

DOKUMENTACJA TECHNICZNO-RUCHOWA



KRAKOWSKA FABRYKA
APARATOW POMIAROWYCH
"MERA-KFAP"

Pamięć na dyskach elastycznych
z formaterem i selektorem typ SP 45 DE.

Tom I. Część III

Opis techniczny

Spis treści :

1.	Część ogólna	1
1.1.	Zasady współpracy poszczególnych podzespołów pamięci	1
1.2.	Opis konstrukcji	2
1.2.1.	Blok sterowania pamięcią	3
1.2.2.	Blok jednostek pamięci	3
1.2.3.	Szafa	3
1.3.	Omówienie techniki realizacji pakietów	4
1.4.	Wymagane zasilanie	5
1.4.1.	Blok sterowania pamięcią	5
1.4.2.	Blok jednostek pamięci	5
1.4.3.	Realizacja zasilania	5
2.	Opis formatera	8
2.1.	Przeznaczenie i opis struktury logicznej formatera ..	8
2.2.	Opis układów płyty I/O	11
2.2.1.	Rejestr wejściowy IPREG	11
2.2.2.	Rejestr wyjściowy OPREG	11
2.2.3.	Układ logiki wyczekiwania	14
2.2.4.	Układ logiki przerwań	15
2.2.5.	Rejestr CRC	17
2.3.	Opis układów płyty CPU	17
2.3.1.	Jednostka arytmetyczno-logiczna ALU	17
2.3.2.	Rejestry główne FILE REG	18
2.3.3.	Rejestry FLAG	19
2.3.4.	Dekoder	21
2.3.5.	Zegar układu	21
2.4.	Opis układów płyty ROM/RAM	21
2.4.1.	Pamięć stała ROM	22
2.4.2.	Licznik adresów MAG	23
2.4.3.	Rejestr instrukcji ORG	24
2.4.4.	Rejestr IREG	24
2.4.5.	Rejestr IJREG	25
2.4.6.	Rejestr IREG	26
2.4.7.	Pamięć buforowa RAM	27
2.4.8.	Logika skoków	28
2.5.	Instrukcja mikroprogramu	29
2.5.1.	Instrukcja ADD	31

2.5.2.	Instrukcja ADC	31
2.5.3.	Instrukcja SUB	32
2.5.4.	Instrukcja SBB	32
2.5.5.	Instrukcja INC	33
2.5.6.	Instrukcja DEC	33
2.5.7.	Instrukcja SHF	33
2.5.8.	Instrukcja RAL	34
2.5.9.	Instrukcja COP	34
2.5.10.	Instrukcja CMP	34
2.5.11.	Instrukcja ORL	35
2.5.12.	Instrukcja ANL	35
2.5.13.	Instrukcja BOL	35
2.5.14.	Instrukcja COM	35
2.5.15.	Instrukcja JUC	36
2.5.16.	Instrukcja CALL	37
2.5.17.	Instrukcja JIS	38
2.5.18.	Instrukcja JIR	38
2.5.19.	Instrukcja IRO	38
2.5.20.	Instrukcja REJ	39
2.5.21.	Instrukcja JEX	39
2.5.22.	Instrukcja RIN	40
2.5.23.	Instrukcja RDC	40
2.5.24.	Instrukcja BRS	41
2.5.25.	Instrukcja RES	41
2.5.26.	Instrukcja MTN	41
2.5.27.	Instrukcja WTM	41
2.5.28.	Instrukcja RFM	43
2.5.29.	Instrukcja WPI	44
2.5.30.	Instrukcja WPJ	45
2.5.31.	Instrukcja NOOP	46
2.5.32.	Instrukcja STOP	46
2.5.33.	Instrukcja ADF	46
2.5.34.	Instrukcja przerwania	46
3.	Opis selektora	49
3.1.	Przeznaczenie i opis struktury logicznej selektora	49
3.2.	Zasady pracy układów pakietu selektora	51
3.2.1.	Układ sekwencyjny	51
3.2.2.	Licznik kroków	51
3.2.3.	Komparator	52
3.2.4.	Licznik stałych	52
3.2.5.	Układ adresu selektora	53
3.2.6.	Układ selekcji	54

3.2.7.	Układ bramkowania i zapisu i odczytu	54
3.2.8.	Układy generujące sygnały <u>LOAD</u> , <u>STPIN</u> , <u>STFOUT</u>	54
3.2.9.	Układy synchronizacji	55
3.3.	Opis zasady działania selektora	56
3.3.1.	Cykl sprowadzania głowicy na ścieżkę spożyznkową.....	57
3.3.2.	Sprawdzanie głowicy na ścieżkę o zadanym adresie.....	60

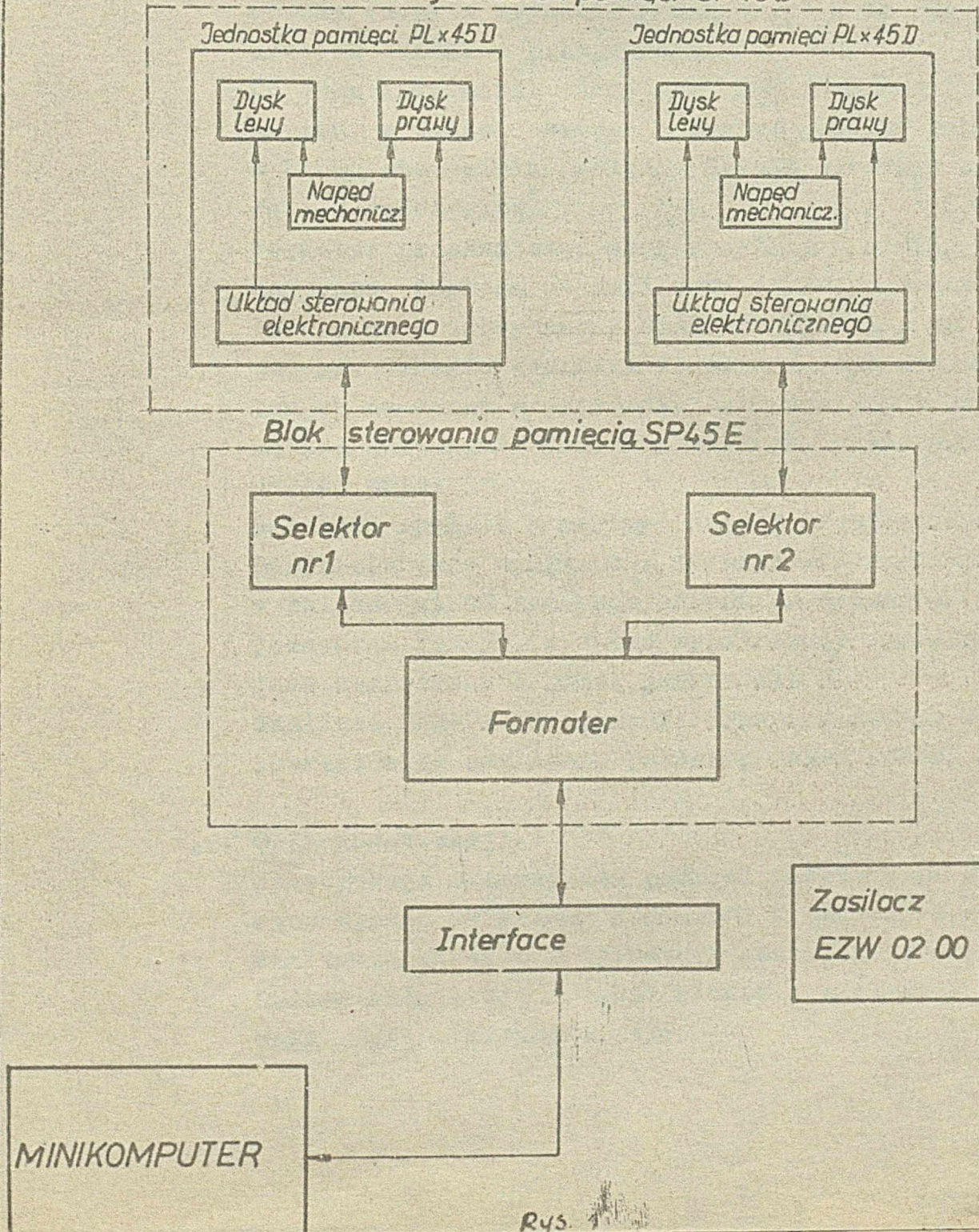
III Opis techniczny

1. Część ogólna

1.1. Zasady współpracy poszczególnych podzespołów pamięci. Pamięć SP45DE zawiera dwie jednostki pamięci PLx45D, oraz formater i dwa selektory skojarzone odpowiednio z jednostkami pamięci PLx45D.

Organizację pamięci przedstawia schemat blokowy rys. 1

Blok jednostek pamięci SP45D



Jednostka pamięci PLx45D umożliwia zapis i odczyt szeregowej informacji na dysku elastycznym w postaci ciągu impulsów. Każda jednostka pamięci PLx45D może jednocześnie obsługiwać dwa dyski elastyczne. Każdy z dysków obsługuje jedna głowica pisać-czytająca, ustawiana na wybranej ścieżce, wspólnym dla danej jednostki pamięci układem mechanicznym napędzanym przez silnik krokowy.

Układ elektroniczny jednostki pamięci PLx45D zawiera: wzmacniacze zapisu, odczytu, obwody sterujące mechanizmami pozycjonowania głowicy, napędu dysków oraz obwody odczytu sygnałów indeksowych i detekcji prawidłowych warunków pracy dysku.

Obróbka informacji odczytanej lub przeznaczonej do zapisu, a także sterowanie ruchem głowicy odbywa się w bloku sterowania pamięcią SP45DE złożonym z formatera i 2 selektorów.

Formater przekształca dane wysyłane z minikomputera na postać dogodną do zapisu na dysku, a także przygotowuje dane odczytane z dysku do wysłania do minikomputera. Ponadto zadaniem formatera jest odszukanie danych na dysku w przypadku odczytu- lub w przypadku zapisu odszukanie zadanego sektora, w którym dane należy umieścić.

Selektor spełnia w pamięci dwie zasadnicze funkcje. Rozdziela dane wysyłane z formatera przekazując je w zależności od zadanego adresu do jednej z dwóch jednostek PLx45D, a także wypracowuje sygnały sterujące bezpośrednio pracą jednostki.

Zasilacz EZW-02-00 zapewnia wymagane napięcia stałe potrzebne do zasilania poszczególnych podzespołów.

1.2. Opis konstrukcji.

Konstrukcja mechaniczna pamięci oparta jest o klasyczny system 19-calowy stosowany w budowie sprzętu minikomputerowego i aparatury kontrolno pomiarowej. Całość składa się z dwóch bloków o wielkości 5U i 7U oraz szafy o wielkości 13U.

1.2.1. Blok sterowania pamięcią.

Konstrukcją nośną bloku jest 19 calowy szkielet o wysokości 5U i głębokości 475 mm. Szkielet jest zbudowany z aluminiowych ram spawanych, wykonanych z profili oraz przykręcanych słupków usztywniających. Wnętrze szkieletu podzielone jest na dwie nierowne części. W części większej umieszczone są na prowadnicach płyty układów elektronicznych. Płyty mają wymiar 300x295 mm i są ustawione horyzontalnie.

W części mniejszej znajduje się zasilacz o budowie panelowej, przystosowany do zabudowy w module 5U. Blok z przodu jest zamykany płytą pulpitu a z tyłu płytą złącz.

Cała konstrukcja nośna bloku oraz płyty pulpitu i złącz są pokryte lakierami piecowymi zapewniającymi dobrą ochronę przed korozją i wysokie walory estetyczne.

1.2.2. Blok jednostek pamięci.

Konstrukcją nośną bloku jest 19 calowy szkielet o wysokości 7U i głębokości 475 mm.

Szkielet jest zbudowany z aluminiowych blach i płaskowników. Blachy stanowią dolną i górną płytę nośną, a płaskowniki są słupkami łączącymi i usztywniającymi. Całość jest skręcona blachowkrętami. Na płycie dolnej są umieszczone dwie pary prowadnic teleskopowych, do których montowane są dwie jednostki pamięci PLx45D.

Szkielet pokryty jest lakierem piecowym zapewniającym dobrą ochronę przed korozją i wysokie walory estetyczne.

1.2.3. Szafa.

Szafa jest konstrukcją stalową. Składa się ze spawanego szkieletu i ścian blaszanych. Szkielet wykonany jest z profili giętych, zapewniających lekkość konstrukcji przy dobrej wytrzymałości.

Wymiary szafy zgodne są z wymiarami zalecanymi dla 19 calowego systemu konstrukcji mechanicznych.

Wymiary wewnętrzne wynoszą:

szerokość x wysokość x głębokość: 490x700x600 mm.

W celu zapewnienia dobrej wentylacji we wnętrzu szafy ściany boczne posiadają perforacje a na ścianie tylnej zamontowany jest wentylator wymuszający przepływ powietrza w kierunku przed-tył.

Szafa posiada dwukołowe podwozie co umożliwia łatwe jej przemieszczanie.

Wierzch szafy pokryty jest blatem w okleinie drewnopodobnej.

Konstrukcja szafy pokryta jest lakierami piecowymi co zapewnia dobrą ochronę przed korozją i estetyczny wygląd.

1.3. Omówienie techniki realizacji pakietów.

Blok sterowania pamięcią SP45E zawiera pięć pakietów: trzy płyty formatera: SP45 I/O, SP45 ROM/RAM, SP45 CPU oraz dwie płyty obu selektorów SP45 SEL.

Pakiety te zrealizowane są w postaci płyt o wymiarach 295x300 mm dwustronnie drukowanych. Na płytach tych zamontowane są cyfrowe układy scalone TTL małej, średniej i wielkiej skali integracji oraz inne elementy elektroniczne: kondensatory, rezystory, diody, tranzystory itd.

Na wszystkich pięciu wymienionych płytach zamontowane są po dwa złącza pośrednie ELTRA 831, a na płytach SP45 I/O i SP45 SEL dodatkowo po jednym złączu bezpośrednim ELTRA 801.

Złącza pośrednie służą do połączenia poszczególnych pakietów z płytą łączącą, na której naniesione są ścieżki realizujące połączenia między płytami.

Złącza bezpośrednie wykorzystywane są przy testowaniu pakietów.

1.4. Wymagane zasilania

1.4.1. Blok sterowania pamięcią SP45E:

Formater:

płyta SP45 I/O zasilana napięciem $5V \pm 0,25V$ -
 - max. pobór prądu 1,5A

płyta SP45 CPU zasilana napięciem $5V \pm 0,25V$ -
 - max. pobór prądu 1,5A

płyta SP45 ROM/RAM zasilana napięciem $5V \pm 0,25V$ -
 - max. pobór prądu 1,5A

Selektor nr 1:

płyta SP45 SEL zasilana napięciem $5V \pm 0,25V$ -
 - max. pobór prądu 1,5A

Selektor nr 2:

płyta SP45 SEL zasilana napięciem $5V \pm 0,25V$ -
 - max. pobór prądu 1,5A

1.4.2. Blok jednostek pamięci SP45D.

Silnik synchroniczny prądu zmiennego napędzający dyski zasilany jest napięciem: 220V /+10%, -15%/; 50Hz.

Maksymalne zapotrzebowanie mocy przez silnik wynosi 50 VA.

Silnik krokowy prądu stałego - zasilany jest napięciem $48V \pm 5\%$, prąd średni 1,5A, prąd maksymalny 4,5A.

Płyta logiki jednostki pamięci PLx45D - zasilana jest napięciem: $+12V \pm 5\%$, max. pobór prądu 300 mA

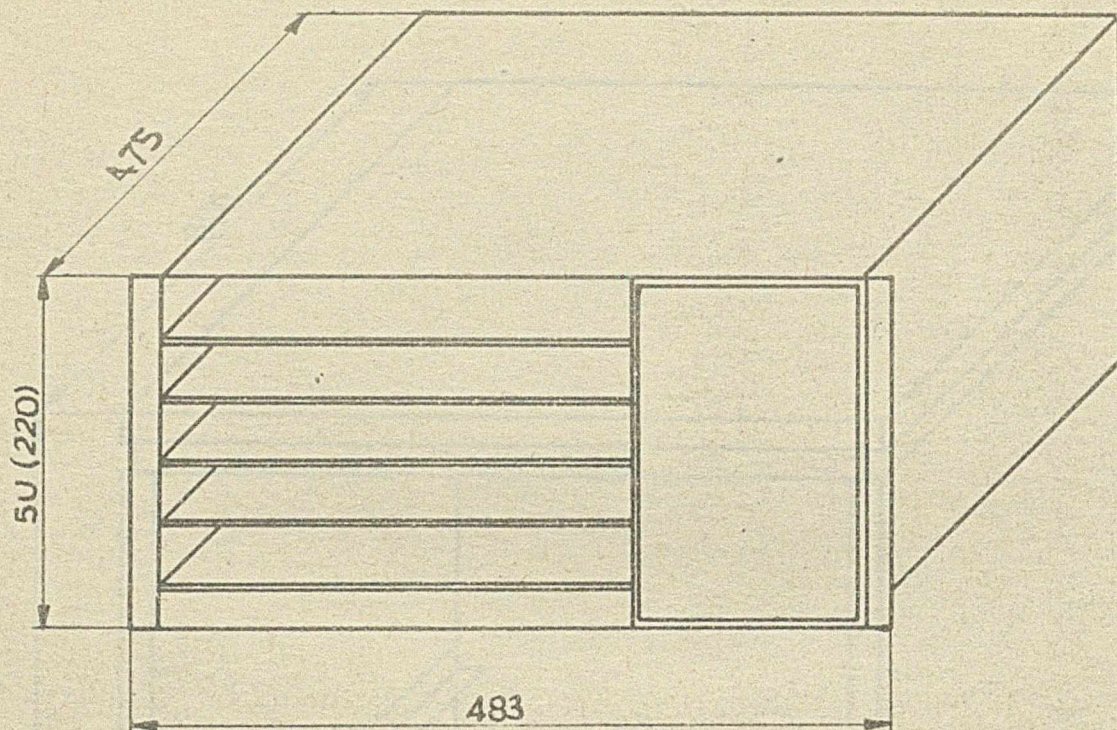
$+5V \pm 5\%$	"	"	"	1,3 A
$-12V \pm 10\%$	"	"	"	100 mA.

1.4.3. Realizacja zasilania.

Napięcia stałe o wymaganych parametrach uzyskiwane są w zasilaczu EZW-02-00 i rozprowadzane wiązkami do odbiorników we wszystkich podzespołach pamięci.

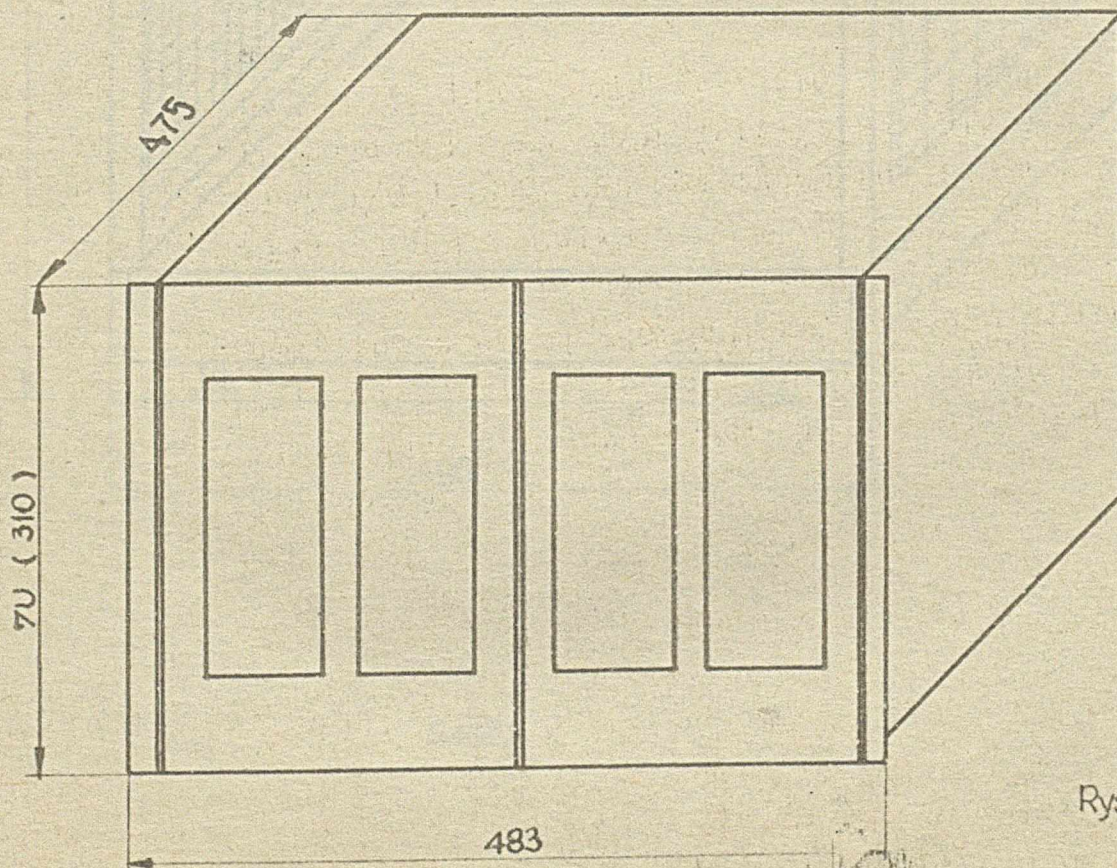
Napięcie zmienne 220V w przypadku wykonania III doprowadzane jest do listwy rozdzielczej i stąd dalej do wszystkich odbiorników.

