



**POLITECHNIKA POZNAŃSKA**

**INSTYTUT AUTOMATYKI**

60-965 Poznań, ul. Piotrowo 3

tel. 782-365, 782-366

<b>Zleceniodawca</b> Zarząd Dróg i Mostów Poznań, ul. Wilczak 16	<b>Nr umowy</b> IA-U/11/74/75 B
<b>Nazwa opracowania</b> System operacyjny SOWIK	<b>Rodzaj opracowania</b> Dokumentacja programu
<b>Urządzenie</b> SCR-5	<b>Egz.</b> 6
	<b>Nr archiwalny</b> 76024





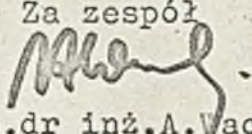
System operacyjny SOWIK

Zleceńodawca: Zarząd Dróg i Mostów  
Poznań, ul. Wilczak 16

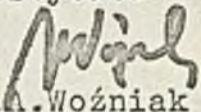
Znak zlecenia: TRE/7132/75

Nasz znak: IA-U/11/74/75 B

Za zespół

  
doc.dr inż. A. Vachowski

Dyrektor Instytutu

  
Prof. dr hab. inż. A. Woźniak

Poznań, listopad 1976





Autorzy:

dr inż. Jacek Martinek

mgr Jerzy Bartoszek

mgr inż. Jan Nadolski





## Spis treści

	str.
1. Wstęp	3
2. Uruchamianie SOWIK-a	5
3. Opis zleceń	9
4. Opis realizacji programu konwersacyjnego	20
5. Cytowane opracowania	25
6. Wykaz zleceń	26
7. Wykaz błędów	27
8. Tekst programu konwersacyjnego.	28





## 1. Wstęp

System operacyjny SOWIK realizuje funkcje, które wynikają z zasad działania WIK-a oraz zapewnia konwersację z użytkownikiem lub wielu użytkownikami.

W skład systemu operacyjnego SOWIK wchodzi:

- komputer wirtualny WIK,
- trzy procesory arytmetyczno-logiczne o nazwach 1,4,5,
- procesor zewnętrzny o nazwie 2,
- procesor wejścia o nazwie 3,
- program wczytujący,
- program początkowy,
- program konwersacyjny.

Zasady działania WIK-a opisano w [1].

W punkcie 2 przedstawiono sposób wykorzystania procesorów o nazwach 1,2,3 oraz programów wczytującego i początkowego. Zadaniem programu konwersacyjnego jest realizowanie zleceń wysyłanych przez użytkownika (użytkowników) oraz sygnalizowanie błędów wykrytych w trakcie działania procesorów arytmetyczno-logicznych. Program konwersacyjny napisano w języku AWIK (patrz [2]).

Sposoby wykorzystania zleceń oraz ich formaty opisano w punkcie 3.

Zlecenia umożliwiają:

- wczytywanie do pamięci wirtualnej programów zapisanych w kodzie heksadecymalnym,
- wczytywanie i wyprowadzanie taśm binarnych,
- tworzenie, usuwanie, startowanie i zatrzymywanie procesorów,





- zmienianie priorytetów procesorów,
- przesyłanie komunikatów do procesorów,
- wyprowadzanie oraz zmienianie zawartości pamięci wirtualnej,
- wyprowadzanie oraz zmienianie zawartości rejestrów procesorów.

Do wysyłania zleceń wykorzystuje się procesory zewnętrzne. Procesory o nazwach 4 i 5 pełnią w SOWIK-u rolę pomocniczą. Zostają one utworzone w trakcie wykonywania programu konwersacyjnego.





## 2. Uruchamianie SOWIK-a

Uruchamianie SOWIK-a jest procesem trzyetapowym.

### Etap 1 - wprowadzenie programu wczytującego

1. Włączyć maszynę cyfrową.
2. Włączyć urządzenie wejścia/wyjścia związane z linią przerwań nr 2 w bloku przerwań we/wy, czyli urządzenie, którego NRUZ jest równy 2 (patrz: [1],[3] )
3. Włączyć urządzenie wejścia związane z klawiszem WPR, znajdującym się na pulpicie technicznym maszyny cyfrowej.  
  
Przyjęto, że urządzenie to związane jest z linią przerwań nr 1 w bloku przerwań we/wy, czyli jego NRUZ jest równy 1.
4. Założyć taśmę oznaczoną napisem PROGRAM WCZYTUJACY do urządzenia, którego NRUZ jest równy 1.
5. Nacisnąć klawisz ZER znajdujący się na pulpicie technicznym.
6. Wyzerować rejestry LR i A maszyny cyfrowej.
7. Obserwując zawartość rejestru A naciskać klawisz WPR aż do pojawienia się pierwszego znaku o kodzie różnym od zera.
8. Wcisnąć klawisz PA0.
9. Nacisnąć klawisz ŁAD.
10. Nacisnąć klawisz LR+1.
11. Nacisnąć klawisz WPR

Czynności wymienione w punktach 9, 10, 11 należy powtarzać tak długo, aż zawartość rejestru LR (wyświetlona przez dolny rząd lamp na pulpicie technicznym) będzie większa od 22g.





12. Wcisnąć klawisz LR.
13. Nacisnąć klawisz ZER.
14. Umieścić w rejestrze LR wartość równą jeden.
15. Nacisnąć klawisz ZER.
16. Wcisnąć klawisz S-S.

W wyniku wykonania wyżej wymienionych czynności zostanie wczytana pozostała część taśmy oznaczonej napisem PROGRAM W CZYTUJACY, a urządzenie, którego NRUZ jest równy 2 wydrukuje napis:

IN =

#### Etap 2 - wczytanie WIK-a oraz programu początkowego

WIK oraz program początkowy znajdują się na taśmie oznaczonej napisem WIK BIN.

Taśmę tę można wczytać do pamięci operacyjnej przy pomocy programu wczytującego. W tym celu za napisem IN= należy napisać dwucyfrową liczbę ósemkową (NRUZ) określającą urządzenie wejścia, przy pomocy, którego, zostanie wczytana taśma WIK BIN. Urządzenie to powinno zostać uprzednio przygotowane do wprowadzenia informacji. Po wczytaniu taśmy WIK BIN następuje automatyczne uruchomienie komputera wirtualnego. Sygnalizowane jest to napisem

\*WIK

drukowanym przez urządzenie, którego NRUZ jest równy 2. Stan ten nazywa się stanem początkowym WIK-a. Istnieją wówczas trzy wirtualne procesory:

- procesor arytmetyczno-logiczny o nazwie 1,





- procesor zewnętrzny o nazwie 2, związany z urządzeniem, którego NRUZ jest równy 2, .
- procesor wejścia o nazwie 3, związany z urządzeniem, którego NRUZ został określony przez użytkownika po wydrukowaniu napisu. IN= .

Z wirtualnym procesorem arytmetyczno-logicznym związany jest program napisany w rozkazach WIK-a, nazywany programem początkowym.

