodana jest automatycznie zawartość rejestru D.
G	Rejestr Długości Programu Normalnego
N	Wskaźnik Nadmiaru
F	Wskaźnik Stanu - zawartość '0' tego wskaźnika odpowiada pracy Programu Normalnego, zawartość '1' - pracy Dyrygenta
Z	Rejestry związane z urządzeniami Wejścia i Wyjścia
S	Sygnały jedno-bitowe, związane z urządzeniami Wejścia i Wyjścia
T	Wskaźniki związane z urządzeniami Wejścia i Wyjścia
PP, PD, PT	Wskaźniki Przyjęcia Przerwania
ZP	Wskaźniki Zgłoszenia Przerwania
ZW	Wskaźniki Zgłoszenia Wejścia lub Wyjścia

Każda maszyna ZAM może współpracować z praktycznie nieograniczoną ilością rejestrów Z, sygnałów S, wskaźników T oraz wskaźników ZW. Rejestry D i G oraz wskaźnik F znacznie ułatwiają wieloprogramowy system pracy maszyny.



Rys. 7. Rejestry i wskaźniki dostępne dla programisty w maszynach ZAM.

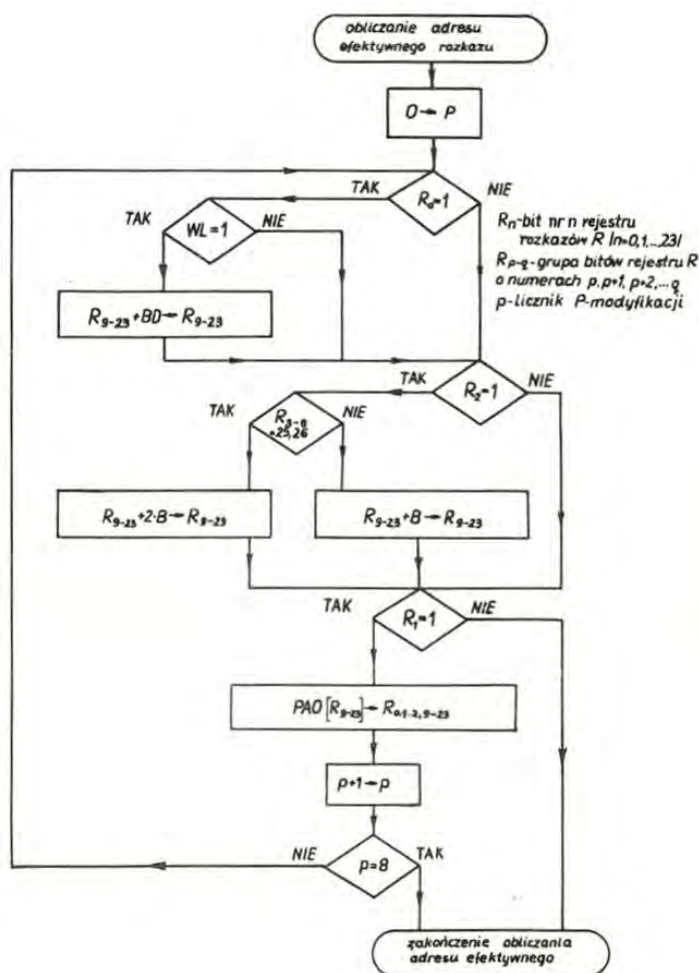
## Obliczanie efektywnego argumentu rozkazu

Argumenty, odnoszące się do pamięci ferrytowej maszyny, nazywamy adresami pamięciowymi lub też krótko adresami.

Zawartość określona przez 15 prawych bitów w rozkazie nazywamy argumentem pierwotnym rozkazu. Efektywny argument rozkazu obliczany jest na podstawie bitów B, P oraz argumentu pierwotnego. Obliczenie to przebiega następująco:

- Indeksowanie. Jeśli wartość bitu B w rozkazie lub w słowie pośrednim przy pośrednim adresowaniu wynosi ulu, to do odpowiadajacego mu argumentu dodana zostaje zawartość rejestru B.
- Pośrednie adresowanie. Jeśli wartość bitu P w rozkazie jest równa '1' to argument rozkazu (ewentualnie po zindeksowaniu) wskazuje miejsce pamięci, skąd należy pobrać słowo pośrednie, zawierające nowy argument oraz nowe bity B i P. Jeśli nowa wartość P jest równa '1', to pobranie nowego słowa pośredniego (ewentualnie znów po zindeksowaniu) powtórzy się podobnie, nie więcej jednak jak siedem razy. Za ósmym razem lub kiedy wartość bitu P jest równa '0' otrzymujemy efektywny argument rozkazu.

Indeksowanie może następować na każdym szczeblu pośredniego adresowania i poprzedza zawsze pobranie następnego argumentu. Indeksowanie i pośrednie adresowanie pozwalają na otrzymywanie efektywnych adresów 18-bitowych.



## Przerywanie Programu

Przerywanie programu w maszynach ZAM następuje wtedy, gdy chociaż jeden wskaźnik Zgłoszenia Przerwania ZP zawiera bit '1' oraz odpowiedni wskaźnik Przyjęć Przerwań PP lub PD zawiera bit '1'. Podstawowe czynności maszyny przy przerywaniu programu to:

- Zakończenie cyklu wykonania bieżącego rozkazu
- Zapamiętanie odpowiednio uzupełnionej zawartości Licznika Rozkazów
- Wpisanie do Licznika Rozkazów 32+P, gdzie P oznacza numer tego wskaźnika ZP, który posiada najwyższy priorytet i zawiera bit '1'. Skutkiem powyższych czynności jest naruszenie sekwencji rozkazów jaka jest zadana przez program i przejście do wykonania rozkazu umieszczonego w miejscu pamięci o adresie 32+P.

Przyczyna przerwania	priorytet
Zbliżanie się do głowicy zapisu/odczytu pamięci bębnowej słowa, którego adres został ustawiony w rejestrze adresów pamięci bębnowej.	1
Zakończenie czytania kolumny karty perforowanej,	2
Zakończenie czytania znaku przez czytnik taśmy 2	3
Przepełnienie zegara	4
Zakończenie czytania karty perforowanej.	5
Zakończenie perforowania znaku przez perforator 2	
Zakończenie czytania znaku przez czytnik taśmy 1.	7
Zakończenie drukowania wiersza i przesuwu papieru przez drukarkę wierszową	8
Zakończenie perforowania znaku przez perforator 1	9
Zakończenie drukowania znaku lub przyjęcie znaku przez monitor.	10
Zakończenie wykonywania polecenia wydanego kanałowi taśmy magnetycznej.	12
Rozkaz nielegalny	14
Naciśnięcie przycisku ZO (zgłoszenie operatora) lub błędy.	15

## Rozkazy Nielegalne

Są to rozkazy, które mogłyby zakłócić pracę innych Programów Normalnych. Jeśli wskaźnik F zawiera '0', to próba wykonania rozkazu nielegalnego powoduje przerwanie programu i przejście do odpowiedniego podprogramu DRYGENTA. Jeśli wskaźnik F zawiera '1' to wszystkie rozkazy nielegalne wykonywane są tak samo, jak rozkazy legalne.

## **Rozkazy Programowane**

Rozkazy programowane powodują wywołanie podprogramu określonego przez KOD tego rozkazu. Dzięki temu ARGUMENT rozkazu może być swobodnie wykorzystany przez ten podprogram. Rozkazy programowane mają więc postać rozkazów wbudowanych, a znaczenie ich może być stosunkowo łatwo określone przez programistę.

