

KOMPUTER WIRTUALNY WIK-2

Dokumentacja oprogramowania

Sprawozdanie z realizacji zadania badawczego
nr 1 tematu "Kompleksowe sterowanie ruchem
ulicznym w podobszarze miejskim" (nr 43-005/13)

Opracował pod kierunkiem
dra inż. Jacka Martinka
mgr JERZY BARTOSZEK

Poznań, czerwiec 1977 rok

Zleceniodawca:

Instytut Automatyki Politechniki
Poznańskiej
jako podzlecenie w temacie nr
43-005/13 "Kompleksowe sterowanie
ruchem ulicznym w podobszarze
miejskim"

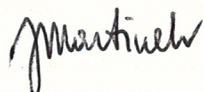
Zleceniobiorca:

Środowiskowy Ośrodek Informatyki
Politechniki Poznańskiej

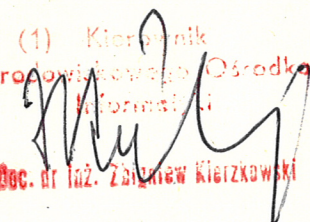
Wykonawcy:

Zespół Środowiskowego Ośrodka
Informatyki Politechniki Poznańskiej

Kierownik zadania



dr inż. Jacek Martinek

(1) Kierownik
Środowiskowego Ośrodka
Informatyki

Doc. dr inż. Zdzisław Kierzkowski

SPIS TREŚCI

	Strona
1. Wstęp	2
2. Organizacja logiczna	3
2.1. Pamięć operacyjna	3
2.2. Strony sterowania	4
2.3. Procesory wirtualne	4
2.4. Przerwania	7
3. Struktury danych wewnętrznych	8
4. Rozkazy wirtualne	15
5. Podprogramy realizujące WIK-a i rozkazy wirtualne	23
5.1. Podprogramy realizujące WIK-a	26
5.2. Podprogramy realizujące rozkazy wirtualne	37
6. Uruchamianie WIK-a	83
7. Cytowane opracowania	86
Dodatek A	
Dodatek B	

1. WSTĘP

Wirtualny komputer WIK-2 został zrealizowany na systemie MERA 300 w oparciu o projekt opracowany w Instytucie Automatyki Politechniki Poznańskiej (patrz [2]).

WIK-2 może być wykorzystany do współbieżnego wykonywania programów, a w szczególności do sterowania w czasie rzeczywistym. Z punktu widzenia programisty jest on komputerem wieloprocesowym. Wirtualne procesory mogą być tworzone, startowane, zatrzymywane i usuwane. Ponadto mogą synchronizować się nawzajem za pomocą przerwań oraz komunikatów.

Stosunkowo ubogą listę rozkazów systemu MERA 300 uzupełniono tzw. rozkazami wirtualnymi. Wykonywane są one przez podprogramy zapisane w instrukcjach systemu MERA 300 w ramach obsługi przerwania programowego. Rozkazy wirtualne ułatwiają adresację pamięci operacyjnej (poprzez adresację pośrednią oraz indeksową), programowanie operacji wejścia i wyjścia, obsługę przerwań oraz sterowanie procesorami. Ponadto umożliwiają one tworzenie nowych rozkazów wirtualnych oraz usuwanie nie wykorzystywanych.

WIK-2 może współpracować ze wszystkimi urządzeniami zewnętrznymi przewidzianymi w systemie MERA 300.

2. ORGANIZACJA LOGICZNA

2.1. Pamięć operacyjna

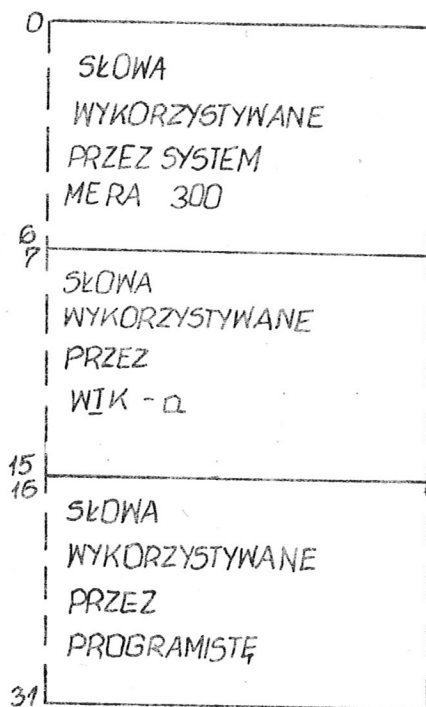
W komputerze WIK-2 pamięć operacyjna może być adresowana dwoma sposobami. Pierwszy z nich zgodny jest z zasadami adresacji w systemie MERA 300. Drugi sposób wymaga stosowania rozkazów wirtualnych. Każde słowo pamięci posiada wówczas 13-bitowy adres. Ignoruje się podział pamięci na strony. Możliwe jest stosowanie modyfikacji pośredniej oraz indeksowej. Do modyfikacji indeksowej wykorzystuje się cztery X-rejestry (X0-X3).

W dalszych opisach zakłada się, że w przypadku pamiętania adresu w dwu kolejnych słowach, pierwsze z nich zawiera 5 najstarszych bitów a drugie 8 pozostałych.

Strona numer zero pamięci operacyjnej (rys.1) wykorzystywana jest we WIKU w sposób specjalny. Dla każdego wirtualnego procesora przechowywany jest odrębny jej egzemplarz (słowa 4-6 oraz 16-31). Słowa 0-6 wykorzystywane są przez system MERA w sposób opisany w [1], a słowa 7-15 wykorzystywane są przez WIK-a. Z tego względu używanie tych słów przez programistę jest ograniczone. Sposób wykorzystania pozostałych słów należy wyłączyć do programisty. W szczególności, w pierwszym sposobie adresacji, słowa 16-23 mogą być wykorzystywane do obliczania adresu efektywnego (patrz [1]), a w drugim sposobie, pary słów: 24-25, 26-27, 28-29, 30-31 jako rejestry X0, X1, X2, X3.

Strony numer 1, 2, 3 wykorzystywane są zgodnie z ich przeznaczeniem w systemie MERA.

Każdy procesor może wykorzystywać dowolny fragment pamięci operacyjnej, z wyjątkiem obszaru przeznaczonego na cele organizacyjne WIK-a.



Rys.1. Schemat strony numer zero

2.2. Strony sterowania

Z obszaru przeznaczanego na cele organizacyjne WIK-a zdecydowano się wyodrębnić 20 stron i wykorzystywać je jako porcje pamięci do sterowania procesorami. Taką porcję pamięci nazywa się stroną sterowania. Strona sterowania może być pusta lub zajęta przez informację sterującą. Wtedy, przechowuje się na niej informacje określające jeden procesor wirtualny lub zawiera ona komunikat.

2.3. Procesory wirtualne

Procesory wirtualne działają współbieżnie dzięki podziałowi czasu procesora fizycznego. Każdy procesor wirtualny posiada nazwę oraz priorytet, które są liczbami z zakresu 1-15. Procesory mogą być tworzone oraz usuwane. We WIK-u może istnieć maksymalnie 15 procesorów. Tworzenie procesora, jego usuwanie, nadawanie nazwy oraz priorytetu odbywa się przez

wykonanie odpowiedniego rozkazu wirtualnego. Procesor wirtualny może znajdować się w jednym ze stanów: przetwarzania, zatrzymania, czekania lub zatrzymania i czekania.

W stanie przetwarzania procesor może wykonywać swoje zadania, o ile zostanie mu przydzielony procesor fizyczny. Procesor fizyczny jest przydzielany procesorowi wirtualnemu, który aktualnie posiada najwyższy priorytet.

W stanie zatrzymania procesor wirtualny istnieje w komputerze WIK, ale nie wykonuje żadnych zadań. Stan czekania oznacza, że procesor oczekuje na wolną stronę sterowania, komunikat od innego procesora lub przerwanie. Podstawowe rejestry procesora wirtualnego są identyczne z rejestrami procesora fizycznego.

Są to zatem:

- 8-bitowy rejestr akumulatora (A),
- 8-bitowy rejestr strony (S),
- 13-bitowy licznik rozkazów (LR),
- 4-bitowy rejestr przyjęć przerw (W),
- 1-bitowy rejestr przeniesienia (P),
- 1-bitowy rejestr strony numer zero (Z),
- 1-bitowy rejestr skoku warunkowego (CI).

Rejestry te wykorzystywane są we WIK-u zgodnie z zasadami przyjętymi w systemie MERA 300. W szczególności przy pomocy rejestru W można określać klasy przerw przyjmowanych w trakcie działania procesora wirtualnego. Stan procesora pamiętany jest w 2-bitowym rejestrze stanu SW. Bit SS tego rejestru wskazuje stan zatrzymania, a bit WS - stan czekania. Ponadto każdy procesor wirtualny posiada cztery 13-bitowe X-rejestry wykorzystywane do adresacji pamięci operacyjnej. W trakcie

działania procesora pamiętana są one na stronie numer zero. Procesor wirtualny może wykonywać rozkazy systemu MERA (z wyjątkiem rozkazów: "przerwanie programowe" i "wyjdz z obsługi przerwania" o kodach odpowiednio: 224 i 228 czyli 340_8 i 344_8) oraz rozkazy wirtualne. Rozkazy wirtualne wykonywane są w ramach obsługi przerwania programowego. W danej chwili we WIK-u może istnieć maksymalnie 16 rozkazów tego typu (o kodach od 0 do 15).

Wykorzystując odpowiednie rozkazy wirtualne procesory mogą synchronizować się wzajemnie za pomocą komunikatów oraz przerwania. Każdy komunikat składa się z następujących elementów dostępnych dla programisty:

- nazwy nadawcy,
- znacznika, który jest liczbą z zakresu 1-7 i umożliwia rozróżnianie komunikatów od tego samego nadawcy, a tym samym wielopoziomowe przesyłanie informacji,
- wskaźnika określającego długość treści czyli ilość przesyłanych słów (od 0 do 29),
- treści.

Zakłada się, że komunikaty nadesłane do procesora przez jednego nadawcę i posiadające jednakowe znaczniki mogą być wykorzystane tylko w kolejności ich nadesłania. Założenie to nie dotyczy komunikatów nadesłanych przez różnych nadawców lub posiadających różne znaczniki.

W przypadku wysłania komunikatu do procesora nieistniejącego w komputerze, procesor - nadawca otrzymuje automatycznie od WIK-a komunikat stwierdzający ten fakt. Jako nadawcę takiego komunikatu przyjmuje się wówczas nieistniejący procesor, jego znacznik jest identyczny ze znacznikiem błędnie wysłanego

komunikatu, a wskaźnik określający długość treści ma wartość 0.

2.4. Przerwania

W komputerze WIK-2 procesory mogą synchronizować się za pomocą przerwania. Wykorzystując rozkazy wirtualne mogą one oczekiwać na jednokrotne lub wielokrotne wystąpienie danego przerwania, a także niektóre z nich generować.

Wyróżniono następujące przerwania:

- procesorowe (PRI),
- wejścia i wyjścia (IOI),
- zewnętrzne (E3I, E2I, E1I),
- czasu rzeczywistego (RTI).

Przerwania PRI generowane są przez procesory przy pomocy rozkazów wirtualnych. Posiadają one numery identyczne z nazwami procesorów, które je generują tj. z przedziału 1-15 ($1-17_8$).

IOI reprezentują we WIK-u te przerwania, które w systemie MERA 300 należą do klasy wejścia/wyjścia (tj. klasy numer 4). We WIK-u posiadają numery z zakresu 16-47 (20_8-57_8).

Przerwaniu numer 0 klasy wejścia/wyjścia przyporządkowano we WIK-u numer 16, przerwaniu numer 1-17 itd.

E3I, E2I, E1I związane są z przerwaniami zewnętrznymi systemu MERA. W komputerze WIK posiadają one numery odpowiednio: 48, 49, 50 ($60_8, 61_8, 62_8$). E3I reprezentuje wszystkie przerwania zewnętrzne klasy 3 występujące w systemie MERA. Pojawienie się dowolnego przerwania należącego do klasy 3 powoduje wystąpienie przerwania E3I we WIK-u, a w akumulatorze każdego procesora czekającego na E3I umieszcza się automatycznie numer przerwania w ramach klasy (0-31).

W ten sposób jeden procesor wirtualny może "obsługiwać" całą klasę przerwania zewnętrznych. Przerwania E2I i E1I dotyczą przerwania zewnętrznych klas odpowiednio 2 i 1 i są analogicznie wykorzystywane we WIK-u. Ponadto w komputerze WIK wprowadzono cztery przerwanie RTI dotyczące czasu rzeczywistego. Posiadają one numery z zakresu 51-54 (63_8 - 66_8) i są generowane przez WIK-a. Częstotliwość ich występowania jest całkowitą wielokrotnością częstotliwości przerwania fizycznego zegara czasu rzeczywistego.

Zakłada się, że przerwanie o numerach 51-54 będą występowały jeden raz w okresie:

- 51 - 0,1 sekundy,
- 52 - 1 sekundy,
- 53 - 1 minuty,
- 54 - 1 godziny.

Zgodnie z zasadami obowiązującymi w systemie MERA 300 przyjęto, że w komputerze WIK przerwanie o wyższym numerze posiada wyższy priorytet. Fakt ten wpływa pośrednio na sposób nadawania priorytetów procesorom, a mianowicie: procesory "obsługujące" przerwanie o wyższym numerze powinny posiadać wyższe priorytety.

3. STRUKTURY DANYCH WEWNĘTRZNYCH

WIK wykorzystuje swoje własne, wewnętrzne struktury danych. Są nimi strony sterowania i tablice. Ponadto strony sterowania zorganizowane są w listy i kolejki.

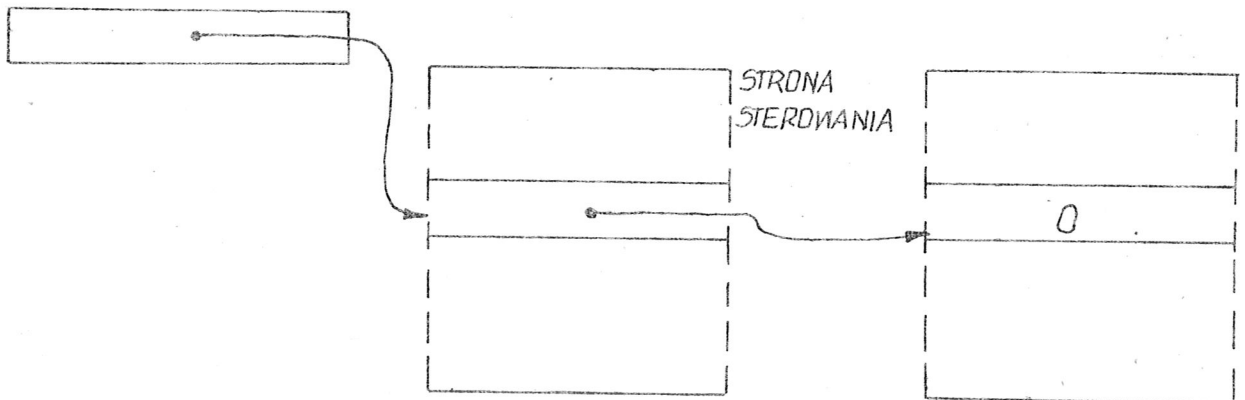
W dalszych opisach będą stosowane następujące oznaczenia: PR-procesor, NK-nazwa k-tego procesora, M-komunikat (message), F-znacznik (flag), EP-strona pusta (empty page), T-tablica,

L-lista, Q-kolejka (queue), P-łącznik (pointer), S-semafor, I-przerwanie (interrupt), IK-przerwanie numer K, A-adres, AK-k-ty adres, PRIOR-priorytet, ERR-błąd.

W tym systemie oznaczeń np. skrót PR-NK-M-Q oznacza kolejkę komunikatów do procesora o nazwie NK, PR-NK-M-Q-P jest łącznikiem w tej kolejce, a PR-NK-M-Q-S jest semaforem z nią związanym.

Omówimy najpierw rodzaje i strukturę kolejek i list WIK-a a następnie strukturę stron sterowania oraz strukturę i rodzaje tablic. Kolejki i listy zorganizowane są w sposób przedstawiony na rys.2.

NAGŁÓWEK KOLEJKI (LISTY)



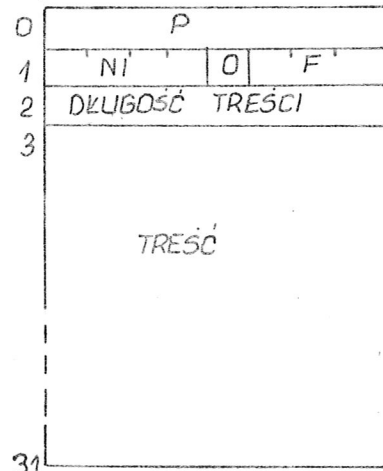
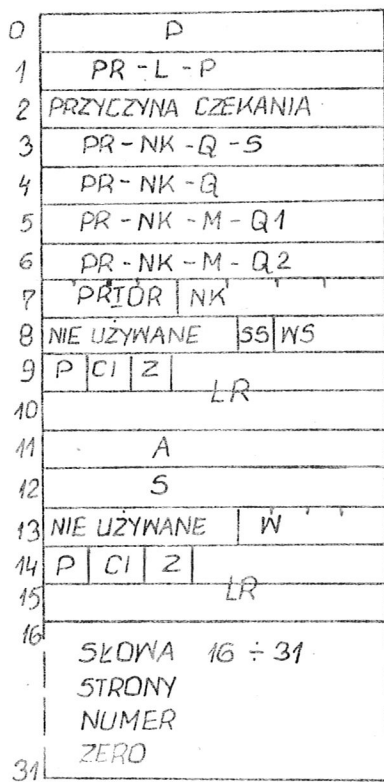
Rys.2. Schemat kolejek i list we WIK-u

W komputerze WIK z dowolnym przerwaniem IK związana jest kolejka IK-PR-Q oraz semafor IK-PR-Q-S. Znajdujące się w tej kolejce procesory oczekują na pojawienie się (jednokrotne lub wielokrotne) przerwania IK. Procesor oczekujący na n-krotne wystąpienie przerwania IK zostaje z IK-PR-Q usunięty, jeżeli

od chwili jego umieszczenia w tej kolejce IK wystąpiło n razy ($0 < n < 256$).

Puste strony sterowania zorganizowane są w listę oznaczoną skrótem EP-L. Z listą tą związany jest semafor EP-L-S oraz kolejka procesorów oczekujących na pustą stronę EP-PR-Q. Kolejka ta obsługiwana jest według zasady: "pierwszy przyszedł - pierwszy zostanie obsłużony". Strony sterowania, które wykorzystano do przechowywania informacji o procesorach zorganizowane są w listę PR-L.

Strukturę strony sterowania zajętej przez procesor wirtualny o nazwie NK pokazano na rys. 3a.



Rys.3a. Schemat strony sterowania procesora NK

Rys.3b. Schemat strony sterowania zajętej przez komunikat

W słowie numer jeden takiej strony znajduje się łącznik listy PR-L. W zerowym słowie natomiast, pamięta się łącznik, gdy procesor jest w stanie czekania i znajduje się w jednej z kolejek WIK-a. Przyczyna czekania pamiętana jest wówczas w drugim słowie strony. Jeżeli procesor oczekuje na n-krotne wystąpienie przerwania IK (czyli znajduje się w kolejce IK-PR-Q), to słowo to zawiera n, a po każdym pojawieniu się IK jego zawartość zmniejszana jest o jeden. W przypadku, gdy procesor oczekuje na wolną stronę sterowania (znajduje się w kolejce EP-PR-Q), zawartość drugiego słowa jest równa zero. Po otrzymaniu strony, w słowie tym umieszcza się jej adres. Wreszcie, gdy procesor oczekuje na komunikat ze znacznikiem F od procesora NI, słowo to zawiera NI na bitach 0-3 oraz F na bitach 4-7. W takim przypadku procesor NK znajduje się w kolejce PR-NK-Q. Nagłówek tej kolejki pamiętany jest w słowie czwartym, a semafor z nią związany (PR-NK-Q-S) w słowie trzecim. Kolejka PR-NK-Q jest dość szczególna, bowiem albo jest pusta, albo czeka w niej tylko procesor NK. W słowach piątym i szóstym znajdują się nagłówki kolejek PR-NK-M-Q1 i PR-NK-M-Q2, czyli kolejek komunikatów do procesora NK. Wszystkie komunikaty, które nadesłano do procesora NK umieszczane są początkowo w kolejce PR-NK-M-Q1 a następnie, w wyniku wykonania pewnych rozkazów wirtualnych, umieszczane są w kolejce PR-NK-M-Q2. Słowo siódme zawiera nazwę procesora oraz jego priorytet. Stany rejestrów procesora po zakończeniu ostatnio wykonywanego rozkazu pamiętane są w słowach 8-12. W słowach od 13 do 15 pamiętane są komórki 4-6 strony numer zero. Słowa 16-31 strony zerowej przechowywane są w pozostałych komórkach.

Strukturę strony sterowania zajętej przez komunikat ilustruje rys.3b.

Zerowe słowo zawiera wówczas łącznik w kolejce PR-NK-M-Q1 lub PR-NK-M-Q2, gdzie NK oznacza odbiorcę komunikatu. Bity 0-3 słowa numer jeden zawierają nazwę nadawcy komunikatu, a bity 4-7 jego znacznik. Słowo numer dwa zawiera informację o długości treści. Pozostałe słowa strony przeznaczone są do pamiętania treści.

Obecnie omówimy wewnętrzne tablice WIK-a. Rysunek 4 przedstawia tablicę VCT zwaną tablicą komputera wirtualnego. Zapamiętane są w niej semafony i nagłówki kolejek związanych z przerwaniem, nagłówki list EP-L i PR-L oraz kolejki EP-PR-Q a także semafor EP-L-S. W tablicy tej pamięta się również numer przerw generowanych przez fizyczny zegar czasu rzeczywistego (RTC), cztery liczniki (CO-C3) umożliwiające prawidłowe generowanie przez WIK-a przerw 51-54 oraz adres strony sterowania (PRA), priorytet (PRIOR) i nazwę (N) procesora wirtualnego aktualnie wykorzystującego procesor fizyczny. Tablica VCT przechowywana jest w pamięci operacyjnej na stronach 4-7.

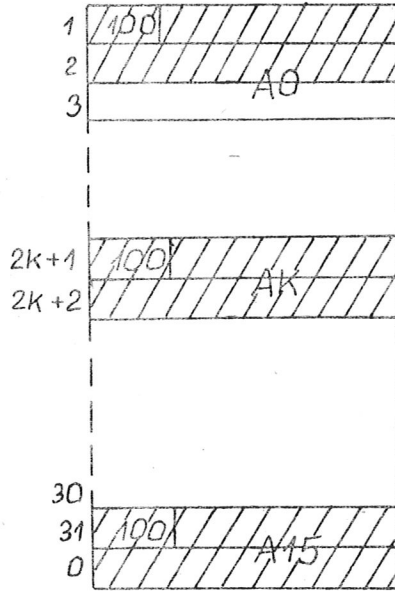
Rysunek 5a przedstawia schemat tablicy VIT. Związana jest ona z rozkazami wirtualnymi. W tablicy VIT pamiętane są rozkazy skoków do programów realizujących poszczególne (aktualnie istniejące we WIK-u) rozkazy wirtualne. W słowach 1 i 2 tablicy umieszcza się skok związany z rozkazem wirtualnym o kodzie 0, w słowach 2 i 3 - o kodzie 1 itd. Słowa odpowiadające kodom nie wykorzystanym zawierają skoki do programu stwierdzającego błąd.

We WIK-u istnieje także tablica (ERRPRT) procesorów, w trakcie działania których wykryto błędy. Jeden element tej tablicy (rys.5b) składa się z nazwy procesora oraz numeru błędu. Początek obszaru nie wykorzystanego wskazuje pierwsze

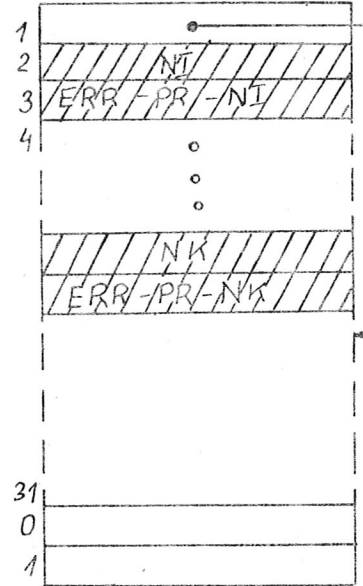
słowo w tablicy. Tablice VIT i ERRPRT przechowywane są w pamięci operacyjnej na stronach numer 8 i 9.

