



# AKADEMIA SZTABU GENERALNEGO WP

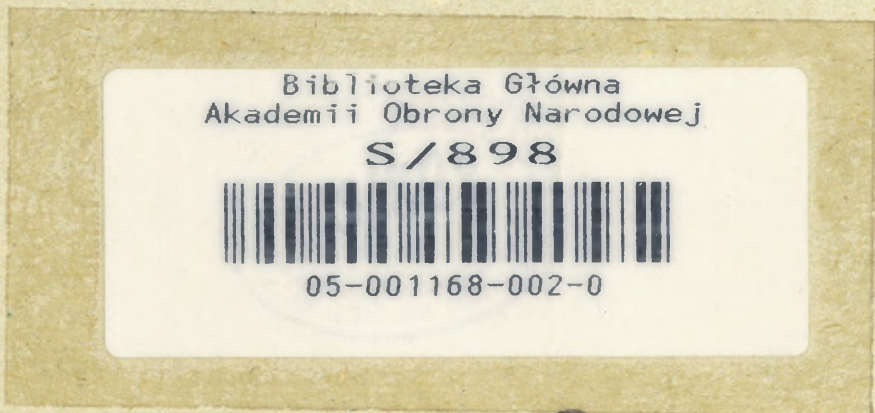
INSTYTUT BADAŃ STRATEGICZNO-OBRONNYCH

ASG WP wewr. 3851/84

Egz. Nr 7

Ppłk mgr inż. Czesław LOCH  
Ppłk mgr inż. Ludwik PIELA

## CHARAKTERYSTYKA SYSTEMU KOMPUTEROWEGO „IRIS — 80”



WARSZAWA

12826

1984



# AKADEMIA SZTABU GENERALNEGO WP

INSTYTUT BADAŃ STRATEGICZNO-OBRONNYCH

ASG WP wewn. 3851/84

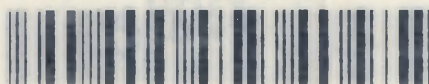
Egz. Nr <sup>7</sup> .....

Ppłk mgr inż. Czesław LOCH  
Ppłk mgr inż. Ludwik PIELA

## CHARAKTERYSTYKA SYSTEMU KOMPUTEROWEGO „IRIS – 80”



Biblioteka Główna  
Akademii Obrony Narodowej  
S/898



05-001168-002-0

WARSZAWA

12826

1984

AKADEMIA SZTABU GENERALNEGO WP

---

INSTYTUT BADAŃ STRATEGICZNO-OBRONNYCH

Egz. nr ...

ASG WP wewn. 3851/84



ppłk mgr inż. Czesław LOCH

ppłk mgr inż. Ludwik PIELA

CHARAKTERYSTYKA  
SYSTEMU KOMPUTEROWEGO  
"I R I S - 80"



---

WARSZAWA

1984 r.

## SPIS TREŚCI

	Str.
1. OGÓLNE WIADOMOŚCI O SYSTEMIE IRIS-80 .....	6
1.1. OPIS SYSTEMU .....	6
1.2. ZESTAW CENTRALNY .....	10
1.2.1. Pamięć operacyjna .....	10
1.2.2. Jednostka centralna .....	11
1.2.2.1. Rejestry ogólne .....	12
1.2.2.2. Słowo stanu programu .....	14
1.2.2.3. Reprezentacja danych .....	18
1.2.2.4. Formaty instrukcji .....	20
1.2.2.5. Adresowanie pamięci .....	22
1.2.2.6. Obliczanie adresu efektywnego .....	22
1.2.2.7. Układ stronicowania PAO IRIS-80.....	24
1.2.2.8. Ochrona pamięci .....	27
1.2.3. System przerwań .....	28
1.2.4. Lista rozkazów IRIS-80 .....	31
1.2.4.1. Pobierania i pamiętania .....	31
1.2.4.2. Arytmetyka stałoprzecinkowa .....	31
1.2.4.3. Porównania .....	32
1.2.4.4. Operacje logiczne .....	32
1.2.4.5. Przesunięcia .....	32
1.2.4.6. Konwersja .....	33
1.2.4.7. Arytmetyka zmienno-przecinkowa .....	34
1.2.4.8. Arytmetyka dziesiętna .....	34
1.2.4.9. Przetwarzanie ciągu bajtów .....	35
1.2.4.10. Operacje na stosach .....	35
1.2.4.11. Rozkazy skokowe .....	35
1.2.4.12. Monitorowanie .....	36
1.2.4.13. Rozkazy uprzywilejowane .....	36
1.3. URZĄDZENIA WEJŚCIA / WYJŚCIA .....	37
1.3.1. Jednostki kanałowe bezpośrednie .....	37
1.3.2. Jednostki kanałowe multiplexerowe .....	38
1.3.3. Jednostki sterujące i urządzenia zewnętrzne .....	40
1.3.3.1. System dyskowy .....	40
1.3.3.2. Pamięci taśmowe .....	41
1.3.4. Urządzenia zewnętrzne elektromechaniczne .....	41

1.3.4.1. Czytnik kart LC-1200 .....	42
1.3.4.2. Perforator kart PC-300 .....	42
1.3.4.3. Drukarka wierszowa LP-1200 .....	42
1.3.4.4. Czytnik-perforator taśmy papierowej .....	42
1.3.5. Konsola operatora .....	43
1.4. JEDNOSTKA TELETRANSMISJI .....	43
1.4.1. Terminale lekkie .....	44
1.4.1.1. Iriscope 200 .....	44
1.4.1.2. Iriscope 300 .....	44
1.4.2. Terminale ciężkie .....	45
1.4.2.1. MITRA-15/RB .....	46
1.4.2.2. MITRA-15/VG .....	47
1.4.2.3. MITRA-15/CD .....	47
Bibliografia do rozdziału 1 .....	49
2. OPROGRAMOWANIE SYSTEMU IRIS-80 .....	50
2.1. CHARAKTERYSTYKI PRAC I ZASOBÓW .....	50
2.1.1. Przetwarzanie wsadowe .....	50
2.1.2. Wielodostęp .....	51
2.1.3. Przetwarzanie w czasie rzeczywistym .....	51
2.1.4. Zasoby .....	52
2.2. OGÓLNA STRUKTURA SIRIS 8 .....	52
2.2.1. Funkcje podstawowe .....	52
2.2.2. Funkcje usługowe .....	55
2.2.3. Funkcje zarządzania .....	56
2.2.4. Funkcje komunikacji z otoczeniem .....	57
2.2.4.1. Zarządzanie zbiorami .....	58
2.2.4.2. Metody dostępu .....	60
2.2.4.3. Przetwarzanie wywołań użytkownika .....	62
2.2.4.4. Język sterujący realizacją programów .....	62
2.2.4.5. Narzędzia uruchamiania .....	64
2.2.5. Zarządzanie transmisją .....	65
2.2.5.1. Wykonywane funkcje .....	65
2.2.5.2. Współpraca programu użytkowego z systemem transmisji ..	66
2.2.5.3. Kolejki .....	67
2.2.5.4. Zarządzanie linią i terminalem .....	67
2.2.5.5. Tryby transmisji .....	68
2.2.5.6. Ochrona przed błędami .....	68
2.2.6. Języki - kompilatory .....	69
2.2.6.1. METASYMBOL .....	69

2.2.6.2. LP-70 .....	70
2.2.6.3. LPS .....	70
2.2.6.4. FORTRAN IV CII - rozszerzony .....	71
2.2.6.5. PL/I .....	72
2.2.6.6. SIMULA 67 .....	73
2.2.6.7. COBOL .....	73
2.2.6.8. ALGOL - IRIS 80 .....	74
2.2.7. Programy usługowe .....	74
2.2.7.1. Procesor konsolidacji programów .....	74
2.2.7.2. Generator konwersji .....	75
2.2.7.3. Sortowanie i łączenie .....	75
2.2.7.3.1. Charakterystyki sortowania .....	75
2.2.7.3.2. Łączenie .....	76
2.2.7.4. Generator zbiorów testowych .....	76
2.2.7.5. Generator systemu .....	76
2.2.7.6. Konserwacja zbiorów .....	76
2.3. WIELODOSTĘP .....	76
2.3.1. Ogólna struktura .....	76
2.3.2. Wymienność prac przetwarzanie wsadowe / wielodostęp .....	78
2.3.3. Zarządzanie terminalem .....	78
2.3.4. Zarządzanie zbiorami .....	79
2.3.4.1. Typy zbiorów .....	79
2.3.4.2. Zbiory przydzielone do terminali .....	79
2.3.5. Opis funkcji .....	80
2.3.5.1. Funkcje podstawowe .....	80
2.3.5.1.1. Zarządzanie zadaniem konwersyjnym .....	80
2.3.5.1.2. Zarządzanie pamięcią i wymianą .....	81
2.3.5.1.3. Sterowanie wejściem / wyjściem terminala .....	82
2.3.5.2. Funkcje sterowania .....	82
2.3.5.3. Funkcje komunikacji z otoczeniem .....	82
2.3.5.4. Funkcje usługowe .....	82
2.3.5.4.1. Zadanie DEMON .....	82
2.3.5.4.2. Wykonawca .....	82
2.3.6. Język sterowania .....	82
2.3.7. Procesory .....	83
2.3.7.1. Edytor tekstu .....	83
2.3.7.2. Narzędzie uruchamiania .....	84
2.3.7.3. BASIC .....	85
2.3.7.4. FORTRAN .....	85
2.3.8. Kontrola działania wielodostępu SIRIS 8 .....	86

2.4. PAKIETY PROGRAMÓW UŻYTKOWYCH .....	86
2.4.1. Biblioteka matematyczna .....	86
2.4.2. Biblioteka statystyczna .....	87
2.4.3. OPALINE .....	88
2.4.4. SOCRATE .....	90
Bibliografia do rozdziału 2 .....	92

## 1. OGÓLNE WIADOMOŚCI O SYSTEMIE IRIS-80

### 1.1. OPIS SYSTEMU

System IRIS-80 jest zestawem o dużej mocy obliczeniowej, pozwalającym na pracę jedno lub wieloprogramową w systemie jedno lub wieloprocesorowym. Strukturę jego budowy oparto na idei umieszczenia głównych funkcji komputera w różnych typach jednostek:

- pamięci operacyjnej,
- procesora centralnego,
- jednostek kanałowych WE/WY
- systemu przerwań.

Częścią centralną całego systemu jest szybka pamięć operacyjna o dużej pojemności, z którą współpracują pozostałe moduły zestawu przy pomocy zgłoszeń. W systemie może znajdować się po kilka jednostek tego samego typu. Może on zawierać jedną lub dwie jednostki centralne przetwarzające informację oraz od jednej do kilku jednostek kanałowych /bezpośrednich lub multiplekserowych/, które organizują współpracę pomiędzy dołączonymi do nich urządzeniami zewnętrznymi a pozostałymi częściami systemu. Kanał bezpośredniego dostępu umożliwia sterowanie urządzeniami pracującymi w czasie rzeczywistym /REAL TIME/.

Rysunek 1 przedstawia konfigurację systemu IRIS-80, zainstalowanego w ASG WP i zawierającą następujące moduły:

#### 1/ Zestaw centralny

- pamięć operacyjna o pojemności 1 Mb,
- jednostka centralna,
- pulpit operatorski /maszyna do pisania, dwa monitory ekranowe bez klawiatury, klawiaturę oraz środki łączności/
- czytnik / perforator taśmy papierowej LPR,
- czytnik kart LC 1200,
- dwie drukarki wierszowe IP-1200,
- perforator kart PC-300,
- jednostka kanałowa,
- jednostka sterująca PT,
- cztery jednostki pamięci taśmowych 1600/800bpi,
- adapter interface'u ATI,
- jednostka sterująca pamięci dyskowych SDA,
- pięć jednostek dyskowych MD 100,
- kanał transmisji blokowej CTQM,
- kanał transmisji znakowej CTC,
- kanał transmisji międzykomputerowej CLIO,
- szafa modemów.



2. Ciężki terminal MITRA-15/RB - z pamięcią operacyjną 32 Ksłowa podłączony do komputera centralnego, poprzez kanał CTQM za pomocą modemów MO-60 i wydzielonej /niekomutowanej/ czteroprzewodowej linii transmisji zapewniającej pracę w trybie synchronicznym z szybkością 2400 bodów.

3. Ciężki terminal MITRA-15/VG - z pamięcią operacyjną 32 Ksłowa podłączony do centralnego komputera poprzez dwa wejścia kanału CTQM za pomocą modemów MO-30, i dwóch czteroprzewodowych, wydzielonych linii, zapewniających pracę w trybie synchronicznym dla linii SL 01 /full duplex/ i ML 01 /half duplex/ z szybkością 2400 bodów.

4. Koncentrator /dyfuzor MITRA-15/CD - z pamięcią operacyjną 24 Ksłowa dołączony do komputera centralnego poprzez kanał CLIO za pomocą modemów MO-60 i wydzieloną, czteroprzewodową linią transmisji danych, zapewniającą pracę w trybie synchronicznym /full duplex/. Do koncentratora / dyfuzora podłączono siedem monitorów alfanumerycznych - asynchronicznych IS-200 za pomocą siedmiu par modemów MO-20 oraz siedmiu dwu lub czteroprzewodowych, wydzielonych /niekomutowanych/ linii transmisji danych, zapewniających pracę w trybie asynchronicznym. Modemy od strony C/D umieszczone zostały w oddzielnej szafie.

5. Dwa monitory alfanumeryczne IS-200 - asynchroniczne podłączone do komputera centralnego poprzez kanał CTC za pomocą dwóch par modemów MO-20 na takich samych zasadach jak i do C/D.

6. Sześć asynchronicznych monitorów alfanumerycznych IS-200 - podłączonych do komputera centralnego poprzez kanał CTC, zapewniających obsługę lokalną, przy czym długość linii łączących /czteroprzewodowych/ nie może być większa niż 150 m.

Na rysunku nie uwzględnione zostały elementy dodatkowe wyposażenia systemu:

- a/ Kopiarki ekranów IS-30 /30 linii/min/ do monitorów alfanumerycznych asynchronicznych IS-200, podłączane na życzenie użytkownika w ilości pięciu egzemplarzy;
- b/ Sygnalizatory telefoniczne MO-85 do ciężkich terminali i koncentratora / dyfuzora dołączane w zależności od potrzeb użytkownika;
- c/ Stanowiska przygotowania danych K-212 - służące do przygotowania danych wejściowych na kartach. Zostaną one umieszczone w oddzielnych pomieszczeniach i nie będą podłączane do systemu.

OZNACZENIA NA RYS. 1:

MP	- Maszyna do pisania konsoli operatora	70001
MO	- Monitor operatora	70001
CPT	- Czytnik / perforator taśmy papierowej	70060
CK	- Czytnik kart - 1200 kart/min.	70140
DW	- Drukarka wierszowa 1200 linii/min.	72244
PK	- Perforator kart 300 kart/min.	70160
KM	- Kanał multiplekserowy	2765
KBD	- Kanał bezpośredniego dostępu	2763
CTQM	- Kanał transmisji blokowej	72660
		72667
CLIO	- Kanał transmisji międzykomputerowej	72671
		72672
CTC	- Kanał transmisji znakowej	70611
ATI	- Adapter interface'u	2781
SDA	- Jednostka sterująca pamięci dyskowych	72252
FD	- Jednostki dyskowe 100 Mb	
FT	- Jednostki taśm magnetycznych 1600/800 bpi	73355
M 3	- Modem MO-30 - 2400 bodów	75630
M 6	- Modem MO-60 2400 bodów	75660
M 2	- Modem MO-20 600/1200 bodów	75620
M15/RB	- Ciężki terminal /remote batch/ MITRA-15	
M15/VG	- Ciężki terminal /remote batch/ MITRA-15 z grafoskopem kolorowym VG 1610	
M15/CD	- Koncentrator /dyfuzor MITRA-15	
MPT	- Maszyna do pisania TELETYPE /konsola operatora terminala/	15001
CK-3	- Czytnik kart 300 kart/min.	15120
DW-4	- Drukarka wierszowa 400 linii/min.	15410A
FTP	- Jednostka taśmy magnetycznej 800 bpi	
XY	- Pisak bębnowy XY /BENSON 122/74/	
PDD	- Jednostka pamięci dyskowej 10 Mb	
VG 1610	- Grafoskop kolorowy z jednostką sterującą /firmy CIT-ALCATEL/	
A	- Monitor alfanumeryczny typu IRISCOPE 200	71525

System IRIS-80 przedstawiony na rysunku 1 pozwala na jednoczesną pracę w trybach:

- przetwarzania lokalnego wieloprogramowego /batch processing/;
- przetwarzania zdalnego z końcówek M-15/RB i M-15/VG poprzez linie transmisji danych;

- podziału czasu /time sharing/ z końcówek alfaskopowych typu IRISCOPE-200 /lokalnie i zdalnie poprzez kanał transmisji znakowej CTC lub zdalnie przez końcówkę M-15 C/D/.

## 1.2. ZESTAW CENTRALNY

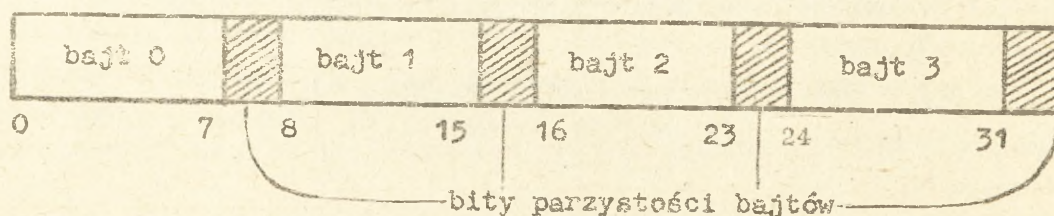
### 1.2.1. Pamięć operacyjna

Pamięć operacyjna systemu IRIS-80 składa się z niezależnych funkcjonalnie bloków. Podstawowym elementem funkcjonalnym jest blok o pojemności 128 Kbajtów, którego pojemność może być zwiększona do 256 Kbajtów przez dodanie dodatkowego modułu. Dwa bloki po 256 Kb tworzą jednostkę operacyjną /znajdującą się w jednej szafie/, posiadającą oprócz bloków rdzeniowych:

- logikę pamięci dla każdego bloku,
- zestaw przełączników umożliwiających ustawienie adresu początkowego danego bloku,
- przełącznik przepłotu adresów umożliwiający pracę: bez przepłotu, przepłot między dwoma blokami danej jednostki, lub przepłot między czterema blokami tejże jednostki,
- logikę zgłoszeń dla każdego bloku PAO do której podłączony jest interfejs jednostki centralnej lub jednostek kanałowych WE/WY.

Każdy blok posiada standardowo dwa zgłoszenia. Ilość ich może być zwiększona do 8 przez dodanie dodatkowych układów, z których każdy posiada logikę dla 2 zgłoszeń i ponumerowane są od 0 do 7. Zgłoszenie o numerze 0 posiada najwyższy priorytet. W rozpatrywanym egzemplarzu IRIS-80 znajdują się 3 układy dostępu dla JC i JK oraz dostęp T służący do testowania pamięci nawet w czasie pracy systemu. Do tego wejścia dołączony jest tester pamięci operacyjnej.

Fizyczna organizacja pamięci jest słowowa. Oznacza to, że odczyt lub zapis informacji z / do pamięci odbywa się w postaci słów 36 bitowych, z których 32 bity są bitami informacyjnymi, zaś pozostałe 4 są bitami kontrolnymi na parzystość dla każdego bajtu. Postać informacji w PAO jest następująca:



Bity parzystości są niedostępne i służą tylko do kontroli poprawności zapisu lub odczytu informacji do / z pamięci.

Zastosowana w IRIS-80 pamięć posiada następujące parametry techniczne:

- czas cyklu 650 ± 40 nS,
- czas dostępu 300 ± 450 nS,
- możliwości przepływu adresów,
- pojemność 1 M bajt /128 Ksłów/,
- wielodostępność.

Pamięć ta może być adresowana bezpośrednio, albo pośrednio przy pomocy stronicowania poprzez pamięć asocjacyjną wbudowaną do jednostki centralnej. W przypadku adresowania pośredniego przez stronicowanie cała pamięć zostaje podzielona na tzw. "strony" po 512 słów /2 Kbajty/ każda. Pozwala to na 16 - krotne zwiększenie pojemności pamięci operacyjnej, ale tylko dla programisty.

Zastosowany system ochrony PAO pozwala ustrzec zapisaną w niej informację przed błędami programowania /zwrot do "cudzych" obszarów PAO/ przez zakaz odczytu lub zapisu informacji, bądź też zakaz wykonania instrukcji znajdującej się na chronionej stronie.

#### 1.2.2. Jednostka Centralna

Każda jednostka centralna składa się z następujących podzespołów:

- układ adresowania pamięci operacyjnej,
- układy odczytu i zapisu informacji,
- układy wykonawcze operacji logicznych i arytmetycznych,
- kontroler rozkazów,
- kontroler wymiany informacji pomiędzy PAO a innymi częściami systemu,
- układ zarządzania systemem przerwań

Jednostka centralna może przetwarzać informacje różnego typu o stałej lub zmiennej długości, wykonywać instrukcje adresujące stałe lub zmienne długości informacji które znajdują się w PAO oraz w rejestrach pamięci topograficznej lub urządzeń zewnętrznych.

Jednostka centralna została zaprojektowana w oparciu o zasadę tzw. wyprzedzania. Oznacza to, że operuje ona w danej chwili trzema instrukcjami, a mianowicie:

- wykonuje instrukcję n,
- przygotowuje do wykonania n+1,
- pobiera instrukcję n+2.

Prowadzi to, co prawda do zwiększenia ilości rejestrów roboczych, lecz znacznie przyspiesza pracę. Wynika to ze sposobu adresowania instrukcji i operandu.

Ilość wykonywanych instrukcji przez JC jest zależna od trybu pracy:

- tryb A - 106 instrukcji
- tryb B } - 117 instrukcji
- tryb C }

W trybie pracy A IRIS-80 wykorzystywany jest częściowo i odpowiada jednostce centralnej typu 10 070.

W trybie pracy B - wykorzystywane są wszystkie resursy dla formatu 10 070 plus siedem instrukcji dodatkowych: LAS, LVAW, LSTP, LRA, PLS, PSS i RIO. Używa systemu stronicowania, systemu przerwania oraz systemu "stos kontekstu".

Tryb pracy C - zarezerwowany jest dla systemu operacyjnego SIRIS8. Jest to tryb, w którym wykorzystywane są wszystkie możliwości systemu IRIS-80. Wykorzystuje instrukcje z 10 070, trybu B oraz dwie dodatkowe: LBR i LIAI. Używa systemu stronicowania, ośmiu rejestrów bazowych, systemu przerwania oraz stosu kontekstu. Tryb ten pozwala na przetwarzanie ciągu bajtów jednej długości do 64 Kbajtów, podczas gdy w trybie A i B tylko 255 bajtów.

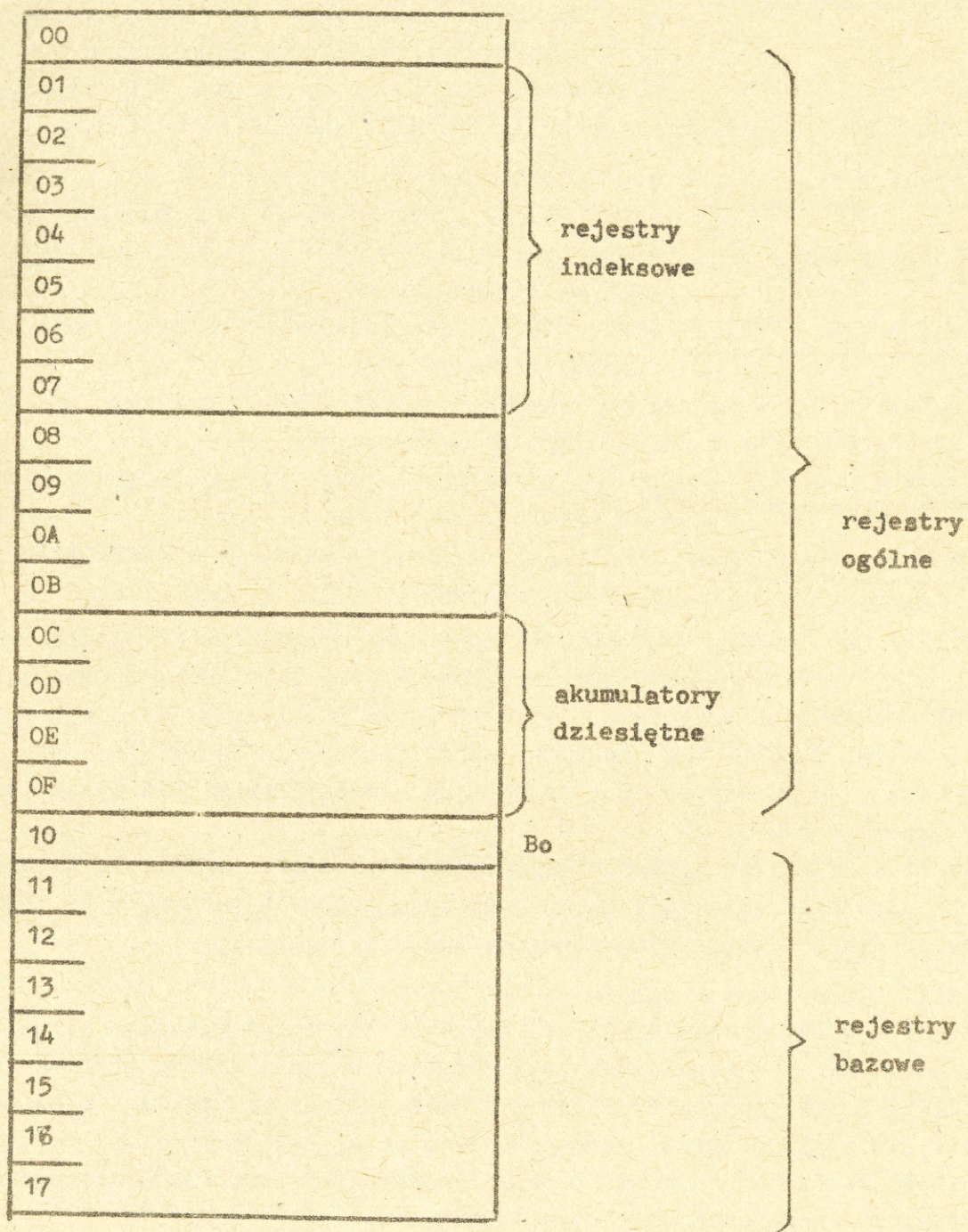
Tryb pracy jednostki centralnej określany jest przez słowo stanu programu które znajduje się w zestawie rejestrów oraz w niektórych przerzutnikach funkcyjnych. Ono właśnie decyduje czy jest tryb pracy A czy B czy C.

#### 1.2.2.1. Rejestry ogólne

Wykonanie, przez jednostkę centralną programu wymaga pewnej ilości rejestrów, które będą zawierały:

- albo operandy pośrednie,
- albo wskaźniki stanu programu,
- albo rozkazy warunkujące działanie JC.

JC IRIS-80 zawiera blok 24 32-bitowych rejestrów ogólnych, do których zwracają się praktycznie wszystkie instrukcje. W celu przyspieszenia wymiany programów można zainstalować do 16 bloków po 24 rejestry. Każdy z tych bloków będzie przydzielony do innego programu. Poniższy rysunek podsumowuje główne funkcje tych rejestrów:



Wykorzystywane one są w następujący sposób:

- Rejestry  $R_{00} + R_{23}$  - wykorzystywane są jako elementy pamięci i mogą zawierać: rozkaz, drugi operand instrukcji lub słowo bezpośrednie.

