

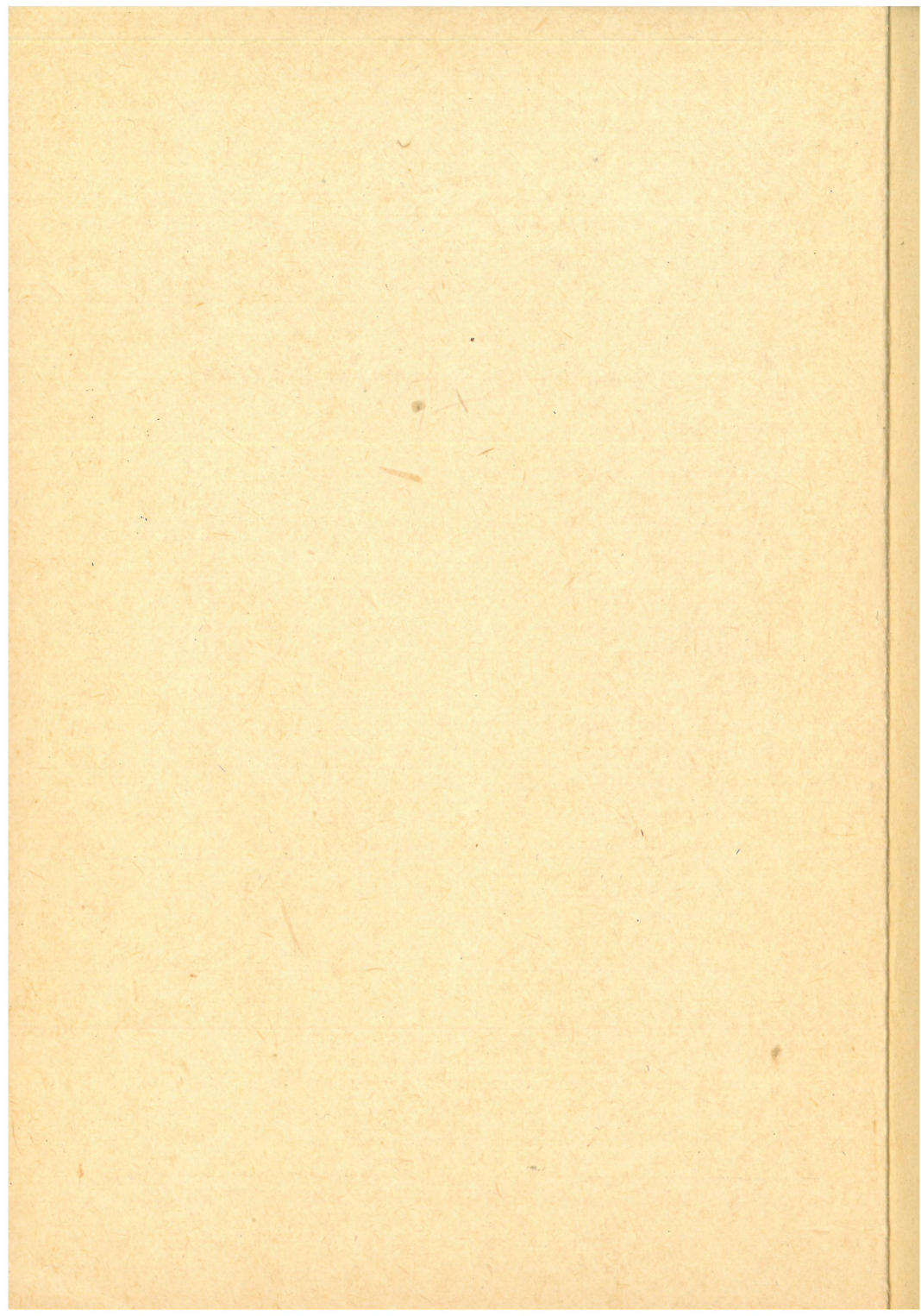
POLITECHNIKA ŁÓDZKA
OŚRODEK ELEKTRONICZNEJ TECHNIKI OBLICZENIOWEJ

EDWARD KĄCKI, STANISŁAW STARZAK

JĘZYK SYMULACYJNY AM-F-1
DLA MASZYN CYFROWYCH ODRA 1013

Łódź 1972

NAKŁADEM POLITECHNIKI ŁÓDZKIEJ



POLITECHNIKA ŁÓDZKA
OŚRODEK ELEKTRONICZNEJ TECHNIKI OBLICZENIOWEJ

EDWARD KĄCKI, STANISŁAW STARZAK

JĘZYK SYMULACYJNY AM-F-1
DLA MASZYN CYFROWYCH ODRA 1013

*Gregorz Siewczyk
Łódź, ul. 22 Lipca 10 w 94*

Łódź 1972

NAKŁADEM POLITECHNIKI ŁÓDZKIEJ

Niniejsza publikacja jest przeznaczona dla osób zajmujących się
modelowaniem procesów dynamicznych na maszynie cyfrowej
Odra 1013

Redaktor Jan Onopa

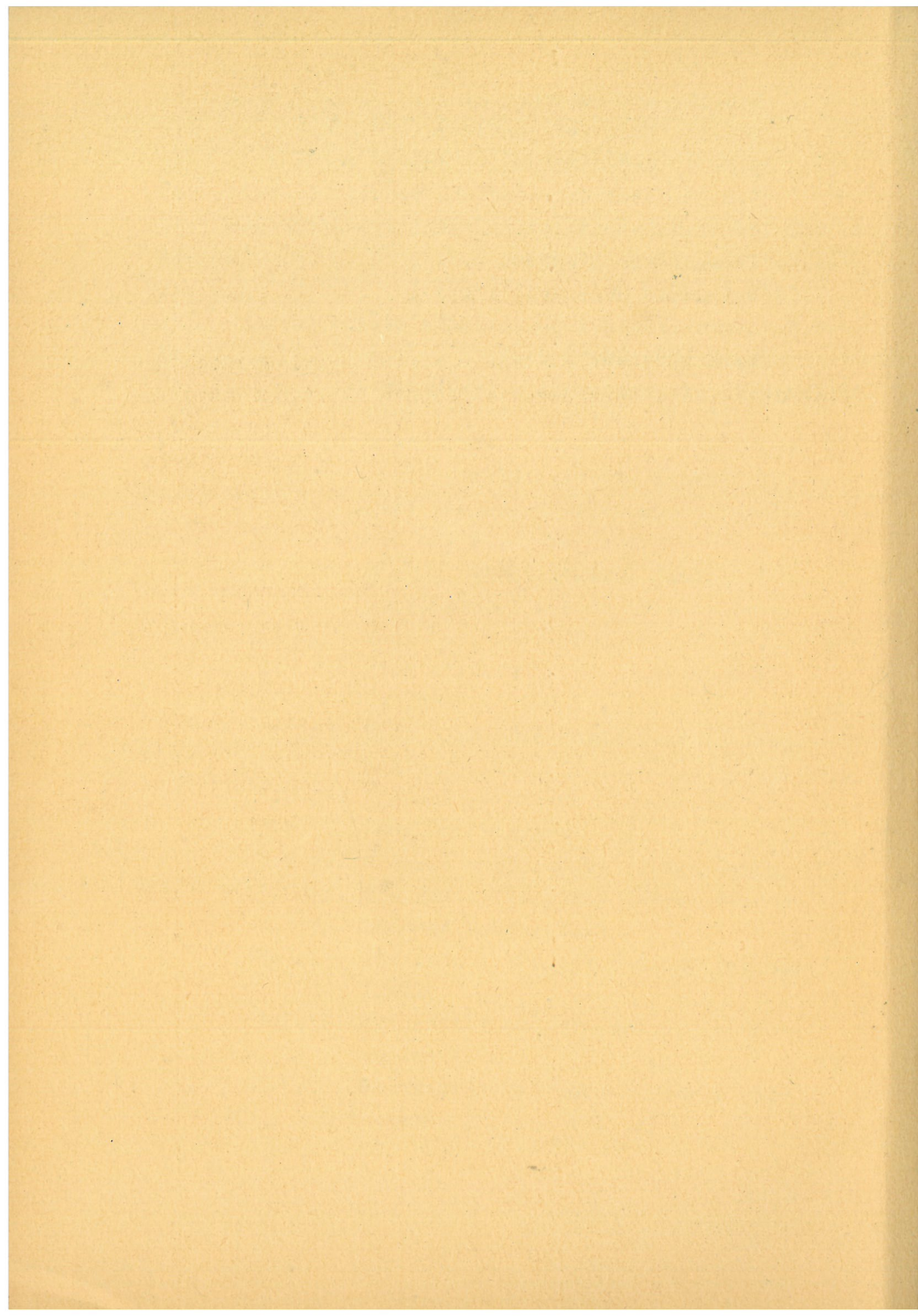
Wydanie I. Nakład 1500 egz. Ark. wyd. 2,0. Ark. druk. 3,375.
Papier powiel. kl. III 61X86. Druk ukończono w październiku 1972 r. Z. 394/72. D-10
Zakład Małej Poligrafii PŁ, Łódź, ul. Worcella 6/8