0	SŁOWA	0	I 15 -PR-Q	0	I 31 -PR-Q	0	I 47 -PR-Q
1	ROBOLZE	1	I 16 -PR-Q-5	1	I 32 -PR-Q-5	1	I 48 -PR-Q-5
2		2	I 16 -PR-Q	2	I 32 -PR-Q	2	I 48 -PR-Q
3	I 1 -PR-Q-5	3		3		3	
4	I 1 -PR-Q		o		o		
5	I 2 -PR-Q-5		o		o	13	I 54 -PR-Q-5
6	I 2 -PR-Q		o		o	14	I 54 -PR-Q
7			o		o	15	RTC
			o		o	16	CO
			o		o	17	C1
			o		o	18	C2
			o		o	19	C3
						20	EP-L
						21	EP-L-S
						22	EP-PR-Q
						23	PR-L
						24	PRA
						25	PRIDR N
28		28		28		26	EXIM
29	I 14 -PR-Q-5	29	I 30 -PR-Q-5	29	I 46 -PR-Q-5		
30	I 14 -PR-Q	30	I 30 -PR-Q	30	I 46 -PR-Q		
31	I 15 -PR-Q-5	31	I 31 -PR-Q-5	31	I 47 -PR-Q-5		

Rys.4. Tablica VCT

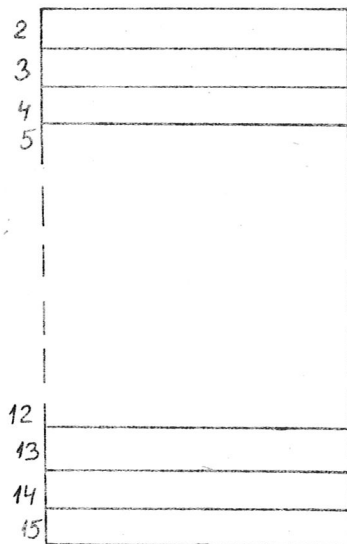


Rys.5a Tablica VIT

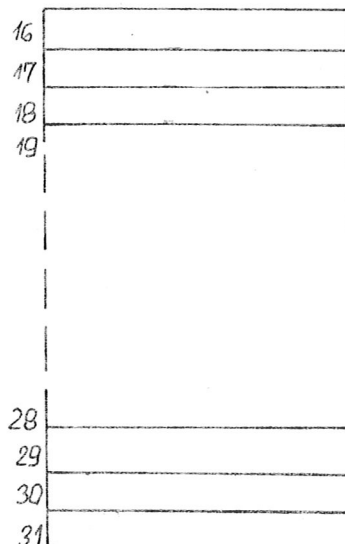


Rys.5b Tablica ERRPRT

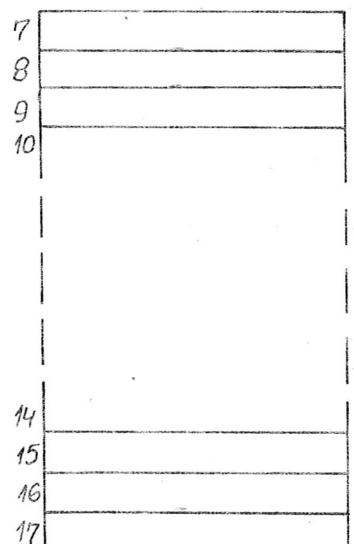
W podprogramach realizujących rozkazy wirtualne wykorzystuje się tablicę RTT (rys.6a). Używa się jej do pamiętania "śladów" podprogramów przy ich wywoływaniu. Każdy element tej tablicy składa się z dwu słów. Dokładne omówienie sposobu wykorzystania tablicy RTT znajduje się w punkcie 5.



Rys.6a Tablica RTT



Rys.6b Pole AUXF

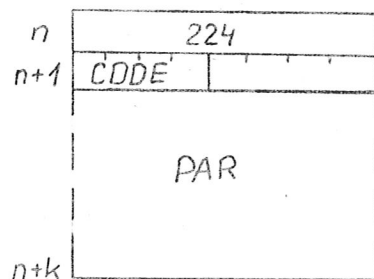


Rys.6c Pole ARGF

Argumenty podprogramów realizujących rozkazy wirtualne pamiętane są na stronie numer zero w polu ARGF (słowa 7-17). Pole to wykorzystywane jest w tych podprogramach również jako obszar roboczy. Schemat pola ARGF przedstawiono na rys. 6c. Uzupełnieniem obszaru ARGF jest pole AUXF (rys.6b). Może być ono wykorzystywane do pamiętania argumentów rozkazów lub do tymczasowego przechowywania niektórych słów z pola ARGF. Tablica RTT oraz pole AUXF pamiętane są w pamięci operacyjnej na stronie numer 10.

4. ROZKAZY WIRTUALNE

Rozkazy wirtualne realizowane są przez podprogramy zapisane w instrukcjach systemu MERA 300 w ramach obsługi przerwania programowego. Posiadają one format pokazany na rysunku 7.



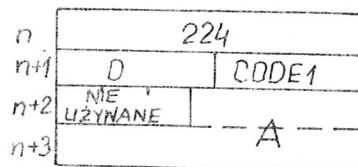
224 -- kod rozkazu "przerwanie programowe w systemie MERA 300
 CODE -- kod rozkazu wirtualnego
 PAR -- parametry rozkazu

Rys.7 Format rozkazów wirtualnych

Ilość słów zajmowanych przez rozkaz wirtualny zależy od parametrów w nim występujących. Zasady pisania podprogramów realizujących rozkazy wirtualne omówiono w punkcie 5.

Obecnie omówione zostaną poszczególne rozkazy wirtualne.

Utwórz rozkaz wirtualny - CVI



CODE1 -- kod tworzony rozkazu
 A -- adres programu realizującego tworzony rozkaz

Rys.8 Format rozkazu CVI

Rozkazem CVI można utworzyć nowy rozkaz wirtualny. Kod tworzonego rozkazu (CODE1) musi być różny od kodów wszystkich rozkazów istniejących we WIK-u w momencie tworzenia. CVI umieszcza w tablicy VIT w miejscu odpowiadającym kodowi CODE1 rozkaz skoku do początku programu realizującego tworzony rozkaz wirtualny (do adresu A). Program ten może być pamiętany w dowolnym obszarze pamięci operacyjnej dostępnym dla procesorów. Od programu realizującego rozkaz zależy również postać parametrów oraz sposób ich wykorzystania. Utworzony rozkaz wirtualny może być natychmiast wykorzystywany.

Usuń rozkaz wirtualny - DVI

n	224	
n+1	1	CODE1

 CODE1 - kod usuwanego rozkazu

Rys.9 Format rozkazu DVI

DVI usuwa z tablicy VIT skok do programu realizującego rozkaz o kodzie CODE1. Usunięty rozkaz nie może być wykorzystywany (do chwili ponownego jego utworzenia), a jego kod może być użyty do utworzenia innego rozkazu. Usunięcie rozkazu nie oznacza usunięcia z pamięci operacyjnej programu z nim związanego. Obszar zajmowany przez ten program może być jednak wykorzystywany do innych celów.

Generuj przerwanie - GIN

n	224	
n+1	2	NIE UŻYWANE

Rys.10 Format rozkazu GIN

Rozkaz ten służy do generowania przerwania przez procesory wirtualne.

Czekaj na przerwanie - WIN

n	224	
n+1	3	NIE UZYWANE
n+2	IK	
n+3	NIK	

IK - numer przerwania ($0 < IK < 59$)

NIK - ilość pojawień się przerwania
IK ($0 < NIK < 256$)

Rys.11 Format rozkazu WIN

Po wykonaniu tego rozkazu procesor zostaje umieszczony w kolejce IK-PR-Q, gdzie oczekuje na NIK-krotne pojawienie się przerwania IK.

Wyślij komunikat - SME

n	224	
n+1	4	NI
n+2	F	
n+3	A	

NI - nazwa procesora - odbiorcy komunikatu

F - znacznik komunikatu

A - adres

Rys.12 Format rozkazu SME

Rozkaz SME umożliwia wysłanie komunikatu ze znacznikiem F do procesora NI. Ilość znaków w treści komunikatu określa słowo o adresie A, Natomiast treść pamiętana jest w kolejnych słowach, poczynając od adresu A+1. Procesor wykonujący rozkaz SME zgłasza zapotrzebowanie na wolną stronę sterowania. Jeżeli takiej strony nie ma, procesor zostaje umieszczony w kolejce EP-PR-Q i przechodzi do stanu czekania. Po uzyskaniu strony umieszcza się na niej nazwę nadawcy, znacznik komunikatu, treść oraz wskaźnik określający jej długość. Tak zorganizowaną stronę sterowania umieszcza się w kolejce PR-NI-M-Q1. Jeżeli procesor NI oczekiwał na ten komunikat, to stan jego zmienia się na stan przetwarzania.

Czekaj na następny komunikat - WNM

n	224	
n+1	5	NI
n+2	F	NIE UŻYWANE

NI - nazwa procesora - nadawcy komunikatu

F - znacznik komunikatu

Rys.13 Format rozkazu WNM

Procesor NK wykonując rozkaz WNM sprawdza, czy w kolejce PR-NK-M-Q1 znajduje się komunikat ze znacznikiem F od procesora NI. Jeżeli komunikat nadesłano, to strona sterowania zawierająca komunikat zostaje umieszczona w kolejce PR-NK-M-Q2.

W przeciwnym wypadku procesor NK przechodzi w stan czekania i zostaje umieszczony w kolejce PR-NK-Q. Zerowa wartość F lub NI w rozkazie oznacza czekanie na komunikat odpowiednio z dowolnym znacznikiem lub od dowolnego nadawcy.

Czytaj komunikat - RME

n	224	
n+1	6	NI
n+2	F	A
n+3		

NI - nazwa procesora - nadawcy komunikatu

F - znacznik komunikatu

A - adres

Rys.14 Format rozkazu RME

Rozkaz RME powoduje, że nazwa nadawcy komunikatu, jego znacznik, wskaźnik określający ilość znaków w treści oraz treść zostają przepisane ze strony sterowania do kolejnych słów pamięci operacyjnej poczynając od adresu A. Strona sterowania zawierająca komunikat staje się wolną stroną.

Przed wykonaniem rozkazu RME przez procesor NK komunikat wskazany rozkazem musi znajdować się w kolejce PR-NK-M-Q2. W przeciwnym wypadku wykryty zostanie błąd. Zerowe wartości F i NI mają identyczne znaczenie jak w rozkazie WNM.

Sprawdź, czy nadesłano komunikat - TME

n	224	
n+1	7	NI
n+2	F	A
n+3		

NI - nazwa procesora - nadawcy komunikat

F - znacznik komunikatu

A - adres

Rys.15 Format rozkazu TME

Rozkazem TME można sprawdzić, czy w kolejce PR-NK-M-Q2 lub PR-NK-M-Q1 znajduje się komunikat ze znacznikiem F nadesłany przez procesor NI. Jeżeli tak jest, to w rejestrze LR procesora NK umieszcza się adres A. W przeciwnym wypadku zawartość LR zwiększana jest o 4. W rozkazie TME testowanie wykonuje się do momentu znalezienia szukanego komunikatu lub osiągnięcia końca kolejki PR-NK-M-Q1. Wszystkie testowane komunikaty znajdujące się w kolejce PR-NK-M-Q1 umieszczane są w PR-NK-M-Q2. Zerowe wartości F i NI mają identyczne znaczenie jak w rozkazie WNM.

Startuj procesor - ST1

n	224	
n+1	8	NI

NI - nazwa procesora

Rys.16 Format rozkazu ST1

Rozkaz ten powoduje przejście procesora NI w stan przetwarzania (jeżeli był w stanie zatrzymania) lub w stan czekania (gdy był w stanie zatrzymania i czekania). Procesor NI musi istnieć w komputerze WIK.

Zatrzymaj procesor - STP

n	224	
n+1	9	NI

NI - nazwa procesora

Rys.17 Format rozkazu STP

Po wykonaniu rozkazu procesor NI przechodzi w stan zatrzymania lub zatrzymania i czekania (jeżeli był w stanie czekania). Procesor NI musi istnieć w komputerze WIK.

Startuj procesor od podanego adresu - ST2

n	224	
n+1	10	NI
n+2	NIE UŻYWANE	A
n+3	A	

NI - nazwa procesora

A - adres

Rys.18 Format rozkazu ST2

Rozkaz ten powoduje umieszczenie w rejestrze LR procesora NI adresu A i przejście procesora w stan przetwarzania. Procesor NI musi istnieć we WIK-u i musi być w stanie zatrzymania.

Utwórz procesor - CRP

n	224	
n+1	11	NI
n+2	PRIOR	W
n+3	NIE UŻYWANE	A
n+4	A	

NI - nazwa tworzonego procesora

PRIOR - priorytet

W - zawartość rejestru W

A - adres - początkowa zawartość LR

Rys.19 Format rozkazu CRP

Rozkaz ten umożliwia tworzenie nowego procesora w komputerze WIK. Nazwa tworzonego procesora (NI) musi być różna od nazw wszystkich procesorów istniejących we WIK-u w momencie tworzenia.

Procesor wykonujący rozkaz CRP zgłasza zapotrzebowanie na wolną stronę sterowania. Jeżeli takiej strony nie ma, to procesor zostaje umieszczony w kolejce EP-PR-Q. Po uzyskaniu strony, procesor organizuje jej strukturę oraz umieszcza ją na liście PR-L. Utworzony procesor uzyskuje priorytet PRIOR,

stan zatrzymania oraz zerowe zawartości rejestrów A, S, P, Z, CI. Początkowe zawartości rejestrów W i LR podane są w rozkazie.

Usuń procesor - DEP

n	224	
n+1	12	NI

NI - nazwa usuwanego procesora

Rys.20 Format rozkazu DEP

W wyniku wykonania rozkazu DEP procesor o nazwie NI zostaje usunięty z komputera WIK. Procesor NI musi być w stanie zatrzymania lub zatrzymania i czekania. Strona sterowania wykorzystywana do przechowywania informacji o procesorze NI staje się wówczas wolną stroną sterowania. Wszystkie komunikaty w kolejkach PR-NI-M-Q1 i PR-NI-M-Q2 zostają potraktowane tak, jak gdyby procesor NI nie istniał w chwilach ich wysyłania.

Wirtualne rozkazy adresowe

n	224			
n+1	13	I	X	XN
n+2	CODE1	A		
n+3				

- I - bit modyfikacji pośredniej
- X - bit modyfikacji indeksowej
- XN - numer X-rejestru
- CODE1 - kod wirtualnego rozkazu adresowego
- A - adres

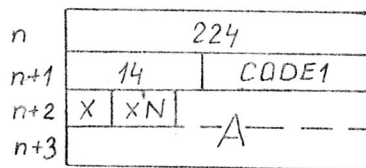
Rys.21 Format wirtualnych rozkazów adresowych

We WIK wprowadzono sześć wirtualnych rozkazów adresowych:

- dodaj do akumulatora - VAM (CODE1=0)
- pamiętaj i zeruj akumulator - VSC (CODE1=1)
- mnoż logicznie - VLM (CODE1=2)
- dodaj jeden do pamięci (VIM (CODE1=3)
- skocz - VJP (CODE1=4)
- wykonaj program - VEP (CODE1=5).

Mają one to samo znaczenie, co odpowiednie rozkazy systemu MERA 300. Różnica polega na sposobie adresacji pamięci operacyjnej oraz możliwości stosowania modyfikacji pośredniej (I=1) i indeksowej (X=1). Jeżeli w rozkazie należy wykonać obydwie modyfikacje, to najpierw wykonywana jest modyfikacja indeksowa.

Wykonaj rozkaz wirtualny z pośrednio wskazanymi argumentami -EXI



CODE1 - kod rozkazu wirtualnego, który należy wykonać

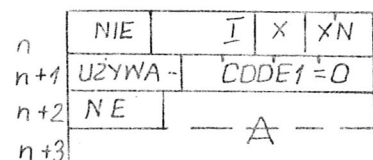
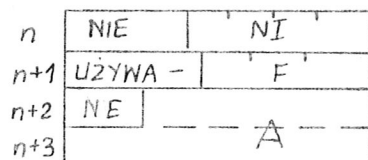
X - bit modyfikacji indeksowej

XN - numer X-rejestru

A - adres

Rys.22 Format rozkazu EXI

Rozkaz ten umożliwia wykonanie rozkazu wirtualnego o kodzie CODE1 z argumentami zapamiętanymi w kolejnych słowach pamięci operacyjnej poczynając od adresu A (modyfikowanego, gdy X=1). Dla rozkazów opisanych powyżej przyjęto, że w jednym słowie znajduje się tylko jeden argument. Nie dotyczy to argumentu A (adresu), który pamiętany jest w dwu słowach oraz argumentów I, X, XN, które zawsze pamiętane są w jednym słowie odpowiednio na bitach 4, 5, 6, 7. Na przykład dla rozkazów SME i VAM pośrednio wskazane argumenty powinny być pamiętane tak, jak pokazuje to rysunek 23.



a) Argumenty rozkazu SME

b) Argumenty rozkazu VAM

Rys.23 Pośrednio wskazane argumenty rozkazów SME i VAM

W rozkazach wprowadzonych przez programistę mogą być one pamiętane według innych zasad.

5. PODPROGRAMY REALIZUJĄCE WIK-a I ROZKAZY WIRTUALNE

Podprogramy realizujące WIK-a oraz realizujące rozkazy wirtualne wykonywane są w ramach obsługi przerwania wejścia, wyjścia, zewnętrznych oraz programowych. Podprogramy realizujące WIK-a dzielą się na dwie grupy:

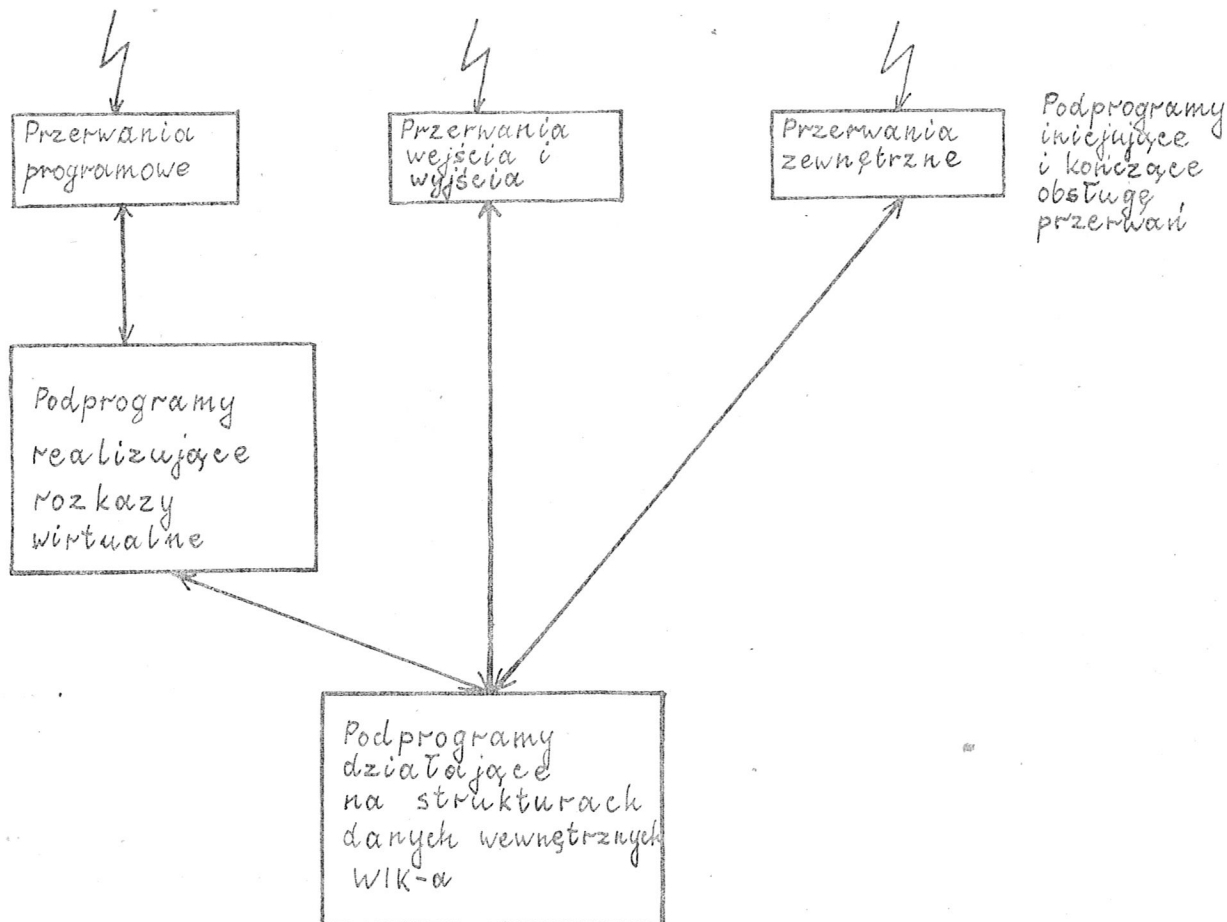
- inicjujące oraz kończące obsługę przerwania,
- działające na strukturach danych wewnętrznych WIK-a.

Z punktu widzenia programisty są one podprogramami stale istniejącymi w pamięci operacyjnej. Adresy początków tych podprogramów podano w dodatku B.

Podprogramy realizujące rozkazy wirtualne wykonywane są w trakcie obsługi przerwania programowych. Programista może wprowadzać je do pamięci operacyjnej oraz z niej usuwać (z wyjątkiem podprogramów realizujących rozkazy CVI oraz DVI, które są nieusuwalne).

W podprogramach realizujących rozkazy wirtualne można wykorzystywać wszystkie podprogramy działające na strukturach danych wewnętrznych WIK-a. Schematycznie przedstawiono to na rys.24.

Podprogramy realizujące WIK-a oraz realizujące rozkazy wirtualne muszą być zakodowane w instrukcjach systemu MERA 300. Argumenty tych podprogramów pamiętane są w polach ARGF i AUXF. Jak wspomniano poprzednio, przerwanie uaktywniają podprogramy inicjujące ich obsługę. Przy uaktywnianiu podprogramów działających na strukturach danych wewnętrznych WIK-a oraz realizujących rozkazy wirtualne wykorzystuje się rozkaz systemu MERA 300 o kodzie równym 229 - "pamiętaj ślad".



Rys.24 Podprogramy realizujące WIK-a i rozkazy wirtualne

Ze względu na sposób jego działania (patrz [1]) zachodzi konieczność programowego organizowania stosu śladów przy wielopoziomowym wywoływaniu podprogramów. Rolę takiego stosu pełni tablica RTT (rys.6a). Ponadto podprogramy realizujące WIK-a oraz realizujące rozkazy wirtualne podzielono na osiem poziomów według następujących zasad:

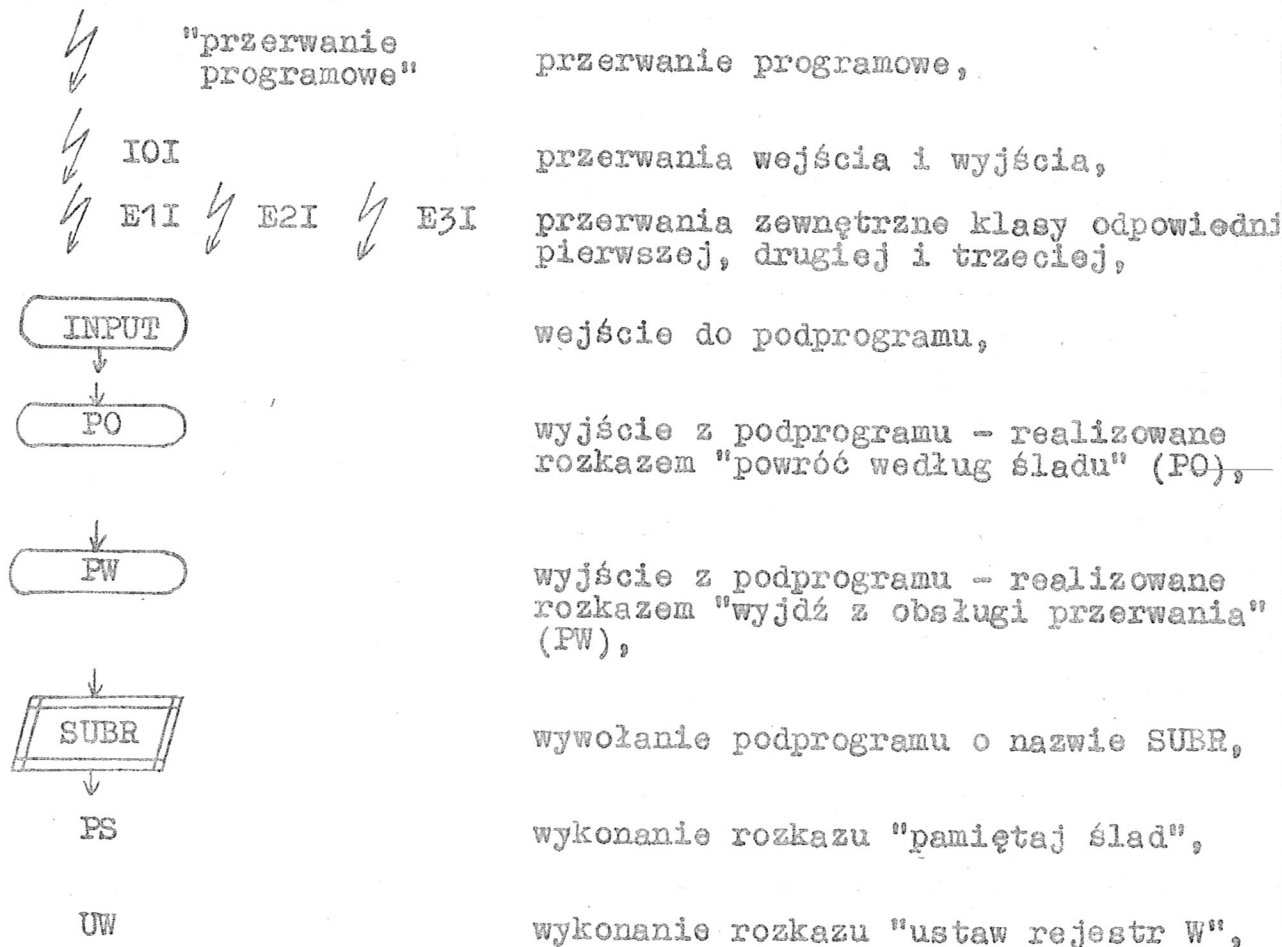
- podprogram niewykorzystujący żadnego innego podprogramu należy do poziomu 0,
- podprogram należy do poziomu $n+1$, jeżeli wykorzystuje przynajmniej jeden podprogram z poziomu n (gdzie n jest najwyższym wykorzystywanym poziomem),
- podprogramy inicjujące oraz kończące obsługę przerwania należą do poziomu 7.

Przyjęto, że do pamiętania śladu (sków 5 i 6 strony zerowej) podprogramu z poziomu n wykorzystuje się w tablicy RTT słowa numer $2n$ i $2n+1$.

Podprogram kończący obsługę przerwania musi być zakończony rozkazem o kodzie równym 228 - "wyjdź z obsługi przerwania", a pozostałe podprogramy rozkazem o kodzie równym 230 - "powrót według śladu".

Obecnie omówione zostaną poszczególne podprogramy. Wśród podprogramów realizujących WIK-a, jako pierwsze omówione zostaną te, które należą do poziomu 0, następnie do poziomu 1, itd. Podprogramy realizujące rozkazy wirtualne omówione zostaną w kolejności odpowiadającej opisom rozkazów w punkcie 4.

