

KATEDRA BUDOWY MASZYN MATEMATYCZNYCH  
POLITECHNIKA WARSZAWSKA

*[Faint, illegible text]*

I. Wstęp 1  
II. Zadanie 1  
III. Dane wyjściowe 1  
IV. PRZEDWSTĘPNY PROJEKT MASZINY UMC-100 1  
V. Obliczenia 4  
VI. Wykresy 5  
VII. Wykresy 6  
VIII. Wykresy 7

Opracowali: dr inż. J. Bańkowski  
dr inż. K. Fiałkowski  
mgr inż. W. Lutosławska  
mgr inż. J. Mieścicki  
mgr inż. M. Miszyński

Warszawa, listopad 1964 r

I. DANE OGÓLNE

Spis treści

I.	Dane ogólne	str. 1
II.	Zapis informacji	str. 1
III.	Schemat organizacji wewnętrznej	str. 3
IV.	Ważniejsze elementy jednostki centralnej	str. 3
V.	Urządzenia zewnętrzne	str. 4
VI.	Cykl pracy UMC-100	str. 5
VII.	Język wewnętrzny UMC-100	str. 6
VIII.	Programowanie	str. 10

II. ZAPIS INFORMACJI

Wprowadzenie do zapisu informacji w UMC-100 jest

składnikiem integralnym systemu i służy do

zapisu informacji w postaci

zapisu informacji w postaci

zapisu

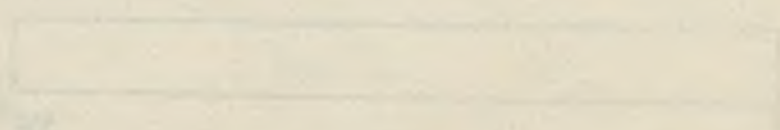
zapisu informacji w postaci

zapisu informacji w postaci

zapisu informacji w postaci

zapisu informacji w postaci

zapisu



Wskazanie na zapis informacji w UMC-100 jest

zapisu informacji w postaci

## I. DANE OGOLNE.

UMC-100 jest maszyną równoległą średniej wielkości. Głównym jej zastosowaniem są obliczenia typu naukowo-technicznego. System zapisu informacji oraz budowa i sposób kodowania rozkazów umożliwiają jednak korzystanie z maszyny przy rozwiązywaniu szerokiej klasy problemów takich, jak np. symulacja układów logicznych, obliczenia statystyczne itp.

Szybkość wykonywania operacji przesyłania i dodawania w stałym przecinku jest rzędu 35000 op/sek.

Maszyna jest zrealizowana w technice dynamicznej tranzystorowej. Częstotliwość podstawowa 200 kHz, co odpowiada podstawowemu taktowi pracy 5 $\mu$ s.

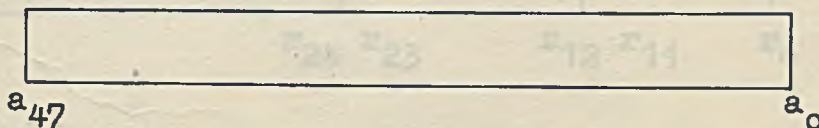
## II. ZAPIS INFORMACJI.

Podstawową jednostką informacji w UMC-100 jest 48-bitowe słowo. Może ono zawierać:

- liczbę stałoprzecinkową
- liczbę zmiennoprzecinkową
- rozkaz
- ciąg 47 zmiennych bulowskich
- 8 znaków alfanumerycznych.

Liczby są przedstawione za pomocą systemu dwujedynkowego.

Słowo zawierające liczbę stałoprzecinkową ma budowę następującą:



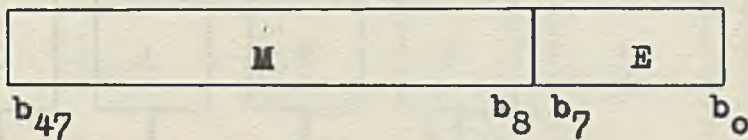
Liczba A przedstawiona za pomocą bitów  $a_i$  ma wartość:

$$A = \sum_{0}^{47} (2 a_i - 1) 2^{i-1} - \frac{1}{2}$$

i mieści się w przedziale:

$$-2^{47} \leq A \leq 2^{47} - 1$$

Słowo zawierające liczbę zmiennoprzecinkową:



Bity  $b_0 \div b_7$  stanowią wykładnik E w zapisie ze stałą  $\frac{1}{2}$ .  
Może więc on przyjmować wartość:

$$-128 \leq E \leq 127$$

Bity  $b_8 \div b_{47}$  stanowią mantysę M w zapisie dwujedynkowym

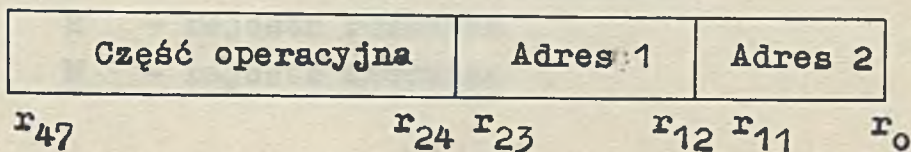
$$M = \sum_8^{47} (2 b_i - 1) 2^{i-48}$$

Mantysa jest liczbą z przedziału obustronnie otwartego:

$$-1 < M < 1$$

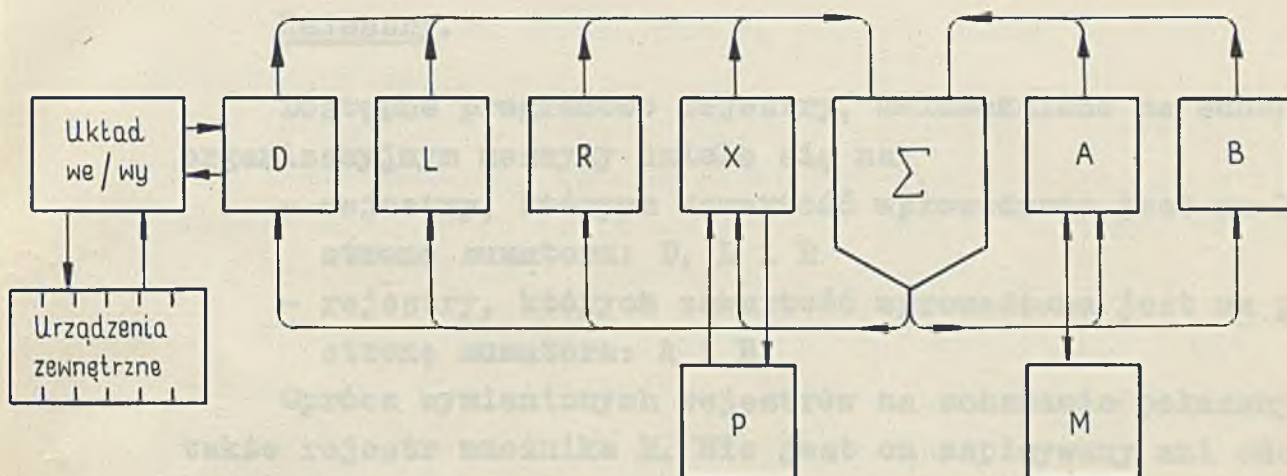
Zero przedstawia się jako liczbę o wykładniku  $E = -128$  i dowolnej mantysie M.

Słowo zawierające rozkaz:



Budowa i przeznaczenie poszczególnych elementów części operacyjnej zostanie szczegółowo omówiona w rozdziale VII.

### III. SCHEMAT ORGANIZACJI WEWNĘTRZNEJ UMC-100.



Podany schemat obejmuje jedynie te elementy maszyny, które są dostępne dla programisty.

#### Oznaczenia:

- $\Sigma$  - sumator
- X - pamięć operacyjna /ferrytowa/
- P - pamięć pomocnicza /bębnowa/
- A - rejestr akumulatora
- B - rejestr modyfikacji
- D - rejestr zewnętrzny
- L - rejestr licznika rozkazów
- R - rejestr rozkazów
- M - rejestr mnożnika

### IV. WAŻNIEJSZE ELEMENTY JEDNOSTKI CENTRALNEJ.

#### Pamięć.

UMC-100 dysponuje dwoma rodzajami pamięci:

- pamięć ferrytowa /operacyjna/ o pojemności 4096 słów i cyklu odczyt-zapis 10 $\mu$ s

- pamięć bębnowa /pomocniczy/ o pojemności  $\frac{n \times}{32768}$  słów.  $\frac{n = 1}{8}$

Między pamięcią operacyjną a pomocniczą informacja jest przesyłana w postaci bloków słów.

### Rejestry.

Dostępne programowo rejestry, uwidocznione na schemacie organizacyjnym maszyny dzielą się na:

- rejestry, których zawartość wprowadzana jest na lewą stronę sumatora: D, L i R
- rejestry, których zawartość wprowadzana jest na prawą stronę sumatora: A i B.