SPIS TREŚCI

Przedmowa	7
1. Wprowadzenie	9
2. Opis języka AM-F-1. Zasady programowania . . .	10
3. Opis struktury formalnej języka	14
3.1. Stałe	14
3.2. Komentarz	15
3.3. Zdanie proste programu	16
3.4. Zdanie ZMIENNE	17
3.5. Zdanie MAX	18
3.6. Zdanie LICZ	18
3.7. Zdania pomocnicze	19
3.8. Opis poszczególnych elementów funkcjonal- nych	19
3.8.1. Generator zmiennej niezależnej . .	20
3.8.2. Generator stałej	21
3.8.3. Generator zmiennej stochastycznej o rozkładzie równomiernym	21
3.8.4. Blok całkujący (integrator)	22
3.8.5. Blok opóźnienia	23
3.8.6. Blok różniczkujący	23
3.8.7. Blok sumujący dwuwejściowy	24
3.8.8. Blok sumujący czterowejściowy . .	25
3.8.9. Blok mnożenia przez stałą	25
3.8.10. Blok mnożenia zmiennych	26
3.8.11. Blok dzielenia zmiennych	26
3.8.12. Blok kwadratora	27
3.8.13. Blok podnoszenia do trzeciej potęgi	27

3.8.14.	Blok pierwiastka kwadratowego . . .	28
3.8.15.	Blok przekształtnika funkcji sinus	28
3.8.16.	Blok przekształtnika funkcji cosinus	29
3.8.17.	Blok przekształtnika funkcji tangens	29
3.8.18.	Blok przekształtnika funkcji cotangens	30
3.8.19.	Blok przekształtnika funkcji wykładniczej	30
3.8.20.	Blok przekształtnika funkcji logarytmicznej o podstawie naturalnej	31
3.8.21.	Blok przekształtnika funkcji logarytmicznej o podstawie dziesiętnej	31
3.8.22.	Blok uniwersalnego przekształtnika funkcji	32
3.8.23.	Blok przekształtnika funkcji moduł z x	33
3.8.24.	Element ze strefą nieczułości . .	34
3.8.25.	Blok ogranicznika	34
3.8.26.	Element wybierający dwuwęściowy	35
3.8.27.	Element wybierający trzywęściowy	36
3.8.28.	Element komparatora	37
3.8.29.	Przełącznik trójpołożeniowy bez histerezy	37
3.8.30.	Przełącznik dwupołożeniowy z histerezą	39
3.8.31.	Przełącznik trójpołożeniowy z histerezą	39

3.8.32. Luz mechaniczny	41
3.8.33. Blok wyprowadzania wyników . . .	41
3.8.34. Blok drukowania znaków odstępu .	42
3.8.35. Blok zatrzymujący proces liczenia	43
4. Obsługa translatora języka symulacyjnego AM-F-1	44
4.1. Tłumaczenie programu	44
4.2. Wykonanie obliczeń	45
4.3. Sygnalizacja i poprawianie błędów formal- nych programu	47
5. Przykład programowania w języku AM-F-1	53



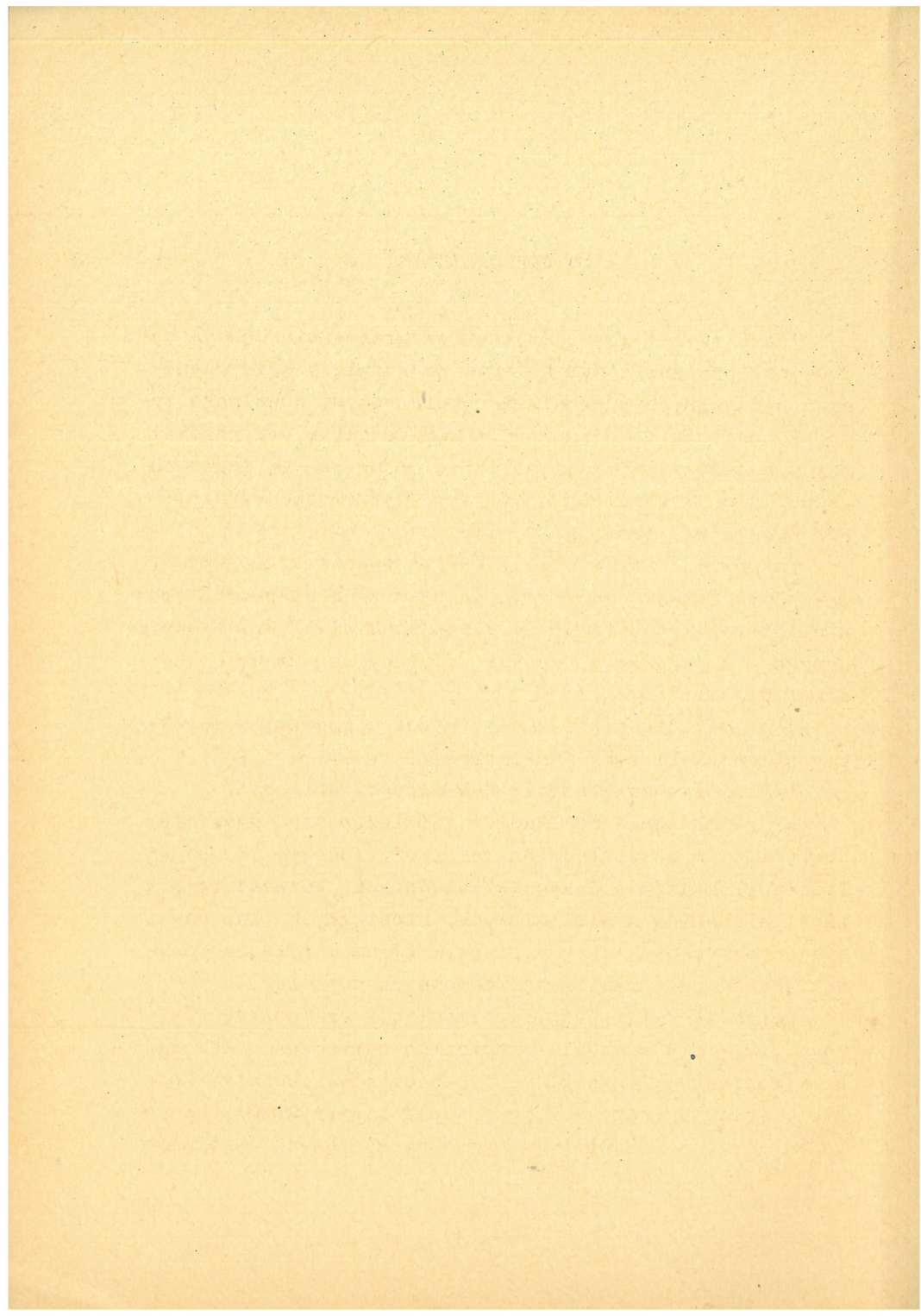
PRZEDMOWA

Translator przedstawionego języka symulacyjnego został całkowicie opracowany w Ośrodku Elektronicznej Techniki Obliczeniowej Politechniki Łódzkiej przez zespół w następującym składzie: doc. dr habil. n.t. Edward Kącki (kierownik zespołu), dr n.t. Mirosław Woźniakowski (konsultant), mgr inż. Jan Makuch, mgr inż. Krzysztof Pionka, mgr inż. Stanisław Starzak.

Prace nad translatorem wraz z wszechstronnym wypróbowaniem jego działania zostały zakończone w lutym 1972 roku.

Niniejsza publikacja ww. opracowania jest pomyślana w pierwszym rzędzie jako pomoc dydaktyczna dla studentów PŁ, a nadto może być bardzo użyteczna dla osób zajmujących się modelowaniem procesów dynamicznych na maszynach cyfrowych Odra 1013.

A u t o r z y



1. WPROWADZENIE

Język AM-F-1 jest językiem programowania maszyn cyfrowych, przeznaczonym głównie do rozwiązywania zagadnień opisywanych równaniami różniczkowymi dowolnego typu. Ze względu na stosowaną w nim metodykę programowania i sposób realizacji obliczeń można go zaliczyć do grupy języków symulacyjnych, tzw. cyfrowych symulatorów maszyn analogowych.

Zachowując takie zalety programowania analogowego, jak: duża łatwość budowania programów, dostępność i prosta interpretacja fizyczna wszystkich wielkości występujących w programie, języki symulacyjne pozwalają jednocześnie na:

- a) wyeliminowanie konieczności wyznaczania współczynników skali amplitud i czasu,
- b) znaczne zwiększenie dokładności obliczeń,
- c) zwiększenie możliwości rozwiązywania zagadnień nieliniowych ze względu na możliwość użycia dowolnej liczby elementów o danych własnościach, łatwość realizacji elementów o nieliniowych, nieciągłych lub nawet niejednoznacznych charakterystyk stanu ustalonego, dużą dokładność przy realizacji dowolnych opóźnień.

Język AM-F-1 i jego implementacja na maszynę cyfrową ODRA-1013 stanowią oryginalne opracowanie Ośrodka Elektronicznej Techniki Obliczeniowej Politechniki Łódzkiej. Jest to druga w Polsce, obok języka CEMMA, udana próba zbudowania autokodu, przeznaczonego do cyfrowego modelowania układów ciągłych.