W zamieszczonych schematach stosuje się następujące oznaczenia:



SD	wykonanie rozkazu "czekaj",
A	zawartość akumulatora
S	zawartość rejestru strony
(\langle strona \rangle , \langle słowo \rangle)	zawartość słowa \langle słowo \rangle na stronie strona w pamięci operacyjnej,
(S, \langle słowo \rangle)	zawartość słowa \langle słowo \rangle na stronie wskazanej przez rejestr S,
\rightarrow	zmiana zawartości słowa lub rejestru,
\wedge	mnożenie logiczne - realizowane rozka- zem "mnóż logicznie",
\cdot	mnożenie arytmetyczne - realizowane rozkazem "przesuń akumulator w lewo",
\circlearrowleft	mnożenie cykliczne - realizowane roz- kazem "przesuń cyklicznie akumulator w lewo",
/	dzielenie - realizowane rozkazem "prze- suń akumulator w prawo",
\bar{A}	negacja logiczna zawartości akumula- tora,
-A	negacja arytmetyczna zawartości aku- mulatora,

Ponadto przyjęto, że wszystkie liczby występujące w schematach są liczbami ósemkowymi.

5.1. Podprogramy realizujące WIK-a

Nazwa: REMF

Poziom: 0

Dane: (0,16) - numer strony }
 (0,17) - numer słowa } adres kolejki lub listy

Działanie: REMF usuwa z kolejki lub listy pierwszą stronę ste-
rowania a jej numer umieszcza w słowie (0,15) (rys.25)

Słowa robocze: (0,14); (0,20); (0,21)

Nazwa: REMS

Poziom: 0

Dane: (0,15) - numer strony

(0,16) - numer strony }
(0,17) - numer słowa } adres kolejki lub listy

Działanie: REMS usuwa stronę wskazaną przez (0,15) z kolejki lub listy (rys.25c)

Słowa robocze: (0,14); (0,20); (0,21); (4,0)

Nazwa: JOINS

Poziom: 0

Dane: (0,15) - numer strony

(0,16) - numer strony }
(0,17) - numer słowa } adres kolejki lub listy

Działanie: JOINS umieszcza stronę wskazaną przez (0,15) na końcu kolejki lub listy (rys.25d)

Słowa robocze: (0,14); (0,20); (0,21); (4,0)

Nazwa: FINDM

Poziom: 0

Dane: (0,15) - nazwa nadawcy komunikatu oraz jego znacznik

(0,16) - numer strony }
(0,17) - numer słowa } adres kolejki

Działanie: FINDM bada, czy w kolejce znajduje się komunikat wskazany przez (0,15). Jeżeli komunikat znajduje się to w (0,15) umieszcza się jego adres. W przeciwnym wypadku w (0,15) umieszcza się zero (rys.25e)

Słowa robocze: (0,14); (0,20).

Nazwa: FINDPR

Poziom: 0

Dane: (0,15) - nazwa procesora

Działanie: FINDPR bada, czy na liście PR-L znajduje się procesor wskazany przez (0,15). Jeżeli procesor znajduje się, to w (0,15) umieszcza się jego adres. W przeciwnym wypadku w (0,15) umieszcza się zero (rys.25f).

Nazwa: CHANGEA

Poziom: 0

Dane: (0,16) }
(0,17) } 13-bitowy adres

Działanie: CHANGEA umieszcza w (0,16) osiem najstarszych bitów adresu, a w (0,17) pięć najmłodszych (rys.25g).

Nazwa: CHANGEA1

Poziom: 0

Dane: (0,16) }
(0,17) } 13-bitowy adres

Działanie: CHANGEA1 umieszcza w (0,16) osiem najstarszych bitów adresu, a w (0,17) pięć najmłodszych. Jeżeli w wyniku tego działania (0,17) zawiera zero, to zawartość(0,16) jest zmniejszona o jeden (rys.25h).

Nazwa: FETCHB

Poziom: 0

Dane: (0,15) - ilość słów ≤ 6

(0,16) - numer strony }
(0,17) - numer słowa } adres pierwszego słowa w pamięci operacyjnej

Działanie: FETCHB przepisuje słowa z pamięci operacyjnej, poczynając od adresu wskazanego przez (0,16) i (0,17) na stronę zerową do słów (0,7) - (0,14). W słowach (0,16) i (0,17) umieszczają się adres pierwszego nie przepisanego słowa (rys.25i).

Słowa robocze: (0,14); (0,20); (0,21).

Nazwa: FETCH1

Poziom: 0

Dane: (0,16) - numer strony }
 (0,17) - numer słowa } adres słowa w pamięci operacyjnej

Działanie: FETCH1 bada, czy zawartość słowa (7,32) (zmieniona przez podprogram realizujący rozkaz EXI) jest równa zero. Jeżeli tak, to nie wykonuje się żadnych działań. W przeciwnym wypadku w (0,7) umieszczają się cztery młodsze bity słowa wskazanego przez (0,16) i (0,17), a w (0,16) i (0,17) umieszczają się adres kolejnego słowa (rys.25j).

Słowa robocze: (0,20).

Nazwa: LRINCR

Poziom: 0

Dane: (0,14) - ilość słów

Działanie: LRINCR bada, czy zawartość słowa (7,32) (zmieniona przez podprogram realizujący rozkaz EXI) jest równa zero. Jeżeli tak, to zawartość słów (0,0) i (0,1) (czyli zawartość LR) zwiększana jest o wartość wskazaną przez (0,14). W przeciwnym wypadku zawartość słów (0,0) i (0,1) zwiększane są o trzy (rys.25k).

Nazwa: LRDECR1

Poziom: 0

Dane: brak

Działanie: LRDECR1 zmniejsza zawartość słów (0,0) i (0,1) (czyli zawartość LR) o jeden (rys.25l).

Nazwa: REMPR

Poziom: 0

Dane: brak

Działanie: REMPR wykonuje czynności związane ze zwolnieniem procesora fizycznego przez procesor wirtualny (rys.25m)

Słowa robocze: (0,11) - (0,21).

Nazwa: FETCHPR

Poziom: 0

Dane: (0,17) - numer strony sterowania procesora wirtualnego

Działanie: FETCHPR wykonuje czynności związane z przydzieleniem procesora fizycznego procesorowi wirtualnemu wskazanemu przez (0,17) (rys.25n)

Słowa robocze: (0,11) - (0,21).

Nazwa: SAFE2021

Poziom: 0

Dane: brak

Działanie: SAFE2021 przepisuje (0,20) i (0,21) do słów (12,36) i (12,37) w polu AUXF (rys.25b).

Nazwa: POP

Poziom: 1

Dane: (0,15) - przyczyna czekania

(0,16) - numer strony }
(0,17) - numer słowa } adres semafora

Działanie: POP (rys.26a) realizuje P-operację na semaforze wskazanym przez (0,16) i (0,17). Jeżeli wartość semafora jest mniejsza niż jeden, to procesor wirtualny aktualnie wykorzystujący procesor fizyczny przechodzi w stan czekania i zostaje umieszczony w kolejce związanej z semaforem. Jako przyczynę czekania przyjmuje się zawartość słowa (0,15). W słowie (0,14) umieszcza się wartość równą jeden. W przeciwnym wypadku, procesor wirtualny pozostaje w stanie przetwarzania, a w słowie (0,14) umieszcza się zero. W obydwu przypadkach wartość semafora jest zmniejszona o jeden.

Słowa robocze: (0,20); (0,21).

Nazwa: VOP

Poziom: 1

Dane: (0,15) - zwalniana strona sterowania lub zero

(0,16) - numer strony }
(0,17) - numer słowa } adres semafora

Działanie: VOP (rys.26b) realizuje V-operacją na semaforze wskazanym przez (0,16) i (0,17). Jeżeli wartość semafora jest mniejsza od zera, to z kolejki z nim związanej zostaje usunięty pierwszy procesor. Usunięty procesor przechodzi w stan przetwarzania (jeśli był w stanie czekania) lub w stan zatrzymania (jeśli był w stanie czekania i zatrzymania). Do słowa numer dwa jego strony sterowania zostaje przepisana zawartość (0,15). W słowie (0,14) umieszcza się wartość równą jeden. W przeciwnym wypadku zawartość słowa (0,15) nie ulega zmianie, a w (0,14) umieszcza się zero. W obydwu przypadkach wartość semafora jest zwiększana o jeden.

Słowa robocze: (0,20); (0,21).

Nazwa: VOPS

Poziom: 1

Dane: (0,14) - strona sterowania procesora

(0,15) - zwalniana strona sterowania lub zero

(0,16) - numer strony

(0,17) - numer słowa

} adres semafora

Działanie: VOPS (rys.26c) realizuje szczególnego typu V-operację na semaforze wskazanym przez (0,16) i (0,17). Zwiększa on wartość semafora o jeden, a z kolejki z nim związanej usuwa procesor wskazany przez (0,14). Usunięty procesor przechodzi w stan przetwarzania (jeśli był w stanie czekania) lub w stan zatrzymania (jeśli był w stanie czekania i zatrzymania). Do słowa numer dwa jego strony sterowania zostaje przepisana zawartość (0,15).

Słowa robocze: (0,20); (0,21).

Nazwa: SCHEDS

Poziom: 1

Dane: (0,17) - strona sterowania procesora

Działanie: SCHEDS (rys.26d) porównuje priorytet procesora wirtualnego aktualnie wykorzystującego procesor fizyczny z priorytetem procesora wskazanego przez (0,17) (o ile jest on w stanie przetwarzania). Następnie procesor fizyczny zostaje przydzielony temu procesorowi wirtualnemu, którego priorytet jest wyższy.

Słowa robocze: (0,7) - (0,21).

Nazwa: SCHED

Poziom: 1

Dane: brak

Działanie: SCHED (rys.26e) przydziela procesor fizyczny temu

procesorowi wirtualnemu we WIK-u, który jest w stanie przetwarzania i aktualnie posiada najwyższy priorytet. W przypadku, gdy żaden z procesorów wirtualnych nie znajduje się w stanie przetwarzania procesor fizyczny wykonuje podprogram WAIT (rys.26g) zawierający rozkaz systemu MERA 300 - "czekaj".

Słowa robocze: (0,11) - (0,21).

Nazwa: FETCH3

Poziom: 1

Dane: (0,16) - numer strony }
(0,17) - numer słowa } adres argumentów w pamięci operacyjnej

Działanie: FETCH3 (rys.26f) przepisuje z pamięci operacyjnej do słów (0,7) - (0,12) argumenty rozkazów wirtualnych: CVI, SME, RME, TME, ST2, VAM, VSZ, VLM, VIM, VJP, VEP oraz EXI. Działanie podprogramu FETCH3 zależy od zawartości słowa (7,32) w tablicy VCT (zmienionej rozkazem EXI).

Słowa robocze: (0,20).

Nazwa: CHECKQ

Poziom: 2

Dane: (0,15) - numer przerwania w ramach klasy (jeśli jest to przerwanie zewnętrzne)
(0,16) - numer strony }
(0,17) - numer słowa } adres semafora związanego z przerwaniem

Działanie: CHECKQ (rys.27a) związany jest z obsługą przerwania. Przegląda on listę procesorów czekających na wystąpienie przerwania wskazanego przez (0,16) i (0,17). Z listy tej usuwane są procesory, które oczekiwały na jednokrotne pojawienie się przerwania. Każdy

taki procesor przechodzi w stan przetwarzania (jeśli był w stanie czekania) lub w stan zatrzymania (jeśli był w stanie czekania i zatrzymania). W przypadku przerwania zewnętrznego zawartość słowa (0,15) przepisywana jest do akumulatorów usuwanych procesorów. Fakt usunięcia przynajmniej jednego procesora sygnalizowany jest w słowie (0,10). U pozostałych procesorów na liście krotność oczekiwanych pojawień się przerwania (pamiętana w słowach numer dwa ich stron sterowania) zmniejszona jest o jeden.

Słowa robocze: (0,14); (0,20).

Nazwa: ERR

Poziom: 2

Dane: (0,17) - numer błędu

Działanie: ERR (rys.27b) umieszcza w tablicy ERRPRT numer błędu pamiętany w (0,17) oraz nazwę procesora wirtualnego aktualnie wykorzystującego procesor fizyczny. Procesor wirtualny przechodzi w stan zatrzymania, a procesor fizyczny może zostać przydzielony innemu procesorowi wirtualnemu.

Słowa robocze: (0,11) - (0,21).

Nazwa: SENDM

Poziom: 2

Dane: (0,15) - strona sterowania zawierająca komunikat

(0,16) - strona sterowania procesora - odbiorcy komunikatu

Działanie: SENDM (rys.27c) wykonuje czynności związane z wysłaniem komunikatu wskazanego przez (0,15) do procesora wskazanego przez (0,16).

Słowa robocze: (0,14); (0,17); (0,20); (0,21).

Nazwa: GETP

Poziom: 2

Dane: brak

Działanie: GETP (rys.27d) wykonuje czynności związane z uzyskaniem przez procesor wolnej strony sterowania.

Słowa robocze: (0,11) - (0,21).

Nazwa: ISERV

Poziom: 3

Dane: (0,15) - numer przerwania w ramach klasy (jeśli jest to przerwanie zewnętrzne)

(0,16) - numer strony }
(0,17) - numer słowa } adres semafora związanego z przerwaniem

Działanie: ISERV (rys.28a) wykonuje czynności związane z obsługą przerwania. W szczególności ISERV generuje ("w imieniu WIK-a") oraz obsługuje przerwania o numerach 51-54 ($63_8 - 66_8$).

Słowa robocze: (0,10) - (0,21).

Nazwa: ERRVI

Poziom: 3

Dane: brak

Działanie: ERRVI (rys.28b) sygnalizuje próbę wykonania rozkazu wirtualnego, który nie istnieje we WIK-u.

Słowa robocze: (0,11) - (0,21).

Nazwa: PRISERV

Poziom: 7

Dane: brak

Działanie: PRISERV (rys.29a) inicjuje oraz kończy obsługę przerw programowych. Jest on uaktywniony po wykonaniu rozkazu systemu MERA 300 "przerwanie programowe". Na podstawie zawartości słów (0,0) i (0,1) (czyli zawartości LR w chwili wystąpienia przerwania) PRISERV oblicza adres słowa w pamięci operacyjnej, w którym znajduje się kod rozkazu wirtualnego, który należy wykonać. Obliczony adres pamiętany jest w słowach (0,16) i (0,17). Następnie, wykorzystując tablicę VIT, PRISERV uaktywnia odpowiedni podprogram realizujący rozkaz wirtualny. Po wykonaniu tego podprogramu obsługa przerwania programowego zostaje zakończona.

Słowa robocze: (0,7) - (0,21)

Nazwa: IOISERV

Poziom: 7

Dane: numer przerwania pamiętany w akumulatorze procesora fizycznego.

Działanie: IOISERV (rys.29b) inicjuje oraz kończy obsługę przerw wejścia i wyjścia. Jest on uaktywniany w chwili wystąpienia takiego przerwania. Na podstawie zawartości akumulatora procesora fizycznego IOISERV oblicza adres semafora związanego z przerwaniem. Adres ten pamiętany jest w słowach (0,16) i (0,17). Następnie uaktywniony zostaje podprogram obsługujący przerwanie (ISERV). Po wykonaniu tego podprogramu obsługa przerwania zostaje zakończona.

Słowa robocze: (0,7) - (0,21).

Nazwa: EISERV

Poziom: 7

Dane: numer przerwania pamiętany w akumulatorze procesora fizycznego

Działanie: EISERV (rys.29c) inicjuje oraz kończy obsługę przerw zewnątrznych. Składa się on z trzech części: E1ISERV, E2ISERV, E3ISERV. Każda część inicjuje obsługę jednej klasy przerw zewnątrznych. Zawartość akumulatora zostaje zapamiętana w słowie (0,15). Oblicza się adres semafora związanego z daną klasą. Adres ten pamiętany jest w słowach (0,16) i (0,17). Następnie uaktywniony zostaje podprogram obsługujący przerwanie (ISERV). Po wykonaniu tego podprogramu obsługa przerwania zostaje zakończona.

Słowa robocze: (0,7) - (0,21).

5.2. Podprogramy realizujące rozkazy wirtualne

Nazwa: CVIR

Poziom: 3

Dane: (0,16) - numer strony }
(0,17) - numer słowa } adres parametrów rozkazu

Działanie: CVIR realizuje rozkaz wirtualny CVI (rys.30a)

Słowa robocze: (0,7) - (0,21).

Nazwa: DVIR

Poziom: 3

Dane: (0,16) - numer strony }
(0,17) - numer słowa } adres parametrów rozkazu

Działanie: DVIR realizuje rozkaz wirtualny DVI (rys.30b)

Słowa robocze: (0,7) - (0,21).

Nazwa: GINR

Poziom: 4

Dane: brak

Działanie: GINR realizuje rozkaz wirtualny GINR (rys.30c)

Słowa robocze: (0,7) - (0,21).

Nazwa: WINR

Poziom: 3

Dane: (0,16) - numer strony }
(0,17) - numer słowa } adres parametrów rozkazu

Działanie: WINR realizuje rozkaz WINR (rys.30d)

Słowa robocze: (0,7) - (0,21).

Nazwa: SMER

Poziom: 3

Dane: (0,16) - numer strony }
(0,17) - numer słowa } adres parametrów rozkazu

Działanie: SMER realizuje rozkaz wirtualny SME (rys.30e)

Słowa robocze: (0,7) - (0,21).

Nazwa: WNMR

Poziom: 2

Dane: (0,16) - numer strony }
(0,17) - numer słowa } adres parametrów rozkazu

Działanie: WNMR realizuje rozkaz wirtualny WNM (rys.30f)

Słowa robocze: (0,7) - (0,21).

Nazwa: RMER

Poziom: 3

Dane: (0,16) - numer strony }
(0,17) - numer słowa } adres parametrów rozkazu

Działanie: RMER realizuje rozkaz wirtualny RME (rys.30g)

Słowa robocze: (0,7) - (0,21).

Nazwa: TMER

Poziom: 2

Dane: (0,16) - numer strony }
(0,17) - numer słowa } adres parametrów rozkazu

Działanie: TMER realizuje rozkaz wirtualny TME (rys.30h)

Słowa robocze: (0,7) - (0,21).

Nazwa: ST1R

Poziom: 3

Dane: (0,16) - numer strony }
(0,17) - numer słowa } adres parametrów rozkazu

Działanie: ST1R realizuje rozkaz wirtualny ST1 (rys.30i)

Słowa robocze: (0,7) - (0,21).

Nazwa: STPR

Poziom: 2

Dane: (0,16) - numer strony }
(0,17) - numer słowa } adres parametrów rozkazu

Działanie: STPR realizuje rozkaz wirtualny STP (rys.30j)

Słowa robocze: (0,7) - (0,21).

Nazwa: ST2R

Poziom: 3

Dane: (0,16) - numer strony }
(0,17) - numer słowa } adres parametrów rozkazu

Działanie: ST2R realizuje rozkaz wirtualny ST2 (rys.30k)

Słowa robocze: (0,7) - (0,21).

Nazwa: CRPR

Poziom: 3

Dane: (0,16) - numer strony }
(0,17) - numer słowa } adres parametrów rozkazu

Działanie: CRPR realizuje rozkaz wirtualny CRP (rys.30l)

Słowa robocze: (0,7) - (0,21).

Nazwa: DEPR

Poziom: 3

Dane: (0,16) - numer strony }
(0,17) - numer słowa } adres parametrów rozkazu

Działanie: DEPR realizuje rozkaz wirtualny DEP (rys.30m).

W trakcie swego działania wykorzystuje on m.innymi podprogram V10P (rys.30n) należący do poziomu 2. Dana V10P - adres strony sterowania - pamiętana jest w słowie (0,15). Podprogram V10P wykonuje V-operację na semaforze EP-L-S. Jeżeli po wykonaniu V-operacji wartość semafora jest większa od zera, to V10P łączy stronę sterowania do listy. EP-L. W przeciwnym wypadku sygnalizowany jest (za pomocą słowa (0,11)) fakt przejścia jednego z procesorów ze stanu czekania (lub czekania i zatrzymania) do stanu przetwarzania (lub zatrzymania). V10P jest wprowadzany do pamięci operacyjnej oraz z niej usuwany równocześnie

z podprogramu DEPR.

Słowa robocze: (0,7) - (0,21).

Nazwa: VAIR

Poziom: 2

Dane: (0,16) - numer strony }
(0,17) - numer słowa } adres parametrów rozkazu

Działanie: VAIR realizuje wirtualne rozkazy adresowe: VAM, VSZ, VLM, VIM, VJP oraz VEP (rys.30o)

Słowa robocze: (0,7) - (0,21).

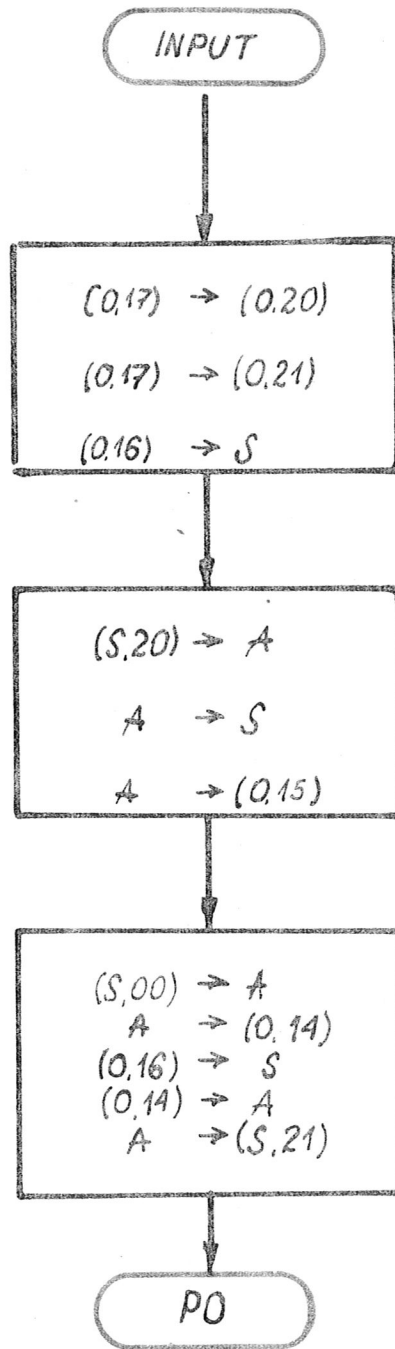
Nazwa: EXIR

Poziom: 6

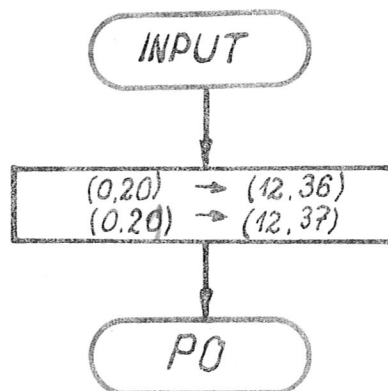
Dane: (0,16) - numer strony }
(0,17) - numer słowa } adres parametrów rozkazu

Działanie: EXIR realizuje rozkaz wirtualny EXI (rys.30p).

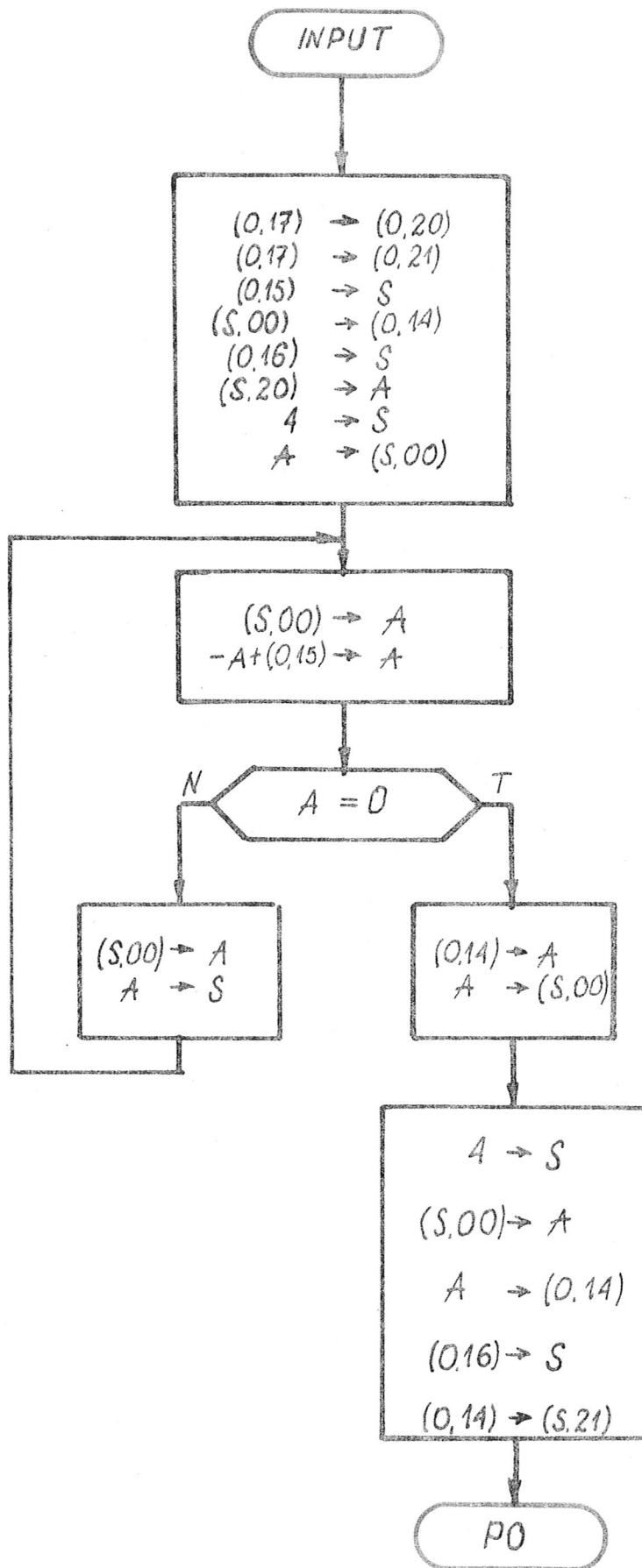
W szczególności oblicza on adres pośrednio wskaza-
nych argumentów, a następnie wykorzystując tablicę
VIT, uaktywnia podprogram realizujący odpowiedni
rozkaz wirtualny (podprogram taki nie może należeć
do poziomów 6 i 7). Obliczony adres argumentów pa-
miętany jest w słowach (0,16) i (0,17), a fakt, że
argumenty zostały wskazane pośrednio sygnalizowany
jest niezerową w zawartością słowa (7,32) w tablicy
VCT.