## **Lista Rozkazów Maszyn ZAM**

### **Rozkazy Sterujące**

<b>SKO</b>	Skocz
<b>SZA</b>	Skocz przy Zerze Akumulatora
<b>SMA</b>	Skocz przy Minusie Akumulatora
<b>SLR</b>	Skocz pamiętając Licznik Rozkazów
<b>SSL</b>	Skocz ze Śladem
<b>SOB</b>	Skocz po Odjęciu jeden od B
<b>SUB</b>	Skocz i Umieść w B
<b>SSE</b>	Skocz do Segmentu
<b>SDY</b>	Skocz do Dyrygenta
<b>WRO</b>	Wróć

### **Rozkazy Programowane**

Pnm Wykonaj rozkaz programowany nm (istnieje 18 takich rozkazów w ZAM 51)

### **Rozkazy rejestru B**

<b>UEB</b>	Umieść Efektywny Adres w B
<b>UMB</b>	Umieść w B
<b>PAB</b>	Pamiętaj B
<b>DOB</b>	Dodaj do B
<b>ODB</b>	Odejmij od B
<b>POB</b>	Porównaj B

### **Przesłania A oraz M**

<b>UMA</b>	Umieść w A
<b>PZA</b>	Pamiętaj i Zeruj A
<b>UMM</b>	Umieść w M
<b>PMM</b>	Pamiętaj M
<b>UAM</b>	Umieść w A oraz M
<b>PAM</b>	Pamiętaj A oraz M
<b>UAD</b>	Umieść Akumulator Długi
<b>PAD</b>	Pamiętaj Akumulator Długi

### **Działania Logiczne**

<b>DOL</b>	Dodaj Logicznie do A
<b>ODL</b>	Odejmij Logicznie od A
<b>MNL</b>	Mnóż Logicznie przez A
<b>POL</b>	Porównaj Logicznie A

### **Działania Stałoprzecinkowe A oraz M**

<b>DOA</b>	Dodaj do A
<b>ODA</b>	Odejmij od A
<b>DOM</b>	Dodaj do M
<b>ODM</b>	Odejmij od M
<b>MNM</b>	Mnóż przez M
<b>DZD</b>	Dziel Akumulator Długi

## **Działania Zmiennoprzecinkowe**

**DOZ** Dodaj w Zmiennym Przecinku  
**ODZ** Odejmij w Zmiennym Przecinku  
**MNZ** Mnóż w Zmiennym Przecinku  
**DZZ** Dziel w Zmiennym Przecinku

## **Działania na słowach w Pamięci**

**DOP** Dodaj do Pamięci jeden  
**ODP** Odejmij od Pamięci jeden i Przeskocz

## **Zmiana zawartości rejestrów**

### **Przeskoki**

**PPB** Przeskocz przy Plusie B  
**PPM** Przeskocz przy Plusie M  
**PRM** Przeskocz przy Różnym od zera M  
**PZN** Przeskocz przy Zerze N  
**PEB** Przeskocz, gdy Efektywny argument nie mniejszy od B

### **Przesłania**

**ZBA** Zamień Zawartość B oraz A  
**ZAM** Zamień Zawartość A oraz M  
**ZMB** Zamień Zawartość M oraz B  
**WMB** Przepisz Wykładnik z M do B  
**WBM** Przepisz Wykładnik z B do M  
**LMB** Mnóż logicznie M przez Efektywny Argument oraz umieść w B

### **Normalizacja, Zaokrąglenia i Działania na Znakach:**

**NOR** Normalizuj  
**OKZ** Zaokrąglij w Zmiennym Przecinku  
**OAM** Zaokrąglij A oraz M  
**ZZN** Zrównaj Znaki  
**NZB** Neguj Znak B

### **Przesunięcia:**

**LCA** Przesuń w Lewo Cyklicznie A  
**PCA** Przesuń w Prawo Cyklicznie A  
**LCM** Przesuń w Lewo Cyklicznie M  
**PCM** Przesuń w Prawo Cyklicznie M  
**LCD** Przesuń w Lewo Cyklicznie AD  
**PCD** Przesuń w Prawo Cyklicznie AD  
**LAR** Przesuń w Lewo Arytmetycznie AD  
**PAR** Przesuń w Prawo Arytmetycznie AD

### **Różne:**

**STO** Stop  
**PDG** Pisz w D oraz G  
**CKA** Czytaj Klucze do A  
**PLA** Pisz Lampki z A  
**NNR** Nic Nie Rób



## **Operacje Wejścia i Wyjścia**

### **Operacje Arytmometru:**

<b>NAS</b>	Nadaj Sygnał
<b>PZS</b>	Przeskocz przy Zerze Sygnału
<b>CRA</b>	Czytaj Równoległe do A
<b>PRA</b>	Pisz Równoległe A
<b>CTA</b>	Czytaj Trzy Znaki do A
<b>PTA</b>	Pisz Trzy Znaki z A

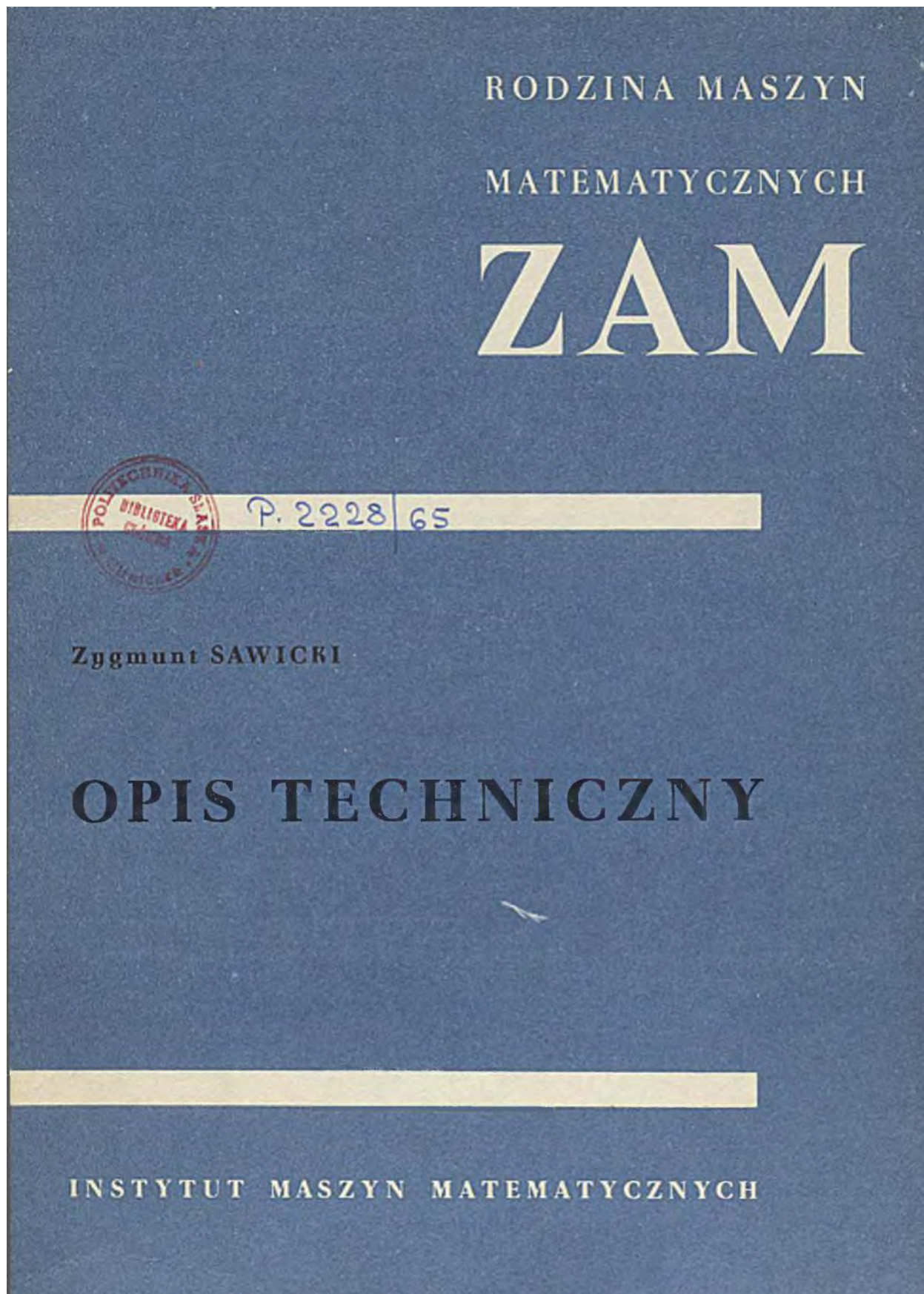
### **Operacje Kanału KS:**

<b>NAS</b>	Nadaj Sygnał
<b>PZS</b>	Przeskocz przy Zerze Sygnału
<b>CRP</b>	Czytaj Równoległe do Pamięci
<b>PRP</b>	Pisz Równoległe z Pamięci
<b>CTB</b>	Czytaj po Trzy Znaki Blok Słów
<b>PTB</b>	Pisz po Trzy Znaki Blok Słów
<b>CRB</b>	Czytaj Równoległe Blok Słów
<b>PRB</b>	Pisz Równoległe Blok Słów

Bliższy opis Listy Rozkazów maszyn ZAM znajduje się w innym opracowaniu wydanym przez IMM (KB - oraz w ZAM-41 Kompendium Oprogramowania Jana Wierzbowskiego).