Oprócz wymienionych rejestrów na schemacie pokazany jest także rejestr mnożnika M. Nie jest on zapisywany ani odczytywany tak jak pozostałe. Programista może doń wprowadzić określoną zawartość jedynie z akumulatora przez przesunięcie łącznej zawartości rejestrów A i M w prawo, a odczytać jego zawartość tylko do A oraz przez analogiczne przesunięcie w lewo.

Rejestry A, M, R i D mają długość 48 bitów

Rejestr B ma długość 24 bity

Rejestr L ma długość 12 bitów.

### Sumator.

Sumator maszyny UMC-100 jest równoległy, wykonujący działania na liczbach przedstawionych za pomocą systemu dwujedynkowego. Jest on zrealizowany jako sumator sum warunkowych.

Dodawanie dwóch liczb pełnej długości trwa 4 takty, liczb 12-bitowych - 3 takty. Możliwe jest podawanie argumentów na wejście sumatora takt po takcie, nie czekając na zakończenie poprzedniej operacji.

## V. URZADZENIA ZEWNĘTRZNE.

W skład typowego zestawu urządzeń zewnętrznych UMC-100 wchodzi:

- dalekopis
- szybki czytnik taśmy perforowanej
- szybka dziurkarka

Wszystkie te urządzenia są przystosowane do taśmy pięciokanałowej.

Istnieje możliwość dołączenia czytnika i perforatora kart dziurkowanych. Ostatnie dwa urządzenia pozwolą na zastosowanie maszyny do obliczeń statystycznych, a także do rozwiązywania problemów logicznych i symulacji.

Maszyna wyposażona jest w rejestr zewnętrzny o długości 1 słowa, tak, że kompletowanie słowa po wprowadzeniu i rozkładzie go na znaki przy wyprowadzaniu, zachodzi autonomicznie i wymaga korzystania z jednostki centralnej tylko raz na jedno wprowadzane lub wyprowadzane słowo.

## VI. CYKL PRACY UMC-100.

Cykl pracy UMC-100 przy wykonywaniu typowych rozkazów, takich, jak:

(A) + (X) → A, X

(A) → X

SKOK ZWYKŁY - (R) → L

SKOK POSREDNI - (X) → L

WYKONAJ ROZKAZ - (X) → R

przedstawiony jest na załączonym schemacie.

## VII. JEZYK WEWNĘTRZNY UMC-100

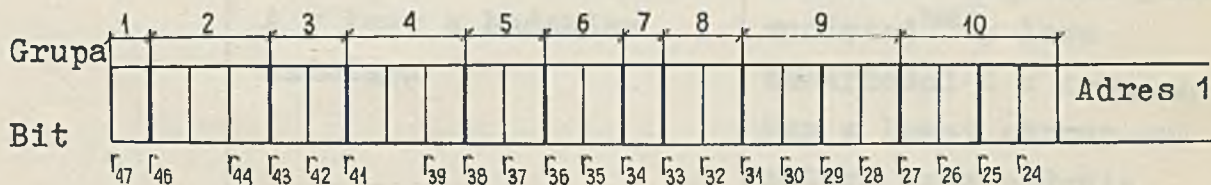
Słowo rozkazowe UMC-100 o długości 48 bitów składa się w zasadzie z:

- 24-bitowej części operacyjnej
- dwóch 12-bitowych części adresowych.

Część operacyjna rozkazu składa się z 10 grup kodowych. Kombinacja bitów wewnątrz grupy opisuje pewną elementarną czynność, jaką maszyna ma wykonać w ramach danego rozkazu. Dość duża swoboda w zestawianiu rozkazów z grup kodowych (ograniczona jednak pewnymi regułami), prowadzi do znacznego rozszerzenia zestawu operacji UMC-100 w porównaniu z maszynami o stałej liście rozkazów.

Długość grup kodowych nie przekracza 4 bitów. Sprawia to, że przedstawiony w języku wewnętrznym rozkaz ma przejrzystą budowę, a jego analiza, zarówno przez programistę jak przez układy sterowania maszyny jest stosunkowo nieskomplikowana.

Budowa części operacyjnej jest następująca:



1.  $r_{47} = 0$  oznacza operację należącą do grupy stałoprzecinkowych

$r_{47} = 1$  oznacza operację należącą do grupy zmiennoprzecinkowych.