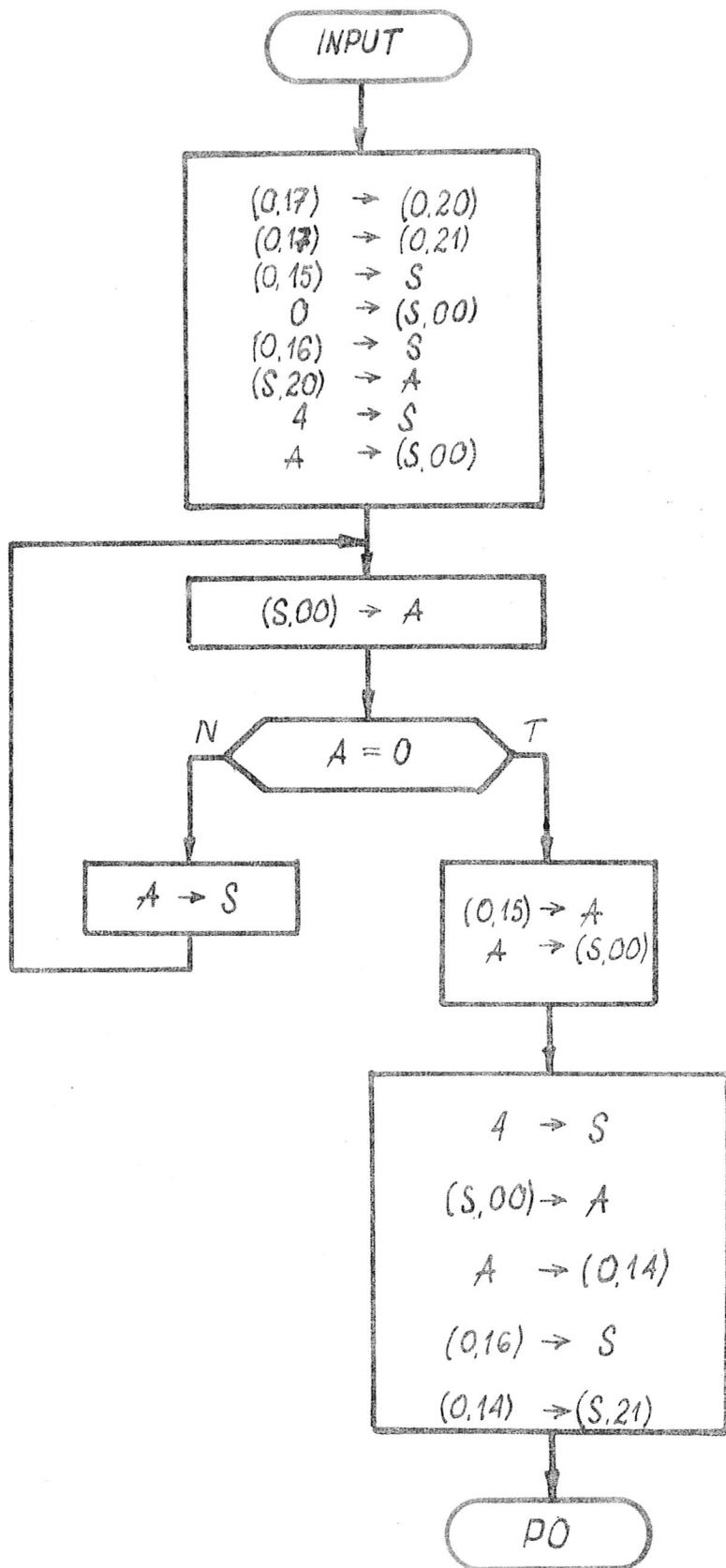
Rys.25a Schemat podprogramu REMF



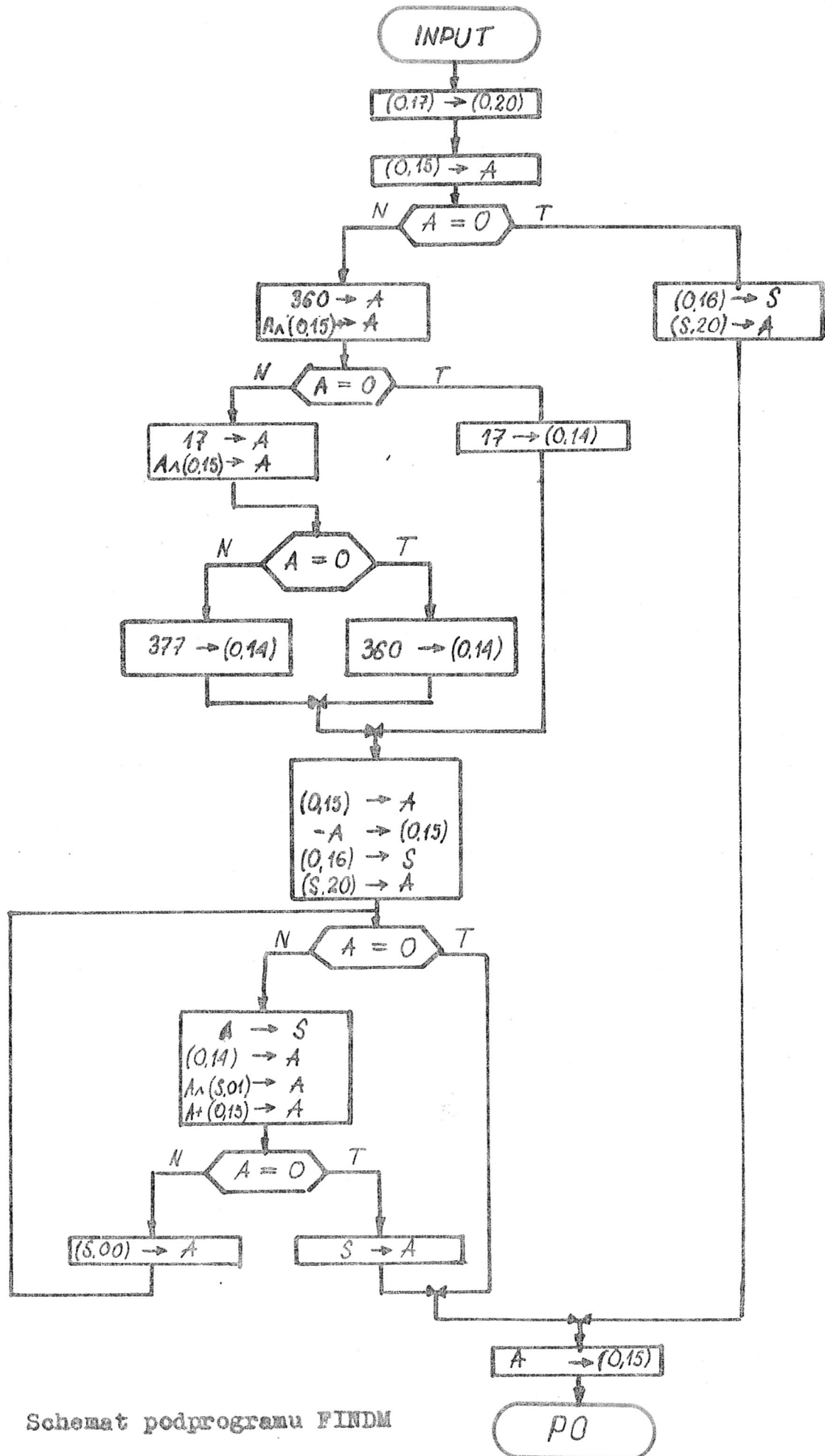
Rys.25b Schemat podprogramu SAFE2021



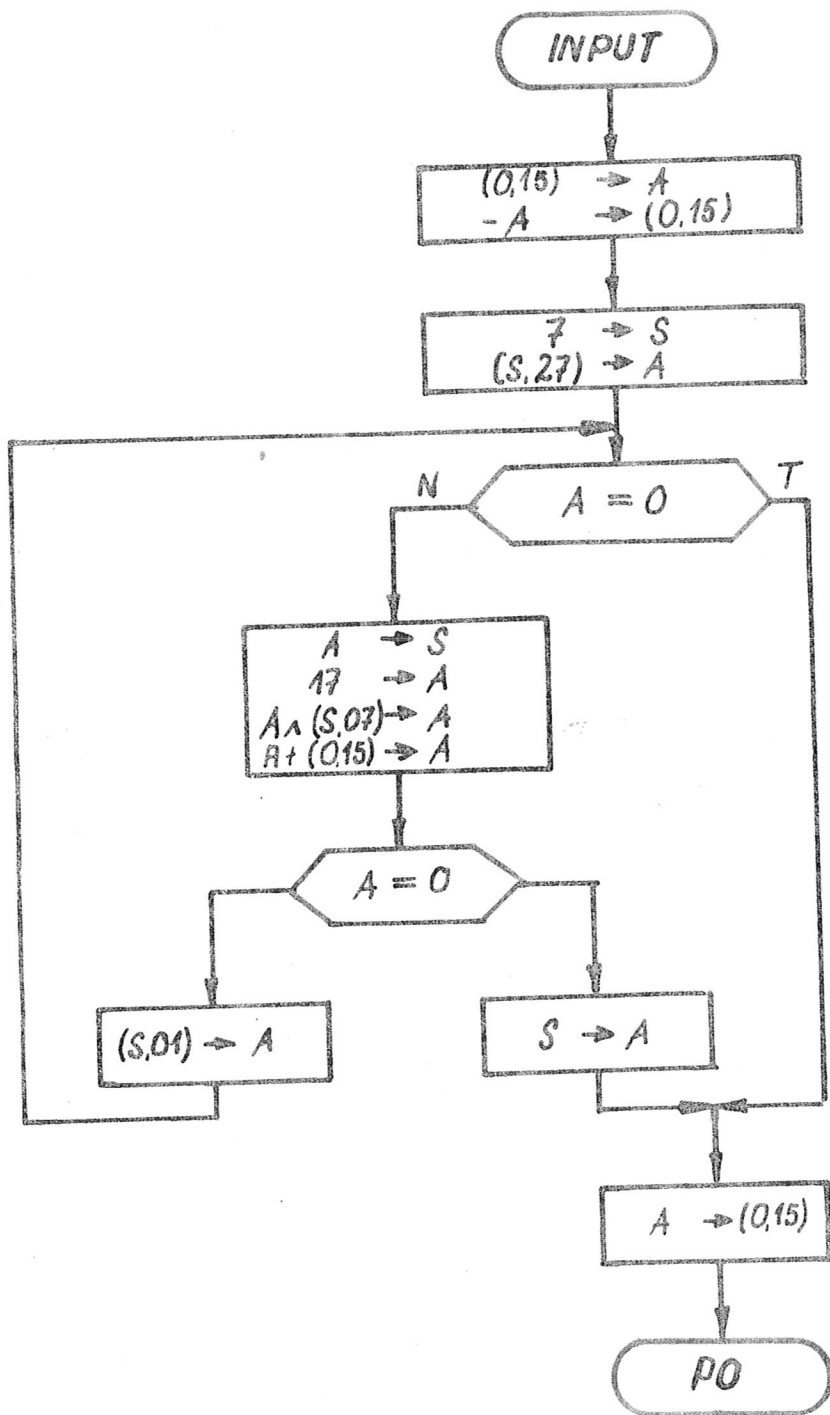
Rys.25c Schemat podprogramu REMS



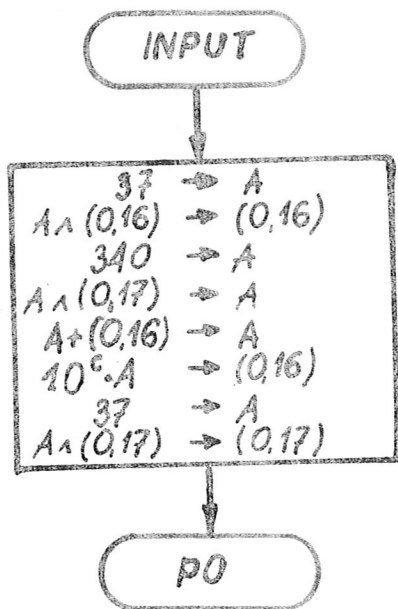
Rys.25d Schemat podprogramu JOINS



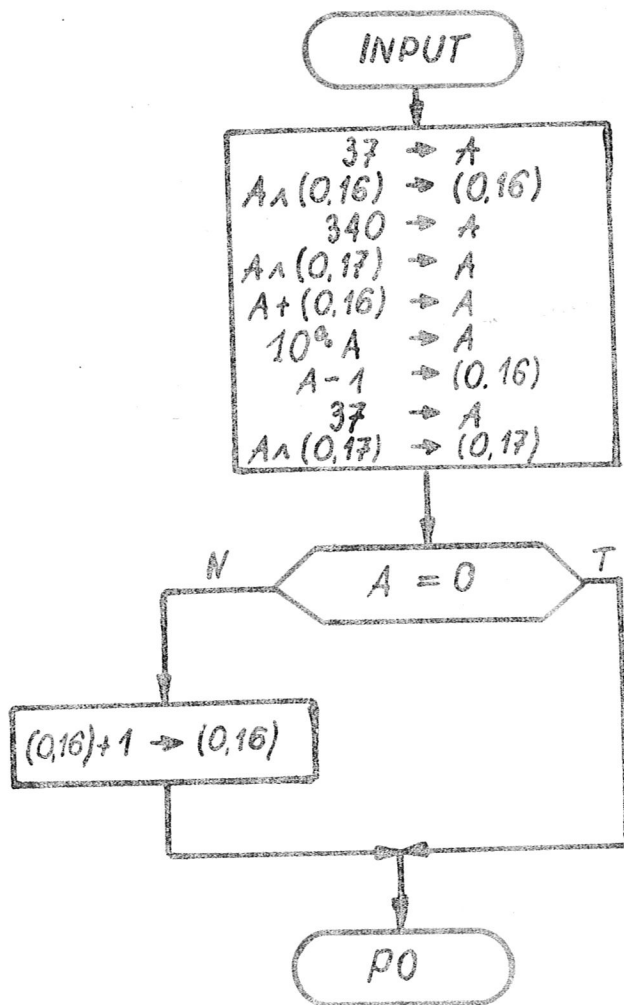
Rys.25e Schemat podprogramu FINDM



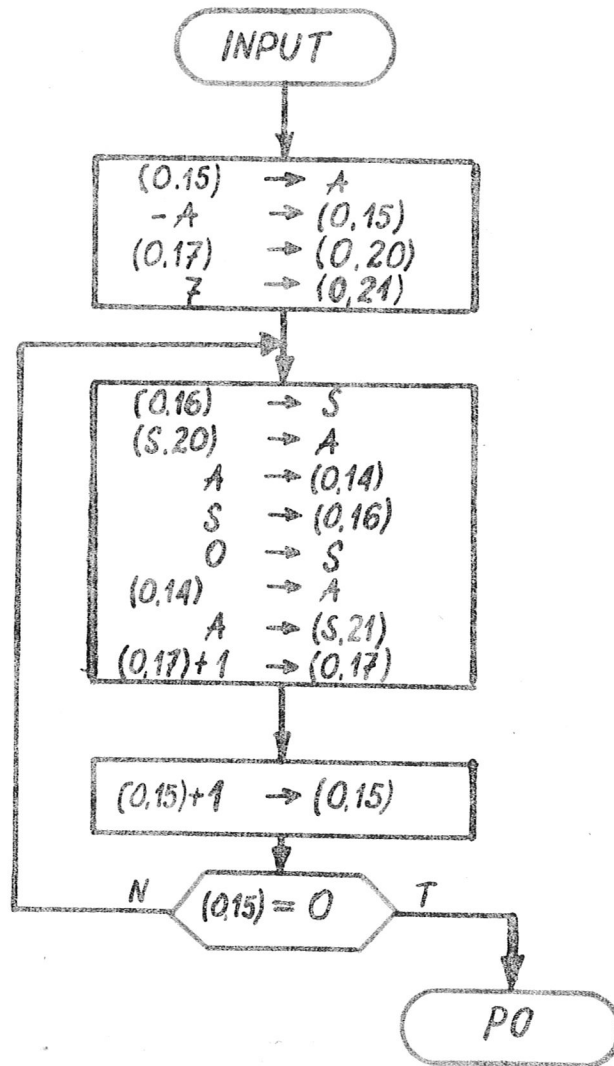
Rys.25f Schemat podprogramu FINDPR



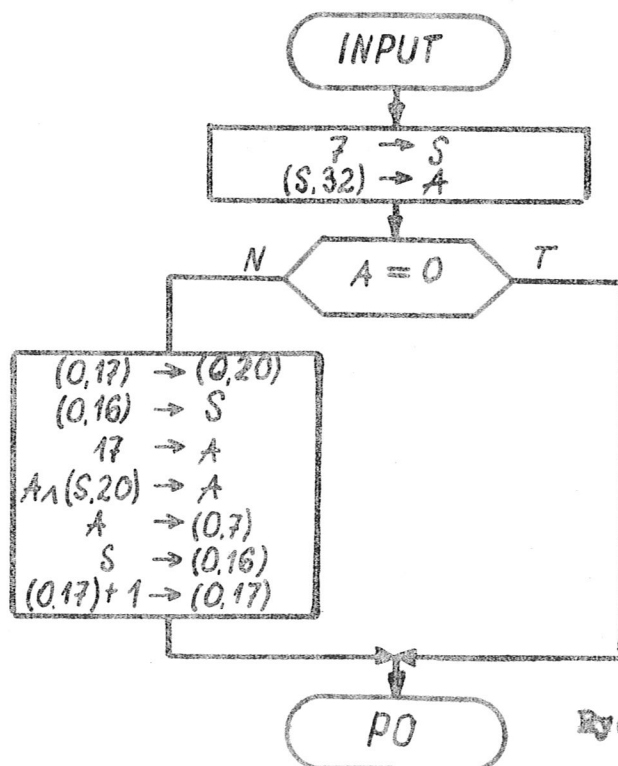
Rys.25g Schemat podprogramu CHANGEA



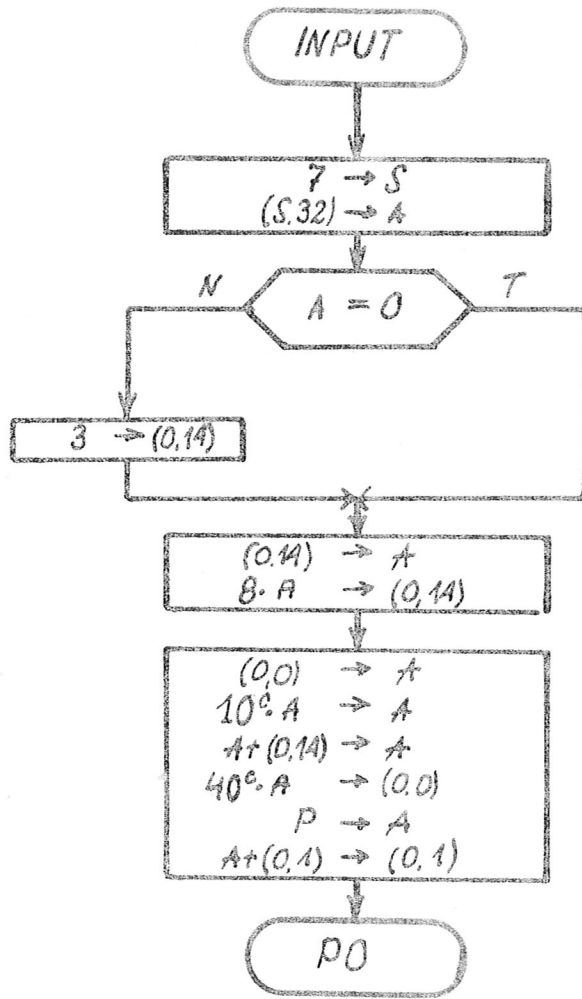
Rys.25h Schemat podprogramu CHANGEA1



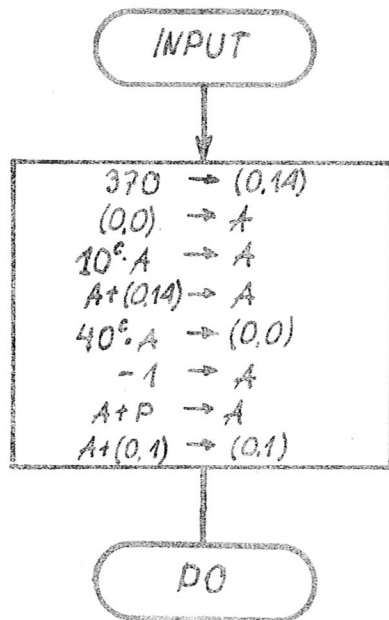
Rys.25i Schemat podprogramu FETCHB



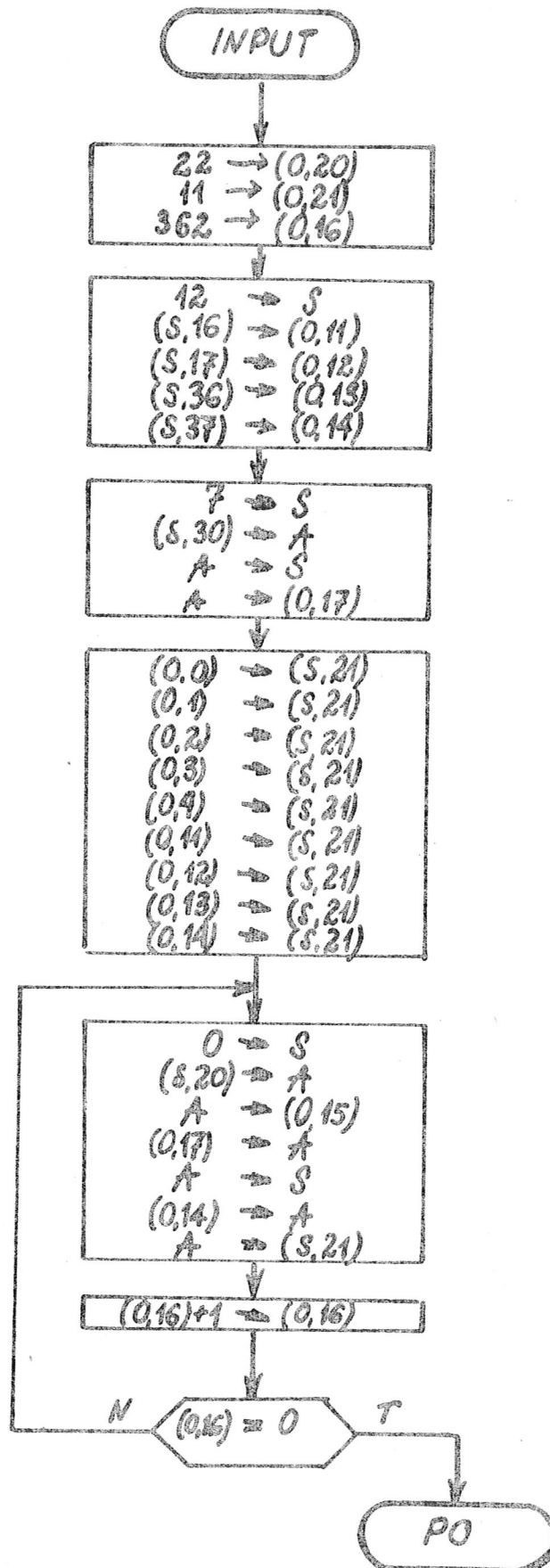
Rys.25j Schemat podprogramu FETCH1



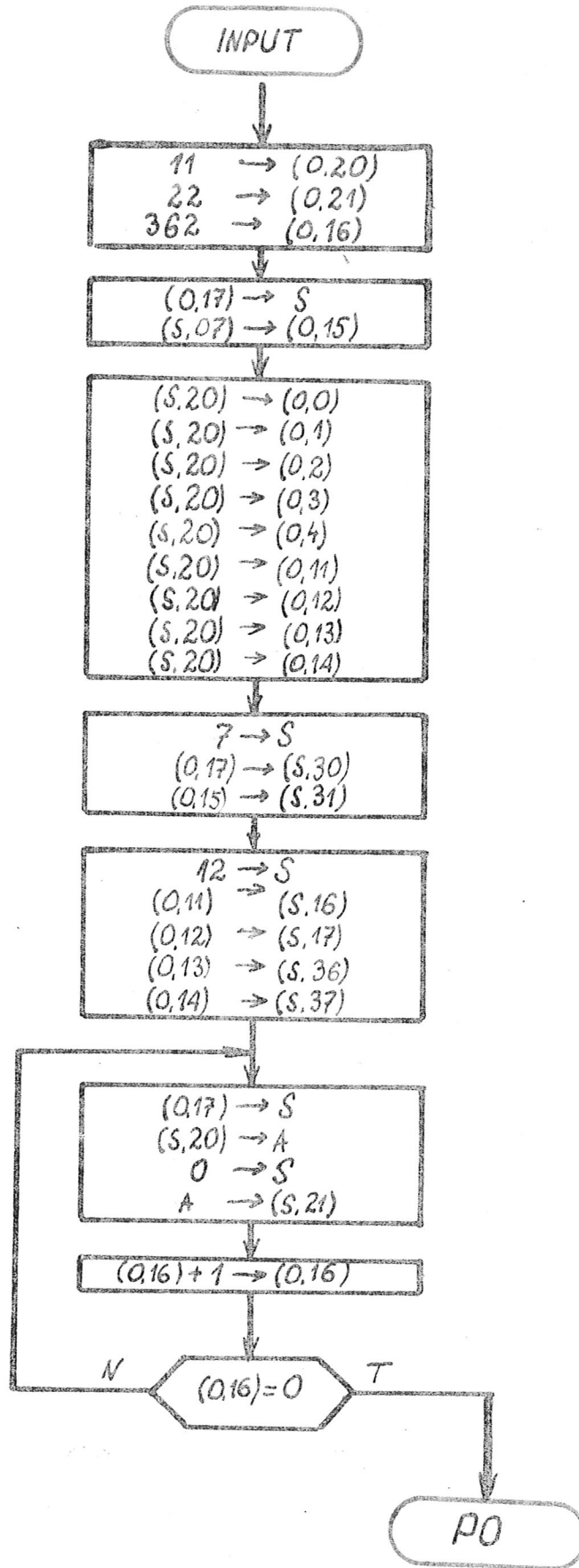
Rys.25k Schemat podprogramu LRINCR



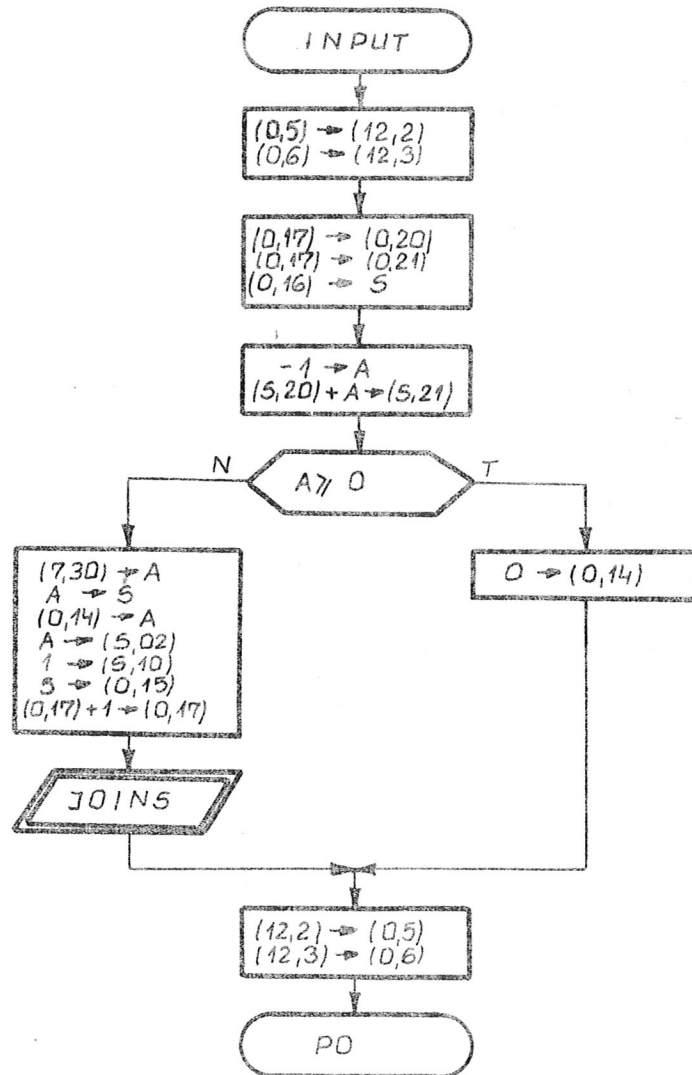
Rys.25l Schemat podprogramu LRDECR1



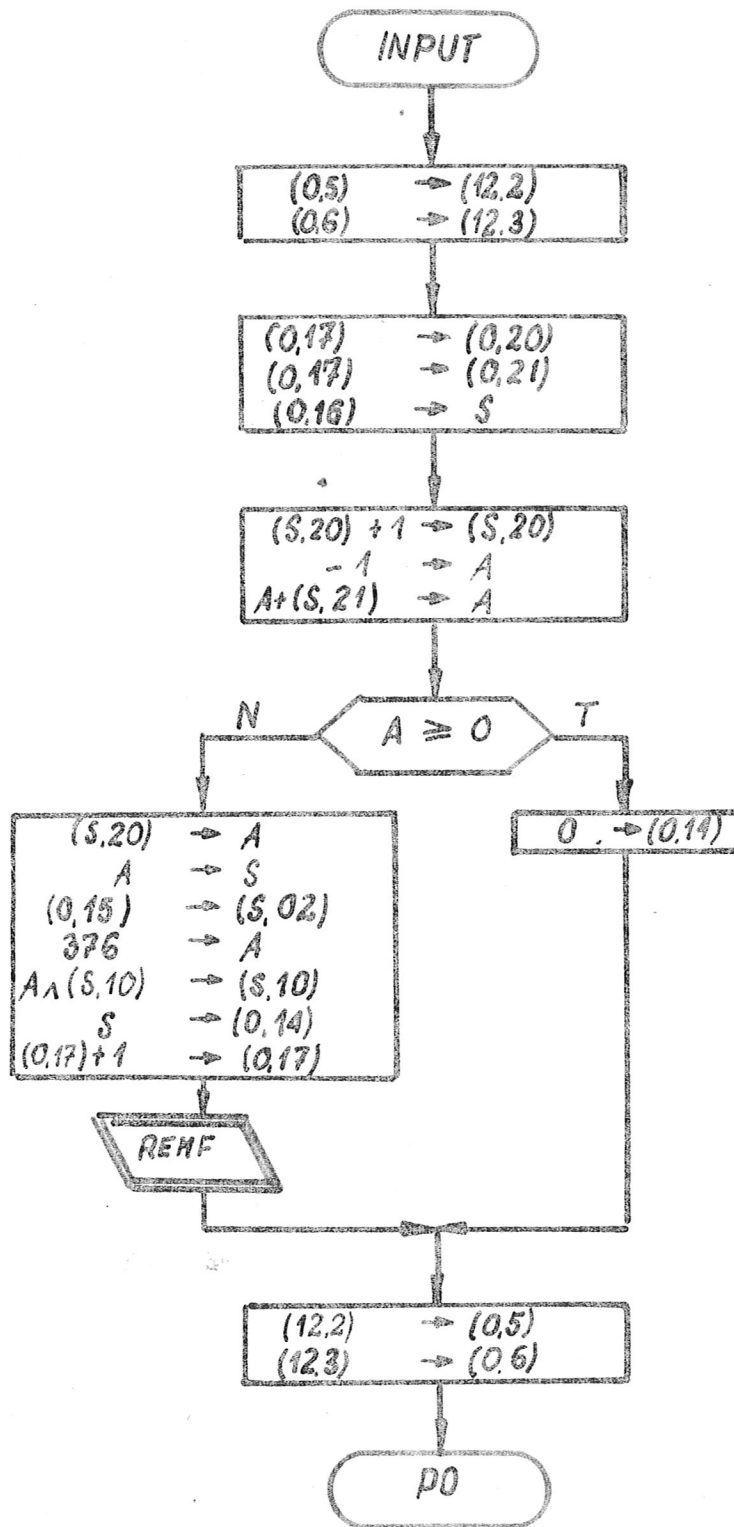
Rys.25m Schemat podprogramu REMPR



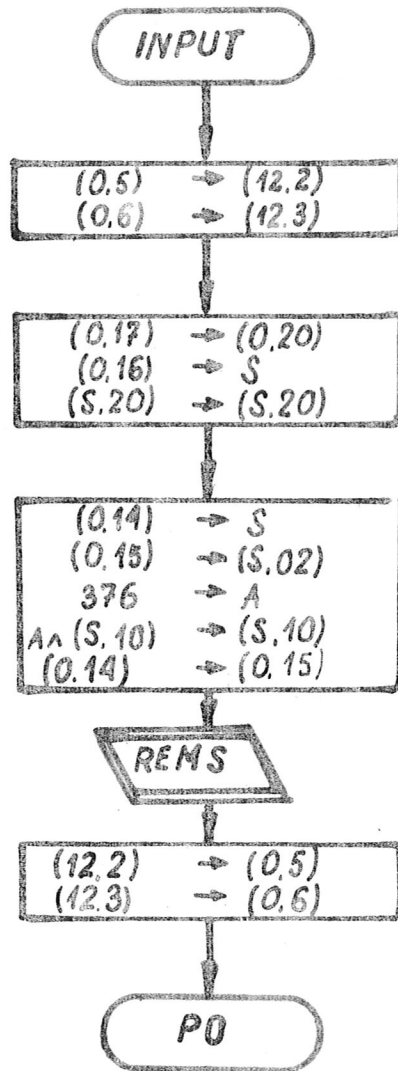
Rys.25n Schemat podprogramu FETCHPR



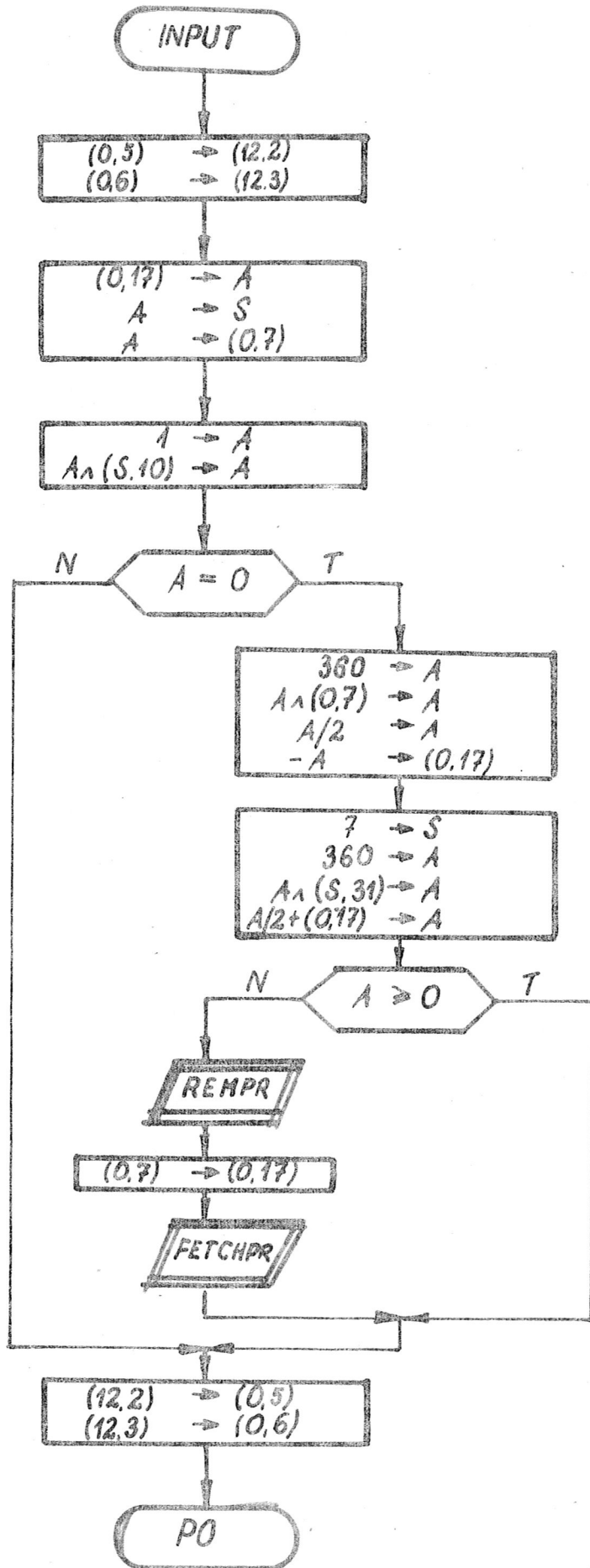
Rys.26a Schemat podprogramu POP



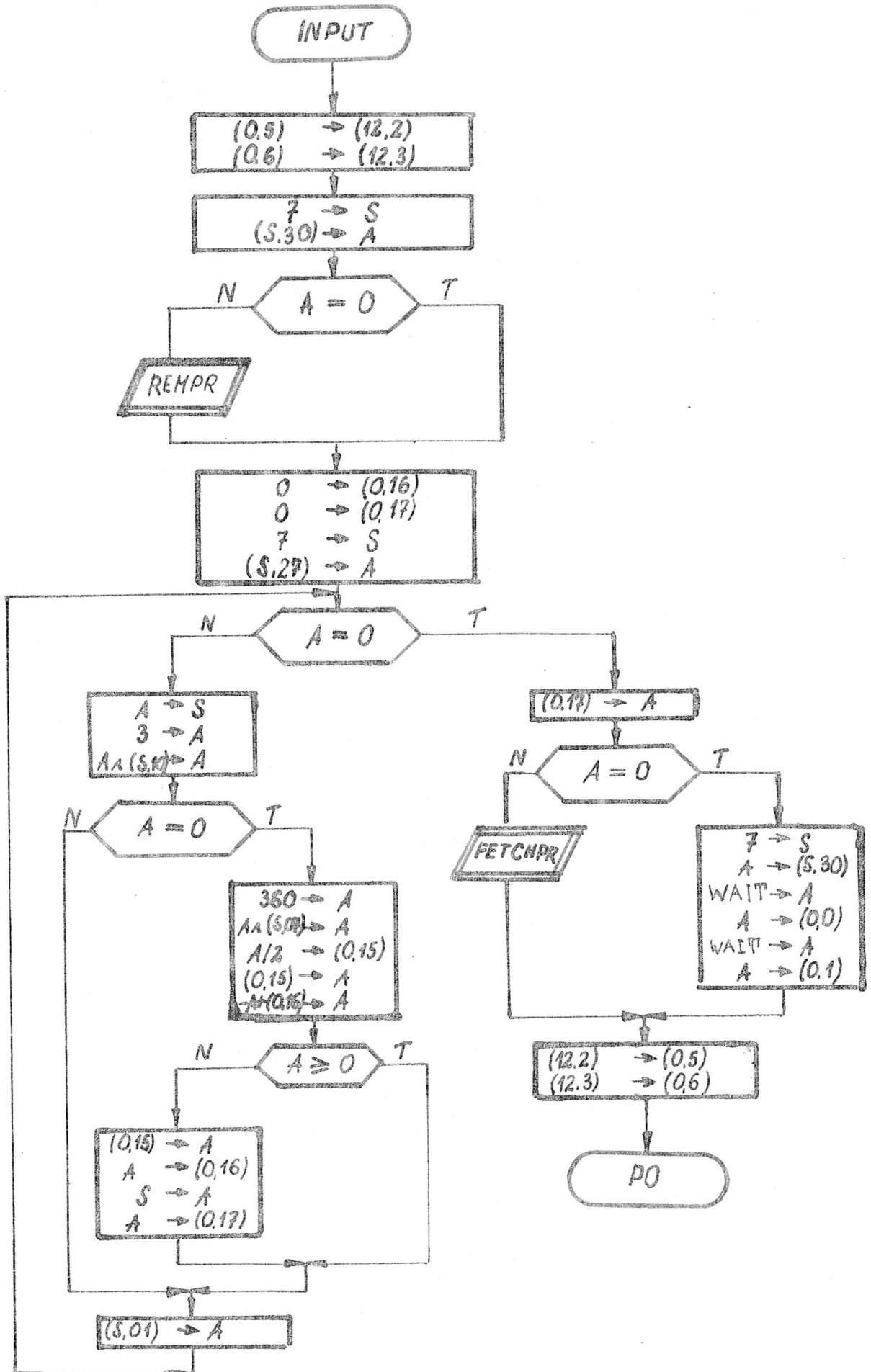
Rys.26b Schemat podprogramu VOP



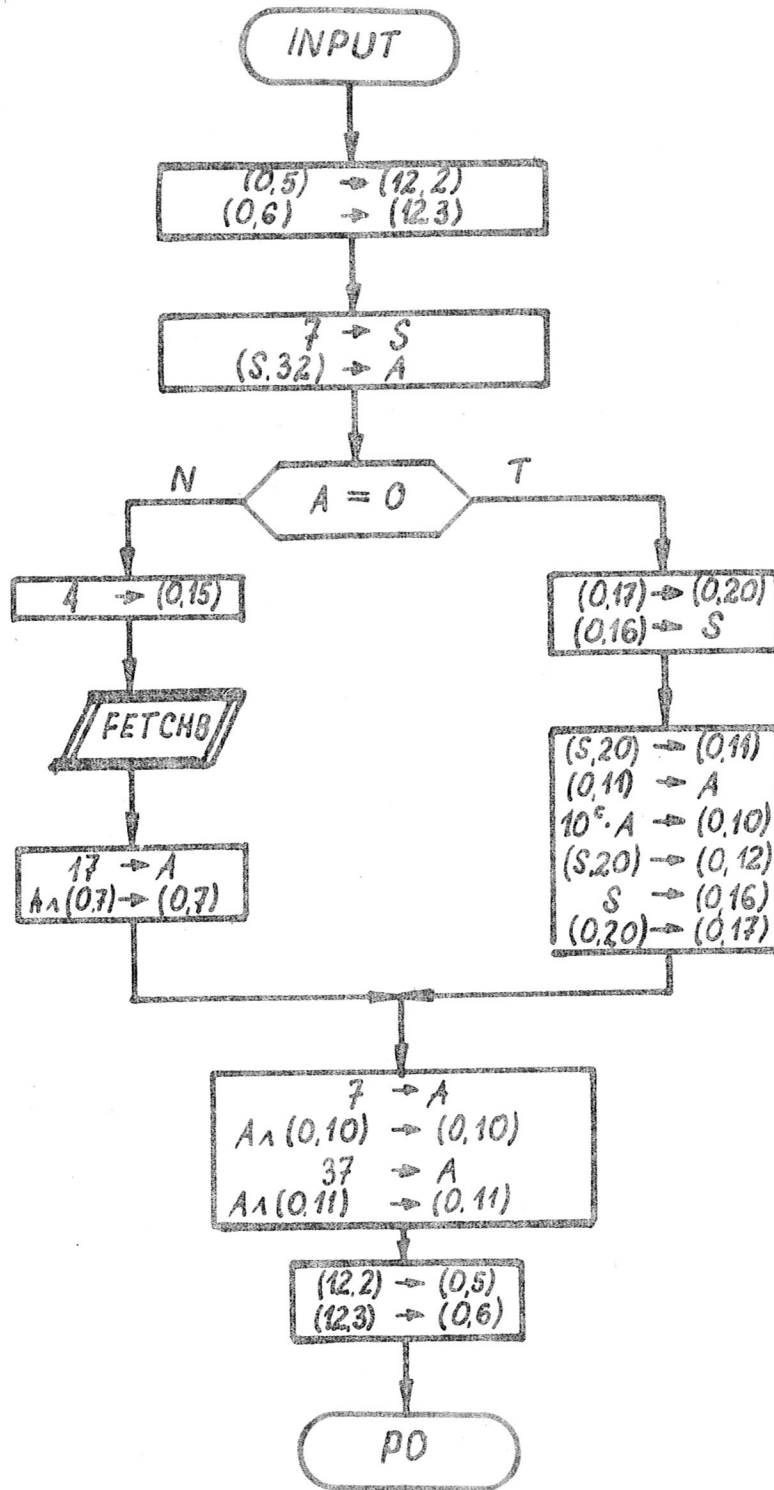
rys.26c Schemat podprogramu VOPS



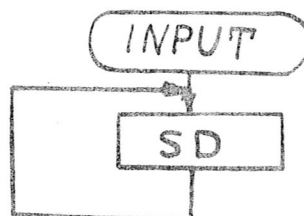
Rys.26d Schemat podprogramu SCHEDS



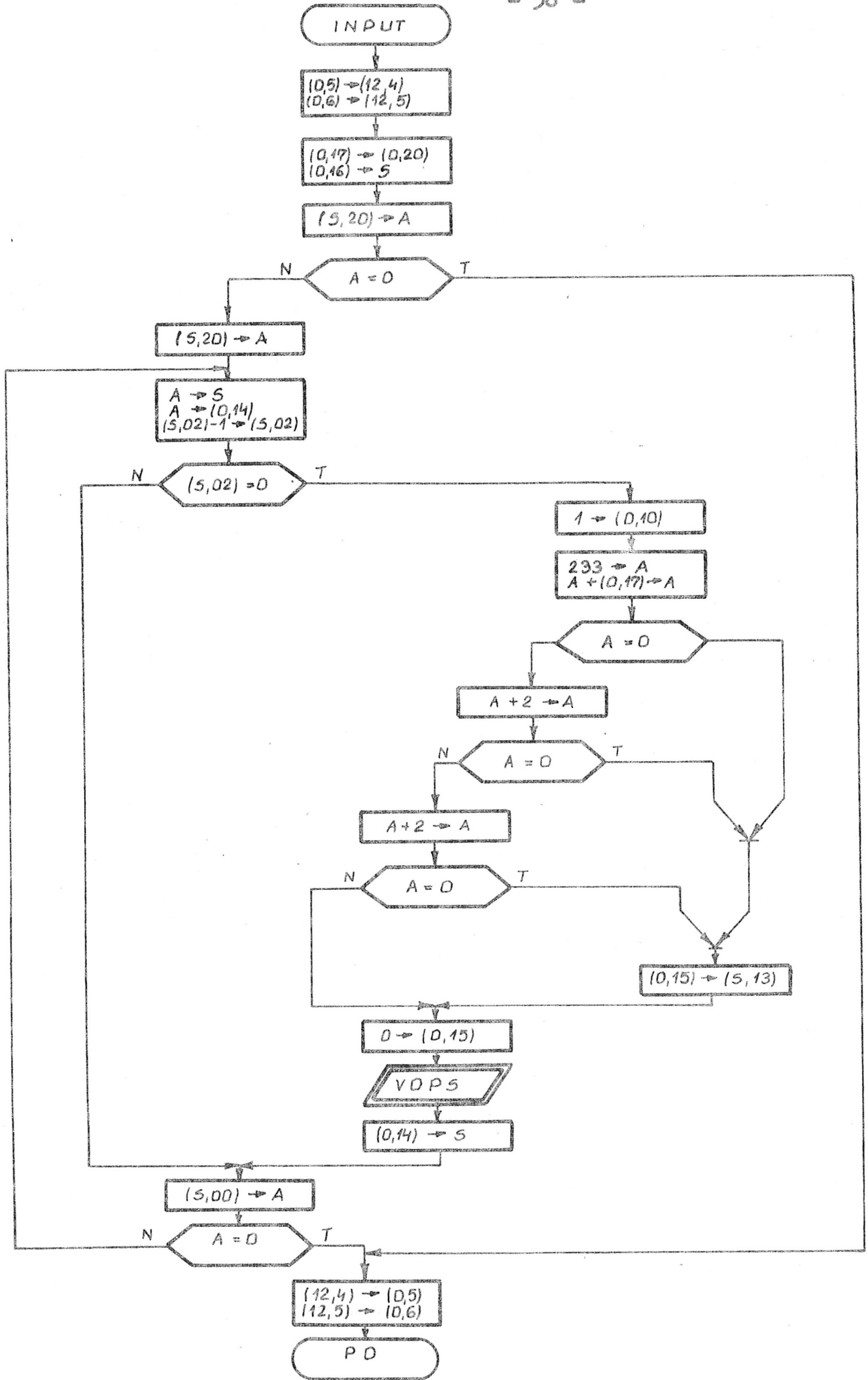
Rys.26e Schemat podprogramu SCHED



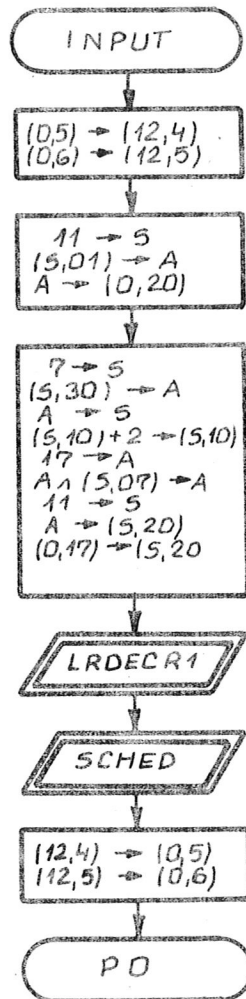
Rys.26f Schemat podprogramu FETCH3



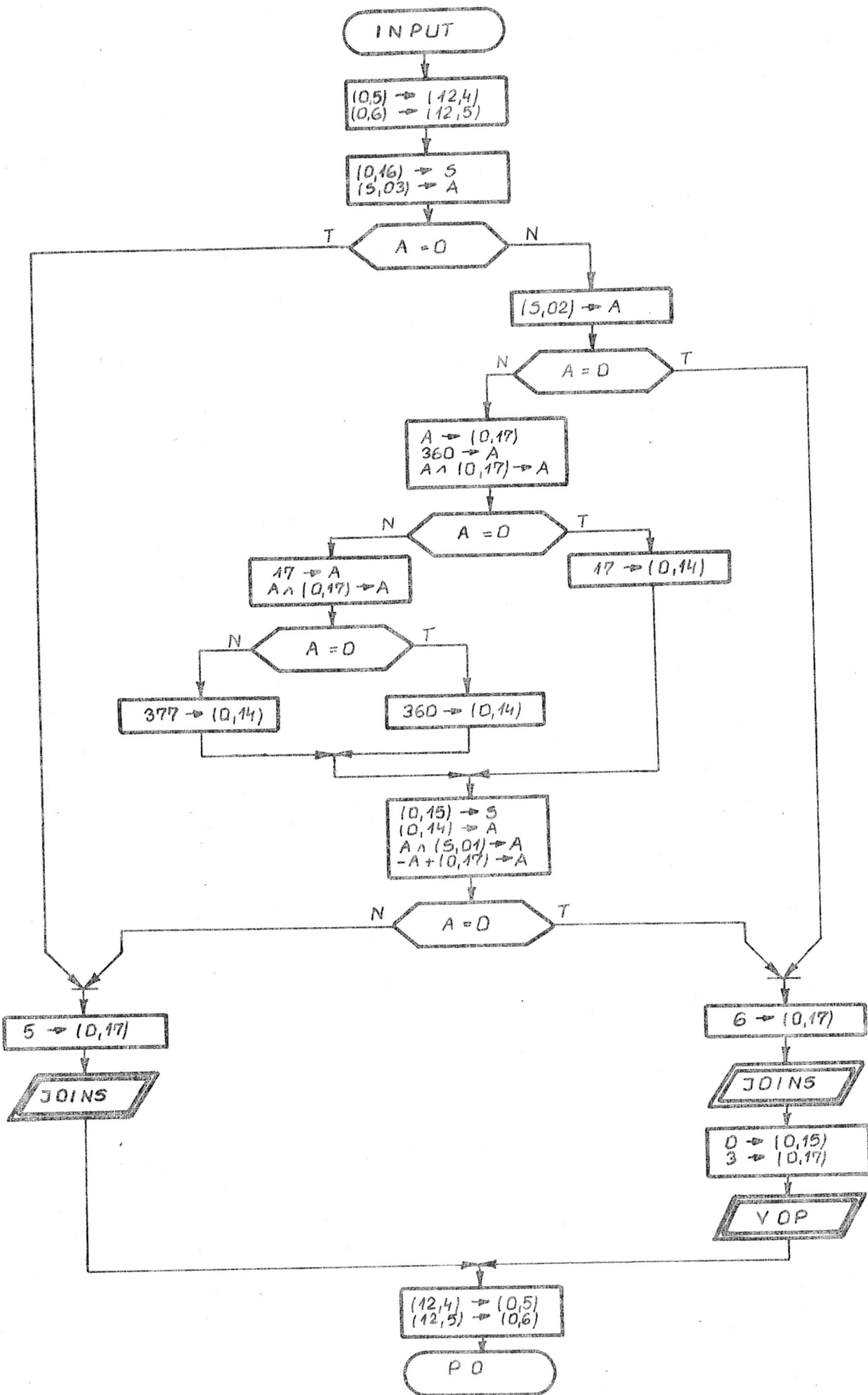