W przypadku wykonania I i II napięcie 220V należy doprowadzić do poszczególnych odbiorników z magistrali zasilającej szafę, wewnątrz której pamięć jest montowana.



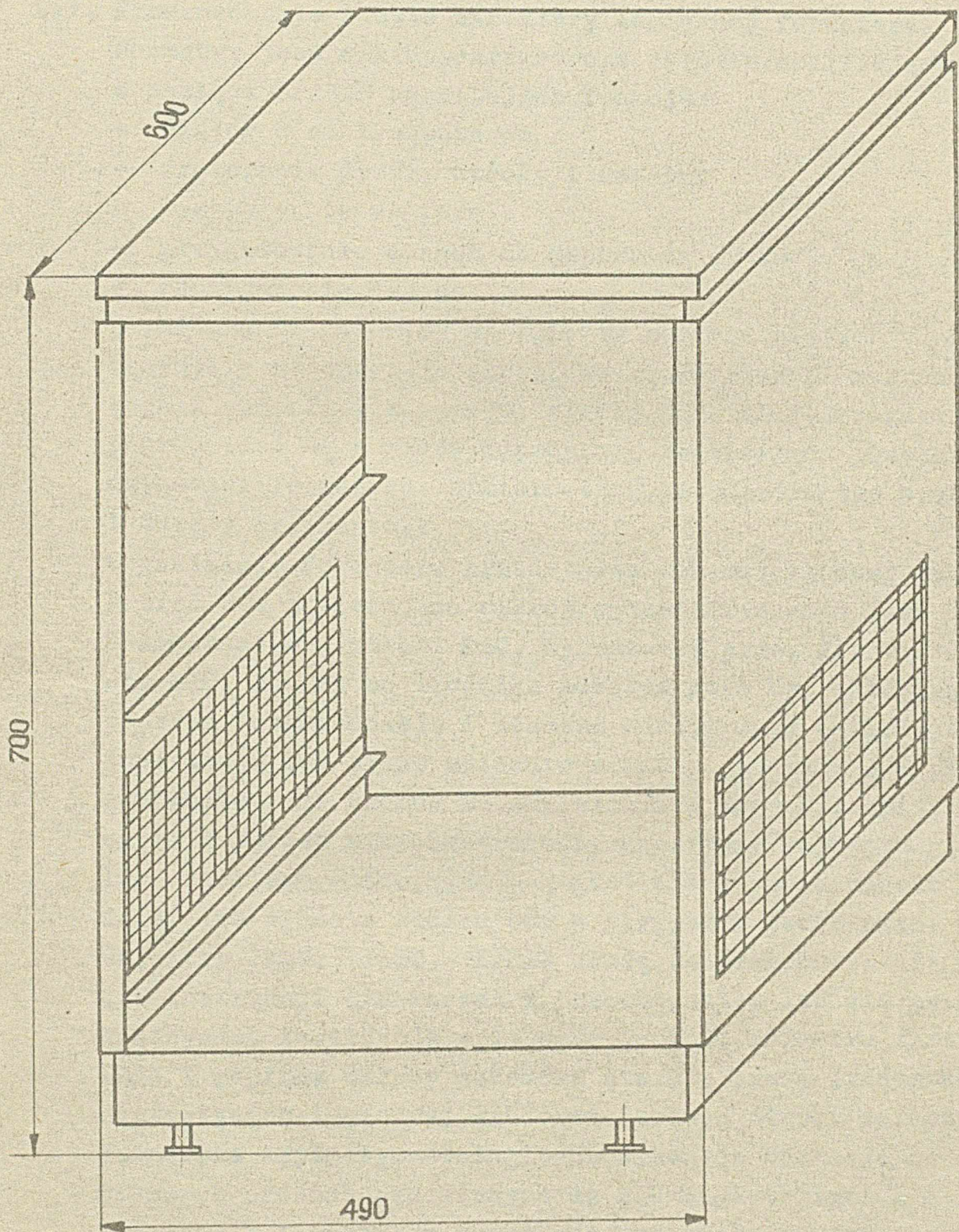
Blok sterowania pamięcią

Rys. 2



Rys. 3

Blok jednostek pamięci



Szafa

Rys. 4

2. Opis formatera

2.1. Przeznaczenie i opis struktury logicznej formatera.

Formater jest mikroprogramowanym układem, realizującym w pamięci SP45DE następujące funkcje:

- dialog z minikomputerem
- wybieranie dysku, strony i ścieżki
- wyszukiwanie sektora
- przygotowanie danych do zapisu na dysku
- odczytywanie danych
- ustawianie głowicy na ścieżce spoczynkowej.

Wszystkie te operacje określone są 16-bitowymi mikroinstrukcjami zawartymi w pamięci stałej ROM. Mikroinstrukcje pogrupowane są w mikropodprogramy realizujące poszczególne czynności formatera. Mikroinstrukcja określa typ operacji i adresy argumentów.

Organizację formatera przedstawia schemat blokowy rys. 5. Pamięć ROM zawierająca mikroprogram adresowana jest stanem 8-liniowej magistrali MAC, wymuszonym przez licznik adresów MAC. Zawartość licznika adresów może być zmieniana poprzez inkrementację i wówczas odczytywane są kolejne instrukcje lub przez wpisanie stanu magistrali A w przypadku instrukcji skoku. Do magistrali A może być wpisywana część adresowa mikroinstrukcji, zawartość rejestru głównego lub wejściowego dostępna w magistrali Y /adresowanie pośrednie/ lub wyjście kodera GOD w przypadku przerwania.

Rejestry JREG, IREG, IJREG służą do przechowywania adresu dla instrukcji przełączeń z przechowaniem lub dla przerwania. Wykonywana instrukcja wpisywana jest do rejestru instrukcji OREG i poprzez układy dekodera steruje pracą jednostki arytmetyczno-logicznej ALU oraz poszczególnych rejestrów. Jednostka arytmetyczno-logiczna wykonuje operacje na osmiobitowych słowach przesłanych do ALU magistralami X i Y. Magistralą X może być przesłana zawartość rejestru głównego FILLREG, magistralą Y zaś zawartość rejestru głównego lub wejściowego IPREG. Wynik operacji poprzez magistralę S może być umieszczony w rejestrze głównym lub wyjściowym. Do przechowywania danych przeznaczonych do zapisu na dysku lub odczytanych z dysku służy pamięć buforowa RAM. Dane z interfejsu do pamięci buforowej przesyłane są magistralą Y, poprzez ALU, magistralą S do rejestru głównego.

a następnie magistralą X do RAM. Wyjście danych realizowane jest poprzez magistralę X, ALU i magistralę S.

Pamięć RAM adresowana jest stanem magistrali Y /zawartość rejestru głównego lub wejściowego/.

Wymiana informacji między formaterem a minikomputerem i selektorami odbywa się poprzez rejestry: wejściowy IPREG i wyjściowy OPREG.

Dane przeznaczone do zapisu na dysku przechodzą przez układ CRC /cykliczna kontrola redundancyjna/. Wyznaczony w tym układzie znak kontrolny dopisywany jest do informacji. Podczas odczytu przejście danych wraz ze znakiem CRC przez układ CRC umożliwia wykrycie ewentualnych przekłamań. Przebieg programu może być uzależniony od zewnętrznych sygnałów. Do tego celu służą układy logiki przerwań i logiki wyczekiwania.

Układ logiki przerwań składa się z rejestru maskującego MASQ.REG, rejestru przerwań P.REG oraz kodera COD.

Wejścia przerwań umożliwiają realizowanie podprogramów przy zatrzymanym programie głównym.

Układ logiki wyczekiwania składa się z rejestru synchronizującego REGW i multiplexera WAIT. Pozwala on na zsynchronizowanie przebiegu programu z zewnętrznym sygnałem asynchronicznym.

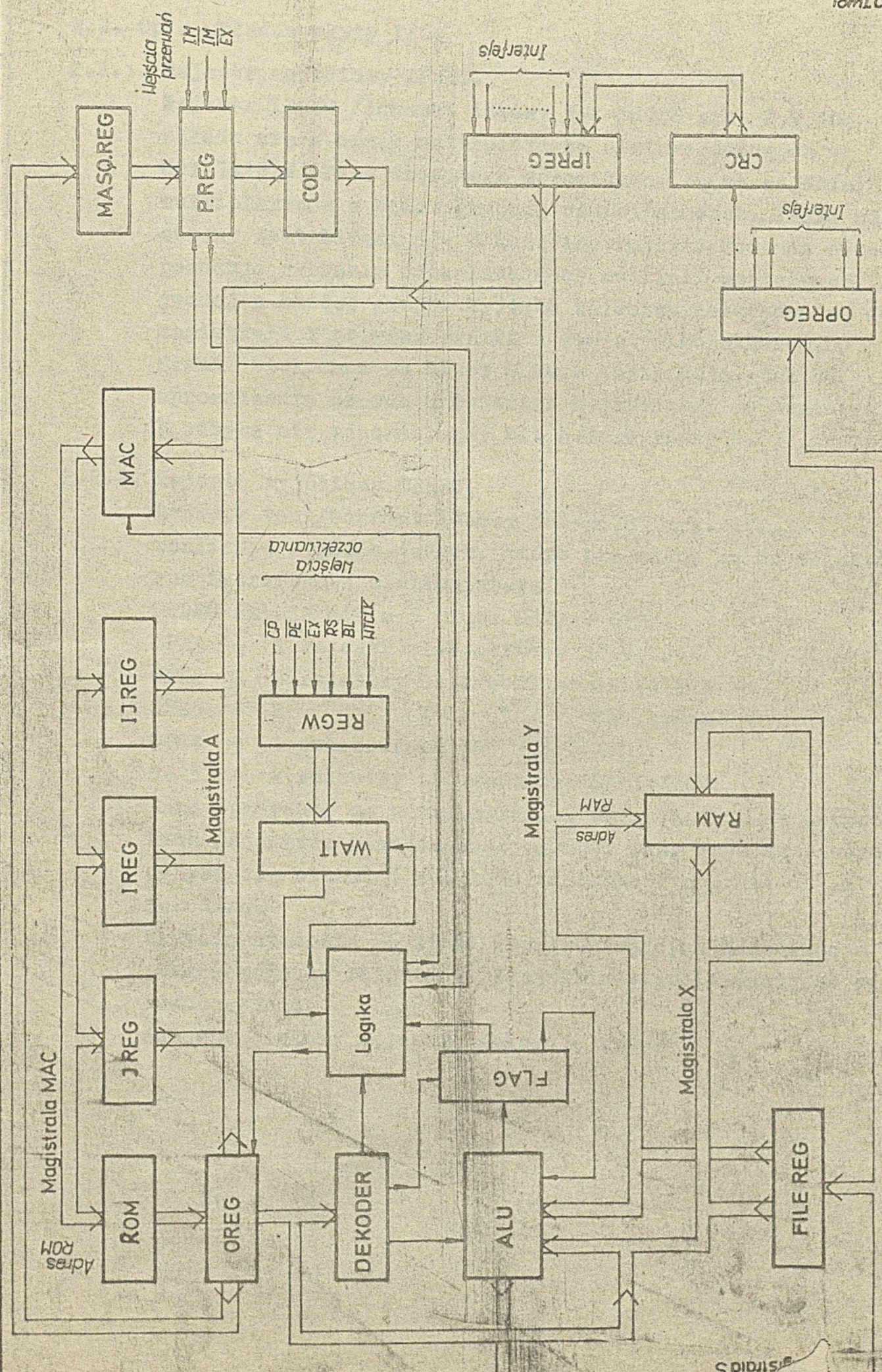
Fizycznie formater stanowią trzy pakiety:

- Płyta SP45 I/O
- Płyta SP45 CPU
- Płyta SP45 ROM/RAM.

Płyta SP45I/O zawiera rejestry OPREG i IPREG, układy przerwania, wyczekiwania oraz rejestr CRC.

Na płycie SP45 CPU znajduje się jednostka arytmetyczno-logiczna ALU, rejestry główne FILE.REG, rejestry FLAG, dekodery, oraz zegar układu.

Płyta SP45 ROM/RAM zawiera pamięć ROM i RAM, licznik adresów MAC, rejestry JREG, IREG, IJREG, rejestr instrukcji OREG oraz układ logiki ścieżek...



Schemat blokowy formatera

Rys. 5

2.2. Opis układów płyty I/O.

2.2.1. Rejestr wejściowy IPREG.

Rejestr IPREG /schemat ideowy 3-L-36785 ark. 8,9,10/ składa się z ośmiu multiplexerów ośmiowejsiowych UCY 74151N. Dane wejściowe wprowadzane są na wejścia multiplexerów w ośmiu grupach ośmioliniowych. IPREG adresowany jest bitami 5 + 8 kodu instrukcji. Podanie adresu powoduje wybranie odpowiadającej mu linii wejściowej, po jednej z każdej grupy. Wyjścia rejestru dołączone są do magistrali Y poprzez bramki z otwartym kolektorem. Bramki blokowane są bitem 8 kodu instrukcji, tak że wprowadzenie danych z rejestru wejściowego do magistrali Y odbywa się wówczas, gdy bit 8 jest równy 1.

2.2.2. Rejestr wyjściowy OPREG.

Rejestr ten /schemat ideowy 3-L-36785 ark. 1,2,3/ realizuje wyjście danych, które przesłane są do selektora lub interfejsu minikomputera.

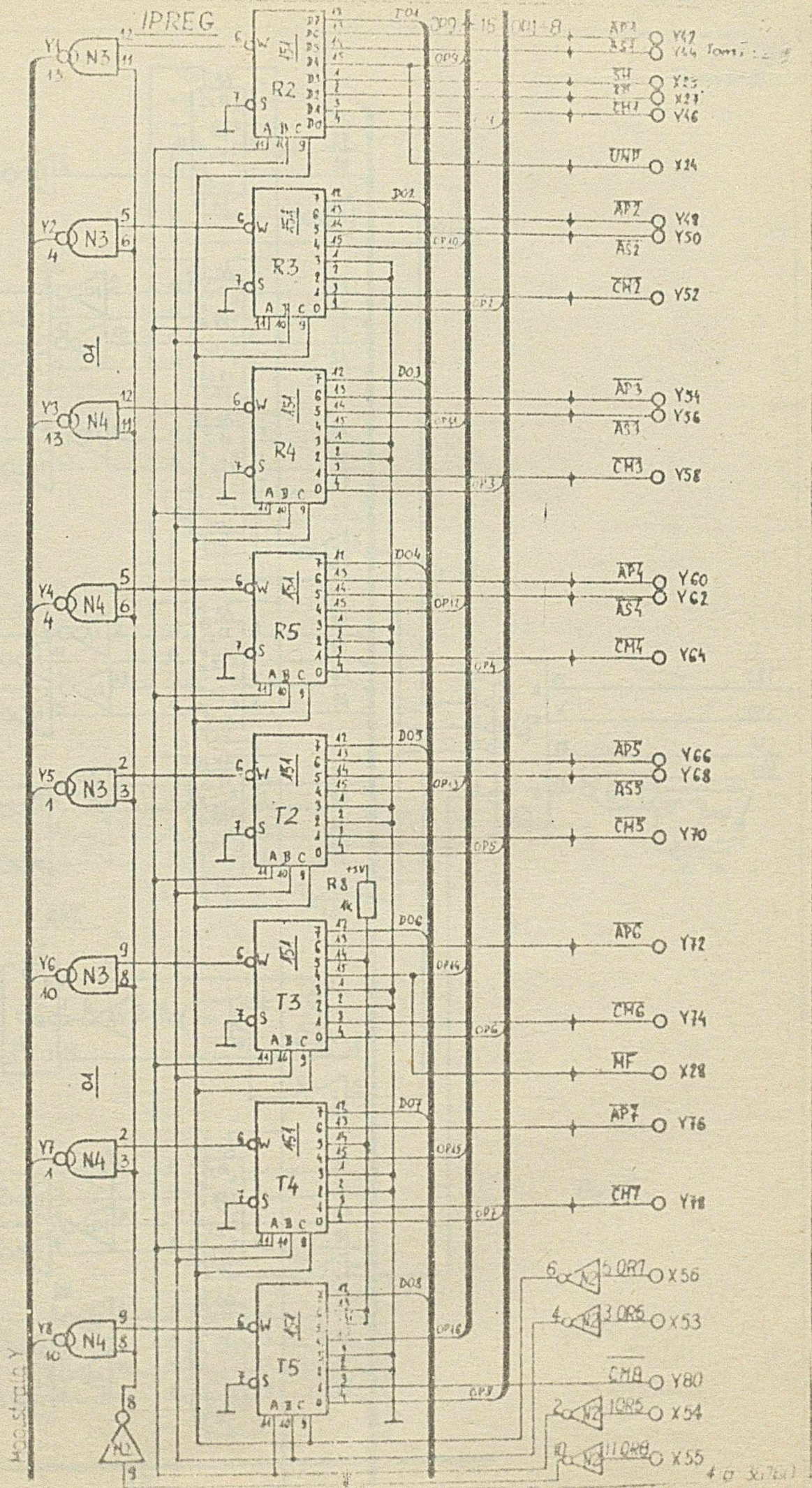
OPREG składa się z 5 grup: OPR1 + OPR5.

- OPR1 - 1 rejestr czterobitowy SN 74194N
- OPR2 - 2 rejestry czterobitowe SN 74194N
- OPR3 - 2 " " SN 74194N
- OPR4 - 1 przerzutnik JK UCY 7473N
- OPR5 - 2 rejestry czterobitowe SN 74194N.

Dane wpisywane są z magistrali S równocześnie do wszystkich rejestrów zaadresowanej grupy, przy czym adresowanie odbywa się bitami 1 + 4 kodu instrukcji poprzez układ dekodera.

Wyjścia stanowią 26 linii reprezentujących zawartość poszczególnych rejestrów. Wszystkie wyjścia dostępne są równocześnie.

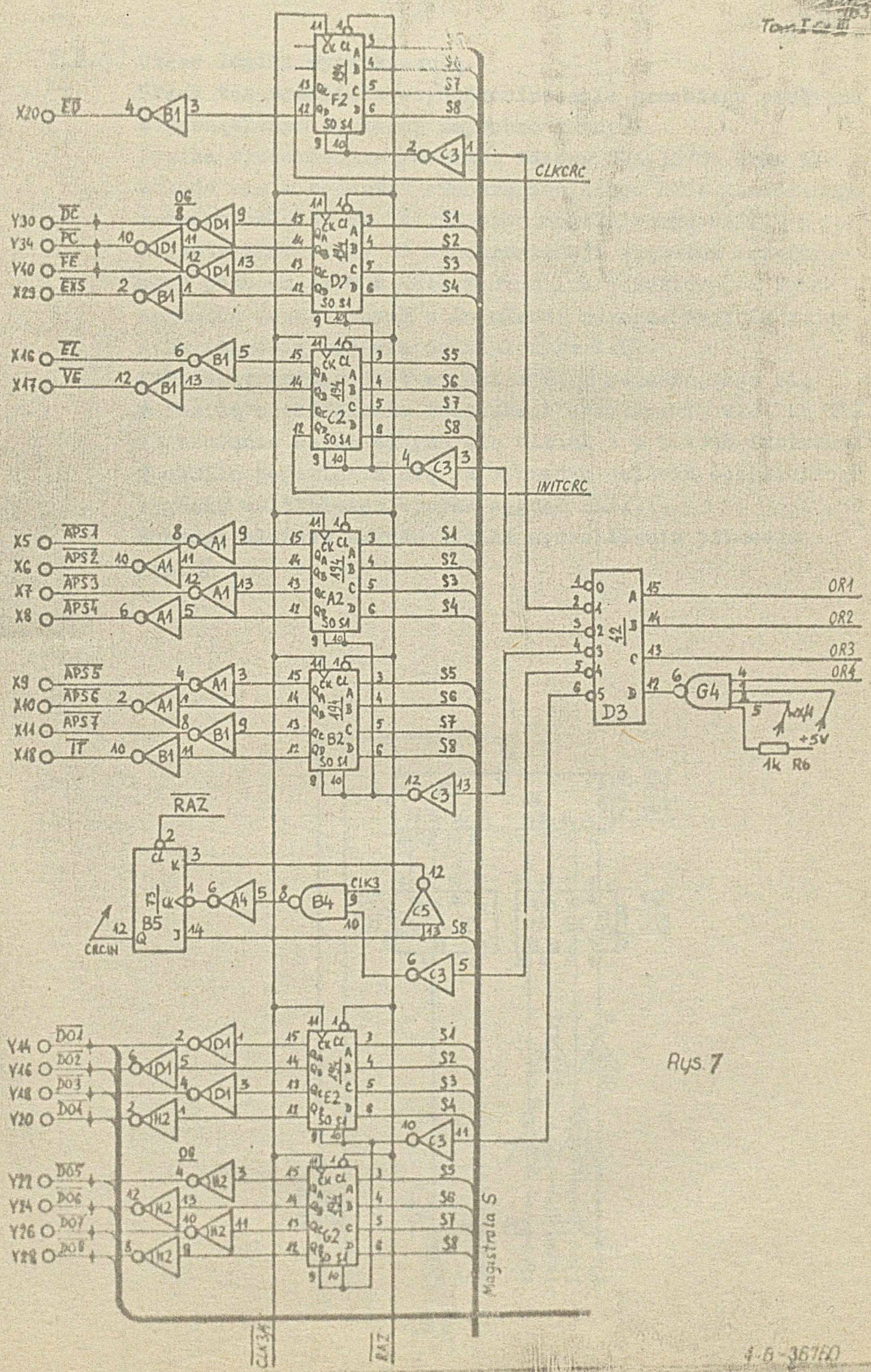
Schemat blokowy rejestru pokazuje rys.7



Rys 6

4001, 4002, 4003, 4004, 4005, 4006, 4007, 4008, 4009, 4010, 4011, 4012, 4013, 4014, 4015, 4016, 4017, 4018, 4019, 4020, 4021, 4022, 4023, 4024, 4025, 4026, 4027, 4028, 4029, 4030, 4031, 4032, 4033, 4034, 4035, 4036, 4037, 4038, 4039, 4040, 4041, 4042, 4043, 4044, 4045, 4046, 4047, 4048, 4049, 4050, 4051, 4052, 4053, 4054, 4055, 4056, 4057, 4058, 4059, 4060, 4061, 4062, 4063, 4064, 4065, 4066, 4067, 4068, 4069, 4070, 4071, 4072, 4073, 4074, 4075, 4076, 4077, 4078, 4079, 4080, 4081, 4082, 4083, 4084, 4085, 4086, 4087, 4088, 4089, 4090, 4091, 4092, 4093, 4094, 4095, 4096, 4097, 4098, 4099, 4100

40 X 1761



Rys 7

2.2.3. Układ logiki wyczekiwania.

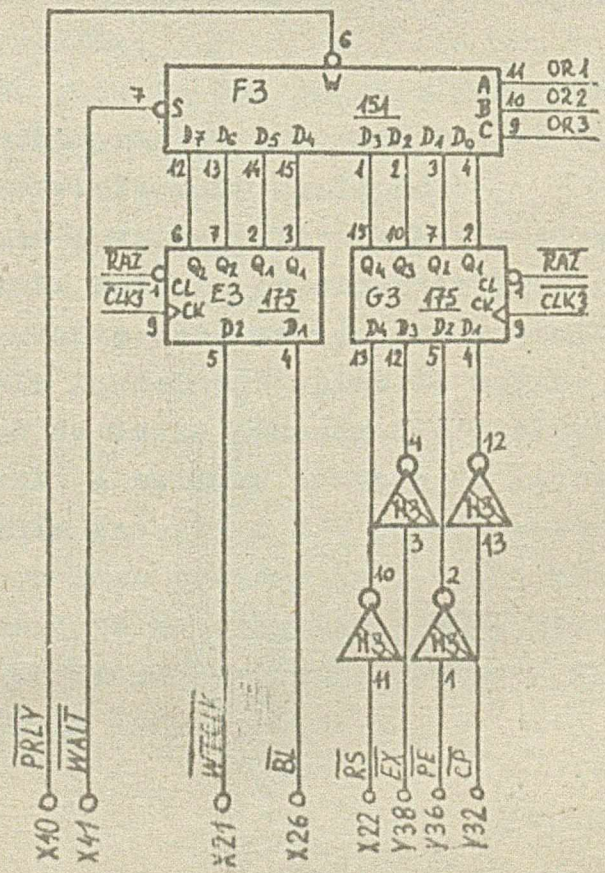
Układ ten umożliwia zsynchronizowanie przebiegu programu z zewnętrznym sygnałem asynchronicznym.