- Rejestry R<sub>00</sub> - R<sub>15</sub> - służą jako rejestry pomocnicze dla operacji sta-  
ło i zmiennoprzecinkowych. Zawierają operandy i  
wyniki operacji logicznych, składniki operacji  
porównania, operand i wynik operacji dziesiętnych.
- Rejestr R<sub>00</sub> - wykorzystywany do operacji WEJŚCIA / WYJŚCIA.
- Rejestry R<sub>01</sub> - R<sub>15</sub> - służą do transmisji informacji do / z urządzeń  
zewnętrznych przydzielonych do programu.
- Rejestry R<sub>01</sub> - R<sub>07</sub> - służą jako rejestry indeksowe przy obliczaniu  
adresów operandów.
- Rejestry R<sub>12</sub> - R<sub>15</sub> - służą także, jako rejestry pomocnicze przy ope-  
racjach dziesiętnych.

Rejestry R<sub>16</sub> + R<sub>23</sub> są rejestrami bazowymi wykorzystywanymi tylko w trybie pracy C i mają następujące przeznaczenie:

- Rejestr R<sub>16</sub> /Bo/ - wykorzystywany przez rozkazy, działające na łań-  
cuchu znaków.
- Rejestry R<sub>17</sub> + R<sub>23</sub> - służą, jako rejestry bazowe przy obliczaniu  
adresu rozkazu.

Wszystkie te rejestry zbudowane są na układach scalonych i poadresowane od 00 + 23. Stanowią jak gdyby początek pamięci operacyjnej. Oznacza to, że jeśli adres rozkazu lub operandu jest mniejszy od 16 dla trybu A i B i mniejszy od 24 dla trybu C - zwrot nastąpi do rejestrów a nie do PAO, czyli, że adresy pamięci od 00 + 23 nie są praktycznie wykorzystywane przez program.

Jeśli w JC - jest więcej niż jeden blok rejestrów ogólnych, to numer wykorzystywanego aktualnie bloku jest podawany na bitach 56 + 59 "słowa stanu programu" dla przetwarzanego aktualnie programu.

#### 1.2.2.2. Słowo stanu programu

Jest to grupa bitów tworzących kontekst dla przetwarzanego aktualnie programu i określającego stan tego programu w danej chwili przetwarzania. Bity te są grupowane w trzy lub cztery słowa maszynowe i zapamiętywane w PAO jeśli program, którego one dotyczą, jest przerywany. W chwili ponownego restartu programu słowo to jest pobierane z PAO i umieszczane w odpowiednich rejestrach i przerzutnikach.

Dla trybu pracy A i B słowo stanu programu tworzone jest przez dwa słowa maszynowe, natomiast dla trybu C przez trzy lub cztery. Zawartość poszczególnych słów jest następująca:



9

- AV - adres wirtualny

Jeśli bit ten = 1, następuje transformacja adresu wirtualnego na rzeczywisty, jeśli = 0 adres efektywny traktowany jest jako rzeczywisty. Istnieją pewne przypadki nie stosujące się do tej reguły /rozkazy XPSD, LPSD i MTW/.

10, 11

- Maska

MD - bit 10 - maska dziesiętna -

dla MD = 1 - zawieszania działają normalnie,

dla MD = 0 - zawieszania dziesiętne są zakazane.

MA - bit 11 - maska arytmetyczna -

dla MA = 1 - zawieszania arytmetyki zmp. działają normalnie,

dla MA = 0 - zawieszania arytmetyki zmp. nie są obsługiwane

12

- AE - adres rozszerzony

Jeśli AE = 0 praca w trybie A - tzn. że instrukcje mają format 10 070.

Jeśli AE = 1 - tryb BIC - format dla IRIS-80.

13

- PT - tablica segmentów

Jeśli PT = 1 - pobierane jest trzecie słowo maszynowe "słowa stanu programu" zawierające wskaźnik tablicy segmentów. W przypadku wykonywania XPSD, obsługującego przerwanie z pamięci topograficznej lub układu stronicowania, adres powodujący powstanie zawieszania jest ładowany do czwartego słowa "słowa stanu programu".

14

- ME - tryb rozszerzony

Dla ME = 0 tryb pracy A.

Dla ME = 1 tryb pracy B i C.

Bity 12 i 14 ściśle określają tryb pracy maszyny.

dla  $\left. \begin{array}{l} AE = 0 \\ ME = 0 \end{array} \right\}$  tryb pracy A

dla  $\left. \begin{array}{l} AE = 0 \\ ME = 1 \end{array} \right\}$  tryb pracy B

dla  $\left. \begin{array}{l} AE = 1 \\ ME = 1 \end{array} \right\}$  tryb pracy C

- 15 + 31** - AI - adres instrukcji  
Na bitach tych znajduje się adres następnej do wykonania instrukcji, lub w przypadku powstania zawieszenia jest to adres instrukcji, która spowodowała zawieszenie.
- 32 i 33** - nie wykorzystywane i zawsze równe są "0".
- 34 + 35** - CA - klucze dostępu do pamięci  
Te dwa bity służą do otwarcia blokady PAO
- 36** - PNI - ochrona poziomu przerwań  
Bit ten w przypadku IRIS-80 zawsze jest równy "0".
- 37** - ICR - blokada zewnętrznych liczników czasu rzeczywistego  
ICR = 0 - przerwania od licznika czasu z zewnątrz są obsługiwane.  
ICR = 1 - zakaz przyjmowania i obsługi przerwań.
- 38** - IES - zakaz przyjmowania przerwań od WE / WY  
IES = 0 - przerwania od urządzeń WE / WY są obsługiwane,  
IES = 1 - zakaz obsługi przerwań od WE / WY.
- 39** - IIE - zakaz przerwań zewnętrznych  
Jeśli IIE = 1 Przerwania zewnętrzne nie są obsługiwane
- 40 + 46** - S - segment  
Jest to siedem najstarszych bitów części adresowej instrukcji formatu IRIS-80. W trybie B, te bity są dołączane do adresu instrukcji w celu utworzenia 24 bitowego adresu, zaś w trybie C stanowi część integralną adresu /z wyjątkiem XPSD i CAL/.
- 48 + 55** - NI - poziom przerwania  
Bity te określają poziom priorytetu aktualnie wykonywanego programu. Bity te są ładowane z PAO na podstawie bitu 26 SSP /słowa stanu programu/ dla trybów B lub C.
- 56 + 59** - PB - wskaźnik bloku  
Na tych bitach ustawiany jest numer bloku rejestrów ogólnych dla JC wyposażonej w więcej niż jeden blok rejestrów. Strefa

ta może być modyfikowana tylko przez program pracujący w trybie uprzywilejowanym.

**60 + 63** - CI - zezwolenie na prace liczników czasu rzeczywistego  
System IRIS-80 wymaga 4 liczników czasu rzeczywistego o różnej częstotliwości. Każdy z tych bitów steruje jednym z nich, jeśli wartość tego bitu = 1.  
CI = 0 - blokada działania liczników

**64** - nie wykorzystany

**65 + 71** - LTS - długość tablicy segmentów  
Na tych siedmiu pozycjach możemy zaadresować maksymalnie 128 segmentów. Jeśli wszystkie pozycje będą równe zero, oznacza to, że mamy do czynienia tylko z jednym segmentem zerowym.

**72 + 73** - nie wykorzystane

**74 + 75** - UC - numer jednostki centralnej  
Te dwa bity mają sens przy pracy wieloprocesorowej, przy pracy jednoprocessorowej, tak jak w naszym przypadku, nie są wykorzystane.

**76 + 95** - ATS - adres tablicy segmentów  
Na tych 20 bitach podawany jest adres rzeczywisty pierwszego słowa tablicy segmentów.

Jeżeli podczas pracy systemu wykorzystywane jest czwarte słowo stanu programu, to znajduje się tam adres instrukcji która spowodowała zawieszenie programu.

#### 1.2.2.3. Reprezentacja danych

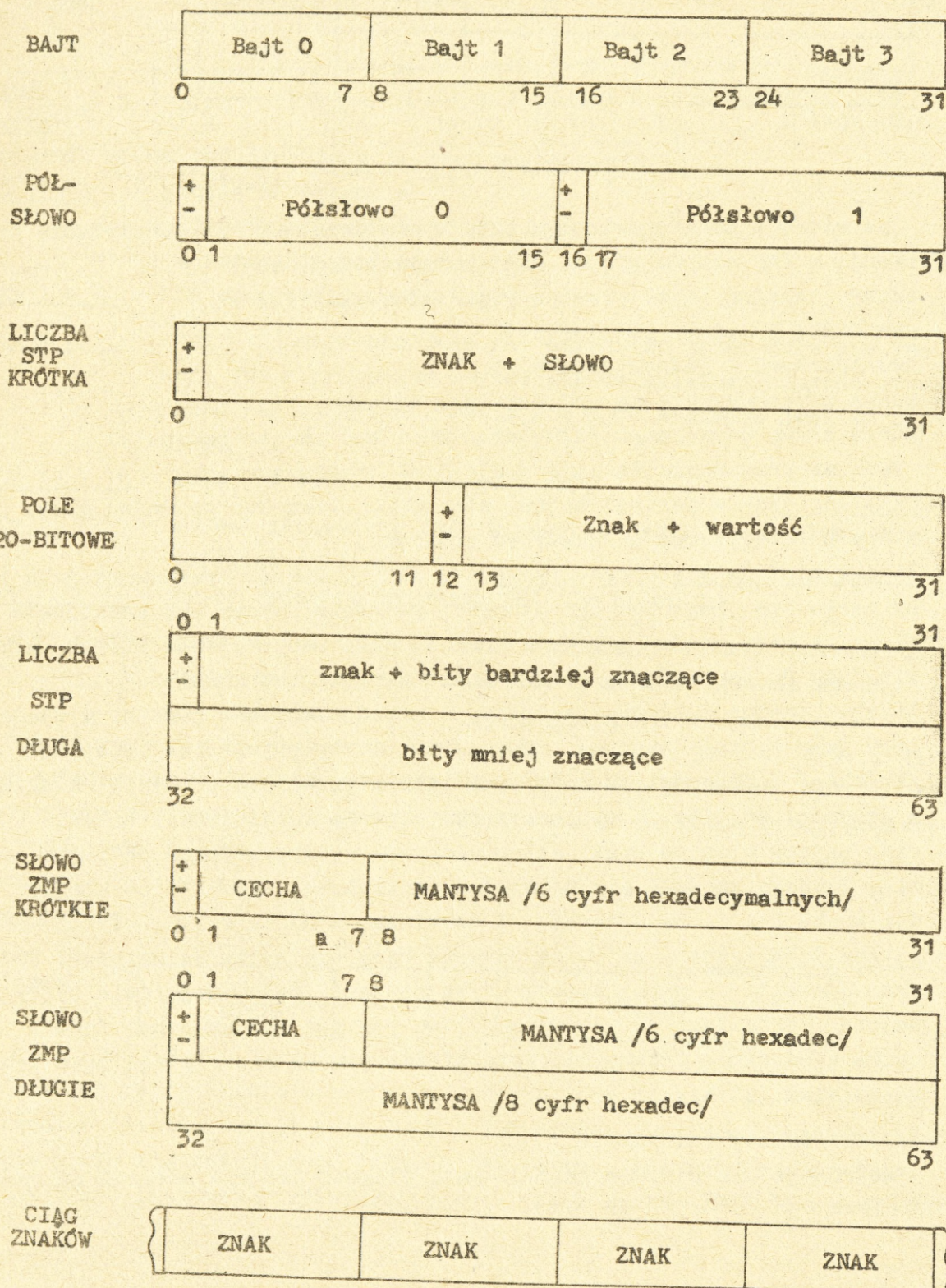
System IRIS-80 przetwarza informację o różnej długości.

Postaci te są:

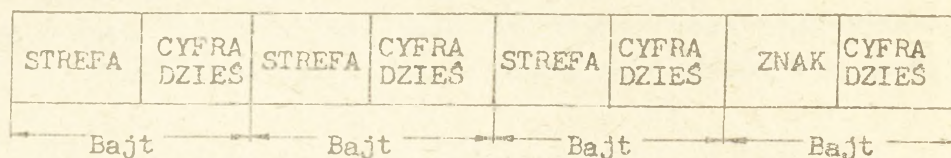
- 8 bitowy bajt lub znak,
- 16 bitowe półsłowo,
- 32 bitowe słowo,
- 64 bitowe słowo podwójne,
- ciąg znaków OO długości od 1 + 256 bajtów,
- liczby dziesiętne o długości od 1 + 31 cyfr.

Wszystkie one fizycznie odnoszone są do jednostki podstawowej, tzn. do

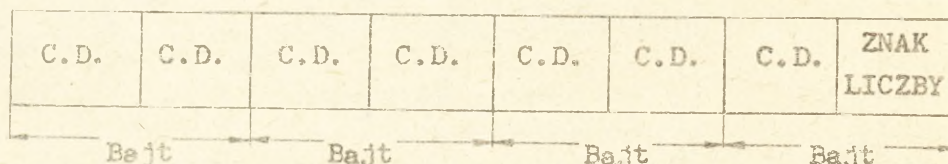
32 - bitowego słowa maszynowego, co wynika z fizycznej organizacji pamięci operacyjnej oraz rejestrów roboczych jednostki centralnej. Graficzny obraz w/w postaci informacji jest następujący:



LICZBA  
DZIESIĘTNA  
ROZPAKOWANA



LICZBA  
DZIESIĘTNA  
UPAKOWANA



Jak widać z przedstawionego obrazu graficznego, przechowywanie informacji w PAO odbywa się na następujących zasadach:

- cztery kolejne bajty przechowywane są w słowie,
- dwa bajty w jednym półsłowie,
- dwa półsłowa w jednym słowie,
- słowa przechowywane są pod dowolnym adresem,
- adres słowa podwójnego /długiego/ musi być zawsze parzysty.

Podczas przetwarzania informacja jest traktowana jako:

- dane logiczne w postaci bajtu, półsłowa, słowa lub podwójne słowa, których interpretacja zależy od programu;
- liczby binarne stałoprzecinkowe tj. liczby zawsze traktowane jako liczby całkowite. Mogą one być długości półsłowa, słowa lub podwójnego słowa. Liczby ujemne przedstawione są jako uzupełnienie liczby całkowitej do 2. Najbardziej znaczący bit liczby jest znakiem.
- liczby binarne zmiennoprzecinkowe - przedstawiane w postaci wykładniczej jako cecha i mantysa. Używane są liczby pojedynczej precyzji /krótkie/ o długości 32 bity oraz liczby podwójnej precyzji /długie/ o długości 64 bity. W obu przypadkach cecha ma długość 7 bitów.
- ciąg bajtów - są to znaki alfanumeryczne wykorzystywane do zarządzania. Zawiera on od 1 + 256 bajtów dla trybu A, oraz 1 + 64 Kbajtów dla trybu B i C. Zaczyna się od dowolnego miejsca w PAO.
- liczby dziesiętne - są to zakodowane binarnie cyfry dziesiętne. Podczas operacji na nich, system używa postaci upakowanej /dwie cyfry w bajcie/, natomiast np. do wydawnictwa postaci rozpakowanej /jedna cyfra w bajcie/. Cztery pierwsze bity bajtu są nie znaczące lub są znakiem liczby.

#### 1.2.2.4. Formaty instrukcji

Instrukcje dla systemu IRIS-80 mają długość jednego słowa i są dwóch rodzajów:

1/ Instrukcje kompatybilne z 10 070 /tryb A i B/.

2/ Instrukcje tylko dla IRIS-80 /tryb C/.

TYP A	*	KOD OPERACJI	R lub L	X	AR						
TYP C	*	KOD OPERACJI	R lub L	X	B	PK					
	0 1	7 8	11 12	14 15	17 18	31					

Znaczenie poszczególnych pozycji jest następujące:

Bit 0 - rodzaj adresowania

-  $x = 0$  adresowanie bezpośrednie

-  $x = 1$  adresowanie pośrednie

Bity 1 + 7 - kod operacyjny instrukcji

Bity 8 + 11 - numer rejestru ogólnego /R/ używanego przez daną instrukcję lub w przypadku arytmetyki dziesiętnej długość argumentu /L/ w bajtach. Jeśli L = 0000 argument jest 16 bajtowy.

Bity 12 + 14 - numer rejestru indeksowego /x/.

Bity 15 + 31 - adres rzeczywisty /AR/ argumentu jeśli  $x = 0$ , lub adres pośredni jeśli  $x = 1$ .

Bity 15 + 17 - numer rejestru bazowego /B/ dla trybu C.

Bity 18 + 31 - jest to tzw. przesunięcie krótkie. Jest ono dodawane do zawartości rejestru bazowego w celu otrzymania adresu wirtualnego.

Postać powyższą ma większość instrukcji. Istnieją jeszcze tzw. instrukcje bezpośrednie, dla których bity 12 + 31 tworzą argument. Bit 12 jest wtedy znakiem. Liczba ta może być rozszerzona do 32 bitów przez powielenie bitu znaku w lewo.

Instrukcje te nie muszą zwracać się do PA0. Są to instrukcje typu B i mają postać:

TYP B	*	KOD OPERACJI	R	+ -	LICZBA
	0 1	7 8	11 12		31

Oprócz tych typów istnieje typ I, w których bity 15 + 31 stanowią adres pośredni - dla instrukcji typu 10-070 lub bity 8 + 31 tzw. przesunięcia długie - dla instrukcji typu IRIS-80. Są to tzw. instrukcje pośrednie. Ogólna ilość instrukcji wykorzystywanych przez system IRIS-80 wynosi 117.

#### 1.2.2.5. Adresowanie pamięci

Przy adresowaniu operujemy pewnymi pojęciami:

- AE - adres efektywny - jest to adres otrzymywany przez dodanie przesunięcia do rejestru bazowego. Używany przy adresowaniu pośrednim i indeksowanym. Dla trybu A i B jest on 19 bitowy, dla trybu C - 26 bitowy. Jest zawsze adresem bajtowym.
- AV - adres wirtualny - inaczej zwany urojonym. Dla trybu A i C bezpośrednio, dla trybu B przez dołączenie numeru segmentu /S/ ze słowa stanu programu. Jest zawsze bajtowy. Długość jak AE.
- AR - adres rzeczywisty - uzyskiwany jest poprzez transformację adresu wirtualnego. Dla trybu A przez pamięć topograficzną, dla B i C przy pomocy układów stronicowania. Adres ten jest zawsze adresem bajtowym i ma długość dla trybu A - 19 bitów, dla trybu B i C - 22 bity.

ADRES REJESTRU: Jeżeli adres jest mniejszy:

- dla trybu A od 24
- dla trybów B i C od 96

będzie to adres rejestru ogólnego lub bazowego a nie adres PAO i zwrot nastąpi do odpowiedniego rejestru.

Adresy te w PAO są dostępne jedynie pośrednio:

- dla trybu A poprzez pamięć topograficzną,
- dla trybów B i C poprzez układ stronicowania.

Ponieważ adres odwołania do pamięci operacyjnej jest słowowy a instrukcje działają na bajtach, półsłowach, słowach lub podwójnych słowach, musi więc być zapewniony dostęp do odpowiedniego bajtu lub półsłowa w słowie. Dla tego też celu rejestr instrukcji został wydłużony w prawo o 2 pozycje /32 i 33/, które będą określały numer bajtu lub półsłowa w całym słowie .

#### 1.2.2.6. Obliczanie adresu efektywnego:

##### a/ Adresowanie bezpośrednio bez indeksacji

Adres efektywny tym sposobem otrzymujemy przez dodanie DK do zawartości rejestru B

$$AE = DK + /B/$$

Jeśli B = 0 wtedy AE = DK

Adres dla tego sposobu zawarty jest na bitach 12 - 31 dla wszystkich trybów pracy A, B i C.

##### b/ Adresowanie pośrednie bez indeksacji

W tym przypadku adres efektywny będzie wskazywał adres argumentu. Dla trybu A będzie on zawarty na bitach 12 - 31, a dla trybów B i C na bitach 8 - 31.

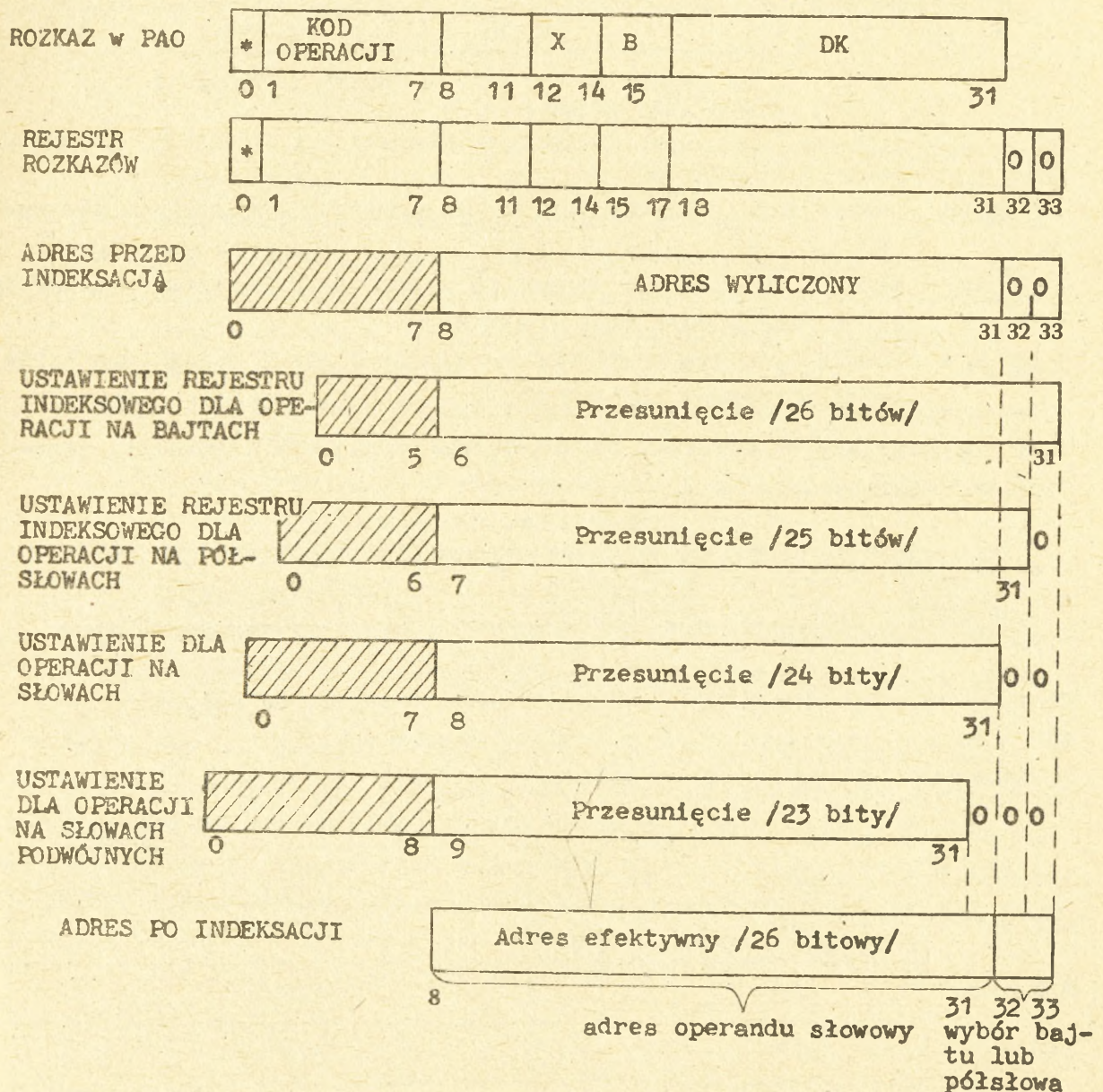
c/ Adresowanie pośrednie z indeksacją

Aby dostać się do odpowiedniego bajtu lub półsłowa należy wprowadzić indeksację, która realizowana jest przy pomocy bitu "0" / $\pi$ / 1 pola X danej instrukcji. Zawartość rejestrów indeksowych X jest traktowana jako przesunięcie i dodawana do rozszerzonego /o pozycje 32 i 33/ rejestru adresowego. Zawartość rozszerzenia wskazuje:

- bit 32 nr półsłowa w słowie,
- bity 32 i 33 nr bajtu w słowie.

Indeksacja jest ostatnią operacją w ciągu obliczeniowym adresu i umożliwia dostęp do wszystkich bajtów w PAO.

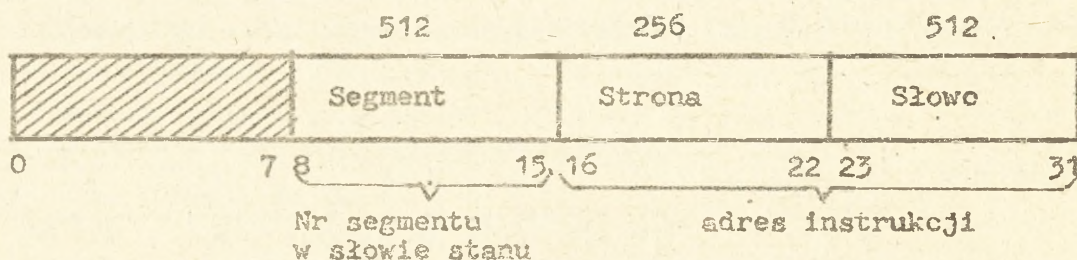
Poniżej przedstawiony jest graficznie sposób wykonania indeksacji dla trybu pracy C.



### 1.2.2.7. Układ stronicowania PAO - IRIS-80

Jest to układ, który pozwala na rozszerzenie pamięci operacyjnej do 64 Mbajtów ale tylko dla programisty. Maksymalna pamięć fizyczna dla IRIS-80 może zawierać 1 M słów - czyli 4 Mbajty, do której zaadresowania potrzebny jest adres na 20 bitach. Programy są pisane dla pamięci wirtualnej /64 Mbajty/ wymagającej adresu 24 bitowego który musi być przetworzony w JC na adres rzeczywisty. Wykonywane to jest przy pomocy tablic. Jest tylko jedno ograniczenie - program nie może być dłuższy niż 32 strony.

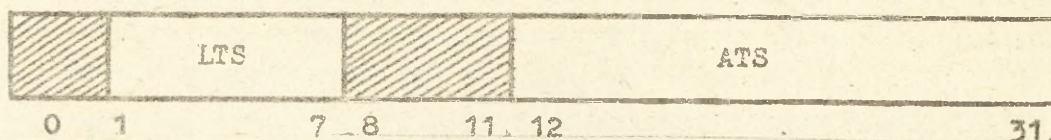
Budowa adresu utworzonego przez JC ma następującą postać:



Dzieli się on na trzy części:

- segment - określa nr segmentu i pobierany jest ze słowa stanu programu.
- strona - określa nr strony w segmencie - jest to 7 bardziej znaczących bitów części adresowej instrukcji.
- słowo - określa nr słowa w stronie - jest to 8 mniej znaczących bitów części adresowej instrukcji.

W JC znajduje się 32 - bitowy rejestr zwany "Wskaźnikiem tablicy segmentów" posiadający budowę:



Określa on przy pomocy LTS - długość tablicy segmentów w słowach oraz ATS - adres pierwszego słowa w tablicy segmentów.