Rys.26g Schemat podprogramu WAIT



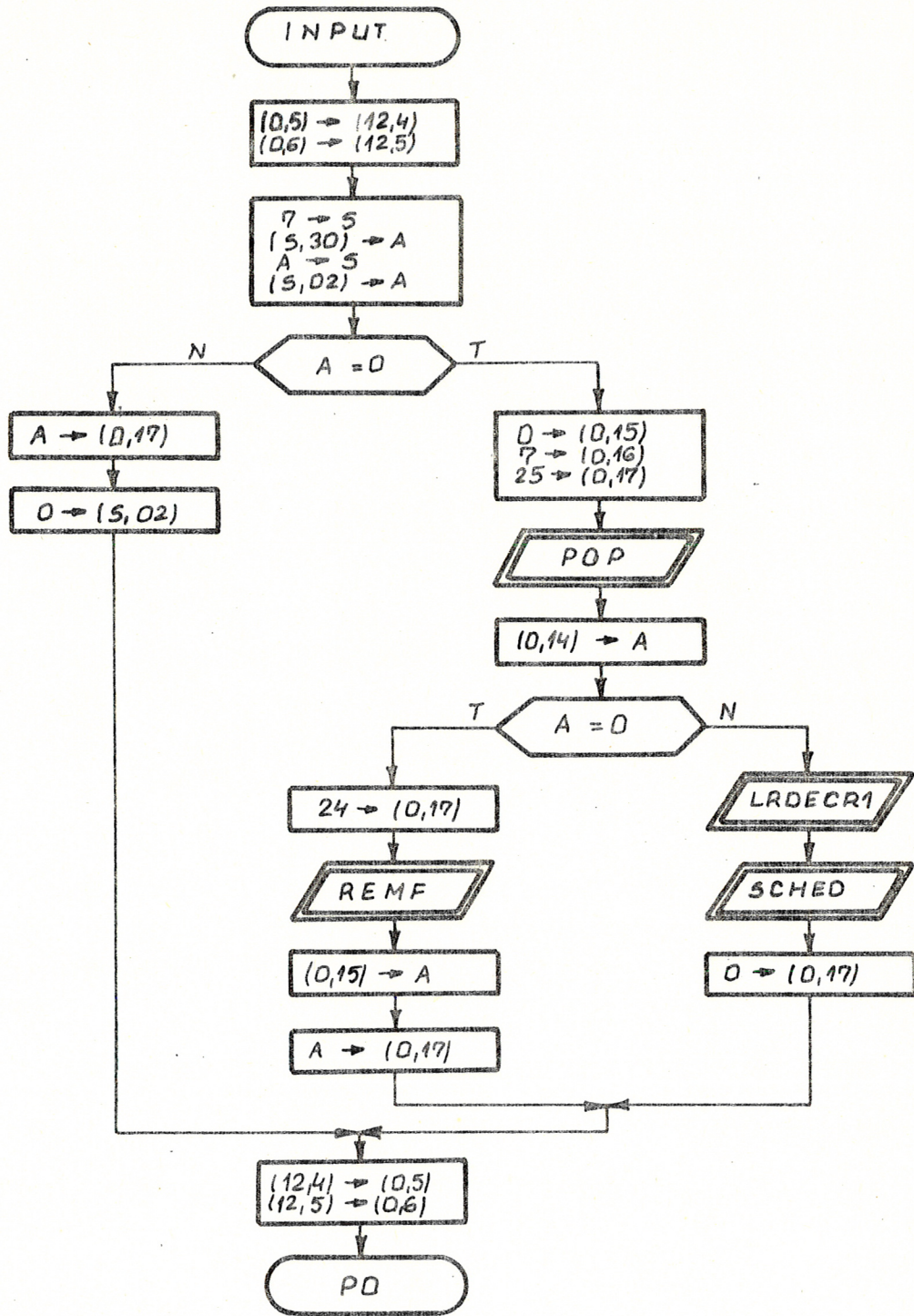
Rys.27a Schemat podprogramu CHECKQ



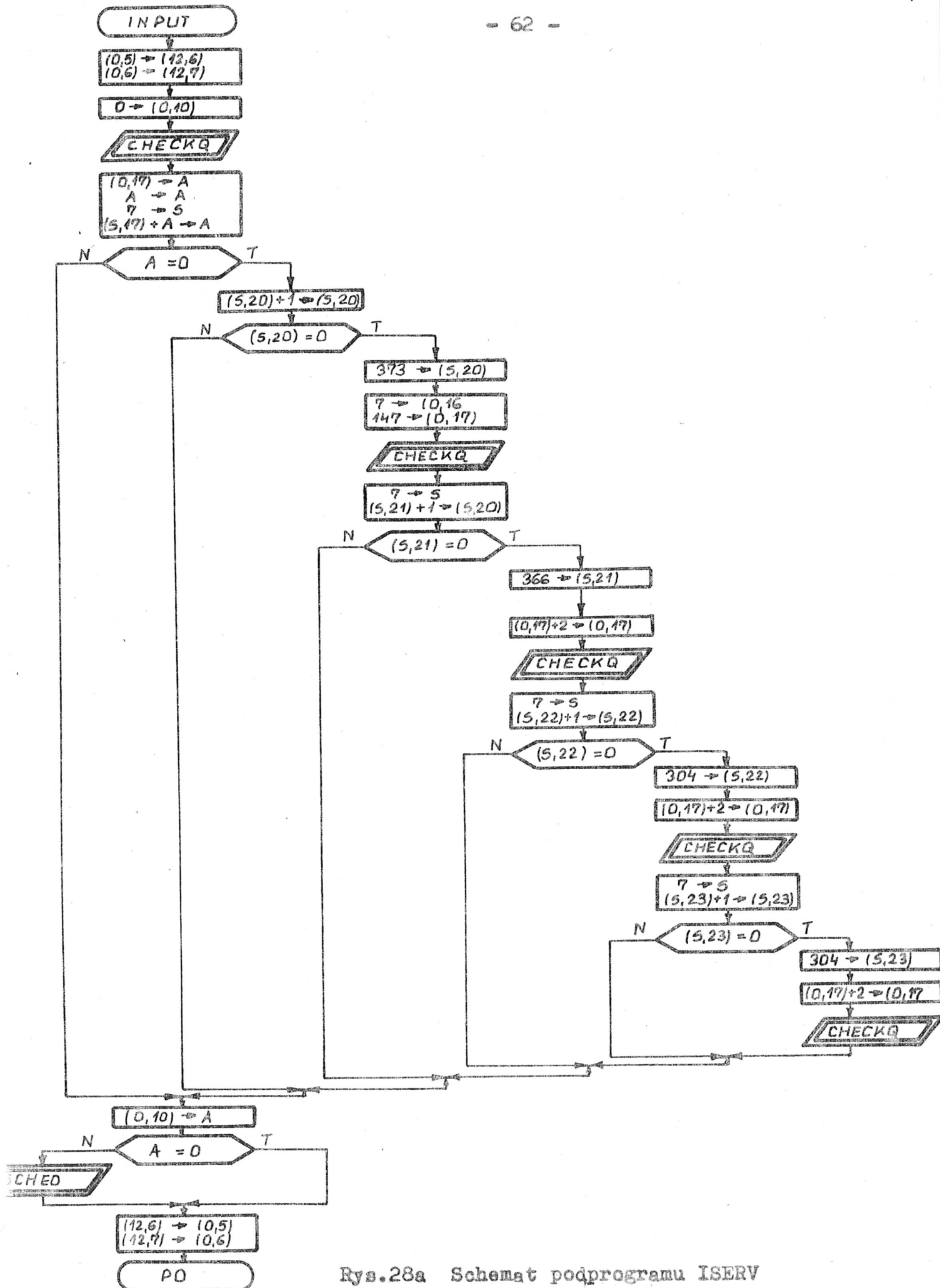
Rys.27b Schemat podprogramu ERR



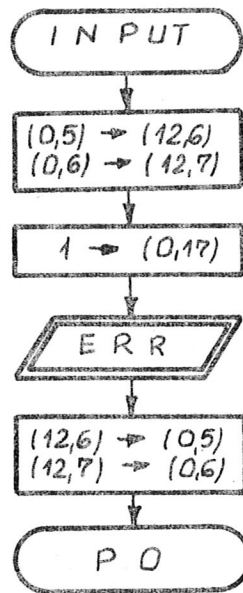
Rys.27c Schemat podprogramu SENDM



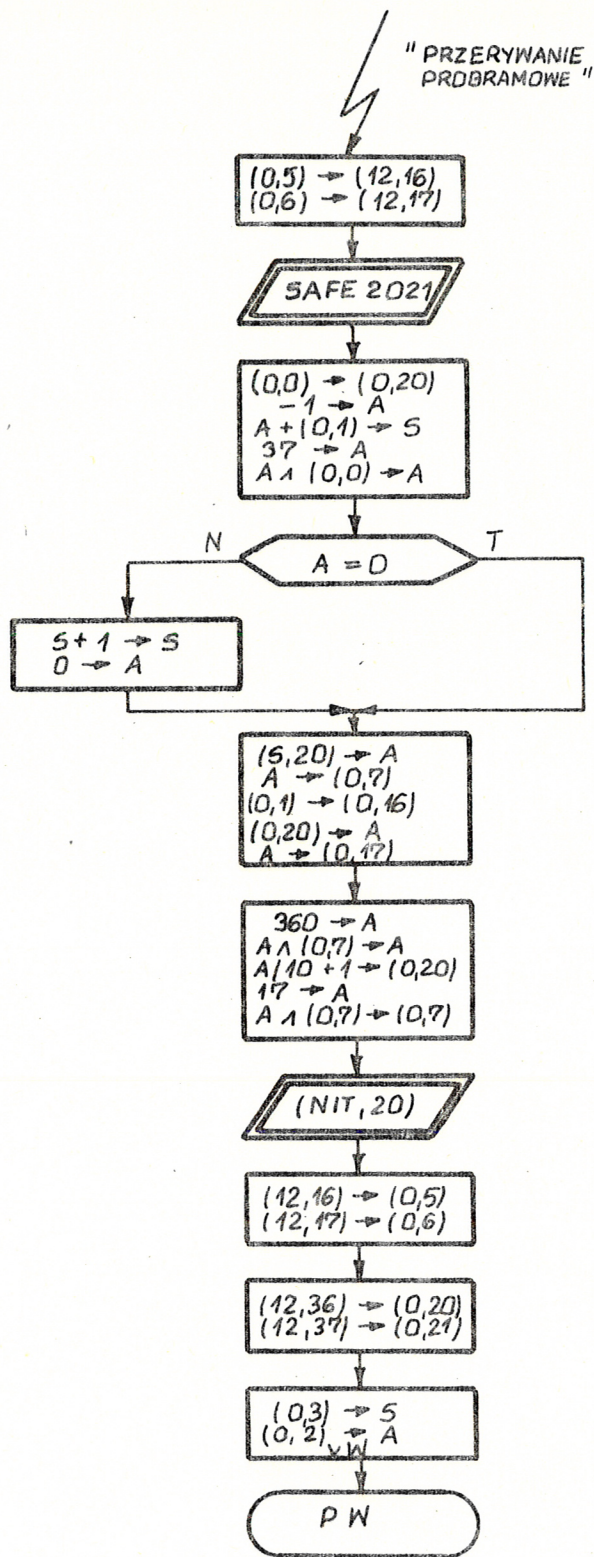
Rys.27d Schemat podprogramu GETP



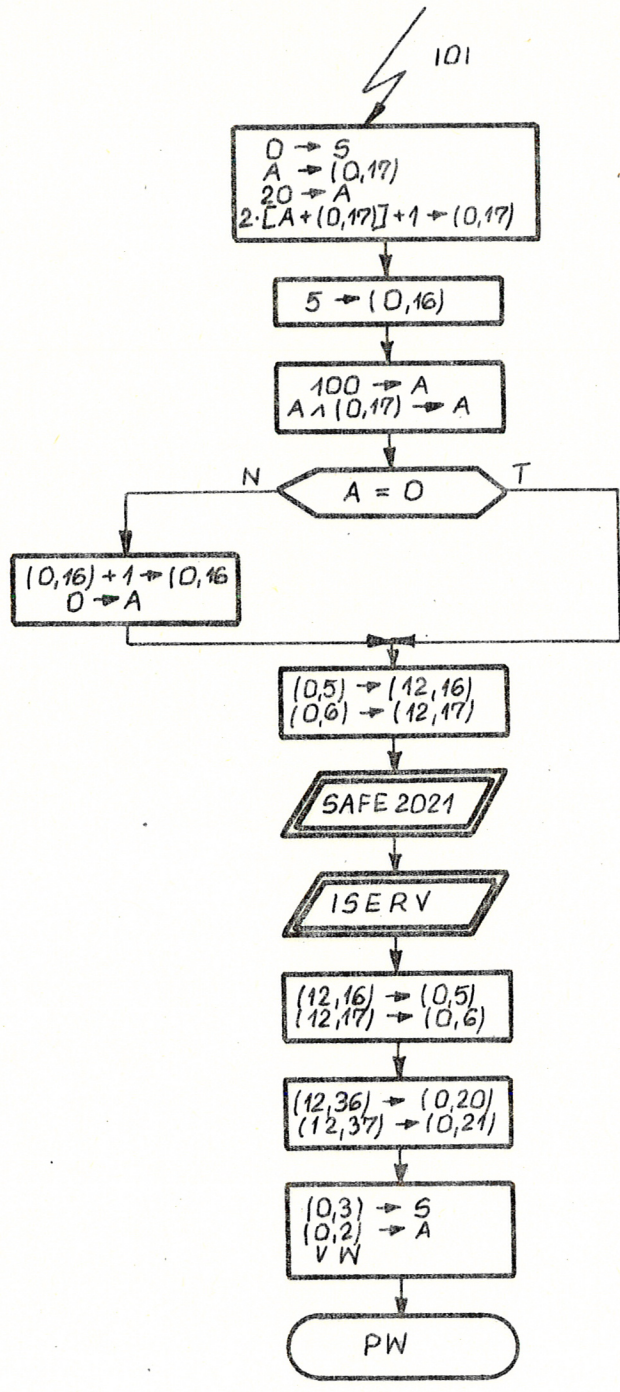
Rys.28a Schemat podprogramu ISERV



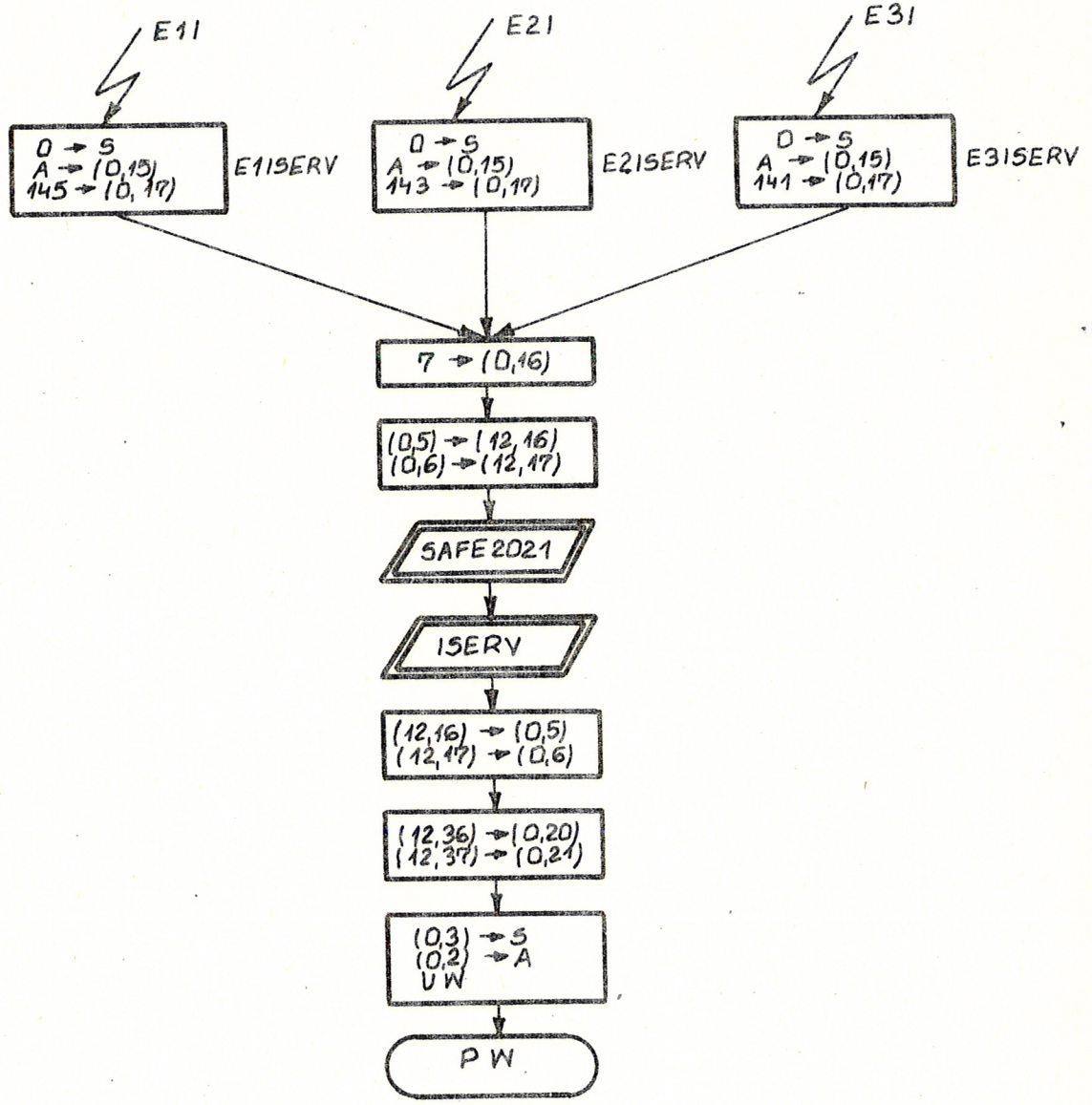
Rys.28b Schemat podprogramu ERRVI



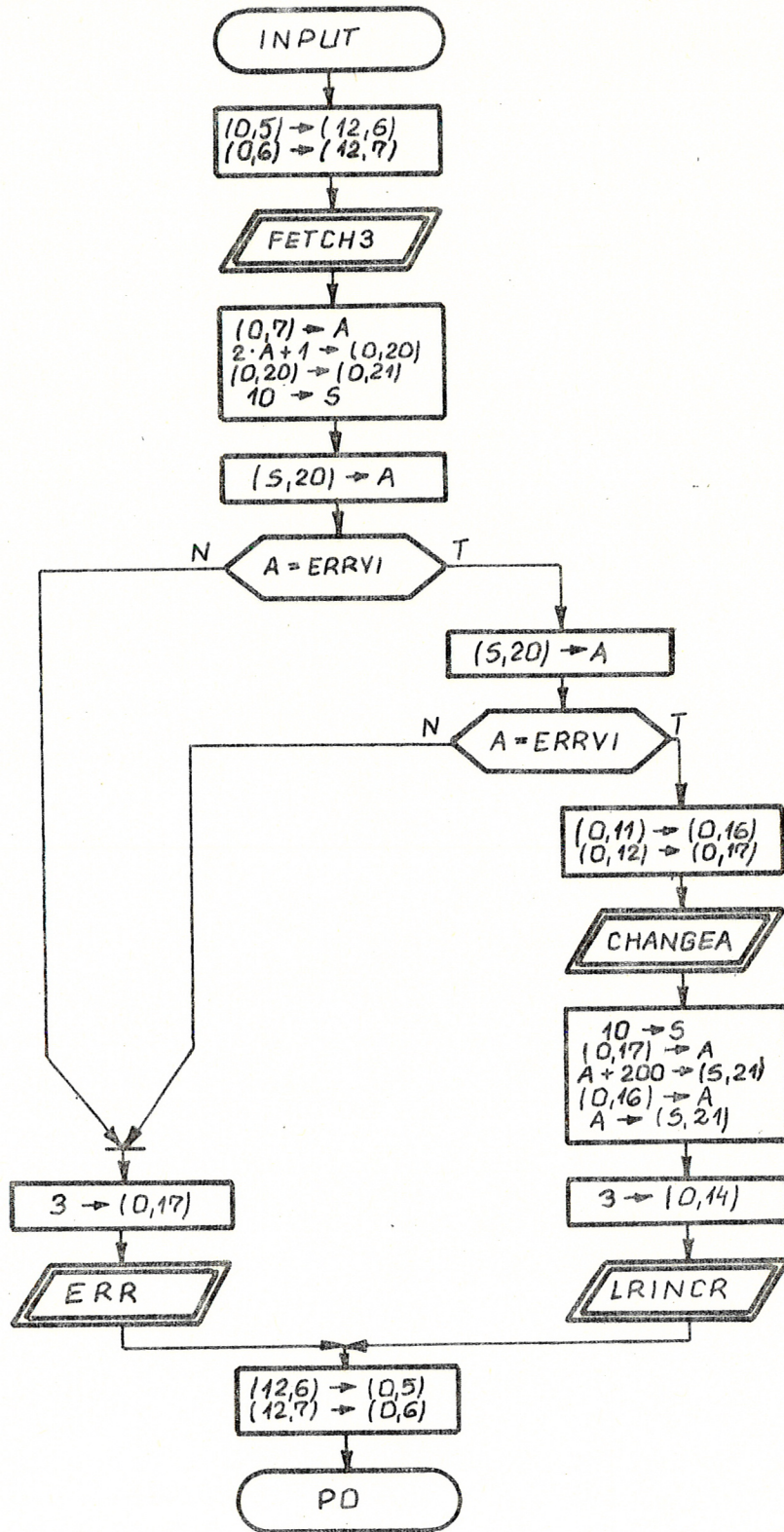
Rys.29a Schemat podprogramu PRISERV



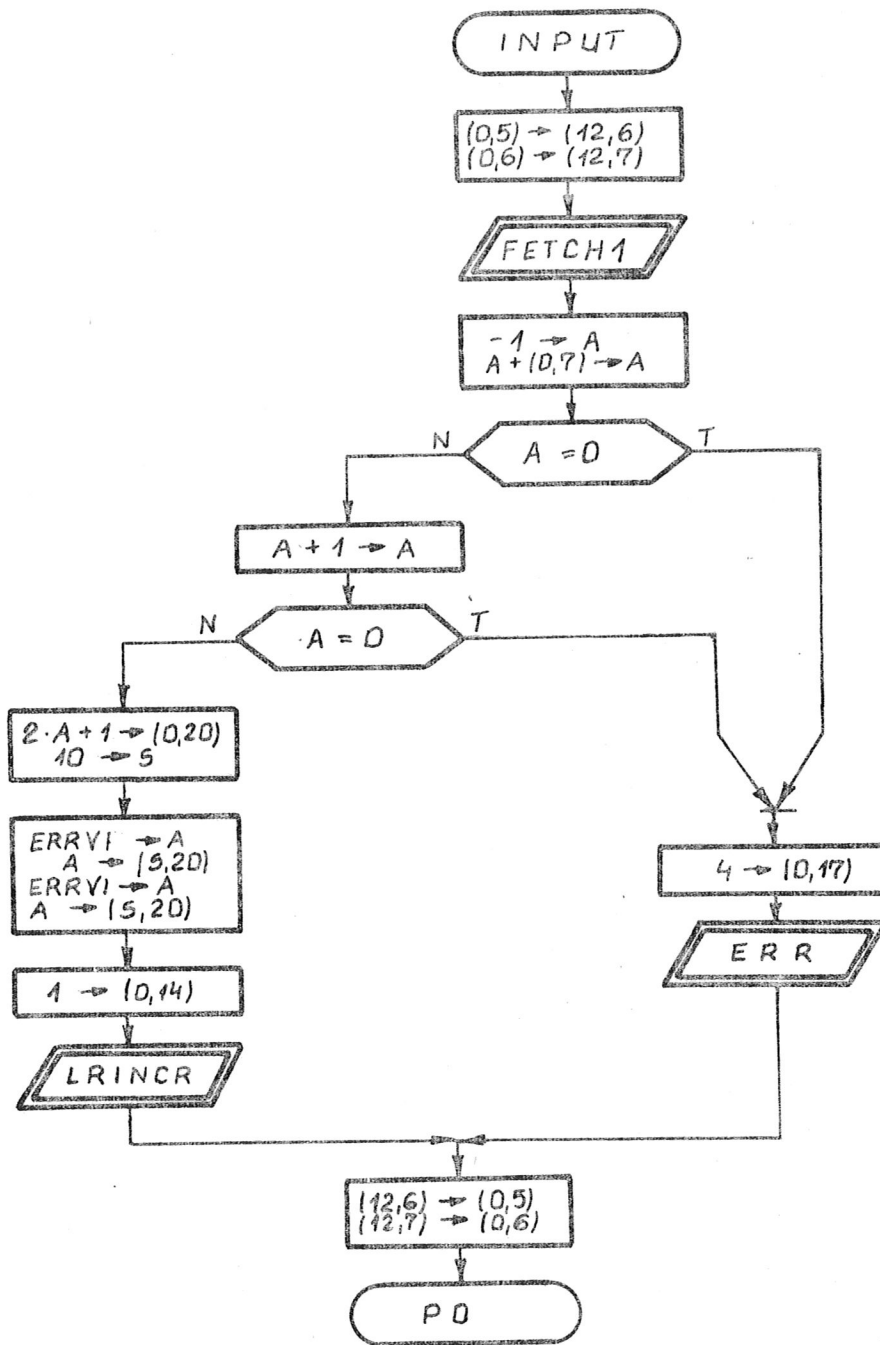
Rys.29b Schemat podprogramu IOISERV



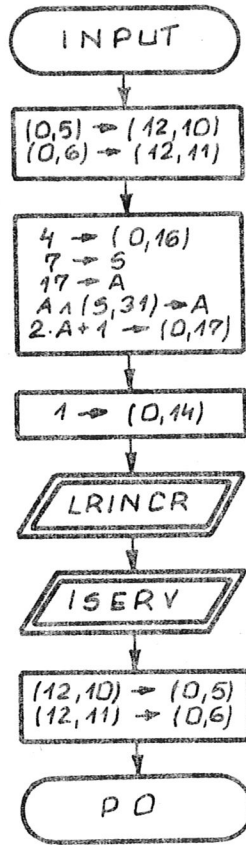
Rys.29c Schemat podprogramu EISERV



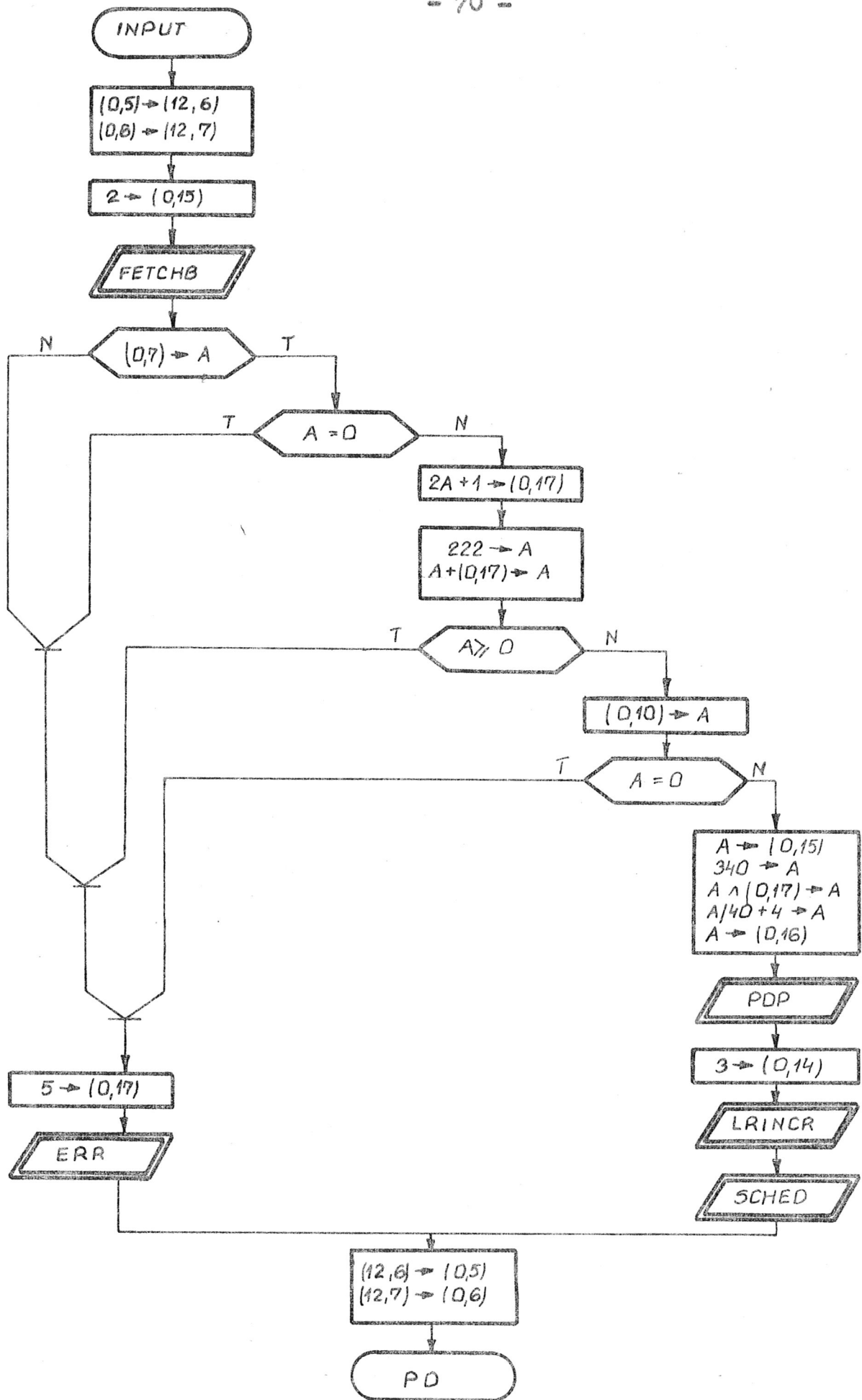
Rys.30a Schemat podprogramu CVIR



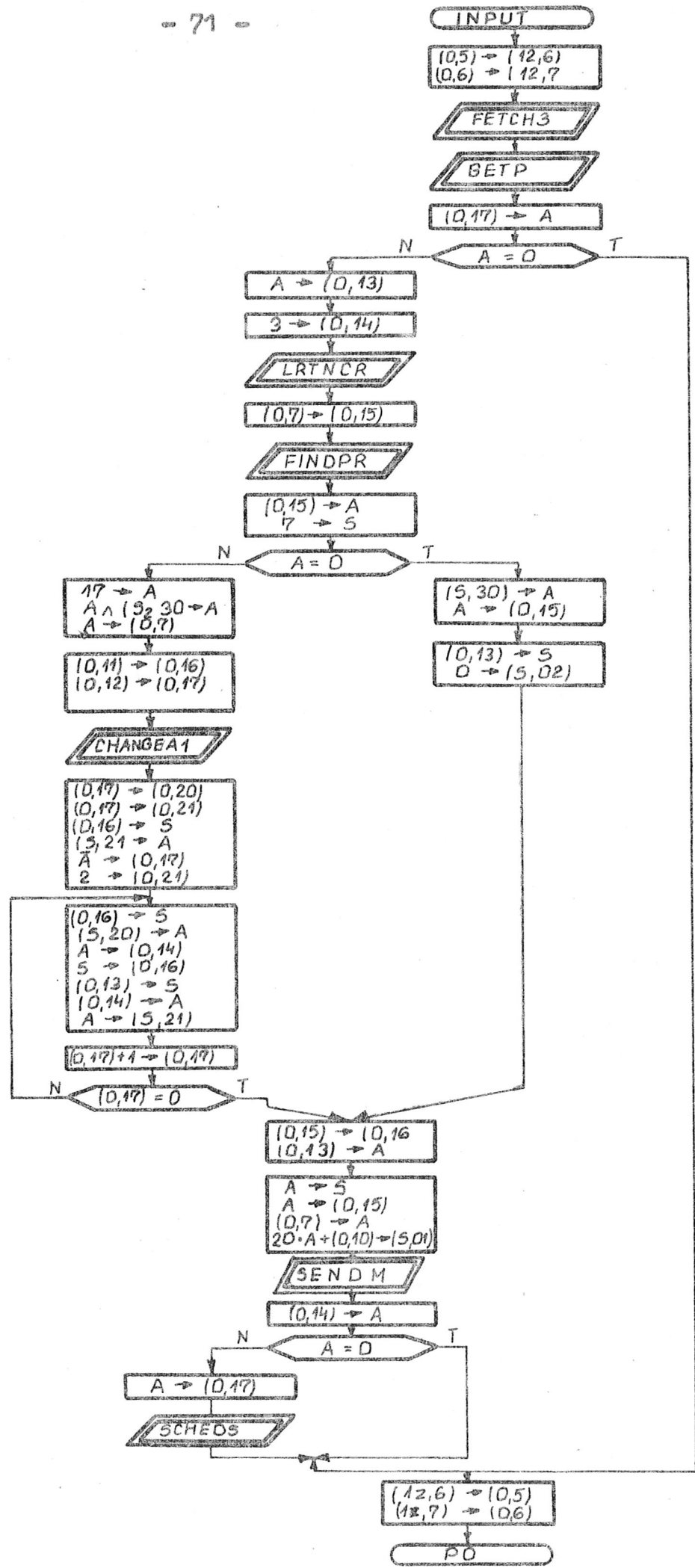
Rys.30b Schemat podprogramu DVIR



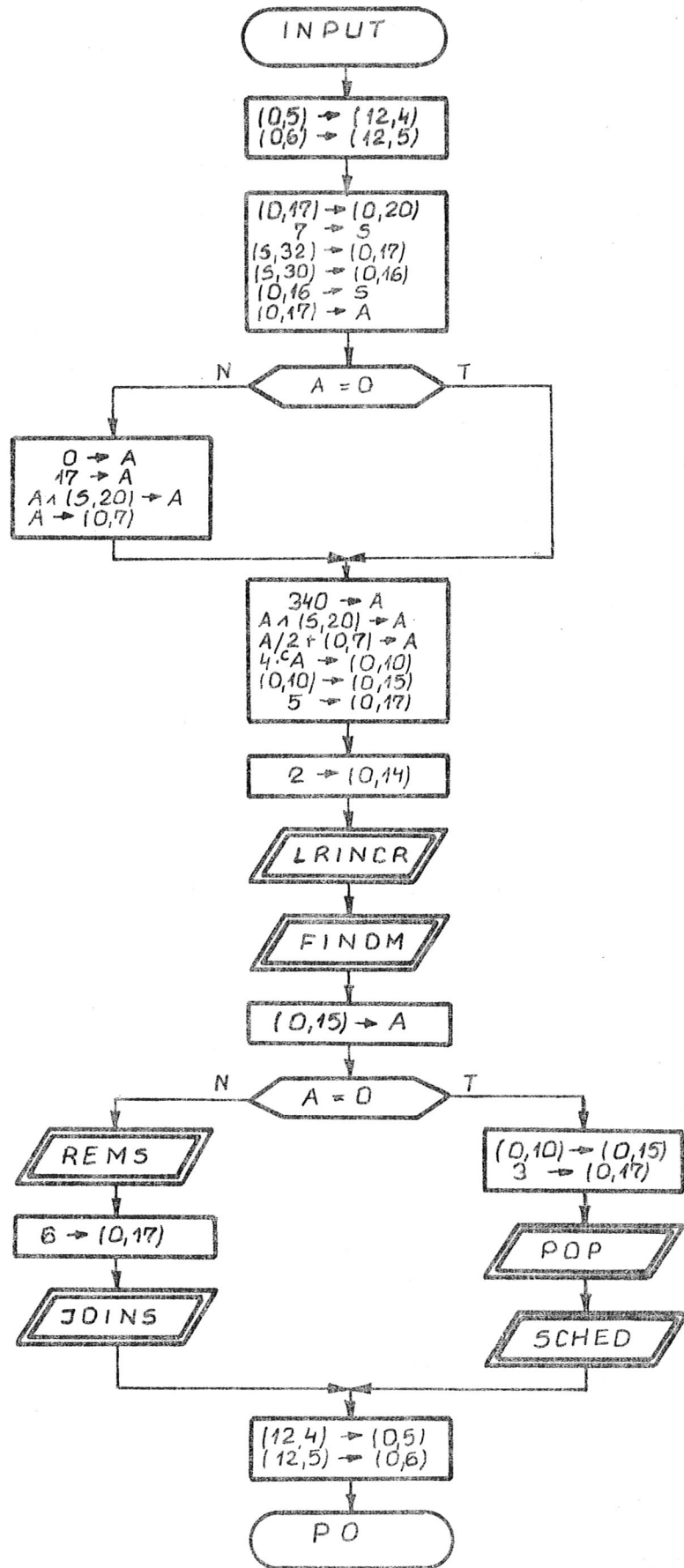
Rys.30c Schemat podprogramu GINR



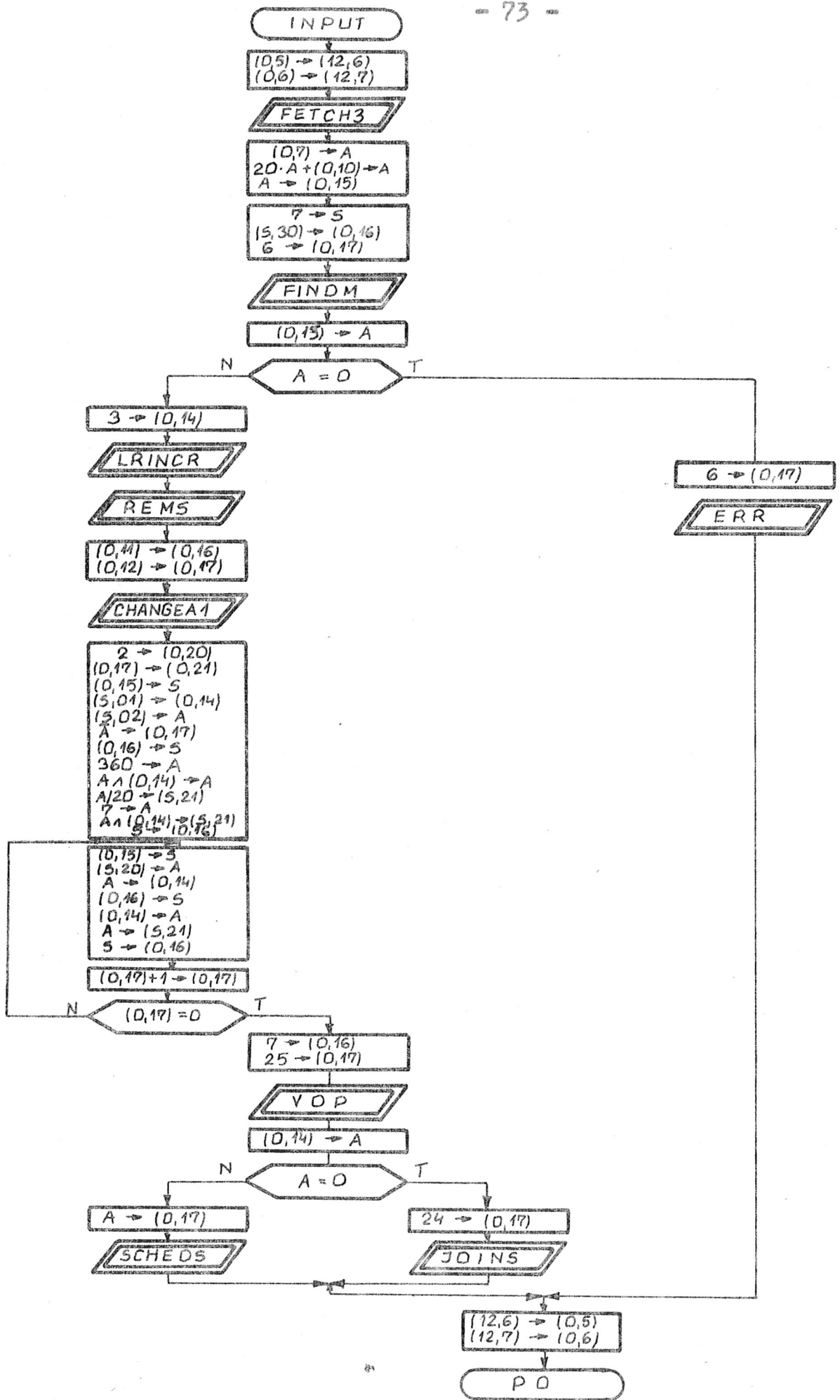
Rys. 30d Schemat podprogramu WINR



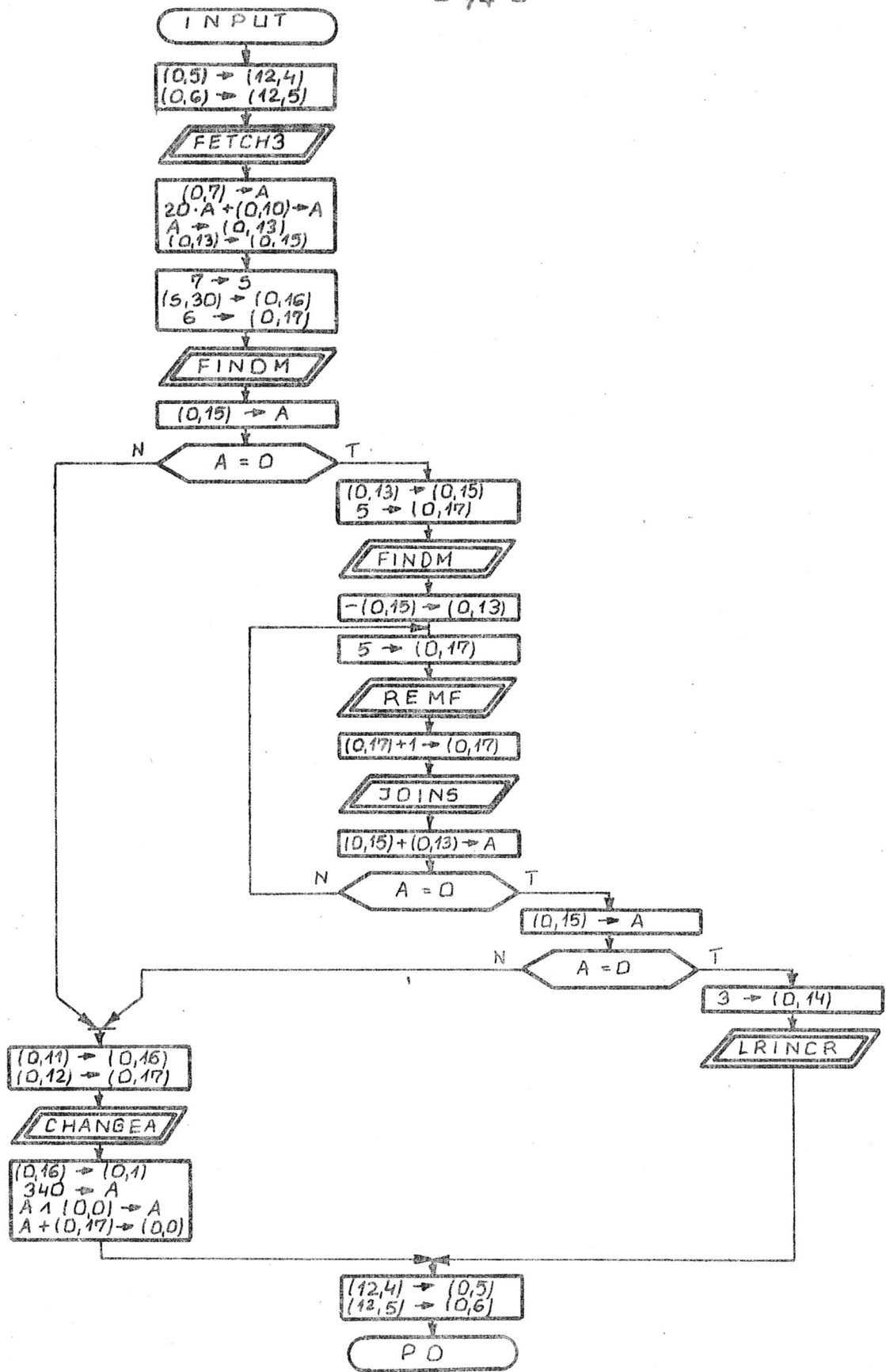
Rys.30e Schemat podprogramu SMER



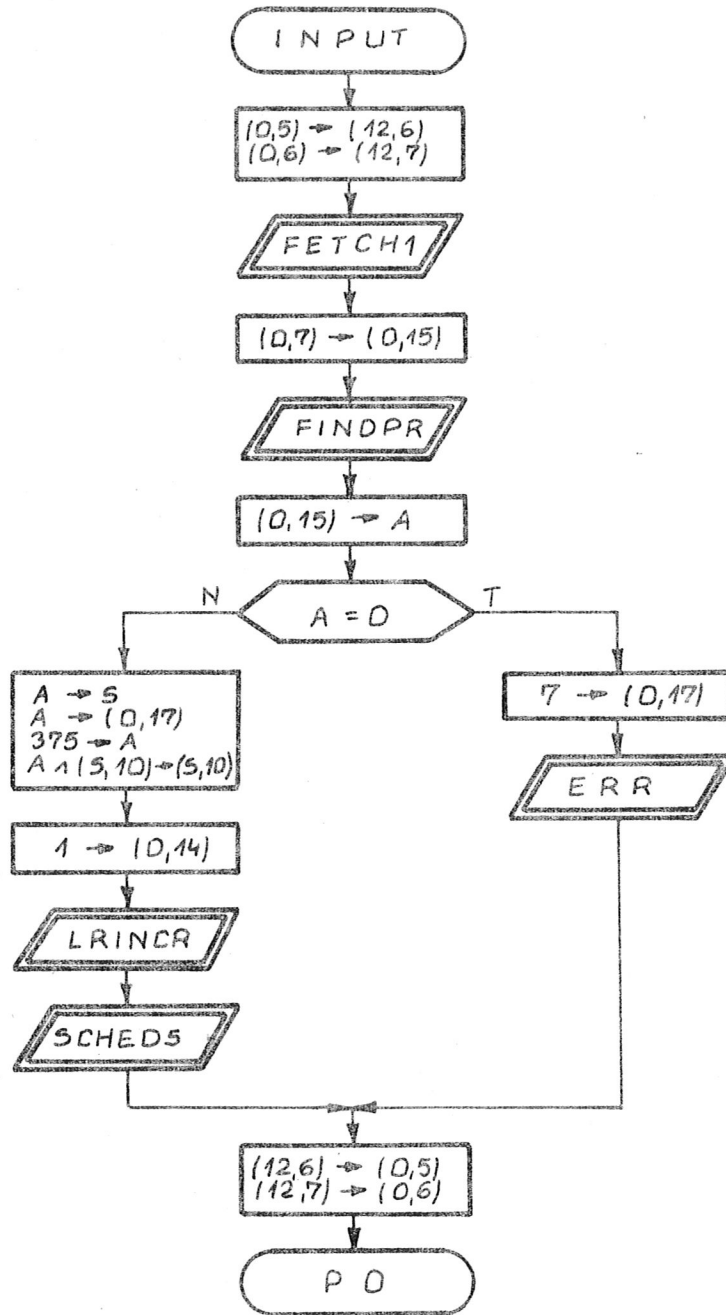
Rys.30f Schemat podprogramu WNM



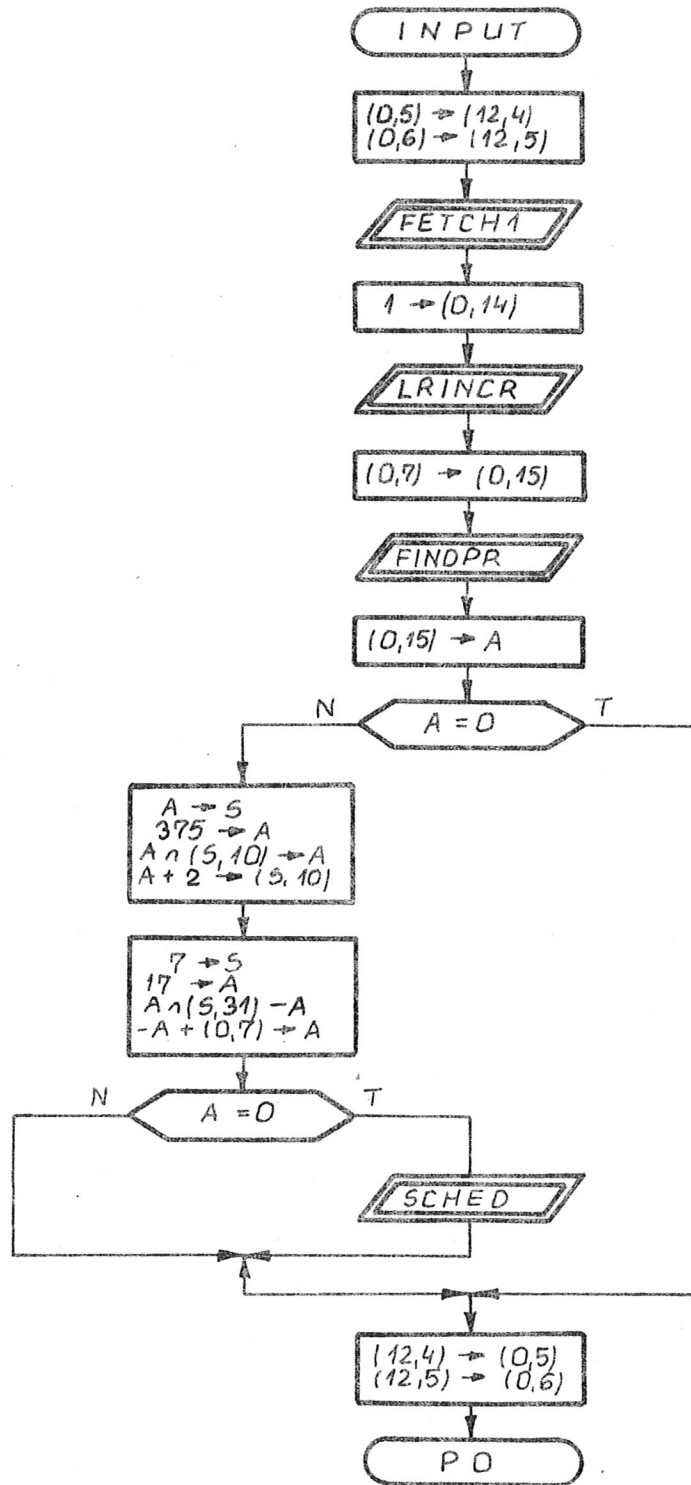
Rys. 30g Schemat podprogramu RMER



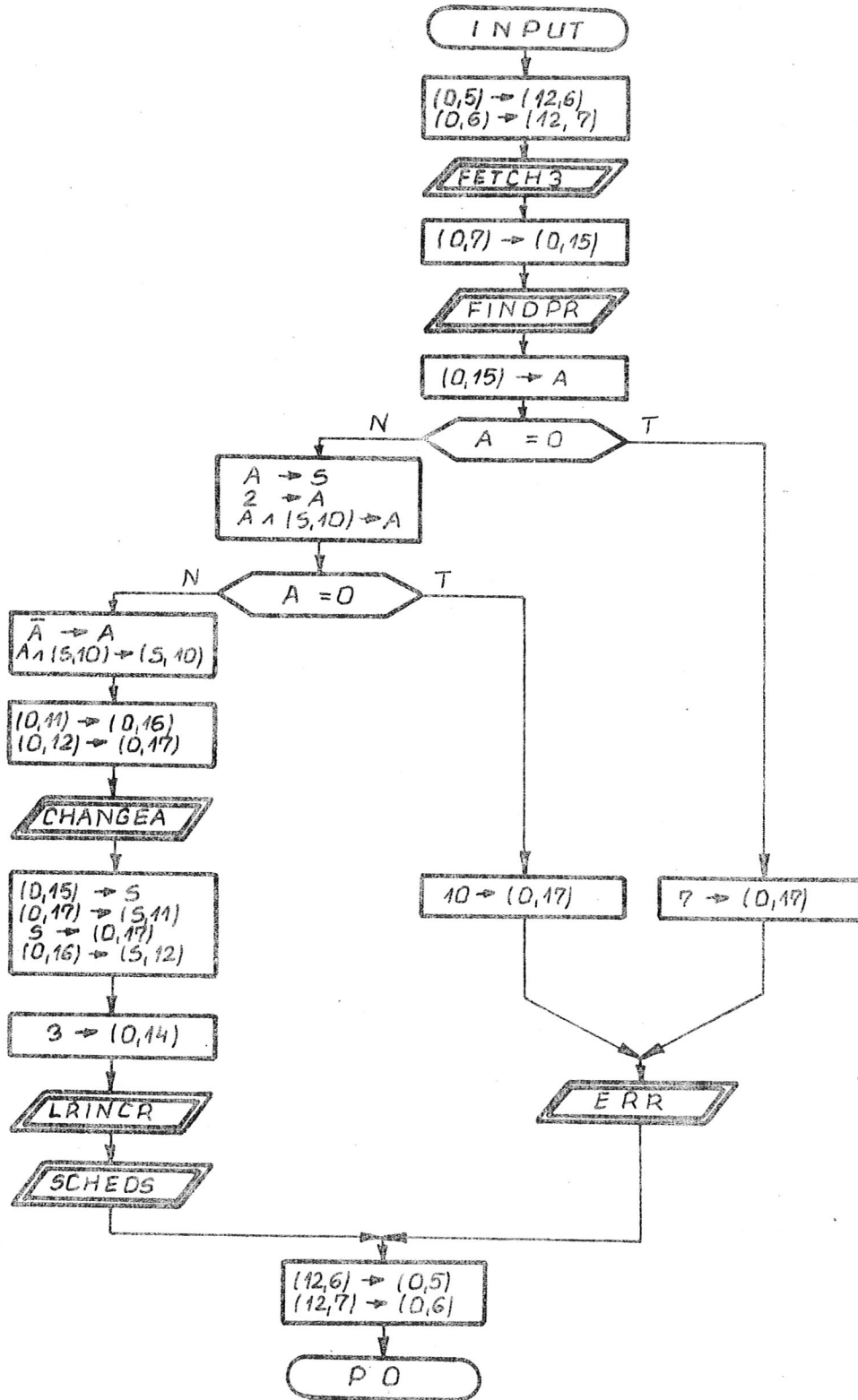
Rys. 30h Schemat podprogramu TMER



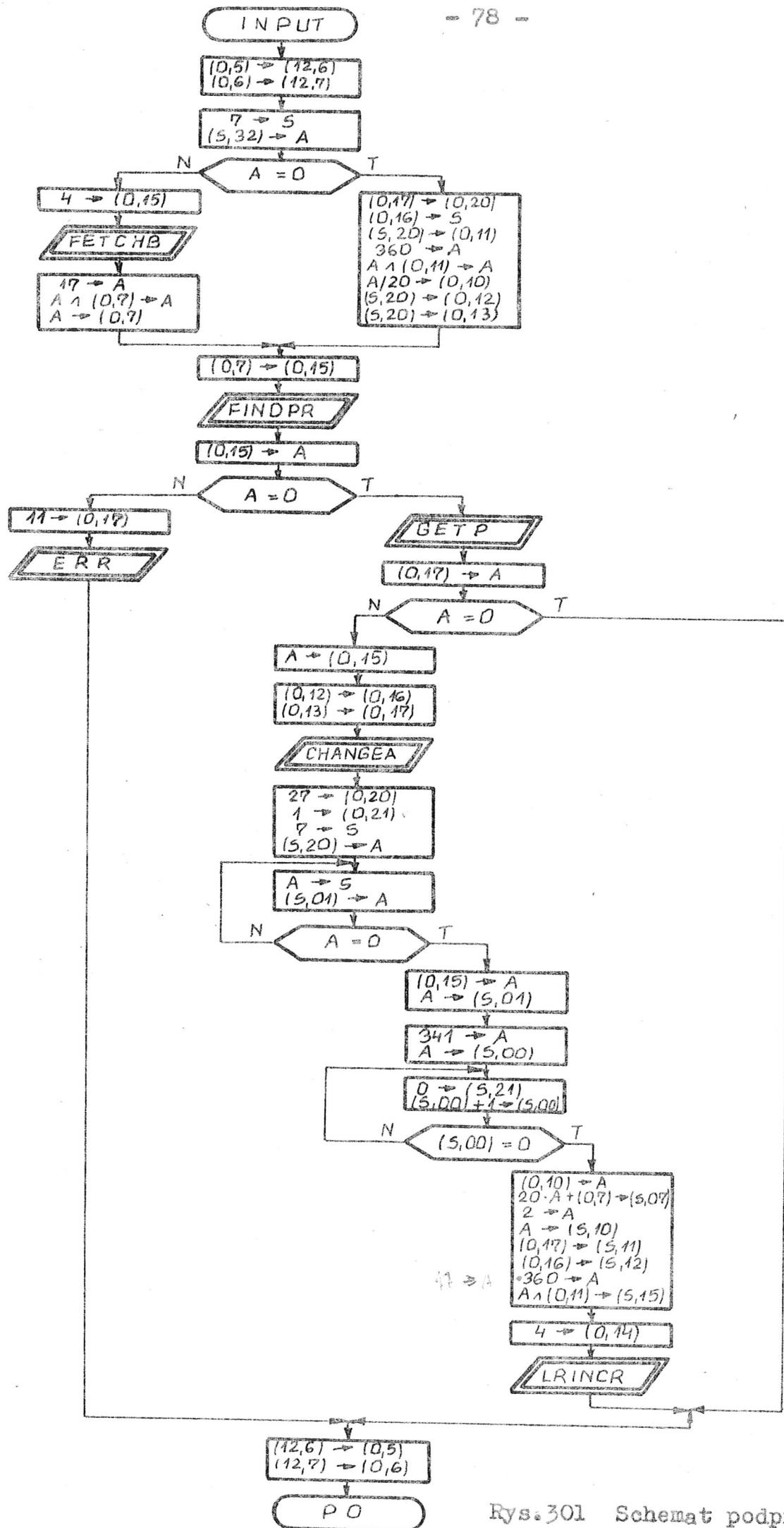
Rys. 301 Schemat podprogramu ST1R



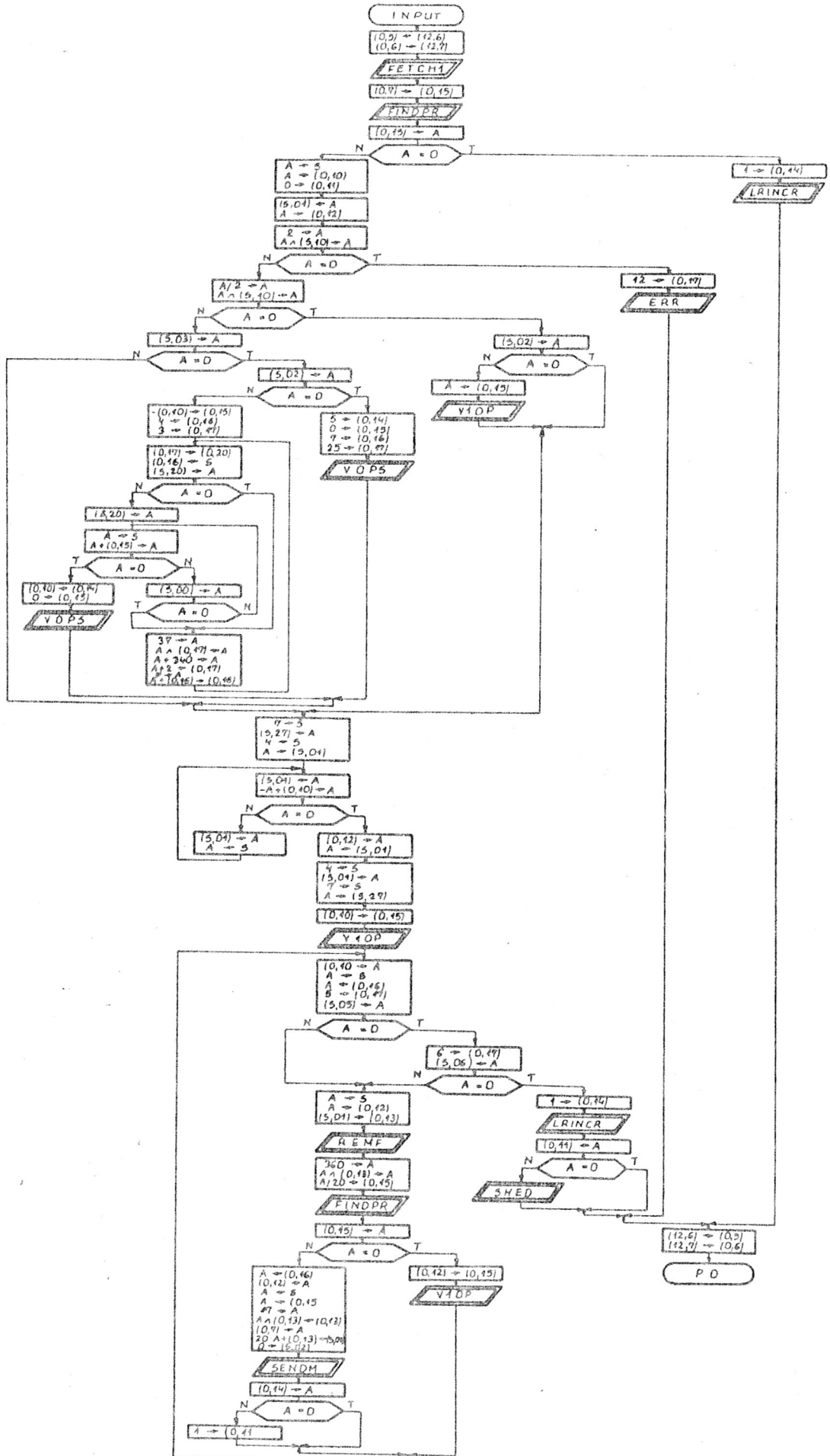
Rys. 30j Schemat podprogramu STPR



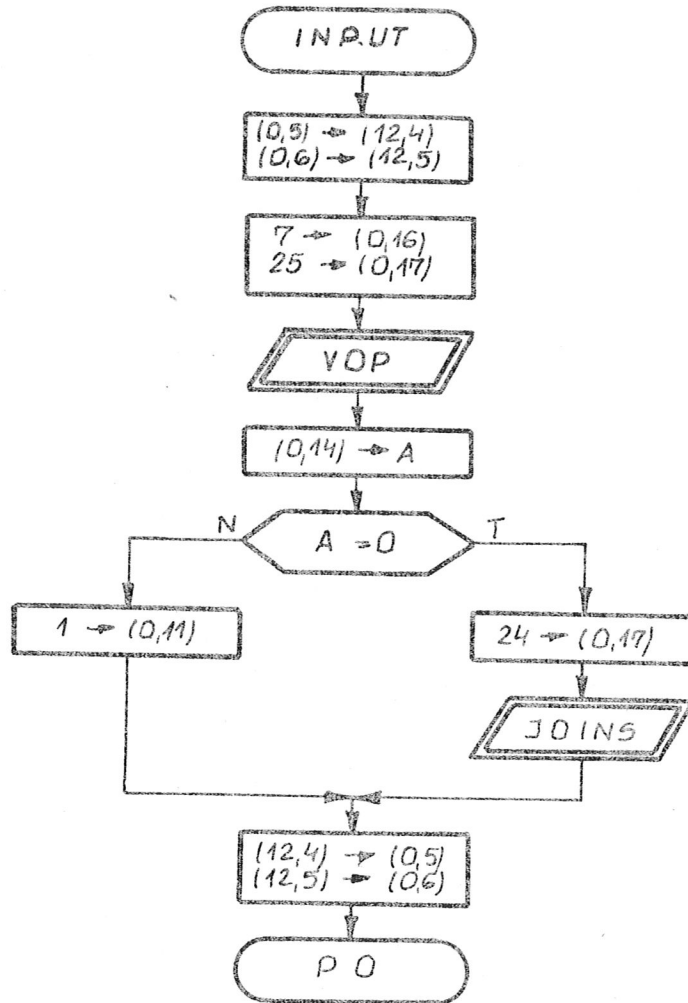
Rys. 30k Schemat podprogramu ST2R



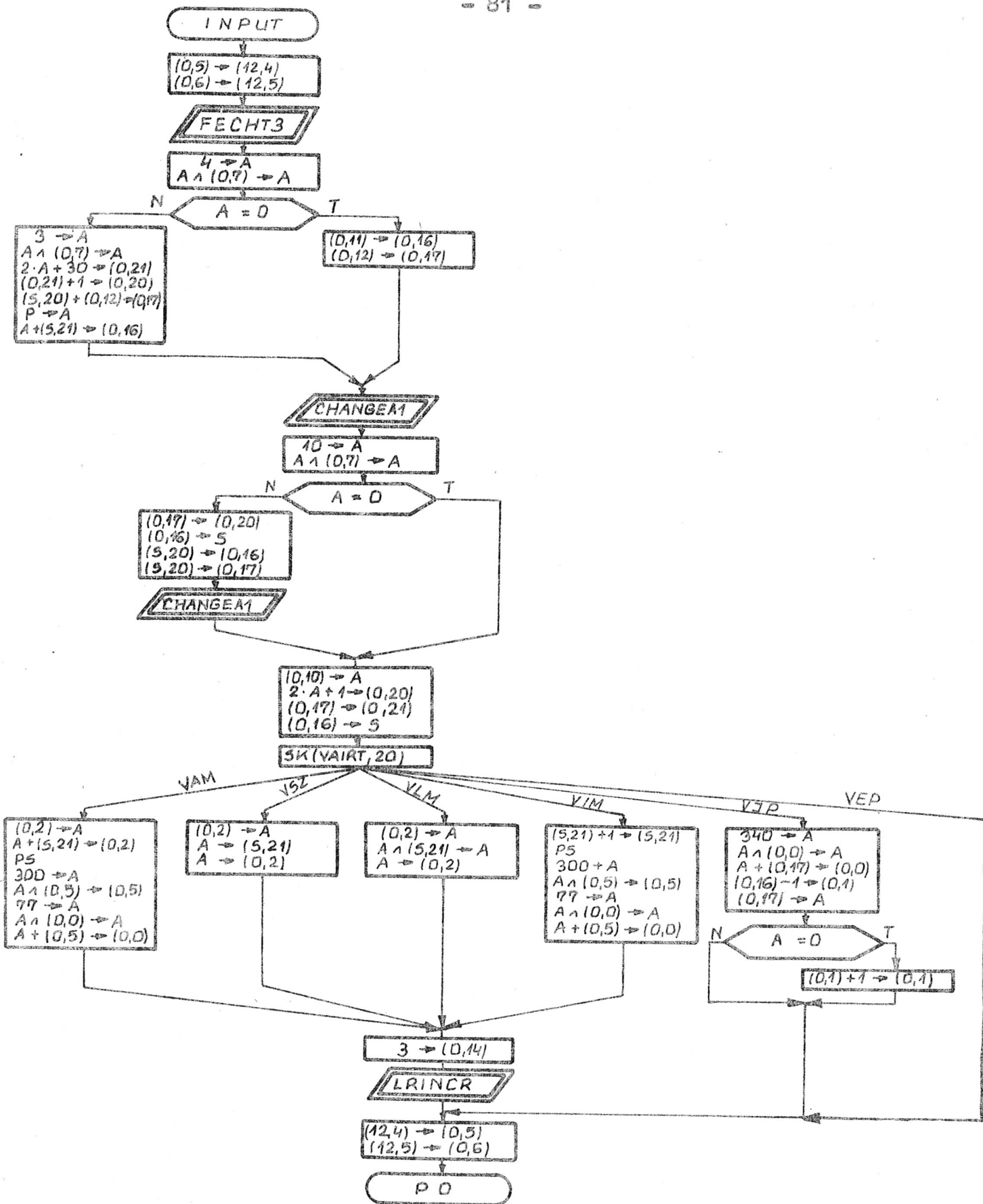