Logika wyczekiwania /schemat ideowy 3-E-36785 ark. 5/ składa się z rejestru synchronizującego REGW zbudowanego z dwóch układów SN 74175N oraz multiplexera WAIT.

Rejestr REGW służy do synchronizacji sygnałów zewnętrznych pochodzących z interfejsu a lub selektora, a także sygnałów wewnętrznych z impulsami zegara. Wyjścia rejestru stanowią wejścia multiplexera WAIT.

WAIT uaktywniany jest sygnałem WAIT pojawiającym się z chwilą zdekodowania instrukcji oczekiwania WPJ lub WPI. Adresowanie wejść odbywa się bitami 1 + 3 kodu instrukcji. Z chwilą pojawienia się na wybranym wejściu oczekiwanego sygnału multiplexer wysyła sygnał PRLY.

Schemat blokowy układu logiki wyczekiwania pokazuje rys. 8



RYS. 8

2.2.4. Układ logiki przerwań.

157
163
Tom I Cx III

Układ logiki przerwań /schemat ideowy 3-L-36785 ark. 6/ umożliwia przerwanie wykonywanego programu w celu zrealizowania podprogramu w zależności od sygnałów IM, IIM, EX.

Układ składa się z rejestru maskującego MASQ.REG, rejestru przerwań P.REG, kodera COD oraz dwóch przerzutników INTD i INT. Rejestry MASQ.REG i P.REG stanowią układy SN 74194N, INTD i INT to przerzutniki JK UCY 7473N, koder COD stanowi zespół bramek.

Każde przerwanie zabronione jest sygnałem MIN. Ustawienie MIN w stan niski powoduje wpisanie bitów 1 + 4 kodu instrukcji do rejestru maskowania MASQ.REG. Wprowadzone bity określają linie przerwań, które będą mogły być realizowane. Pojawiające się przerwanie zostaje wpisane do rejestru przerwań. Z chwilą zdekodowania go przez koder COD pojawia się sygnał INTG powodujący wstrzymanie następnego wpisu do P.REG.

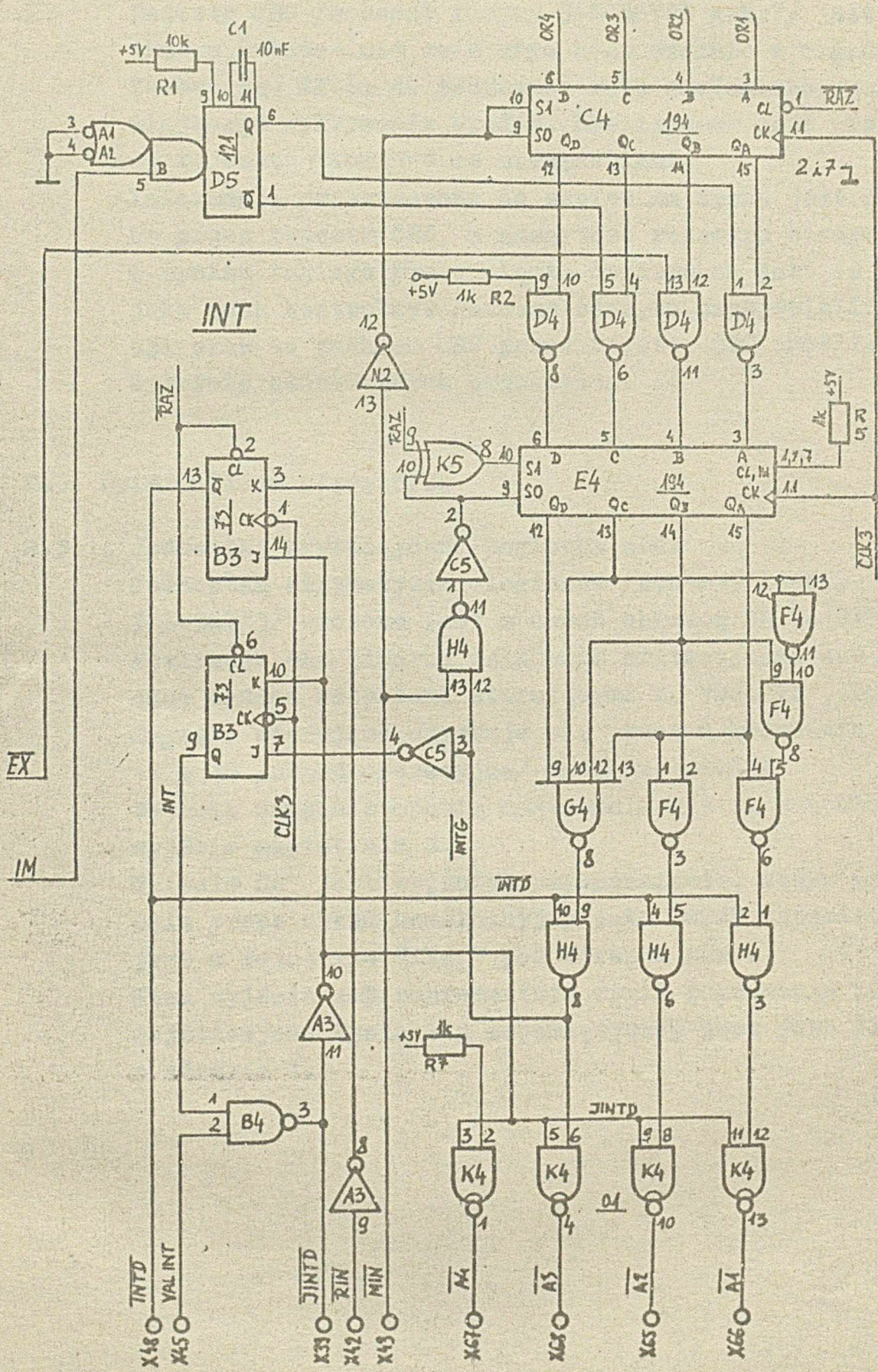
Koder COD koduje cztery linie wejściowe na trzy linie wyjściowe binarne, ustalając między nimi priorytet. Linie wyjściowe stanowią trzy pierwsze bity kodu adresu. Czwarty bit jest równy 1.

Sygnał uwzględnienia przerwania INTG ustawia przerzutnik INT /zgłoszenie przerwania/, który w iloczynie z VALINT daje sygnał JINTD.

Sygnał VALINT pojawia się podczas określonych instrukcji i warunków wykonywanego programu i jest to jedyna chwila, w której przerwanie może nastąpić.

JINTD powoduje podanie czterech bitów adresu na magistralę A, równocześnie ustawia przerzutnik INTD definiujący stan obsługi przerwania oraz zeruje przerzutnik INT. Powoduje to wstrzymanie następnego wpisu do magistrali A i do kodera COD aż do czasu zakończenia obsługi przerwania. Stan magistrali A wpisany zostaje do licznika adresów MAC. Poprzednia zawartość MAC przechowywana jest w rejestrze IREG. Obsługa przerwania powinna być zakończona instrukcją powrotu po przerwaniu RIN, która powoduje wyzerowanie przerzutnika INTD oraz przepisanie zawartości rejestru IREG do MAC.

16/63
Tom I Cz III



Rys. 9

2.2.5. Rejestr CRC.

Rejestr CRC /schemat ideowy 3-2-36785 ark.7/ jest dodatkowym układem nie związanym bezpośrednio z organizacją formatera. Służy do tworzenia kodu wielomianowego umożliwiającego wykrywanie błędów. Realizowany jest przy pomocy rejestru przesuwającego ze sprzężeniami.

Informacja przeznaczona do zapisu na dysku jest przesuwana przez rejestr CRC, a zawartość rejestru otrzymana w wyniku takiego przesunięcia dopisywana jest do danych jako znak kontrolny. Podczas odczytu przejście informacji wraz ze znakiem CRC przez rejestr CRC umożliwia wykrycie ewentualnych przekłamań.

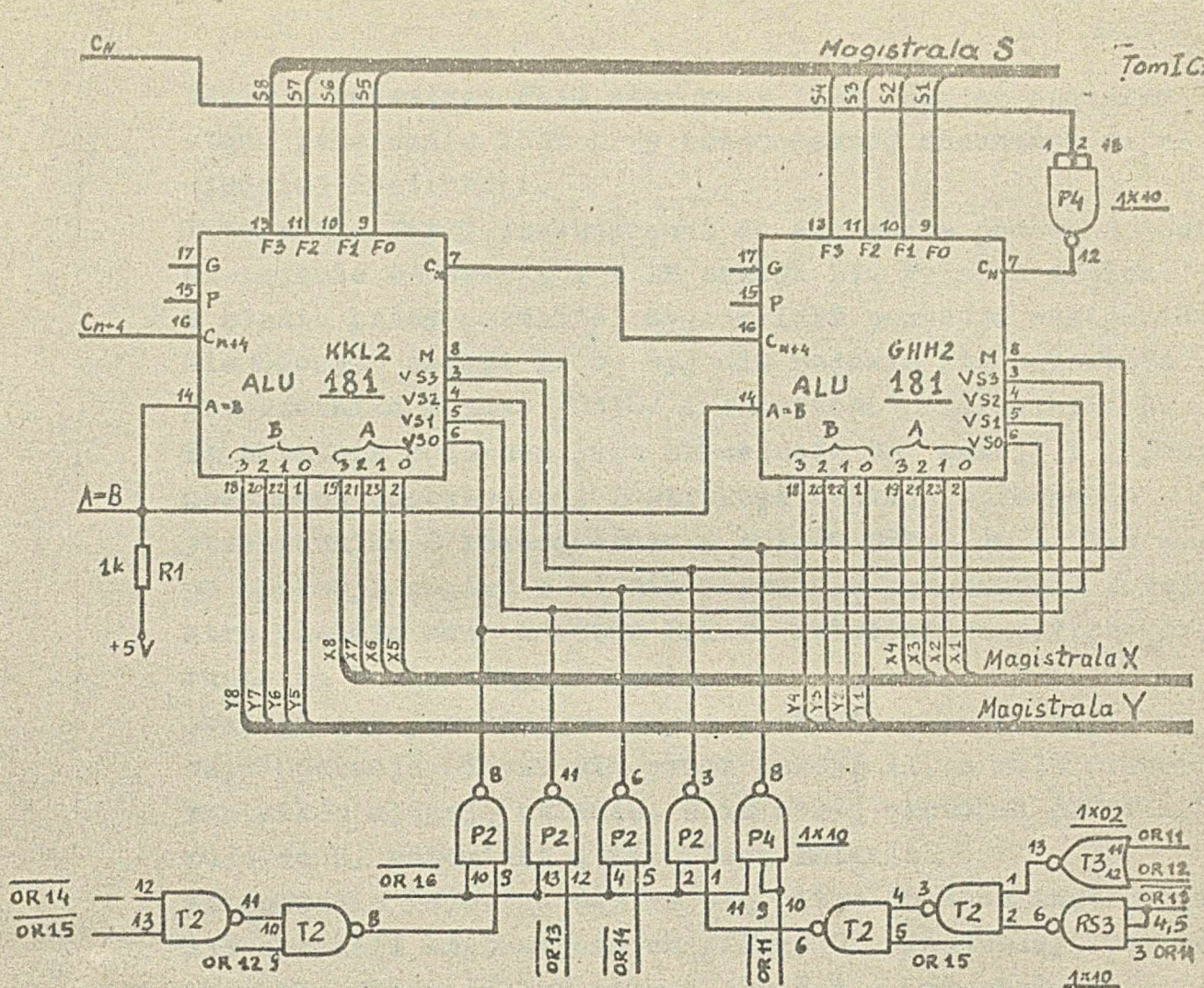
2.3. Opis układów płyty CPU.

2.3.1. Jednostka arytmetyczno-logiczna ALU.

Jednostka arytmetyczno-logiczna /schemat ideowy 3-2-36783/ złożona jest z dwóch obwodów SN 74181N. Realizuje ona binarne działania arytmetyczne lub logiczne zadane wejściami sterującymi M, VSO, VS1, VS2, VS3. Sygnały sterujące uzyskuje się przez dekodowanie bitów 11 + 16 /część operacyjna/ kodu instrukcji. Wejścia danych stanowią magistrale X i Y, natomiast wyjście magistrala S.

Wyjście Cn jest wyjściem przeniesienia, które po przejściu przez układ kombinacyjny jako DSTAT zapamiętywane jest w rejestrze "flag" pod adresem zero.

Stan wyjścia A=B reprezentuje wynik porównania wielkości wejściowych. Wynik ten zapamiętywany jest jako flaga o adresie 1.



2.3.2. Rejestry główne FILE REG.

Rejestr główny /schemat ideowy 3-L-36785 ark.5/ zbudowany jest z czterech obwodów SN 74172N. Każdy obwód to osiem dwubitowych rejestrów skojarzonych z dwoma multiplexerami i dwoma demultiplexerami tworzącymi sekcję I i sekcję II. Wejścia rejestru stanowią magistrale X i Y, a wyjście magistrala S.

Wpisywanie do rejestru odbywa się poprzez sekcję I uaktywnioną sygnałem WX. Adresowanie jednego z ośmiu rejestrów dokonywane jest bitami 1 + 3 kodu instrukcji. Odczyt może odbywać się przez sekcję I lub II względnie przez obie równocześnie. W przypadku odczytu przez sekcję I, rozkazem odczytu jest sygnał RX, a dane są wówczas wpisywane do magistrali X. Adresowanie rejestru, z którego odczytywane są dane odbywa się bitami 1 + 3. Odczyt przez sekcję II jest uaktywniany sygnałem RY. Dane są wówczas wprowadzane magistralą Y, a adresowanie odbywa się bitami 5 + 7.

2.3.4. Dekoder.

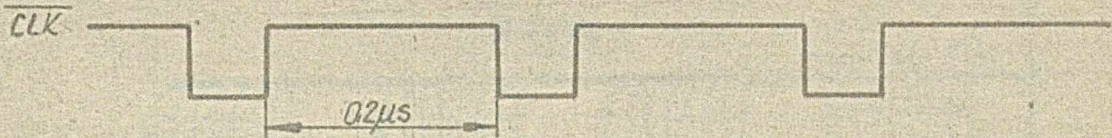
Dekoder dekoduje bity 11 + 16 /część operacyjna/ kodu instrukcji na sygnały sterujące pracą poszczególnych rejestrów.

Instrukcje arytmetyczno-logiczne i instrukcje przesyłania danych między rejestrami dekodowane są przez układ UCY 7442N oraz zespół bramek związanych z poszczególnymi rejestrami.

Instrukcje dotyczące zapisu i odczytu bufora, instrukcje związane z przetwarzaniem i wyczekiwaniem oraz instrukcje skoków dekodowane są przez element SN 74154N.

2.3.5. Zegar układu.

Układ ten składa się z generatora impulsów prostokątnych stabilizowanego rezonatorem kwarcowym. Częstotliwość impulsów wynosi 10 MHz. Impulsy te są dzielone przez dwa przy pomocy przerzutnika JK UCY 7473N, a następnie formowane są w układzie bramek w sygnał CLK o częstotliwości 5 MHz i wypełnieniu 1:3.



Rys. 13

Sygnał ten służy jako zegar do synchronizacji pracy wszystkich rejestrów i przerzutników formatera. Ponadto płyta CPU zawiera układ dzielnika, dzielącego sygnał CLK przez dwadzieścia. W wyniku podzielenia otrzymuje się ciąg impulsów WCLK o częstotliwości 250 kHz, który służy do synchronizacji selektora.

2.4. Opis układów płyty ROM/RAM.

2.4.1. Pamięć stała ROM.

Pamięć ROM /schemat ideowy 3-2-36784 ark. 1 i 2/ zawiera zestaw 256 instrukcji 16-bitowych stanowiących mikroprogram.

Zrealizowana jest przy pomocy czterech układów scalonych SN 74S387N /93426/.

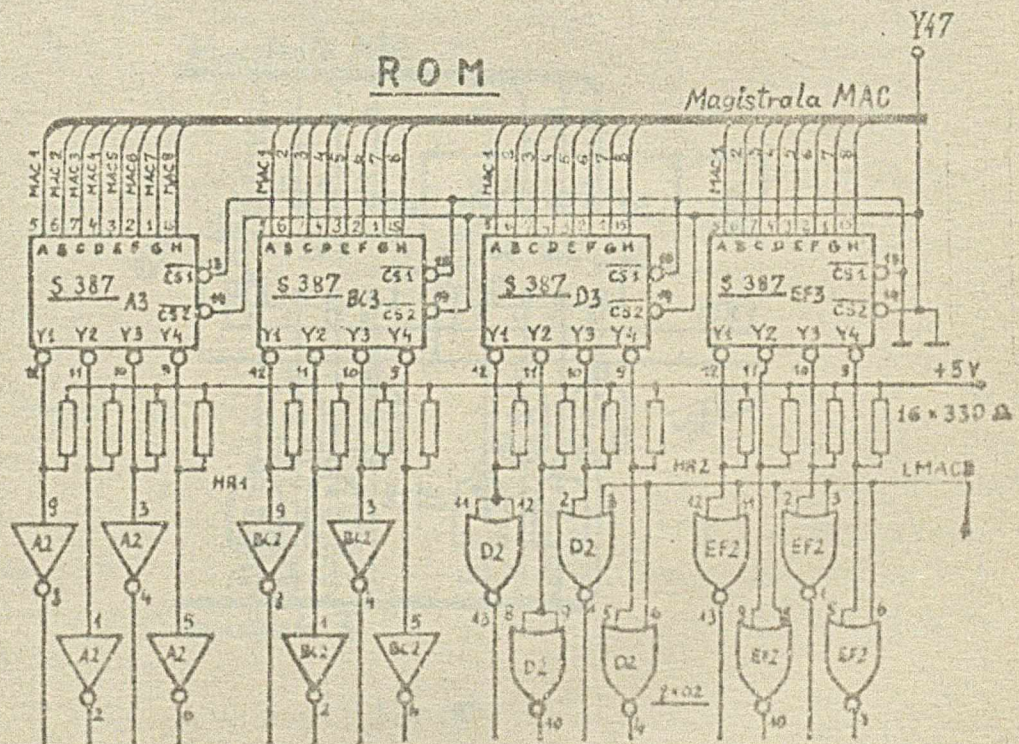
Czas dostępu do pamięci wynosi 40 ns.

Pamięć ROM adresowana jest licznikiem adresów, który jest inkrementowany przy każdej mikroinstrukcji, lub do którego wpisuje się część adresową samej mikroinstrukcji w przypadku instrukcji skoku, zawartość rejestru głównego /skok z adresowaniem pośrednim/ lub zawartość odpowiedniego rejestru adresowego przy skoku buforowanym czy też przerwaniu /w rejestrze zachowany jest adres, do którego trzeba wrócić/.