(KB) - Zwrócić tu należy uwagę na rozbudowaną implementację rozkazu **SDY** (Skocz Do Dyrygenta) udostępniającego komunikację programu z Dyrygentem, sprzętem, szczególnie z urządzeniami Wejścia-Wyjścia, Biblioteką Systemu, Monitorem, Jednostką Centralną i Stolikiem Operatora (p. 3.4.4 Kompendium). Rozkazy **SDY** podzielone są na dwie grupy, Legalne, dla programów użytkowych i Nielegalne, zastrzeżone do użytku przez System Operacyjny.

Opis techniczny maszyn matematycznych rodziny ZAM



## **Wstęp**

Opis techniczny zawiera wstępną informację dla przyszłych użytkowników elektronicznych maszyn cyfrowych rodziny ZAM. Opis ten pozwoli czytelnikowi wyrobić sobie wstępny pogląd o technicznych właściwościach maszyn rodziny ZAM, w szczególności o szybkości liczenia, przyjętych zasadach konstrukcji, niezawodności pracy i możliwości wyboru optymalnego zestawu maszyny dla określonego zakresu zastosowań.

Użytkownikami maszyn rodziny ZAM mogą być instytucje zainteresowane częściowym lub kompleksowym automatyzowaniem zarządzania przy pomocy przetwarzania danych, obliczeniami naukowymi i technicznymi lub sterowaniem procesami technologicznymi.

Do rodziny maszyn ZAM należy pięć typów maszyn, mianowicie: ZAM 11, ZAM 21, ZAM 31, ZAM 41 i ZAM 51. Maszyny rodziny ZAM pokrywają praktycznie cały zakres zastosowań maszyn cyfrowych, pracujących w warunkach stacjonarnych, przy spełnieniu normalnych wymagań w zakresie odporności mechanicznej i klimatycznej.

## **Ogólna charakterystyka maszyn rodziny ZAM**

Szybki rozwój elektronicznej techniki obliczeniowej w świecie postawił nowe wymagania konstruowanym obecnie lub wprowadzanym do produkcji maszynom matematycznym.

Do tych wymagań należy zaliczyć:

- opracowywanie modułowych konstrukcji pozwalających na budowę optymalnych zestawów maszyn dla określonego zakresu zastosowań tj. przetwarzania danych, obliczeń naukowych i technicznych oraz sterowania procesami
- konstruowanie ujednoczonych systemów maszyn obejmujących
- maszyny małe, średnie i duże, tworzących tzw. "rodzinę maszyn"
- unifikowanie w ramach rodziny maszyn:
  - języka wewnętrznego
  - systemów programowania
  - standardów konstrukcyjnych
  - systemów eksploatacji

Elektroniczne maszyny cyfrowe rodziny ZAM spełniają powyższe kryteria warunkujące ich wysoce ekonomiczną eksploatację.

Wszystkie maszyny rodziny ZAM są przystosowane do pracy w warunkach "czasu realnego" poprzez kanały automatyki lub kanały transmisji danych. W związku z tym, maszyny ZAM są wyposażone w układy powodujące przerwanie bieżącego programu maszyny i przełączenie na program, którego sygnał spowodował przerwanie.

Każda maszyna z rodziny ZAM posiada system operacyjny, automatycznie organizujący pracę maszyny w systemie równoczesnej pracy kilku urządzeń

wejścia i wyjścia oraz jednoczesnego wykonywania kilku programów. Maszyny rodziny ZAM mogą być wyposażone w komplet autokodów i bibliotekę standardowych programów do różnych zastosowań, w szczególności:

- przetwarzanie danych
- obliczenia naukowe i techniczne
- rejestracja i sterowanie procesami technologicznymi.

Czas trwania poszczególnych operacji maszyn rodziny ZAM przedstawia Tablica 1.

Tablica 1.

Czas trwania poszczególnych operacji maszyn rodziny ZAM

Operacje		Czas w $\mu$ sek				
		ZAM-11	ZAM-21	ZAM-31	ZAM-41	ZAM-51
Stały przeci- nek, 24 bit.	Dodawanie	200	20	20	20	20
	Odejmowanie	200	20	20	20	20
	Mnożenie	750	130	130	130	130
	Dzielenie	800	160	160	160	160
Zmienny przeci- nek wykl. 9 bitów mantyssa 39 bi- tów	Dodawanie	2 000	450	80	450	80
	Odejmowanie	2 000	450	80	450	80
	Mnożenie	3 000	850	240	850	240
	Dzielenie	3 000	1 100	300	1 100	300
Rozkazy organizac.		200	15	15	15	15

Maszyny ZAM posiadają organizację równoległą, jedno-adresową, o długości słowa 24 bitów, arytmetykę stało- i zmiennoprzecinkową z możliwością zwiększania precyzji obliczeń:

Pamięć ferrytowa wszystkich maszyn ZAM posiada:

- czas cyklu 10  $\mu$ sek
- pojemność do 32768 słów, składaną z bloków po 496 słów,
- lub po 8192 słów,
- pojemność do 262144 słów w wykonaniu specjalnym.

## Technika maszyn rodziny ZAM

Maszyny ZAM są budowane z podzespołów półprzewodnikowych opartych na technice S-400. Układy techniki S-400 zgrupowane są na pakietach wykonywanych technologią obwodów drukowanych. Pakiety tworzą uniwersalny zestaw pozwalający na realizację dowolnego bloku maszyn cyfrowych. Dużą niezawodność pracy układów, realizowanych w technice S-400, osiągnięto dzięki uwzględnieniu w konstrukcji szeregu warunków, mianowicie:

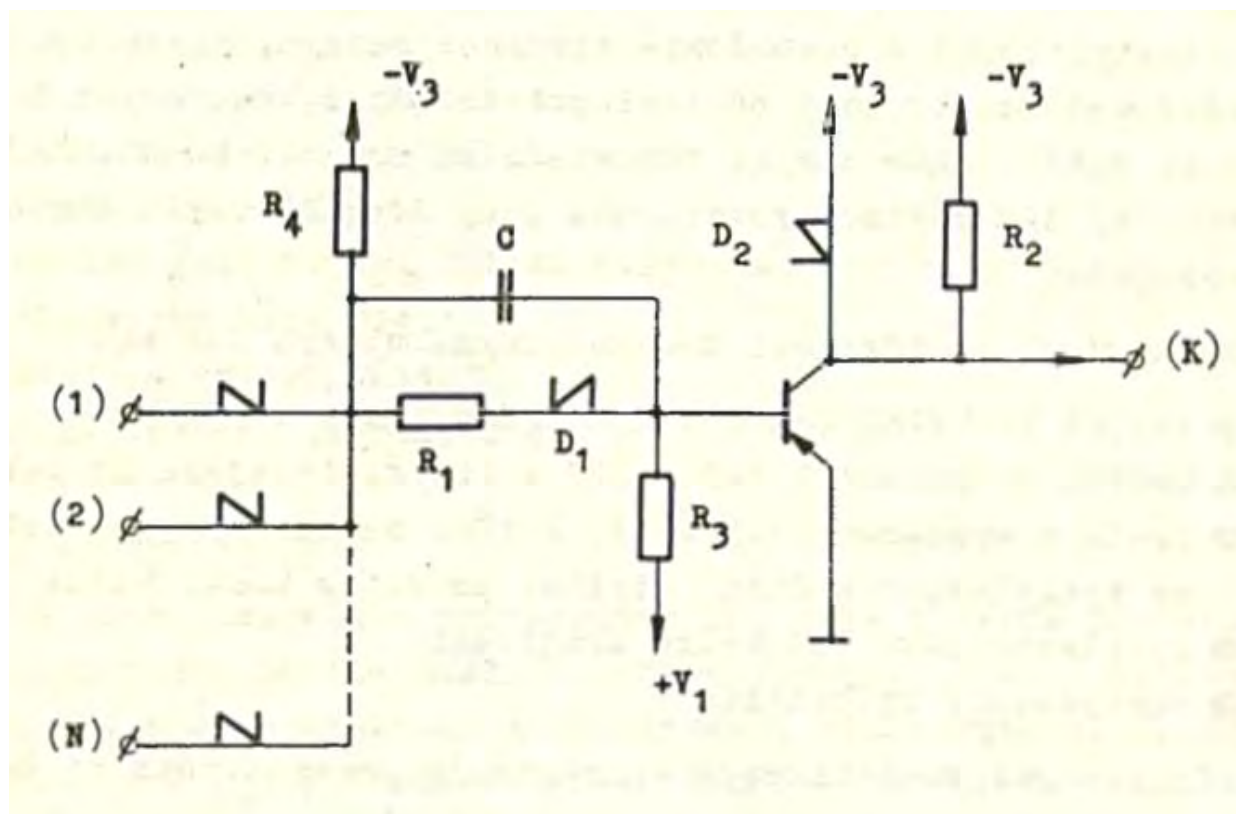
- wszystkie elementy techniki S-400 są obciążone nie więcej niż
- 30% wartości nominalnej
- układy techniki S-400 dopuszczają duże zmiany w czasie parametrów elementów składowych
- instalowane elementy w układach techniki S-400 spełniają warunek stosunkowo małego rozrzutu parametrów, względem wartości nominalnych i są stabilne w czasie
- poszczególne bloki maszyn zasilane są ze źródeł napięć stabilizowanych.