### Etap 3 - uczytanie programu konwersacyjnego

Jeżeli użytkownik napisze na urządzeniu zewnętrznym związanym z procesorem o nazwie 2 sekwencję znaków postaci

\*RS or lf

gdzie cr oznacza naciśnięcie klawisza "powrót karetki",

lf oznacza naciśnięcie klawisza "wysuw o wiersz",

wtedy procesor o nazwie 1, wykorzystując procesor o nazwie 3, wczytuje przy pomocy programu początkowego dowolny inny program zapisany w kodzie heksadecymalnym. Wystąpienie błędu w trakcie wykonywania powyższych czynności jest sygnalizowane napisem

\*RS <nb>

gdzie <nb> określa rodzaj błędu w następujący sposób:

1 - nieoperatywność urządzenia,

2 - błąd w trakcie realizacji transmisji,

inna - błędna sekwencja znaków napisana przez użytkownika, drukowanym przez urządzenie związane z procesorem o nazwie 2. Procesor o nazwie 1 czeka wówczas na kolejną informację od użytkownika.





Poprawne wczytanie programu sygnalizowane jest napisem

\*RS,

a wczytany program jest wykonywany przez procesor o nazwie 1 poczynając od zerowego słowa pierwszego wczytanego segmentu. Nazwa tego segmentu nie może być mniejsza od  $25_8$  i jest umieszczana w S-rejestrze nr  $\emptyset$ .

Powtórne "wywołanie" programu początkowego jest możliwe poprzez wykonanie następującej sekwencji rozkazów WIK-a:

LS $\emptyset$ , 2 $\emptyset$ ;

JP,  $\emptyset$  ;

W szczególnym przypadku programem wczytanym w sposób opisany powyżej może być program konwersacyjny. Zajmuje on wówczas segmenty o nazwach  $25_8$  do  $31_8$ . Rozpoczęcie wykonywania programu konwersacyjnego sygnalizowane jest napisem postaci  
\*SOWIK

drukowanym przez urządzenie zewnętrzne związane z procesorem o nazwie 2.

Jak wspomniano w punkcie 1, program konwersacyjny napisano w języku AWIK. Jego postać w kodzie heksadecymalnym otrzymano wykorzystując translator języka AWIK zrealizowany na maszynie cyfrowe K202 (patrz [2]). Taśmę zawierającą program konwersacyjny w kodzie heksadecymalnym oznaczono napisem

PROGRAM KONWERSACYJNY.

#### Uwaga !

W czasie działania WIK-a zabrania się naciskania klawisza ZER, znajdującego się na pulpicie technicznym (również wtedy, gdy klawisz S-S jest wyciśnięty).





### 3. Opisy zleceń

Wszystkie zlecenia przesyłane są do procesora arytmetyczno-logicznego o nazwie 1. Ponieważ do ich przesyłania wykorzystuje się procesory zewnętrzne, każde zlecenie musi być poprzedzone znakiem " \* " i zakończone znakiem "nowa linia" (znak o kodzie równym 12<sub>8</sub>). Ogólny format zleceń jest następujący:

\* <kod> , <par 0> , <par 1> , ... , <par N>

gdzie:

<kod> oznacza dwuznakową nazwę zlecenia np. CA, RS, W1 , .

<par i>  $0 \leq i \leq N$  jest liczbą ósemkową lub ciągiem znaków w kodzie ISO-7 (z wyjątkiem znaku "nowa linia").

Poprawne wykonanie zlecenia (z wyjątkiem zlecenia RS) sygnalizowane jest napisem postaci

SOWIK , <kod> , <par 0>

drukowanym bezpośrednio po tym wykonaniu.

Błędne zlecenie oraz takie, w trakcie transmisji którego wystąpił błąd w działaniu urządzenia zewnętrznego, sygnalizowane jest napisem postaci

\* SOWIK , <kod> , <par 0> , ER , <nb>

gdzie:

<kod> i <par 0> oznaczają to samo co poprzednio,

<nb> oznacza numer błędu.

Wszystkie informacje o zleceniu (z wyjątkiem zlecenia RS) drukowane są na urządzeniu zewnętrznym użytkownika, od którego zlecenie otrzymano:





Sposób drukowania informacji dotyczących wykonania zlecenia RS opisano w punkcie 2. Błędy wykryte w trakcie działania innych procesorów arytmetyczno-logicznych sygnalizowane są w sposób analogiczny, jak błędne zlecenia. W takich przypadkach  $\langle \text{kod} \rangle$  oznacza napis PR, a  $\langle \text{par } \emptyset \rangle$  jest nazwą procesora, który błąd wykrył. Napis powyższy poprzedzony jest wówczas wydrukiem zawartości rejestrów SW i LR tego procesora. Zawartości rejestrów SW i LR drukowane są w sposób przedstawiony w opisie zlecenia W2.

Informacje o błędach wykrytych w czasie działania procesorów arytmetyczno-logicznych drukowane są na urządzeniu zewnętrznym związanym z procesorem o nazwie 2.

Formaty poszczególnych zleceń podano w dalszych częściach tego opisu.

W opisach formatów  $\langle \text{par } i \rangle \quad \emptyset \leq i \leq N$  zastąpiono następującymi symbolami

- $\langle S_n \rangle$  oznacza jednocyfrowy numer rejestru bazowego (S-rejestru) lub literę „E”,
- $\langle P \rangle$  oznacza jednocyfrową liczbę określającą priorytet procesora lub literę „E”,
- $\langle NP \rangle$  oznacza dwucyfrową nazwę procesora,
- $\langle NS_n \rangle$  oznacza dwucyfrową nazwę segmentu lub literę „E”,
- $\langle NU \rangle$  oznacza dwucyfrowy numer urządzenia zewnętrznego (NRUZ),
- $\langle \text{ADR}_n \rangle$  oznacza trzycyfrowy adres w ramach segmentu lub literę „E”,
- $\langle \text{PAR}_n \rangle$  oznacza trzycyfrową liczbę lub literę „E”,
- $\langle \text{TREŚĆ KOMUNIKATU} \rangle$  oznacza ciąg znaków w kodzie ISO-7 (z wyjątkiem znaku "nowa linia").





Literę „E” można używać jedynie w zleceniach zmieniających zawartość pamięci rejestrów lub zmieniających priorytet procesora. Użycie litery „E” oznacza wówczas, że zawartość odpowiedniego słowa pamięci, rejestru (jego części) lub priorytet procesora nie ulegają zmianie.

Uwaga !

W żadnym zleceniu nie można dodawać nowych znaków, ani opuszczać znaków wymienionych. Dotyczy to również znaku "odstęp" (znak o kodzie równym  $40_8$ ).

Wczytaj system operacyjny

Format zlecenia:

\*RS

Użycie tego zlecenia powoduje wczytanie przy pomocy programu początkowego oraz procesora wejścia o nazwie 3 (w sposób analogiczny jak to opisano w punkcie 2) "nowego" programu konwersacyjnego. Uzyskuje się w ten sposób "nową" wersję SOWIK-a.