Magistrala adresowa pamięci ROM zawiera 8 linii wychodzących z licznika adresów MAC.

Wyjście danych stanowi 16 linii tworzących magistralę wejściową rejestru instrukcji OREG.

Zespół bramek sterowanych sygnałem LMACE umożliwia automatyczne wpisywanie do OREG instrukcji NOOP podczas wykonywania niektórych instrukcji, niezależnie od stanu magistrali adresującej ROM.



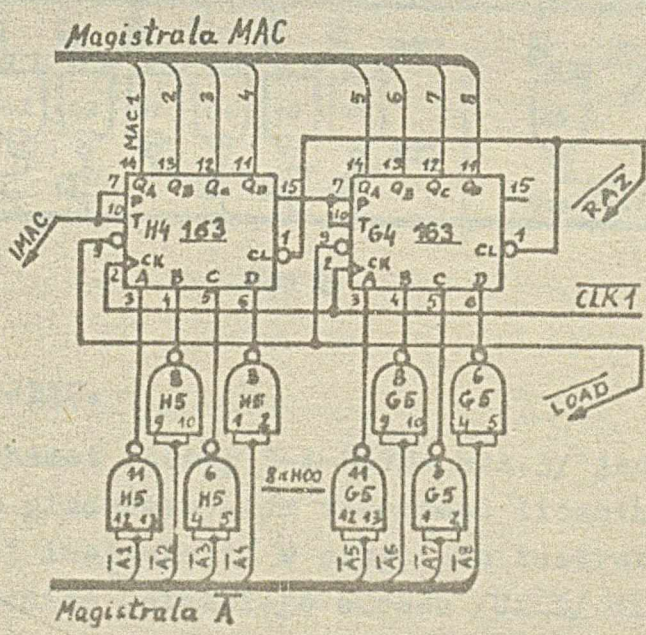
2.4.2. Licznik adresów MAC.

MAC /schemat ideowy 3-2-36784 ark.6/ jest ośmiobitowym licznikiem, którego stan poprzez magistralę MAC adresuje pamięć ROM.

Zrealizowany jest przy pomocy 2 obwodów SN 74163N. Zawartość rejestru jest inkrementowana z chwilą wpisywania do OREG wykonywanej instrukcji. Reprezentuje więc ona adres następnej instrukcji.

W przypadku instrukcji o skoku lub przerwania do licznika adresów wpisywany jest stan magistrali A stanowiący adres instrukcji, do której następuje skok. Wpis uaktywniany jest sygnałem sterującym LOAD.

W przypadku instrukcji wykonywanych w czasie dłuższym niż 1 okres impulsów zegarowych /0,2 μ s/ inkrementacja licznika jest wstrzymywana niskim poziomem sygnału IMAC.



Rys. 15

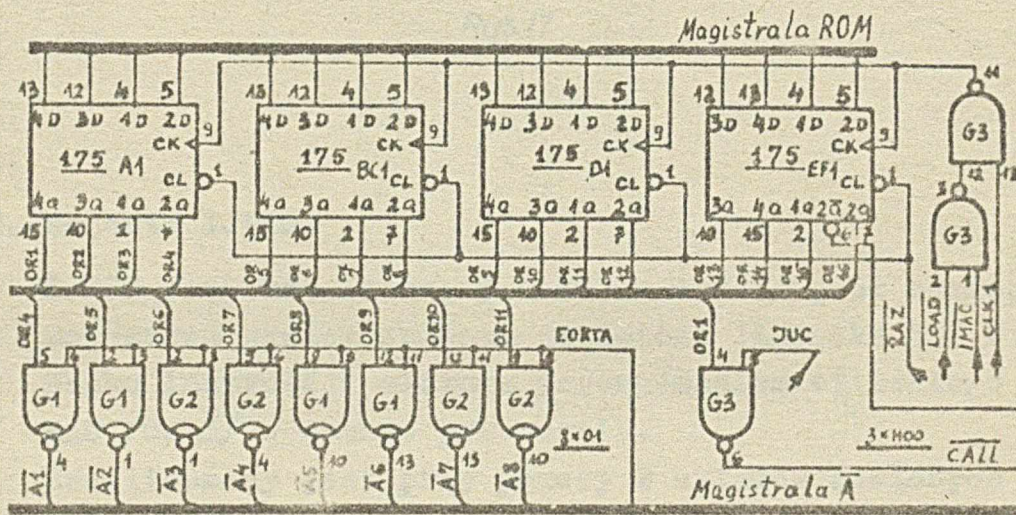
2.4.3. Rejestr instrukcji OREG.

OREG /schemat ideowy 3-2-36784 ark. 1,2/ jest szesnastobitowym rejestrem przechowującym wykonywaną instrukcję pobraną z pamięci ROM. Zrealizowany został przy użyciu czterech układów SN 74175N.

Wejście rejestru stanowi magistrala ROM, wyjście magistrala OREG.

Instrukcje wpisywane są do OREG kolejnymi impulsami zegarowymi, przy czym wpis może zostać wstrzymany odpowiednią kombinacją sygnałów \overline{IMAC} i \overline{LOAD} .

Bitów wyjściowych OREG /OR1 + OR16/ sterują pracą jednostki arytmetyczno-logicznej, a także adresują poszczególne rejestry.



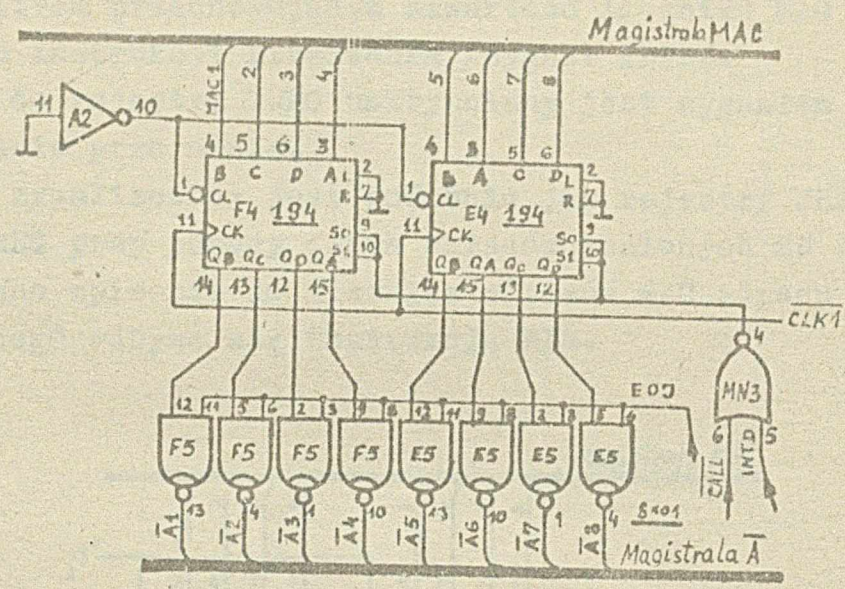
Rys. 16

2.4.4. Rejestr JREG.

JREG /schemat ideowy 3-2-36784 ark.3/ jest osmiobitowym rejestrem przechowującym zawartość licznika MAC /adres następnej instrukcji/ w przypadku instrukcji skoku z przechowaniem ostatniego adresu /CALL/ dla pracy w stanie normalnym.

Zbudowany jest z dwóch układów scalonych SN 74194H. Sterowaniem wyjścia zajmuje się instrukcja powrotu do stanu pierwotnego po skoku - REJ.

Instrukcja REJ przepisuje zawartość JREG do licznika MAC poprzez magistralę A.



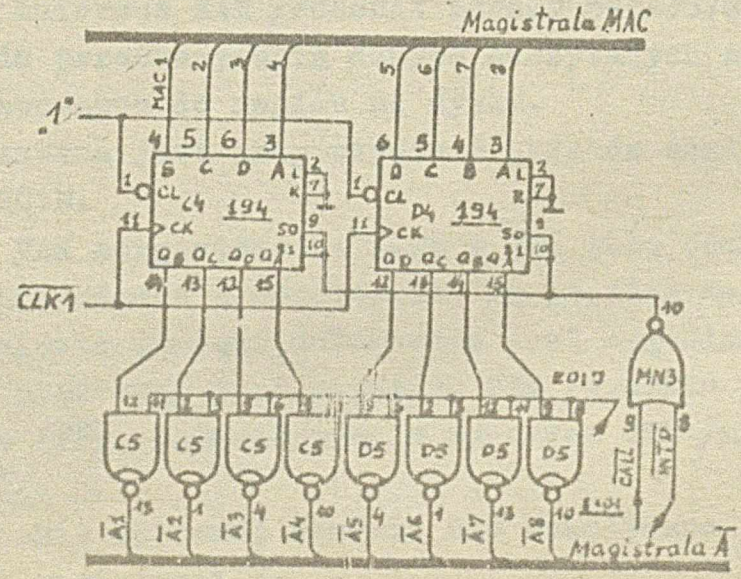
Rys. 17

2.4.5. Rejestr IJREG.

IJREG /schemat ideowy 3-L-36784 ark.5/ jest osmiobitowym rejestrem przechowujacym zawartosc licznika MAC w przypadku instrukcji skoku z przechowaniem ostatniego adresu MAC /CALL/ w stanie przerwania.

Zrealizowany jest przy pomocy 2 układow scalonych SN 74194N.

Ponowne wpisanie do licznika MAC adresu przechowywanego odbywa się instrukcją REJ podczas stanu przerwania.



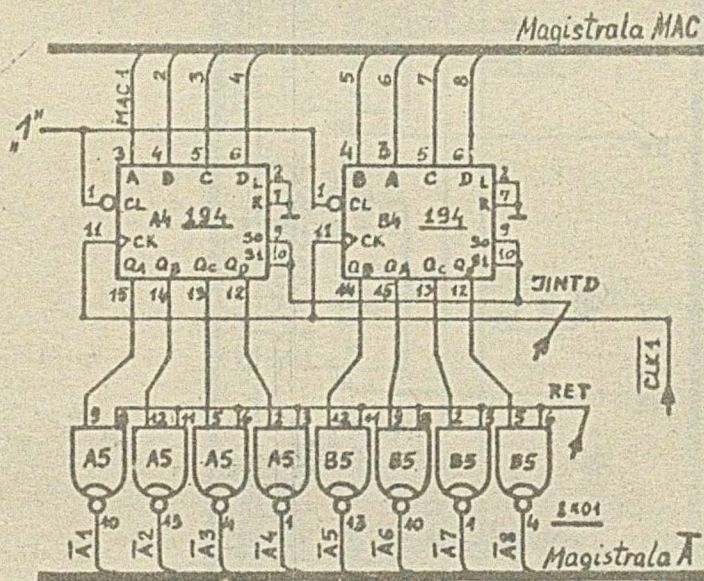
Rys. 18

2.4.6. Rejestr IREG.

IREG /schemat ideowy 3-1-36784 ark.4/ jest ośmiobitowym rejestrem przechowującym zawartość licznika MAC w przypadku instrukcji przerwania /sygnał INTD/.

Wpis do rejestru IREG uaktywniany jest sygnałem JINTD w stanie przerwania.

IREG zrealizowany jest podobnie jak rejestry JREG i IJREG przy pomocy dwóch układów scalonych SN 74194N. Ponowne wpisanie do licznika adresów MAC adresu przechowywanego odbywa się instrukcją RIN.



Rys.19

2.4.7. Pamięć buforowa RAM.

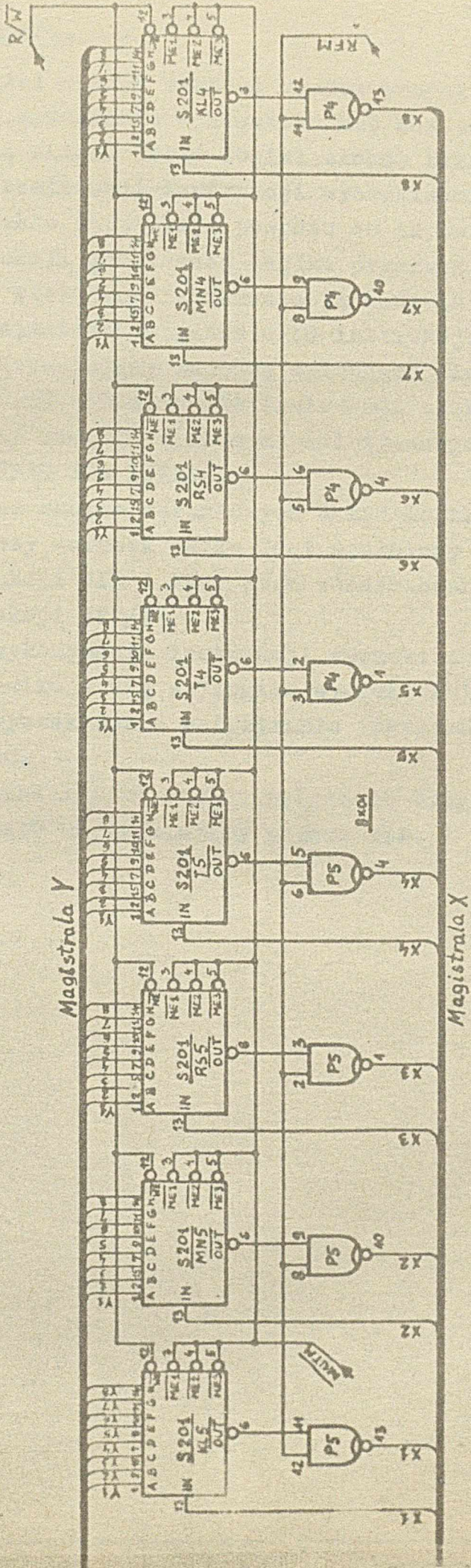
Pamięć buforowa RAM /schemat ideowy 3-1-36785 ark.7,8/ służy do przechowywania danych odczytanych z dysku lub przeznaczonych do zapisu na dysku.

Zrealizowana jest przy użyciu 8 układów scalonych SN 74S201N.

Pamięć RAM zorganizowana jest w 256 słów ośmiobitowych. Dane wprowadzane i odczytywane są poprzez magistralę X. Uaktywnienie pamięci dokonywane jest sygnałem WAITM. Odczyt sterowany jest sygnałem RFM /wysokim poziomem sygnału R/W/, natomiast wpis sygnałem WTM /niskim poziomem R/W/.

Adresowanie wybranej komórki RAM dokonywane jest ośmioma liniami magistrali Y.

27/63
Tom I Ce VI



Rys 20

2.4.8. Logika skoków.

Układ logiki skoków stanowi zespół bramek, którego zadaniem jest wymuszenie w liczniku MAC adresu określonego instrukcją skoku. Układ logiki skoków bierze również udział w realizacji instrukcji wyczekiwania oraz przerw, a także instrukcji powrotu do zapamiętanego adresu. Dla instrukcji przełączeń logika przerw generuje sygnały IMAC - sterujący inkrementacją MAC, LMACB - wymuszający na wejściach rejestru OREG instrukcję NOOP oraz LOAD - uaktywniający wejście wpisujące licznika MAC i rejestru OREG. Ponadto dla instrukcji skoku z przechowaniem /CALL/ steruje wpisem zapamiętywanego adresu do rejestru JREG lub IJREG.

W przypadku skoków warunkowych układ logiki skoków sprawdza czy warunek skoku jest spełniony /stan flag dla instrukcji JIR i JIS, stan zaadresowanego rejestru dla instrukcji JRO/.

W czasie wykonywania instrukcji wyczekiwania WPI, WPJ, układ komunikuje się z "logiką wyczekiwania" powodując na okres wyczekiwania wstrzymanie inkrementacji licznika adresów MAC.

Ponadto układ logiki skoków wysyła do "logiki przerw" sygnał VALINT umożliwiający przerwanie.

2.5. Instrukcje mikroprogramu.

Formater operuje instrukcjami 16-bitowymi umieszczonymi przed wykonaniem w rejestrze instrukcji OREG. Bity 11 + 16 stanowią część operacyjną kodu instrukcji, natomiast pozostałe w zależności od rodzaju instrukcji mogą stanowić część adresową.

Kod adresowy rejestrów

Rejestr wejściowy IPREG

	8	7	6	5	4	3	2	1
IPRO	1	0	0	0	X	X	X	X
IPR1	1	0	0	1	X	X	X	X
IPR2	1	0	1	0	X	X	X	X
IPR3	1	0	1	1	X	X	X	X
IPR4	1	1	0	0	X	X	X	X
IPR5	1	1	0	1	X	X	X	X
IPR6	1	1	1	0	X	X	X	X
IPR7	1	1	1	1	X	X	X	X

Rejestr wyjściowy OPREG

	8	7	6	5	4	3	2	1
OPRO	X	X	X	X	1	0	0	0
OPR1	X	X	X	X	1	0	0	1
OPR2	X	X	X	X	1	0	1	0
OPR3	X	X	X	X	1	0	1	1
OPR4	X	X	X	X	1	1	0	0
OPR5	X	X	X	X	1	1	0	1
OPR6	X	X	X	X	1	1	1	0
OPR7	X	X	X	X	1	1	1	1

Lista adresów "Flag"

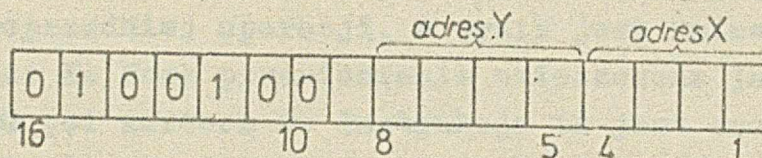
przeniesienie 00
 równość 01
 użytkownicy 02
 03
 04
 05
 06
 07

Lista kodowa instrukcji
OREG

	16	15	14	13	12	11	1
JIS	1	1	1	1	0	X	X
JIR	1	1	1	0	1	X	X
JUC	1	1	0	0	1	X	0
CALL	1	1	0	0	1	X	1
JRO	1	1	0	0	0	X	X
JEX	1	1	1	0	0	X	X
REJ	1	1	0	1	0	X	X
RIN	1	1	0	1	1	X	X
SRS	1	0	1	0	0	1	X
RES	1	0	1	0	0	0	X
MIN	1	0	1	1	0	1	X
WTM	1	0	0	1	0	1	X
RFM	1	0	0	1	0	0	X
WPJ	1	1	1	1	1	X	X
WPI	1	0	1	1	0	0	X
NOOP	0	0	0	0	0	0	X
STOP	0	0	0	1	0	0	X
LDC	1	0	0	0			
ADD	0	1	0	0	1	0	X
SUB	0	0	1	1	0	0	X
INC	0	1	0	0	0	0	X
DEC	0	1	1	1	1	0	X
SHF	0	1	1	0	0	0	X
GOP	0	1	0	1	0	1	X
CMP	0	0	1	0	1	1	X
ORL	0	1	1	1	0	1	X
ANL	0	1	0	1	1	1	X
EOL	0	0	1	1	0	1	X
COM	0	1	1	1	0	0	X
ADC	0	1	0	0	1	0	X
SBB	0	0	1	1	0	0	X
RAL	0	1	1	0	0	0	X
ADF	0	0	0	0	1	0	X

2.5.1. Instrukcja ADD.

$$ADD = X + Y \rightarrow X$$



Instrukcja ADD pozwala dodać zawartość rejestru X do zawartości rejestru Y bez uwzględnienia przeniesienia z poprzedniej operacji, a wynik umieszczany jest w rejestrze X. Przeniesienie umieszczane jest w rejestrze "flag" o adresie 0.

Bity 5 + 8 określają źródło Y /rejestr główny lub wejściowy zależnie od bitu 8/

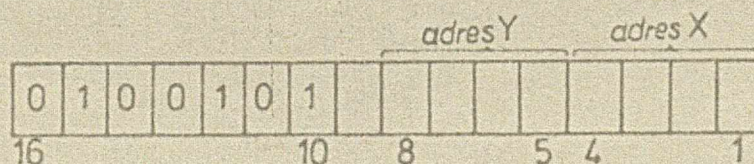
Bity 1 + 4 określają źródło X /zawsze rejestr główny niezależnie od bitu 4/ i adres dla rezultatu działania /rejestr główny lub wyjściowy zależnie od bitu 4/.

Instrukcja jest wykonywana w czasie jednego okresu impulsów zegarowych. W tym czasie instrukcja zostaje zdekodowana, przy czym z części operacyjnej uzyskuje się odpowiednią kombinację sygnałów sterujących pracą jednostki arytmetyczno-logicznej ALU.

Magistralą Y podawana jest do ALU zawartość zaadresowanego bitami 5 + 7 rejestru głównego /bit 8 równy 0/ lub wejściowego /bit 8 równy 1/. Magistralą X natomiast zawartość zaadresowanego bitami 1 + 3 rejestru głównego. W ALU zostaje wykonana operacja dodawania, a wynik umieszczony może być w zaadresowanym bitami 1 + 3 rejestrze głównym /bit 4 równy 0/ lub wyjściowym /bit 4 równy 1/.

2.5.2. Instrukcja ADC.

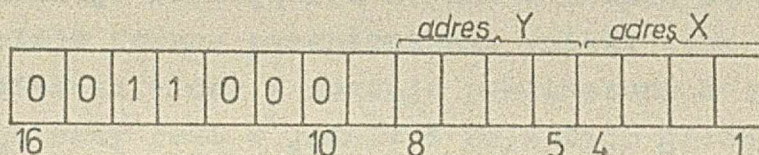
$$ADC = X + Y + C_{n-1} \rightarrow X$$



Instrukcja ADC pozwala dodać zawartość rejestru X do zawartości rejestru Y z uwzględnieniem przeniesienia z poprzedniej operacji, a wynik jest umieszczany w rejestrze X. Nowe przeniesienie umieszczone jest w rejestrze FLAG pod adresem 0. Instrukcja ta jest realizowana analogicznie jak ADD.