Technika S-400 wykorzystuje w zasadzie jeden funkcyjny logiczny realizujący negację iloczynu lub negację sumy argumentów, co zapewnia systemowi jednorodność, przejrzystość konstrukcji i prostotę działania układów.

### Podstawowy układ techniki S-400

Podstawowym układem techniki S-400 jest inwerter. Jest to wzmacniacz tranzystorowy, poprzedzony jednowarstwową bramką diodową. Bramka diodowa dla konwencji, w której  $-6\text{ V}$  odpowiada "1" "logicznej", realizuje iloczyn argumentów wejściowych. Tranzystor pracuje w układzie wspólnego emitera, stąd oprócz wzmocnienia napięciowego i prądowego daje odwrócenie /inwersję/ sygnału wejściowego. W efekcie, dla przytoczonej konwencji, układ realizuje negację iloczynu. Oba poziomy sygnały wyjściowego układu są ściśle zdefiniowane. Poziom  $0\text{ V}$  określony jest stanem nasycenia tranzystora, poziom  $-6\text{ V}$  określony jest diodą poziomującą D2. W obu ustalonych stanach wyjście charakteryzuje się niską opornością, co wpływa na silne tłumienie zakłóceń o charakterze pojemnościowym. Dzielnik oporów wejściowych R1, R3, R4 oraz dioda D1 zapewniają istnienie marginesów odporności układu na zakłócenia, na wejściu o amplitudzie ok.  $1,5\text{ V}$ , na obu poziomach sygnału.

Rys. 1. Schemat ideowy podstawowego układu techniki S-400.



Układ ten charakteryzuje się dużym wzmocnieniem logicznym  $K=5$ , dużym współczynnikiem spływu logicznego  $N$ . Stanowi on podstawę budowy wszystkich układów techniki S-400.

Podstawowe dane techniczne układów techniki S-400

- poziom sygnałów: 0V i -6
- progi przeciwzakłócenieniowe: powyżej 1V
- czas propagacji Jednego układu: 400 nsek
- maksymalna częstotliwość pracy przerzutnika: 400 Kc/sek
- napięcie zasilające w technice S-400: +12.7, -6V, i -15V
- zakres temperatury otoczenia układów wykonywanych w technice S-400 wynosi od +5°C do +45°C.

## Konstrukcja maszyn rodziny ZAM

Wymagania stawiane nowoczesnym konstrukcjom polegają na: standaryzacji i unifikacji poszczególnych zespołów i bloków maszyn cyfrowych, elastyczności w rozbudowie zestawów maszyn, przez wprowadzenie modułowej konstrukcji oddzielnych urządzeń, łączonych ze sobą za pomocą kabli zakończonych odpowiednimi mufami. Konstrukcja maszyn rodziny ZAM została rozwiązana przy uwzględnieniu wymienionych wymagań.

Podstawowymi standardami mechanicznymi maszyn ZAM są:

- pakiet techniki S-400 o wymiarach 126,4 x 205 mm
- kaseta o wymiarach 145 x 225 x 400 zawierająca 25 pakietów
- szafa o wymiarach 625 x 675 x 1700 mm zawierająca dwie ramy wychylne, przednią i tylną, po osiem kaset każda
- zasilacze jako oddzielne urządzenie
- wentylatory chłodzenia

Szafa stanowi zunifikowaną konstrukcję przeznaczoną do budowy modułów maszyn cyfrowych, opartych na standardowych zespołach: pakiety, kasety, zasilacze, wentylatory.

Kaseta jest blaszaną konstrukcją przestrzenną, skręconą z poszczególnych elementów. W tylnej części kasety zamocowane są łączówki; każda łączówka posiada 32 styki.

Pakiet jest płytką z laminatu szklano-epoksydowego z naniesionym obwodem drukowanym połączeń. Pakiet na końcu zaopatrzony jest w 32 styki współpracujące z łączówką kasety.

## **Moduły maszyn rodziny ZAM**

Moduły maszyn rodziny ZAM stanowią integralne jednostki spełniające ściśle określone funkcje. Budowane są w zasadzie ze standardowych zespołów opisanych w rozdziałach 3, 4 i 5. Modułami maszyn rodziny ZAM są:

- pamięć ferrytowa PA05
- pamięć bębnowa PB5
- pamięć taśmowa SPT2
- stolik operatora S02
- czytnik taśmy papierowej CP1
- dziurkarka taśmy papierowej PD1
- drukarka wierszowa DW1
- czytnik kart CK2
- monitor /dalekopis lub elektryczna maszyna do pisanania/ MD2
- dziurkarka kart DK1
- kanał automatyki KA1
- kanał transmisji danych KT1

### ***Pamięć ferrytowa PA05***

PA05 Jest pamięcią z wybieraniem liniowym o dwóch rdzeniach na bit. Wybieranie adresu pamięci odbywa się przy pomocy biernego przełącznika magnetycznego z polaryzacją stałoprądową. Układy elektroniczne - w pełni tranzystorowe. Pamięć Jest dostosowana do współpracy z maszynami rodziny ZAM i może być dostosowana do współpracy z innymi typami maszyn.

Dane techniczne modułu:

- pojemność: 4 096 słów
- długość słowa: 24 bity
- czas dostępu: 3,5  $\mu$ s
- czas cyklu: 10  $\mu$ s
- zakres temperatur otoczenia od +15°C do 35°C
- pamięć przystosowana do pracy ciągłej
- poziomy sygnałów: 0V i -6V.

W jednej standardowej szafie mieszczą się dwa moduły pamięci operacyjnej po 4 096 słów, które mogą być wykorzystane niezależnie.

- zasilanie: 3 × 220/380 V 50 Hz 0,8 kVA dla dwóch modułów PA05
- wymiary modułu: 625 × 675 × 1700 mm
- ciężar: 260 kg



## **Pamięć bębnowa PB5**

PB5 zaprojektowana jest w zasadzie do współpracy z maszynami należącymi do rodziny ZAM. Modułowa konstrukcja /całość wraz z bębniem mieści się w jednej szafie standardowej/ pozwala Jednak na przystosowanie jej do dowolnej maszyny cyfrowej.