Sama tablica segmentów ma budowę:

	0	7	8	9	10	11	12		31
ATS + 0	LTP		PS				ATP		
ATS + 1	LTP		PS				ATP		
ATS + 2	LTP		PS				ATP		
ATS + 3	LTP		PS				ATP		

Podaje - LTP - długość tablicy stron,  
 - PS - kod ochrony segmentu - zabezpieczający programy na poziomie pamięci wirtualnej.  
 - ATP - adres tablicy stron - jest to adres rzeczywisty pierwszego słowa w tablicy stron, która odpowiada temu segmentowi.  
 Każdy segment skojarzony jest z tablicą stron, utworzoną z kolejnych pól słów o następującej budowie:

	0	1	2	3	4	5	
U	A	PPG	PP				PR
U	A	PPG	PP				PR
U	A	PPG	PP				PR
U	A	PPG	PP				PR

- gdzie: U - BIT UŻYCIA - pozwala systemowi rozpoznać, czy dana strona była używana, czy nie. Bit ten ustawiany jest na "1" przy pierwszym zwróceniu się do danej strony.
- A - BIT WYMIANY - system rozpoznaje po nim czy dana strona została zmieniona, czy nie. Jeśli w stronie nie wprowadzono zmian, można ją wykorzystać bez powtórnego przepisywania z pamięci zewnętrznej.
- PPG - BIT OBECNOŚCI STRONY - jeśli = 1 - oznacza, że dana strona znajduje się w PAO, jeśli = 0 - strona wirtualna nie została przesłana do PAO. Każdy zwrot do strony, dla której PPG = 0 będzie powodował zawieszenie.
- PP - BIT PROTEKCJI - zapewnia ochronę programów lub poziomów pamięci wirtualnej.
- PR - STRONA RZECZYWISTA - jest to nr strony rzeczywistej, która odpowiada stronie wirtualnej, figurującej jako adres początkowy.

Każdorazowe obliczenie adresu przy pomocy opisanych powyżej tablic jest pracochłonne. W celu przyspieszenia przekształcania stron wirtualnych na strony rzeczywiste została zastosowana w układzie stronicowania IRIS-80 pamięć asocjacyjna składająca się z 16 słów. Pozwala ona otrzymywać bezpośrednio adresy 16 stron rzeczywistych bez potrzeby zwracania się do tablic. Budowa słowa tej pamięci jest następująca:

0	6 7	14 15 16 17 18	28 29 30 31
Segment	Strona wirtualna	A	PPE PR C <sub>1</sub> C <sub>2</sub>
Segment	Strona wirtualna	A	PPE PR C <sub>1</sub> C <sub>2</sub>

PPE - ochrona strony efektywnej.

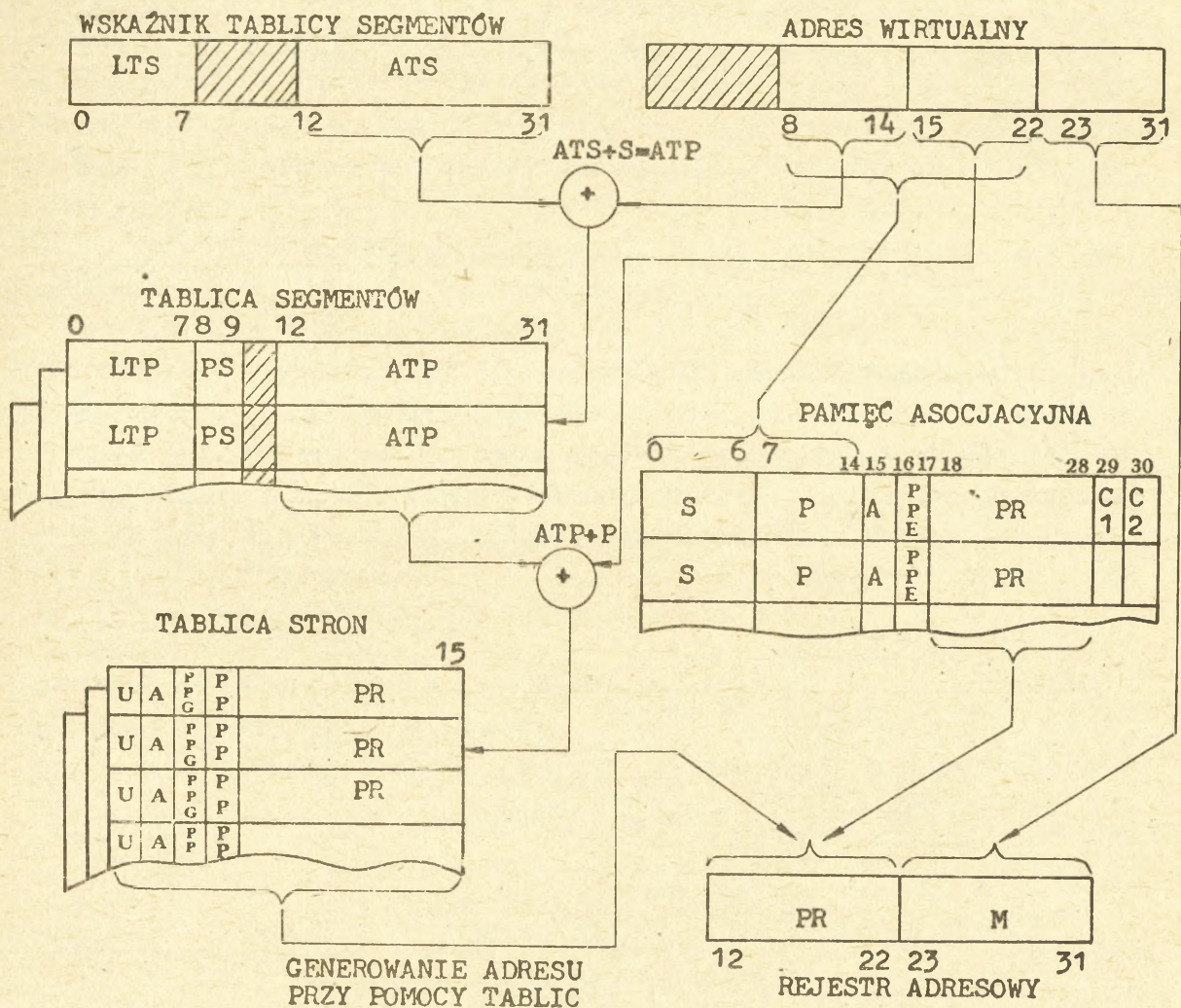
C<sub>1</sub>, C<sub>2</sub> - wskaźniki odnowienia zawartości pamięci asocjacyjnej sygnalizujące, że coś w tej stronie zostało zmienione.

Zasada działania układu stronicowania jest następująca:

- do adresu tablicy segmentu /ATS - ze wskaźnika tablicy segmentów/ dodawany jest nr segmentu /S - ze słowa stanu programu/ - przez co otrzymujemy adres segmentu. Jeśli S = 0 - mamy do czynienia tylko z segmentem zerowym. Uzyskany adres pozwala odczytać adres pierwszej strony w tablicy stron, po dodaniu do której nr strony /część P adresu instrukcji/ mamy dostęp do wybranej strony rzeczywistej - /PR/. Jeśli do tego dołączymy adres słowa /część M adresu instrukcji/ otrzymamy adres słowa rzeczywistego. Adres ten jest ładowany do rejestru adresowego jednostki centralnej /L/.

Procedura ta obowiązuje podczas obliczania adresu rzeczywistego przy pomocy tablic. Sprawa się upraszcza, jeśli mamy do dyspozycji pamięć asocjacyjną. Jest ona przeszukiwana na zawartość S i P adresu wirtualnego. W przypadku znalezienia tej wartości część PR z pamięci asocjacyjnej odpowiadająca danemu S i P przepisywana jest do rejestru adresowego. Jeżeli natomiast brak jest tej wartości w pamięci, adres liczony jest przy pomocy tablic i strona przepisywana jest z pamięci zewnętrznej do PAO, a jej adres wirtualny i rzeczywisty łącznie ze wskaźnikami umieszczany jest w pamięci asocjacyjnej.

Najlepiej zasadę działania układu stronicowania zobrazuje poniższy rysunek:



Cała ta procedura nie jest znana programiście, który zwraca się zawsze do początku obszaru wirtualnego. Ułatwia ona zarządzanie pamięcią, szczególnie przy paracy wieloprogramowej oraz uwalnia system od okresowej translacji programów.

#### 1.2.2.8. Ochrona pamięci

Organizowana jest zarówno na poziomie pamięci wirtualnej, jak i pamięci rzeczywistej. Ma na celu zabezpieczenie przydzielonych danemu użytkownikowi obszarów pamięci przed dostępem przez innych użytkowników tego systemu. Wykorzystywana jest tylko w NORMALNYM trybie pracy maszyny, natomiast nie jest brana pod uwagę w trybie UPRZYWILEJOWANYM.

Na poziomie pamięci wirtualnej ochronie podlegają zarówno segmenty /decydują o tym bity PS/, jak i strony /bity PP/. Ich odpowiednikiem w pamięci asocjacyjnej są bity PPE, których stan decyduje o:

PPE

- 00 - JC może czytać, pisać lub wykonywać instrukcje w obszarze danej strony.
- 01 - może tylko czytać i wykonywać instrukcje. Zakaz zapisu.
- 10 - tylko odczyt.
- 11 - zakaz dostępu do strony.

Podobnie ma się rzecz na poziomie pamięci rzeczywistej, gdzie ochronie podlega strona z tym, że ochronie podlega tylko pierwsze 512 K bajtów pamięci rzeczywistej. Również i tu ochrona działa tylko w trybie NORMAL-NYM. O dostępie do danego obszaru decyduje klucz, czyli bity 34 i 35 słowa stanu programu.

1.2.3. System przerwania

Różnego rodzaju zdarzenia zachodzące na zewnątrz systemu jak i wewnątrz będą powodowały zmianę stanu procesora centralnego i będą wymagały odpowiedniej jego reakcji. System IRIS-80 rozpoznaje i obsługuje 10 klas zdarzeń które powodują przerwania, są to:

1. Operacje nielegalne.
2. Przekroczenie zakresu stosu.
3. Nadmiar operacji stałoprzecinkowych.
4. Błąd operacji zmiennoprzecinkowych.
5. Błąd operacji dziesiętnych.
6. Błąd niesprawności logiki układowej.
7. Przerwania programowe.
8. Instrukcje wywołania systemu operacyjnego.
9. Wewnętrzne liczniki czasu rzeczywistego i przerwania programowane.
10. Uszkodzenie zasilania.

Niektóre z nich wymagają natychmiastowej reakcji, inne mogą być obsłużone nieco później. Ze względu na to przydzielone zostały im priorytety i tak:

- 1-szy priorytet posiada klasa 6 - błąd logiki,
- 2-gi " posiadają klasy 1, 2, 3, 4, 5, 6, 8,
- 3-ci " posiada klasa 7,
- 4-ty " " " 9,
- 5-ty " " " 10.

Jeśli jedno z tych zdarzeń pojawi się w czasie wykonywania programu, jest on przerywany na czas obsługi przerwania, a następnie ponownie restartowany i wykonywany dalej.

System przerwania IRIS-80 jest systemem hierarchicznym i może zawierać do 224 poziomów przerwania podzielonych na 16 grup, z których każda zawiera 16 poziomów. Grupy są tworzone przez wewnętrzne połączenia logiczne i możemy je zmieniać, natomiast w samej grupie poziom priorytetu jest ściśle określony i nie może być zmieniany. Numer priorytetu poziomu posiada długość ośmiu bitów i jest on wyższy w miarę wzrostu zawartości tych bitów. Numery te są przyporządkowane do poszczególnych grup przerwania oraz do poziomów wewnątrz grupy i powinny być zgodne z zaszytymi w logice układowej.

Na każdy poziom przerwania składa się wiele elementów:

- miejsce w PAO - gdzie zawarta jest instrukcja, która pozwala odwołać się do odpowiedniego programu obsługi danego przerwania;
- numer priorytetu przerwania - składa się z 8 bitów i jest podzielony na dwie części po 4 bity z których bardziej znaczące określają grupę przerwania dla danego poziomu /może być 16/, a mniej znaczące 4 bity określają miejsce poziomu wewnątrz grupy /16/;
- numer priorytetu w JC - składa się też z 8 bitów, z których część bardziej znaczącą przesyła urządzenie powodujące przerwanie. Zawartość ta powinna być identyczna;
- przerzutnik obsługi - ustawiany programowo, decyduje o tym czy dany poziom przerwania ma być obsługiwany czy też nie /maskowanie/;
- dwa przerzutniki stanu - określające stan poziomu przerwania w danej chwili:
  - Nie przygotowane /desarme/ - w tym stanie żadne żądanie przerwania nie będzie brane pod uwagę przez system przerwania.
  - Przygotowane /arme/ - jest to normalny stan poziomu i będzie obsługiwane przez maszynę po spełnieniu warunków.
  - Oczekiwanie /attente/ - w tym stanie przerwanie zostanie przyjęte, lecz obsłużone dopiero wtedy gdy nie będzie przerwania o wyższym priorytecie.
  - Aktywny /actif/ - JC - pobiera przerwanie do wykonania i odwołuje się do programu obsługi.

W przypadku zgłoszenia przerwania w stanie oczekiwania - system porównuje jego priorytet z priorytetem wykonywanego programu i jeśli jest on wyższy od priorytetu programu, program jest przerywany. Następuje obsługa przerwania a następnie powrót do poprzedniego programu. W zasadzie JC dopuszcza przerwanie programu w czasie ładowania jego kontekstu lub w czasie wykonywania operacji na stosach.

Przerwanie, po jego obsłużeniu, przechodzi do stanu przygotowany lub nieprzygotowany.

Ze względu na kolejność obsługi, przerwania dzieli się na 16 grup:

- Grupa 0 /najwyższy priorytet obsługi/ - posiadająca 6 poziomów:
  - 4 poziomy dla liczników wewnętrznych czasu,
  - 1 poziom dla błędu PAO /np. BP/.
  - 1 poziom wolny.
- Grupa 1 /wyzerowanie liczników/ - posiada 4 poziomy i obsługuje do czterech liczników. Może być blokowana przez 37 bit słowa stanu programu.
- Grupa 2 /wejście / wyjście/ - posiada jeden poziom do obsługi przerwania od urządzeń WE / WY oraz jeden do obsługi przerwania od pulpitu operatora. Może być blokowana przez 38 bit słowa stanu programu.
- Grupy 3 + 15 /przerwania zewnętrzne/ - system IRIS-80 akceptuje aż 13 grup przerwania zewnętrznych jako opcję, z których każda posiada 16 poziomów. Cały ten zbiór może być blokowany przez 39 bit słowa stanu programu.

Grupy 0,1 i 2 należą do tzw. przerwania wewnętrznych. Są to sytuacje wyjątkowe, które mogą wystąpić w czasie realizacji programu. Obsługa ich jest taka jak opisano powyżej. Powodują one zawieszenie /a nie przerwanie/ wykonywanego programu. Przyczynami powodującymi powstanie przerwania wewnętrznych będą:

- Nadmiar podczas wykonywania operacji arytmetycznych stało i zmiennie-przecinkowych oraz dziesiętnych.
- Nielegalny rozkaz, którego kod nie odpowiada żadnej z dozwolonych instrukcji.
- Błąd licznika czasu działania programu.
- Błąd informacji wykryty w czasie przetwarzania np. liczba niedziesiętna w danych dla operacji dziesiętnych.

Jak więc widzimy przerwania zewnętrzne powstają niezależnie od wykonywanego programu, natomiast wewnętrzne są ściśle związane z jego realizacją.

#### 1.2.4. Lista rozkazów IRIS-80

Jednostka centralna IRIS-80 akceptuje rozkazy dwóch typów:

1/ typowe tylko dla IRIS-80.

2/ kompatybilne z 10 070.

Pierwszy typ używany jest w trybie pracy C, zaś drugi w trybie pracy A i B.

Lista rozkazów IRIS-80 zawiera 112 rozkazów, z których 14 są to rozkazy uprzywilejowane, tzn. że mogą one być wykonywane tylko w specjalnym trybie pracy JC w tzw. "trybie uprzywilejowanym". JC może przejść do tego trybu tylko z powodu przerwania. Są to instrukcje zarezerwowane wyłącznie dla systemu operacyjnego i użytkownik /programista/ nie ma do nich dostępu. Oprócz systemu wykorzystywane one są też przez konserwatorów sprzętu podczas testowania, gdyż umożliwiają dostęp do obszarów /np. PAO/ normalnie niedostępnych. Do tych operacji należą:

RD; WD; SIO; TIO; AIO; HIO; TDV; RIO - oraz

LPSD i XPSD - dla słowa stanu programu i

PSS i PLS - do działania na stosach kontekstów.

Pozostałe rozkazy, w zależności od rodzaju wykonywanej przez nie operacji dzielimy na następujące grupy:

##### 1.2.4.1. - Pobierania i pamiętania

Grupa ta zawiera 28 instrukcji, które umożliwiają przesyłanie informacji między rejestrami a pamięcią, wpisywanie do rejestrów wartości logicznych lub numerycznych oraz ładowania wskaźników. Do grupy tej należą: LI; LB; LH; LW; LD; LCH; LAH; LCW; LAW; LCD; LAD; LS; LM; LAS; LRA; LYAW; LIAI; LCFI; LCF; LBR; XW; STB; STH; STW; STD; STS; STM; STCF.

Umożliwiają one operacje na bajtach, półsłowach, słowach i słowach podwójnych.

##### 1.2.4.2. - Arytmetyka stałoprzecinkowa

Do grupy tej należy 16 rozkazów wykonujących operacje binarne dodawania, odejmowania, mnożenia i dzielenia na operandach całkowitych, z których jeden będzie pobierany z rejestru ogólnego a drugi bądź bezpośrednio z instrukcji bądź z rejestru ogólnego, bądź z PAO. Dla większości tych instrukcji oba operandy umieszczone są w rejestrach. Ułatwia to podnoszenie do kwadratu lub zerowanie. Operują one na półsłowach, słowach i słowach podwójnych. Wszystkie zaś ustawiają, w zależności od wyników, wskaźniki warunków. Do grupy tej należą: AI; AH; AW; AD; SH; SW; SD; MI; MH; MW; DH; DW; AMW; MTB; MTH; MTW.

#### 1.2.4.3. - Porównania

Do grupy tej należy 8 instrukcji, które porównują zawartość rejestru z argumentem z PAO bez zmiany zawartości obu operandów. Wynik porównania odzwierciedlony jest we wskaźnikach warunków. Umożliwiają one operacje na bajtach, półsłowach, słowach, słowach podwójnych oraz na bitach. Do grupy tej należą: CI; CB; CH; CW; CD; CS; CLR; CLM.

#### 1.2.4.4. - Operacje logiczne

3 rozkazy umożliwiające wykonanie operacji boolowskich. Wykonywane one są bit po bicie na dwóch operandach, z których jeden jest w rejestrze R a drugi w PAO. Wynik tych operacji znajduje się w rejestrze R. Do grupy tej należą: OR; EOR; AND; One również na podstawie wyniku ustawiają wskaźniki warunków i działają na słowach.

#### 1.2.4.5. - Przesunięcia

Należą tu instrukcje S i SF, które umożliwiają przesunięcie rejestru lub rejestrów w lewo lub w prawo na trzy sposoby:

- 1/ logicznie - zawartość rejestru lub rejestrów jest przesuwana w lewo lub w prawo o zadaną ilość pozycji. Bity wychodzące poza rejestr są gubione a na miejsca zwolnione wchodzi zera.
- 2/ arytmetycznie - zawartość rejestru lub rejestrów jest przesuwana. Jeśli przesunięcie jest w lewo na miejsca wolne wchodzi zera, jeśli zaś w prawo to na zwalnianych pozycjach powielany jest bit zerowy /znak/. Bity wychodzące poza rejestr są gubione.
- 3/ cyklicznie - w tym przypadku bity, które wychodzą poza rejestr są wprowadzane na miejsca zwalniane.

Określenie rodzaju przesunięcia, jego wielkości oraz kierunku odbywa się przez odpowiednie ustawienie bitów 21 + 23 i 25 + 31 części adresowej rozkazu zgodnie z poniższą tabelą:

BITY			Typ przesunięcia
21	22	23	
0	0	0	przesunięcie logiczne krótkie /32 bity/
0	0	1	" " długie /64 bity/
0	1	0	" cykliczne krótkie
0	1	1	" " długie
1	0	0	" arytmetyczne krótkie
1	0	1	" " długie
1	1	0	} nieokreślone
1	1	1	

Bity 25 + 31 są traktowane jako liczba algebraiczna stała, dla której bit 25 będzie znakiem. Znak ten będzie decydował o kierunku przesunięcia a zawartość bitów 26 + 31 określa ilość pozycji do przesunięcia. Liczba ujemna jest przedstawiona w postaci uzupełnienia do dwóch. Liczba dodatnia powoduje przesunięcie w lewo, liczba ujemna w prawo. Wartość ta może być zawarta:

$$-64 \leq N \leq +63$$

Dla przesunięć długich pole R powinno być parzyste. W przypadku nieparzystej wartości nastąpi dwukrotne przesunięcie rejestru R. Dla liczb zmiennoprzecinkowych następuje podział liczby na cechę i mantysę. Przesuwana jest mantysa a cecha zwiększana lub zmniejszana odpowiednio do ilości pozycji przesunięcia. Znak liczby po przesunięciu jest taki sam jak i przed przesunięciem.

#### 1.2.4.6. - Konwersja

Są dwie instrukcje konwersji CVA i CVS - pozwalające przechodzić z dowolnego kodu na binarny lub odwrotnie. Każda z tych instrukcji operuje na słowach. W rejestrze R umieszczony jest kod binarny a w Ru 1 kod do konwersji. Adres instrukcji jest adresem 32 bitowej tablicy kodu zawierającego wagi poszczególnych pozycji. Ponieważ są to operacje długie mogą one być przerywane bez ładowania zawartości rejestrów R i Ru 1. Przerywanie jest dopuszczalne do 16-ej iteracji, powyżej nie jest przerywana aż do zakończenia.

#### 1.2.4.7. - Arytmetyka zmiennoprzecinkowa

Grupa ta zawiera 8 instrukcji pozwalających wykonać operacje na liczbach zmiennoprzecinkowych. Należą do nich: FAS; FAL; FSS; FSL; FMS; FML; FDS; FDL.

Rozróżnia się dwa rodzaje liczb zmiennoprzecinkowych:

- liczby zmp. krótkie,
- liczby zmp. długie.

Liczba zmp. krótka zawiera znak /bit 0/, 7 bitową cechę /bity 1 + 7/ oraz sześć cyfrową mantysę /bity 8 + 31/.

Liczba zmp. długa podobna jest do krótkiej i jest rozszerzona o 8 cyfrową mantysę. W obu przypadkach cyfry cechy i mantysy przedstawione są w kodzie szesnastkowym. Dla operacji zmp. istnieją cztery wskaźniki trybu działania. Są to:

- AF - zaokrąglenie zmp.
- PF - precyzja zmp.
- ZF - zero zmp.
- NF - normalizacja zmp.

Zawarte są one na 4, 5, 6 i 7 bicie słowa stanu programu i ustalają tryb działania zmp. Można je ustawiać przy pomocy rozkazów. Wskaźnik AF wykorzystywany jest tylko w trybach pracy B i C.

#### 1.2.4.8. - Arytmetyka dziesiętna

Wszystkie operacje dziesiętne operują na liczbach dziesiętnych upakowanych, zawierających od 1 + 31 cyfr dziesiętnych + znak liczby dziesiętnej, przedstawiony w postaci kodu: 1010 /X'A'/; 1100 /X'C'/; 1110 /X'E'/; 1111 /X'F'/ - oznaczającego liczbę dodatnią oraz w postaci 1011 /X'B'/ i 1101 /X'D'/ - co oznacza liczbę ujemną. Przyjęta w systemie IRIS-80 postać znaku wyniku operacji: 1100 /X'C'/ dla liczb dodatnich i 1101 /X'D'/ dla liczb ujemnych. Do grupy tej należy 10 instrukcji, z których cztery są operacjami arytmetycznymi, dwie służą do przesyłania między akumulatorami dziesiętnymi a pamięcią operacyjną, porównania i przesunięcia. Dwie instrukcje pozwalają na rozpakowanie /umieszczenie jednej cyfry dziesiętnej na bitach 4 + 7 bajtu/ tzn. umieszczenie jednej cyfry na bajt, oraz na upakowanie - tzn. umieszczenie dwóch cyfr dziesiętnych w jednym bajcie. Operacje te ułatwiają wydawnictwo.

Przed wykonaniem każdego rozkazu liczby dziesiętne sprawdzane są na obecność znaków niedziesiętnych tzn. znaków zawar-

tych między X'A'a X'F; z wyjątkiem pozycji znaku liczby. Wszystkie te operacje ustawiają wskaźniki warunków. Adresowanie jest bajtowe. Rozkazy mnożenia i dzielenia dziesiętnego mogą być przerywane w trakcie wykonywania.

#### 1.2.4.9. - Przetwarzanie ciągu bajtów

Jest pięć rozkazów operujących na ciągach bajtów: MBS; CBS; TBS; TTBS i EBS. Adresowane są bajtowo bezpośrednio i operują z PAO do PAO. Przetwarzają ciągi znaków alfanumerycznych występujące w systemach zarządzania. Pozwalają przenieść ciąg z jednego miejsca PAO do drugiego, porównać ze sobą dwa ciągi, dekodować kolejne bajty jednego z ciągów zgodnie z tabelą oraz przeprowadzać wydawnictwo zgodnie z przyjętym wzorcem. Wszystkie rozkazy /z wyjątkiem EBS/ ze względu na długi czas wykonywania mogą być przerywane.

#### 1.2.4.10. - Operacje na stosach

Operacje te są obsługiwane przez 5 rozkazów: PSW; PLW; PSM; PLM i MSP, adresowanych dwusłowowo. Wykorzystywane są w wielu typach przetwarzania a w szczególności przy tłumaczeniu języków. Każdy ze stosów zdefiniowany jest przez tzw. "wskaźnik stosu" zajmujący dwa słowa maszynowe i zawierający aktualny stan przetwarzanego stosu tzn.

- adres wierzchołka stosu,
- długość stosu w słowach,
- ilość słów pozostałych do wypełnienia.

Każde przesłanie do stosu lub pobranie z niego powoduje automatycznie aktualizację wskaźnika którego adres znajduje się w części adresowej rozkazu.

#### 1.2.4.11. - Rozkazy skokowe

Jest to grupa rozkazów umożliwiająca warunkową lub bezwarunkową zmianę sekwencji programu. Dla skoków bezwarunkowych następną wykonywaną instrukcją będzie instrukcja, której adres jest zawarty w części adresowej rozkazu skoku. Dla skoków warunkowych skok będzie efektywny jeśli narzucony warunek zostanie spełniony, w przeciwnym zaś przypadku zostanie wykonany rozkaz następny po rozkazie skoku. Skok ze śladem umożliwia wyskok i powrót do wykonywanego programu. Dla grupy tej, przy programowaniu w assemblerze stosowane są tzw. "pseudokody", na podstawie których translator assemblera ustawia odpowiednie warunki skoku. Właściwość ta

upraszcza pracę programisty. Do grupy tej należy 6 rozkazów: EXU; BCS; BCR; BIR; BDR; BAL. Wskaźniki warunków ustawiane są przez następnym wykonywany rozkaz.

#### 1.2.4.12. - Monitorowanie

Każdy z czterech rozkazów tej grupy: CAL 1; CAL 2; CAL 3; i CAL 4 powoduje automatyczne odwołanie się pod odpowiedni adres pamięci operacyjnej: X'48'; X'49'; X'4A'; X'4B' w celu obsługi przerwania, które powoduje ten rozkaz. Oznacza to, że sterowanie jest przekazywane do systemu operacyjnego. Zazwyczaj każdy z tych rozkazów powoduje wykonanie rozkazu XPSD /ładuj słowo stanu programu/ którego strefa R umożliwia rozróżnienie 16 różnych sposobów monitorowania. Oznacza to, że grupa ta umożliwia wykonanie 64 różnych podprogramów. Adresowanie jest słowowe. Rozkazy te nie ustawiają wskaźników warunków.