2. OPIS JĘZYKA AM-F-1. ZASADY PROGRAMOWANIA

Proces programowania w języku AM-F-1 obejmuje następujące etapy:

- a) sformułowanie fizycznego i matematycznego opisu zjawiska,
- b) zbudowanie analogowego schematu blokowego,
- c) napisanie programu w języku AM-F-1,
- d) wprowadzenie danych i wykonanie obliczeń na maszynie cyfrowej.

Podstawę do napisania programu stanowi schemat blokowy, zbudowany zgodnie z zasadami obowiązującymi przy programowaniu maszyny analogowej. Następnie programista: przyporządkowuje każdemu elementowi operacyjnemu numer porządkowy; określa parametry opisujące dany element operacyjny zgodnie z wymaganiami języka; dla każdego elementu operacyjnego buduje jedno zdanie programu.

Z d a n i e m p r o g r a m u nazywamy ciąg znaków dalekopisowych, które zgodnie z określonym kodem, opisują jednoznacznie własności elementu operacyjnego i jego położenie na schemacie blokowym.

N a z a d a n i e p r o g r a m u składa się:

- a) numer porządkowy elementu,
- b) kod literowy operacji wykonawczej przez element,
- c) numery porządkowe elementów przyłączonych do wejścia elementu opisywanego,
- d) wagi przyporządkowane poszczególnym sygnałom wejściowym, bądź też parametry określające realizowaną operację.

Język AM-F-1 dysponuje wyszczególnionymi elementami operacyjnymi, podanymi w tabeli 1.

T a b e l a 1

Elementy operacyjne języka AM-F-1

Kod literowy	Nazwa bloku ew. realizowana funkcja
GZN	generator zmiennej niezależnej
GST	generator stałej
GZS	generator zmiennej stochastycznej o rozkładzie równomiernym
INT	element całkujący
OPZ	element opóźnienia
DIV	element różniczkujący
SUM	element sumujący, dwuwejściowy
DOD	element sumujący, czterowejściowy
MST	element mnożenia przez stałą
MNZ	element mnożenia zmiennych
DZL	element dzielenia zmiennych
KWR	element realizujący funkcję $y=x^2$
SZN	element realizujący funkcję $y=x^3$
PWD	element realizujący funkcję $y=\sqrt{x}$
SIN	element realizujący funkcję $y=\sin x$
KOS	element realizujący funkcję $y=\cos x$
TAN	element realizujący funkcję $y=\operatorname{tg} x$
CTA	element realizujący funkcję $y=\operatorname{ctg} x$
EXP	element realizujący funkcję $y=e^x$
LON	element realizujący funkcję $y=\ln x$
LOG	element realizujący funkcję $y=\lg x$
FUN	element uniwersalnego przekształtnika funkcji
MOD	element realizujący zależność $y=(x)$

Kod literowy	Nazwa bloku ew. realizowana funkcja
STN	element realizujący strefę nieczułości
OGR	element realizujący ograniczenie
EWD	element przełączający, dwuwejściowy
EWT	element przełączający, trzywejściowy
KOM	element realizujący charakterystykę komparatową
PRZ	element realizujący charakterystykę przekaźnika trójpołożeniowego bez histerezy
PHZ	element realizujący charakterystykę przekaźnika dwupołożeniowego z histerezą
PTH	element realizujący charakterystykę przekaźnika trójpołożeniowego z histerezą
LUZ	element realizujący charakterystykę luzu mechanicznego
DRU	blok wyprowadzania wyników obliczeń
CLS	blok wyprowadzania znaków zmiany linii i odstępu
KON	blok określający koniec procesu obliczeń

Każdy program w języku AM-F-1 składa się z następujących części: komentarza i programu obliczeniowego.

KOMENTARZ jest to ciąg znaków dalekopisowych służących do słownego opisu programu obliczeniowego i uzyskiwanych kolumn wyników. W programie może występować co najwyżej jeden komentarz. O ile w programie występuje komentarz, musi on poprzedzać program obliczeniowy.

PROGRAM OBLICZENIOWY jest to zbiór zdań opisujących poszczególne elementy schematu blokowego i zdań typu organizacyjnego. Każdy program musi zaczynać się od instrukcji MAX, względnie ZMIENNE i kończyć instrukcją LICZ. Innych ograniczeń na budowę programu obliczeniowego nie nakłada się.

3. OPIS STRUKTURY FORMALNEJ JĘZYKA

Rozdział niniejszy stanowi krótki, słowny opis wymagań stawianych programowi jako całość i poszczególnym jego elementom tak, aby rozwiązane zagadnienie mogło być jednoznacznie interpretowane przez maszynę cyfrową.

Zwykle opis taki ma postać ściśle matematyczną i zostaje wyrażony w specjalnym metajęzyku. Do celów praktycznych wystarczą jednak w zupełności poniższe uwagi.

W dalszej części rozdziału użyto między innymi następujących oznaczeń:

- sp - znak odstępu,
- cr - znak powrotu karetki,
- lf - znak zmiany wiersza,
- ⦿ - znak krzyża maltańskiego,
- Ⓢ - separator, znak rozdzielający.

3.1. Stałe

W języku AM-F-1 stosuje się dwa rodzaje liczb: stałoprzecinkowe i zmiennoprzecinkowe.

Liczby stałoprzecinkowe są używane do oznaczania numerów porządkowych poszczególnych bloków operacyjnych na schemacie analogowym. Są to liczby naturalne z przedziału (1,62). Liczb zmiennoprzecinkowych używa się do określania parametrów elementów operacyjnych.

Dopuszcza się dwie postacie liczb zmiennoprzecinkowych:

a) Postać normalna. Jest to taki sposób zapisu liczby, w którym część całkowitą liczby od części ułamkowej odziela kropka, np.:

$$378.25 \quad 27.51 \quad -.135$$

b) Postać półlogarytmiczna. Jest to taki sposób zapisu, w którym wartość liczby podajemy za pomocą dwóch liczb: mantysy m i cechy c :

$$N = M' c,$$

zgodnie z zależnością:

$$N = m \cdot 10^c,$$

gdzie: c - liczba całkowita, np:

$$- 12.7' - 1 = - 1.27$$

$$0.37' + 3 = 370$$

Do obu postaci liczb zmiennoprzecinkowych odnoszą się następujące uwagi:

- a) znak $+/$ przed liczbą dodatnią można pominąć,
- b) o ile liczba nie posiada części całkowitej, zera przed kropką można pominąć,
- c) o ile liczba nie posiada części ułamkowej, zera po kropce a także samą kropkę można pominąć, np:

$$+23.7 = 23.7$$

$$-0.058 = -.058$$

$$21.0'3 = 21'3 = 21000$$

3.2. Komentarz

Komentarz, czyli słowny opis programu, umieszczamy między znakami \boxtimes ... \boxtimes . Należy przy tym stosować się do następujących uwag:

a) w tekście komentarza nie wolno używać znaków krzyża maltańskiego do innych celów jak tylko do oznaczania początku i końca komentarza,

b) program, w którym komentarz nie występuje, jest programem poprawnym,

c) o ile komentarz występuje w programie, musi on znajdować się na jego początku,

d) w każdym programie może być, co najwyżej, jeden komentarz.

3.3. Zdanie proste programu

Zdanie proste programu opisuje pojedynczy element operacyjny. Składa się ono z następujących składników: numer porządkowy elementu ⑤ kod literowy elementu ⑤ lista numerów ⑤ = ⑤ lista parametrów ⑤ ,

gdzie: ⑤ - jest separatorem czyli znakiem rozdzielającym. W języku AM-F-1 przyjęto, że separatorem może być sp, lf bądź też dowolny ciąg tych znaków.

numer porządkowy elementu - jest to numer przyporządkowany danemu elementowi operacyjnemu na schemacie analogowym.

kod literowy elementu - jest to ciąg trzech liter określający realizowaną przez element operację.

lista numerów - jest to zestawienie numerów porządkowych elementów operacyjnych przyłączonych do wejścia elementu rozpatrywanego. Poszczególne numery są oddzielane znakiem separatora.

lista parametrów - jest to zestawienie parametrów charakteryzujących realizowaną przez dany element operację. Poszczególne parametry należy rozdzielić za pomocą separatora. Parametry można podawać w dwojakiej postaci: jako liczby zmiennoprzecinkowe, jako litery; w tym przypadku wartości liczbowe przyporządkowane tym literom wprowadza się do maszyny jako dane przed rozpoczęciem obliczeń.

Znakiem kończącym zdanie programu jest przecinek /, /.

3.4. Zdanie ZMIENNE

Zdanie to ma postać:

ZMIENNE © abc ... z © ,

gdzie: a,b,c,... , z są parametrami, które w zdaniach prostych występują pod postacią liter.

Należy przy tym stosować się do następujących uwag:

- a) jako parametrów literowych można używać 26 liter alfabetu łacińskiego bez indeksów,
- b) każdej litery można użyć tylko raz zarówno w zdaniu ZMIENNE, jak i w zdaniach prostych,
- c) zdanie ZMIENNE musi występować w programie bezpośrednio po komentarzu, a przed instrukcją MAX (patrz 3.5),
- d) o ile w programie nie użyto parametrów literowych, zdanie ZMIENNE można pominąć.