Dane techniczne modułu:

- wymiary bębna: średnica 280 mm wysokość 255 mm
- ekscentryczność /"bicie"/ powierzchni w stosunku do osi:  $<2 \mu$  .
- średni czas dostępu: około 20 ms, natomiast w systemie wieloprogramowym około 0,6 ms
- obroty bębna: około 1500 obr/min
- ilość ścieżek: 150, w tym: 128 informacyjnych, 1 zegarowa, 1 adresowa, 20 zapasowych
- pojemność pamięci: 32 768 słów 27-bitowych, w tym 24 bity informacyjne i 3 bity kontrolne
- gęstość zapisu: 9 bitów/mm w metodzie NRZ
- szybkość przekazywania informacji 6 400 słów/sek
- czas przełączania układu wybierania głowic z jednej ścieżki na drugą: mniejszy niż 0,5 msek
- odległość głowicy od stojącego bębna: około 16  $\mu$
- szerokość szczeliny roboczej głowicy: 40  $\mu$
- ilość zwojów głowicy: 2  $\times$  26
- zakres temperatur pracy pamięci od +10°C do +35°C
- poziomy sygnałów;- 0V i -6V
  - zasilanie: 3  $\times$  220/380 V +5%/-10%, 50 Hz; 0,6 kVA
  - wymiary: 625  $\times$  675  $\times$  1700 mm
  - ciężar: 275 kg

## **Pamięć na taśmie magnetycznej PT2**

Magnetyczna pamięć taśmowa PT2 przeznaczona jest do współpracy z maszynami cyfrowymi. Stanowi ona pamięć zewnętrzną tych maszyn, niezbędną do przetwarzania danych oraz wykonywania bardzo dużych obliczeń.

Dane techniczne modułu:

- szybkość przekazywania informacji: 24 tys. 8-bitowych znaków alfanumerycznych/sekundę
- szerokość taśmy: 1/2"
- forma zapisu informacji: zmienna długość bloku
- długość taśmy: 750 m - przy grubości taśmy 52  $\mu$  , 1100 m - przy grubości taśmy 35  $\mu$
- liczba ścieżek: 9
- gęstość zapisu: ISO - 8 rzędów/mm i normalna 12 rzędów/mm
- pojemność: od 7 do 10 min znaków alfanumerycznych
- prędkość przesuwu taśmy pod głowicami - 2 m/sek

- czas startu taśmy: 6 ms
- czas stopu taśmy: około 10 ms
- średnia prędkość taśmy przy szybkim przewijaniu szpuli: 5 m/sek
- zakres temperatur: 20°C +/- 3°C
- wilgotność względna: 50% +/- 10%
- poziomy sygnałów wyjściowych: 0V i -6V
  - zasilanie: 3 × 220/380V 50 Hz, 2,0 kVA
  - moc wydzielana na ciepło: 1,2 kW wymiary: 700 × 675 × 1700 mm
  - ciężar: ok. 300 kg

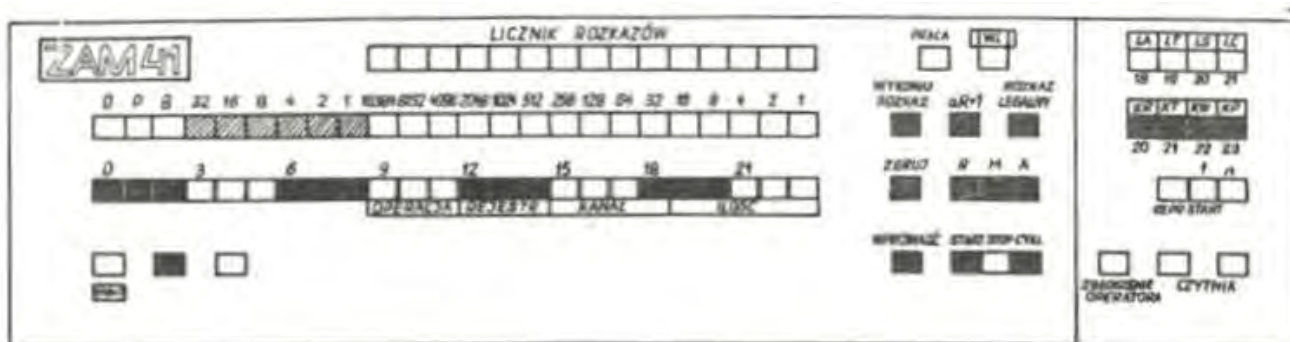
**Uwaga:** w czasie eksploatacji PT2 pomieszczenie musi być odpylane i klimatyzowane.

## Stolik operatora SO2

Stolik operatora stanowi segment zwarty konstrukcyjnie. Stolik składa się z pulpitu sterującego, kanałów czytnika i dziurkarki taśmy papierowej. Czytnik typu TR5B firmy ICT oraz dziurkarka typu PE-1500 firmy Facit umieszczone są na stoliku. W przyszłości będą stosowane: czytnik i dziurkarka taśmy papierowej produkcji krajowej.

Dane techniczne stolika:

- szybkość czytania taśmy: do 1000 znaków/sek, dla czytnika TR5B: 300 znaków/sek
- szybkość dziurkowania taśmy: 150 znaków/sek
- kod dowolny
- ilość ścieżek: 5 do 8
- poziomy sygnałów: 0V i -6V
  - zasilanie: 3 × 220/380V 50 Hz; 0,4 kVA
  - wymiary: 1686 × 710 × 966 mm
  - ciężar: 195 kg



## **Moduł czytnika taśmy CP1**

Moduł czytnika taśmy papierowej stanowi konstrukcyjnie zwarty segment. Moduł czytnika składa się z kanału synchronizującego pracę czytnika z dowolną maszyną rodziny ZAM oraz czytnika taśmy typu TR5B firmy ICT, stojącego na pulpicie segmentu. W przyszłości będzie stosowany czytnik produkcji krajowej.

Dane techniczne modułu:

- szybkość odczytu: do 1000 znaków/sek; dla czytnika TR5B do 300 znaków/sek
- kod dowolny
- ilość ścieżek: 5 do 8
- poziomy sygnałów: 0V i -6V
  - zasilanie: 3 x 220/380 50 Hz; 0,3 kVA
  - wymiary: 710 x 710 x 760 mm
  - ciężar: 100 kg

## **Moduł dziurkarki taśmy papierowej PD1**

Moduł dziurkarki taśmy papierowej stanowi konstrukcyjnie zwarty segment. Moduł dziurkarki składa się z kanału synchronizującego pracę dziurkarki z dowolną maszyną rodziny ZAM oraz perforatora taśmy typu PE 1500 firmy Facit, stojącego na pulpicie segmentu. W przyszłości będzie stosowana dziurkarka produkcji krajowej.

Dane techniczne modułu:

- szybkość dziurkowania: 150 znaków/sek
- kod dowolny
- ilość ścieżek: 5 t 8
- poziomy sygnałów: 0V i -6V
  - zasilanie: 3 x 220/380V 50 Hz; 0,3 kVA
  - wymiary: 710 x 710 x 760 mm
  - ciężar: 100 kg

## **Moduł drukarki wierszowej DW1**

Moduł drukarki wierszowej stanowi urządzenie konstrukcyjnie zwarte. Moduł składa się z kanału synchronizującego pracę drukarki z dowolną maszyną rodziny ZAM oraz samego urządzenia drukującego, napędzanego z odpowiednich układów elektronicznych. W module zastosowano drukarkę typu ICT-665 firmy ICT. Moduł wyposażono w pamięć buforową o pojemności 128 znaków. Pamięć zbudowana jest na rdzeniach ferrytowych.

Dane techniczne modułu:

- czas ładowania pamięci buforowej o pojemności 128 znaków: około 3 msek
- szybkość drukowania: 10 wierszy/sek
- ilość różnych znaków: 50
- ilość znaków w wierszu: 120
- ilość egzemplarzy druku: 1 oryginał + 4 kopie
- gęstość drukowania: 10 znaków/cal, 6 wierszy/cal
- format papieru z perforacją obrzeżną, szerokość max. 420 mm, min. 120 mm
- poziomy sygnałów: 0V i -6V
  - zasilanie: 3 × 220/380V; 50 Hz; 1,0 k7A
  - wymiary: 674 × 1638 x1411 mm
  - ciężar: 570 kg

## **Moduł czytnika kart CK2**

Moduł czytnika kart stanowi segment konstrukcyjnie zwarty. Moduł czytnika kart składa się z kanału synchronizującego pracę czytnika z dowolną maszyną rodziny ZAM oraz samego czytnika stojącego na pulpicie segmentu. W module zastosowano czytnik typu B-42 firmy Elliott.

Dane techniczne modułu:

- szybkość czytania: 400 kart/min
- rodzaj kart: 80- i 65-kolumnowe
- sposób odczytu kart: kolumnami
- rodzaj kodu: dowolny
- poziomy sygnałów: 0V i -6V
  - zasilanie: 3 × 220/380V 50 Hz; 0,4 kVA
  - wymiary: 710 × 710 × 760 mm
  - ciężar : 100 kg

## **Moduł monitora MD2**

Moduł monitora stanowi zwarty konstrukcyjnie segment; składa się z kanału synchronizującego pracę monitora z dowolną maszyną rodziny ZAM oraz samego monitora, stojącego na pulpicie segmentu. W module zastosowano dalekopis typu LO-15B firmy Lorenz. Przy pomocy klawiatury dalekopisu, operator bezpośrednio kontaktuje się z maszyną cyfrową.