#### 1.2.4.13. - Rozkazy uprzywilejowane

W grupie tej rozróżniamy dwie podgrupy:

- 1/ Rozkazy dotyczące systemu operacyjnego i procesora
- 2/ Rozkazy dotyczące urządzeń WE / WY.

Do pierwszej podgrupy należą: PSS; PLS; LPSD; XPSD; LRP; LSTP; MMC; WAIT; RD i WD. Rozkazy te nie mogą być wykonywane w trybie normalnym, a tylko w trybie uprzywilejowanym. Próby wykonania w trybie normalnym powodują zawieszenie i skok do komórki X'40'. Rozkazy te są zarezerwowane dla systemu i umożliwiają mu zarządzanie przetwarzaniem przez możliwość wymiany kontekstu, zmianę słowa stanu programu, zmianę rejestrów kluczy i blokady PAO. Rozkaz MMC może być przerywany w trakcie wykonywania. Rozkaz RD pozwala wczytać program z urządzenia zewnętrznego do PAO /np. binarny, wczytywanie systemu z dysków itp./.

Do drugiej podgrupy należy pięć rozkazów: SIO; HIO; RIO; TIO; TDV i AIO. Są one uprzywilejowanymi i w trybie normalnym traktowane przez system jako nielegalne. System wykorzystuje je do sterowania transmisją do lub z urządzeń zewnętrznych, poprzez uruchamianie transmisji, zatrzymywanie jej, sprawdzanie stanu kanału lub urządzenia zewnętrznego.

Część adresowa rozkazu nie jest adresem komórki PAO lecz numerem kanału i urządzenia w tym kanale. Dla rozkazów SIO; HIO; TIO i TDV jest to 12 ostatnich bitów adresu. Wszystkie te rozkazy powodują ustawienie wskaźników warunków w zale-

żności od rodzaju odpowiedzi UZ. Całą grupą rozkazów uprzywilejowanych posługują się konserwatorzy sprzętu, w szczególności przy testach. Istnieje wtedy potrzeba wejścia w obszary systemu normalnie niedostępne.

### 1.3. URZĄDZENIA WEJŚCIA / WYJŚCIA

W systemie IRIS-80 wszystkie operacje wejścia / wyjścia wykonywane są pod kontrolą jednostek kanałowych. Pozwala to uwolnić jednostkę centralną od szczegółów operacji WE / WY, która w tym czasie może poświęcić się wykonywaniu programu. Każda z jednostek kanałowych jest jak gdyby specjalizowanym procesorem pośredniczącym pomiędzy PAO a dołączonymi do niego urządzeniami zewnętrznymi. Istnieją dwa typy jednostek kanałowych:

- Jednostki kanałowe bezpośrednie - posiadające jeden tor przesyłowy i mogące pracować z dużą prędkością.
- Jednostki kanałowe multiplexerowe - posiadające jeden multiplexerowy tor przesyłowy do którego można dołączyć kilka urządzeń zewnętrznych wolnych i dwa tory bezpośrednie.

#### 1.3.1. Jednostki kanałowe bezpośrednie /JKB/

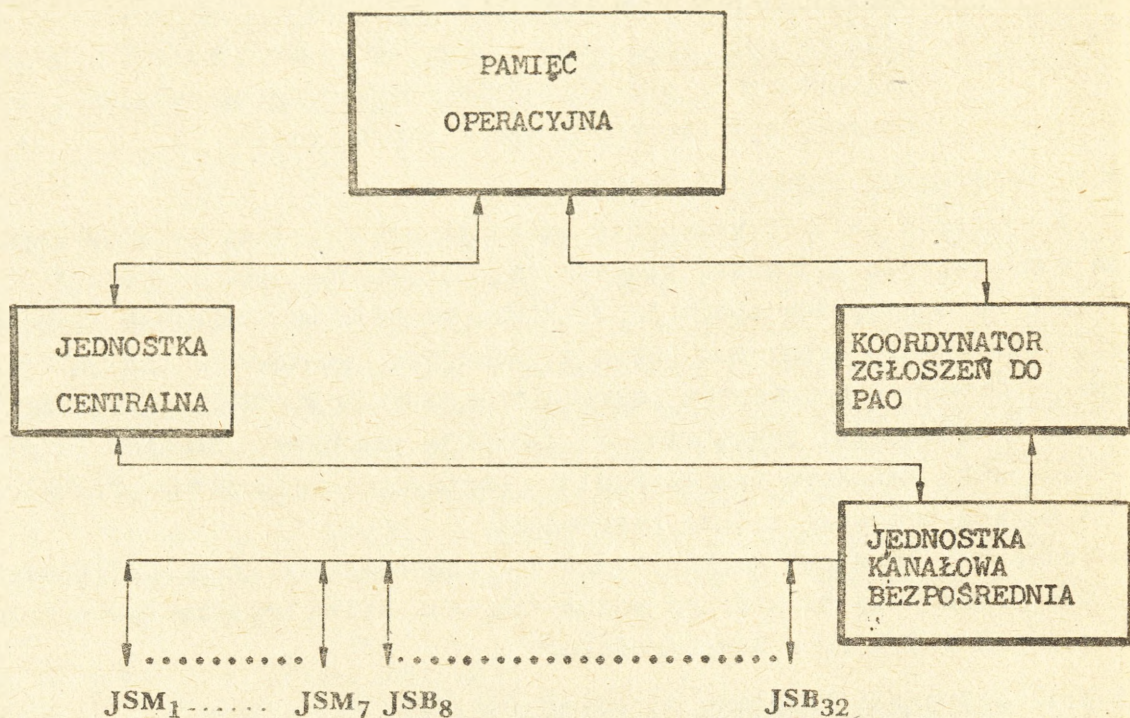
Są to właściwie specjalizowane komputery, służące do obsługi transmisji z bardzo szybkich urządzeń zewnętrznych takich jak np. dyski magnetyczne. Posiadają one własną pamięć operacyjną, do której wprowadzany jest program obsługi kanału. Program ten jest przygotowywany przez jednostkę centralną inicjującą operacje WE / WY. Po zainicjowaniu operacji WE / WY dalsza praca kanału odbywa się bez udziału jednostki centralnej. Całe sterowanie przesyłaniem danych z UZ do jednej lub kilku jednostek PAO przejmują jednostki kanałowe, sygnalizując jedynie do JC sytuacje awaryjne i zakończenie transmisji WE / WY.

JKB steruje interfejsem kanałowym 32 bitowym i może przesyłać 1, 2 lub 4 bajty jednocześnie do lub z UZ do niej dołączonych. Pozwala to teoretycznie osiągnąć prędkość transmisji:

- 1,25 Mbajta/S dla przesyłania 1-bajtowego,
- 3,5 Mbajta/S dla przesyłania 4-bajtowego,

Jeśli do toru tego będą dołączone dyski - szybkość transmisji będzie 2,5 Mbajta/S.

Przez JKB możemy dołączyć do 32 Jednostek Sterujących U.Z., z których 8 powinno być Jednostkami Multiplexerowymi. Interfejs łączący JKB z JS jest również interfejsem słowowym umożliwiającym transmisję bądź bajtową bądź słowową. Ponieważ JKB posiada tylko jedną pamięć kanału, współpracować z nią w danej chwili może tylko jedno UZ. Sposób dołączenia JKB i miejsce jej w systemie IRIS-80 obrazuje poniższy rysunek:



Dodatkowy moduł koordynatora dostępu do PAO pozwala na zwiększenie ilości JK podłączonych do PAO. Np. koordynator typu 2777 pozwala na dołączenie dwóch jednostek kanałowych dowolnego typu: albo 2 JKM albo 2 JKB albo 1 JKM i 1 JKB.

### 1.3.2. Jednostki kanałowe multiplexerowe /JKM/

W rzeczywistości jednostka ta zawiera dwa typy kanałów multiplexerowy i bezpośredni. Ponieważ JKM posiada tylko tor bajtowy, to kanał bezpośredni jest typu bajtowego. Teoretyczna szybkość kanału typu multiplexerowego wynosi 350 K bajtów/S, zaś kanał szybkiego 700 K bajtów/S. Przy jednoczesnej pracy kanałów multiplexerowego i dwóch szybkich prędkość całego toru będzie:

- do 250 K bajtów/S dla kanału multiplexerowego,
- do 300 K bajtów/S dla kanału szybkiego B,
- do 500 K bajtów/S dla kanału szybkiego A.

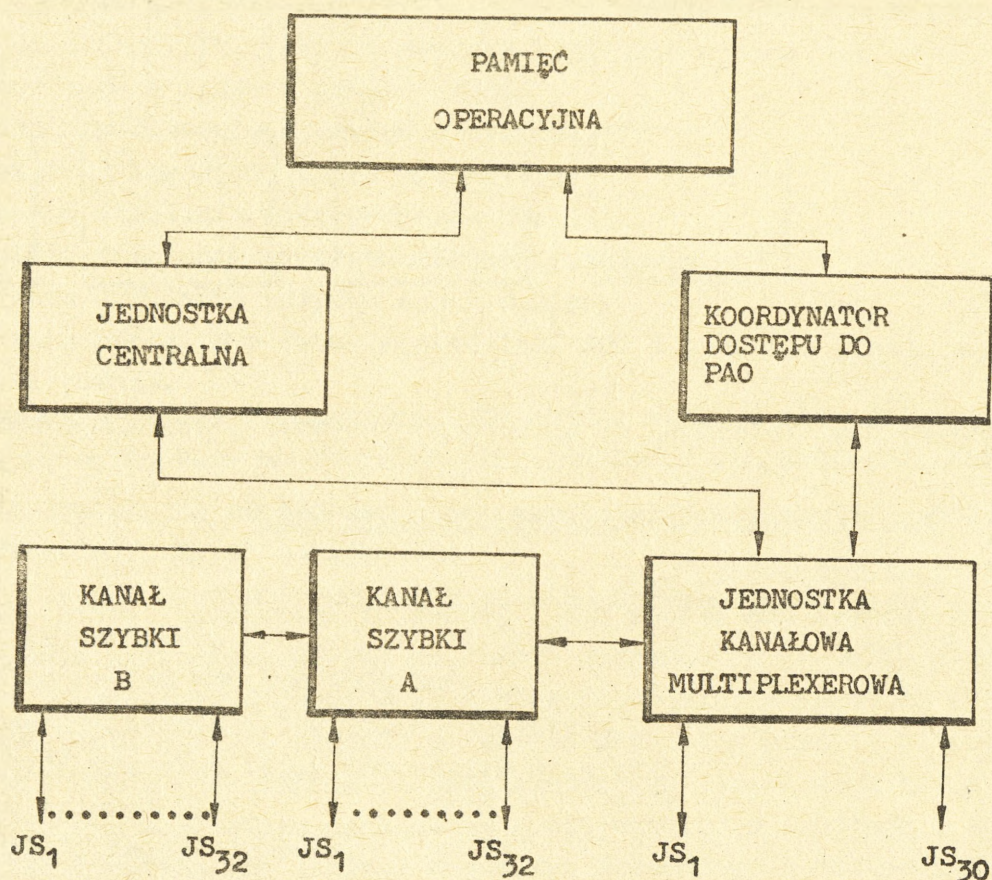
Do JKM można dołączyć do 30 JS mogących pracować jednocześnie, natomiast do szybkich po 32 JS. Oznacza to, że sumarycznie JKM może obsłużyć aż 94 urządzenia zewnętrzne.

Równoczesna praca urządzeń w kanale multiplexerowym jest możliwa dzięki zastosowaniu buforów dla każdego podkanału /30 UZ/. Oprócz tego każdy podkanał posiada przeznaczoną do jego obsługi tzw. "pamięć kanału".

w której przechowywane są informacje dotyczące obsługi transmisji takie jak:

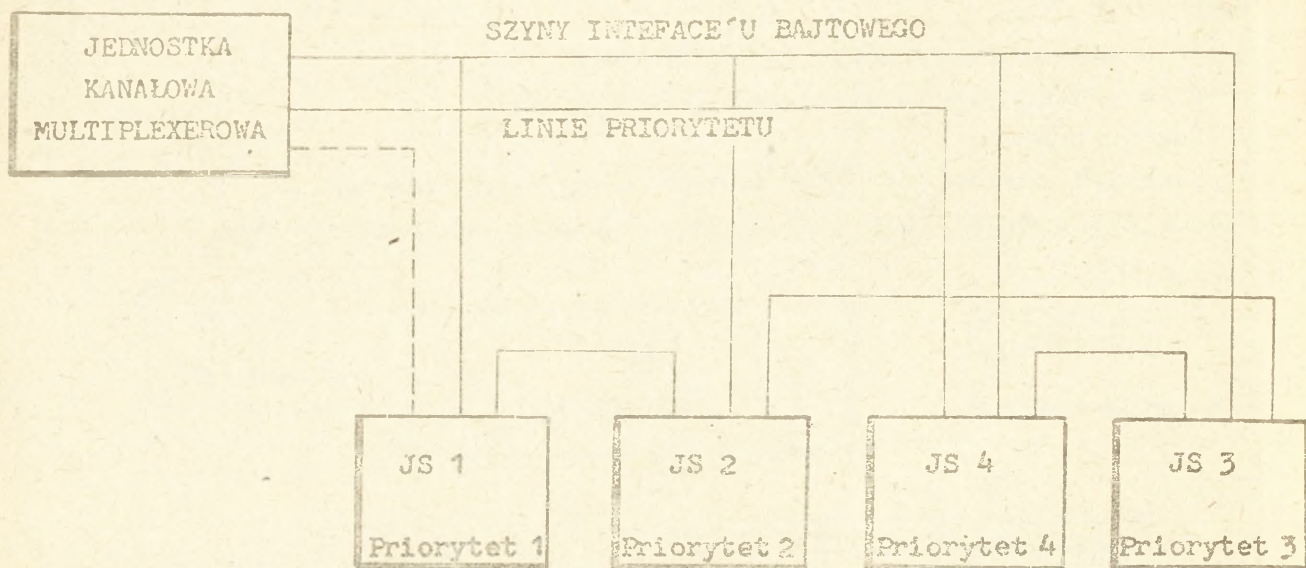
- adres bajtu przesyłanego z lub do PAO,
- adres podwójnego słowa rozkazowego kanału,
- licznik bajtów,
- wskaźniki charakteryzujące sposób wykonywania transmisji.

Sposób dołączenia JKM i jej miejsce pokazuje poniższy rysunek:



Do JKM może być dołączony adapter stacji zewnętrznych /końcówek/ typu multiplexerowego, co umożliwia jednoczesne sterowanie wszystkimi podłączonymi do niego końcówkami. Zapewnia on teoretyczną szybkość transmisji 4 Mbajty/sekundę.

Wszystkie JS dołączone do JKM posiadają swój priorytet, który uzyskiwany jest przez odpowiednie połączenia kablowe między JKM a JS. Zasadę ustalania priorytetu poszczególnych JS obrazuje poniższy rysunek:



Układ priorytetu nie określa adresu JS, lecz ustala kolejność obsługi w przypadku jednoczesnego zgłoszenia się dwóch JS do obsługi przez JKM.

### 1.3.3. Jednostki sterujące i urządzenia zewnętrzne

#### 1.3.3.1. System dyskowy

System dyskowy SDA składa się z jednostki sterującej, którą jest minikomputer MITRA-15 i jednostek dyskowych MD-100. Jednostka sterująca umożliwia sterowanie od 2 do 16 jednostek dyskowych wymiennych, która po odebraniu rozkazu przejmuje całkowitą obsługę dialogu DYSKI-PAO zwalniając z tej funkcji JC. Wyposażona jest dodatkowo we FLOPPY DISC o pojemności 5 Mbajtów, z którego można wprowadzać system operacyjny JS lub testy. W stosunku do jednostek dyskowych spełnia ona następujące funkcje:

- zarządza dołączeniem odpowiedniej JD do pracy z PAO;
- automatycznie koryguje błędy przy pomocy kodów samokorygujących;
- automatycznie restartuje operację WE / WY w przypadku błędu.

Z pamięcią operacyjną pracuje poprzez adapter dopasowujący logikę PAO do logiki MITRA-15.

Jednostki dyskowe /JD/ - są urządzeniami o bardzo dobrej jakości i posiadają następujące parametry:

- pojemność pakietu dyskowego - 89 Mbajtów,
- średni czas dostępu kątownego - 8,33 ms,
- średni czas dostępu promieniowego - 30 ms,
- szybkość transmisji - 800 Kbajtów/S,
- jednostka adresowania: sektor = 1024 bajty.

Niedogodnością w całym systemie dyskowym jest brak możliwości inicjowania u użytkownika pakietów dyskowych. Musi się to odbywać u producenta na specjalnym przyrządzie, ze względu na dużą gęstość zarówno zapisu, jak i ilości ścieżek.

#### 1.3.3.2. Pamięci taśmowe

Jednostka sterująca pamięciami taśmowymi zapewnia współpracę pomiędzy JC a przewijakami taśmy magnetycznej, poprzez kanał bezpośredniego dostępu. Jest to podobnie, jak i przy systemie dyskowym MITRA-15, z systemem służącym do obsługi TM, który spełnia następujące funkcje:

- zapewnia połączenie z kanałem KBD,
- zapewnia połączenie z pamięcią taśmową,
- zapewnia pracę linii zapis - odczyt,
- zapewnia działanie mikroprogramu obsługi,
- zapewnia działanie programu diagnostycznego,
- zapewnia działanie pulpitu technicznego,
- zapewnia pracę TM z dwoma gęstościami 800 i 1600 bpi przełączanymi automatycznie,
- zapewnia możliwość adresowania od 8 do 16 jednostek taśm magnetycznych.

Z pamięcią operacyjną współpracuje poprzez interface słowowy tzn. 32 bitowy w systemie 1, 2 lub 4-bajtowym.

Do JS dołączonych jest równoległe od 4 do 8 przewijaków taśmy magnetycznej o parametrach:

- gęstość zapisu 800 lub 1600 bpi,
- szybkość transmisji 320/160 000 bajtów/S,
- szybkość przesuwu taśmy 200 cali/S /5 m/S/,
- ilość ścieżek 9,
- szerokość taśmy 1/2 cala,
- zapis z modulacją fazy /NRE/,
- czas przewijania krążka taśmy 45 sek.

Taśmy zainstalowane w systemie IRIS-80 mają mocno rozbudowany system sterowania pneumatycznego, oraz automatyczne ładowanie taśmy. Wystarczy, że operator założy krążek taśmy na prawy napęd a wprowadzenie taśmy pod głowice, do kieszeni, nawinięcie na lewą szpulę i ustawienie na znaczniku początku odbywa się samoczynnie.

#### 1.3.4. Urządzenia zewnętrzne elektromechaniczne

Są to urządzenia bardzo wolne, pracujące bajtowo i w systemie REMOTE BATCH. Są one podłączone do kanału multiplexerowego równoległe. Do urządzeń tych należą:

- czytnik kart LC-1200 - typ 70140

- perforator kart PC-300 - typ 70160
- drukarka wierszowa IP-1200 - typ 72444
- czytnik - perforator taśmy papierowej LPR
- konsola operatora

#### 1.3.4.1. Czytnik kart LC-1200

Urządzenie elektromechaniczne, służące do wprowadzenia informacji z 80-kolumnowych kart perforowanych o następujących parametrach:

- szybkość czytania - 1200 kart/min,
- system odczytu - fotooptyczny /jeden oświetlacz z systemem światłowodów/,
- podawanie kart - pneumatyczne,
- kontrola poprawności informacji przez ponowne wczytanie karty,
- kod - binarny EBCDIC
- technika germanowa - firmy CDC.

#### 1.3.4.2. Perforator kart PC-300

Urządzenie to służy do wyprowadzania informacji z systemu IRIS-80 na 80 kolumnowe karty perforowane. Przydatne w przypadkach, gdy dane wyjściowe z jednego programu są danymi wejściowymi dla drugiego, szczególnie gdy on jest przetwarzany w innym terminie lub w innym Ośrodku Obliczeniowym. Jest to urządzenie firmy BOOLL, zbudowane w technice mieszanej DTL i TTL. Posiada parametry:

- szybkość perforacji 300 kart/min,
- kontrola - przez odczyt szczotkowy i porównanie,
- kody - binarny EBCDIC

#### 1.3.4.3. Drukarka wierszowa LP-1200

Jest urządzeniem wyjściowym służącym do wyprowadzania rezultatów przetwarzania w postaci tabulogramów. Zbudowana w technice mieszanej DTL i TTL. Rozwiązania mechaniczne ułatwiają pracę operatora i konserwatora. Posiada następujące parametry:

- szybkość druku max. 1200 wierszy/min,
- szybkość druku min. 875 wierszy/min,
- 132 znaki w wierszu,
- 64 różnorodne znaki pisarskie,
- regulacja młotków elektroniczna.

Model ten był specjalnie opracowany dla systemu IRIS-80.

#### 1.3.4.4. Czytnik - perforator taśmy papierowej

Jest to zestaw trzech urządzeń elektromechanicznych w jednym module:

- czytnik taśmy papierowej,
- perforator taśmy papierowej,
- przewijarka taśmy papierowej.

Czytnik taśmy o odczycie fotoelektrycznym z jedną żarówką centralną pozwala czytać taśmę 5, 6, 7 i 8-ścieżkową z szybkością 300 znaków/S  $\pm 20$  znaków przy pracy ciągłej i z szybkością 100 znaków/S przy pracy start-stopowej. Posiada przełącznik oświetlenia ze względu na gęstość taśmy. Perforator pracuje z prędkością 55 znaków/S. Można do niego stosować taśmę papierową lub typu MYLLAR.

Przewijarka taśmy papierowej o pojemności 300 mb. taśmy. Zapewnia przewinięcie całych 300 m w czasie krótszym niż 50 sekund.

#### 1.3.5. Konsola operatora

Jest to niezależna jednostka umożliwiająca dialog operatora z systemem operacyjnym i składa się z:

- maszyny do pisania TELETYPE pracującej z prędkością 35 znaków/min,
- 2 alfaskopów IS-300 bez klawiatury,
- części technicznej pulpitu z przyciskami.

Podłączona jest do systemu poprzez kanał tzw. pseudo CLIO /Jednostka sterująca transmisją międzykomputerową/. W systemie IRIS-80 oznaczona jako PED /pulpit sterowania z odległości/. Posiada także zegar czasu rzeczywistego.

Pierwszy z alfaskopów wyświetla dane o stanie systemu i o programach. Są to:

- ilość pamięci operacyjnej /w stronach/ dostępne dla użytkownika,
- czas,
- ilość przetwarzanych programów, czas pracy programu, ilość UZ przydzielonych do programu.

Drugi używany jest do kontroli informacji wychodzących z maszyny. Są to informacje typu: - ile stron pozostało do wydruku na DW, ilość kart do perforowania na PK, którą taśmę założyć na który przewijak, lub który pakiet dyskowy na który napęd itp.

Pojemność ekranu każdego z nich wynosi 1008 znaków, po 85 znaków w linii.

#### 1.4. JEDNOSTKA TELETRANSMISJI

Zapewnia współpracę z jednej strony z kanałem multiplexerowym a z drugiej z liniami zdalnego przetwarzania wsadowego /REMOTE BATCH/ lub z podziałem czasu /TIME SHARING/, umożliwiając korzystanie z bazy danych dużego komputera bez potrzeby rozbudowy własnego zestawu. Oba te sposoby wymiany informacji są stosowane w systemie IRIS-80 w oparciu o minikomputery MITRA-15, modemy i terminale /np. IRISCOPE 200/.

System IRIS-80 został wyposażony w procedury teletransmisji:

- TMM - VU - procedura wizualna - pracująca w kodzie ASC, realizując polting i selekting. Transmisja odbywa się blokami tylko

w jednym kierunku. Realizuje połączenia dwu i wielopunktowe. Jednostka sterująca dla tej procedury jest jednostką bierną - tzn. że pracuje całkowicie pod nadzorem komputera.

- TMM - RB - /remote batch/ - procedura przetwarzania wsadowego na odległość. Terminalem w tym przypadku jest minikomputer MITRA-15 posiadający system operacyjny MTR RB lub MTRD RB /dyskowy/. W tym przypadku możliwa jest transmisja poprzez linie telefoniczne na odległość do 15 m bez modemów, na dalsze poprzez modemy. Dopuszcza się pracę alternatywną lub w dupleksie integralnym blokami o stałej długości.

- TMM - IC - zapewnia wymianę informacji w dwóch typach połączeń:

- UC<sub>1</sub> - CTQM ————— CTQM - UC<sub>2</sub>

- UC<sub>1</sub> - CLIØ ————— CLIØ - UC<sub>2</sub>

W przypadku transmisji - stacją nadrzędną jest ta, która rozpocznie transmisję.

Do jednostki sterującej możemy dołączyć od 8 + 64 linii asynchronicznych pogrupowanych w bloki po 8 linii. Może ona pracować z linią telegraficzną i telefoniczną z szybkością od 50 do 1200 bodów kodami 5 i 8 bitowymi. Transmisja może się odbywać simpleksowo, półdupleksowo lub w pełnym dupleksie. Jako terminale służą dostarczone przez firmę CII monitory ekranowe typu IRISCOPE-200 z klawiaturami z podłączonymi do nich termo-kopiarkami lub IRISCOPE-300.

#### 1.4.1. Terminale lekkie

##### 1.4.1.1.

###### - IRISCOPE-200

Są to monitory ekranowe typu telewizyjnego z klawiaturą. Posiadają ekran o pojemności 8 linii po 80 znaków, z możliwością rozszerzenia do 16 linii. Transmisja odbywa się z prędkością 110, 220 lub 800 bodów przez linię telefoniczną w pół lub pełnym dupleksie. Wyświetlać można zestaw 96 znaków. Własnością opcjonalną jest drukarka termiczna FCIS30 pracująca z prędkością 30 znaków/sekundę i długością wiersza 80 znaków. Może być ona dołączona do jednego z 8 IRISCOPE-200 w celu przekopiowania ekranu na papier.

##### 1.4.1.2.

###### - IRISCOPE-300

Są to monitory ekranowe typu telewizyjnego z klawiaturą o pojemności ekranu 8 lub 16 linii 80 znakowych. Transmisja odbywa się blokowo po specjalnie wydzielonych liniach telefonicznych z prędkością 1200, 2400

lub 4800 bodów. Monitory te charakteryzują się następującymi parametrami:

- możliwość generowania 100 symboli graficznych /duże i małe litery/,
- posiadają dwa rodzaje 80 znakowej klawiatury alfanumerycznej,
- duże pole wyświetlania umożliwia uzyskanie czystego i wyraźnego obrazu,
- możliwość wyświetlania negatywu obrazu tzw. video inversion,
- możliwość adresowania na poziomie znaku,
- możliwość kopiowania ekranu na drukarce,
- możliwość rozszerzenia repertuaru znaków od 640 do 1280,
- możliwość stosowania tabulacji i ochrony obszaru,
- możliwość kasowania i wstawiania znaków wewnątrz linii.

#### 1.4.2. Terminale ciężkie

Do tego rodzaju terminali będą należały:

- MITRA-15/RB
- MITRA-15/VG
- MITRA-15/CD

Są to minikomputery przeznaczone do pracy w czasie rzeczywistym i mogące mieć różnorodne zastosowanie. Posiadają one własną pamięć operacyjną o pojemności od 4 + 32 Ksłów, 16 bitowych oraz pamięć mikroprogramowaną o pojemności 512 lub 1024 słów, od zawartości której zależy czy to jest JC czy JS czy specjalnie adaptowana do nietypowych zastosowań. Każdy z minikomputerów MITRA-15 posiada swój wewnętrzny interfejs, przez który dołączony jest do JC odpowiedni dla spełnionej funkcji zestaw urządzeń zewnętrznych. Wykonuje 87 rozkazów z których:

- 40 - rozkazów współpracy z PAO,
- 29 - rozkazów operacji na rejestrach,
- 12 - rozkazów operacji dziesiętnych,
- 6 - rozkazów specjalnych.