2.5.3. Instrukcja SUB.

$SUB = X - Y \rightarrow X$

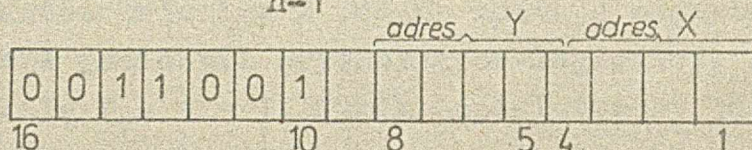


Instrukcja SUB powoduje odjęcie zawartości rejestru Y od zawartości rejestru X bez uwzględnienia przeniesienia /pożyczki/ z poprzedniej operacji i umieszczenie wyniku w rejestrze X. Przeniesienie w wyniku tej operacji jest wpisywane do rejestru FLAG pod adresem 0.

Realizacja instrukcji odbywa się podobnie jak w przypadku instrukcji ADD.

2.5.4. Instrukcja SBB.

$SBB = X - Y - C_{n-1} \rightarrow X$

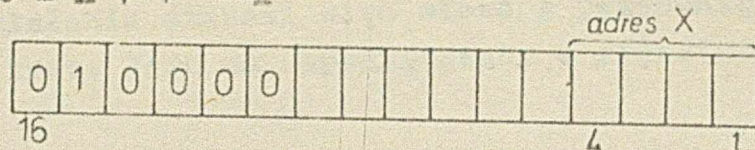


Instrukcja SBB powoduje odjęcie zawartości rejestru Y od zawartości rejestru X z uwzględnieniem przeniesienia ujemnego /pożyczki/ z poprzedniej operacji i umieszcza wynik w rejestrze X. Nowe przeniesienie wpisywane jest jako flaga 0.

Realizacja tej instrukcji - podobnie jak ADD.

2.5.5. Instrukcja INC.

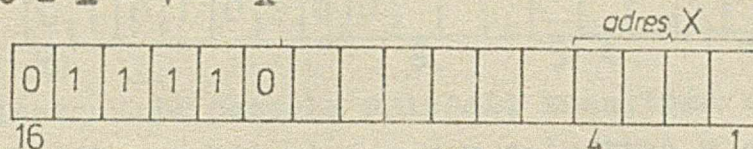
$$\text{INC} = X + 1 \rightarrow X$$



Instrukcja pozwala powiększyć o 1 zawartość rejestru X o adresie określonym bitami 1 + 4 kodu instrukcji. Zawartość rejestru X podana zostaje na wejście A jednostki arytmetyczno-logicznej ALU, która dla tej instrukcji realizuje funkcję $F = A$. Równocześnie na wejściu Cn ALU zostaje podane przeniesienie /niski stan wejścia Cn/. W efekcie słowo A zostaje powiększone o 1. Wynik zapamiętywany jest w jednym z rejestrów głównych, a ewentualne przepełnienie we fładze 0. Instrukcja INC wykonywana jest w czasie jednego okresu impulsów zegarowych.

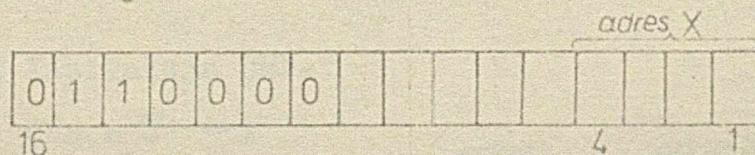
2.5.6. Instrukcja DEC.

$$\text{DEC} = X - 1 \rightarrow X$$



Instrukcja powoduje zmniejszenie o 1 zawartości rejestru X określonego bitami 1 + 4. Zawartość rejestru X podana zostaje magistralą X na wejście A jednostki arytmetyczno-logicznej. ALU realizuje funkcję $F = A - 1$. Wynik zapamiętywany jest w rejestrze X.

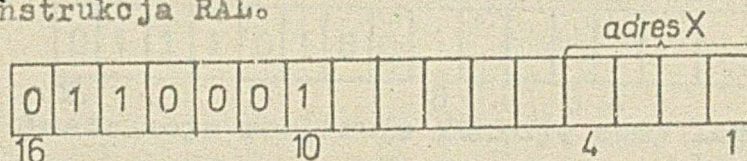
2.5.7. Instrukcja SHF.



Instrukcja powoduje przesunięcie zawartości rejestru X określonego bitami 1 + 4 o jedną pozycję w lewo, w kierunku bardziej znaczących pozycji. Najbardziej znaczący bit zapamiętany zostaje w przerzutniku /flaga 0/. Na najmniej znaczącą pozycję wpisywane jest 0.

Realizacja instrukcji jest następująca: zawartość rejestru X podana zostaje na wejście jednostki arytmetyczno-logicznej ALU. ALU realizuje funkcję $F = A + A$. Wynik działania stanowi więc słowo A przesunięte o 1 bit w lewo. Jest on zapamiętywany w rejestrze X.

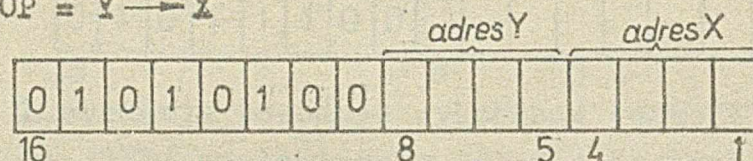
2.5.8. Instrukcja RAL.



Instrukcja ta powoduje rotację zawartości rejestru X poprzez flagę przeniesienia o jedną pozycję w lewo. Działa ona tak jak instrukcja SHF z tym, że na najmniej znaczącą pozycję wpisywana jest zawartość flagi przeniesienia /FLAG O/ z poprzedniej operacji.

2.5.9. Instrukcja COP.

$COP = Y \rightarrow X$

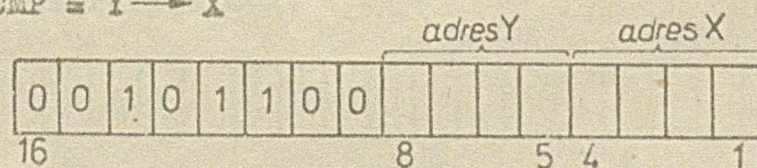


Instrukcja pozwala umieścić zawartość rejestru o adresie określonym bitami 5 + 8 w rejestrze o adresie określonym bitami 1 + 4.

Realizacja instrukcji odbywa się następująco: zawartość zaadresowanego rejestru podana zostaje na wejście B ALU, które dla tej instrukcji realizuje funkcję $F = B$. Wynik umieszczony zostaje w jednym z rejestrów głównych lub wyjściowych określonych bitami 1 + 4 kodu instrukcji.

2.5.10. Instrukcja CMP.

$CMP = \bar{Y} \rightarrow X$

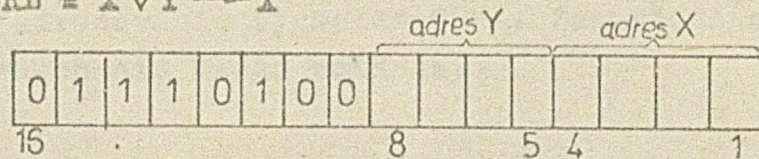


Instrukcja pozwala umieścić dopełnienie zawartości rejestru określonego bitami 5 + 8 w rejestrze o adresie określonym bitami 1 + 4.

Realizacja jest taka sama jak dla instrukcji OMP, przy czym ALU realizuje funkcję $F = \bar{B}$.

2.5.11. Instrukcja ORL.

$$\text{ORL} = X \vee Y \rightarrow X$$

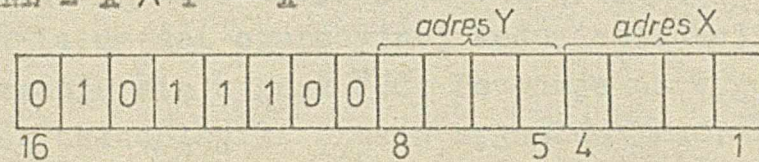


Instrukcja realizuje funkcję sumy logicznej /OR/ na odpowiadających sobie bitach zawartości rejestrów X /adres określony bitami 5 + 8/. Wynik zapamiętywany jest w rejestrze X.

Funkcja sumy logicznej realizowana jest w ALU.

2.5.12. Instrukcja ANL.

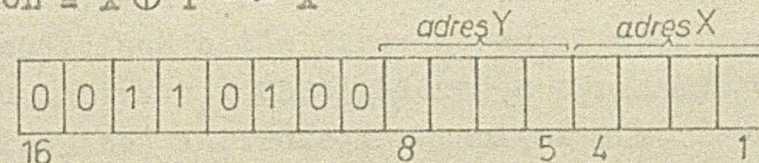
$$\text{ANL} = X \wedge Y \rightarrow X$$



Instrukcja powoduje wykonanie funkcji iloczynu logicznego /AND/ analogicznie jak dla instrukcji ORL.

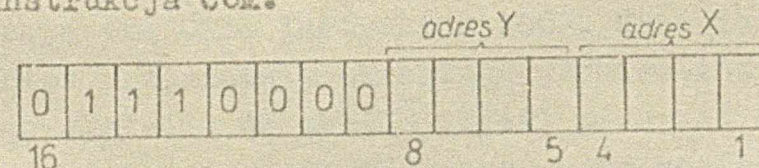
2.5.13. Instrukcja EOL.

$$\text{EOL} = X \oplus Y \rightarrow X$$



Instrukcja powoduje wykonanie funkcji sumy modulo dwa /exclusive OR/ analogicznie jak dla instrukcji ORL.

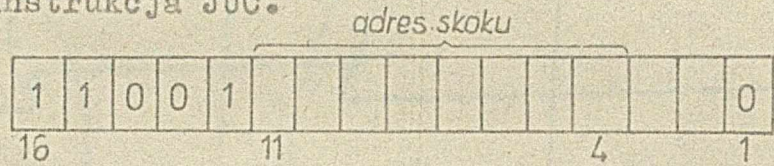
2.5.14. Instrukcja COM.



Instrukcja umożliwia porównanie zawartości rejestru X określonego bitami 1 + 4 z zawartością rejestru Y określonego bitami 5 + 8. Wynik porównania przechowywany jest w rejestrze flag 0 adrese 1 /"1" wpisane do przetwornika oznacza równość/.

Realizacja instrukcji jest podobna jak w przypadku instrukcji SUB /taka sama funkcja ALU/, przy czym wykorzystywane jest wyjście A = B jednostki arytmetyczno-logicznej. Stan tego wyjścia podawany jest na wejście rejestru flag /sygnał DSTAT/ i wpisywany do przerzutnika 01 /adresowanie sygnałem COM/. Zawartość rejestrów głównych nie ulega zmianie.

2.5.15. Instrukcja JUC.



Instrukcja powoduje skok bezwarunkowy do instrukcji programu o adresie określonym bitami 4 + 11. Bit 1 jest zawsze równy 0.

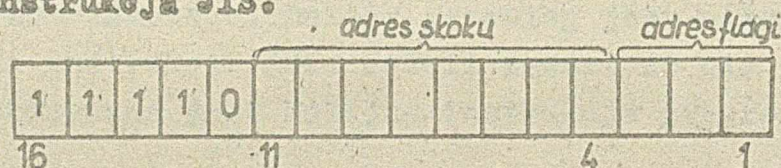
Realizacja instrukcji jest następująca: w wyniku dekodowania części operacyjnej kodu instrukcji /bity 12+16/ otrzymuje się sygnał \overline{JUC} . Powoduje on wygenerowanie przez układ logiki skoków sygnałów IMAC, \overline{LOAD} , IMACB, EORTA. Sygnał EORTA powoduje podanie na magistralę A bitów 4 + 11, które stanowią żądany adres następnej instrukcji programu.

Sygnał IMAC powoduje wstrzymanie inkrementacji licznika adresów MAC, a sygnał \overline{LOAD} uaktywnia jego wejście wpisyjące. Sygnał IMACB powoduje ustawienie stanów niskich na wejściach bitów 12 + 16 /część operacyjna/ do rejestru instrukcji OREG, blokując tym samym wpis części operacyjnej instrukcji z pamięci ROM.

Równocześnie na wejściu J przerzutnika definiującego koniec wykonywania instrukcji podany zostaje stan wysoki. Z chwilą pojawienia się impulsu zegarowego do licznika adresów MAC wpisany zostaje stan magistrali A, natomiast do rejestru OREG instrukcja NOOP. Ustawiony zostaje również przerzutnik określający koniec instrukcji, powodując wysłanie do układu logiki przerwań sygnału VALINT. W wyniku dekodowania NOOP skasowane zostają sygnały IMAC, \overline{LOAD} , IMACB, EORTA. Następnym impuls zegarowy wpisuje do OREG instrukcję, do której nastąpił skok i wznowia inkrementację MAC. Powoduje również wyzerowanie przerzutnika końca instrukcji i zależnie od innych warunków skasowanie VALINT.

IJREG w przypadku przerwania. Impuls zegarowy wpisujący nowy adres do MAC powoduje również wpisanie do jednego z tych rejestrów starej zawartości licznika adresów określającej adres następnej z kolei instrukcji. Dalsza część realizacji instrukcji jest taka sama jak dla JUC.

2.5.17. Instrukcja JIS.



Instrukcja powoduje skok do instrukcji o adresie określonym bitami 4 + 11, gdy przerzutnik flagi o adresie określonym bitami 1 + 3 jest ustawiony na "1".

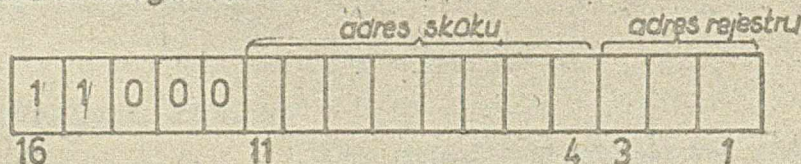
Instrukcja JIS realizowana jest podobnie jak JUC. W wyniku dekodowania otrzymywany jest sygnał JIS. Równocześnie sprawdzany jest stan przerzutnika flagi /sygnał \overline{CF} /. Jeżeli jest on ustawiony, dalszy przebieg jest taki sam jak dla instrukcji JUC, jeżeli nie, wykonywana jest kolejna instrukcja programu.

2.5.18. Instrukcja JIR.



Instrukcja powoduje skok do instrukcji o adresie określonym bitami 4 + 11, gdy przerzutnik flagi o adresie określonym bitami 1 + 3 kodu instrukcji jest wyzerowany. Realizacja instrukcji jest taka sama jak dla instrukcji JIS.

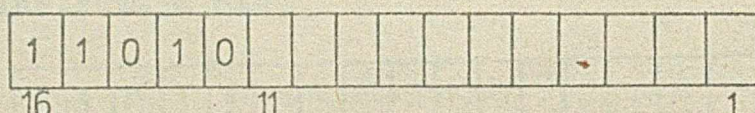
2.5.19. Instrukcja JRO.



Instrukcja powoduje skok do instrukcji o adresie określonym bitami 4 + 11 w przypadku gdy zawartość rejestru określonego bitami 1 + 3 wynosi zero.

Realizacja instrukcji przebiega podobnie jak w przypadku poprzednio omówionych instrukcji skoków. W wyniku dekodowania otrzymywany jest sygnał JRO, który uaktywnia układ wykrywający zerowy stan magistrali X. Równocześnie na magistralę X podawana jest zawartość rejestru głównego zaadresowanego bitami 1 + 3. Jeżeli zawartość ta jest równa zero otrzymywany jest sygnał, który spełnia identyczną rolę jak sygnał JUC w przypadku instrukcji skoku bezwarunkowego. Dalszy przebieg jest więc taki sam jak dla instrukcji JUC. Jeżeli zawartość nie jest równa zero wykonywana jest kolejna instrukcja programu.

2.5.20. Instrukcja REJ.

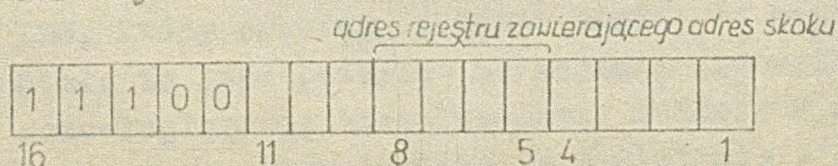


Instrukcja ta powoduje powrót do instrukcji o adresie zapamiętanym w rejestrze JREG /w stanie pracy normalnej/ lub IJREG /w czasie przerwania/ po zrealizowaniu skoku. W wyniku dekodowania instrukcji otrzymywany jest sygnał REJ, który powoduje wygenerowanie przez układ logiki skoków sygnałów LMAC, LOAD, LMACE. Działanie tych sygnałów jest takie samo jak dla instrukcji JUC. Ponadto sygnał REJ powoduje pojawienie się jednego z sygnałów EOJ lub EOIJ. EOJ jest otrzymywany wtedy gdy sygnał INTD definiujący stan przerwania jest niski /nie ma przerwania/, EOIJ gdy INTD jest wysoki /trwa przerwanie/. EOJ powoduje podanie na magistralę A zawartości rejestru JREG, EOIJ natomiast zawartość IJREG.

Rejestry JREG i IJREG /pierwszy dla pracy normalnej, drugi dla stanu przerwania/ przechowują adres instrukcji do której należy wrócić po wykonaniu programu.

Dalsza realizacja instrukcji jest taka sama jak dla instrukcji JUC.

2.5.21. Instrukcja JEX.

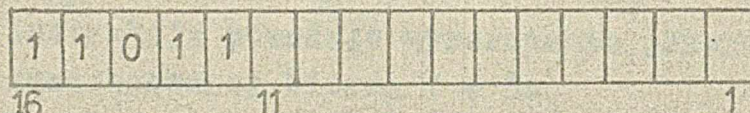


Instrukcja ta umożliwia adresowanie pośrednie. Pozwala mianowicie na skok do instrukcji programu, której adres znajduje się w rejestrze określonym bitami 5 + 8 instrukcji JEX.

Realizacja instrukcji polega na przeładowaniu zawartości zaadresowanego rejestru do licznika adresów MAC. Sygnał JEX powoduje podanie na magistralę A stanu magistrali Y. Magistralą Y zaś przesyłana jest zawartość zaadresowanego rejestru.

Dalszy przebieg instrukcji jest taki sam jak dla instrukcji JUC.

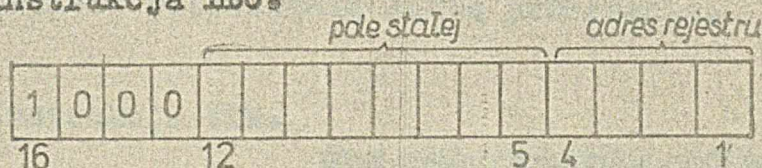
2.5.22. Instrukcja RIN.



Instrukcja umożliwia powrót do programu głównego po zakończeniu przerwania. Powoduje ona przeładowanie zawartości rejestru IREG, stanowiącej adres instrukcji, która miała być wykonywana w momencie pojawienia się przerwania, do licznika adresów MAC.

Realizacja instrukcji jest taka sama jak dla instrukcji REJ.

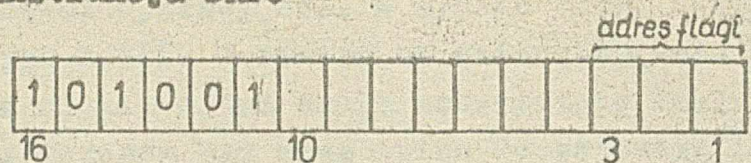
2.5.23. Instrukcja LDC.



Instrukcja powoduje zapis w rejestrze /głównym lub wyjściowym/ zaadresowanym bitami 1 + 4 stałej określonej bitami 5 + 12. Stała jest liczbą naturalną i jej wartość nie może przekraczać 256.

Instrukcja LDC jest wykonywana w czasie jednego taktu zegara. W wyniku dekodowania otrzymywany jest sygnał LDC, który powoduje podanie stałej na magistralę X. Stała ta jest podawana na wejście jednostki arytmetyczno-logicznej, która wykonuje funkcję $F = A$. Tak więc zostaje ona przeniesiona na magistralę S i z chwilą pojawienia się impulsu zegarowego zapamiętana w zaadresowanym rejestrze.

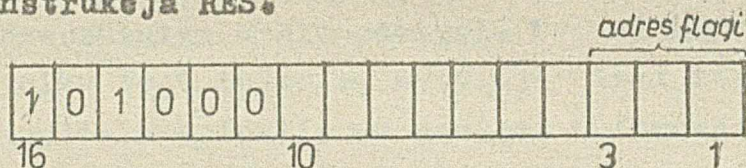
2.5.24. Instrukcja SRS.



Instrukcja powoduje ustawienie przerzutnika flagi zaadresowanej bitami 1 + 3.

Sygnal SRS otrzymany w wyniku dekodowania instrukcji powoduje ustawienie wysokiego stanu na wejście danych rejestru flag oraz uaktywnia wejście wpisujące sekcji 1 lub 2 w zależności od sygnału INTD.

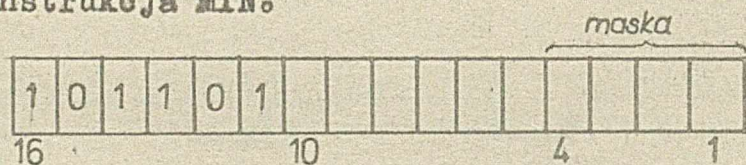
2.5.25. Instrukcja RES.



Instrukcja powoduje wyzerowanie przerzutnika flagi zaadresowanej bitami 1 + 3.