Dane techniczne modułu:

- szybkość drukowania: 7 znaków/sek
- ilość różnych znaków: 52
- szerokość arkusza papieru: 210 mm
- ilość znaków w wierszu: 69
- gęstość druku: 10 znaków/cal
- dalekopis posiada:
  - dziurkarkę taśmy papierowej

- czytnik taśmy papierowej
- szybkość dziurkowania i czytania: 7 znaków na sekundę
- poziomy sygnałów: 0V i -6V
  - A zasilanie: 3 × 220/380V; 50 Hz; 0,4 kVA
  - A wymiary: 710 × 710 × 760 mm
  - A ciężar: 100 kg

### **Moduł dziurkarki kart DK1**

Moduł dziurkarki kart stanowi zwarte konstrukcyjnie urządzenie. Moduł dziurkarki kart składa się z kanału, synchronizującego pracę dziurkarki z dowolną maszyną rodziny ZAM, oraz samej dziurkarki napędzanej odpowiednimi układami elektronicznymi. W module zastosowano dziurkarkę kart typu 582 firmy ICT.

Dane techniczne modułu:

- szybkość dziurkowania: 100 kart/min
- rodzaj kart: 80 kolumnowe
- sposób dziurkowania: wierszami
- rodzaj kodu: dowolny
- poziomy sygnałów: 0V i -6V
  - zasilanie: 3 × 220/380V 50 Hz; 0,8 kVA
  - wymiary: 670 × 1600 × 1410 mm
  - ciężar: 500 kg

### **Moduł kanału automatyki KA1**

Kanał automatyki stanowi zwarty konstrukcyjnie moduł, zapewniający komunikację dowolnej maszyny ZAM z przełącznikiem sygnałów analogowych, przetwornikiem a/o i o/a, oraz źródłami i odbiornikami sygnałów cyfrowych w systemie centralnej kontroli, rejestracji i sterowania procesem przemysłowym.

Dane techniczne modułu:

- maksymalna liczba wejść analogowych: 1024
- maksymalna liczba jedno-bitowych wejść cyfrowych: 768
- maksymalna liczba wyjść analogowych: 128
- maksymalna liczba jedno-bitowych wyjść cyfrowych: 768
- poziomy sygnałów: 0V i -6V
  - zasilanie: 3 × 220/380V; 50 Hz; 0,6 kVA
  - wymiary: 710 × 710 × 760 mm
  - ciężar: 150 kg

### **Moduł kanału transmisji danych XT1**

Moduł kanału transmisji danych stanowi zwarty konstrukcyjnie segment i zapewnia bezpośrednią dwukierunkową współpracę dowolnej maszyny ZAM z łączem synchronicznego przesyłania danych.

Dane techniczne modułu:

- szybkość przesyłania: 600, 1200 i 1800 bitów/sek
- długość słowa: 30 bitów
- długość przerwy między słowami: 5 do 10 bitów
- poziom sygnałów: 0V i -6 V
  - zasilanie: 3 × 220/380 V 50 Hz; 0,4 kVA
  - wymiary: 710 × 710 × 760 mm
  - ciężar: 80 kg

## System zasilania maszyn rodziny ZAM

Zasilanie stanowi ważną część maszyn cyfrowych i decyduje w znacznej mierze o jakości całej konstrukcji. W celu spełnienia warunku modułowości konstrukcji maszyn rodziny ZAM przyjęto zdecentralizowany system zasilania modułów, odpowiadających poszczególnym urządzeniom.

Wprowadzono dwa standardowe typy zasilania ZS1 i ZS2, różniące się jedynie wymiarami i mocą wyjściową. Zasilacz opracowano w oparciu o elementy półprzewodnikowe: diody prostownicze, tranzystory i diody Zenera. Zastosowano kompensację termiczną w układach stabilizatorów napięciowych, dzięki czemu osiągnięto dużą stałość napięcia funkcji temperatury. Zasilacze wyposażono w układy tranzystorowego bezpiecznika nadmiarowego, który zapewnia dużą odporność stabilizatorów na zwarcia i przeciążenia.

Podstawowe dane techniczne:

W skład zasilaczy wchodzi cztery zespoły stabilizatorów napięć stałych:

- $U1 = +12V$ ;  $U2 = -6V$ ;  $U3 = -15V$ ;  $U4 = -15V$
- maksymalny dopuszczalny prąd obciążenia:  
 $I_{max} = 10 \text{ A}$  dla ZS1; oraz  $I_{max} = 5,0 \text{ A}$  dla ZS2
- płynna regulacja każdego napięcia w granicach:  $\pm 4\%$
- oporność wewnętrzna  $\leq 5 \text{ m}$
- napięcie tętnień sieci:  $\leq 5 \text{ mV}$
- temperaturowy współczynnik napięcia:  $K_t = 2,5 \text{ mV}/^\circ\text{C}$
- zakres temperaturowy otoczenia: od  $+10^\circ\text{C}$  do  $+40^\circ\text{C}$
- napięcie sieci:  $3 \times 220/380V \pm 5\%/-10\%$ ;  $50 \text{ Hz}$

Napięcie  $+12 \text{ V}$  jest przystosowane do regulacji w celu przeprowadzania badań marginesowych poszczególnych modułów i całej maszyny. Regulacja może odbywać się ręcznie lub automatycznie według określonego programu. Badaniom marginesowym mogą podlegać w tej samej chwili grupy modułów lub jednocześnie wszystkie moduły naraz. Zasilacze typu ZS1 przeznaczone są do zasilania modułów: centralnej części maszyny, pamięci ferrytowych, pamięci bębnowych i drukarki wierszowej. Zasilacze typu ZS2 przeznaczone są do zasilania modułów: czytnika i dziurkarki taśmy papierowej, czytnika kart, monitora i innych urządzeń zewnętrznych.

Całe zasilanie maszyn typu ZAM jest kontrolowane z centralnej tablicy rozdzielczej zwanej pulpitem zasilania PZ2. Pulpit jest wyposażony w układy zabezpieczeń, sygnalizacji, pomiarowe, oraz włączniki i wyłączniki ręczne.

## Zestawienie typowych wariantów maszyn rodziny ZAM

Każda maszyna rodziny ZAM zawiera centralną część /procesor/ i szereg modułów pamięci ferrytowych, bębnowych, ewentualnie taśmowych oraz modułów urządzeń wejścia i wyjścia.

Wyróżniamy trzy centralne części maszyn rodziny ZAM:

- CC11 dla maszyny ZAM-11
- CC21 dla maszyn ZAM-21 i ZAM-41
- CC31 dla maszyn ZAM-31 i ZAM-51

Centralna część CC11 maszyny ZAM-11 jest najmniejsza i mieści się w jednej ramie standardowej szafy. W drugiej ramie tej samej szafy znajduje się moduł pamięci ferrytowej PA05 o pojemności 4096 słów 24-bitowych. Jest to typowa szafa o następujących danych:

- zasilanie:  $3 \times 220/380 \text{ V } +5\%/-10\%$ ; 50 Hz, 0,8 kVA
- wymiary: 625 × 675 × 1700 mm
- ciężar: 260 kg

Centralna część CC21 maszyny ZAM-21 jest identyczna z centralną częścią maszyny ZAM-41 i mieści się w dwóch ramach standardowych szaf. W przeciwległych ramach mieszczą się dwa moduły ferrytowej pamięci PA05, o pojemności po 4096 słów 24-bitowych każdy. Całość składa się z dwóch standardowych szaf.

- zasilanie:  $3 \times 220/380 \text{ V } +5\%/-10\%$ ; 50 Hz, 1,6 kVA
- wymiary: 1250 × 675 × 1700 mm
- ciężar: 530 kg

Centralna część CC31 maszyny ZAM-31 jest identyczna z centralną częścią ZAM-51 i mieści się w dwóch ramach standardowych szaf. W przeciwległych ramach mieszczą się moduły pamięci ferrytowej PA05 o pojemności 4096 słów 24-bitowych każdy. Całość CC31 zajmuje dwie szafy.