3.5. Zdanie MAX

Postać zdania:

$$\text{MAX } \textcircled{S} N_{\text{max}} \textcircled{S} ,$$

gdzie: N_{max} - maksymalny użyty w programie numer bloku.

Programiście stawia się następujące wymagania:

a) Bloki operacyjne można numerować dowolnie, używając jednak kolejnych numerów od 1 do N_{max} . Wówczas N_{max} jest jednocześnie ilością bloków użytych w programie.

b) Zdania opisujące poszczególne bloki mogą być wymienione w programie w dowolnej kolejności.

c) Wyniki obliczeń są wyprowadzane w kolejności rosnących numerów bloków drukowania wyników DRU.

d) W implementacji języka AM-F-1 na maszynie Odra 1013 przyjęto, że względu na bardzo małą pojemność pamięci ferrytowej, że liczba użytych jednorazowo w programie bloków nie może być większa od 62:

$$N_{\text{max}} \leq 62 .$$

e) Zdanie MAX występuje w programie bezpośrednio po zdaniu ZMIENNE lub po komentarzu.

3.6. Zdanie LICZ

Postać zdania:

$$\text{LICZ } \textcircled{S} = \textcircled{S} P1 \textcircled{S} ,$$

gdzie: $P1 = \Delta t_{\text{dru}}$ - żądany odcinek czasu maszynowego między kolejnymi wyprowadzeniami wyników.

Przy czym wyniki wyprowadzane są w rzeczywistości co czas:

$$\Delta t \cdot \text{ENT} \left[\frac{\Delta t_{dru} + \frac{1}{2} \Delta t}{\Delta t} \right],$$

gdzie: Δt - krok całkowania (obliczeń).

Zdanie LICZ jest ostatnim zdaniem programu.

3.7. Zdanie pomocnicze

Do sterowania pracą czytnika w trakcie tłumaczenia programu używa się następujących rozkazów:

CZEKAJ © , - po przetłumaczeniu tego rozkazu maszyna drukuje na dalekopisie słowo CZEKAM i przestawia się na czytanie z dalekopisu. Gotowość wprowadzania zdań programu z dalekopisu jest sygnalizowana poprzez wydrukowanie znaku ☉.

CZYTAJ © , - napisanie tej instrukcji na dalekopisie powoduje dalsze czytanie i tłumaczenie programu wprowadzanego przez czytnik.

Instrukcji CZEKAJ i CZYTAJ można używać do zatrzymania czytnika w celu zmiany tasiemki z programem lub do napisania części programu za pomocą dalekopisu. Instrukcji CZYTAJ używa się także po poprawieniu na dalekopisie błędnego zdania programu.

3.8. Opis poszczególnych elementów funkcjonalnych

Podaną w punkcie 3.1 gramatykę języka AM-F-1 należy uzupełnić przez podanie dodatkowych informacji dotyczących opisu poszczególnych elementów operacyjnych za pomocą zdefiniowanego już zdania prostego programu. Zamieszczony poniżej opis każdego elementu operacyjnego zawiera:

- a) pełną postać opisującego go zdania,
- b) matematyczny opis realizowanej operacji,
- c) szczegółowy opis poszczególnych elementów zdania,
- d) propozycję symbolu graficznego elementu używanego na schemacie blokowym.

Jak wynika z punktu 3.1, zdania programu zawierają informację dotyczące własności elementów i topologii układu. Zastosowano przy tym następujące oznaczenia:

N - numer własny bloku,

N z indeksem - numery bloków przyłączonych do wejścia bloku danego,

P z indeksem - parametry charakteryzujące własności bloku.

3.8.1. Generator zmiennej niezależnej

Postać zdania:

$$N \text{ GZN} = P_1 P_2 ,$$

np.:

$$5 \text{ GZN} = 0. 0.005 ,$$

gdzie: $P_1 = t_0$ - czas początkowy,

$P_2 = \Delta t$ - krok obliczeń.

Realizowana funkcja:

$$y = t_0 + i \cdot \Delta t ,$$

gdzie: i - numer kolejnego kroku obliczeniowego.

Warunek: $P_2 > 0$.

Symbol:



Rys. 1

Blok GZN pełni w programie rolę zegara. W każdym programie musi znajdować się jeden blok GZN. Można z niego także korzystać jako ze źródła sygnału narastającego liniowo.

3.8.2. Generator stałej

Postać zdania:

$$N \text{ GST} = P1 ,$$

np.:

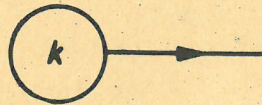
$$2 \text{ GST} = 1. ,$$

gdzie: P1 - stała generowana przez blok.

Realizowana funkcja:

$$y = k \quad k = \text{const.}$$

Symbol:



Rys. 2

3.8.3. Generator zmiennej stochastycznej o rozkładzie równomiernym

Postać zdania:

$$N \text{ GZS} = P1 P2 ,$$

np.:

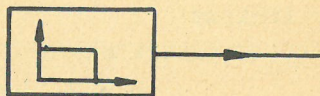
$$12 \text{ GZS} = 3. 0.5 ,$$

gdzie: $0 < P1 < 4,$

$0 < P2 < 1.$

Blok generuje ciąg liczb losowych o rozkładzie równomiernym z przedziału $\langle 0,1 \rangle$.

Symbol:



Rys. 3

3.8.4. Blok całkujący (integrator)

Postać zdania:

$$N \text{ INT } N1 = P1 \ P2 ,$$

np.:

$$7 \text{ INT } 5 = 0.2 \ 15. ,$$

gdzie: $N1$ - nr porządkowy bloku przyłączonego do wejścia,

$P1 = y$ - wartość początkowa całki,

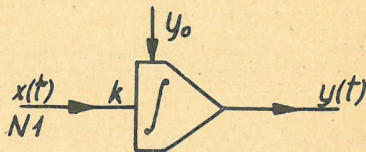
$P2 = k$ - współczynnik wzmocnienia.

Realizowana zależność:

$$y = y_0 + k \int_{t_p}^t x(t) dt,$$

gdzie: $y_0 = y(t_p)$.

Symbol:



Rys. 4

Całkowanie odbywa się metodą Eulera:

$$Y_i = Y_{i-1} + \frac{\Delta t}{2} k(3X_i - X_{i-1}),$$

gdzie: $i = 1, 2, \dots, n, \dots$

$$x_0 = x_1.$$

3.8.5. Blok opóźnienia

Postać zdania:

$$N \text{ OPZ } N1 = P1 \text{ P2},$$

np.:

$$47 \text{ OPZ } 15 = 5. 12.5' -7$$

gdzie: $P1 = y_0$ - wartość początkowa,

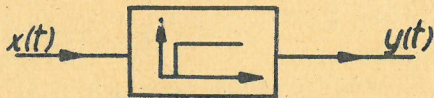
$P2 = t_0$ - czas opóźnienia.

Realizowana funkcja:

$$y = \begin{cases} y_0 & \text{dla } 0 \leq t < t_0 \\ x(t-t_0) & \text{dla } t \geq t_0. \end{cases}$$

Warunek: $P2 = t_0 > 0$.

Symbol:



Rys. 5

3.8.6. Blok różniczkujący

Postać zdania:

$$N \text{ DIV } N1 = P1,$$

np.:

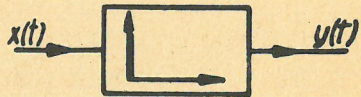
$$10 \text{ DIV } 31 = 0.27 ,$$

gdzie: $P1 = y_0$ - wartość początkowa pochodnej.

Realizowana funkcja:

$$y = \begin{cases} y_0 & \text{dla } t = 0 \\ \frac{dx}{dt} & \text{dla } t > 0 . \end{cases}$$

Symbol:



Rys. 6

3.8.7. Blok sumujący dwuwęściowy

Postać zdania:

$$N \text{ SUM } N1 \ N2 = P1 \ P2 ,$$

np.:

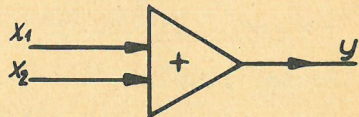
$$62 \text{ SUM } 51 \ 48 = 1.5 \ -3. ,$$

gdzie: $P1 = k_1$ - waga przyporządkowana zmiennej x_1 , $P2 = k_2$ - waga przyporządkowana zmiennej x_2 .

Realizowana funkcja:

$$y = k_1 \cdot x_1 + k_2 \cdot x_2 .$$

Symbol:



Rys. 7

3.8.8. Blok sumujący czterowejściowy

Postać zdania:

$$N \text{ DOD } N1 \ N2 \ N3 \ N4 = P1 \ P2 \ P3 \ P4 ,$$

np.:

$$15 \text{ DOD } 7 \ 10 \ 1 \ 05 = 1. \ 2. \ -3. \ 0.15 ,$$

gdzie: $P1 = k_1$ - waga przyporządkowana zmiennej x_1 ,

$P2 = k_2$ - waga przyporządkowana zmiennej x_2 ,

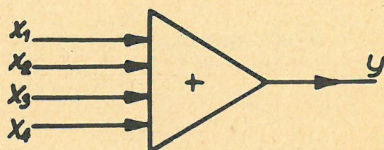
$P3 = k_3$ - waga przyporządkowana zmiennej x_3 ,

$P4 = k_4$ - waga przyporządkowana zmiennej x_4 .