2. Rodzaj operacji.

$r_{46}$	$r_{45}$	$r_{44}$	Grupa stałoprzecinkowa	Grupa zmiennoprzecinkowa
0	0	0	Operacja 1-go stopnia <sup>*/</sup>	Operacja 1-go stopnia <sup>*/</sup>
0	0	1	normalizacja	normalizacja
0	1	0	mnożenie	mnożenie
0	1	1	przesunięcie zawartości A w lewo cyklicznie	dzielenie
1	0	0	Badanie nadmiaru przy operacji 1-go stopnia	Zamiany liczby na postać stałoprzecinkową i od- rzucenie cz. ułamkowej
1	0	1	Operacja następnika przy operacji 1-go stopnia	Operacja bębnowa
1	1	0	Przesunięcie zawartości A w prawo	Sumowanie logiczne prze- sunętej <sup>*/</sup> w prawo zawartości A z argumen- tem z lewej strony sumatora
1	1	1	Przesunięcie zawartości A w lewo z badaniem nadmiaru	Sumowanie logiczne prze- sunętej <sup>*/</sup> w lewo zawartości A z argumen- tem z lewej strony su- matora oraz badanie nadmiaru.

<sup>\*/</sup> tzn. operacja opisywana przez pozostałe grupy kodowe  
części operacyjnej rozkazu

<sup>\*/</sup> przesuwanie długie /w tym wypadku/.

3. Odczyty argumentów z lewej strony sumatora.

$r_{43}$	$r_{42}$	Czynność
0	0	odczytaj X
0	1	odczytaj R
1	0	odczytaj L
1	1	odczytaj D

4. Zapisy.

$r_{41}$	$r_{40}$	$r_{39}$	Czynność
0	0	0	Nie zapisuj nigdzie
0	0	1	Zapisz do R
0	1	0	Zapisz do L
0	1	1	Zapisz do X
1	0	0	Zapisz do A
1	0	1	Zapisz do B
1	1	0	Zapisz do D
1	1	1	Zapisz do A i X

5. Wskaźnik odczytów.

$r_{38}$	$r_{37}$	Czynność
0	0	Nie odczytuj
0	1	Odczytaj /wprowadź/ lewy argument do sumatora
1	0	Odczytaj /wprowadź/ prawy argument do sumatora
1	1	Odczytaj argumenty z obu stron sumatora

6. Zmiana znaku.

$r_{36}$	$r_{35}$	Czynność
0	0	Nie wykonuj zmiany znaku
0	1	Zmień znak liczby wchodzącej jako lewy argument do sumatora
1	0	Zmień znak liczby wchodzącej jako prawy argument do sumatora
1	1	Jeżeli operacja stałoprzecinkowa /tzn. $r_{47}=0$ / wykonaj iloczyn logiczny. Gdy $r_{47}=1$ zmień znaki obydwu argumentów

7. Odczyty argumentów prawej strony sumatora

$r_{34} = 0$	Odczytaj rejestr A
$r_{34} = 1$	Odczytaj rejestr B

### 8. Operacje na częściach adresowych

$r_{33}$	$r_{32}$	Czynność
0	0	Wykonaj operację na słowach pełnej długości
0	1	Wykonaj operację na 1-szych częściach adresowych /bity 12-23/
1	0	Wykonaj operację na 2-gich częściach adresowych /bity 0-11/
1	1	Wykonaj operację na obu częściach adresowych /bity 0-23/

### 9. Operacje warunkowe

$r_{31}$	$r_{30}$	Czynność
0	0	-
0	1	Jeżeli wynik operacji da zero, ustaw rejestr warunku $W_1=1$
1	0	Jeżeli wynik operacji da zero, ustaw rejestr warunku $W_2=1$
1	1 <sup>*/</sup>	Jeżeli wynik operacji jest mniejszy od zera wykonaj skok do adresu 1 Jeżeli wynik operacji jest większy od zera wykonaj skok do adresu 2 Jeżeli wynik operacji jest równy zero, wykonaj następny rozkaz

\* / używane tylko w rozkazach skoków

$r_{29}$	$r_{28}$	Czynność
0	0	Wykonaj rozkaz bezwarunkowo
0	1	Wykonaj rozkaz gdy $W_1=0$
1	0	Wykonaj rozkaz gdy $W_2=0$
1	1	Wykonaj rozkaz gdy $W_1$ lub $W_2 = 0$

### 10. Modyfikacje i adresy względne

$r_{27}$	$r_{26}$	Czynność
0	0	-
0	1	Wykonaj modyfikację 1-go adresu
1	0	Wykonaj modyfikację 2-go adresu
1	1	Wykonaj modyfikację 1-go i 2-go adresu

$r_{25}$	$r_{24}$	Czynność
0	0	-
0	1	Adres 1-szy jest adresem względnym
1	0	Adres 2-gi jest adresem względnym
1	1	Oba adresy są adresami względnymi

### VIII. PROGRAMOWANIE.

Programista - użytkownik posługuje się w zasadzie wyłącznie językiem zewnętrznym. Język ten charakteryzuje się następującymi cechami ogólnymi.