- zasilanie:  $3 \times 220/380 \text{ V } +5\%/-10\%$ ; 50 Hz, 1,6 kVA
- wymiary: 1250 × 675 × 1700 mm
- ciężar: 530 kg

Przedstawione centralne części maszyn rodziny ZAM mogą być wyposażone w moduły opisane w rozdziale wyżej. Typowe zestawy tych maszyn są przedstawione w tabelicy nr 2. Oczywiście mogą być budowane i inne zestawy maszyn, w zależności od zakresu zastosowań i przede wszystkim zgodnie z życzeniem zamawiającego.



Tablica 2

Typowe zestawy maszyn rodziny ZAM

Moduły	Warianty ZAM-11			Warianty ZAM-21			ZAM-31	ZAM-41	Warianty ZAM-51	
	I	II	III	I	II	III			I	II
1	2	3	4	5	6	7	8	9	10	11
PA05	1	1	1	2	2	2	2	4	4	6
PB5		1	1	1	1	1	1	2	1	4
PT2			2*)			2*)		8	4	12
SO2	1	1	1	1	1	1	1	1	1	1
CP1						1		1	1	1
CK2			1			1		1		2
DW1			1			1	1	1	1	1
MD2	1	1	1	1	1	1	1	1	1	1
DX1								1		1
KA1	1			1						
KT1								1		2

\*) taśmy współpracują z maszyną za pośrednictwem akumulatora.

W przyszłości maszyny rodziny ZAM będą wyposażone w nowe typy pamięci, mianowicie:

- moduły pamięci ferrytowej PA07 o pojemności 8192 słów 24-bitowych
- moduły pamięci bębnowej PB6 o pojemności 128 tys. słów 24-bitowych
- moduły pamięci taśmowych PT3 o wysokich parametrach eksploatacyjnych.

# Warunki eksploatacji technicznej maszyn rodziny ZAM

## Niezawodność maszyn

Wysoką niezawodność działania maszyny uzyskano przez:

- Wyłączne stosowanie półprzewodników i ferrytów z całkowitym pominięciem układów lampowych.
- Liczne układy elektroniczne, kontrolujące poprawność pracy maszyny.
- Kontrolę programową włączoną automatycznie w podprogramy czytania i pisania danych.
- Profilaktykę uszkodzeń niezależną we wszystkich modułach maszyny poprzez ręczne lub programowe próby marginesowe.

Dzięki zastosowaniu powyższych środków średni czas użyteczny centralnej części maszyny, przy prawidłowej eksploatacji, jest nie mniejszy niż 22 godziny na dobę.

## Obsługa maszyny

- Konstrukcja maszyn ZAM zapewnia wygodną obsługę, łatwość przenoszenia oddzielnych zespołów oraz łatwość dostępu do poszczególnych elementów dla konserwacji i napraw.

Konstrukcja maszyn ZAM zapewnia wmontowanie i zainstalowanie typowego zestawu każdego typu maszyny względnie jej zdemontowanie w czasie nie dłuższym niż 12 godzin.

Celem zapewnienia prawidłowej obsługi technicznej maszyny ZAM 41 w zestawie podanym na rysunku 3 wraz z typowym zestawem urządzeń do przygotowywania danych oraz przy pracy na trzy zmiany niezbędny jest następujący personel techniczny:

- Jeden inżynier elektronik – kierownik zespołu, dobrze zaznajomiony z konstrukcją i programowaniem maszyny.
- Czterech techników elektroników wyszkolonych w obsłudze i konserwacji elektronicznych części maszyny.
- Czterech techników mechaników wyszkolonych w naprawach i konserwacji urządzeń wejścia i wyjścia oraz urządzeń do przygotowywania danych.

Obsługa maszyny ZAM 51 wymaga podobnej ilości personelu jak ZAM41. W przypadku maszyn ZAM 11 ZAM 21 lub ZAM 31 ilość techników mechaników może być odpowiednio zmniejszona, ze względu na mniejsze wyposażenie maszyny w urządzenia mechaniczne.

Każda osoba należąca do personelu obsługi technicznej powinna umieć spełniać funkcję operatora maszyny.

## **Wymagania mechaniczne i klimatyczne**

Maszyna w wykonaniu standardowym wytrzymałe przechowanie w granicach temperatur od +5°C do +45°C.

Maszyna bez taśm magnetycznych i w wykonaniu standardowym może pracować w następujących warunkach:

- Temperatura otoczenia od +15 do +30°C,
- Wilgotność względna otoczenia do 75%.

Przy stosowaniu taśm magnetycznych w pomieszczeniu powinny być spełnione następujące warunki:

- Temperatura otoczenia 20°C ± 3°C,
- Wilgotność 52% ± 5%,
- Pomieszczenie odpylane.

## **Pomieszczenie dla maszyny**

Wygodne pomieszczenie dla maszyny, nie licząc powierzchni pozostałych części ośrodka obliczeniowego, jest następujące (w metrach kwadratowych):

ZAM	11	21	31	41	51
Pomieszczenie dla maszyny	30	40	50	80	80
Powierzchnia dla obsługi technicznej	20	20	30	40	40
Razem	50	60	80	120	120

Wielkość całego ośrodka obliczeniowego zależna jest od konkretnego zastosowania maszyny i na ogół mieści się w granicach od 100 do 500 m<sup>2</sup> powierzchni użytkowej.

## **Zasilanie maszyny**

Napięcie sieci: 3 fazy 380/220 V; 50 Hz.

Zainstalowana moc zasilania dla dużego zestawu maszyny ZAM 41 lub ZAM 51 wynosi około 30 kVA, natomiast dla dużego zestawu maszyny ZAM 11, ZAM 21 lub ZAM 31 wynosi około 4 kVA.

Moc tracona na ciepło wydzielane w pomieszczeniu wynosi około 15 kW dla maszyn ZAM 41 lub ZAM 51 oraz około 2,5 kW dla maszyn ZAM 11, ZAM 21 lub ZAM 31.

## **Uwagi końcowe**

- a. Rodzina maszyn matematycznych ZAM została opracowana w instytucie Maszyn Matematycznych, Warszawa, Koszykowa 79.
- b. Produkcja doświadczalna maszyn rodziny ZAM przebiega w Zakładzie Doświadczalnym Instytutu.
- c. Produkcja seryjna maszyn ZAM przebiega we Wrocławskich Zakładach Elektronicznych ELWRO, Wrocław, Ostrowskiego 30.
- d. Początek dostaw maszyn ZAM 21 i ZAM 41 przewidziany jest w roku 1966, natomiast maszyn ZAM 11, 31 oraz 51 – w roku 1967.
- e. Rozdzielnik maszyn ZAM pozostaje w dyspozycji Pełnomocnika Rządu do Spraw Elektronicznej Techniki Obliczeniowej, Warszawa, Wawelska 3.
- f. Zastrzega się możliwość pewnych zmian w informacjach, podanych w całym niniejszym opracowaniu.

## Mgr inż. Krzysztof Bytnerowicz



Ukończył Wydział Mechaniczny Technologiczny Politechniki Warszawskiej, specjalizując się w zastosowaniach komputerów, a dokładniej – w organizacji, ekonomice i planowaniu w przemyśle budowy maszyn.

Od 1969 r. Podejmuje pierwszą pracę jako projektant i programista systemów EPD podejmuje w Stołecznym Ośrodku Elektronicznej Techniki Obliczeniowej, programując w językach SAS i SAKO na EMC ZAM-2 Gamma. Następnie jeden z programistów odpowiedzialnych za wdrożenie EMC ZAM-41 w ścisłej współpracy z IMM. Autor podprogramów w SAS dla ZAM-2 i ZAM-41 oraz rozszerzeń systemu SO-141.

Od 1972 r. zatrudniony w Pracowni Systemów Operacyjnych IMM, jeden z programistów odpowiedzialnych za opracowanie, obsługę i wdrożenie ostatnich wersji systemu SO-41 dla EMC ZAM-41. Autor szeregu programów pomocniczych, podprogramów, dekodery WE/WY i fragmentów SO-141 pisanych w języku PJEK.