Rys. 301 Schemat podprogramu CRPR



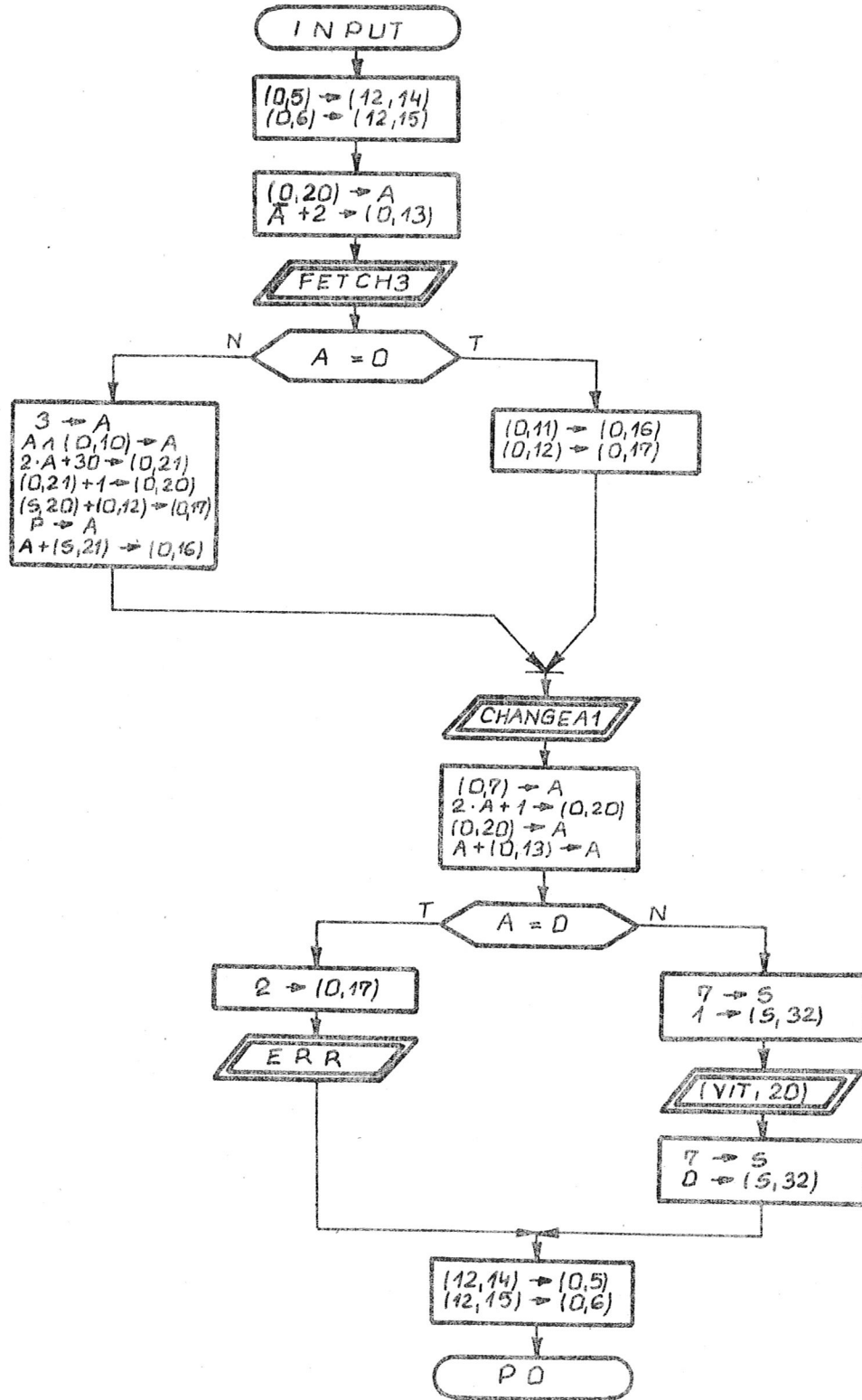
Экз.30a Схемат подпрограмм DEPR



Rys. 30n Schemat podprogramu V10P



Rys. 300 Schemat podprogramu VAIR



Rys.30p Schemat podprogramu EXIR

6. URUCHAMIANIE WIK-a

Uruchamianie WIK-a jest procesem dwuetapowym. W etapie 1 wprowadza się do pamięci operacyjnej "program wczytujący", który jest programem pomocniczym. Jego zadaniem (w etapie 2) jest wprowadzenie do pamięci WIK-a oraz tzw. programu początkowego.

E t a p 1

1. Włączyć maszynę cyfrową.
2. 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 wejścia i wyjścia (patrz [1]).
3. Założyć taśmę oznaczoną napisem PROGRAM WCZYTUJACY do urządzenia wejścia.
4. Włączyć drukarkę DZM 180 z klawiaturą (lub maszynę do pisania FACIT). Przyjęto, że urządzenie to związane jest z linią przerwań nr 2 w bloku przerwań wejścia i wyjścia.
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 PAO.
9. Nacisnąć klawisz IAD.
10. Nacisnąć klawisz LR+1.
11. Nacisnąć klawisz WPR.

Czynności wymienione w pkt. 9,10,11 należy powtarzać tak długo, aż zawartość rejestru LR (wyświetlana przez dolny

rzádek lampek na pulpicie technicznym) będzie większa od

12. Wcisnąć klawisz LR.
13. Nacisnąć klawisz ZER.
14. Wyzerować rejestr LR.
15. Wcisnąć klawisz S-S.

W wyniku wykonania wyżej wymienionych czynności pozostała część taśmy oznaczonej napisem PROGRAM WCZYTUJACY zostanie wczytana oraz zostanie wydrukowany napis " *IN," .

E t a p 2

WIK-2 oraz program początkowy znajdują się na taśmie oznaczonej napisem WIK-2 BIN.