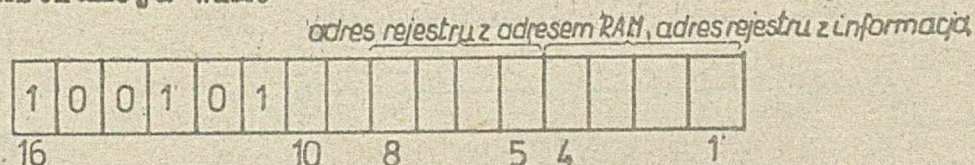
Realizacja instrukcji jest taka sama jak dla SRS.

2.5.26. Instrukcja MIN.



Instrukcja powoduje wpisanie maski do rejestru maskującego wejścia rejestru przerwania. Maskę stanowią bity 1 + 4, przy czym ustawienie na "1" jednego z tych bitów otwiera odpowiednią linię przerwania.

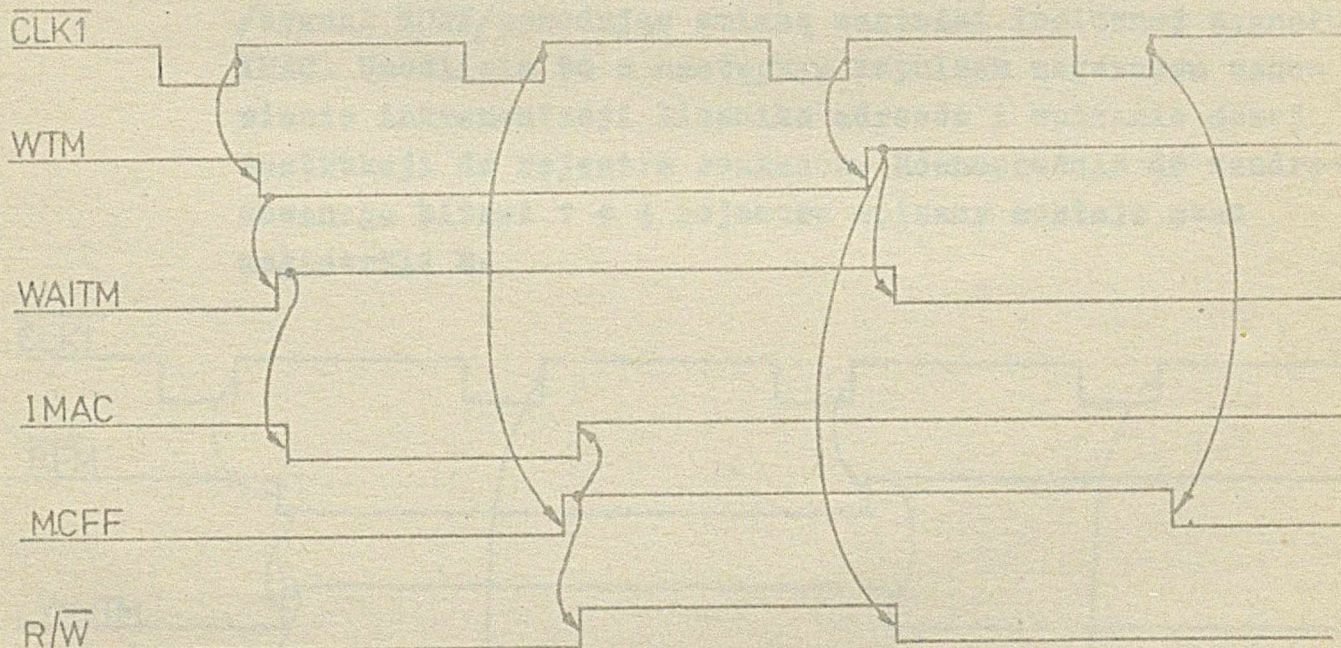
2.5.27. Instrukcja WTM.



Instrukcja umożliwia wpisanie do komórki pamięci buforowej RAM o adresie zawartym w rejestrze określonym bitami 5 + 8 informacji umieszczonej w rejestrze określonym bitami 1 + 4.

Realizacja instrukcji rozpoczyna się od dekodowania części operacyjnej, w wyniku czego otrzymuje się sygnał \overline{WTM} . Powoduje on zmianę sygnału \overline{IMAC} z wysokiego na niski, a tym samym wstrzymanie inkrementacji licznika adresów \overline{MAC} oraz wpisu do rejestru instrukcji \overline{OREG} . Równocześnie pojawia się sygnał \overline{WAITM} , który uaktywnia pamięć buforową oraz \overline{WAITM} wymuszający wysoki stan na wejściu J przerzutnika cyklu pamięci. Następuje również podanie na magistralę Y zawartości rejestru adresowanego bitami 5 + 8 oraz na magistralę X zawartości rejestru adresowanego bitami 1 + 3.

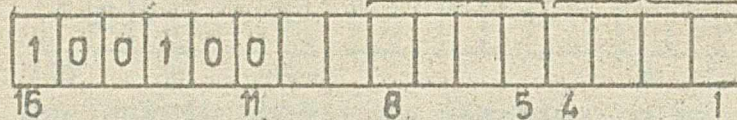
Najbliższy impuls zegarowy nie powoduje zmiany zawartości rejestrów \overline{MAC} i \overline{OREG} , natomiast powoduje ustawienie przerzutnika cyklu pamięci. Sygnał \overline{MCFF} , który się w wyniku tego pojawia, wywołuje niski stan $\overline{R/W}$, a więc umożliwia wpisanie do pamięci buforowej danych dostępnych na magistrali X . Równocześnie pojawia się ponownie sygnał \overline{IMAC} powodując z nadejściem kolejnego impulsu zegarowego wpisanie nowej instrukcji do \overline{OREG} jak i inkrementację \overline{MAC} .



Rys. 22

2.5.28. Instrukcja RFM.

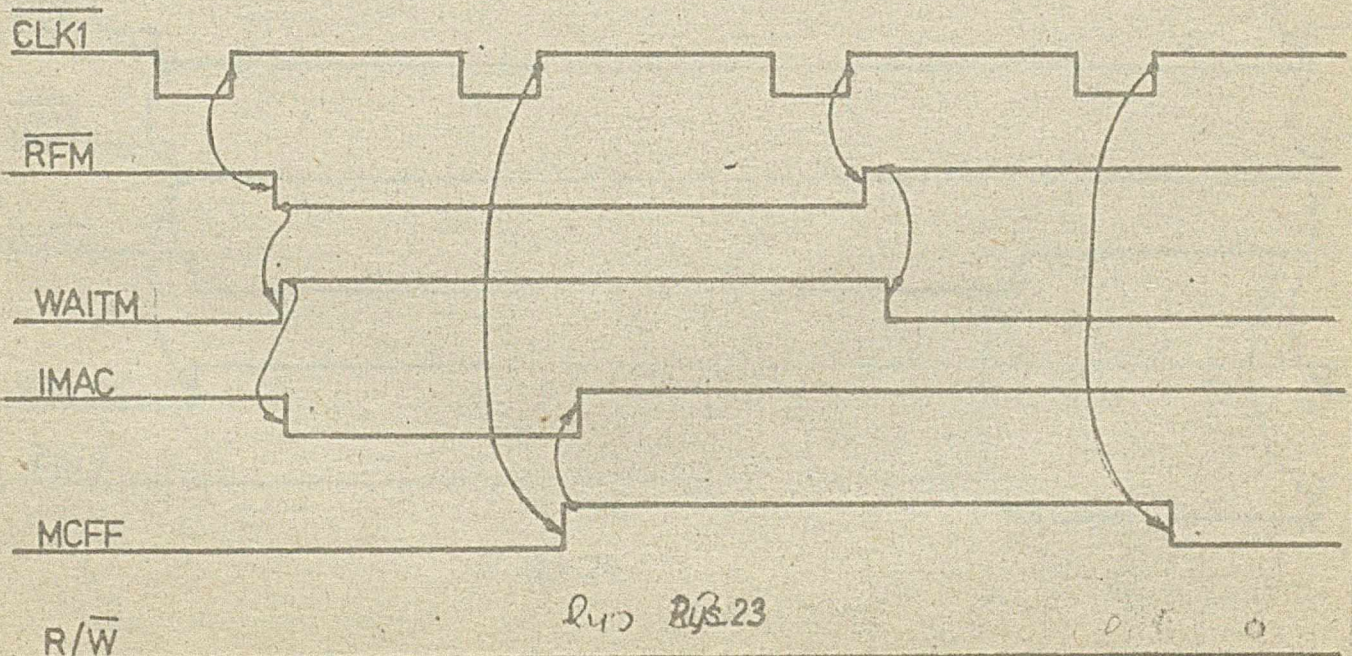
adres rejestru z adresem RAM, adres rejestru dla informacji



Instrukcja umożliwia wpisanie danych zawartych w komórce pamięci buforowej, której adres zawarty jest w rejestrze określonym bitami 5 → 8, do rejestru zaadresowanego bitami 1 → 4.

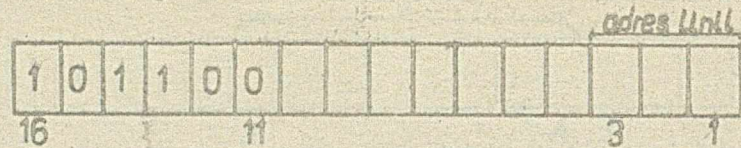
W wyniku dekodowania instrukcji otrzymuje się sygnał RFM, który powoduje pojawienie się WAITM uaktywniający pamięć buforową. Ponieważ sygnał R/W jest w czasie całego cyklu zapisu wysoki oraz otwarte zostają sygnałem RFM branki wyjściowe z pamięci, na magistralę X podana zostaje zawartość zaadresowanej komórki bufora.

Dane podawane są na magistralę X na wejście A jednostki arytmetyczno-logicznej ALU, która wykonuje funkcję $F = A$ przenosząc zawartość magistrali X na magistralę S. WAITM wywołuje również niski poziom sygnału IMAC, tak że z chwilą nadejścia impulsu zegarowego nie następuje inkrementacja MAC ani wpisanie nowej instrukcji do OREG. Ustawiony zostaje natomiast przerzutnik cyklu pamięci /sygnał MOFF/ powodując zmianę wartości logicznej sygnału IMAC. Umożliwia to z następnym impulsem zegarowym wznowienie inkrementacji licznika adresów i wpisanie nowej instrukcji do rejestru rozkazów. Równocześnie do zaadresowanego bitami 1 → 4 rejestru wpisany zostaje stan magistrali S.



Rys. 23

2.3.29. Instrukcja WPI.

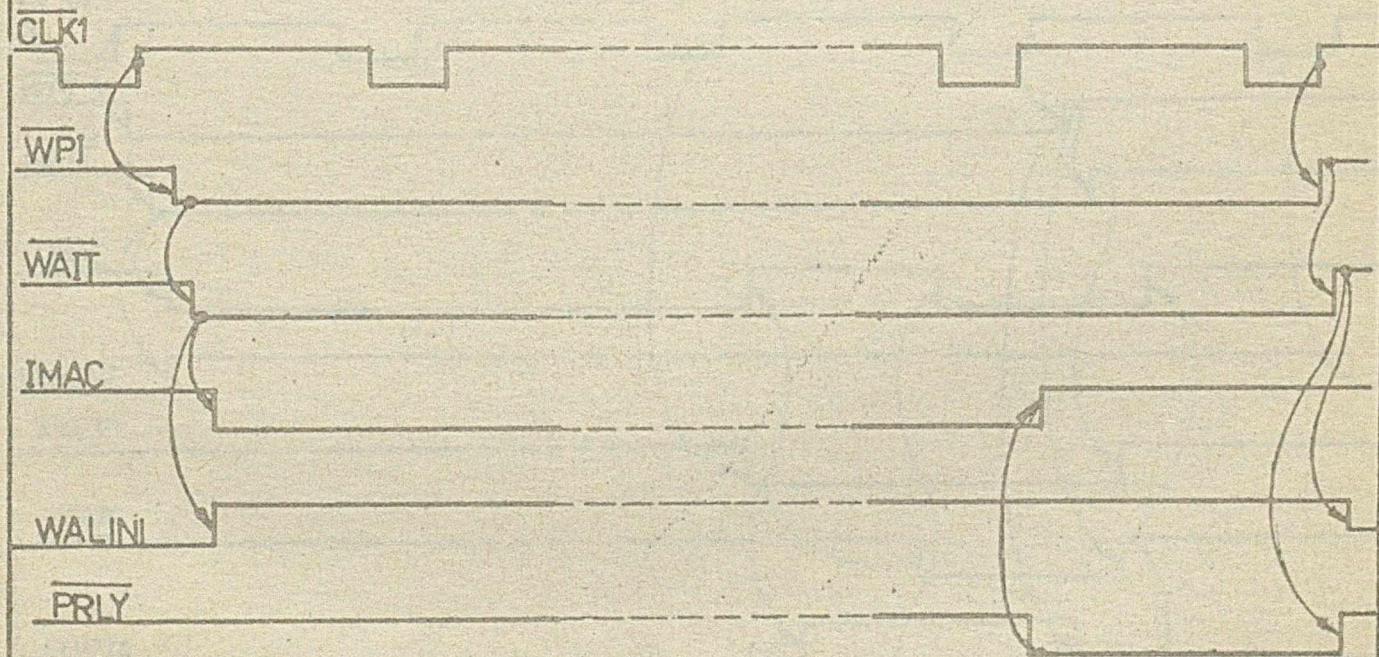


Instrukcja umożliwia oczekiwanie na sygnał zewnętrzny na linii określonej bitami 1 i 3. Pojawienie się sygnału powoduje przejście do następnej instrukcji.

Sygnał WPI pojawiający się w wyniku dekodowania instrukcji powoduje wygenerowanie sygnału WAIT oraz niski poziom IMAC. W związku z tym następuje wstrzymanie inkrementacji licznika adresów MAC, wstrzymanie wpisu do rejestru instrukcji OREG oraz uaktywnienie sygnału WAIT multiplexera WAIT.

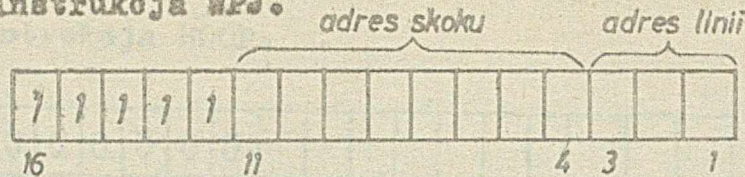
Wysyłany zostaje również do układu logiki przerwań sygnał VALINT.

Pojawiający się na wybranej linii oczekiwany sygnał zostaje zsynchronizowany w rejestrze REGW, a następnie podany na wejście multiplexera WAIT. WAIT wysyła wówczas sygnał PRLY powodujący zmianę wartości logicznej sygnału IMAC, co umożliwia z chwilą pojawienia się impulsu zegarowego wznowienie inkrementacji MAC i wpisanie nowej instrukcji do OREG.



Rys. 243

2.5.30. Instrukcja WPJ.



Instrukcja pozwala oczekiwać na sygnał z linii wejściowej określonej bitami 1 + 3 przed skokiem do instrukcji o adresie podanym bitami 4 + 11.

Realizacja instrukcji do momentu pojawienia się sygnału PRLY jest identyczna jak w przypadku instrukcji WPI.

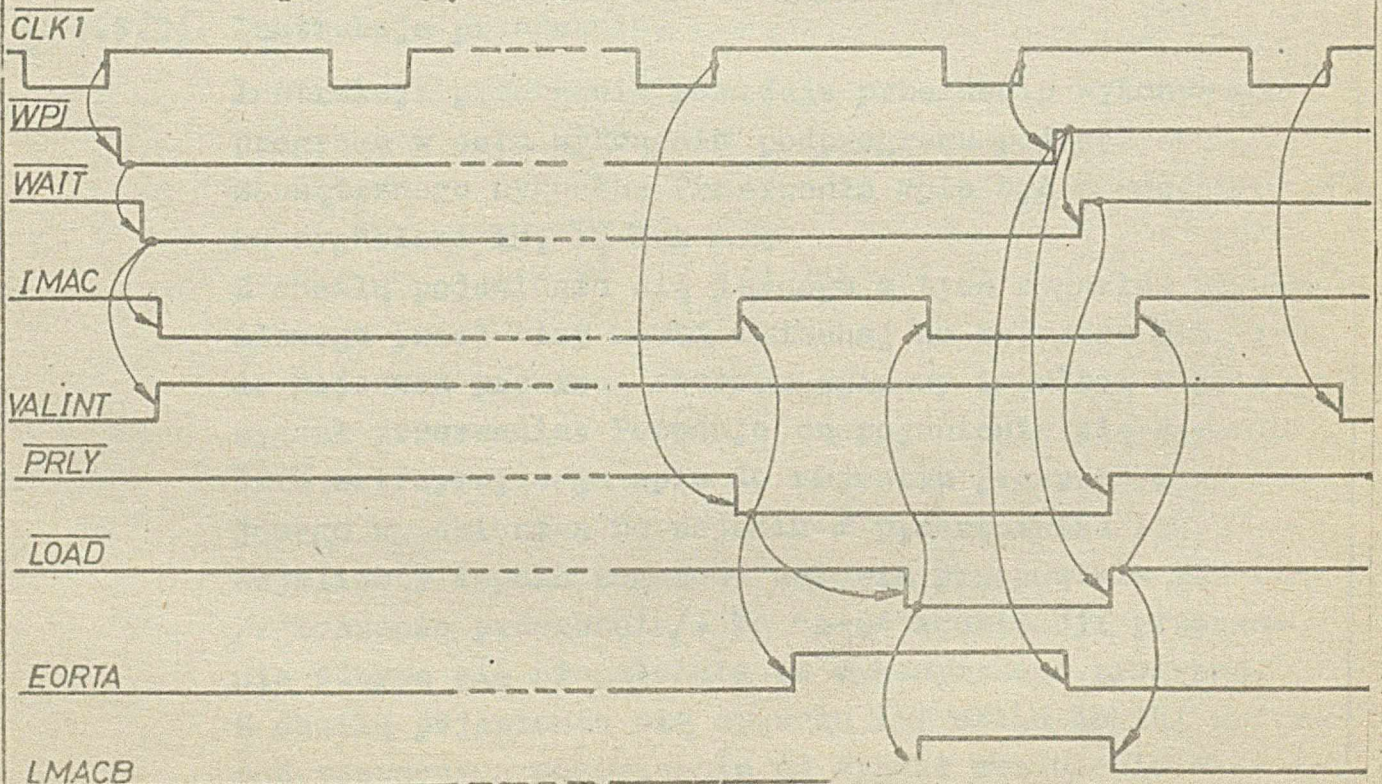
Sygnał PRLY nie powoduje jednak wznowienia inkrementacji MAC. Pojawiają się natomiast sygnały LOAD uaktywniający wejście wpisujące MAC, EORTA powodujący podanie na magistralę A bitów 5 + 11 oraz LMACB wprowadzający na wejście OREG instrukcję NOOP.

Najbliższy impuls zegarowy powoduje wpisanie do MAC stanu magistrali A, który stanowi adres następnej instrukcji programu, natomiast do OREG instrukcji NOOP.

Ustawiony zostaje również przerzutnik końca instrukcji wysyłając do układu logiki przerwań sygnał VALINT.

W wyniku dekodowania NOOP skasowane zostają sygnały LOAD, EORTA i LMACB. Pojawia się również IMAC powodujący z następnym impulsem zegarowym wznowienie inkrementacji MAC oraz wpisanie nowej instrukcji do OREG.

Wyzerowany zostaje również przerzutnik końca instrukcji przestając wysyłać sygnał VALINT.



Rys. 25

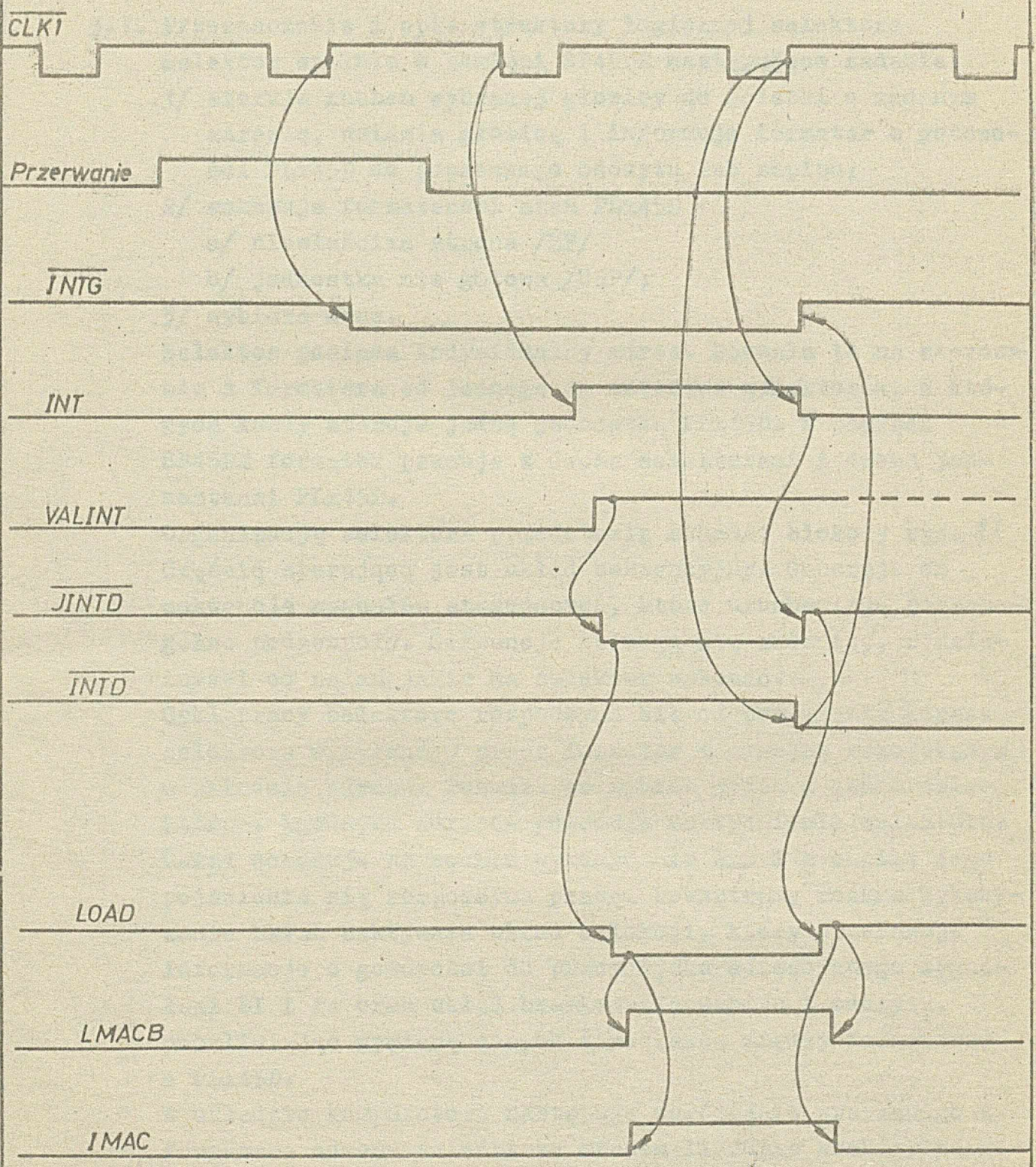
VALINT, w zależności od rodzaju instrukcji może pojawiać się w różnych fazach jej wykonywania.