1. Posiada strukturę zdaniową i jest językiem typu Compiler.
2. Przystosowany jest do obliczeń naukowo-technicznych.
3. Translator umieszczony jest na stałe w pamięci pomocniczej maszyny wraz z podprogramami najczęściej występujących funkcji.
4. Użytkownik ma możliwość rozbudowania zapisanej w pamięci na stałe biblioteki podprogramów zależnie od swoich potrzeb.

Opisana w poprzednim rozdziale budowa rozkazu UMC-100 i związana z nią metoda kodowania operacji jest dogodna dla translatora. Przejrzysty podział części operacyjnej na krótkie grupy kodowe ułatwia proces zestawiania rozkazów w języku wewnętrznym. W skład translatora wchodzi także macierz operacji dozwolonych w języku wewnętrznym maszyny.

Wprowadzenie języka zewnętrznego typu Compiler ogranicza zawsze możliwości maszyny, mimo, że sam język jest dla użytkownika instrumentem niezwykle wygodnym. Dlatego też, w maszynie UMC-100 umożliwiono programowanie w zewnętrznym języku maszyny /typu "jeden na jeden"/, tzn. komunikowanie się z maszyną za pomocą prostego programu wprowadzającego. Posługując się językiem bezpośrednim doświadczony programista

Cyfel pracy... (faint text)

może w pełni wykorzystać możliwości maszyny, budując programy bardziej optymalne /pod względem czasu wykonywania lub zajmowania miejsca w pamięci/ niż translator w czasie zestawiania programu wynikowego.

4. Blok... (faint text)

5. Wykonalny... (faint text)

Na podstawie... (faint text)

$I_0$  -... (faint text)

$I_1$  -... (faint text)

$\left. \begin{matrix} 22 \\ 20 \\ 18 \\ 16 \end{matrix} \right\}$	... (faint text) ... bity 0 - 17
	... (faint text) ... bity 18 - 23
	... (faint text) ... bity 24 - 31
	... (faint text) ... bity 32 - 35

$I_0$  -... (faint text)

$I_1$  -... (faint text)

$I_{20}$  -... (faint text)

$I_{21}$  -... (faint text)

$I_{22}$  -... (faint text)

$I_{23}$  -... (faint text)

$I_{24}$  -... (faint text)

$I_{25}$  -... (faint text)

$\left. \begin{matrix} 22 \\ 20 \\ 18 \\ 16 \end{matrix} \right\}$	... (faint text) ... bity 0 - 17
	... (faint text) ... bity 18 - 23
	... (faint text) ... bity 24 - 31
	... (faint text) ... bity 32 - 35

Cykl pracy maszyny wykonującej następujące rozkazy:

1.  $(A) + (X) \rightarrow A, X$
2.  $(A) \rightarrow X$
3. Skok zwykły  $(R) \rightarrow L$
4. Skok pośredni  $(X) \rightarrow L$
5. Wykonaj rozkaz  $(X) \rightarrow R$

Za podstawową jednostkę czasu przyjęto 1 takt = 5  $\mu$ sek

$X_o$  - bramki odczytu pamięci ferrytowej

$X_z$  - bramki zapisu pamięci ferrytowej

$\Sigma_{we}$	}	zespół bramek wejściowych sumatora bity 0 - 47
$\Sigma_o$		bity 0 - 11
$\Sigma_{12}$		bity 12 - 23
$\Sigma_{24}$		bity 24 - 35

$L_o$  - bramka odczytu licznika rozkazów

$L_z$  - bramka zapisu licznika rozkazów

$R_{pf}$  - bramki rejestru adresów pamięci ferrytowej

$R_m$  - bramki odczytu rejestrów modyfikacji

$R_{po}$  - bramki odczytu rejestru pomocniczego

$R_{pz}$  - bramki zapisu rejestru pomocniczego

$A_o$  - bramki odczytu akumulatora

$A_z$  - bramki zapisu akumulatora

$\Sigma_{wy}$	}	zespół bramek wyjściowych sumatora; bity 0 - 47
$\Sigma_{wyo}$		bity 0 - 11
$\Sigma_{wy12}$		bity 12 - 23
$\Sigma_{wy24}$		bity 24 - 35

# Cykl pracy