Wczytaj program (RP)

Format zlecenia:

\*RP , <NP>

Zlecenie to pozwala wczytać do pamięci wirtualnej dowolny program zapisany w kodzie heksadecymalnym. W trakcie realizacji tego zlecenia wykorzystuje się procesor wejścia o nazwie <NP>





### Wczytaj taśmę binarną

Format zlecenia:

\*RB, <NP>

Przy pomocy tego zlecenia wczytuje się do pamięci wirtualnej taśmę binarną (otrzymaną przy pomocy zlecenia WB) wykorzystując procesor wejścia o nazwie <NP>

### Wyprowadź taśmę binarną

Format zlecenia:

\*WB, <NP>, <NS>, <ADR<sub>1</sub>>, <ADR<sub>2</sub>>

Przy pomocy tego zlecenia wyprowadza się na taśmę (wykorzystując procesor wyjścia o nazwie <NP>) zawartość segmentu o nazwie <NS> poczynając od adresu <ADR<sub>1</sub>> a kończąc na adresie <ADR<sub>2</sub>>.

Tak otrzymaną taśmę będziemy nazywać taśmą binarną.

### Utwórz procesor arytmetyczno-logiczny

Format zlecenia:

\*CA, <NP>, <S<sub>n</sub>>, <ADR>, <NS>, <P>

Zlecenie to pozwala utworzyć procesor arytmetyczno-logiczny o nazwie <NP> i priorytecie <P>.

Początkowe zawartości rejestrów A, A' i X tego procesora są równe zero. W dwu najbardziej lewych bitach rejestru LR zostanie umieszczony numer rejestru bazowego <S<sub>n</sub>> a w pozostałych bitach adres <ADR>. W rejestrze bazowym <S<sub>n</sub>> zostanie umieszczona nazwa segmentu <NS>.





Pozostałe rejestry bazowe są nieokreślone. Wskaźniki P, C, L, E, G, V, B rejestru SW są równe zero. Utworzony procesor znajduje się w stanie zatrzymania.

Wykonanie tego zlecenia jest równoważne wykonaniu rozkazu CAP WIK-a (patrz [1]).

#### Utwórz procesor wejścia

Format zlecenia:

\* CI, <NP>, <NU>

Przy pomocy tego zlecenia tworzy się procesor wejścia o nazwie <NP> związany z urządzeniem zewnętrznym o numerze <NU>. Po utworzeniu procesor wejścia znajduje się w stanie zatrzymania.

Wykonanie tego zlecenia jest równoważne wykonaniu rozkazu CIP WIK-a (patrz [1]).

#### Utwórz procesor wyjścia

Format zlecenia:

\* <CO>, <NP>, <NU>

Zlecenie to pozwala utworzyć procesor wyjścia o nazwie <NP> związany z urządzeniem zewnętrznym o numerze <NU>.

Po utworzeniu procesor wyjścia znajduje się w stanie zatrzymania.

Wykonanie tego zlecenia jest równoważne wykonaniu rozkazu COP WIK-a (patrz [1]).





### Utwórz procesor zewnętrzny

Format zlecenia:

\*CE, <NP>, <NU>

Przy pomocy tego zlecenia tworzy się procesor zewnętrzny o nazwie <NP> związany z urządzeniem zewnętrznym o numerze <NU>. Po utworzeniu procesor zewnętrzny znajduje się w stanie zatrzymania.

Wykonanie tego zlecenia jest równoważne wykonaniu rozkazu CEP WIK-a (patrz [1]).

### Usuń procesor

Format zlecenia:

\*DP, <NP>

Przy pomocy tego zlecenia usuwa się procesor o nazwie <NP> będący w stanie zatrzymania lub zatrzymania i zablokowania. Wykonanie tego zlecenia jest równoważne wykonaniu rozkazu DEP WIK-a (patrz [1]).

### Startuj procesor

Format zlecenia:

\*ST, <NP>

Przy pomocy tego zlecenia wyprowadza się procesor o nazwie <NP> ze stanu zatrzymania. Wykonanie tego zlecenia jest równoważne wykonaniu rozkazu ST1 WIK-a (patrz [1]).

### Zatrzymaj procesor

Format zlecenia:

\*SP, <NP>





Po wykonaniu tego zlecenia procesor o nazwie  $\langle NP \rangle$  będzie w stanie zatrzymania, jeżeli był w stanie przetwarzania, oraz będzie w stanie zablokowania i zatrzymania, jeśli był w stanie zablokowania.

Wykonanie tego zlecenia jest równoważne wykonaniu rozkazu STP WIK-a (patrz [1]).

### Wyślij komunikat

Format zlecenia:

\*SM,  $\langle NP \rangle$ ,  $\langle \text{TREŚĆ KOMUNIKATU} \rangle$

Przy pomocy tego zlecenia można wysłać komunikat o treści określonej przez  $\langle \text{TREŚĆ KOMUNIKATU} \rangle$  (maksymalnie 22 znaki) do procesora arytmetyczno-logicznego o nazwie  $\langle NP \rangle$ .

Po wykonaniu zlecenia procesor o nazwie 1 czeka na odpowiedź pochodzącą od procesora o nazwie  $\langle NP \rangle$ .

Wykonanie tego zlecenia jest równoważne wykonaniu rozkazu SHE WIK-a (patrz [1]).

### Umieść w pamięci

Format zlecenia:

\*RM,  $\langle NS \rangle$ ,  $\langle ADR \rangle$ ,  $\langle PAR_1 \rangle$ ,  $\langle PAR_2 \rangle$

Zlecenie to pozwala umieścić wartości określone przez  $\langle PAR_1 \rangle$  i  $\langle PAR_2 \rangle$  w dwu kolejnych słowach segmentu o nazwie  $\langle NS \rangle$ .

Adres pierwszego słowa określony jest przez  $\langle ADR \rangle$  (wartość  $\langle ADR \rangle \leq 254$ ).





### Wyprowadź zawartość pamięci

Format zlecenia:

\*WM, <NS>, <ADR>

Przy pomocy tego zlecenia można wyprowadzić zawartości dwu kolejnych słów segmentu o nazwie <NS> poczynając od adresu <ADR> (wartość <ADR> ≤ 254). Zawartość każdego słowa wyprowadzona jest w postaci trzycyfrowej liczby ósemkowej, przy czym jako pierwsza wyprowadzana jest zawartość słowa o adresie <ADR>. Następnie drukowany jest napis sygnalizujący poprawne wykonanie zlecenia.

### Umieść w rejestrach A i X

Format zlecenia:

\*R1, <NP>, <PAR<sub>1</sub>>, <PAR<sub>2</sub>>, <PAR<sub>3</sub>>, <PAR<sub>4</sub>>

Zlecenie to pozwala zmienić zawartość rejestrów A i X procesora o nazwie <NP>.