Dla instrukcji wykonywanych w czasie jednego taktu zegara /z wyjątkiem instrukcji MIN/ oraz instrukcji wyczekiwania VALINT pojawia się z chwilą ich zdekodowania.

Dla instrukcji przełączeń, VALINT generowany jest równocześnie z wpisem NOOP do OREG i nowego adresu do licznika MAC.

VALINT w iloczynnie z INT daje sygnał JINTD, który powoduje wygenerowanie przez układ logiki skoków sygnałów LOAD, IMACB, IMAC, analogicznie jak w przypadku instrukcji skoków. Równocześnie na magistralę A podany zostaje zakodowany przez koder COD, adres pierwszej instrukcji podprogramu /4 bity/. Adres ten z chwilą nadejścia impulsu zegarowego zostaje wpisany do MAC. Równocześnie do rejestru IREG uaktywnionego sygnałem JINTD przepisana zostaje poprzednia zawartość licznika adresów do OREG wpisana zostaje instrukcja NOOP wymuszona sygnałem IMACB, natomiast niski poziom IMAC wstrzymuje w tym momencie inkrementację MAC. Ten sam impuls zegarowy powoduje również wyzerowanie przerzutnika INT i ustawienie przerzutnika INTD, który wysyła sygnał INTD definiujący stan przerwania. Wyzerowanie INT powoduje skasowanie sygnału JINTD, a więc i sygnałów IMACB, LOAD i IMAC. Niski poziom JINTD powoduje także wstrzymanie wpisu z kodera na magistrali A. Uaktywniony zostaje natomiast wpis do rejestru przerwań.

Następny impuls zegarowy wpisuje do OREG pierwszą instrukcję podprogramu i powoduje inkrementację MAC. Dalsze wykonywanie odbywa się tak jak w przypadku pracy normalnej z tym, że adresy w przypadku instrukcji skoków z przechowaniem, umieszczane są w rejestrze IJREG.



Rys. 26

64

3. Opis selektora.

3.1. Przeznaczenie i opis struktury logicznej selektora.

Selektor spełnia w pamięci SP45DE następujące zadania:

- 1/ steruje ruchem wybranej głowicy do ścieżki o zadanym adresie, ustawia głowicę i informuje formater o gotowości PLx45D do poprawnego odczytu lub zapisu;
- 2/ wskazuje formaterowi stan PLx45D .
 - a/ niewłaściwa strona /MF/
 - b/ jednostka nie gotowa /UNP/;
- 3/ wybiera dane.

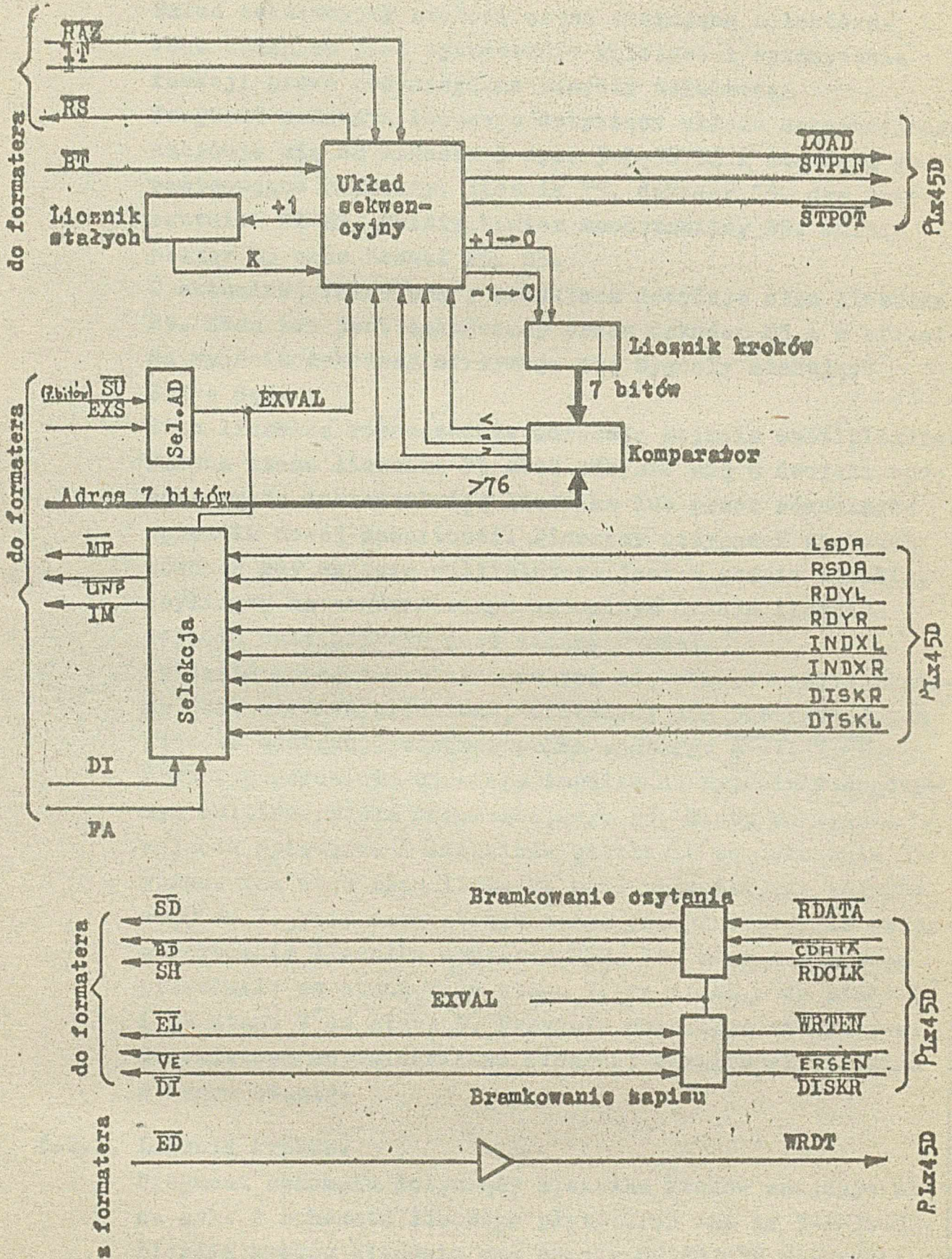
Selektor posiada indywidualny adres. Pozwala to na sterowanie z formatera od jednego do czterech selektorów, z których każdy steruje jedną jednostką PLx45D. W pamięci SP45DE formater pracuje z dwoma selektorami i dwoma jednostkami PLx45D.

Organizację selektora przedstawia schemat blokowy rys. 27 Częścią sterującą jest układ sekwencyjny. Generuje on sekwencje sygnałów sterujących, które uruchamiają poszczególne podzespoły. Sekwencje te mogą się zmieniać, w zależności od zadań jakie ma selektor wykonać.

Cykl pracy selektora rozpoczyna się od porównania adresu selektora wysyłanego przez formater z adresem zakodowanym w układzie adresu. Pozwala to wybrać jeden z dwóch selektorów. Zgodność adresów powoduje uaktywnienie selektora. Układ oczekuje na rozkaz wykonywania EXS i z chwilą jego pojawienia się rozpoczyna pracę. Wewnętrzny rozkaz wykonywania EKVAL uaktywnia układ selekcji, który przekazuje informacje o gotowości do pracy dysku adresowanego sygnałami DI i FA oraz układ bramkowania zapisu i odczytu, umożliwiając wymianę danych i rozkazów między formaterem a PLx45D.

W układzie komparatora następuje porównanie wysyłanego z formatera adresu ścieżki ze stanem licznika kroków, reprezentującego położenie głowicy i w zależności od potrzeb wysłane zostają sygnały sterujące do jednostki PLx45D. Zachowanie wymaganych odstępów czasowych między poszczególnymi sygnałami zapewnia licznik stałych.

O zakończeniu cyklu pracy selektora, formater jest informowany sygnałem \overline{RS} .



Rys. 27

3.2. Zasada pracy układów pakietu selektora.

3.2.1. Układ sekwencyjny.

Układ sekwencyjny stanowi część sterującą selektora. Jego zadaniem jest wyznaczenie kolejności wykonywania funkcji przez poszczególne zespoły selektora.

Fragment schematu ideowego dotyczący układu sekwencyjnego znajduje się na arkuszu 5 rys. 3-L-36782 i stanowią go następujące elementy: licznik F5, dekodery G5, dwa przetrzutniki JK-E5, multiwibrator monostabilny N5, multiplexer D5 oraz bramki P5, G5.

O aktualnej fazie pracy selektora decyduje stan licznika F5. Stan ten jest dekodowany przez dekodery G5 i w efekcie na wyjściu dekodera otrzymuje się sygnały sterujące SQ1 + SQ7.

Stan licznika równocześnie adresuje wejścia multiplexera. Zmiana stanu licznika F5 może odbywać się w dwojaki sposób: przez inkrementację licznika lub przez równoległe wpisanie nowej zawartości. Pierwszy przypadek zachodzi wówczas gdy wyjście multiplexera jest w stanie wysokim, czyli gdy na zaadresowanym aktualnym stanem licznika wejściu multiplexera jest żądany sygnał.

Wpisanie nowej zawartości odbywa się wówczas gdy spełniony jest warunek przeskoku, określony dla poszczególnych stanów. Następuje wówczas przez sygnały: 4→7, 7→3, 6→5 wygenerowanie ujemnego impulsu na wyjściu zanegowanym multiwibratora monostabilnego N5, który uaktywnia wejście wpisujące L licznika. Umożliwia to dokonanie wpisu. Tak więc stan licznika może zmieniać się kolejno od 0 do 7, przy czym stan 0 można osiągnąć jedynie przez wyzerowanie licznika sygnałem FRZ, lub mogą następować przeskoki: ze stanu 4 do stanu 7, ze stanu 6 do stanu 5 i ze stanu 7 do stanu 8. Przyjęta numeracja stanów jest dziesiętnym odpowiednikiem binarnej zawartości licznika w danym stanie.

3.2.2. Licznik kroków.

Fragment schematu dotyczący licznika kroków znajduje się na ark. 2 schematu ideowego płyty SP45 SEL nr 3-L-36782. Licznik kroków stanowią dwa obwody SN 74193N /liczniki rewersyjne/ A2 i A3. W zależności od tego, na które

wejście CU czy CD podawane są impulsy zegarowe, następuje dodawanie jedynki do zawartości licznika lub odejmowanie jedynki od zawartości licznika.

Licznik kroków odwzorowuje położenie głowicy jednostki pamięci PLx45D. Stan licznika reprezentuje numer ścieżki, na której głowica aktualnie się znajduje .

3.2.3. Komparator.

Fragment schematu dotyczący komparatora znajduje się na ark. 2 schematu ideowego 3-L-36782.

Na układ komparatora składają się dwa obwody UCY 7485N - A2, B2 oraz zespół bramek O2, D3, E3.

Komparator ma za zadanie porównanie adresu ścieżki podawanego liniami APO + AP6 ze stanem licznika kroków. Efektem porównania są sygnały wyjściowe komparatora $\overline{AP < CPP}$, $\overline{AP = CPP}$ i $\overline{AP > CPP}$. Jeżeli adres jest mniejszy od zawartości licznika pojawia się sygnał $\overline{AP < CPP}$, jeżeli adres jest większy od zawartości licznika a zawartość licznika mniejsza od 76, to pojawia się sygnał $\overline{AP > CPP}$, jeżeli adres jest równy zawartości licznika lub jeżeli zawartość licznika jest większa od 76, pojawia się sygnał $\overline{AP = CPP}$. Sygnały wyjściowe z komparatora służą do sterowania pracą licznika kroków i układów generujących impulsy STPIN, STPOUT oraz stanowią informację dla układu sekwencyjnego o tym, czy głowica została naprowadzona na zadaną ścieżkę.

3.2.4. Licznik stałych.

Fragment schematu dotyczący licznika stałych znajduje się na ark. 4 schematu ideowego 3-L-36782.

Licznik stałych zbudowany jest z czterech obwodów SN74193 A4, B4, C4, D4. Sterowanie odbywa się sygnałami SQ = 0, FRZZ, SQ = 5, SQ = 6 poprzez układ bramek M4, N4, P4 oraz inwerty E4.

Do licznika wpisywane są stałe poprzez wejścia równoległe poszczególnych obwodów. Poszczególne bity tych stałych tworzone są w układzie bramek A5, R4, P4, N4 oraz inwertorów E4, a ich wartości zależą od aktualnej fazy pracy selektora. Możliwe jest wpisanie następujących stałych /w kodzie 16-tkowym/: FEOO, FD80, EOOO, OFOO, odpowiednio dla czasów 2,5ms, 5ms, 40ms, 100ms.

Cykl pracy licznika stałych jest następujący: sygnał sterujący $SQ = 5$ oraz ujemny impuls zegarowy /BT/2/1 powodują uaktywnienie wejść wpisujących i stała zostaje wpisana do licznika. W momencie gdy sygnał $SQ = 6$ bramkujący zegar licznika BT/2 staje się równy "1" rozpoczyna się zliczanie. Każdy impuls zegarowy podawany na wejście Cu licznika powoduje inkrementację jego zawartości. Zliczanie trwa aż do przepełnienia licznika. Pojawia się wówczas na wyjściu CO obwodu A4 sygnał przepełnienia, który zamyka bramkę P4.

Sygnałem wyjściowym informującym układ sekwencyjny o tym, że stała została obliczona, jest sygnał CPC = F.

Licznik stałych pozwala na uzyskanie stałych czasowych o długości będącej wielokrotnością okresu impulsów zegarowych $BT/2 - 8/\mu s$. Stałe czasowe są podyktowane wymaganiami jednostki PLx45D.

3.2.5. Układ adresu selektora.

Fragment schematu dotyczący układu adresu selektora znajduje się na ark. 8 schematu ideowego 3-L-36782. Składają się na niego bramki: K1, G1 oraz inwentyry B1, C1, N1.

Adres selektora jest determinowany przez kombinację mostków 1, 2 zgodnie z tabelą. Układ porównuje nastawiony adres z adresem przesyłanym liniami 30T, 30Z, w przypadku zgodności tych adresów i przy równoczesnej obecności sygnału EXS pojawia się sygnał EXVAL, który jest rozkazem wykonywania dla selektora.

Nr selektora	Nr mostka	
	1	2
1	-	-
2	+	-
3	-	+
4	+	+

"-" brak mostka

"+" obecność mostka

3.2.6. Układ selekcji.

Fragment schematu dotyczący układu selekcji znajduje się na ark. 6 schematu ideowego 3-2-36782.

Układ składa się z multiplexera M1, przerzutnika J1 oraz zespołu bramek K1, E1, L1, H1 oraz inwertorów P1, N1, F1. Na wejścia układu selekcji podawane są pary sygnałów: LSDA i RSDA, RDYL i RDYR, INDXL i INDXR, DISKR i DISKR/1, odnoszące się odpowiednio do lewego i prawego dysku jednostki PLx45D. W zależności od adresu DI na wejściu M multiplexera wybierany jest jeden sygnał z każdej pary. Sygnały LSDA i RSDA są ponadto porównywane z FA tak, że sygnał MF/1 stanowi informację o zgodności jednego z nich z zadaniem adresem.

Przerzutnik J1 służy do zapamiętania adresu DI, przy czym wpis odbywa się w czasie wysyłania przez układ sekwencyjny sygnału sterującego $Sq = 4$. Zapamiętany adres w postaci sygnału DISKR przesyłany jest do jednostki PLx45D.

3.2.7. Układ bramkowania zapisu i odczytu.

Fragment schematu dotyczącego układu bramkowania znajduje się na ark. 7 schematu ideowego 3-2-36782.

Układ składa się z bramek E1, D1. Sygnał FEXVAL uaktywnia bramki, co pozwala na przekazywanie sygnałów RDATA, RULK, ODATA z dysku do formatera oraz WRLEN i ERSEN z formatera do PLx45.

3.2.8. Układy generujące sygnały LOAD, STPIN, STPOUT.

Odpowiedni fragment schematu znajduje się na ark. 3 schematu ideowego 3-2-36782.

Układ generujący sygnał LOAD składa się z przerzutnika S3 oraz bramek P3, L3 i inwertora R2. Sygnał LOAD jest wysyłany jeżeli przerzutnik S3 zostaje ustawiony. Wpisanie jedynki do przerzutnika następuje w czasie trwania sygnału sterującego $SQ = 5$ pod warunkiem obecności sygnału W100. Wyzerowanie przerzutnika czyli skasowanie sygnału LOAD następuje w czasie trwania sygnału sterującego $SQ = 0$ lub w czasie trwania $SQ = 3$ przy równoczesnym niskim stanie sygnału MLOAD.

Układ generujący impulsy STPIN składa się z przerzutnika S2, bramek P2, P3 i inwertora R2.

Działanie układu jest następujące: sygnał sterujący $SQ = 5$ uaktywnia bramkę P2 i jeżeli spełniony jest warunek $/AP > CPP + FFHOME/ \cdot \overline{RP} = 1$ to najbliższy impuls zegarowy $/BT/2/1$ wpisuje jedynkę do przerzutnika. Na wejściu K tego przerzutnika pojawia się wówczas stan wysoki i następny impuls zegarowy powoduje zmianę stanu przerzutnika. Odpowiada temu wygenerowanie na wyjściu \overline{Q} przerzutnika ujemnego impulsu o czasie trwania równym okresowi impulsów zegarowych.

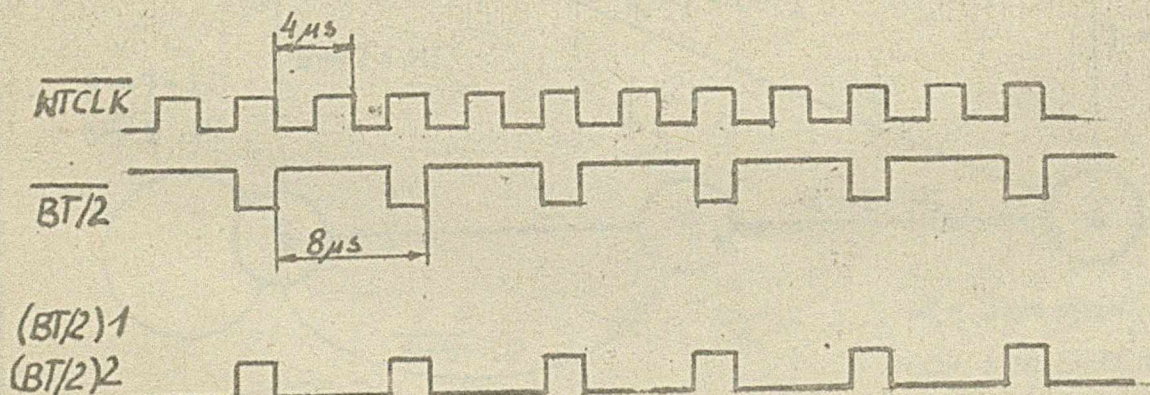
Warunek $/AP > CPP + FFHOME/ \cdot \overline{RP} = 1$ odpowiada następującej sytuacji: zawartość licznika kroków jest mniejsza od adresu lub głowica znajduje się na ścieżce spoczynkowej, a równocześnie nie ma rozkazu sprowadzenia głowicy na ścieżkę spoczynkową.

Układ generujący sygnały $\overline{STPOINT}$ składa się z przerzutnika S2 oraz bramek P2, P3.

Działanie tego układu jest podobne jak układu generującego impulsy \overline{STPIN} . Różnicę stanowi warunek generacji impulsu, który tutaj ma postać $/AP < CPP + RP/ \cdot \overline{FFHOME} = 1$, co odpowiada sytuacji gdy stan licznika kroków jest większy od zadanego adresu lub głowica jest w trakcie sprowadzania na ścieżkę spoczynkową, przy czym jeszcze tej ścieżki nie osiągnęła.