Po zakończeniu prac nad ZAM-41 członek zespołu odpowiedzialnego za adaptację i wdrożenia systemu IBM OS/360 do pracy na EMC Jednolitego Systemu. Współautor Technologii Woluminów Dokumentacyjnych (TWD), rozszerzonej adaptacji systemu SMAD dla ZAM-41. Odpowiedzialny za wdrożenie i obsługę, we współpracy z IBM, systemów operacyjnych IBM OS/360 MFT, MVT, OS/VS1 i VM/370 na EMC IBM/370 model 148.

Od 1977 r. po zaprzestaniu prac nad OS/JS w IMM, po porozumieniu pomiędzy ZDO IMM i OBRI przenosi się z częścią zespołu OS/JS do OBRI, aby wdra-

zać OS/JS, zwany Technologiczną Wersją OS, w ośrodkach obliczeniowych sieci ZETO. Prowadzi szkolenia i prezentacje OS/JS i TWD, prezentuje TWD w Akademii Nauk ZSRR i w NRD (Robotron). Wdraża HASP z RTAM na IBM/360 model 50, następnie bierze udział w adaptacji HASP i RTAM do pracy w OS/JS. Współpracuje przy pracach nad RODAN z wykorzystaniem TWD, pomaga w instalowaniu u klientów.

Od 1979 r. zatrudniony w Ośrodku Obliczeniowym Uniwersytetu Warszawskiego, odpowiedzialny za obsługę systemów operacyjnych IBM i JS. Wdraża do użytku, z wykorzystaniem TWD, bibliotekę podprogramów CERNLIB (ponad 1000 podprogramów). Pracuje przy przygotowaniach do instalacji EMC JS/65 produkcji ZSRR we współpracy z producentem.

Od 1980 w MERA-SYSTEM, odpowiedzialny za wdrożenie i obsługę OS/JS, HASP i TWD w ośrodku obliczeniowym. Generuje wersje OS/JS do użytku przez ośrodki obliczeniowe Zjednoczenia MERA. Przygotowuje się do pracy konsultanta w Kuwejcie.

Od 1981 r. programista systemowy w Government Computer Centre w Kuwejcie. Obsługuje systemy operacyjne OS/370 SP, OS/VS2, i VM/370, odpowiada za obsługę JES2. Opracowuje szereg rozszerzeń (exit) JES2, prezentowanych później na SHARE i opublikowanych na taśmie modyfikacji JES2 SHARE. Kontynuuje prace nad TWD. Wykorzystując TWD przeprowadza konwersję z Fortranu CDC na IBM Fortran IV systemu OSSM (Ocean Spill Simulation System system przewidywania rozlewu ropy naftowej na morzu) autorstwa NOAA dla Kuwejckiej Agencji Ochrony Środowiska. Szkoli operatorów systemu.

Jednocześnie od 1982 programista systemowy (pół etatu) w KISR (Kuwejt Institute of Scientific Research). Obsługuje IBM OS/VS1 i szereg podsystemów pracujących pod kontrolą VM/370. Instaluje szereg pakietów graficznych dla obsługi terminali, drukarek i plotterów Tektronix. Autor rozszerzenia (exit) JES/VS dla OS/VS1 obsługującego wydruki OS/VS1 pod kontrolą VM/370. Rozszerzenie to było opublikowane w XEPHON. Autor systemu przygotowania danych i prezentacji wyników obliczeń finansowych wsadu dla OS/VS1 pod kontrolą VM/CMS.

Od 1986 r. Specjalista obsługi oprogramowania EMC produkcji Hitachi (kompatybilnych z IBM/370) w Australii, Nowej Zelandii, Azji i Ameryce Południowej (APLA) w National Advanced Systems, później Hitachi Data Systems. Udział w instalacji tych EMC u klientów, między innymi pierwsza instalacja w Tajlandii. Rozwiązywanie problemów kompatybilności oprogramowania systemowego we współpracy z producentem. Odpowiedzialny za całokształt obsługi ośrodka obliczeniowego pracującego pod kontrolą VM/SP, MVS/SP, MVS/XA wykorzystującego własne EMC AS/6600, później AS/80X3. Odpowiedzialny za telekomunikację z USA, Europą i wewnątrz Regionu Azji/Pacyfiku.

Od 1992 r. przeniesiony do EDS (Electronic Data Systems), nadal obsługując HDS. Odpowiedzialny za konwersje telekomunikacji HDS z własnej sieci do EDSNET. Zapewnienie zdalnego dostępu do sieci z komputerów personalnych z zastosowaniem PCKET/3270 za pomocą opracowanej nakładki napisanej w języku C.

Obsługa wewnętrznych sieci LAN i WAN, instalacja i obsługa oprogramowania LAN firmy Novell. Instalacja i obsługa serwerów LAN. Obsługa telekomunikacji biur HDS w Regionie. Nagradzany kilkakrotnie.

W latach 1999 i 2000 tłumacz ochotnik Olimpiady Sydney'2000. Obsługa wioślarskich zawodów przedolimpijskich (Mistrzostwa Świata) w 1999 r. w języku rosyjskim. W 2000 r. obsługa konkurencji wioślarskich w języku polskim. Autor olimpijskich słowniczków angielsko-polskich dla konkurencji jachtowych, konnych i strzeleckich.

Dwa ciekawe kontrakty między pracami: konwersja systemu rozliczania połączeń telekomunikacji morskiej i satelitarnej z PDP-11 na server OS2/Warp, obsługa routerów Cisco obsługujących australijską sieć Zurich Insurance.

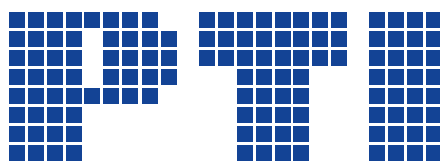
Od 2001 r. po ponad rocznej przerwie, w Computer Associates jako projektant-programista systemów Netmaster (jeden z dwóch, oprócz Netview produkcji IBM) zarządzającego telekomunikacją SNA, X.25 i TCPIP w środowisku EMC IBM i Operations, automatycznego operatora dla tego samego środowiska. Oba te australijskie systemy zostały zakupione przez Computer Associates.



Jeden z trzech programistów odpowiedzialnych za interfejs pomiędzy Systemem Operacyjnym (z/OS) a Netmaster i Operations. Autor podsystemów, modyfikacji, rozszerzeń, emulacji (Netwiew pod Netmaster) programowanych w Assembler, Rexx, wyspecjalizowanych językach (OML, NCL, itp.). Autor aplikacji (Rexx, OML) dekodujących zapamiętane (packet trace) pakiety TCP (IP i UDP). Współpracuje z IBM przy uruchomieniu podsystemu BCPI do użytku. Autor interfejsu Assembler-C-Java umożliwiającego przekazywanie informacji systemu z/OS to stron internetowych programowanych w Java. Kilkakrotnie nagradzany.

Od 2011 r. emeryt na skutek reorganizacji Computer Associates.

Translacja tekstów na język Polski dla paru aplikacji Androida.



## **POLSKIE TOWARZYSTWO INFORMATYCZNE**

Polskie Towarzystwo Informatyczne (PTI) od 1981 r. skupia specjalistów branży teleinformatycznej. Wśród członków Towarzystwa są zarówno informatycy pracujący na uczelniach, w jednostkach administracji publicznej, jak i w biznesie. PTI posiada 14 regionalnych oddziałów na terenie całego kraju. Pasjonaci poszczególnych zagadnień są dodatkowo skupieni w 12 sekcjach tematycznych PTI.

### **NAJWAŻNIEJSZE OBSZARY DZIAŁAŃ**

- przygotowywanie branżowych **KONFERENCJI I SEMINARIÓW**,
  - wydawanie specjalistycznych **PUBLIKACJI**,
  - wspieranie **ROZWOJU KOMPETENCJI** specjalistów IT,
  - promocja **WIZERUNKU** polskich informatyków,
- przygotowywanie **OPINII I EKSPERTYZ** przez Izbę Rzecznawców PTI,
- **CERTYFIKOWANIE** umiejętności komputerowych przez Polskie Biuro ECDL działające przy PTI,
- wspieranie **EDUKACJI INFORMATYCZNEJ**, w tym organizowanie konkursów dla uczniów i studentów,
- przygotowywanie obchodów **ŚWIATOWEGO DNIA SPOŁECZEŃSTWA INFORMATYCZNEGO**.



**[www.pti.org.pl](http://www.pti.org.pl)**