<PAR<sub>1</sub>> i <PAR<sub>2</sub>> określają nową zawartość rejestru A;

<PAR<sub>3</sub>> i <PAR<sub>4</sub>> dotyczą rejestru X.

### Umieść w rejestrach SW i LR

Format zlecenia:

\*R2, <NP>, <PAR>, <Sn>, <ADR>

Przy pomocy tego zlecenia można zmienić zawartość rejestrów SW i LR procesora o nazwie <NP>.

<PAR> określa nową zawartość rejestru SW.





$\langle S_n \rangle$  określa nową zawartość dwu najbardziej lewych bitów rejestru LR;

$\langle ADR \rangle$  dotyczy pozostałych bitów tego rejestru.

Umieść w rejestrach  $S_0$  i  $S_1$

Format zlecenia:

\*R3,  $\langle NP \rangle$ ,  $\langle NS_1 \rangle$ ,  $\langle NS_2 \rangle$

Przy pomocy tego zlecenia w rejestrach  $S_0$  i  $S_1$  procesora arytmetyczno-logicznego o nazwie  $\langle NP \rangle$  można umieścić odpowiednio nazwy segmentów  $\langle NS_1 \rangle$  i  $\langle NS_2 \rangle$ .

Wykonanie tego zlecenia jest równoważne wykonaniu dwóch rozkazów WIK-a:

LS0 i LS1 (patrz [1]).

Umieść w rejestrach  $S_2$  i  $S_3$

Format zlecenia:

\*R4,  $\langle NP \rangle$ ,  $\langle NS_1 \rangle$ ,  $\langle NS_2 \rangle$

Zlecenie to pozwala umieścić w rejestrach  $S_2$  i  $S_3$  procesora arytmetyczno-logicznego o nazwie  $\langle NP \rangle$  nazwy segmentów określone odpowiednio przez  $\langle NS_1 \rangle$  i  $\langle NS_2 \rangle$ .

Wykonanie tego zlecenia jest równoważne wykonaniu dwóch rozkazów WIK-a: LS2 i LS3 (patrz [1]).

Wyprowadź zawartość rejestrów A i X

Format zlecenia:

\*W1,  $\langle NP \rangle$





Przy pomocy tego zlecenia można wyprowadzić zawartość rejestrów A i X procesora o nazwie  $\langle NP \rangle$ . Zawartość każdego rejestru wyprowadzana jest w postaci dwu liczb ósemkowych trzycyfrowych. Jako pierwsza wyprowadzana jest zawartość rejestru A. Po wyprowadzeniu zawartości obu rejestrów drukowany jest napis sygnalizujący poprawne wykonanie zlecenia.

Wyprowadź zawartości rejestrów SW i LR

Format zlecenia:

\*W2,  $\langle NP \rangle$

Zlecenie to pozwala wyprowadzić zawartości rejestrów SW i LR procesora o nazwie  $\langle NP \rangle$ .

Jako pierwsza wyprowadzana jest zawartość rejestru SW w postaci liczby ósemkowej trzycyfrowej. Zawartość rejestru LR wyprowadzana jest przy pomocy dwu liczb ósemkowych. Pierwsza – jednocyfrowa określa zawartość dwu najbardziej lewych bitów rejestru LR; druga – trzycyfrowa dotyczy pozostałych bitów tego rejestru. Następnie wyprowadzamy jest napis sygnalizujący poprawne wykonanie zlecenia.

Wyprowadź zawartości rejestrów S0 i S1

Format zlecenia:

\*W3,  $\langle NP \rangle$

Zlecenie to pozwala wyprowadzić zawartości rejestrów S0 i S1 procesora arytmetyczno-logicznego o nazwie  $\langle NP \rangle$ .

Jako pierwsza wyprowadzana jest, w postaci liczby ósemkowej dwucyfrowej, zawartość rejestru S0. Następnie wyprowadzana jest w analogiczny sposób zawartość rejestru S1.





Po wyprowadzeniu zawartości obu rejestrów drukowany jest napis sygnalizujący poprawne wykonanie zlecenia.

Wyprowadź zawartości rejestrów S2 i S3

Format zlecenia:

\*W4, <NP>

Przy pomocy tego zlecenia można wyprowadzić zawartości rejestrów S2 i S3 procesora arytmetyczno-logicznego o nazwie <NP>. Jako pierwsza wyprowadzana jest, w postaci liczby ósemkowej dwucyfrowej, zawartość rejestru S2. Zawartość rejestru S3 wyprowadzana jest w analogiczny sposób. Po wyprowadzeniu zawartości obu rejestrów drukowany jest napis sygnalizujący poprawne wykonanie zlecenia.

Zmień priorytet procesora

Format zlecenia:

\*PC, <NP>, <P>

Zlecenie to pozwala zmienić priorytet procesora arytmetyczno-logicznego o nazwie <NP>.

"Nowy" priorytet procesora jest określony przez <P> i musi być różny od 0 i 1.





#### 4. Opis realizacji programu konwersacyjnego

Jak wspomniano w punkcie 1 zadaniem programu konwersacyjnego jest obsługiwanie zleceń wysyłanych przez użytkownika (użytkowników) oraz sygnalizowanie błędów wykrytych w trakcie działania procesorów arytmetyczno-logicznych. Do realizacji powyższych celów wykorzystuje się trzy procesory arytmetyczno-logiczne o nazwach: 1, 4, 5, oraz priorytetach równych odpowiednio: 7, 6, 2.

Z tego względu program konwersacyjny należy rozpatrywać jako konfigurację trzech programów wykonywanych równolegle.

Zadaniem procesora 1 jest realizowanie zleceń otrzymywanych od użytkownika (użytkowników). W zleceniach tych mogą wystąpić (z oczywistych powodów) błędy składniowe oraz błędy semantyczne.

Obydwa rodzaje błędów są sygnalizowane nadawcy zlecenia, ale wystąpienie błędu semantycznego powoduje przejście procesora 1 w stan zatrzymania (patrz [1]). Zaistniała więc konieczność utworzenia procesora 4, którego zadaniem jest "wyprowadzanie" procesora 1 ze stanu zatrzymania.

Procesor 4 jest w stanie przetwarzania jedynie w czasie realizacji poszczególnych zleceń.

Schematy działań procesorów 1 i 4 przedstawiono na rysunkach 1 i 2.

Procesor 5 służy do sygnalizacji błędów wykrytych w trakcie działania procesorów arytmetyczno-logicznych istniejących w komputerze WIK. Informacje o błędach oraz związane z nimi nazwy procesorów znajdują się w specjalnej tablicy błędów.