Wszystkie rozkazy są tego samego formatu i mają postać:

TRYB	FUNKCJA	PRZESUNIĘCIE
0	2 3	7 8 15

TRYB	FUNKCJA	PRZESUNIĘCIE
0	3 4	7 8 15

Format ten jest aktualny dla bajtów, dla słów, dla słów podwójnych oraz dla łańcucha znaków o dowolnej długości.

Stosowane są następujące typy adresowania:

- adresowanie bezpośrednio dla operandów bajtowych,
- adresowanie pośrednie i bezpośrednio indeksowane z wykorzystaniem rejestru bazowego lokalnego,
- adresowanie pośrednie i bezpośrednio indeksowane z wykorzystaniem rejestru bazowego ogólnego.

Możliwość programowania w językach:

- ASSEMBLER MITRAS I,
- ASSEMBLER rozszerzony MITRAS II,
- makro-generator,
- LP 15,
- BASIC,
- FORTRAN IV.

Posiada możliwość zastosowania następujących systemów operacyjnych:

- monitor podstawowy MOB,
- monitor czasu rzeczywistego MTR,
- monitor czasu rzeczywistego dyskowy MTRD /tylko dla MITR wyposażonych w dysk - np. MITRA-15/VG/.

Rodzaj zastosowanego systemu narzuca wielkość PAO. Poszczególne rodzaje MITRA-15 posiadają następujące parametry JC oraz urządzeń zewnętrznych.

1.4.2.1. MITRA-15/RB - z pamięcią operacyjną 32 Ksłów podłączony do IRIS -80 przez kanał CTQM za pomocą modemów MO-60 i czteroprzewodowej wydzielonej /niekomutowanej/ linii transmisji danych zapewniających pracę w trybie synchronicznym tzw. DUPLEX INTEGRALNY przy szybkości transmisji 2400 bodów. W skład tego zestawu wchodzi następujące urządzenia:

- jednostka centralna MITRA-15 z PAO = 32 Ksłów,
- monitor operatora - maszyna do pisania TELETYPE ASR 33 z czytnikiem i perforatorem taśmy papierowej i szybkości 15 zn/S,
- czytnik taśmy papierowej o szybkości 300 zn/S,
- perforator taśmy papierowej o szybkości 60 zn/min,
- czytnik kart o szybkości 300 kart/min,
- drukarka wierszowa /132 znaki w wierszu/ o szybkości 400 linii/min,
- taśma magnetyczna o gęstości 800/1600 bpi i szybkości transmisji 40 Kbajtów/S,
- pisak bębnowy XY typu BENSON 1220.

Zestaw ten został ustawiony i uruchomiony autonomicznie w WAT.

1.4.2.2. MITRA-15/VG z pamięcią operacyjną 32 Ksłów podłączony do IRIS-80 przez dwa wejścia kanału CTQM za pomocą modemów MO-30 i dwóch czteroprzewodowych, wydzielonych linii transmisji danych zapewniających pracę w trybie synchronicznym dla linii SL 01 - FULL DUPLEX i ML 01 - HALF DUPLEX z szybkością transmisji 2400 bodów. Posiada następujący zestaw UZ:

- jednostka centralna MITRA-15 z PAO = 32 Ksłów,
- monitor operatora - maszyna do pisania TELETYPE ASR-33 o szybkości 15 znaków/S z czytnikiem i perforatorem taśmy papierowej,
- czytnik taśmy papierowej o szybkości 300 zn/S,
- perforator taśmy papierowej o szybkości 60 zn/S,
- czytnik kart LC 300 o szybkości 300 kart/min,
- drukarka wierszowa IP-400 o szybkości 400 linii/min i długości wiersza 132 znaki,
- pamięć taśmowa o gęstości 800/1600 bpi i szybkości transmisji 40 Kbajtów/sek,
- pamięć dyskowa o pojemności 10 Mbajtów /jeden pakiet stały 5 Mbajtów i jeden wymienny 5 Mbajtów/ ze średnim czasem dostępu 90 m/S i prędkości transmisji 150 KBajtów/S.
- pisak bębnowy XY typu BENSON 1220,
- zestaw transmisji dla 2 linii synchronicznych FULL DUPLEX o szybkości 1200/2400 bodów do współpracy z IRIS-80,
- grafoskop kolorowy trójbarwny /czerwony, żółty i pomarańczowy/ z piórem świetlnym typu CIT ALCATEL VG 1610.

Zestaw ten został uruchomiony autonomicznie w ASG WP.

1.4.2.3. MITRA-15/CD - z pamięcią operacyjną 24 Ksłów podłączona do IRIS-80 przez kanał CLIO za pomocą modemów MO 60 i czteroprzewodowej, wydzielonej linii transmisji danych, zapewniającej pracę w trybie synchronicznym FULL DUPLEX. Koncentrator - Dyfuzor obsługuje siedem monitorów ekranowych alfanumerycznych asynchronicznych typu IRISCOPE-200 za pomocą 7 par modemów MO 20 oraz siedmiu dwu - lub czteroprzewodowych, wydzielonych /niekomutowanych/ linii transmisji danych zapewniających pracę w trybie asynchronicznym z prędkością 600/1200 bodów. Modemy MO 20 od strony MITRY-15 C/D muszą być umieszczone w oddzielnej szafie. Posiada następujący zestaw urządzeń:

- jednostka centralna MITRA-15 z PAO = 24 Ksłowa,
- monitor operatora - maszyna do pisania TELETYPE ASR-33 o szybkości 15 znaków/S z czytnikiem i perforatorem taśmy papierowej.

- czytnik kart LC-300 z szybkością 300 kart/min,
  - zestaw transmisji dla 7 linii asynchronicznych, pracujących w trybie FULL DUPLEX z prędkością 600/1200 bodów.
- System operacyjny, obsługujący transmisję do MITRA-15/CD jest wprowadzany z JC IRIS-80.

W skład systemu IRIS-80 wchodzi jeszcze urządzenia służące do przygotowania danych wejściowych. W zasadzie dostarczane są przez producenta urządzenia dziurkujące karty 80-kolumnowe typu K-212. Jest to jedno urządzenie, które wykonuje trzy funkcje:

- dziurkarki kart,
- sprawdzarki kart,
- opisywacza.

Karty dziurkowane są w kodzie EBCDIC.

Bibliografia do rozdziału 1

1. Ordinateur IRIS-80 - manuel d'utilisation.
2. Systeme IRIS-80 - manuel de fonctionnement.TOM 1 - TOM 9.
3. Lecteur de cartes - manuel de maintenance.
4. Systeme d'interruptions - manuel de fonctionnement
5. Unites de disques MD-100.
6. Coupleur BM-3000 - manuel de fonctionnement.
7. Unite centrale MITRA-15.
8. Lecteur - perforateur de bande.
9. Canal multiplexe.
10. Bandes magnetiques STC-3000 et 3200.
11. Guide de couplage MITRA-15.
12. Manuel de reference MITRA-15 tom 1.

## 2. OPROGRAMOWANIE SYSTEMU IRIS-80

System operacyjny IRIS-80 składa się z monitora, zestawu kompilatorów i generatorów programów usługowych.

Monitor steruje wszystkimi aspektami realizacji zadań i optymalizuje je, kompilatory i generatory umożliwiają użytkownikom tworzenie swoich własnych programów i tłumaczenie ich na postać możliwą do realizacji, programy usługowe ułatwiają konserwację oprogramowania i realizację operacji usługowych.

System nie jest stały, ilość i rodzaje programów mogą się zmieniać i stale wzrastać, odpowiednio do specyficznych wymagań użytkowników. Specjalny program, zwany "generatorem systemu", umożliwia adaptację i aktualizację systemu operacyjnego.

SIRIS 8 zaprojektowany został do działania w trybie wieloprogramowania, dlatego zdolny jest do jednoczesnej realizacji kilku programów w celu najbardziej efektywnego wykorzystania różnych zasobów systemu komputerowego.

### 2.1. CHARAKTERYSTYKI PRAC I ZASOBÓW

Pracą nazywamy ciąg operacji, które system realizuje dla jednego użytkownika.

Praca może być podzielona na kilka etapów, które automatycznie realizowane są jeden po drugim i przekazują sobie dane przy pomocy zbiorów stałych lub czasowych /roboczych/.

SIRIS 8 obsługuje trzy podstawowe tryby przetwarzania:

a/ przetwarzanie wsadowe, obejmujące:

- lokalne przetwarzanie wsadowe,
- zdalne przetwarzanie wsadowe,
- zarządzanie transakcjami,

b/ przetwarzanie wielodostępne /time-sharing/,

c/ przetwarzanie w czasie rzeczywistym,

Przedstawione tryby przetwarzania mają swoje własne charakterystyki.

Wymagają zatem różnych mechanizmów planowania i inicjowania oraz używają specyficznych programów usługowych.

#### 2.1.1. Przetwarzanie wsadowe

Lokalne przetwarzanie wsadowe.

Prace wprowadzane są przy pomocy jednego lub więcej czytników kart, zapamiętywane w pamięci dyskowej i wstawione do kolejek oczekiwania. Monitor zapamiętuje każdą pracę zgodnie z jej priorytetem i dostępnymi zasobami, a pracę o najwyższym w danej chwili priorytecie przesyła do jednostki centralnej. Praca może utracić sterowanie jednostką centralną z powodu różnych przyczyn:

- żądanie wejścia / wyjścia,
- wystąpienia zdarzenia dotyczącego programu o wyższym priorytecie, który aktualnie jest w stanie oczekiwania,
- zakończenia wejścia / wyjścia.

Zdarzenia te przerywają aktualnie realizowany program i zmuszają jednostkę centralną do realizacji programu monitora. Po zakończeniu obsługi przerwania, monitor zwalnia jednostkę centralną, która może wtedy ponownie uruchomić albo przerwany program, albo inny program, który oczekiwał na zdarzenie, które właśnie wystąpiło.

#### Zdalne przetwarzanie wsadowe

Użytkownik może wprowadzić pracę z pewnej odległości od systemu komputerowego i tam odebrać wyniki, przy pomocy terminala składającego się z czytnika kart, drukarki i klawiatury. Prace te są przyjmowane w taki sam sposób, jak prace lokalne. Oddalony użytkownik może używać wszystkich urządzeń zewnętrznych w taki sam sposób, jak użytkownik lokalny.

#### Zarządzenie transakcjami

SIRIS 8 umożliwia wymianę danych między zbiorami umieszczonymi centralnie w pamięci masowej i pewną ilością oddalonych użytkowników. Możliwy jest zatem dialog przy pomocy terminala typu dalekopisu lub alfaskopu.

Przy pomocy terminala użytkownik może prowadzić dialog ze specjalizowanym programem i przeglądać oraz aktualizować wspólne zbiory.

#### 2.1.2. Wielodostęp

Edytor tekstu umożliwia wstawianie, wymianę lub usuwanie zdań lub znaków wewnątrz dowolnego tekstu, również programu.

Konwersacyjne kompilatory FORTRAN i BASIC umożliwiają konwersacyjne pisanie i kompilowanie programów.

Przy pomocy terminala użytkownik może:

- mieć dostęp do wszystkich zbiorów bibliotecznych, w taki sam sposób, jak użytkownicy lokalni,
- tworzyć i używać zbiory czasowe,
- żądać bezpośredniej realizacji programu, z możliwością przydzielenia oddalonej konsoli do operacji wejścia / wyjścia,
- wstawić pracę do kolejki przy pomocy zbioru skatalogowanych kart sterujących.

Ten zbiór może być tworzony przy pomocy klawiatury lub w inny sposób.

#### 2.1.3. Przetwarzanie w czasie rzeczywistym

SIRIS 8 steruje procedurami, które bardzo szybko reagują na zdarzenia zewnętrzne. Dostarcza użytkownikowi procedur, które umożliwiają mu realizację niezbędnych skrótów, aby lepiej reagować na ograniczenia czasowe w tym rodzaju działalności.

#### 2.1.4. Zasoby

Zasobami systemu przetwarzania danych są wszystkie potencjalne możliwości tego systemu.

Rolą systemu operacyjnego jest przydzielanie zasobów różnym użytkownikom w celu osiągnięcia maksymalnej efektywności.

W SIRIS 8 istnieją trzy typy zasobów:

- Zasoby, które mogą być dzielone między kilku użytkowników: czas jednostki centralnej, obszar pamięci, obszar zbiorów czasowych w systemie dyskowym.
- Zasoby robocze, przydzielane określonym użytkownikom z wewnętrznej puli na ich żądanie.
- Zasoby rezerwowane, wywoływane indywidualnie.

Przed zainicjowaniem pracy SIRIS 8 oszacowuje zasoby każdego typu, które będą żądane, inicjuje w zasadzie pracę, kiedy wszystkie żądane zasoby są dostępne. Może być jednak osiągnięte bardziej wyrafinowane zarządzanie na poziomie etapu pracy, dla pewnych zasobów i kosztem pewnej ostrożności.

#### 2.2. OGÓLNA STRUKTURA SIRIS 8

Monitor systemowy SIRIS 8 składa się z modułów programowych i tablic. Poniżej opisane są różne elementy struktury monitora i realizowane przez nie funkcje.

##### 2.2.1. Funkcje podstawowe

Funkcje podstawowe tworzą jądro monitora, które steruje możliwościami sprzętowymi systemu.

##### a/ Zarządzanie zadaniami

Po przyjęciu przez monitor każdej pracy tworzone jest zadanie, które może przechodzić w różne stany. Celem zarządzania zadaniami jest przetwarzanie tych zmian stanów.

Zadanie może być w jednym z następujących stanów:

- aktywnym,
- przerwania,
- wstrzymania,
- oczekiwania.

Zadania obecne w systemie w danym czasie są podzielone na kilka poziomów przerwania zewnętrznych; dlatego akcje wybierania i przełączania, które są ściśle związane ze sprzętem IRIS-80, są bardzo szybkie.

Stan zadania jest w rzeczywistości określony przez poziom przerwania, do którego jest ono podłączone.

Przejęcie w stan aktywny realizowane jest przez przełączenie przerwania wewnętrznego. Jeśli aktywny jest wyższy poziom, wtedy zadanie nie mo-

że przejąć sterowania i pozostaje w stanie wstrzymania. W przeciwnym przypadku zadanie aktywne przechodzi w stan przerwania, z korzyścią dla nowych zadań, które stają się aktywne.

Przechodzenie zadania w stan oczekiwania odpowiada dezaktywacji poziomu przerwania. Zatem mogą być uruchomione zadania o niższym priorytecie.

#### b/ Zarządzanie pamięcią operacyjną

SIRIS 8 wykorzystuje mechanizm stronicowania. Celem tej funkcji monitora jest sterowanie przydzielaniem i uwalnianiem stronic pamięci rzeczywistej odpowiednio do pamięci wirtualnej używanej przez różne aktywne programy.

Każde zadanie ma swoją odpowiednią pamięć wirtualną; mechanizm stronicowania dostarcza wzajemnych relacji między dostępną pamięcią rzeczywistą i pamięcią wirtualną, przy pomocy tablicy stronic zadania aktualizowanej przez SIRIS 8.

Obszar wirtualny podzielony jest na kilka części:

- obszar monitora rezydentnego,
- obszar zadania obejmujący:
  - jedną stronicę bloku sterującego dostępną tylko dla odczytu,
  - obszar samego programu. W zasadzie nie wypełnia on całego obszaru, który był żądany dla zadania w czasie inicjowania pracy lub etapu; pozostaje zatem:
    - obszar, który nie jest początkowo przydzielony, lecz który może być przydzielony dynamicznie;
    - obszar, który nie jest początkowo przydzielony i który może być dynamicznie podzielony na bloki czterosłowe.
- obszar monitora nierezydentnego, który będzie zawierał moduły nierezydentne, które będą ładowane na żądanie.

Po zakończeniu zadania uwolnione stronice są zwracane do puli stronic wolnych, bez zakłócenia innych zadań obecnych w systemie w danej chwili.

Używając funkcji komunikacji z otoczeniem użytkownik dysponuje czterema procedurami zarządzania pamięcią, które umożliwiają mu żądanie lub uwalnianie pamięci w miarę potrzeb.

#### c/ Zarządzanie pamięcią pomocniczą

Pamięć pomocnicza obejmuje wszystkie typy pamięci dyskowych. Obszar dyskowy zarządzany jest tą funkcją w podobny sposób, jak obszar pamięci operacyjnej. Po zażądaniu obszaru użytkownik będzie pracował w obszarze ciągłym, chociaż przydział może być nieciągły. Reorganizacje i zmiany nośników nie wpływają na obszar wirtualny użytkownika.

Zarządzanie SIRIS 8 jest optymalizowane, aby uniknąć nadmiernych przestoju pamięci pomocniczej:

- W czasie przydzielania żądanie jest realizowane poprzez minimalną ilość obszarów fizycznych.
- Po każdym uwolnieniu obszaru wolne obszary są przegrupowywane tak, aby utworzyć największe możliwe obszary fizyczne.

d/ Superwajzor wejścia / wyjścia

Cele tej funkcji są następujące:

- Centralizacja żądań wejścia / wyjścia urządzeń zewnętrznych lokalnych lub linii transmisji w celu utrzymania maksymalnej możliwej szybkości transmisji dla wszystkich jednostek. W tym celu tworzona jest kolejka do urządzenia zewnętrznego tak, aby użytkownik nie musiał brać pod uwagę:
  - adresu fizycznego urządzenia zewnętrznego,
  - jego tożsamości /robocze urządzenia zewnętrzne/,
  - jego stanu w czasie żądania
- Optymalizacja dostępu do pamięci dyskowej.

Dla dysków o ruchomych głowicach tworzone są dwie kolejki dla jednego pakietu dyskowego i żądania są przełączane do jednej z nich lub do drugiej w zależności od tego, czy żądanie dotyczy cylindra, nad którym już przeszły głowice, czy nie. W każdej kolejce żądania są rozmieszczane i łączone razem, aby umożliwić realizację jak powyżej. Kiedy jedna kolejka jest pusta, aktywowana jest druga. Ruch ramienia jest zatem regularny i optymalizowany.

e/ Zarządzanie monitorem nierezydentnym

Cały monitor nie może stale rezydować w pamięci operacyjnej. Rolą tej funkcji jest zatem wywoływanie określonych modułów nierezydentnych do pamięci, na żądanie programów, oraz uwalnianie odpowiedniego obszaru po użyciu.

Funkcja zarządzania modułami monitora nierezydentnego stale kontroluje obszar wirtualny, który pozostaje wolny wewnątrz obszaru przydzielonego do monitora nierezydentnego, dla każdego zadania. Gdy moduł nierezydentny jest żądany przez program użytkowy, niezbędna ilość stronic rzeczywistych pobierana jest z puli wolnych stronic i przydzielana do odpowiedniego obszaru wirtualnego. Modyfikowana jest tablica stronic zadania tak, że pola adresowe załadowanych modułów nie wymagają żadnej modyfikacji. Po użyciu modułu jest on traktowany jako dostępny i pozostaje w pamięci. Uwalnianie obszaru, w miarę potrzeby, osiagane jest przez użycie algorytmu, który bierze pod uwagę częstotliwość wywołań modułów nierezydentnych, które zostały załadowane.

### 2.2.2. Funkcje usługowe

#### a/ Komunikacja z operatorem

Funkcja ta obejmuje i centralizuje wszystkie informacje wymieniane między systemem i operatorem. Wymiany mogą być inicjowane albo przez operatora albo przez system.

Minimalna konfiguracja konsoli sterującej obejmuje klawiaturę i dwa alfaskopy, które są zarządzane niezależnie i są podzielone na trzy obszary:

- linię górną, która przekazuje informację ogólną,
- dziesięć linii środkowych, które są zarezerwowane dla szczególnych komunikatów określonych przez operatora,
- linię dolną używaną dla komunikatów,

Przedstawiana informacja nie zawsze może być dopasowana do jednego ekranu. Wygodniej będzie traktować ekran rzeczywisty jak okno przesuwające się po ekranie wirtualnym o tej samej szerokości i zmiennej długości i traktowanym jak obwód zamknięty.

#### Komunikacja operator-system

Definiuje się trzy typy wymiany:

- Wymiana typu S.O. /system - operator/. Ten typ komunikatów nie wymaga odpowiedzi. W zasadzie dotyczy on zmian w stanach prac.
- Wymiana typu S.O.S. /system - operator z operacją odwrotną/. Te komunikaty w zasadzie dotyczą urządzeń zewnętrznych:
  - żądań założenia /zdjęcia wolumenów wymiennych,
  - wskazania niewłaściwej pracy urządzeń, co może być poprawione przez interwencję operatora.
- Wymiana typu O.S.O. /operator - system, z odpowiedzią systemu/. Operator może zainicjować ten typ wymiany przy pomocy klawiatury.

#### b/ Symbionty. Telesymbionty.

Symbiont jest pracą równoległą uruchamianą przez operatora, która zapewnia wprowadzanie i wyprowadzanie prac całkowicie asynchronicznie z ich przetwarzaniem przy pomocy pamięci pośredniej na dysku systemowym. Umożliwia to najlepsze wykorzystanie wolnych urządzeń zewnętrznych oraz uniknięcie zwolnienia pracy jednostki przetwarzającej. Telesymbiont dostarcza tych samych funkcji dla prac zdalnych.

Symbionty i telesymbionty zarządzają pewną ilością elektromechanicznych urządzeń zewnętrznych: drukarki, czytniki kart i perforatory.

- Symbiont wejściowy.

Dane i programy na kartach czytane są przez symbiont wejściowy. Informacja ta zapamiętywana jest na dysku systemowym, w obszarze przydzielonym dynamicznie. Technika ta dostarcza pełnego asynchronizmu działania wolnych urządzeń zewnętrznych i dysku systemowego, zapewnia optymalne

działanie czytnika kart. Jeśli nie ma wystarczającego obszaru dyskowego, symbiont wejściowy przechodzi w stan oczekiwania.

- Symbiont wyjściowy.

Symbiont wyjściowy wyprowadza wyniki na drukarkę lub perforator kart i może być realizowany po zakończeniu tworzenia zbioru symbiontu lub bezpośrednio po jego utworzeniu. Operator może zawiesić, skasować lub ponownie zainicjować symbionty. Dodatkowo operator może określić numer kategorii zbioru wyprowadzonego przez symbiont. Numeru tego można używać do lepszego sterowania wydrukiem. Przy użyciu numeru można także wyprowadzać zbiór symbiontu na drukarkę oddalonej stacji.

- Telesymbionty

Telesymbionty umożliwiają zdalne wprowadzanie i wyprowadzanie prac w taki sam sposób jak symbionty. Dla osiągnięcia tego używają one programów specjalizowanych w zarządzaniu liniami superwajzora wejścia / wyjścia.

c/ Interpretacja kart sterujących /CCI/.

Karty sterujące, które opisują prace, są analizowane przez tę funkcję, która sprawdza ich poprawność i inicjuje określone działania.

Funkcja ta kompiluje karty sterujące opisujące wszystkie prace. Kompilacja powoduje:

- Bezpośrednie wykrywanie dowolnych błędów, które przerywają pracę.
- Przekształcenie danych na postać upakowaną dając optymalne łańcuchowanie różnych etapów prac.

Funkcja ta może być używana na trzy różne sposoby:

- Analiza kart bez realizacji, w celu kontroli poprawności.
- Analiza i katalogowanie.
- Analiza i realizacja.

Katalogowanie polega na przechowywaniu kart sterujących pracą w specjalizowanym zbiorze na dysku systemowym. Mogą one być później wywołane przez pojedyncze odwołanie /karta sterująca lub komunikat operatora/. Katalogowanie może być połączone z parametryzacją elementów zawartych w kartach sterujących. Aktualne wartości parametrów będą określone w czasie wywołania.

### 2.2.3. Funkcje zarządzania

#### a/ Regulator prac

Wybiera on prace, które będą aktywowane zgodnie z trzema kryteriami:

- klasą kolejki,
- żądanymi zasobami,
- priorytetem.

SIRIS 8 tworzy automatycznie dwie kolejki:

- ogólną, zasilaną przez czytnik kart,
- kolejkę prac skatalogowanych, wyzwalaną przez operatora.

Dodatkowo użytkownik może zdefiniować od jednej do pięciu dodatkowych kolejek sekwencyjnych zgodnie ze swoimi wymaganiami. Może on zatem zmienić mechanizm regulatora prac i dopasować pewien poziom wieloprogramowania w celu optymalizacji wydajności pracy.

Aby dać użytkownikowi możliwość kierowania wyzwalaniem prac w trybie przetwarzania wsadowego, pole karty sterującej pracą może włączyć nową pracę do jednej z następujących kolejek:

- Kolejka ogólna /opcja domyślna/.

Prace przechodzące w stan oczekiwania w tej kolejce są traktowane przez regulator zgodnie z priorytetami zewnętrznymi i dostępnością żądanych zasobów. Kilka prac z tej kolejki może być aktywowanych jednocześnie w trybie wieloprogramowania.

- Zbiory sekwencyjne.

W czasie generowania systemu użytkownik może określić kilka dodatkowych kolejek oczekiwania. Prace zawarte w jednym z tych zbiorów aktywowane są w ściśle chronologicznej kolejności i tylko wtedy kiedy poprzednia praca w kolejce zostanie zakończona. Regulator prac aktywizuje jedną pracę z każdej kolejki. Wewnętrzne poziomy priorytetu względem ogólnej kolejki są ustalone w czasie generowania systemu.

- Kolejka prac skatalogowanych.

Prace skatalogowane mogą być aktywowane z klawiatury operatora lokalnego lub z oddalonego terminala. Kilka prac z tej kolejki może być aktywnych jednocześnie w trybie wieloprogramowania.

#### b/ Zarządzanie pracami.

Celem tej funkcji jest inicjowanie pracy po tym, gdy regulator prac podjął decyzję o jej aktywacji. Przed uruchomieniem programu funkcja ta przygotowuje niezbędne urządzenia zewnętrzne, a w szczególności kontroluje obecność wolumenów wymiennych. Pod koniec programu lub etapu funkcja kontroluje zakończenie operacji wejścia / wyjścia i rozładowuje wolumeny wymienne, które dłużej nie są potrzebne.

Zarządzanie pracami przygotowuje niezbędne dane dla analizy obciążenia systemu i przygotowania rachunków. Te dane zapamiętywane są w zbiorze, który należy do każdej pracy oraz w zbiorze centralnym, który jest używany okresowo.

#### 2.2.4. Funkcje komunikacji z otoczeniem

Dostarczają one interfejsu między użytkownikiem i jądrem monitora oraz dostarczają użytkownikowi pewnej ilości usług.

#### 2.2.4.1. Zarządzanie zbiorami.

Funkcja ta wiąże się z umieszczaniem struktur danych w pamięci i na pomocniczych taśmach magnetycznych. Lokalizacja zapewniona jest przez nazwy umieszczone w katalogach, które należą do każdego użytkownika. System zarządzania zbiorami umożliwia łatwy dostęp wraz z możliwym jednoczesnym odczytem i zapisem, do rekordów zawartych w tych zbiorach.

Zarządzanie zbiorami dostarcza następujących usług:

- powiązania programów użytkowych ze zbiorami danych, które nie mogą rezydować w pamięci operacyjnej,
- zapewnia łatwe sterowanie programowe odczytem i zapisem /tworzeniem lub aktualizacją/ rekordów zbioru.