Realizowana funkcja:

$$y = \sum_{i=1}^4 k_i \cdot x_i .$$

Symbol:



Rys. 8

U w a g i: 0 ile do wejścia elementu doprowadzamy mniej niż 4 sygnały, to do wejść nie wykorzystanych należy przyłączyć dowolny blok występujący w programie i przypisać im wagę 0.0; operacja SUM jest w maszynie realizowana dwukrotnie szybciej niż operacja DOD.

3.8.9. Blok mnożenia przez stałą

Postać zdania:

$$N \text{ MST } N1 = P1 ,$$

np.:

$$07 \text{ MST } 60 = 12. ,$$

gdzie: $P1 = k = \text{stała}$.

Realizowana funkcja:

$$y = k \cdot x$$

Symbol:



Rys. 9

3.8.10. Blok mnożenia zmiennych

Postać zdania:

$$N \text{ MNZ } N1 \text{ } N2 \text{ ,}$$

np.:

$$6 \text{ MNZ } 15 \text{ } 24 \text{ ,}$$

Realizowana funkcja:

$$y = x_1 \cdot x_2$$

Symbol:



Rys. 10

3.8.11. Blok dzielenia zmiennych

Postać zdania:

$$N \text{ DZL } N1 \text{ } N2 \text{ ,}$$

np.:

$$1 \text{ DZL } 15 \text{ } 20 \text{ ,}$$

Realizowana funkcja:

$$y = \frac{x_1}{x_2} .$$

Symbol:



Rys. 11

3.8.12. Blok kwadratora

Postać zdania:

N KWR N1 ,

np.:

7 KWR 43 ,

Realizowana zależność:

$$y = x^2 .$$

Symbol:



Rys. 12

3.8.13. Blok podnoszenia do trzeciej potęgi

Postać zdania:

N SZN N1 ,

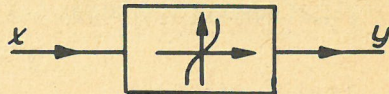
np.:

20 SZN 07 ,

Realizowana funkcja:

$$y = x^3 .$$

Symbol:



Rys. 13

3.8.14. Blok pierwiastka kwadratowego

Postać zdania:

N PWD N1 ,

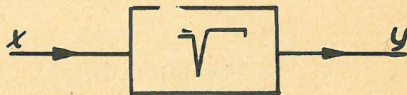
np.:

3 PWD 5 ,

Realizowana funkcja:

$$y = \sqrt{x} .$$

Symbol:



Rys. 14

3.8.15. Blok przekształtnika funkcji sinus

Postać zdania:

N SIN N1 ,

np.:

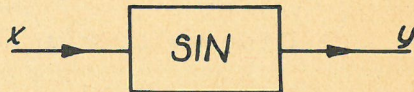
7 SIN 35 ,

Realizowana funkcja:

$$y = \sin x ,$$

gdzie: x - argument w radianach.

Symbol:



Rys. 15

3.8.16. Blok przekształtnika funkcji cosinus

Postać zdania:

$$N \text{ COS } N1 ,$$

np.:

$$38 \text{ COS } 37 ,$$

Realizowana funkcja:

$$y = \cos x ,$$

gdzie: x - argument w radianach.

Symbol:



Rys. 16

3.8.17. Blok przekształtnika funkcji tangens

Postać zdania:

$$N \text{ TAN } N1 ,$$

np.:

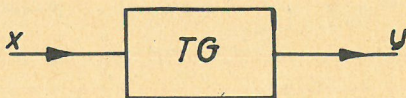
$$27 \text{ TAN } 26 ,$$

Realizowana zależność:

$$y = \text{tg } x ,$$

gdzie: x - argument w radianach.

Symbol:



Rys. 17

3.8.18. Blok przekształtnika funkcji cotangens

Postać zdania:

N CTA N1 ,

np.:

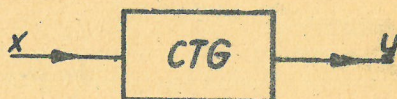
2 CTA 17 ,

Realizowana funkcja:

$$y = \operatorname{ctg} x ,$$

gdzie: x - argument w radianach.

Symbol:



Rys. 18.

3.8.19. Blok przekształtnika funkcji wykładniczej

Postać zdania:

N EXP N1 ,

np.:

19 EXP 1 ,

Realizowana funkcja:

$$y = e^x ,$$

gdzie: e - podstawa logarytmów naturalnych.Ze względu na zakres liczb używanych w maszynie cyfrowej ODRA 1013 musi być spełniona nierówność $x < 176$.

Symbol:



Rys. 19

3.8.20. Blok przekształtnika funkcji logarytmicznej
o podstawie naturalnej

Postać zdania:

N LON N1 ,

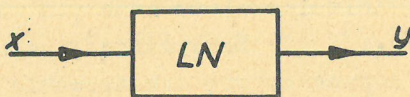
np.:

23 LON 22 ,

Realizowana funkcja:

$$y = \ln x.$$

Symbol:



Rys. 20

3.8.21. Blok przekształtnika funkcji logarytmicznej
o podstawie dziesiętnej

Postać zdania:

N LOG N1 ,

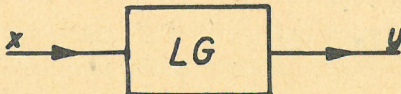
np.:

3 LOG 21 ,

Realizowana funkcja:

$$y = \log_{10} x = \lg x .$$

Symbol:



Rys. 21

3.8.22. Blok uniwersalnego przekształtnika funkcji

Postać zdania:

$$N \text{ FUN } N1 = P1 \ P2 \ P3 \ \dots \ P(N-1) \ PN ,$$

gdzie: $P(K-1)$, PK są współrzędnymi punktów aproksymowanej funkcji na płaszczyźnie x , y podawanymi w kolejności rosnących wartości zmiennej niezależnej x ,

np.:

$$3 \text{ FUN } 5 = 3. \ 12. \ 4. \ 15. \ 5. \ 17. \ 7. \ 18. ,$$

Minimalna ilość zadanych punktów - 2(4 parametry x_1, y_1, x_2, y_2). Maksymalna liczba zadanych punktów - 60. Nie dopuszcza się deklarowania w bloku FUN parametrów w postaci zmiennych literowych.

Realizowana funkcja:

dla $x_{k-1} \leq x \leq x_k$:

$$y = y_{k-1} + (x - x_{k-1}) \frac{y_k - y_{k-1}}{x_k - x_{k-1}} ,$$

w przypadku gdy wartość zmiennej x znajdzie się w trakcie obliczeń poza deklarowanym przedziałem tzn.:

$$x < x_1 \quad \text{lub} \quad x > x_N ,$$

wartość wyjściowa elementu może być ekstrapolowana zgodnie z życzeniem programisty za pomocą dwóch punktów brzegowych

dla $x < x_1$:

$$y = y_1 + (x - x_1) \frac{y_2 - y_1}{x_2 - x_1} ,$$

dla $x > x_N$:

$$y = y_{N-1} + (x - x_{N-1}) \frac{y_N - y_{N-1}}{x_N - x_{N-1}},$$

Symbol:



Rys. 22

U w a g a: W wymienionych poniżej blokach podawane parametry są zgodnie z poniższą, łatwą do zapamiętania regułą:

- warunek początkowy,
- punkty na osi x w kierunku rosnących wartości x ,
- punkty na osi y w kierunku rosnących wartości y ,
- pozostałe parametry.