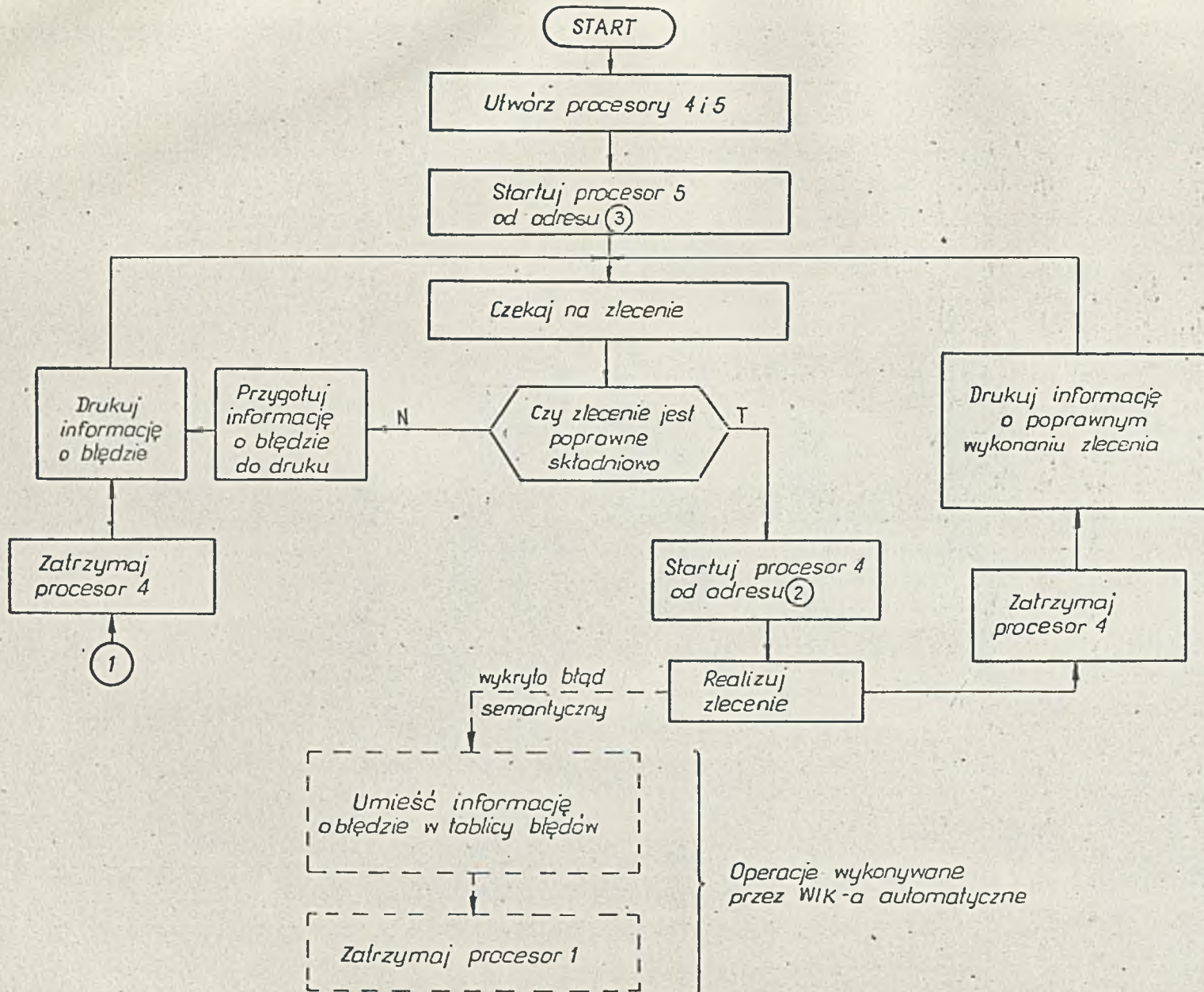


Dostęp do tej tablicy mają: komputer wirtualny WIK, procesor 4 oraz procesor 5. Do sygnalizacji błędów procesor 5 wykorzystuje urządzenie zewnętrzne związane z procesorem 2. Bardziej szczegółowe informacje dotyczące sygnalizacji błędów zamieszczono w punkcie 3.

Ze względu na sposób działania WIK-a (a zwłaszcza procesorów zewnętrznych) procesor 5 nie może być wykorzystany do realizacji zadania wykonywanego przez procesor 4.

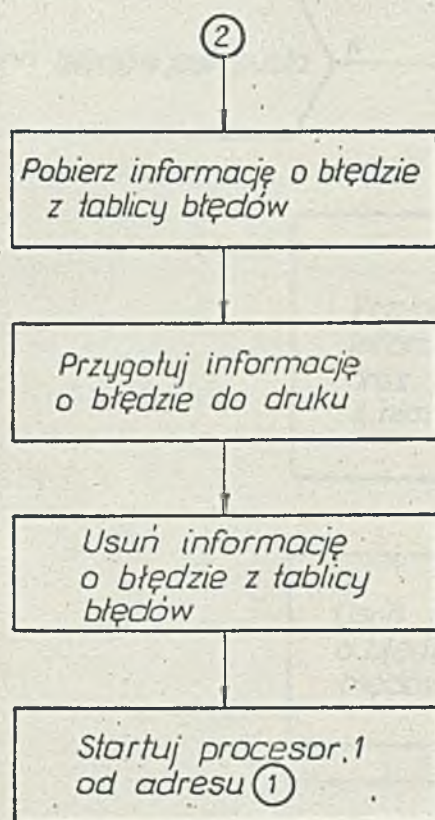
Schemat działania procesora przedstawiono na rysunku 3.