W zależności od potrzeb użytkownik może pracować na poziomie logicznym, bez brania pod uwagę fizycznych nośników i używanych urządzeń zewnętrznych.

Moduły dostosowane do różnych struktur zbiorów zapewniają standardowe funkcje podstawowe:

- transmisję danych,
- kontrolę transmisji,
- kontrolę nośników,
- grupowanie rekordów logicznych, w celu lepszego wykorzystania jednostek pamięci,
- zamianę obszarów wejścia / wyjścia, umożliwiającą transmisję do jednego z obszarów podczas gdy drugi jest przetwarzany.

Określenie danych i ich lokalizacja zależą od organizacji zbioru:

Organizacja standardowa zbioru: charakterystyki i lokalizacja danych każdego zbioru zapisywane są w katalogu przechowywanym w nagłówku wolumenu.

Wolumen może być:

- Prywatny: dowolny nośnik, który musi być określony przez użytkownika.
- Kontowy: na dysku, wymienny lub nie, skatalogowany w systemie.

Organizacja niestandardowa zbioru: lokalizacja danych całkowicie zależy od użytkownika. System zapewnia wszelkie możliwe środki lokalizacji wymienne z możliwymi urządzeniami zewnętrznymi.

Organizacja zbioru i nośnika

W celu lepszego wykorzystania pojemności nośnika i przyspieszenia transmisji danych, struktury danych muszą być standaryzowane:

- dla organizacji zbioru na jego nośniku /struktura nośnika/,
- dla organizacji danych wewnątrz każdego zbioru /struktura zbioru/.

##### Struktura nośnika

- Dyski: zbiór dyskowy może być podzielony na kilka nieciągłych obszarów o wielkości 8 K bajtów każdy.

- Dyski wymienne: w organizacji standardowej struktura pakietu opisana jest w nagłówku pakietu, który zawiera:
  - etykietę wolumenu,
  - tablicę przydziału obszaru pakietu,
  - wykaz zbiorów na pakiecie.

Przetwarzane są następujące organizacje standardowe:

- Zbiór jednowolumenowy lub wielowolumenowy,
- Wolumen wielozbiorowy / o wielu zbiorach /,
- Wiele zbiorów na wielu wolumenach.

W organizacji niestandardowej SIRIS 8 przyjmuje, że cały wolumen zajęty jest przez zbiór.

- Taśmy magnetyczne: w organizacji standardowej taśmy mają standardowe etykiety ANSI i ECMA. Organizacja jest taka sama jak na dysku. W organizacji niestandardowej cały wolumen zajmowany jest przez zbiór.

#### Struktura zbioru

Pojęcia bloku, rekordu i formatu:

- Na taśmie magnetycznej blok odpowiada rekordowi fizycznemu.
- Na dysku blok jest stałą ilością danych zbioru, która zaczyna się od początku sektora.
- Rekord jest rekordem logicznym, kilka rekordów może być grupowane w tym samym bloku.
- Format stały /F/: rekordy, a następnie bloki, mają stałą długość.
- Format zmienny /V/: pierwsze 2 bajty każdego rekordu podają jego długość.
- Format nieokreślony /U/: rekordy i bloki są takie same.

Struktura zbioru dyskowego:

- Organizacja sekwencyjna /C/: dostęp sekwencyjny do bloku i / lub rekordów, jak na taśmie magnetycznej.
- Organizacja indeksowa /I/: dostęp do rekordu przy pomocy klucza, który ma stałą pozycję. Bloki mają stałą długość, a rekordy mają długość stałą lub zmienną. Lokalizacja danych realizowana jest przy pomocy indeksu, który ma kilka poziomów.
- Organizacja bezpośrednia /D/: dostęp do bloku określonego bezpośrednio przez jego numer.
- Organizacja partycjonowana /P/: zbiór zawiera grupy rekordów lub partycje. Każda grupa dostępna jest przy pomocy indeksu jednopoziomowego.

Struktura zbioru na taśmie magnetycznej.

Możliwa jest tylko organizacja sekwencyjna: możliwe są wszystkie formaty.

Struktura zbioru na nośniku niemagnetycznym. Zawsze jest jeden rekord na blok. Struktura danych w istotny sposób zależy od nośnika.

#### Metody implementacji zbioru

Z punktu widzenia systemu operacyjnego głównymi stanami w czasie istnienia zbioru są:

- Allokacja: przydział identyfikatora i nośnika.
- Tworzenie: wstawienie danych początkowych.
- Katalogowanie,
- Eksploatacja: odczytywanie, rozszerzanie, aktualizowanie.
- Degeneracja.
- Reorganizacja,
- Kasowanie.

Ochrona i wspólne używanie:

- Ochrona nośnika: tylko właściciel może tworzyć zbiory na już przydzielonym nośniku.
- Ochrona zbioru: dla każdego zbioru określony jest wykaz użytkowników upoważnionych do odczytu lub zapisu. Dodatkowo może być używane hasło.
- Wspólne używanie wolumenu: kilka zadań może mieć jednocześnie dostęp do wolumenu określonego jako wspólnie używany.
- Wspólne używanie zbioru: zbiór wspólnie używany umieszczony jest na wspólnie używanym wolumenie. Aktywny zbiór wspólnie używany może być w jednym z trzech następujących stanów:
  - Zapis: tylko jeden użytkownik może zapisywać; inni będą umieszczani przez SIRIS 8 w stanie oczekiwania.
  - Odczyt: kilku użytkowników może jednocześnie odczytywać zbiór.
  - Wspólne używanie danych: jeśli ten tryb jest dopuszczalny dla zbioru, można zezwolić na jednoczesny dostęp dla dowolnego użytkownika.

- Dostęp do zbioru:

W celu komunikowania się z systemem zarządzania zbiorami użytkownik dysponuje zestawem makroinstrukcji, które umożliwiają mu wybór spośród różnych metod dostępu do zbioru.

#### 2.2.4.2. Metody dostępu

Metody dostępu dostarczają interfejsu między programami użytkowymi i superwajzorem wejścia / wyjścia. Uwalniają one użytkownika od ograniczeń związanych z fizycznym nośnikiem wejścia / wyjścia.

Istnieje siedem następujących metod dostępu:

a/ Wspomagana sekwencyjna metoda dostępu /ASAM/.

Zapewnia sekwencyjne odczytywanie i zapisywanie rekordów zbioru. Rekordy są automatycznie łączone w bloki i rozłączane.

- Dopuszczalne formaty: F, V, U. Rekordy łączone w bloki lub nie.
- Tryb dostępu: odczyt lub zapis.

b/ Indeksowo-sekwencyjna metoda dostępu /ISAM/.

Umożliwia tworzenie zbioru indeksowanego, rekordy mogą być dostępne przy pomocy swojego klucza lub sekwencyjnie, poprzez pobieranie rekordów. Zbiór może być także aktualizowany poprzez modyfikowanie lub wstawianie rekordów.

- Przetwarzane formaty: F i V.
- Tryby dostępu: odczyt, zapis, aktualizacja.

c/ Partycjonowana metoda dostępu /PAM/.

Umożliwia:

- Lokalizację początku partycji o danej identyfikacji;
- Wstawianie lub kasowanie identyfikatorów;
- Sekwencyjne przetwarzanie rekordów;
- Przetwarzane formaty: F i V;
- Tryby dostępu:
  - Lokalizacja partycji i odczyt,
  - Wstawienie partycji i zapis,
  - Aktualizowanie.

d/ Niewspomagane metody dostępu.

Działają one na rekordach fizycznych. Ich rolą jest uproszczenie przesyłania rekordów z jednostek wejścia / wyjścia /zbiorów urządzeń zewnętrznych/ do programów lub odwrotnie. Z każdym przesyłaniem może być skojarzone "zdarzenie" umożliwiające użytkownikowi przejście w stan oczekiwania do czasu zakończenia jednego lub więcej przesyłań.

- Wirtualna sekwencyjna metoda dostępu /VSAM/.

Umożliwia odczyt lub zapis dowolnego zbioru /tj. na urządzeniu zewnętrznym/:

- Format: bez formatu,
- Tryby dostępu: odczyt, zapis, odczyt odwrotny, odczyt / zapis.
- Wirtualna bezpośrednia metoda dostępu /VDAM/.
- Używana jest dla nośników o dostępie bezpośrednim i umożliwia odczyt lub zapis rekordów fizycznych dowolnego formatu, rozpoczynając od początku bloku.
- Tryby dostępu: odczyt, zapis, wszystkie dostępy.
- Bazowa /podstawowa/ bezpośrednia metoda dostępu /BDAM/.
- Używana dla nośników o dostępie bezpośrednim, umożliwia odczyt lub zapis przy pomocy rzeczywistych adresów dyskowych.

e/ Transmisyjna metoda dostępu /SGT/.

Jest ona kompatybilna z innymi metodami i w znacznym stopniu niezależna od używanych procedur transmisji.

SGT używany jest w trybie blokowym /transmisja synchroniczna/ i w trybie znakowym /transmisja asynchroniczna/.

Linie i terminale traktowane jako zasoby opisane są podczas generowania systemu. Po wprowadzeniu pracy rozkaz ASSIGN określa linie przydzielone do tej pracy.

#### 2.2.4.3. Przetwarzanie wywołań użytkownika

W IRIS-80 i SIRIS 8 zwrot do monitora powoduje przerwanie sprzętowe. Funkcja obsługi tych przerw steruje zwrotami do monitora w sposób scentralizowany: podłącza obsługę aktualnie żądanej usługi i przesyła określone argumenty.

Można wyróżnić następujące grupy makroinstrukcji SIRIS 8

a/ Zarządzanie obszarem pamięci pracy - GP, FP, GSP, FSP.

b/ Zarządzanie programem - LDTRC, LINK, SEGLD, CHKPT, RESTART, TRAP.

c/ Zarządzanie zbiorami i operacjami wejścia / wyjścia - DCB, OPEN, CLOSE, GET, PUT, DELREC, CVOL, FIND, STOW, READ, WRITE, CHECK, SETDCB, ASSIGN, NOTE, TRUNC, POINT.

d/ Zliczanie czasu - TIME, STIMER, TIMER.

e/ Komunikacja zewnętrzna - INT, KEYIN, TYPE, PRINT.

f/ Zarządzanie pseudokluczami pracy - ESS, RSS, TSS.

g/ Powiązania program-system - WAIT, RETURN, ERR, IDLE, GETDAY.

h/ Procedury uprzywilejowane - MASTER, SLAVE, ATTACH, CLEAR, DETACH, TRIGGER, TASK, LOCK, FREE, RCB, ENTER, ECB, UNITECB, EVENT, DELTASK, CLOCK, RCLOCK, TRAP, GETMEM, RELMEM.

#### 2.2.4.4. Język sterujący realizacją programów /JCL/

Język sterujący realizacją programów służy do opisu każdej pracy za pomocą kart sterujących. Karty sterujące są środkiem komunikacji pomiędzy użytkownikiem i systemem. W szczególności służą do powiązania różnych etapów składających się na pracę. Za etap uważa się każdorazowe załadowanie pamięci programem i jego realizację. Zespół kart sterujących i kart danych tworzy zadanie w momencie wprowadzania go do systemu. Z tego samego wejścia można kolejno wprowadzać do systemu wiele prac, aktywowanie ich odbywa się według ściśle określonego priorytetu, zdefiniowanego w parametrach kart sterujących. JCL monitora SIRIS 8 składa się z 20 typów kart sterujących:

- JOB

Karta rozpoczynająca każdą pracę. Służy również do określania końca poprzedniej pracy. Określa, poprzez swoje opcje, typ realizacji żądany przez użytkownika:

- Analiza kart sterujących bez realizacji.
  - Analiza kart sterujących z realizacją.
  - Analiza i katalogowanie wprowadzanych kart, bez realizacji.
- RUN
- Karta ta powoduje załadowanie do pamięci operacyjnej modułu ładowalnego i wykonanie danego etapu pracy. Sposób wykonania etapu określony jest poprzez opcje tej karty.
- EXEC
- Karta ta powoduje wywołanie grupy kart rozkazowych, które uprzednio były skatalogowane w zbiorze: CTRLCARDS, na koncie użytkownika lub w zbiorze: CTRLCARDS, na koncie systemowym: SYS.
- PROC
- Wywołuje i inicjuje programy skatalogowane na koncie systemowym : SYS /kompilatory, itp./. Karta ta pozwala na przydzielenie parametrom formalnym zdefiniowanym w kartach skatalogowanych wartości efektywnych przez domniemanie.
- LIMIT
- Karta ta zawiera następujące informacje dla systemu:
- a/ określa liczbę stron jaka będzie wydrukowana na drukarce i liczbę kart perforowanych przy wyjściu z systemu.
  - b/ określa maksymalne wartości zasobów jakie dana praca będzie wykorzystywać: obszar pamięci operacyjnej i dyskowej, czas pracy jednostki centralnej.
- SLIMIT
- Karta ta definiuje ograniczenia na wspólnie używane zasoby systemu na okres realizacji etapu, którego dotyczy karta. Zasobami tymi są: obszar pamięci operacyjnej i obszar roboczy dysku.
- RESOURCE
- Karta ta wskazuje systemowi minimalne żądane wartości zasobów /urządzeń zewnętrznych/ dla aktywowanej pracy.
- ASSIGN
- Karta ta realizuje połączenie pomiędzy zespołem operacji we / wy zdefiniowanym w programie a zbiorem /i czasami nośnikiem/.
- SWITCH
- Pozwala na określenie wartości każdej z 32 pozycji binarnych JSW /JOB SWITCH WORD/ przydzielonego do danej pracy.

- TITLE

Karta ta pozwala na wydrukowanie dowolnego nagłówka na początku każdej strony.

- MESSAGE

Pozwala na przesłanie komunikatu /polecenia/ operatorowi w momencie aktywowania danego etapu.

- COMMENT

Karta ta pozwala włączyć dowolny komentarz do zestawu kart sterujących.

- DATA

Karta ta wskazuje początek kart zawierających dane.

- EOD

Wskazuje koniec kart zawierających dane.

- BCD

Wskazuje symbiontowi, że karty są w kodzie BCD.

#### 2.2.4.5. Narzędzia uruchamiania

Użytkownik może korzystać z odpowiednich procedur w dwojaki sposób:

- przez zastosowanie procesora /z lub bez modyfikacji programu, z włączeniem czasowym lub stałym modyfikacji/.
- przez zastosowanie procedur systemowych.

Zastosowanie procesora DEBUG.

- Bez modyfikacji programu.

Program użytkownika jest w postaci modułu ładowalno-wykonalnego /LMEX/ i możliwe jest, bez nowej kompilacji i konsolidacji, jego zmodyfikowanie i dołączenie do niego odwołań do DEBUG.

- Bez modyfikacji programu ale z tworzeniem programu modyfikowanego w zbiorze stałym. Jeżeli użytkownik życzy sobie zachować program zmodyfikowany, może tego dokonać za pomocą procesora DEBUG z opcjami PERM lub LDKEY.

- Z modyfikacją programu.

Jeżeli użytkownik chce zachować program zmodyfikowany bez tworzenia nowego zbioru, korzysta z procesora DEBUG z opcją ALTER.

Zastosowanie procedur systemowych.

- Bez modyfikacji programu.

Korzystanie z procedur wymaga nowej kompilacji i konsolidacji. Odwołania do procedur włączone są do programu użytkownika do momentu nowej kompilacji.

### 2.2.5. Zarządzanie transmisją

Programy realizujące transmisję używają:

- superwajzora wejścia / wyjścia,
- transmisyjnej metody dostępu,

Superwajzor wejścia / wyjścia, oprócz swoich podstawowych funkcji, zawiera dwie specjalne funkcje transmisyjne:

- Wykrywanie błędów linii.
- Obsługę dwóch przerwań zewnętrznych niezbędnych dla jednostki sterującej, działającej w trybie znakowym:

Interfejs superwajzora wejścia / wyjścia jest jednolity i wspólny dla wszystkich systemów, a także dla systemu transmisji.

#### 2.2.5.1. Wykonywane funkcje

Celem metody dostępu jest dostarczenie użytkownikowi pewnej ilości usług, aby pomóc w zarządzaniu terminalami poprzez linie transmisji oraz uwolnienie użytkownika od ograniczeń związanych z transmisją i umożliwienie mu pracy niezależnej od procedur transmisji.

Użytkownik musi posiadać pewne niezbędne informacje o terminalach:

- typy terminali /dostęp sekwencyjny poprzez zgłaszanie się i wybór lub dostęp bezpośredni, dopuszczalna długość bloku/.
- typy podłączonych urządzeń zewnętrznych.

Jest to zatem metoda dostępu, która będzie całkowicie zarządzać procedurą transmisji /nagłówki i ciągi usługowe/ zgodnie z fizycznymi charakterystykami linii i terminali.

Można wyróżnić trzy poziomy implementacji programu wykorzystującego środki telekomunikacji:

#### 1/ Generowanie systemu

W czasie generowania systemu linie i terminale opisywane są w podobny sposób, jak inne elementy systemu komputerowego.

Opis ten zawiera:

- nazwy symboliczne, fizyczne adresy podkanałów, do których podłączone są linie i typy tych linii;
- nazwy symboliczne terminali, ich adresy zgłaszania się i wybierania, ich status /stacja główna lub pomocnicza/ i ich typ; linie, do których są one podłączone i status komputera na tej linii;
- typy terminali oraz ich wszystkie charakterystyki /szczególne funkcje, możliwe formaty danych, procedury transmisji, własności opcjonalne/. Wszystkie te informacje zapamiętywane są w tablicach urządzeń systemu i używane są przez metodę dostępu do realizacji operacji transmisji użytkownika.

#### 2/ Poziom asemblacji lub kompilacji

Tworzony jest blok sterowania wejścia / wyjścia, który zawiera wszystkie charakterystyki grupy linii podłączonych do użytkownika. W tym bloku sterowania użytkownik definiuje:

- ogólną charakterystykę grupy terminali;
- adres wykazu terminali, jeśli chce używać wykazów jawnych;
- adresy wyjścia z programu w przypadku błędu lub powrotu normalnego;
- maksymalną długość bloków użytkownika;
- używany kod /binarny, EBCDIC, ASCII/.

### 3/ Poziom realizacji

Allokacja: na początku pracy przydzielana jest grupa linii. Uzyskuje się to poprzez użycie karty ASSIGN.

Otwarcie: otwarcie grupy linii /makro OPEN/ jest operacją, która:

- uzupełnia blok sterowania wejścia / wyjścia o tablicę rozszerzeń zawierającą wszystkie zmienne parametry wejścia / wyjścia /kontekst transakcji/;
- przygotowuje bloki wejścia / wyjścia i programy procesora wejścia / wyjścia. Celem tego jest automatyzacja konwencjonalnych operacji na linii /transmisja bloku, odbiór bloku, ciągi usługowe/;
- sprawdza podłączenia linii i odpowiedniość między grupą linii żadaną przez użytkownika i tą, która została mu przydzielona;
- wskazuje regulatorowi prac, że użytkownik będzie realizował transmisję i dlatego musi otrzymać odpowiedni poziom priorytetu.

Odczyt i zapis: użytkownik może realizować operacje transmisji i odbioru przy pomocy makroinstrukcji WRITE i READ. Przesyłanie odbywa się bezpośrednio do obszaru pamięci określonego przez makroinstrukcję.

#### 2.2.5.2. Współpraca programu użytkowego z systemem transmisji

Program użytkowy komunikuje się z siecią transmisji przy pomocy makroinstrukcji, które są krótko opisane poniżej:

- tworzenie /lub modyfikacja/ bloku sterowania przy pomocy M : DCB /lub M : SETDCB/;
- otwarcie lub zamknięcie grupy linii przy pomocy OPEN i CLOSE;
- tworzenie /lub modyfikacja/ wykazu zgłaszania się przy pomocy makro LIST /lub MDF LIST/;
- wysyłanie i odbiór poprzez WRITE i READ;
- szczególne operacje na urządzeniach zewnętrznych przy pomocy DEVICE.

Bufory określone są przez użytkownika w każdej makroinstrukcji READ lub WRITE. Mogą być one zarezerwowane w obszarze dynamicznym zadania /makroinstrukcje GSP i FSP/.

Po każdym żądaniu bufora /GSP/ następuje operacja badania. W przypadku nadmiaru jest to wykazane i wykonywany jest powrót do programu

użytkowego. Długość każdego bufora jest zmienna i jest określona przez użytkownika.

#### 2.2.5.3. Kolejki

Kolejki obsługiwane są sekwencyjnie. Istnieje jedna kolejka na koncentrator lub jednostkę sterującą i na terminal w przypadku bezpośredniego adresowania terminala. Sprawdzenie kolejki zapewnione jest przez superwajzor wejścia / wyjścia.

Istnieją trzy poziomy:

- umieszczenie w kolejce, na pierwszym poziomie superwajzora;
- optymalizacja na drugim poziomie;
- inicjowanie operacji i zakończenie akcji przetwarzania na poziomie wejścia / wyjścia.

#### 2.2.5.4. Zarządzanie linią i terminalem

W trybie blokowym używane jest wybieranie zgłoszeń, a następnie procedura TMM.

W trybie znakowym używane są dwa przerwania.

W trybie blokowym możliwe jest przydzielenie priorytetu do terminali przy pomocy wygodnie wybranych wykazów zgłoszeń lub poprzez użycie specjalnych zgłoszeń /zgłoszony jest terminal zamiast koncentratora/.

Jednoczesne przetwarzanie transakcji - transakcje mogą być przetwarzane jednocześnie bez duplikowania programów. Wszystkie usługi są "czyste" umożliwiając pisanie "czystych" programów.

Dodatkowo użytkownik ma dostęp do pewnych ułatwień:

- Dynamiczne zarządzanie buforem.
- Wielozadaniowość, umożliwiającą równoległe /współbieżne/ przetwarzanie różnych zadań bez zmiany poziomu priorytetu.

#### Ochrona

Możliwe jest zabronienie pewnym terminalom tworzenia pewnych typów transakcji:

- na poziomie przydziału, poprzez przydzielenie tylko pewnych linii do danej pracy;
- na poziomie programu użytkowego.

#### Statystyka błędów

Błąd pojedynczy i jakiegokolwiek powtórzenia są obsługiwane automatycznie przez metodę dostępu. Użytkownik będzie tylko ostrzeżony, jeśli błąd utrzyma się po osiągnięciu żądanej ilości powtórzeń. Badanie kodu zakończenia operacji wejścia / wyjścia umożliwi mu wtedy podjęcie decyzji czy chce próbować znowu, czy ostrzec operatora czy przerwać realizację programu.

W dowolnym przypadku metoda dostępu odnotuje ten błąd, wraz z pełną charakterystyką, w specjalnym zbiorze. Charakterystyka zapisywanego błędu zawiera: nazwę linii, nazwę terminala, rodzaj wykonywanej operacji, ilość powtórzeń, stan linii. Informacja ta może być później badana w miarę potrzeby, w celu gromadzenia statystyki, zgodnie z wymaganiami zastosowania. To nie jest tylko informacja rozliczeniowa. Liczby wysłanych komunikatów i bloków są także aktualizowane przez linię i terminal oraz czasy otwierania i zamykania linii.

#### 2.2.5.5. Tryby transmisji

Tryb transmisji jest asynchroniczny, w trybie znakowym, na liniach wolnych o szybkości od 50 do 1200 bodów.

Tryb transmisji jest synchroniczny powyżej 1200 bodów i czasami na poziomie 1200 bodów, tj. kiedy komunikat /blok/ poprzedzony jest znakami dającymi znakowy poziom synchronizacji.

#### 2.2.5.6. Ochrona przed błędami

Ochrona przed błędami uzyskiwana jest na dwóch etapach:

- wykonywanie błędów przez jednostkę sterującą transmisją,
- podejmowanie decyzji przez procesor wejścia / wyjścia o kolejności operacji, które będą wykonane.

##### a/ Wykrywanie błędów przez jednostkę sterującą

Jednostka sterująca dostarcza znaków parzystości i kontroli parzystości wzdułżnej. Wykonuje ona cykliczne wyliczanie kodu /EBCDIC/ i porównanie z odpowiednimi odebranymi znakami. Wykrywa także różne błędy transmisji lub anomalie takie jak: niepoprawna długość, niepoprawny blok, błąd modemu, błąd generatora fali, błąd gotowości transmisji, itd./. Kontroluje również przerwy w pracy:

- przerwę krótką do wykrycia braku odpowiedzi;
- przerwę długą dla wskazania braku aktywności linii;

Wykrywa błędy między jednostką sterującą i komputerem.

##### b/ Rola procesora wejścia / wyjścia

Procesor wejścia / wyjścia jest komputerem zdolnym do realizacji skomplikowanych programów rządzących wymianą z jednostkami sterującymi. Po otwarciu linii monitor przygotowuje ciąg programów procesora wejścia / wyjścia. Za inicjowanie tych programów całkowicie odpowiedzialna jest jednostka centralna. W przypadku zdarzenia /np. powtórzenia/ jednostka centralna ostrzegana jest poprzez przerwanie, analizuje zdarzenie i inicjuje niezbędne programy procesora wejścia / wyjścia

W przypadku błędów powtarzanie bloków jest automatyczne.

## 2.2.6. Języki - kompilatory

### 2.2.6.1. METASYMBOL

Jest to język przeznaczony dla programistów znających organizację wewnętrzną komputera IRIS-80. Translator języka METASYMBOL jest wieloprzebiegowym makroassemblerem, w którym dwie fazy asemblacji /II i III/ poprzedzone są wstępną fazą I rozkodowania programu źródłowego. METASYMBOL pozwala na pisanie programów optymalnych pod względem wykorzystania zasobów komputera IRIS-80 i minimalizacji czasów ich realizacji. Cechy języka:

- Samodefiniujące się stałe /szesnastkowe, dziesiętne, ósemkowe, stałe i zmiennoprzecinkowe/,
- Łatwość pisania dużych programów podzielonych na segmenty lub moduły,
- Wartości mogą zajmować w pamięci bajt, pół słowa, słowo lub podwójne słowo /odpowiednia instrukcja zwiększa adres na początek słowa/,
- Mnemotechnicznie oznaczone procedury standardowe dla przetwarzania z uwzględnieniem różnorodnych opcji sprzętowych.
- Dyrektywy TEXTC i TEXT dla prostego kodowania wyprowadzonych komunikatów bez tworzenia licznika znaków:
- Pole argumentów może zawierać stałą lub zmienną ilość wyrażeń arytmetycznych lub logicznych.
- Możliwość pełnego komentowania i stosowania opisów.
- Dyrektywy DO i WHILE umożliwiają selektywne generowanie sekwencji kodowych przy zadanych parametrach, albo wyrażeń określonych podczas asemblacji.
- Zdolność do generowania wielu rozkazów dla jednej instrukcji języka. Można przy tym uzyskiwać całkowicie sparametryzowane kodowanie dla procedur wykorzystywanych w wielu programach.
- Procedury funkcyjne zwracają wartość do wiersza wywołania w programie.
- Instrukcje wywołania i parametry aktualne procedury mogą być kontrolowane arytmetycznie i logicznie.
- Mogą być stosowane procedury wielopoziomowe i jedna procedura może wywoływać inną.
- Dozwolone jest stosowanie w procedurach operatorów arytmetycznych i logicznych.
- Podczas asemblacji następuje wykrywanie błędów w składni całego programu.
- Dostępna jest biblioteka podprogramów matematycznych /napisanych w FORTRAN'ie/, które mogą być wywoływane przez program w METASYMBOL'u.