Jak wspomniano poprzednio, taśmę tę można wczytać do pamięci operacyjnej przy pomocy programu wczytującego. W tym celu za napisem " *IN," należy napisać dwucyfrową liczbę ósemkową określającą urządzenie wejścia (numer przerwań we WIK-u generowanych przez urządzenie), przy pomocy którego, zostanie wczytana taśma WIK-2 BIN. Urządzenie to powinno zostać uprzednio przygotowane do wprowadzania informacji. Po wczytaniu taśmy WIK-2 BIN następuje automatyczne uruchomienie komputera wirtualnego. Sygnalizowane jest to napisem " *WIK-2".

Stan taki nazywa się stanem początkowym WIK-a. Istnieją wówczas:

- procesor wirtualny o nazwie 1 - wykonujący program początkowy,

- rozkazy wirtualne: CVI, DVI, GIN, WIN.

Program początkowy umożliwia realizację dwóch zleceń:

- wczytaj program,
- wykonaj program.

Wczytaj program

Format zlecenia:

* RP, <numer urządzenia wejścia> <cr> <lf>

gdzie: <numer urządzenia wejścia> jest dwucyfrową liczbą ósemkową określającą numer przerw w WIK-u generowanych przez urządzenie wejścia, <cr> oznacza znak "powrót karetki", <lf> oznacza znak "wysuw papieru".

Zlecenie to pozwala wczytać do pamięci operacyjnej dowolny program zapisany w kodzie heksadecymalnym (patrz [3]).

Wykonaj program

Format zlecenia:

* EP, <adres> <cr> <lf>

gdzie: <adres> jest pięciocyfrową liczbą ósemkową z przedziału $\langle 0, 17777_8 \rangle$, <cr> oznacza znak "powrót karetki", <lf> oznacza znak "wysuw papieru".

Zlecenie to umożliwia wykonanie (przez procesor o nazwie 1) programu poczynając od adresu <adres>.

Umieszczenie w wykonywanym programie rozkazu skoku do adresu ^{5300_s} umożliwia powtórne wywołanie programu początkowego. Sygnalizowane jest to napisem "*WIK-2".