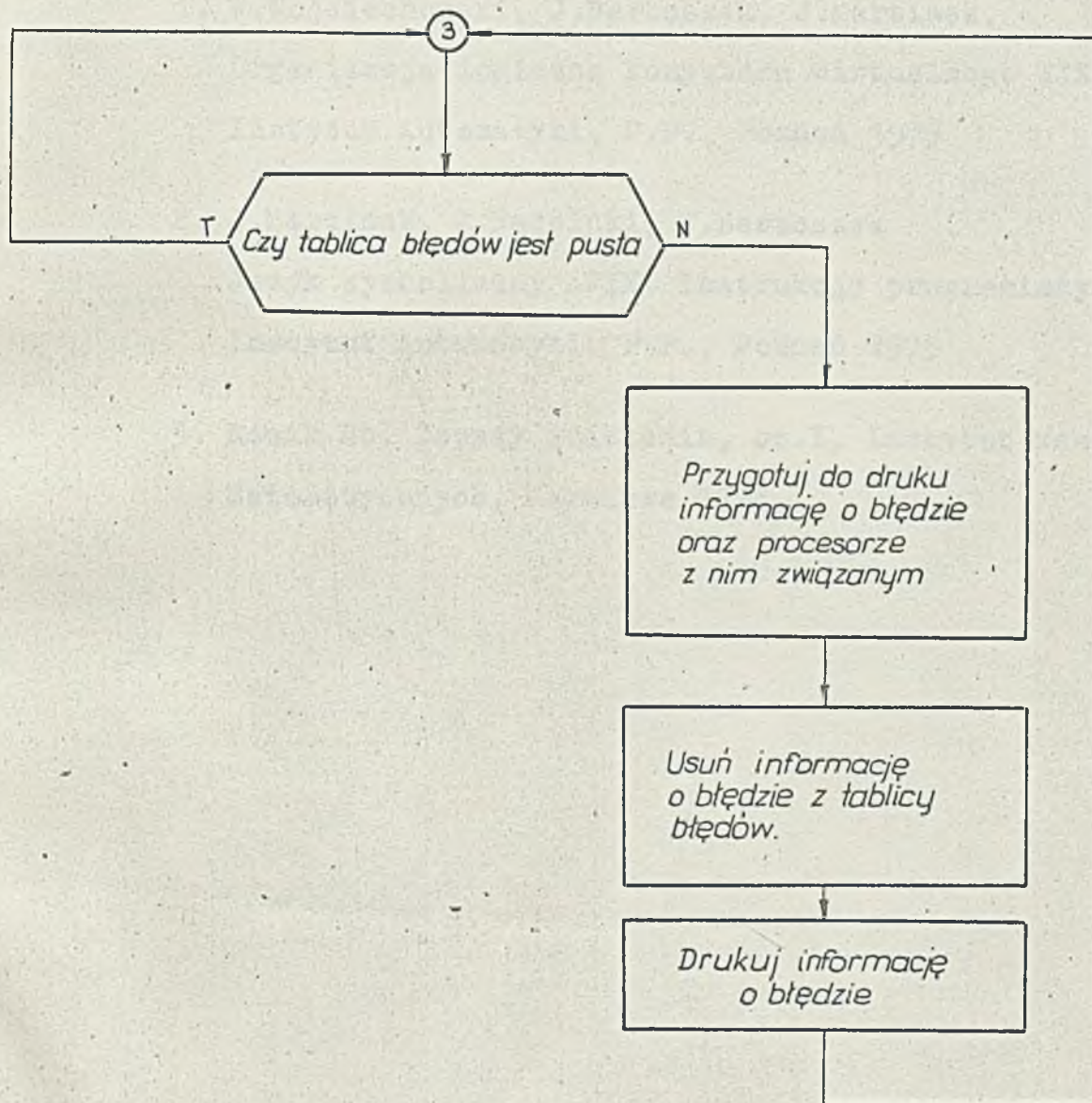
Rys. 1. Schemat działania procesora 1





Rys. 2. Schemat działania procesora 4.





Rys. 3. Schemat działania procesora 5





## 5. Cytowane opracowania

1. W.Wojciechowski, J.Bartoszek, J.Martinek,  
Organizacja logiczna komputera wirtualnego WIK,  
Instytut Automatyki, P.P., Poznań 1975
2. J.Martinek, J.Nadolski, J.Bartoszek  
Język symboliczny AWIK. Instrukcja programisty,  
Instytut Automatyki, P.P., Poznań 1975
3. Momik 8b. Zasady działania, cz.I, Instytut Maszyn  
Matematycznych, Warszawa 1972.





## 6. Wykaz zleceń

Nazwa zlecenia	Znaczenie	Strona
RS	wczytaj system operacyjny	11
RP	wczytaj program	11
RB	wczytaj taśmę binarną	12
WB	wyprowadź taśmę binarną	12
CA	utwórz procesor arytmetyczno-logiczny	12
CI	utwórz procesor wejścia	13
CO	utwórz procesor wyjścia	13
CE	utwórz procesor zewnętrzny	14
DP	usuń procesor	14
ST	startuj procesor	14
SP	zatrzymaj procesor	14
SM	wyślij komunikat	15
RM	umieść w pamięci	15
WM	wyprowadź zawartość pamięci	16
R1	umieść w rejestrach A i X	16
R2	umieść w rejestrach SW i LR	16
R3	umieść w rejestrach S0 i S1	17
R4	umieść w rejestrach S2 i S3	17
W1	wyprowadź zawartości rejestrów A i X	17
W2	wyprowadź zawartości rejestrów SW i LR	18
W3	wyprowadź zawartości rejestrów S0 i S1	18
W4	wyprowadź zawartości rejestrów S2 i S3	19
PC	zmień priorytet procesora	19





## 7. Wykaz błędów

Numer błędu	Znaczenie
00	błędny rozkaz
01	nieoperatywność urządzenia zewnętrznego
02	błąd (przekłamanie) w trakcie realizacji transmisji danych
03	Urządzenie nie realizuje transmisji w określonym kierunku
04	błędna taśma w transmisji typu binarnego
05	błędny numer urządzenia zewnętrznego (NRUZ) w rozkazie tworzenia procesora wejścia, wyjścia lub zewnętrznego
06	błędny adres
07	brak procesora o takiej nazwie
10	próba "podwójnego" tworzenia procesora
11	próba usunięcia procesora, który nie jest w stanie zatrzymania
12	próba wykonania rozkazu ST2 (patrz [1]), gdy procesor nie jest w stanie zatrzymania
13	próba wysłania odpowiedzi gdy nie otrzymano komunikatu
14	odpowieź generowana przez komputer WIK ponieważ brak procesora odbiorcy komunikatu
16	wykonano rozkaz STA (patrz [1])
17	błędne zlecenie



8. TEKST PROGRAMU KONWERSACYJNEGO



ZSEG 025N  
 ZLABN  
 LS3 030;  
 JP START;  
 (CODE)  
 TXT R S R P R B W B C A C I  
 C O C E D P S T S M S P R M  
 W M R 1 R 2 R 3 R 4 W 1 W 2  
 W 3 W 4 P C ; (ERINS)  
 PAR 061 B; PAR 067 B;

## (ADRESS)

PAR RS B B;  
 PAR RP B B;  
 PAR RWB B B;  
 PAR RWB B B;  
 PAR CA B B;  
 PAR CIOE B B;  
 PAR CIOE B B;  
 PAR CIOE B B;  
 PAR DSS B B;  
 PAR DSS B B;  
 PAR SM B B;  
 PAR DSS B B;  
 PAR RM B B;  
 PAR WM B B;  
 PAR R1 B B;  
 PAR R2 B B;  
 PAR R3 B B;  
 PAR R4 B B;  
 PAR W1 B B;  
 PAR W2 B B;  
 PAR W3 B B;  
 PAR W4 B B;  
 PAR PC B B;

## (ZERO)

PAR 0 B;

## (P3)

PAR 3 B B;

## (P7)

PAR 7 B B;

## (P0377)

PAR 0377 B B;

## (ISD)

PAR 060 B; PAR 060 B;

## (ZAZA)

PAR 036 B; PAR 036 B;

## (E)

PAR 054 B;

PAR 0105 B ;

## (ANS)

ZRES 31N

## (MES)

PAR 0123 B;

PAR 0117 B;

PAR 0127 B;

PAR 0111 B;

PAR 0113 B;

PAR 015 B;