#### 2.2.6.2. LP-70

Jest to język utworzony na bazie ALGOL'u, przeznaczony przede wszystkim do pisania programów dla celów symulacji i tworzenia systemów operacyjnych /bezpośredni dostęp do rejestrów, instrukcje sterowania generacją kodu wynikowego/.

Cechy języka:

- format instrukcji - dowolny;
- struktura programu - blokowa /zespół deklaracji i instrukcji zawarty pomiędzy początkiem bloku BEGIN i końcem END/;
- deklaracja danych /np. różne segmenty programu korzystają ze wspólnych danych zadeklarowanych w GLOBAL DATA/.
- instrukcje warunkowe:
  - IF warunek THEN instrukcja;
  - IF warunek THEN instrukcja 1 ELSE instrukcja 2;
- instrukcja zakładania stosu /STACK/;
- instrukcje pętlenia programu /LOOP - instrukcja tworząca pętlę "niekończoną", wyjście z pętli za pomocą instrukcji EXIT i GOTO, REPEAT - instrukcja pozwalająca wykonywać instrukcję określoną ilość razy: FOR - pozwala wykonać instrukcję określoną /stanem rejestru/ ilość razy;
- wyrażenia arytmetyczne i logiczne i wszystkie możliwości assemblera;
- możliwość wywoływania podprogramów napisanych w językach FORTRAN i METASYMBOL.

#### 2.2.6.3. LPS /makro LPS/

Język programowania systemowego LPS zastępuje język assemblera. Pisanie programów w tym języku jest uproszczone, zwiększa się ich czytelność i łatwość poprawiania. Uruchamianie programów jest znacznie szybsze dzięki specjalnym instrukcjom wspomagającym uruchamianie oraz ze względu na brak konieczności bezpośredniego operowania rejestrami, co zwykle powoduje większość błędów. W zwykłym języku assemblera trudno czytelna jest zazwyczaj struktura logiczna programu. Natomiast w LPS instrukcje maszynowe zastąpione są przez operatory arytmetyczne bądź logiczne, które w pewnym zakresie uniezależniają programistę od fizycznej struktury maszyny. LPS umożliwia deklarowanie danych i bezpośredni opis ich struktury. Instrukcje tworzące funkcje logiczne mogą być grupowane za pomocą "procedur", które odpowiadają blokom w sensie ALGOL-u, ale nie są odpowiednikami procedur w językach ALGOL i PL/I.

LPS pozwala na programowanie niezależnie od jednostki centralnej komputera, ponieważ nie wykorzystuje bezpośrednio rejestrów. Jest on uzależniony od systemu operacyjnego oraz od konfiguracji maszyny.

Poza translatorem języka LPS istnieje także procesor Macro-LPS, który dokonuje wstępnego tłumaczenia programu /zawierającego definicje lub odwołania do makroinstrukcji/ na tekst w języku LPS.

Translator Macro-LPS korzysta z jednej strony z:

- biblioteki użytkownika,
- definicji makroinstrukcji użytkownika,
- biblioteki systemowej.

Z drugiej zaś strony z tekstu źródłowego zawierającego:

- instrukcje LPS,
- wywołania makroinstrukcji,
- meta - instrukcje /precyzujące sposób generowania makroinstrukcji/.

#### 2.2.6.4. FORTRAN IV - CII - rozszerzony

Język ten stanowi nadzbiór w stosunku do innych odmian FORTRAN-u IV, które mogą być bez istotnych zmian tłumaczone i realizowane w komputerze IRIS-80. Ponadto możliwe jest stosowanie bardziej rozbudowanej składni oraz dużej liczby dodatkowych udogodnień niespotykanych w innych systemach FORTRAN. Obejmuje to między innymi:

- wyrażenia w wykazie OUTPUT,
- rozbudowane instrukcje przyporządkowania,
- rozbudowane wyrażenia relacyjne,
- tablice N-wymiarowe,
- uogólnione grupy tablic,
- uogólnione podpisy,
- zmienne globalne /określone przez nazwy, a nie przez pozycję w wykazie COMMON/,
- nieograniczona długość identyfikatora,
- instrukcje mieszane,
- uogólnione instrukcje DO i REPEAT,
- podprogramy wewnętrzne /działające na tych samych zmiennych co program główny i kompilowane łącznie z nim/,
- możliwość pisania kilku instrukcji w linii rozdzielonych " ; ",
- automatyczna podwójna precyzja,
- wielokrotne wejścia do podprogramów,
- zmienna liczba argumentów w wywołaniach podprogramów,
- etykiety jako argumenty podprogramów,
- etykiety lokalne i globalne,
- konwersja danych pao-pao /ENCODE/DECODE/,
- formaty bezwymiarowe, gdzie liczba znaków zależy od wyprowadzanej wartości,
- przetwarzanie końca zbioru /EOF/,

- uproszczone instrukcje wejścia / wyjścia,
- wstawianie sekwencji w języku METASYMBOL,
- kompilacja warunkowa,
- rozbudowana i czytelna diagnostyka błędów,
- dualne zbiory znaków /BCD lub EBCDIC/.

Charakterystyka kompilatora:

- tłumaczenie programów przystosowane do realizacji w systemie wielo-  
programowym,
- minimalizacja zajętości pao podczas kompilacji i realizacji programów,
- redukcja czasu realizacji programu,
- zbiory wejściowe:
  - a/ symboliczne z programem źródłowym. Są to zbiory sekwencyjne o re-  
kordach 80 bajtowych z tekstem programu.
  - b/ skondensowane z programem źródłowym. Są to zbiory sekwencyjne lub  
dzielone /partycjonowane/ z tekstem skondensowanym.
  - c/ symboliczne z poprawkami.

Deklaracje zmian wraz z wierszami zaktualizowanymi mogą być wpro-  
wadzane w postaci symbolicznej np. z kart, tekst źródłowy programu  
wprowadzany jest wtedy z wejścia skondensowanego.
- zbiory wyjściowe:
  - a/ binarne wynikowe, przetwarzane następnie przez procesor konsolidacji  
/link edytor/; są to zbiory sekwencyjne lub partycjonowane.
  - b/ symboliczne z postacią źródłową poprawionego programu, które można  
wykorzystać przy następnej kompilacji jako wejściowe; są to zbiory  
sekwencyjne.
  - c/ skondensowane z postacią źródłową poprawionego programu, które mo-  
żna następnie wykorzystać jako wejściowe; są to zbiory sekwencyjne  
lub partycjonowane.
- edytor tekstu wbudowany w kompilator /poprawianie tekstu źródłowego  
podczas kompilacji/.

#### 2.2.6.5. PL/I

Język PL/I implementowany na komputerze IRIS-80 odpowiada normom programowania PL/I stosowanym przez IBM. Podręcznik języka PL/I zawiera szczegółowy opis wszystkich opcji reprezentacji wewnętrznej danych, programu wynikowego, biblioteki procedur standardowych, komunikatów o błędach podczas kompilacji i realizacji programu. Program w PL/I może odwoływać się do procedur kompilowanych oddzielnie przez kompilator FORTRAN-u.

#### 2.2.6.6. SIMULA 67

Jest to uniwersalny język programowania wyższego rzędu, stanowiący rozszerzenie ALGOL-u 60 o język symulacji obiektów.

Język SIMULA wprowadza, oprócz pojęć języka ALGOL, nowe pojęcia: klasy obiektów, prefiksowania klas i systemy quasi - równoległe. Posiada wszelkie udogodnienia dla obliczeń numerycznych, działań na tekstach, działań na zbiorach i strukturach listowych. Posiada bogaty zestaw pojęć standardowych i procedur wejścia - wyjścia, który może być łatwo rozbudowany przez użytkownika do postaci dla niego wygodnych. Możliwości symulacji obiektów oparte są na językach wbudowanych SIMSET /działania na zbiorach skończonych reprezentowanych przez listy dwukierunkowe/ i SIMULATION /narzędzia ułatwiające symulację procesów rzeczywistych, tzn. tworzenie i badanie zachowania się modeli systemów rzeczywistych/. Do celów symulacji potrzebne są również procedury symulacji liczb losowych o różnych rozkładach. Są one również wbudowane w język.

#### 2.2.6.7. COBOL

Język COBOL w systemie IRIS-80 stanowi realizację standardu COBOL-ANS. Kompilator został rozszerzony o dwie funkcje nie znajdujące się w standardzie:

- kompilację cząstkową;
- środki ułatwiające uruchamianie programów /DEBUGGER/.

Kompilacja cząstkowa - możliwość niezależnego pisania, kompilacji i uruchamiania programów w języku COBOL, które później mogą być łączone w jeden większy program. Należy przy tym przestrzegać następujących reguł:

- jeden z programów cząstkowych stanowi program główny, a następne stają się podprogramami;
- program główny musi zawierać wszystkie deklaracje zmiennych, zbiorów i rekordów stosowanych we wszystkich programach cząstkowych.

Środki ułatwiające uruchamianie programów.

Wprowadzono dwie instrukcje ułatwiające i przyspieszające uruchamianie programów:

- TRACE,
- EXHIBIT.

Instrukcja TRACE powoduje wydruk nazwy paragrafu lub sekcji programu w chwili rozpoczęcia jej wykonania. Instrukcja EXHIBIT powoduje wydruk wartości bieżącej zmiennej.

Programy w języku COBOL mogą być segmentowane. Przetwarzane zbiory mogą być sekwencyjne lub sekwencyjno-indeksowe. Instrukcja ENTER pozwala na oddanie sterowania do podprogramu napisanego w językach META-SYMBOL lub FORTRAN przy zachowaniu określonych reguł przekazania parametrów.

#### 2.2.6.8. ALGOL - IRIS-80

Język akceptowany przez kompilatory implementowane w komputerze IRIS-80 obejmuje poziom 1 ALGOL 60. Jest to podzbiór poziomu 0 co sprawia, że każdy program napisany w języku poziomu 1 jest także programem poziomu 0. Posiada on następujące ograniczenia w stosunku do ALGOL-u 60:

- litery tylko duże,
- deklaracja 'OWN' nie jest akceptowana,
- nazwy rozróżniane są z dokładnością od sześciu pierwszych znaków,
- przy nieokreślonym typie wyrażenia arytmetycznego, wartości lub parametru aktualnego - następuje sprowadzenie do typu 'REAL' już na poziomie kompilacji,
- etykieta jest używana wyłącznie jako identyfikator,
- w instrukcji procedury typ i rodzaj parametrów aktualnych musi być taki sam, jak w odpowiednich parametrach formalnych,
- użycie funkcji w formie instrukcji procedury równoznaczne jest z instrukcją pustą,
- w deklaracji procedury muszą być zadane specyfikacje wszystkich parametrów formalnych.

Rozszerzenia w stosunku do ALGOL 60.

Przy napotkaniu procedury, której ciało utworzone zostało przez symbol bazowy 'FORTRAN' lub 'CODE', kompilator odwołuje się do podprogramu w FORTRAN-ie lub METASYMBOL-u. Instrukcje wejścia / wyjścia, przewijania taśmy, zamykania zbioru, wykonywane są przez specjalne procedury wywoływane przez użycie symboli bazowych 'READ', 'WRITE', 'BACKSPACE', 'ENDFILE' lub 'REWIND' w programie. Instrukcja 'FORMAT' stosowana jest do specyfikowania konwersji danych transmitowanych przy wykonywaniu "zredagowanych" instrukcji wejścia / wyjścia.

#### 2.2.7. Programy usługowe

##### 2.2.7.1. Procesor konsolidacji programów /Link edytor/

Procesor konsolidacji programów przeznaczony jest do łączenia między sobą modułów programowych różnego pochodzenia, dla spełnienia ich odwołań zewnętrznych w celu utworzenia programu akceptowanego przez procesor ładowania i realizacji.

Procesor konsolidacji tworzy zbiór albo partycję zbioru, który nazywany jest modułem ładowalnym /LM - Load Module/. Zbiór ten tworzony jest z wyników translacji i uprzednich konsolidacji, tzn.:

- modułów wynikowych /OM - Object Module/;
- modułów ładowalnych /LMED/, które powstały w wyniku uprzednich konsolidacji modułów;
- partycji biblioteki wynikowej /LOM/.

Moduły wynikowe są zbiorami przechowywanymi na nośnikach magnetycznych i odpowiadają wyjściom BO/GO assemblera i kompilatorów systemu SIRIS 8. Są opisane w języku pośrednim i pierwszym zadaniem procesora konsolidacji jest przetłumaczenie tego języka na czysty język maszynowy wraz z adresami absolutnymi.

Kolejnymi zadaniami są:

- połączenie modułów za pomocą symboli zewnętrznych określonych w trakcie działania assemblera lub kompilatora z tworzeniem struktury nakładkowej lub bez;
- tworzenie zbiorów i partycji bibliotecznych /LOM, LMED/ oraz używanie ich w celu przechowania żądanych modułów programu.

#### 2.2.7.2. Generator konwersji

Program generator, inicjowany poprzez karty sterujące, umożliwia użytkownikowi tworzenie programów konwersji nośników zbiorów. Programy te będą realizować wszystkie standardowe akcje konwersji oraz pewną ilość funkcji własnych użytkownika, które mogą być zdefiniowane przy pomocy specjalnych procedur.

Możliwe konwersje:

Karty na	:	dysk
		taśmę magnetyczną
Taśma		dysk
na	:	taśmę
Dysk		karty
		drukarke

#### 2.2.7.3. Sortowanie i łączenie

Program generator tworzy programy sortowania i łączenia dostosowane do wymagań użytkownika. Program ten inicjowany jest poprzez zestaw kart sterujących, które w rzeczywistości tworzą język generacji.

##### 2.2.7.3.1. Charakterystyki sortowania

Sortowanie sekwencyjne jest to przetwarzanie rekordów o stałej lub zmiennej długości. Istnieją dwa typy sortowania: wielofazowe i zrównoważone. Sortowanie wielofazowe może używać czytania do tyłu i działa na 2 do 6 jednostkach. Sortowanie zrównoważone jest sortowaniem dyskowym.

Przy pomocy programu sortującego użytkownik może manipulować zbiorami i rekordami wejściowymi i wyjściowymi. Klucze sortowania mogą mieć długość od 1 do 255 bajtów i mogą być:

- alfanumeryczne /kod EBCDIC/,
- alfanumeryczne z dekodowaniem określonym przez użytkownika,

- dziesiętne, rozszerzone lub upakowane,
- binarne.

W dowolnym przypadku kolejność sortowania może być rosnąca lub malejąca. Zbiory mogą być na taśmie lub na dysku. Sekwencje umożliwiają użytkownikowi:

- wstawianie, kasowanie lub modyfikowanie rekordów,
- odczyt i zapis zbioru,
- kontrolę etykiet.

#### 2.2.7.3.2. Łączenie

Rolą tego programu jest łączenie posortowanych zbiorów. Wszystkie możliwości sortowania dostępne są dla łączenia.

#### 2.2.7.4. Generator zbiorów testowych

Generator zbiorów testowych używa prostego języka, podobnego do COBOL'u, do tworzenia zbiorów. Generator tworzy zbiory na dysku, taśmie magnetycznej lub kartach.

#### 2.2.7.5. Generator systemu

Generator systemu wytwarza system operacyjny SIRIS 8 i system programowania dostosowany do konfiguracji instalacji komputera i do specjalnych wymagań użytkownika.

#### 2.2.7.6. Konserwacja zbiorów

Programy konserwowania zbiorów umożliwiają:

- wstępne oznakowanie wolumenu,
- wstawianie zbiorów, konwersję i reorganizację.

### 2.3. WIELODOSTĘP /TIME SHARING/

#### 2.3.1. Ogólna struktura

Specyficzne funkcje SIRIS 8, które oferują użycie komputera IRIS-80 przez niezależnych użytkowników, posiadających terminale typu konwersacyjnego, zawarte są w podsystemie.

Działanie w trybie wielodostępu umożliwia każdemu użytkownikowi żądanie realizacji programu, bez związanego z tym opóźnienia, z terminala używanego, jako jednostka wejścia / wyjścia. Edytor tekstu, BASIC, FORTRAN i programy uruchamiania dostosowane są do konwersacyjnego charakteru takiego działania. Użytkownik może kompilować programy, uruchamiać je, kontrolować ich realizację i w końcu realizować je normalnie. Dane i wyniki mogą być bezpośrednio przesyłane przez terminal albo poprzez wcześniej przygotowane zbiory albo analizowane przez edytor tekstu.

W SIRIS 8 operacje wielodostępu mają miejsce jednocześnie z konwencjonalnym przetwarzaniem wsadowym, lecz możliwości oferowane przez to ostatnie pozostają dostępne użytkownikowi konwersacyjnemu.

Program, którego realizacja wymaga zbyt wielu zasobów dla zadowalającego działania w trybie wielodostępu, może być wstawiony z terminala do kolejki przetwarzania wsadowego, dwoma sposobami:

- Jako praca priorytetowa /jeśli została ona wcześniej skatalogowana/.
- Jako normalna praca zdefiniowana przez zbiór przygotowany w trybie wielodostępu przy pomocy edytora tekstu, i zawierający w postaci obrazów kart, karty sterujące, program w języku źródłowym i dane.

Z koncepcji działania w trybie wielodostępu wynika, że system przyjmuje żądania użytkownika i próbuje spełnić je bezpośrednio, poprzez dzielenie zasobów komputera tak sprawiedliwie, jak jest to tylko możliwe. Zadania konwersacyjne, które są tworzone zgodnie z żądaniami, a nie tylko zgodnie z dostępnymi zasobami, ogólnie biorąc będą zbyt duże aby być umieszczone jednocześnie w pamięci operacyjnej, musi zatem wystąpić wymienianie /swapping/. Operacja ta obejmuje kopiowanie programu do pamięci pomocniczej, a następnie ładowanie go z powrotem do pamięci operacyjnej i aktywowanie go.

Ta rotacja zadań konwersacyjnych między różnymi poziomami pamięci tak jest zorganizowana, aby minimalizować czas odpowiedzi widoczny dla użytkownika. Dla osiągnięcia tego mogą być wprowadzone dwie koncepcje. Pierwsza, to dzielenie czasu /time slicing/ lub maksymalny czas realizacji zadania zanim będzie ono skasowane w celu przyjęcia innego zadania. Druga, to zmienne priorytety względne, w zależności od historii zadania i chwilowego środowiska, zarządzane przez oprogramowanie. Dodatkowo zadania konwersacyjne realizowane są z absolutnym wspólnym priorytetem /poziomem przerwania/, który jest wyższy niż priorytety prac konwencjonalnych zatem prace konwersacyjne mogą przerywać konwencjonalne.

Wielodostęp zorganizowany jest jako podsystem pod kontrolą SIRIS 8, jego obecność w pamięci operacyjnej niezbędna jest tylko wtedy, kiedy podjęta zostanie decyzja o pracy w trybie wielodostępu. Z tego powodu inicjowanie tej fazy powoduje aktywowanie zadań typu real time, których żądania zasobów /linie transmisji, obszar dyskowy, pamięć operacyjna/ pokrywane są przez te zadania konwersacyjne, które są wtedy tworzone.

Istnieje jednak duża elastyczność w przydziale pamięci operacyjnej, która może być używana przez zadania konwersacyjne. Wielkość tej pamięci może być regulowana w czasie działania, w celu optymalizacji całkowitej działalności systemu.

Komunikacja między programem użytkownika a usługami monitora jest taka sama w przetwarzaniu wsadowym jak i w wielodostępie, umożliwia to programom generowanym i katalogowanym w trybie przetwarzania wsadowego, realizację w trybie wielodostępu i odwrotnie.

### 2.3.2. Wymiennosc prac, przetwarzanie wsadowe / wielodostep

Programy realizowane w trybie wielodostepu uzyskują wiekszość uslug dostarczanych programom realizowanym w przetwarzaniu wsadowym. Różnice sprowadzają się częściowo do podmiiany dziennika systemu i konsoli operatora przez terminal użytkownika i częściowo do utworzenia specjalnych opcji dla procedur wejścia / wyjścia. /Lecz użycie tych opcji w programie nie zmienia możliwości normalnego działania w trybie wsadowym/. Wymiennosc ta istnieje na poziomie języka źródłowego oraz na poziomie realizacji. Nie ma żadnej specjalnej procedury systemowej wielodostepu.

### 2.3.3. Zarządzanie terminalem

Podczas działania fazy wielodostepu monitora SIRIS 8 terminale, które są do niej przydzielone, znajdują się w jednym z trzech następujących stanów, w zależności od okoliczności:

#### Wyjście - Wejście i Czuwanie

- Terminal jest w stanie wyjścia, kiedy drukuje komunikaty na polecenie SIRIS 8 lub dla programu realizowanego pod jego kontrolą. W tym stanie użytkownik nie może pisać żadnych znaków oprócz kilku specjalnych. /np. może on wypisać specjalny znak do kasowania wyjścia, które jest zbyt długie/.
  - Terminal jest w stanie wejścia kiedy SIRIS 8 lub program realizowany pod jego kontrolą oczekuje, że użytkownik napisze dla niego komunikat.
  - Terminal jest w stanie czuwania przez pozostały czas. W tym stanie użytkownik ma możliwość drukowania dwóch specjalnych znaków, które wśród innych konsekwencji, przenoszą terminal do stanu wejścia.
- Terminale mogą działać w trybie półduplex lub echoplex.
- W trybie półduplex dowolny, możliwy do wydrukowania znak, wystukany przez użytkownika, pojawia się na wydruku, użytkownik musi być także świadomy stanu swojego terminala, ponieważ drukowane znaki niekoniecznie muszą być brane pod uwagę.
  - W trybie echoplex wystukane znaki są faktycznie drukowane na żądanie sekwencji monitora przetwarzania wejścia / wyjścia. Występują one zatem na wydruku, jeśli terminal jest w stanie wejścia.

Ważne jest zatem, że użytkownik ostrzegany będzie o zmianach stanu terminala, w szczególności powinien on wiedzieć, kiedy terminal przechodzi w stan wejścia. Będzie to wskazywane przez wydrukowanie znaku przedrostka przed aktualnym przejściem do stanu wejścia.

Rolę tego znaku jest także określenie z kim użytkownik jest w kontakcie. A zatem znak "!" używany jest na poziomie SIRIS 8. Każdy kompilator konwersacyjny ma swój własny znak i podobnie, może on być określony przez programy użytkowe.

## 2.3.4. Zarządzanie zbiorami

### 2.3.4.1. Typy zbiorów

W trybie wielodostępu użytkownik może tworzyć lub manipulować:

- zbiorami przydzielonymi do jego terminala,
- zbiorami czasowymi na dysku systemowym,
- zbiorami stałymi na wolumenie kontowym.

Wolumeny te koniecznie muszą być pseudowolumenami i odpowiednie zbiory są zatem przechowywane na dysku systemowym. Czas życia zbiorów czasowych ograniczony jest do realizacji jednego rozkazu /który może obejmować kilka etapów pracy/. Wśród zbiorów stałych istnieją takie, które są chronione i takie, które chronione nie są. Zbiory chronione są w stanie używalności od sesji do sesji, podczas gdy zbiory, które nie są chronione, niszczone są na końcu sesji.

Dowolny zbiór stały utworzony w czasie sesji traktowany jest, domyślnie, jako niechroniony na końcu sesji, kiedy użytkownik ma możliwość wyraźnego zarządzania, że zbiór będzie chroniony. Eliminuje to wszelkie ryzyko wypełnienia dysku systemowego bezużytecznymi /niepotrzebnymi/ zbiorami, które niedbali użytkownicy zapomnieli oczyścić.

Dostęp do wymiennych wolumenów nie przydzielonych do systemu /taśmy magnetyczne, prywatne pakiety dyskowe/ jest w trybie wielodostępu zabroniony po to, aby nie blokować zasobów instalacji komputerowej. To samo stosuje się do urządzeń zewnętrznych wywoływanych przez adres. Pomimo tego użytkownik zawsze ma możliwość wstawiania prac do przetwarzania wsadowego.

Zbiory czasowe, które istnieją tylko w czasie trwania sesji, mogą być używane tylko przez użytkownika, lecz zbiory stałe mogą być używane do komunikowania się między użytkownikami lub być przetwarzane w pracach konwencjonalnych /i odwrotnie/.

### 2.3.4.2. Zbiory przydzielane do terminali

Aby uzyskać dostęp do swojego terminala, użytkownik przydziela zbiory do systemowych pseudourządzeń wejścia i wyjścia /etykiety IN dla klawiatury i etykiety OUT dla drukarki terminala/.

Programowanie wejścia / wyjścia na terminal wykonywane jest w taki sam sposób, jak dla konwencjonalnych urządzeń zewnętrznych, zarządzanych przez wspomaganą metodę dostępu ASAM.

Użytkownicy konwersacyjni mogą określić funkcje specjalne /takie jak tabulacja/ przez wstawienie parametrów do swojego DCB. Parametry te nie będą rozpoznawane w czasie realizacji w trybie przetwarzania wsadowego. Rekordy mają format stały, zmienny lub nieokreślony, a ich długość może osiągnąć maksymalnie 140 znaków.

Relacja między przetwarzaniem wsadowym i operacją wielodostępu dostarczana jest przez system, który w ostatnim przypadku, używa metody dostępu TSAM.

Podczas czytania TSAM steruje wydrukiem znaku przedrostka, bez interwencji ze strony programu użytkowego. Program użytkownika dostaje poprawiony tekst odebranego komunikatu bez znaku końca komunikatu. Jeśli został zadeklarowany stały format, to rekord będzie uzupełniony spacjami.

Podczas zapisu TSAM wykonuje, zgodnie z żądaniem, określone tabulacje, układ wydruku i bierze pod uwagę fizyczną ilość znaków w linii tak, aby rozdzielić komunikat na kilka linii, jeśli to konieczne i zakończyć poprzez pozycjonowanie na początku następnej linii.

Taka postać działania jest szczególnie dostosowana do konwersacji typu "linia po linii". Pozwala także na konwersację w tej samej linii, w której występuje żądanie użytkownika, a następnie odpowiedź programu.

#### 2.3.5. Opis funkcji

Faza wielodostępu wykorzystuje te same zasady co faza przetwarzania wsadowego. Funkcje, które ona zawiera mogą równie dobrze być mocno związane z odpowiednimi funkcjami tego ostatniego jak i ogólnie rozdzielone, kiedy konwersacyjny charakter prac jest podstawowy.

Rozkład funkcjonalny jest taki sam, tj.:

- Funkcje podstawowe, które zapewniają interfejs ze sprzętem.
- Funkcje zarządzania, które wyzwalają i podłączają prace.
- Funkcje komunikacji z otoczeniem, które zapewniają połączenie między programami i monitorem.
- Funkcje usługowe, które zapewniają połączenie między programami a użytkownikiem.

##### 2.3.5.1. Funkcje podstawowe

###### 2.3.5.1.1. Zarządzanie zadaniem konwersacyjnym

Zadanie konwersacyjne w pamięci operacyjnej jest w zasadzie zarządzane w taki sam sposób, jak zadanie konwencjonalne. Różnice dotyczą pewnej ilości jasno określonych punktów, w których stan zadania usprawiedliwia wyższy poziom zarządzania.

Okoliczności, w których demonstrowany jest konwersacyjny charakter zadań, są takie, w których utrzymanie zadania w pamięci operacyjnej naraża na ukaranie inne zadania tego samego typu, tj.:

- Kiedy upłynął przydzielony odstęp czasu,
- Kiedy zadanie przechodzi w oczekiwanie na zdarzenie, które może wystąpić po zakończeniu stosunkowo długiego okresu czasu.
- Kiedy zadanie przechodzi w stan oczekiwania na zasób, bez posiadania informacji, że ten zasób stanie się szybko dostępny.