3.2.9. Układy synchronizacji.

Wszystkie przerzutniki i liczniki są wyzwalane impulsami zegarowymi $/BT/2/1$, $/BT/2/2$, $\overline{BT/2}$. Impulsy te są formowane przez przerzutnik J1 i bramkę 4.1 z ciągu impulsów $\overline{BTCLK} /BT/$. Przebiegi czasowe przedstawia rysunek



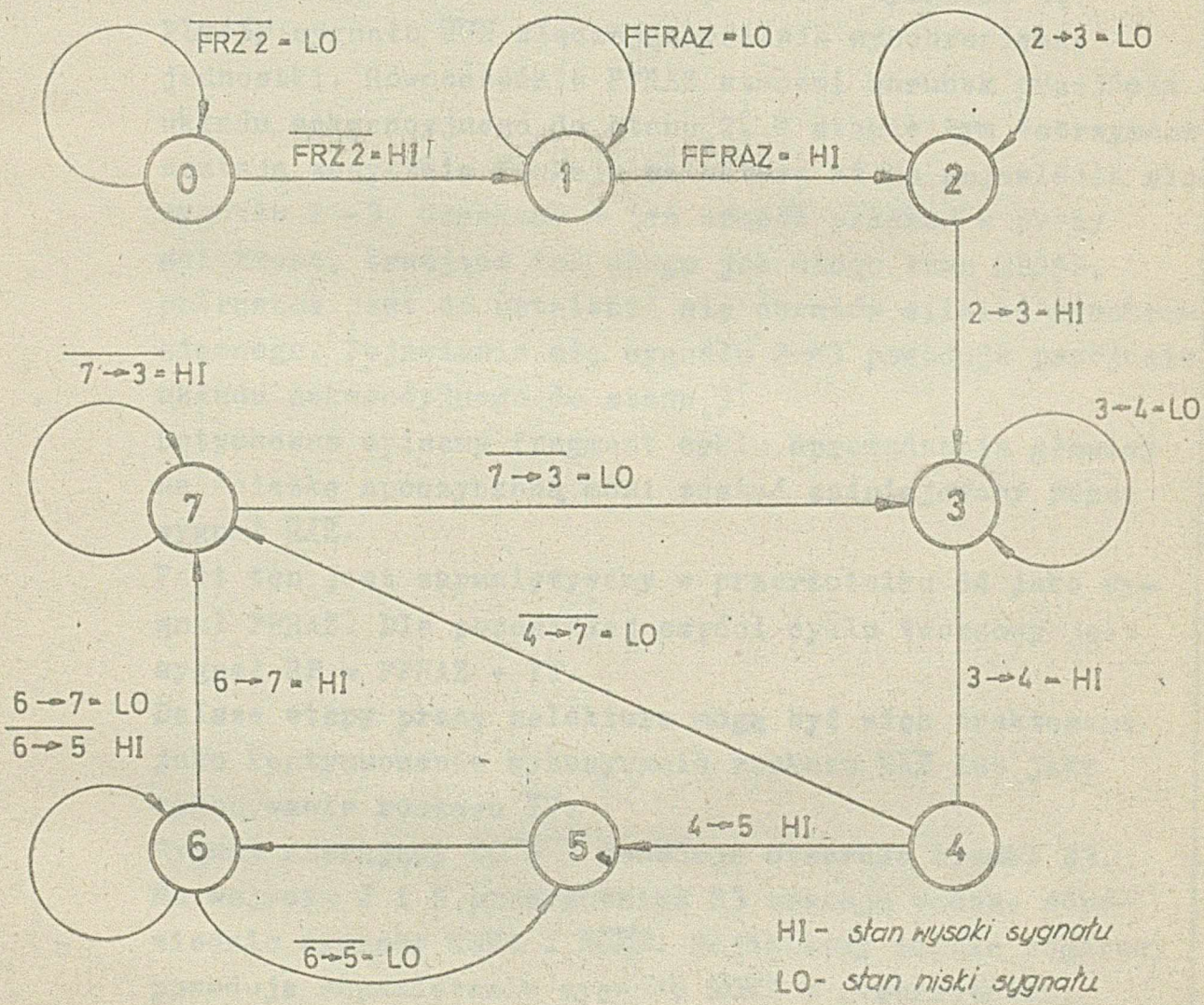
Do synchronizacji sygnałów asynchronicznych służy rejestr L2 i przerzutnik M3. Na wejściach tych obwodów sygnały mogą się zmieniać w dowolnych momentach czasu, natomiast na wyjściach zmiany zachodzą równocześnie z pojawieniem się impulsów zegarowych.

3.3. Opis zasady działania selektora.

Zasada działania selektora omówiona zostanie w oparciu o schemat ideowy 3-2-36782.

Pracę selektora można podzielić na kilka cykli. Cykle te wyznaczone są przez sekwencję stanów sterujących układem sekwencyjnego i zależą od rozkazów wysyłanych z formatera, oraz od położenia głowicy jednostki PLx45D w stosunku do zaadresowanej ścieżki.

Możliwe sekwencje stanów i warunki przejścia z jednego stanu do drugiego przedstawia graf. stanów rys. 29



Rys. 29

3.3.1. Cykl sprowadzania głowicy na ścieżkę spoczynkową.

Cykl ten zostaje zapoczątkowany rozkazem RAZ wysyłanym przez formater. Po synchronizacji rozkaz ten przekształcony zostaje na sygnały FRZ, FRZ2, FRZ3. Powodują one wyzerowanie wszystkich liczników i przerzutników. Tak więc wyzerowany zostaje również licznik F5 układu sekwencyjnego i układ ten znajduje się w stanie "0". Stan "0" trwa tak długo jak długo trwa sygnał FRZ2. Przejście układu do stanu "1" następuje po zniknięciu FRZ2. Wysyłany sygnał $\overline{SQ} = 1$ powoduje wygenerowanie przez multiwibrator M2 impulsu MLOAD synchronizowanego następnie przez przerzutnik M3. Przez cały czas trwania MLOAD sygnał 2→3 jest niski.

Sygnał sterujący $SQ = 1$ podawany jest również na wejście J przerzutnika S4. W związku z tym najbliższy impuls zegarowy powoduje ustawienie przerzutnika. Pojawia się wówczas sygnał FFRAZ. FFRAZ powoduje wysłanie do PLx45D sygnału RUN włączającego silnik synchroniczny jednostki. Równocześnie FFRAZ stanowi warunek przejścia układu sekwencyjnego do stanu "2". W stanie tym wstrzymane zostają wszystkie funkcje selektora aż do pojawienia się sygnału 2→3. Uzyskana w ten sposób przerwa w pracy selektora, trwająca tak długo jak długo trwa MLOAD, potrzebna jest do ustalenia się obrotów silnika synchronicznego. Pojawienie się sygnału 2→3 powoduje przejście układu sekwencyjnego do stanu "3".

Dotychczas opisany fragment cyklu sprowadzania głowicy na ścieżkę spoczynkową musi zostać zainicjowany przez sygnał RAZ.

Fakt ten jest zapamiętywany w przerzutniku S4 jako sygnał FFRAZ. Dla pozostałej części cyklu tworzony jest sygnał $RP = FFRAZ + IT$.

Dalsze etapy pracy selektora mogą być więc traktowane jako kontynuowanie wykonywania rozkazu RAZ lub jako wykonywanie rozkazu IT.

Sygnał sterujący $\overline{SQ} = 3$ powoduje otwarcie bramki R3. Na wejścia J i K przerzutnika S3 zostają podane odpowiednio sygnały HOME i \overline{HOME} . Najbliższy impuls zegarowy powoduje zapamiętanie sygnału HOME w przerzutniku.

Sygnal wyjściowy FFHOME stanowi więc informację o tym, czy głowica znajduje się na ścieżce spoczynkowej, czy nie.

Warunek przejścia do następnego stanu ma postać $\overline{\text{FEVL}} + \text{FFRAZ}$. Ponieważ dla opisywanego cyklu sygnał FFRAZ jest wysoki, więc układ sekwencyjny przechodzi do stanu, 4.

W czasie trwania stanu sterującego, 4 wysyłany jest sygnał $\overline{\text{SQ}} = 4$ uaktywniający bramki H1. Na wejścia J i K przerzutnika J1 podane zostają odpowiednio sygnały DI i $\overline{\text{DI}}$. Najbliższy impuls zegarowy powoduje więc wpisanie do przerzutnika adresu DI. Adres ten jako sygnał $\overline{\text{DISKR}}$ jest wysyłany do jednostki, PLx45D. Ponadto w przypadku wykonywania rozkazu IT sygnał $\overline{\text{SQ}} = 4$ powoduje uaktywnienie bramki D2. Umożliwia to wyzerowanie licznika kroków najbliższym wysokim stanem sygnału /BT/2/2 /jeżeli wykonywany jest rozkaz RAZ licznik został już wyzerowany w stanie 0/. Warunkiem przejścia układu sekwencyjnego do następnego stanu jest wysoki poziom sygnału $\text{RP} + \text{FUENF}$. Dla cyklu sprowadzania głowicy na ścieżkę spoczynkową RP jest wysokie, a więc warunek jest spełniony i układ przechodzi do stanu, 5.

W czasie trwania sygnału sterującego $\overline{\text{SQ}} = 5$ uaktywniona zostaje jedna z bramek P2, druga natomiast jest blokowana niskim poziomem RP. Jeżeli teraz FFHOME jest wysokie, wysokie staje się również wejście J przerzutnika S2 i wygenerowany zostaje impuls STFOUT, który powoduje przesunięcie głowicy PLx45D na ścieżkę o adresie o 1 mniejszym od poprzedniego.

Jeżeli FFHOME jest niskie /głowica na ścieżce spoczynkowej/ wejście J przerzutnika pozostaje niskie i impuls nie zostaje wygenerowany.

Sygnał $\overline{\text{SQ}} = 5$ powoduje otwarcie bramki N4. Impuls zegarowy /BT/2/1 podawany jest wówczas na wejście L licznika D4 i powoduje wpisanie stałej do licznika kroków. Wartość stałej zależy od tego czy głowica znajduje się na ścieżce spoczynkowej czy nie. Jeżeli FFHOME jest niski wpisywana zostaje stała F80, jeżeli jest wysoki stała E80. Stan, 5 trwa przez jeden okres impulsów zegarowych, po czym układ sekwencyjny przechodzi do stanu, 6.

Stan sterujący „6” wyznacza kolejny etap pracy selektora - obliczanie stałej czasowej. Odbywa się ono w sposób następujący:

sygnał $SQ = 6$ przy równoczesnym wysokim poziomie $CPC = F$ powoduje otwarcie bramki P4. Impulsy zegarowe są wówczas podawane na wejście CU licznika D4. Każdy kolejny impuls powoduje inkrementację licznika stałych. Zliczanie zostaje zakończone z chwilą przepełnienia licznika. Pojawia się wówczas sygnał $CPC = F$, który blokuje bramkę P4. Czas zliczania wynosi ok. 5 ms w przypadku gdy wpisana była stała ED80. Jest to odstęp czasu między kolejnymi impulsami STPOUT.

Jeżeli wpisana była stała ED80 zliczanie trwa przez ok. 38 ms.

Sygnał $CPC = F$ stanowi warunek przejścia układu sekwencyjnego do innego stanu. Przejście może nastąpić do stanu „5” jeżeli FFHOME jest wysoki /głowica nie jest na ścieżce spoczynkowej/ lub stanu „7” jeżeli FFHOME jest wysoki. W pierwszym przypadku powtarza się fragment cyklu sterowany sygnałami $SQ = 5$ i $SQ = 6$ /wygenerowany zostaje impuls STPOUT, wpisana stała do licznika kroków, a następnie obliczona stała czasowa/. Trwa to dotąd, dokąd FFHOME nie stanie się niski.

Sygnał $\overline{6 \rightarrow 5}$ powodujący przejście układu sekwencyjnego ze stanu „6” do stanu „5” uaktywnia również bramki R3. Dzięki temu równocześnie ze zmianą stanu następuje wpisanie do przerzutnika S3 aktualnej wartości sygnału HOME.

Jeżeli układ sekwencyjny znajduje się w stanie „6” i FFHOME jest niski, to następnym stanem jest stan „7”.

W stanie „7” wyzerowany zostaje przerzutnik S4, tym samym sygnał FFRAZ staje się niski.

Sygnał $SQ = 7$ uaktywnia bramkę N3. Dzięki temu impulsy zegarowe podawane są na wejście B multiwibratora M2. Przednie zbocze pierwszego z nich powoduje wygenerowanie impulsu MLOAD.

Warunkiem przejścia do następnego stanu jest wysoki poziom sygnału RS/7. Warunek ten jest spełniony w czasie całego cyklu prowadzenia głowicy na ścieżkę spoczynkową.

Stan „7” trwa więc tylko przez jeden okres impulsów zegarowych, a następnie układ przechodzi do stanu „3”.

Sygnal $SQ = 3$ powoduje wpisanie aktualnej wartości HOME do przerzutnika S3, a także uaktywnia bramkę L3. Jeżeli stan „3” trwa dłużej od zainicjowanego w stanie „7” impulsu MLOAD, to z chwilą gdy MLOAD stanie się na powrót niski, na wejściu K przerzutnika S3 pojawia się „1”. Najbliższy impuls zegarowy powoduje więc wyzerowanie przerzutnika. Do PLx45D zostaje wysłany sygnał zwalniający ewentualny docisk głowicy. Stan „3” kończy cykl sprowadzania głowicy na ścieżkę spoczynkową. W stanie tym układ sekwencyjny oczekuje na sygnał FEXVAL, który stanowi warunek przejścia do stanu „4” i tym samym rozpoczyna następny cykl pracy selektora.

3.3.2. Sprowadzanie głowicy na ścieżkę 0 zadanym adresie.

Cykl ten rozpoczyna się od stanu „3”. Układ sekwencyjny oczekuje na sygnał FEXVAL, który stanowi warunek przejścia do stanu „4”. Przed pojawieniem się tego sygnału formater musi wysłać adres selektora SU1, SU2, adres dysku DI, adres strony dysku FA oraz adres ścieżki $AP0 + AP6$. Jeżeli adres SU1, SU2 jest zgodny z adresem selektora determinowanym przez kombinację mostków 1, 2, to z chwilą wysłania przez formater rozkazu wykonywania EXS pojawia się sygnał EXVAL.

EXVAL uaktywnia układ selekcji i do formatera wysyłane są sygnały MF, UNP, IM. Poziom sygnału UHT zależy od tego, czy aktualny adres DI jest zgodny z adresem DISKR zapamiętanym w przerzutniku J1 pochodzącym z poprzedniego cyklu. Jeżeli oba adresy nie są zgodne UHT staje się niski i powoduje wyzerowanie przerzutnika S3, a tym samym wysłanie do PLx45D wysokiego sygnału LOAD.

EXVAL jest synchronizowany w rejestrze L2 i w wyniku tego pojawiają się sygnały FEXVAL i \overline{FEXVAL} . FEXVAL powoduje uaktywnienie bramek E1, D1 układu zapisu i odczytu. Umożliwia to wymianę informacji między PLx45D a formaterem. \overline{FEXVAL} powoduje przejście układu do stanu „4”.

W stanie „4” następuje wpisanie do przerzutnika J1 adresu DI. Sygnał UHT staje się wysoki zwalniając zerowanie przerzutnika LOAD /S3/.

Dalsza część cyklu jest uzależniona od sygnału 4→5 i 4→7.

Sygnal $4 \rightarrow 7$ powodujący przeskok do stanu 7, pojawia się wówczas, gdy $4 \rightarrow 5$ jest wysokie, czyli gdy nie jest spełniony warunek przejścia do stanu 5. Tak więc przeskok do stanu 7 ma miejsce wówczas, gdy zbędne są funkcje wykonywane przez selektor w stanach 5 i 6.

Przejście do stanu 5 odbywa się pod warunkiem $4 \rightarrow 5 = 1$, co jest równoznaczne dla opisywanego cyklu z niskim poziomem sygnału $\overline{FUENF} / \overline{RP}$ jest cały czas równe 1/. Ponieważ $\overline{FUENF} = \overline{FFHOME} + \overline{AP=CPP} + \overline{LOAD} + \overline{MF/1} + \overline{UNP}$, a więc warunek $4 \rightarrow 5 = 1$ jest spełniony gdy sygnały MF/1 i UNP są niskie i równocześnie wysoki jest przynajmniej jeden z sygnałów FFHOME, $\overline{AP=CPP}$, \overline{LOAD} . Odpowiada to sytuacji gdy adres strony dysku jest zgodny z $\overline{FA} / \overline{MF/1} = 0/$, jednostka PLx45D jest przygotowana do pracy $\overline{UNP} = 0/$ i równocześnie: albo głowica znajduje się na ścieżce spoczynkowej $\overline{FFHOME} = 1/$ albo zawartość licznika kroków nie jest zgodna z adresem ścieżki $\overline{AP=CPP} = 1/$ albo głowica nie jest dociągnięta $\overline{LOAD} = 1/$.

W stanie 5 wygenerowany zostaje impuls \overline{STPIN} /gdy $\overline{AP} > \overline{CPP} = 0$ lub $\overline{FFHOME} = 0/$ lub \overline{STPOUT} /gdy $\overline{AP} < \overline{CPP} = 0$ i $\overline{FFHOME} = 1/$. Ponadto sygnał sterujący $\overline{Sq} = 5$ powoduje wpisanie stałej do licznika stałych. Wartość stałej zależy od tego, czy adres ścieżki jest zgodny z zawartością licznika kroków, czy nie. Jeżeli nie jest zgodny $\overline{AP} = \overline{CPP}$ - niskie/ to wpisywana jest stała ED80. Najmniej znaczące bity tej stałej zależą od stanu przerzutnika S4. Przerzutnik ten jest zerowany pod koniec każdego cyklu w stanie 7. Dla rozpatrywanego momentu przerzutnik jest więc wyzerowany.

Przejście do stanu 6 odbywa się bezwarunkowo.

W stanie 6 następuje ustawienie przerzutnika S4.

Ponadto sygnał sterujący $\overline{Sq} = 6$ powoduje uruchomienie licznika stałych. Zliczanie trwa przez ok. 5 ms do przepełnienia licznika. Sygnał przepełnienia $\overline{CPC} = \overline{F}$ uaktywnia bramkę D3. Równocześnie wysoki poziom $\overline{Sq} = 6$ powoduje otwarcie bramki C3. Impuls zegarowy $\overline{BT/2/2}$ podany zostaje na jedno z wejść: CU /w przypadku $\overline{AP} > \overline{CPP} = 1/$ lub CD $\overline{AP} < \overline{CPP} = 0/$ licznika kroków, powodując jego inkrementację lub dekrementację. Jeżeli głowica znajduje się na ścieżce spoczynkowej sygnał \overline{FFHOME} powoduje blokowanie bramki C3 i nie następuje powiększenie zawartości

licznika. Równocześnie ze zmianą zawartości licznika kroków następuje wpisanie aktualnej wartości HOME do przerzutnika S3 i przeskok układu sekwencyjnego do stanu „5”.

Dalsza praca selektora odbywa się identycznie jak poprzednio w stanie „5”. Zmianie ulega jedynie wartość wpisywanej do licznika stałej, która wynosi teraz FE00. Zmiana ta spowodowana jest innym niż poprzednio stanem przerzutnika S4.

Stan „6”, który następuje po stanie „5” różni się od poprzednio opisanego jedynie wartością stałej czasowej, która wynosi ok. 2,5 ms.

Układ sekwencyjny oscyluje między stanem „5” i „6” tak długo, jak długo zawartość licznika kroków nie stanie się zgodna z zadaniem adresem ścieżki. Z chwilą pojawienia się sygnału AP = CPP sygnał W100 staje się wysoki i w czasie trwania stanu „5” następuje ustawienie przerzutnika S3. Do PLx45D wysyłany zostaje LOAD powodujący dociśnięcie głowicy. W tym samym momencie następuje wpis do licznika stałych i przeskok układu do stanu „6”.

Wartość stałej zależy od sygnału LOAD/T. Jeżeli LOAD/T = 1 /głowica nie była dociśnięta/ wpisywana jest stała CF00, jeżeli LOAD/T = 0 /głowica była już dociśnięta/ - stała EC00.

W stanie „6” następuje odliczenie stałej czasowej, która wynosi w pierwszym przypadku ok. 100 ms, a w drugim ok. 41 ms. Czasy te potrzebne są dla uspokojenia pozycjonera.

Po odliczeniu stałej czasowej pojawia się sygnał CPC = F i powoduje wysoki poziom sygnału 6→7. Tym samym spełniony zostaje warunek przejścia do stanu „7”. Sygnał 6→7 powoduje również pojawienie się wysokiego poziomu na wejściu J przerzutnika M3. W związku z tym równocześnie ze zmianą stanu układu sekwencyjnego przerzutnik M3 zostaje ustawiony i do formatera zostaje wysłany sygnał RS informujący o zakończeniu cyklu pracy selektora.

Układ sekwencyjny pozostaje w stanie „7” aż do skasowania rozkazu wykonywania EXS. W momencie gdy EXS staje się wysoki przerzutnik M3 zostaje wyzerowany niskim poziomem

EXVAL. Selektor przestaje wysyłać RS, a równocześnie niski poziom osiąga sygnał RS/T. Sygnał ten po synchronizacji jako ERS/1 powoduje przejście układu do stanu 3. Stan ten kończy cykl pracy selektera, który oczekuje na następny rozkaz wykonywania.

EXVAL. PRZEJŚCIE
ADWALNY POROZUMIENIE
M.P. - KAP
M.P. - KAP

Przebieg pracy selektera
w warunkach i warunkach typ. 15 20

Fig. 3. Przebieg pracy

Przebieg pracy