PAR 012 B;  
PAR 0 B;  
PAR 054 B;  
PAR 0 B B;  
PAR 054 B;  
PAR 0 B;  
PAR 0122 B;  
PAR 054 B;  
PAR 0 B B;  
PAR 015 B;  
PAR 012 B;

(MES1)  
PAR 014 B B;  
PAR ANS PLUS 010 B B;  
PAR 052 B;  
PAR 012 B;

(MES2)  
PAR 0 B B;  
PAR 0 B B;  
PAR 052 B;  
PAR 012 B;

(ERFIN)  
SA MES PLUS 15;  
JP PRINT;

(FININS)  
LA MES PLUS 17;  
SA MES PLUS 11;

(PRINT)  
LA PRINT1;  
AD PRINT2;  
SA PRINT1;  
LAS ANS;  
SA PRINT2;  
PAR 0 B;

(PRINT1)  
PAR 0 B B;  
SAN 0 MES PLUS 17;  
STP 4;  
PAR 3 B;

(PRINT2)  
PAR 0 B B;  
LS3 030;  
JP WMES;

(ERROR)  
PAR 0 B B;  
LA ANS;



```

JS ERROR1;
CO ISO;
JE;
JP ERFIN;
JP I ERROR;

```

```

(ERROR1)
PAR 0 B B;
SA ERROR12;
LA P7;
ML ERROR12;
AD ISO;
SA ERROR13;
LA P7;
LL 3;
ML ERROR12;
LL 5;
AD ERROR13;
JP I ERROR1;

```

```

(ERROR12)
PAR 0 B B;
(ERROR13)
PAR 0 B B;

```

```

(INT2)
PAR 0 B B;
LA P7;
ML X 0;
LL 3;
SA INT21;
LA P7;
ML X 1;
AD INT21;
XDS 3;
JP I INT2;

```

```

(INT21)
PAR 0 B B;

```

```

(INT3)
PAR 0 B B;
LA P3;
ML X 0;
LL 6;
SA INT31;
XDS 1;
JS INT2;
AD INT31;
JP I INT3;

```

```

(INT31)
PAR 0 B B;
(LOOK1)
PAR 0 B B;
SA LOOK11;

```



SX LOOK12;  
 LS3 0;  
 LX P3 PLUS 1;  
 LXS 0200;

(LOOK13)  
 LA X 1;  
 ML P0377;  
 ADS 3;  
 CO P7;  
 JG;  
 JP I LOOK1;  
 SUS 3;  
 LL 5;  
 SA LOOK14;  
 ML P0377;  
 AD P3 PLUS 1;  
 OPT AX;  
 LA LOOK14;  
 LAS 044;  
 SA LOOK14;  
 PAR 017 B;

(LOOK14)  
 PAR 0 B;  
 LA X 011;  
 ML P0377;  
 CO LOOK11;  
 JE;  
 JP LOOK13;  
 OPT XA;  
 LX LOOK12;  
 JP I LOOK1;

(LOOK11)  
 PAR 0 B B;

(LOOK12)  
 PAR 0 B B;

(WINT3)  
 PAR 0 B B;  
 JS WINT21;  
 LA P7;  
 XSS 1;  
 ML I WINT22;  
 LL 8;  
 JS WINT1;  
 JP I WINT3;

(WINT21)  
 PAR 0 B B;  
 SA WINT22;  
 LA P3;  
 LL 6;  
 ML I WINT22;



```

LL 2;
AD ISO;
SA X 0;
XDS 1;
LA P7;
LL 3;
ML I WINT22;
LL 5;
JS WINT1;
JP I WINT21;

```

```

(WINT22)
PAR 0 B B;

```

```

(WINT2)
PAR 0 B B;
JS WINT21;
JP I WINT2;

```

```

(WINT1)
PAR 0 B B;
AD ISO;
LAS 054;
SA X 0;
XDS 2;
LA WINT22;
ADS 1;
JP I WINT1;

```

```

(W21)
PAR 0 B B;
ADS 020;
LXS ANS PLUS 21;
SX PRINT1;
JS WINT3;
JS WINT3;
LX ANS PLUS 27;
SX ANS PLUS 25;
LX ZERO;
LXS ANS PLUS 27;
JS WINT3;
JP I W21;

```

```

(LSEG3)
PAR 0 B B;
JS INT2;
LL 8;
LAS 0251;
SA LSEG31;
PAR 15 B;

```

```

(LSEG31)
PAR 0 B B;
PAR LSEG3 B;

```



(RP)  
 LXS ANS PLUS 4;  
 JS INT2;  
 LSO 024;  
 LL 2;  
 ADS 1;  
 SA 0303;  
 AD RP6;  
 JF RP1;

(SM)  
 LXS ANS PLUS 4;  
 JS INT2;  
 LL 2;  
 SA SM1;  
 AD P3 PLUS 1;  
 SA SMRWB;

(SM1)  
 PAR 0 B B;  
 PAR ANS PLUS 8 B;  
 STP 4;  
 JF SMRWB;

(RWB)  
 LXS ANS PLUS 4;  
 JS INT2;  
 LL 2;  
 SA RWB1;  
 SA RWB3;  
 AD P3 PLUS 1;  
 SA RWB2;  
 SA SMRWB;

(RWB1)  
 PAR 0 B B;  
 PAR MES1 B;

(RWB2)  
 PAR 0 B B;  
 PAR ANS B;  
 JS ERROR;  
 JS LSEC3;  
 JS INT3;  
 SA RWB1;  
 AD P3 PLUS 1;  
 SA MES2 PLUS 2;  
 JS INT3;  
 SU RWB1;  
 ADS 1;  
 SA MES2;

(RWB3)  
 PAR 0 B B;



PAR MES2 B;  
(SMRWB)  
PAR O B B;  
PAR ANS B;  
JS ERROR;  
JP FININS;

(R1)  
JS LOOK;  
ADS 014;  
JS RINT3;  
JS RINT3;  
SUS 4;  
JS RINT3;

(R11)  
JS RINT3;  
JP FININS;

(R2)  
JS LOOK;  
ADS 020;  
JS RINT3;  
JS RINT1;  
JP R11;

(R3)  
JS LOOK;  
ADS 023;

(R31)  
JS RINT2;  
JS RINT2;  
JP FININS;

(R4)  
JS LOOK;  
ADS 025;  
JP R31;

(RM)  
LXS ANS PLUS 4;  
JS LSEG3;  
JS INT3;  
AD P3 PLUS 1;  
SUS 1;  
JP R11 MINUS 2;

(PC)  
JS LOOK;  
ADS 010;  
JS RINT1;  
LA I COE1;  
ADS 2;



SA I COE1;  
JP FININS;

(COE)  
PAR 0 B B;  
SA COE1;  
LA I COE1;  
LAS 0;  
SA COE2;  
LA X 0;  
CO E;  
JE;  
JP I COE;  
XDS 2;  
JP RINTF1;

(COE1)  
PAR 0 B B;