7. CYTOWANE OPRAWOWANIA

1. Momik 8b. Zasady działania, cz.1, Instytut Maszyn Matematycznych, Warszawa 1972
2. Jerzy Bartoszek, Projekt komputera wirtualnego WIK-2, Instytut Automatyki Politechniki Poznańskiej, Poznań 1976
3. Leszek Masadyński, Jacek Martinek, Jerzy Bartoszek, Język symboliczny AWIK-2, Instrukcja programisty, Raport Środowiskowego Ośrodka Informatyki Politechniki Poznańskiej, Poznań 1977.

Dodatek A

Wykaz błędów sygnalizowanych w tablicy ERRPRT

numer błędu	znaczenie
01	próba wykonania rozkazu wirtualnego, który nie istnieje we WIK-u
02	rozkazem, którego argumenty wskazano pośrednio jest EXI
03	"podwójne" tworzenie rozkazu wirtualnego
04	usuwanie rozkazu CVI lub DVI
05	złe argumenty rozkazu WIN
06	czytanie komunikatu, który nie został przysłany do procesora
07	startowanie procesora, który nie istnieje we WIK-u
10	wykonanie rozkazu ST2 w stosunku do procesora, który nie jest w stanie zatrzymania
11	"podwójne" tworzenie procesora
12	usuwanie procesora, który nie jest w stanie zatrzymania lub zatrzymania i czekania

Dodatek B

Adresy początków podprogramów realizujących WIK-a

Nazwa	Poziom	Adres (ósemkowo)
REMF	0	540
REMS	0	567
JOINS	0	650
FINDM	0	721
FINDPR	0	1035
CHANGEA	0	1074
CHANGEA1	0	1115
FETCHB	0	1145
FETCH1	0	1177
LRINCR	0	1227
LRDECR1	0	1266
REMPR	0	1311
FETCHPR	0	1423
SAFE2021	0	1546
POP	1	1565
VOP	1	1656

Nazwa	Poziom	Adres (ósemkowo)
VOPS	1	1753
SCHEDS	1	2031
SCHED	1	2127
FETCH3	1	2272
CHECKQ	2	2405
ERR	2	2537
SENDM	2	2623
GETP	2	2771
ISERV	3	3105
ERRVI	3	3260
PRISERV	7	3315
IOISERV	7	3411
E3ISERV	7	3457
E2ISERV	7	3470
E1ISERV	7	3501