3.8.23. Blok przekształtnika funkcji moduł z x

Postać zdania:

$$N \text{ MOD } N1,$$

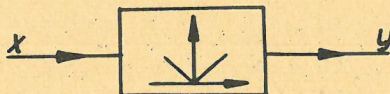
np.:

$$5 \text{ MOD } 47,$$

Realizowana funkcja:

$$y = |x|.$$

Symbol:



Rys. 23

3.8.24. Element ze strefą nieczułości

Postać zdania:

$$N \text{ STN } N1 = P1 \text{ } P2 \text{ } P3 ,$$

gdzie: $P1 = a,$

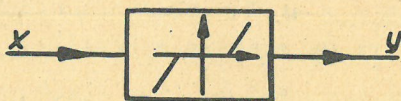
$P2 = b,$

$$P3 = k = \operatorname{tg} \alpha ,$$

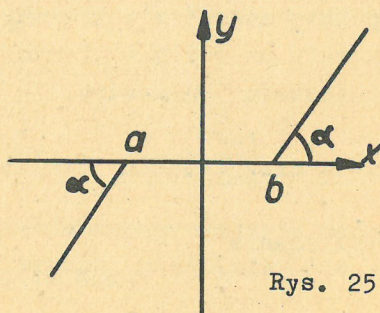
np.:

$$7 \text{ STN } 5 = -1. \ 2. \ 3. ,$$

Symbol:



Rys. 24



Rys. 25

Warunek: $a \leq b.$

3.8.25. Blok ogranicznika

Postać zdania:

$$N \text{ OGR } N1 = P1 \text{ } P2 \text{ } P3 \text{ } P4 ,$$

gdzie: $P1 = a,$

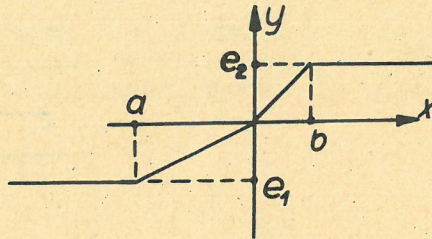
$P2 = b,$

$P3 = e_1,$

$P4 = e_2,$

np.:

$$7 \text{ OGR } 43 = -3. 7. -1 1. ,$$

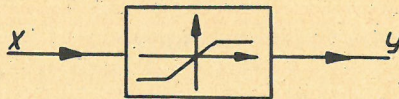


Rys. 26

Realizowana funkcja:

$$y = \begin{cases} e_1 & \text{dla } x < a \\ e_1/a \cdot x & \text{dla } a \leq x < 0 \\ e_2/b \cdot x & \text{dla } 0 \leq x < b \\ e_2 & \text{dla } x > b. \end{cases}$$

Symbol:



Rys. 27

Warunki: $a \leq 0$, $b \geq 0$.

3.8.26. Element wybierający dwuwęściowy

Postać zdania:

$$N \text{ EWD } N1 \text{ } N2 \text{ } N3 = P1 ,$$

gdzie: $P1 = a$,

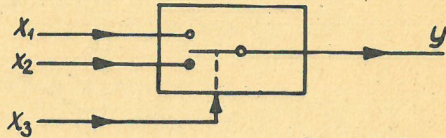
np.:

$$17 \text{ EWD } 54 \text{ } 6 \text{ } 12 = 3. ,$$

Realizowana funkcja:

$$y = \begin{cases} x_2 & \text{dla } x_3 \leq a \\ x_1 & \text{dla } x_3 > a. \end{cases}$$

Symbol:



Rys. 28

3.8.27. Element wybierający trzywejściowy

Postać zdania:

$$N \text{ EWT } N_1 \ N_2 \ N_3 \ N_4 = P_1 \ P_2 ,$$

gdzie: $P_1 = a,$ $P_2 = b,$

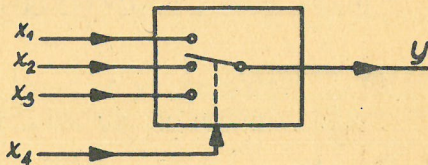
np.:

$$13 \text{ EWT } 5 \ 6 \ 7 \ 8 = -1 \ 5. ,$$

Realizowana funkcja:

$$y = \begin{cases} x_1 & \text{dla } x_4 > a+b \\ x_2 & \text{dla } a-b < x_4 \leq a+b \\ x_3 & \text{dla } x_4 \leq a-b. \end{cases}$$

Symbol:



Rys. 29

3.8.28. Element komparatora

Postać zdania:

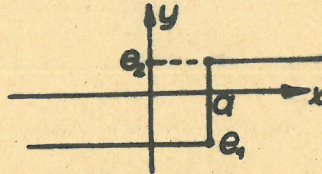
$$N \text{ KOM } N1 = P1 \text{ } P2 \text{ } P3 \text{ } P4 ,$$

gdzie: $P1 = y_0$ - wartość początkowa,

$$P2 = a,$$

$$P3 = e_1,$$

$$P4 = e_2,$$



Rys. 30

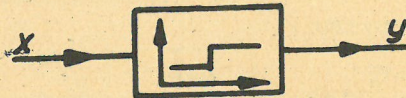
np.:

$$15 \text{ KOM } 7 = -1. \ 3. \ -1.1 \ 2. ,$$

Realizowana funkcja:

$$y = \begin{cases} e_1 & \text{dla } x < a \\ e_2 & \text{dla } x > a \\ \text{wartość z kroku poprzedniego dla } x = a. \end{cases}$$

Symbol:



Rys. 31

3.8.29. Przekaznik trójpołożeniowy bez histerezy

Postać zdania:

$$N \text{ PRZ } N1 = P1 \text{ } P2 \text{ } P3 \text{ } P4 \text{ } P5 ,$$

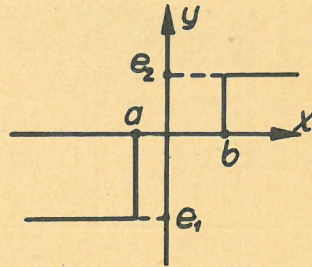
gdzie: $P1 = y_0$ - wartość początkowa,

$P2 = a$,

$P3 = b$,

$P4 = e_1$,

$P5 = e_2$,



Rys. 32

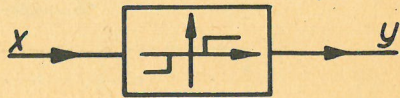
np.:

1 PRZ 4 = 3. -2.5 1. -1.5 3. ,

Realizowana funkcja:

$$y = \begin{cases} e_1 & \text{dla } x < a \\ 0 & \text{dla } a < x < b \\ e_2 & \text{dla } x > b \\ \text{wartość z chwili poprzedniej} & \text{dla } x = a \text{ i } x = b. \end{cases}$$

Symbol:



Rys. 33

warunek: $a \leq b$.

3.8.30. Przekaznik dwupołożeniowy z histerezą

Postać zdania:

$$N \text{ PHZ } N1 = P1 \text{ } P2 \text{ } P3 \text{ } P4 \text{ } P5 ,$$

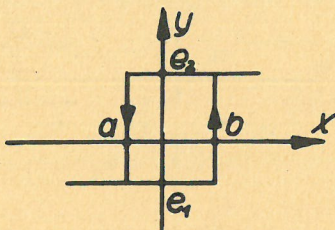
gdzie: $P1 = y_0$ - wartość początkowa,

$$P2 = a,$$

$$P3 = b,$$

$$P4 = e_1,$$

$$P5 = e_2,$$



Rys. 34

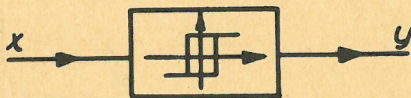
np.:

$$27 \text{ PHZ } 13 = -7.5 \text{ } -1. \text{ } 2. \text{ } -7.5 \text{ } 0.5 ,$$

Realizowana funkcja:

$$y = \begin{cases} e_1 & \text{dla } x < a \\ e_2 & \text{dla } x > b \\ \text{wartość poprzednia} & \text{dla } a < x < b. \end{cases}$$

Symbol:



Rys. 35

Warunek: $a \leq b$.

3.8.31. Przekaznik trójpołożeniowy z histerezą

Postać zdania:

$$N \text{ PTH } N1 = P1 \text{ } P2 \text{ } P3 \text{ } P4 \text{ } P5 \text{ } P6 \text{ } P7 ,$$

gdzie: $P_1 = y_0$ - wartość początkowa,

$P_2 = a$,

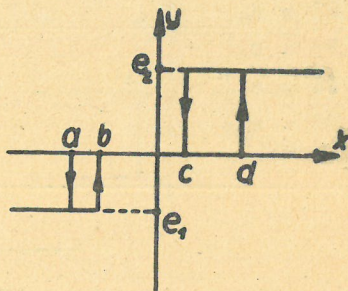
$P_3 = b$,

$P_4 = c$,

$P_5 = d$,

$P_6 = e_1$,

$P_7 = e_2$,



Rys. 36

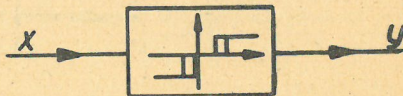
np.:

3 PTH 4 = -1 -2. -1. 1 2. -1. 1. ,

Realizowana funkcja:

$$y = \begin{cases} e_1 & \text{dla } x < a \\ e_2 & \text{dla } x > d \\ 0 & \text{dla } b < x < c \\ \text{wartość poprzednia dla } a < x < b \text{ i } c < x < d. \end{cases}$$

Symbol:



Rys. 37

Warunki: $a \leq b \leq c \leq d$.

3.8.32. Luz mechaniczny

Postać zdania:

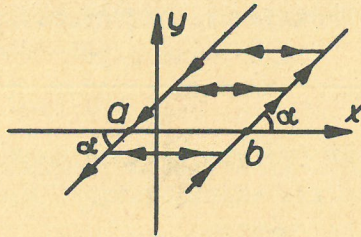
$$N \text{ LUZ } N1 = P1 \ P2 \ P3 \ P4 ,$$

gdzie: $P1 = y_0$ - wartość początkowa,

$$P2 = a,$$

$$P3 = b,$$

$$P4 = k = \operatorname{tg}\alpha,$$

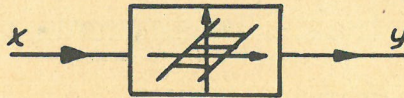


Rys. 38

np.:

$$3 \text{ LUZ } 17 = 4. \ -1. \ 2. \ 7. ,$$

Symbol:



Rys. 39

Warunki: $a < b$, $k \neq 0$.