(COE2)  
PAR 0 B B;

(RINTF)  
AD COE2;  
SA I COE1;

(RINTF1)  
LA COE;  
SUS 4;  
AD RINTF2;  
SA COE;  
LA COE1;  
ADS 1;  
JP I COE;

(RINTF2)  
PAR 010 B;  
PAR 0 B;

(RINT3)  
PAR 0 B B;  
JS COE;  
JS INT3;  
JP RINTF;

(RINT2)  
PAR 0 B B;  
JS COE;  
JS INT2;  
LL 3;  
ADS 1;  
JP RINTF;

(RINT1)



PAR 0 B B;  
JS COE;  
XDS 2;  
ML P7;  
JP RINTF;

(W1)  
JS LOOK;  
ADS 014;  
LXS ANS PLUS 15;  
SX PRINT1;  
JS WINT3;  
JS WINT3;  
SUS 4;

(W11)  
JS WINT3;  
JS WINT3;  
JP FININS;

(W2)  
JS LOOK;  
JS W21;  
JP FININS;

(W3)  
JS LOOK;  
ADS 023;  
LXS ANS PLUS 25;  
SX PRINT1;  
JS WINT2;  
JS WINT2;  
JP FININS;

(W4)  
JS LOOK;  
ADS 025;  
JP W3 PLUS 4;

(WM)  
LXS ANS PLUS 4;  
JS LSEG3;  
JS INT3;  
AD P3 PLUS 1;  
SUS 1;  
LXS ANS PLUS 23;  
SX PRINT1;  
JP W11;

(LOOK)  
PAR 0 B B;  
LXS ANS PLUS 4;  
JS INT2;  
JS LOOK1;  
CO P7;



```

JE;
JP I LOOK;
JS ERROR1;
JP ERFIN;

```

```

ZSEG 030N
ADR := 3 S 0
(START)
JS MOD;
CAP 4 P4 031 6;
CAP 5 P5 031 0;
ST1 5;
SME 2 MES;
WAN 2 ANS;

```

```

(WMES)
LX ZERO;
WME 0 ANS;
ST2 4 P4;
LA ANS, MINUS 1;
LL 10;
SA PRINT2;
ZA;
LAS MES;
SA PRINT1;
LA E;
SA MES PLUS 5;
SA MES PLUS 11;
LA ANS PLUS 2;
SA MES PLUS 6;
LA ANS PLUS 5;
SA MES PLUS 9;
JS ERROR;

```

```

(COMP)
LA X CODE;
CO ANS PLUS 2;
JE;
JI;
JP I X ADRESS;
CO ERINS;
XDS 2;
JE;
JP COMP;
JP ERFIN;

```

```

(CIOE)
JS CONSINS;
JS INT2;
AD CODE1 PLUS 1;
SUS 036;
SA CODE1 PLUS 1;

```

```

(CODE1)
PAR 0 B B;
PAR 0 B B;
JP FININS;

```



```

(CONSINS)
PAR 0 B B;
OPT XA;
LL 7;
SA CODE1;
LXS ANS PLUS 4;
JS INT2;
LL 2;
AD CODE1;
SA CODE1;
LA ZAZA;
SA CODE1 PLUS 2;
JP I CONSINS;

```

```

(DSS)
JS CONSINS;
JP CODE1;

```

```

(CA)
JS CONSINS;
LA P3;
ML X 0;
AD CODE1;
SA CODE1;
XDS 2;
JS INT3;
OPT CHA;
SA CODE1 PLUS 2;
JS INT2;
LL 3;
AD CODE1 PLUS 2;
SA CODE1 PLUS 2;
LA P7;
ML X 0;
AD CODE1 PLUS 2;
SA CODE1 PLUS 2;
JP CODE1;

```

```

(RS)
SAN 0 MES PLUS 17;
STP 4;
STP 5;
DEP 4;
DEP 5;
LS0 024;

```

```

(RS1)
LS1 011;
LA 1 S 81;
ADS 1;
SA 1 S 81;
LS1 023;
LA 1 S 0370;
LAS 024;

```



SA 1 S 0370;  
SA 0 S 0367;  
JP 0 S 37;

(MOD)  
PAR 0 B B;  
LS1 011;  
LA 1 S 81;  
SUS 1;  
SA 1 S 81;  
LS0 025;  
LS1 026;  
LS2 027;  
JP I MOD;

(RP1)  
SA 0306;  
LA RP3;  
SA 0334;  
LA RP4;  
SA 0336;  
JP RS1;

(RP2)  
OPT AX;  
ZA;  
LAS 011;  
SA 0336;  
LAS 015;  
SA 0303;  
AD RP6;  
SA 0308;  
LA RP5;  
SA 0334;  
JS MOD;  
OPT XA;  
CD ZERO;  
JG;  
JP FININS;  
SUS 060;  
JS ERROR1;  
JP ERFIN;

(RP3)  
LS3 030;

(RP4)  
PAR 0240 S RP2 B B;

(RP5)  
PAR 0101 B;  
PAR 0364 B;

(RP6)



PAR 3 S 0 B B;  
ZSEG 031N  
ADR := 2 S 0

(P4)  
LS0 025;  
LS1 026;  
LS3 0;

(P41)  
LX P3 PLUS 1;  
LXS 0240;

(P42)  
LA X 0;  
ML P0377;  
SUS 1;  
CO ZERO;  
JE;  
JP P43;  
LAS MES;  
SA PRINT1;  
JS P451;  
ST2 1 PRINT;

(P43)  
OPT XA;  
CO P44;  
XDS 2;  
JE;  
JP P42;  
JP P41;

(P44)  
PAR 3 S 0276 B B;

(P451)  
PAR 0 B B;  
LA X 1;  
ML P0377;  
JS ERROR1;  
SA MES PLUS 15;  
ZA;  
SA X 1;  
JP I P451;

(P5)  
LS0 025;  
LS1 026;



(P51)  
LX P3 PLUS 1;  
LXS 0240;

(P52)  
LS3 0;  
LA X 1;  
CO ZERO;  
JE;  
JP P53;  
OPT XA;  
CO P44;  
XDS 2;  
JE;  
JP P52;  
JP P51;

(P53)  
STP 1;  
SX P56;  
LA E;  
SA MES PLUS 5;  
SA MES PLUS 11;  
LA CODE PLUS 3;  
SA MES PLUS 6;  
LA X 0;  
ML P0377;  
SA P55;  
JS ERROR1;  
SA MES PLUS 9;  
JS P451;  
LA P55;  
JS LOOK1;  
CO P7;  
JE;  
JP P57;  
ST1 1;  
LX P56;  
JP P52;

(P57)  
LX ZERO;  
JS W21;  
SME 2 ANS PLUS 21;  
ST1 1;  
WAN 2 ANS;  
JP P57 MINUS 4;  
(P55)  
PAR 0 B B;  
(P56)  
PAR 0 B B ;  
ZPROGENDN