W takich przypadkach zadanie dobrowolnie przekazuje sterowanie do systemu, przy pomocy podstawowych usług zarządzania i wskazuje to zadaniu systemowemu o nazwie DISPATCHER, którego rolą jest podjęcie decyzji i wykonanie operacji wymiany. Na poziomie zadania DISPATCHER pełni rolę części regulatora prac w przetwarzaniu wsadowym z pewną różnicą: w działaniu konwersacyjnym zadanie inicjowane jest nie raz, lecz wiele razy, żeby mogły pracować wszystkie zadania. DISPATCHER musi podjąć decyzję, które zadania mają być aktywowane. Dla osiągnięcia tego celu, jego wybór jest kierowany przez pewną ilość kolejek przechowujących zadanie konwersacyjne:

- kolejka zadań bieżących zawiera wszystkie zadania konwersacyjne, które są pod kontrolą sterowania zadaniami, a zatem działają normalnie;
- kolejka zadań zakończonych zawiera wszystkie zadania konwersacyjne, które są w procesie przekazującym sterowanie systemowi, lecz wciąż pozostających czasowo pod kontrolą zarządzania zadaniami;
- kolejka zadań kandydatów jest w rzeczywistości pewną ilością kolejek równą ilości priorytetów względnych. Zawiera wszystkie zadania konwersacyjne, do których może być przekazywane sterowanie;
- kolejka zadań bezczynnych zawiera wszystkie zadania konwersacyjne, które mogą stać się kandydatami w wyniku zdarzenia zewnętrznego.

#### 2.3.5.1.2. Zarządzanie pamięcią i wymianą

Kiedy wprowadzana jest faza wielodostępu SIRIS 8 pewna ilość stron pamięci operacyjnej rezerwowana jest dla niej. Regulator prac przetwarzania wsadowego zapobiega używaniu tych stron przez prace konwencjonalne. Stronice przydzielone do wielodostępu używane będą dla wymagań zadań konwersacyjnych tak, jak obszar roboczy dla zadań systemu. Ilość stron możliwych do użycia w celu spełnienia wymagań zadań konwersacyjnych ustalona jest podczas inicjowania i może być regulowana w czasie działania. Element ten będzie używany przez DISPATCHER, jako kryterium rządzące ilością zadań, które mogą być w kolejce zadań bieżących, w dowolnej chwili. Dla osiągnięcia tego DISPATCHER bierze pod uwagę ustalone wymagania każdego zadania, a nie tylko ich wymagania chwilowe.

Jedną z zasad wymiany jest, że zadanie nie będzie kopiowane na dysk, jeśli to nie jest konieczne. Zatem wymiana "out" inicjowana jest tylko przez DISPATCHER, gdy chce on aktywować zadanie kandydata, którego nie ma w pamięci i dla którego istnieje aktualnie niewystarczająca ilość wolnych stron.

Zadanie, które będzie wymieniane "out", będzie najpierw wybierane z kolejki zadań bezczynnych, następnie z kolejki zadań kandydatów, rozpoczynając od niskich priorytetów. Zatem wymiany częściowe mogą być inicjowane dla zakończenia zadań.

#### 2.3.5.1.3. Sterowanie wejściem / wyjściem terminala

Faza wielodostępu wykorzystuje usługi SGT w trybie znakowym dla uzyskania wejścia / wyjścia na terminalach.

#### 2.3.5.2. Funkcje sterowania

Celem tych funkcji jest sprawdzenie tożsamości użytkownika i wykonanie rozliczenia dla każdej pracy. Funkcje te są także używane jako kontrola żądań podczas wchodzenia do systemu tak, aby uniknąć przeładowania komputera. Kontrola ta wykonywana jest przy użyciu parametrów zasobów ustalonych podczas generowania systemu i w fazie inicjowania, a także parametrów wstawionych w czasie działania.

#### 2.3.5.3. Funkcje komunikacji z otoczeniem

Faza wielodostępu umożliwia użytkownikowi dostęp do większości usług i do zarządzania zbiorami. Obejmuje to także usługi specjalne dostarczane przez metodę dostępu TSAM.

#### 2.3.5.4. Funkcje usługowe

##### 2.3.5.4.1. Zadanie DEMON

Zadanie DEMON pełni częściowo rolę symbiontu lub telesymbiontu w wielodostępie. Zadanie to wykonuje wszystkie operacje wejścia / wyjścia na terminalach dla zadań konwersacyjnych komunikatów przygotowanych lub eksploatowanych przez metodę dostępu TSAM. Dodatkowo DEMON komunikuje się z operatorem dla inicjowania i sterowania fazą wielodostępu.

##### 2.3.5.4.2. Wykonawca

Wykonawca obejmuje funkcje, które rozpoznają komendy użytkownika, dekodują je i wtedy inicjują odpowiednie sekwencje wymagane dla spełnienia żądań. Funkcje te działają w pamięci wirtualnej zadań konwersacyjnych.

#### 2.3.6. Jezyk sterowania

Istnieją trzy typy komend:

- komendy, które mogą być wykonywane przez sekwencje w trybie monitora lub komendy pierwszej kategorii;
  - komendy łączenia z fazą przetwarzania wsadowego;
  - komendy, które mogą być realizowane wewnątrz zadań konwersacyjnych przez sekwencje w trybie użytkownika lub komendy drugiej kategorii.
- Komendy te odpowiadają realizacji podstawowych programów bibliotecznych lub programów użytkowych.

##### Komendy pierwszej kategorii

Istnieje ograniczona ilość tych komend i są one ustalone w czasie generacji. Obejmują one:

LOGIN                      Otwarcie sesji

CANCEL	Kasowanie żądania połączenia terminal - system
PROCEED	Fonowne uruchomienie jałowej sesji
DESTROY	Unicestwienie jałowej sesji
RESTART	Związane z użyciem "uwaga"
STOP	Zatrzymanie programu
LOGOUT	Zakończenie sesji
SUSPEND	Przeniesienie sesji do stanu jałowego
CLOCK	Daje datę i czas
CHARGE	Wydruk rozliczenia
MAIL	Wydruk komunikatów operatora centralnego
TAB	Symulowanie tabulacji
COMMENT	Wstawianie komentarzy
EXECUTE	Realizacja etapu pracy w przetwarzaniu wsadowym w czasie trwania sesji
CATAL	Tworzenie komend drugiej kategorii poprzez katalogowanie zestawu kart sterujących
BATCH	Inicjowanie z terminala pełnej pracy przetwarzania wsadowego

### Komendy drugiej kategorii

Komendy te są wyróżniane przez fakt, że są one możliwe do zdefiniowania i rozszerzenia zgodnie z życzeniami użytkownika. Nazwa komendy drugiej kategorii musi być traktowana jak skrót oznaczający zestaw rozkazów napisanych w języku sterowania SIRIS 8. W praktyce nazwa tej komendy będzie nazwą, pod którą został skatalogowany zestaw kart sterujących SIRIS 8 z możliwością parametryzacji. Komenda będzie wprowadzona przez napisanie jej nazwy, po której w miarę potrzeby występuje wykaz dający wartości wszystkich parametrów formalnych, dla każdego elementu wykazu mający postać:

/identyfikator = wykaz wartości/

Wśród komend drugiej kategorii rozróżniamy:

- komendy wspólne, tj. dostępne dla wszystkich użytkowników. Będzie to przypadek komend wywołujących programy, które są częścią biblioteki standardowej;
- komendy specjalne dostępne tylko dla użytkownika o danym numerze konta.

### 2.3.7. Procesory

#### 2.3.7.1. Edytor tekstu

Edytor tekstu umożliwia użytkownikowi tworzenie, rozszerzanie, kasowanie, dublowanie i przechowywanie zbiorów z konsoli. Zbiory te mogą zawierać programy oraz dane i mają organizację sekwencyjną lub indeksowo-sekwencyjną.

### Charakter przetwarzanych zbiorów

Istnieją trzy typy:

- organizacja indeksowo-sekwencyjna, format zmienny, długość rekordu ograniczona do 144 znaków;
- organizacja sekwencyjna, format zmienny, długość rekordu ograniczona do 140 znaków;
- organizacja sekwencyjna w formacie kart.

### Język sterowania

BUILD	Tworzenie zbioru
DELETE	Kasowanie zbioru
COPY	Kopiowanie zbioru
FIND	Szukanie zbioru
QUERY	Wydruk wykazu zbiorów w katalogu lub charakterystyki zbioru
END	Zatrzymanie edytora
LIST	Wydruk linii umieszczonych między dwoma granicami i spełniających dane kryterium, poprzedzony numerem linii
PRESENT	Wydruk linii umieszczonych między dwoma granicami i spełniających dane kryterium
SUBSTITUTE	Zastępowanie jednego ciągu znaków innym
DELETE	Kasowanie linii między dwoma granicami
INSERT	Wstawienie jednej lub więcej linii między dwie kolejne linie
REPLACE	Wymiana grupy linii
GO TO	Komenda skoku. Modyfikuje komendy źródłowe edytora
OPTION	Modyfikuje operacje domyślne edytora

#### 2.3.7.2. Narzędzie uruchamiania /Debugger/

Narzędzie uruchamiania jest programem służącym do uruchomienia programów. Wykonuje on następujące funkcje:

- analizuje programy wynikowe;
- kontroluje realizację programów;
- wstawia i modyfikuje elementy programu.

#### Wykaz komend

- L Wydruk, w zależności od kryterium, lub nie, wszystkich słów zawartych między dwoma granicami.
- S Modyfikowanie obszaru pamięci
- I Wstawianie
- B Definiowanie punktów przerwania

- D Kasowanie punktów przerwania
- G Kontrola realizacji /umożliwia uruchomienie programu od dowolnego adresu/

### 2.3.7.3. BASIC

Jest to język konwersacyjny, który zapewnia bezpośrednio kontrolowanie składni. Zawiera on pewną ilość wbudowanych funkcji i operacji macierzowych.

#### Wykaz instrukcji

DATA, READ, RESTORE, DEF, DIM, FOR, NEXT, GO TO, ON...GO TO, GO SUB, RETURN, IF...THEN, LET, REM, STOP, MAT, MAT SIZE, MAT READ, FILES, PRINT, INPUT.

#### Wykaz komend

xxxx	yyyy	wstawienie instrukcji
		— kod instrukcji
		— numer linii

LIST	Wydruk jednej lub więcej instrukcji
DELETE	Kasowanie jednej lub więcej instrukcji
INCLUDE	Wstawienie ciągu z programu skatalogowanego do programu, który będzie realizowany
TRACE	Komenda uruchamiania

### 2.3.7.4. FORTRAN

Język FORTRAN, wykorzystywany w trybie wielodostępu zawiera pewne różnice w stosunku do wsadowego FORTRAN-u IV CII:

- jedna linia może zawierać tylko jedną instrukcję;
- kompilacja warunkowa wymaga wypisania w kolumnach 1 i 2 znaków CI;
- w kompilowanej jednostce programu musi być zachowana następująca kolejność:
  - a/ deklaracje typu /SUBROUTINE, FUNCTION, BLOCK DATA/;
  - b/ instrukcje nierealizowalne;
  - c/ instrukcje realizowalne, FORMAT, ENTRY;
  - d/ END;
- wejściowy zbiór z programem źródłowym może mieć organizację sekwencyjną lub sekwencyjno-indeksową;
- kompilacja może odbywać się w trybie wielodostępu lub wsadowym;
- korekta tekstu źródłowego realizowana jest przy pomocy oddzielnego programu edytora tekstu, który umożliwia również analizę składni FORTRAN-u bez kompilacji.

Istotną cechą kompilatora jest rozbudowana diagnostyka kompilowanego programu podczas jego realizacji. Z kompilatora wyodrębniona jest funkcja poprawiania tekstu źródłowego.

### 2.3.8. Kontrola działania wielodostępu SIRIS 8

Decyzja o aktywacji lub dezaktywacji podsystemu wielodostępu zależy wyłącznie od operatora centralnego. W momencie aktywacji operator centralny definiuje różne zasoby, które będą przydzielone do wielodostępu wraz z parametrami operacyjnymi.

Zasoby obejmują:

- rozmiar pamięci operacyjnej;
- rozmiar pamięci pomocniczej /zbiory czasowe/;
- ilość pseudowolumenów /zbiory stałe/.

Parametrami operacyjnymi są:

- maksymalna ilość jednoczesnych użytkowników;
- maksymalny procent użycia jednostki centralnej.

Jednak te zasoby i parametry nie są ustalone na całą sesję wielodostępu i operator centralny może je modyfikować, jeśli chce. Operator może, w dowolnej chwili czasu, znać stan podsystemu wielodostępu. System dostarczy mu następującej informacji:

- wielkość pamięci operacyjnej możliwej do użycia przez zadania konwersacyjne;
- wielkość pamięci pomocniczej;
- aktualna ilość użytkowników jednoczesnych;
- aktualny procent użycia jednostki centralnej.

Operator centralny może zatem, w każdej chwili, skontrolować i zmodyfikować działanie wielodostępu tak, aby najlepiej rozdzielić zasoby systemu między przetwarzanie wsadowe i zadania konwersacyjne, umożliwiając maksymalne wykorzystanie tych zasobów.

Na przykład, może on zwiększyć lub zmniejszyć zasoby pamięci operacyjnej, zgodnie z ilością jednoczesnych użytkowników, lub zmniejszyć procent czasu jednostki centralnej przydzielonej do zadań konwersacyjnych, jeśli chce zwiększyć ilość dla przetwarzania wsadowego.

## 2.4. PAKIETY PROGRAMÓW UŻYTKOWYCH

### 2.4.1. Biblioteka matematyczna

Biblioteka matematyczna składa się z pewnej liczby modułów w formie podprogramów. Dane i wyniki przekazywane są przy pomocy argumentów formalnych. Biblioteka zawiera wiele funkcji dla różnorodnych zastosowań z zakresu analizy numerycznej. Można wyróżnić osiem zasadniczych grup:

- operacje macierzowe,

- operacje na wielomianach,
- rachunek całkowy,
- obliczenia pierwiastków,
- ekstrapolacja, interpolacja,
- równania i układy równań różniczkowych,
- wielomiany specjalne,
- minimalizacja funkcji nieograniczonych.

Tablice używane w podprogramach biblioteki matematycznej są jednowymiarowe o rozmiarach ściśle dostosowanych do potrzeb. Zajętość pamięci określona jest dla każdego z podprogramów. W skład pamięci nie są wliczane tablice posiadające argumenty formalne, których rozmiary są definiowane w programie wywołującym. Dla ułatwienia stosowania podprogramów, dla każdego z nich wyprowadzony jest kod błędu, który może być wykorzystany w przypadku stwierdzenia błędnej pracy programu.

Szczegółowy opis parametrów i argumentów formalnych podany jest w opisie każdego podprogramu. Do opisu macierzy i wielomianów używane są wektory. Większość podprogramów może wykonywać obliczenia w dwóch wersjach:

- standardowej,
- z podwójną precyzją.

#### 2.4.2. Biblioteka statystyczna

Biblioteka statystyczna składa się z dwóch części:

##### 1. Grupy podprogramów statystycznych, w skład której wchodzi:

- podprogramy statystyczne uniwersalne, które mogą być dołączane do dowolnych programów. Przekazywanie wartości pomiędzy programami i ich modułami realizowane jest poprzez argumenty formalne;
- podprogramy statystyczne przewidziane do obliczeń dotyczących prognozowania krótkoterminowego, tj. ekstrapolacji przeszłości w przyszłość.

##### 2. Programy kompletne, do których należy 9 programów realizujących zasadnicze funkcje analizy statystycznej wielu zmiennych. Programy te realizują jednocześnie obliczenia i obsługę we / wy. Zadaniem użytkownika jest dostarczenie danych do obliczeń.

W skład podprogramów statystycznych uniwersalnych wchodzi:

- podprogramy obliczenia: sumy, wartości średniej, wariancji, itp;
- podprogramy obliczania funkcji rozkładu;
- podprogramy testy: STUDENT'a, KOŁMOGOROW'a, SMIRNOW'a, de MANN-WHITNEY'a, chi kwadrat;

W podprogramach prognozowania krótkoterminowego uwzględniono:

- metodę najmniejszych kwadratów;
- ekstrapolację wykładniczą BROWN'a;

- ekstrapolację wielomianową z:
  - wielomianami ortogonalnymi,
  - wielomianami Czebyszewa,
- ekstrapolację auto-adaptacyjną.

Podprogramy te napisane są w języku FORTRAN IV. Przekazywanie danych i wyników realizowane jest poprzez argumenty formalne. Wszystkie tablice są jednowymiarowe i posiadają rozmiary zmieniające się w zależności od rozwiązywanego problemu.

Programy analizy statystycznej wielu zmiennych wykorzystują, w zależności od rozwiązywanych problemów:

- metody czysto opisowe do obliczeń korelacji, analizy czynnikowej;
- metody wykorzystujące regresję prostą, wielokrotną lub wielomianową, analizę wariancji i kowariancji ESOPE, analizę harmoniczną.

Zmienne stosowane w analizie statystycznej wielu zmiennych mogą mieć charakter ilościowy, jak np: wiek, temperatura, płeć, region. Dane przedstawione są w postaci tablicy zawierającej w kolumnach zmienne, a w wierszach obserwacje.

#### 2.4.3. OPALINE - pakiet programowania liniowego

OPALINE jest pakietem programowania liniowego opracowanym przez firmę CII - HB, który maksymalizuje funkcję liniową przy ograniczeniach liniowych i zmiennych, które same mogą być ograniczone. Pakiet ten jest napisany w języku METASYMBOL z wykorzystaniem pamięci dynamicznej.

W OPALINE wykorzystana została zrewidowana metoda simplex oraz technika odwracania macierzy w postaci iloczynowej. Pakiet programowania liniowego składa się z części rezydentnej oraz nakładek. Część rezydentna zapewnia wzajemne powiązanie elementów programu: czytanie i dekodowanie danych, zakładanie zbiorów wejściowych, optymalizację i inwersję, redagowanie wyników.

Dane wejściowe pogrupowane są w pliki kart, każdy z nich poprzedzony jest kartą sterującą definiującą rodzaj pliku. Możliwe rodzaje plików: ROWS, COLUMNS, RHS, RANGES, BOUNDS, NAME, ENDATA. Ponadto specjalne karty sterujące pozwalają na łańcuchowanie operacji i ewentualne modyfikacje parametrów standardowych.

Wprowadzanie danych możliwe jest w jednym z dwóch trybów:

##### a/ Tryb zakładania

Stosowany jest przy pierwszym przetwarzaniu danego problemu i obejmuje:

- czytanie i dekodowanie kart danych z rozbudowaną diagnostyką błędów,
- generowanie tablicy wyjściowej do optymalizacji,
- wybór dogodnej bazy wyjściowej dla optymalizacji.

b/ Tryb poprawiania

Pozwala na powtórzenie przetwarzania problemu niezakończonoego podczas poprzedniego przebiegu optymalizacji lub tego samego problemu, ale z innymi opcjami, funkcjami, wyrażeniami czy ograniczeniami.

W przetwarzaniu można wyróżnić dwa etapy:

- Odwracanie macierzy bazowej poprzez określenie formy iloczynowej inwersji z porządowaniem kolumn celem otrzymania macierzy quasi-trójkątnej, w której:
  - redukcja liczby operacji skraca czas przetwarzania,
  - minimalne wypełnienie macierzy odwróconej zmniejsza zajętość pamięci.
- Optymalizacja. Zastosowano zrewidowany algorytm SIMPLEX o następujących cechach:
  - efektywne uwzględnienie zmiennych ograniczonych,
  - ustalenie liczby kolumn przez użytkownika,
  - szczególne traktowanie wierszy odpowiedzialnych za zwyrodnienie problemu,
  - wykrywanie zjawisk stałego pętlenia się,
  - automatyczna adaptacja niektórych parametrów mających wpływ na wybór bazy.

Pod koniec działania pakietu w sposób automatyczny realizowana jest faza wydawnicza. Wyniki drukowane są w sposób czytelny dla użytkownika. W przypadku zatrzymania nienormalnego wskazywana jest przyczyna. W przypadku znalezienia rozwiązania optymalnego otrzymuje się następujące wyniki:

- nazwa funkcji, zestaw wyrażen prawostronnych i ograniczeń, liczba iteracji i odwracania macierzy, dokładność obliczenia zredukowanych kosztów związanych ze zmiennymi bazowymi, dokładność obliczania ograniczeń,
- zmienne /nazwa, wartość, stan, współczynnik kosztu, ograniczenia dolne i górne, koszt zredukowany/,
- ograniczenia /nazwa, wartość, stan, kresy dolny i górny, koszt marginalny/,

W przypadku rozwiązania nierealizowalnego drukowany jest wykaz niespełnionych ograniczeń i zmienne, których wartość znalazła się poza zakresem zmienności.

Pakiet programowania liniowego umożliwia otrzymanie wyników bez szczegółowej znajomości technik programowania liniowego.

#### 2.4.4. SOCRATE

Jest to pełny system zarządzania bazą danych. Obejmuje on:

- język do definiowania istoty danych i opisu zależności logicznych między danymi. Mogą być reprezentowane struktury hierarchiczne i sieciowe;
- język zapytań i aktualizacji, który umożliwia dostęp do bazy danych w trybie wsadowym lub konwersacyjnym.

SOCRATE zapewnia:

- możliwość dialogu z bazą przy użyciu słowników, które są właściwe dla różnych zastosowań;
- interfejs z programami wsadowymi napisanymi w innych językach programowania /assembler, COBOL, itd/. Szczególnie interesujący jest fakt, że mogą być używane programy istniejące przed utworzeniem bazy danych;
- zachowanie dyskrecji poprzez kontrolowanie dostępu do danych zawartych w bazie;
- wsparcie w trybie konwersacyjnym z terminali lokalnych lub zdalnych;
- bezpieczeństwo danych, z pomocą procedur: oczyszczania i ponownego ładowania bazy, tworzenia historii zbioru, rekonstrukcji;
- ułatwienia w tworzeniu i reorganizacji bazy danych.

System SOCRATE umożliwia wybór fizycznej organizacji bazy najlepiej dostosowanej do metod dostępu używanych w najczęściej wykorzystywanych programach.

Dostęp do bazy danych, w celu tworzenia, odpytywania i aktualizacji, udzielany jest albo użytkownikom z terminali konwersacyjnych, albo programom przetwarzanym w trybie wsadowym.

Dostęp do elementu bazy można uzyskać poprzez:

- adres elementu,
- zawartość elementu.

Dostęp poprzez adres używany jest do przetwarzania znanego elementu. Dostęp poprzez zawartość używany jest w przypadkach, gdy element lub elementy, które mają być otrzymane, nie są znane w sposób jawny. Wykaz charakterystyk, które muszą być spełnione, tworzy "filtr" używany do wyizolowania jednego lub więcej elementów, które są następnie dostarczane w odpowiedzi. Ten typ dostępu może być optymalizowany poprzez tworzenie zbiorów odwrotnych zarządzanych przez system.

Język zapytań i aktualizacji umożliwia definiowanie skomplikowanych filtrów i określenie znanych elementów. Zestaw komend może być używany do definiowania, w powiązaniu z filtrem lub określeniem, wykonywanych akcji lub "wywołań".

Język SOCRATE umożliwia także:

- grupowanie kilku akcji w tym samym wywołaniu,
- realizowanie zapytań lub aktualizacji warunkowych.

W system wbudowany jest makrogenerator umożliwiający adaptację języka SOCRATE do szczególnych wymagań użytkowników. Gdy pewne formy języka występują często, możliwe jest tworzenie wywołań standardowych /makroinstrukcji/, które mogą, w miarę potrzeby, zawierać parametry dla brania pod uwagę zmiennych elementów. Makrogenerator umożliwia katalogowanie tych instrukcji umożliwiając bezpośrednio wyrażenie tych wywołań. Dzięki temu otrzymuje się znaczne uproszczenie dialogu.

Dla prac obejmujących duże ilości danych, takich jak wydruk części bazy danych lub przygotowanie ogólnej statystyki, użytkownik może otrzymać dostęp do bazy przy pomocy programów napisanych w języku COBOL, realizowanych w trybie wsadowym. Użytkownik dysponuje zatem metodą dostępu, która umożliwia mu osiągnięcie dowolnej danej ze struktury bazy zgodnie z regułami zdefiniowanymi w języku opisu.

SOCRATE zapewnia wielodostęp konwersacyjny do bazy danych przy pomocy monitora wielokonsolowego. System dialogu zawiera także:

- kompilator języka SOCRATE,
- edytor tekstu, dla modyfikacji tekstów zawartych w bazie i dla poprawiania wywołań napisanych w języku SOCRATE.

Przy pomocy makrogeneratora mogą być przechowane wcześniej skompilowane zapytania. W przypadku wywołania są one bezpośrednio realizowane.

SOCRATE zapewnia związek logiczny danych. Blokowanie w czasie dostępu do podstawowych charakterystyk jest ukryte i zapewnia fizycznie związek logiczny danych. Użytkownik może także żądać blokowania na poziomie obiektu w celu sprawdzenia semantycznego związku logicznego. Ponadto, system SOCRATE zapewnia ochronę przed niedyskrecją poprzez kontrolowanie prawa dostępu do języka SOCRATE i do skatalogowanych makroinstrukcji.

Bezpieczeństwo bazy danych zależy od metod zastosowanych do ochrony jej przed przypadkami odpowiedzialnymi za spowodowanie zniszczenia danych w niej zawartych. System SOCRATE dostarcza dwóch typów zabezpieczeń:

- ponowne uruchomienie po wystąpieniu przypadków,
- regeneracja bazy.

SOCRATE zapewnia punkty chwilowego zawieszenia w czasie tworzenia, które eliminują konieczność powrotu do początku sesji w przypadku awarii systemu. SOCRATE prowadzi dziennik operacji realizowanych w bazie. Specjalna procedura umożliwia odtwarzanie bazy w razie awarii na podstawie poprzedniej wersji bazy i dziennika.

0233!

Bibliografia do rozdziału 2

1. Systeme d'exploitation et de programmation.
2. Moniteurs SIRIS 7/SIRIS 8. Temps partagé.
3. Systeme de gestion de fichiers SGF sous SIRIS 7/SIRIS 8 ver.B09/C09 - manuel d'utilisation et d'operations.
4. Editeur de liens SIRIS 7/SIRIS 8 - ver.B09/C09 - manuel d'utilisation et d'operations.
5. Langage de commande SIRIS 7/SIRIS 8 version B09/C09.
6. Procedures systemes SIRIS 8.
7. Bibliothèque mathématique.
8. Bibliothèque statistique.
9. Editeur de textes SIRIS 7/SIRIS 8. Temps partagé.
10. Aide a la mise au point SIRIS 8.
11. Tri - fusion sous SIRIS 7/SIRIS 8 - manuel d'utilisation et d'operations.
12. Generateur de fichiers d'essais GEF sous SIRIS 7/SIRIS 8 - manuel d'utilisation et d'operations.
13. Conversions de supports sous SIRIS 7/SIRIS 8.
14. Processeurs associes au temps partagé SIRIS 8 utilisation.
15. LPS. Makro LPS. Manuel d'utilisation.
16. Gestion de transmissions SGT sous SIRIS 7/SIRIS 8 B09/C09. Manuel d'utilisation.
17. SOCRATE sous SIRIS 7/SIRIS 8 - manuel d'utilisation.
18. COBOL ANS sous SIRIS 7/SIRIS 8 - manuel d'utilisation.
19. PL/I sous SIRIS 7/SIRIS 8 - manuel d'operations.
20. SIMULA sous SIRIS 7/SIRIS 8 - manuel d'utilisation.
21. BASIC sous SIRIS 7/SIRIS 8 en temps partagé, manuel d'utilisation et d'operations.

Druk ASG WP OXV - 9846 zam. nr 1928 z dn. 31.10.84 r.