3.8.33. Blok wyprowadzania wyników

Postać zdania:

$$N \text{ DRU } N1 = P1 ,$$

gdzie: $P1$ - parametr wskazujący na sposób drukowania zmiennej określonej przez adres $N1$.

Dopuszcza się dwie postacie liczb zmiennoprzecinkowych wyprowadzanych z maszyny. Deklarując parametr P1 w postaci:

m.n - uzyskuje się drukowanie liczby w postaci normalnej, tzn. drukowane jest m cyfr przed kropką dziesiętną i n cyfr po kropce.

np.:

6 DRU 15 = 3.1 ,

m' - uzyskuje się drukowanie liczby w postaci półlogarytmicznej:

np.:

6 DRU 15 = 4' ,

W obu przypadkach po znakach cyfr drukuje się dodatkowo dwa znaki odstępu. Wyniki wyprowadzane są tylko na perforator.

W programie musi wystąpić przynajmniej jeden element typu DRU.

Warunki: dla liczby w postaci normalnej $0 < m + n \leq 9$,
dla liczby w postaci półlogarytmicznej $0 < m \leq 9$.

3.8.34. Blok drukowania znaków cr, lf, cr, sp

Postać zdania:

N CLS ,

Blok ten służy do uzyskiwania odpowiedniej szaty graficznej wyników. Należy go stosować wówczas, gdy wyprowadzana jednorazowo seria wyników nie mieści się w jednej linii. Blok wyprowadza na perforator znaki:

cr lf cr oraz 13 znaków sp.

3.8.35. Blok zatrzymujący proces liczenia

Postać zdania:

$$N \text{ KON } N1 = P1 \text{ P2 } P3 ,$$

gdzie: $P1 = y_{\min}$,

$P2 = y_{\max}$,

$P3 = t_{\text{kon}}$.

Blok ten zatrzymuje pracę maszyny w przypadku, gdy znajdzie jedna z trzech możliwości:

a) wartość wielkości wyjściowej bloku N1 osiągnie poziom niższy od y_{\min} ,

b) wartość wielkości wyjściowej bloku N1 osiągnie poziom wyższy od y_{\max} ,

c) czas maszynowy przekroczy wartość t_{kon} .

W programie musi występować tylko jeden element KON.

4. OBSŁUGA TRANSLATORA JĘZYKA SYMULACYJNEGO

AM-F-1

Napisany na dalekopisie i wydziurkowany na pięciokanałowej taśmie perforowanej program podlega dwóm etapom przetwarzania na maszynie cyfrowej: a) tłumaczenie i b) wykonywanie obliczeń.

W trakcie procesu tłumaczenia program tłumaczący tzw. translator dokonuje zamiany rozkazów języka AM-F-1 na instrukcje zrozumiałe dla maszyny i organizuje cały proces obliczeń. W drugim etapie, po wprowadzeniu danych do pamięci maszyny, rozpoczyna się proces obliczeń.

4.1. Tłumaczenie programu

Po podłożeniu taśmy z programem pod fotokomórkę czytelnika i włączeniu dalekopisu towarzyszącego tzw. monitora rozpoczyna się proces tłumaczenia. W tym celu należy dokonać następujących czynności:

- a) zerować akumulator,
- b) wcisnąć na klawiszach rejestry rozkazów nr 10000 (ósemkowo)
- c) załadować rejestr rozkazów,
- d) przycisnąć przycisk START.

Po tych czynnościach maszyna wydrukuje na monitorze napis AM-F-1, po czym przystąpi do tłumaczenia programu. Przy uruchamianiu trudniejszych programów może okazać się korzystne tłumaczenie krokowe. W tym przypadku, po przetłumaczeniu jednej instrukcji, maszyna zatrzymuje się i kontynuuje tłumaczenie po ponownym na-

ciśnięciu klawisza START. Pracę kropkową translatora uzyskujemy przez wciśnięcia na klawiaturze akumulatora przycisku nr 38.

Ewentualne błędne ilustracje programu są wypisywane na monitorze w trakcie tłumaczenia (patrz 4.3).

Po zakończeniu tłumaczenia czytnik zatrzymuje się, a translator przechodzi do organizowania procesu obliczeń. Na tym etapie wykrywane są tzw. pętle arytmetyczne (patrz 4.3). O ile program jest całkowicie poprawny, na monitorze jest drukowany napis:

PROGRAM POPRAWNY
Z POWAŻANIEM
ODRA 1013

po czym maszyna zatrzymuje się.

4.2. Wykonywanie obliczeń

Przed przystąpieniem do drugiego etapu przetwarzania należy do maszyny wprowadzić wartości tych parametrów, dla których w programie zostały zarezerwowane oznaczenia literowe. W tym celu należy:

a) napisać na dalekopisie taśmę z danymi, trzymając się następujących dwóch zasad:

- wartości liczbowe odpowiadające poszczególnym literom należy pisać w kolejności zgodnej z deklaracją w rozkazie ZMIENNE,

- poszczególne liczby należy oddzielać znakiem separatora (znak odstępu, zmiana wiersza lub dowolny ciąg tych znaków),

- b) włożyć taśmę perforowaną do czytnika,
- c) włączyć monitor,
- d) zerować akumulator i rejestr rozkazów,
- e) przycisnąć dwukrotnie przycisk START.

Po wprowadzeniu wszystkich danych czytnik zatrzymuje się i należy go wyłączyć. Maszyna przechodzi teraz do sprawdzenia czy wartości liczbowe parametrów, charakteryzujących poszczególne elementy operacyjne, są zgodne z wymaganiami języka. O ile program jest poprawny, na monitorze jest drukowana informacja o obszarze pamięci zajęтым przez program. Ma ona postać:

M (liczba ósemkowa)

W (liczba ósemkowa)

np.:

M 00276

W 07753

Pozwala to na orientacyjną ocenę, jak dalece można program dalej rozbudować. W przypadku gdy:

a) na taśmie z danymi znajduje się niewłaściwy znak (np. litera),

b) liczba na taśmie z danymi jest błędnie napisana,

c) wartości parametrów są niezgodne z wymaganiami języka,

d) program zajmuje więcej niż 10 000 (ósemkowo), komórek pamięci maszyna sygnalizuje błąd (patrz 4.3).

Przed przystąpieniem do obliczeń perforator wprowadza około 20 centymetrowy odcinek czystej taśmy, następnie tekst komentarza. Po wykonaniu obliczeń maszyna zatrzymuje się, o ile zajdzie jeden z trzech warunków określonych w punkcie 3.8.35:

a) czas maszynowy przekroczy wartość t_{kon} - w rejestrze rozkazów zostaje wyświetlony w części AR adres 17 701,

b) zostanie przekroczona wartość y_{max} - wyświetlony adres 17 702,

c) zostanie osiągnięty poziom niższy niż y_{min} - wyświetlony adres 17 703.

4.3. Sygnalizacja i poprawianie błędów formalnych programu

Błędy popełnione przez programistę w trakcie pisania programu są wykrywane przez translator i sygnalizowane na monitorze.

W języku AM-F-1 wyróżnia się dwie zasadnicze grupy błędów:

1 grupa - błędy, które można poprawić na monitorze bez konieczności przepisywania programu.

Po wykryciu przez translator błędnej instrukcji tego typu na monitorze jest drukowany następujący napis:

? (błędna instrukcja)
PATRZ INSTRUKCJA NR (nr błędu)



W celu ułatwienia i przyspieszenia poprawiania programów większość możliwych do popełnienia błędów została oznaczona odpowiednimi numerami. Spis błędów i odpowiadających im numerów zamieszczono na końcu rozdziału.

Błędne instrukcje programu należy poprawiać kolejno jedna po drugiej, czekając każdorazowo aż maszyna obok poprawnej instrukcji wydrukuje znak ⊕, co jest sygnałem, że można pisać następne zdanie programu.

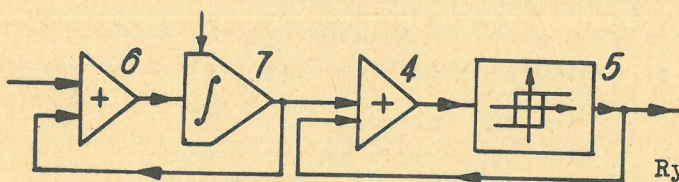
Gdy wszystkie błędne instrukcje zostaną poprawione i przetłumaczone, na monitorze należy napisać instrukcję CZYTAJ i maszyna przechodzi do dalszego wprowadzania programu z czytnika.

2 grupa - błędy, których nie można poprawić na monitorze. Po wykryciu błędu tego typu na monitorze drukowany jest napis:

PATRZ INSTRUKCJA NR (nr błędu)
POPRAW PROGRAM, PONOWNY START OD 10 000

Powyższe uwagi dotyczą błędów wykrywanych w trakcie tłumaczenia programu. Translator sygnalizuje błędy także na innych etapach pracy.

Pętla arytmetyczna. Ze względu na przyjęty sposób wykonywania procesu obliczeń (wynikający z możliwości obliczeniowych maszyny cyfrowej Odra 1013) nie dopuszcza się, aby na schemacie blokowym programu występował zamknięty ciąg bloków realizujących operacje algebraiczne. Sytuację taką nazywamy pętlą arytmetyczną.



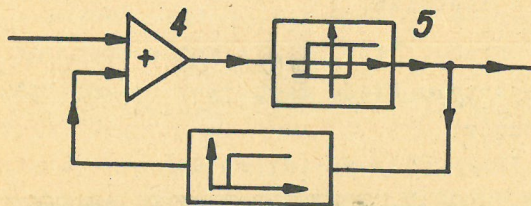
Rys. 40

Elementy 4 i 5 tworzą pętlę arytmetyczną, bowiem do wyznaczenia wielkości wyjściowej któregośkolwiek z nich trzeba mieć wielkość wyjściową elementu drugiego.

Elementy 6 i 7 nie tworzą pętli arytmetycznej, bowiem wartość wielkości wyjściowej elementu całkującego 7 jest zawsze określona przez warunki początkowe.

Likwidacja pętli arytmetycznej przez programistę polega na:

- a) zbudowaniu innego schematu blokowego,
- b) przerwanie pętli arytmetycznej przez włączenie w nią elementu o jednostkowym opóźnieniu równym Δt .



Rys. 41

Stosując odpowiednio mały krok całkowania Δt , błąd wynikający z takiego postępowania można uczynić dowolnie małym.

Sygnalizacja błędów na taśmie z danymi. W przypadku błędnego napisania taśmy z danymi drukowany jest napis

ZMIENNA (OZNACZENIE LITEROWE ZMIENNEJ)

Po poprawieniu błędu należy taśmę jeszcze raz wprowadzić pod czytnik, ponowić czynności wymienione.

Sygnalizacja błędnych parametrów. Jeżeli okaże się, że wartość jakiegokolwiek parametru jest niezgodna z wymaganiami języka, wówczas maszyna drukuje na dalekopisie napis:

NIEDOPUSZCZALNY PARAMETR W BLOKU (Nr BLOKU)

W tym przypadku należy po poprawieniu wartości parametru ponownie wprowadzić program lub tylko taśmę z danymi, o ile błędny parametr został zadeklarowany w zdaniu ZMIENNE.

Sygnalizacja błędów w trakcie wykonywania obliczeń. W trakcie wykonywania obliczeń może wystąpić przypadek, gdy wartości sygnałów wejściowych nie będą mieściły się w dopuszczalnym zakresie (np. sygnał wejściowy przekształtnika funkcji pierwiastek kwadratowy lub logarytm będzie miał wartości ujemne). W takich sytuacjach maszyna zatrzymuje się, wyświetlając w rzędzie AR rejestrów rozkazów numer błędu (ósemkowo).

W tabeli 2 zamieszczono spis błędów wykrywanych przez translator języka symulacyjnego AM-F-1.

Płędy wykrywane przez translator

Nr Błędu	Grupa Błędu	RODZAJ BŁĘDU
10001	2	brak bloku MAX lub blok MAX znajduje się w niewłaściwym miejscu programu
10002	2	komentarz umieszczony w niewłaściwym miejscu programu lub drugi komentarz
10003	2	dwa bloki GZN
10004	2	deklaracja zdania MAX większa od 62
10005	2	brak bloku DRU
10006	2	brak bloku GZN
10007	2	brak bloku KON
10010	2	dwa bloki KON
00011	2	niewłaściwa budowa zdania FUN
10012	2	za mało zmiennych literowych wprowadzonych w programie w stosunku do zadeklarowanych
10013	2	za mało bloków wprowadzonych w programie w stosunku do zadeklarowanych
00014	1	niedopuszczalny znak dalekopisowy na taśmie
00015	1	adres zerowy lub brak adresu
00016	1	nazwa bloku nie poprzedzona numerem
00017	1	adres większy od zadeklarowanego w zdaniu MAX
00020	1	dwa bloki oznaczone tym samym numerem
00021	1	brak separatora
00022	1	błędna nazwa bloku lub brak nazwy
00023	1	brak znaku "=" przed parametrami
00024	1	dodatkowy znak "=" przed parametrami
00025	1	brak parametru

T a b e l a 2 (cd.)

Nr Błędu	Grupa Błędu	RODZAJ BŁĘDU
00026	1	użycie w zdaniu zmiennej nie rezerwowanej lub użycie po raz drugi tej samej zmiennej w programie
00027	1	dwukrotna rezerwacja tej samej litery jako zmiennej
00030	1	zbyt wiele elementów zadania lub brak przecinka
00031	1	przekroczenie dopuszczalnej długości tablicy w zdaniu FUN
00032	1	nieparzysta liczba parametrów w zdaniu FUN
00033	1	niedopuszczalna długość instrukcji
00034	1	niewłaściwy znak dalekopisowy między mantysą a cechą liczby
00035	1	brak cechy po apostrofie l. zmp.
00036	1	parametr wykracza poza zakres liczb maszyny
00037	1	błędna deklaracja parametru z zdaniu DRU
10040	2	ujemny lub równy zeru parametr w zdaniu LICZ
10041	2	parametr w zdaniu LICZ lub zadeklarowany czas opóźnienia wyrażone w krokach całkowania są większe od 8191
10042	2	błędny parametr w bloku o podanym numerze
00043	1	błędny znak dalekopisowy w parametrach zdania FUN lub brak przecinka po zdaniu FUN
16000	1	Błąd w zmiennych
17777	2	przetłumaczony program nie mieści się w pamięci maszyny
010--	2	pierwiastek kwadratowy z liczby ujemnej

T a b e l a 2 (cd.)

Nr Błędu	Grupa Błędu	RODZAJ BŁĘDU
020--	2	logarytm z liczby niedodatniej
030--	2	argument funkcji EXP większy od 176
040--	2	wartości funkcji TAN lub CTA $\rightarrow \infty$
050--	2	dzielenie przez zero

5. PRZYKŁAD PROGRAMOWANIA W JĘZYKU AM-F-1

W celu zobrazowania metody rozwiązywania zagadnień na maszynie cyfrowej za pomocą języka symulacyjnego AM-F-1 przytoczono poniżej przykład analizy dynamiki układu opisanego równaniem różniczkowym nieliniowym.

Należy rozwiązać równanie postaci:

$$\frac{d^2x}{dt^2} + k \left[\text{sign} \left(\frac{dx}{dt} \right) \right] \left(\frac{dx}{dt} \right)^2 + x = 0, \quad (1)$$

przy: $k = \text{var}$,

$$\frac{dx}{dt} \Big|_{t=0} = 10.0 \text{ m/s},$$

$$x=0.0 \text{ m},$$

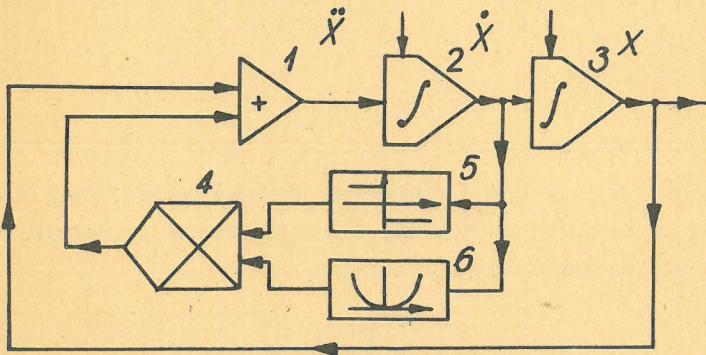
w przedziale czasu $t \in < 0, 10 >$ (sekund).

Tok postępowania:

a) z równania (1) wyznaczamy pochodną najwyższego rzędu:

$$\frac{d^2x}{dt^2} = -k \left[\text{sign} \left(\frac{dx}{dt} \right) \right] \left(\frac{dx}{dt} \right)^2 - x, \quad (2)$$

b) na podstawie równania (2) budujemy schemat analogowy,



Rys. 42

- c) numerujemy kolejno wszystkie elementy,
 d) piszemy program w języku AM-F-1, opisując kolejno poszczególne elementy schematu, dodając na końcu także bloki: DRU, GZN i KON.

T a b e l a 3

Tabulogram programu rozwiązywanego zagadnienia

✠
 ANALIZA DYNAMIKI UKŁADU DRGAJĄCEGO W OŚRODKU
 O TŁUMIENIU PROPORCJONALNYM DO KWADRATU PRĘDKOŚCI

T	V	X
(SEK)	(M/SEK.)	(M)

✠

ZMIENNE KTD ,

MAX 11 ,

01 SUM 03 04 = -1 K ,

02 INT 01 = 10. 1. ,

03 INT 2 = 0. 1. ,

04 MNZ 05 06 ,

05 KOM 02 = -1. 0. 1. -1. ,

06 KWR 02 ,

07 GZN = 0. t ,

08 KON 07 = -1. 10. 10. ,

09 DRU 07 = 1.2 ,

10 DRU 02 = 2.2 ,

11 DRU 03 = 3.1 ,

LICZ = D ,

Napisany w ten sposób program należy wydrukować na dalekopisie łącznie z danymi, po czym można przystąpić do obliczeń na maszynie cyfrowej.

