

WROCLAWSKIE ZAKŁADY ELEKTRONICZNE ELWRO

mgr inż. Jur Lesiński

ODRA 1103

Opis Ogólny

1967

1103-I-1

S P I S T R E Ś C I

- 0. Wstęp.
- 1. Organizacja logiczna..
 - 1.1. Słowo liczbowe i rozkazowe w pamięci operacyjnej.
 - 1.1.1. Przedstawienie w pamięci liczb dziesiętnych.
 - 1.1.2. Przedstawienie w pamięci liczb binarnych.
 - 1.2. Opis zasadniczych bloków Kalkulatora.
 - 1.2.1. Bloki Kalkulatora.
 - 1.2.2. Blok Wejścia - Wyjścia.
 - 1.2.3. Blok pamięci ferrytowej.
 - 1.2.4. Blok sterowania.
 - 1.2.5. Blok arytmometru.
 - 1.2.6. Ręczne sterowanie Kalkulatora.
- 2. Arytmetyka i algorytmy zastosowane w Kalkulatorze.
 - 2.1. Arytmetyka dla liczb dziesiętnych.
 - 2.1.1. Operacje dodawania.
 - 2.1.2. Operacje mnożenia.
 - 2.1.3. Operacje dzielenia.
 - 2.2. Arytmetyka operacji binarnych.
 - 2.2.1. Operacje binarne, których wynik pozostaje w akumulatorze.
 - 2.2.2. Operacja binarna, której wynik zostaje przesłany do komórki pamięci ferrytowej.
 - 2.3. Algorytm operacji przesunięć krótkich.
- 3. Lista rozkazów Kalkulatora.
 - 3.1. Oznaczenia i skróty.
 - 3.2. Zasady przyporządkowania kodów.
 - 3.3. Tabele rozkazów.
 - 3.4. Rozkazy ujęte w tabeli pierwszej.
 - 3.4.1. Rozkazy skoków.
 - 3.4.2. Operacje na parametrze N.
 - 3.4.3. Operacje na binarnej zawartości komórek technicznych.
 - 3.4.4. Operacje ułatwiające organizowanie podprogramów.
 - 3.5. Rozkazy ujęte w tabeli drugiej.
 - 3.5.1. Operacje dziesiętne "krótkie".
 - 3.5.2. Operacje dziesiętne "długie".
 - 3.6. Rozkazy ujęte w tabeli trzeciej.
 - 3.6.1. Operacje przesunięć krótkich.
 - 3.6.2. Operacje pomocnicze dla podprogramów mnożenia i dzielenia.
 - 3.6.3. Operacje specjalne.

- 3.6.4. Operacje Wejścia - Wyjścia.
- 4. Współpraca Kalkulatora z urządzeniami Wejścia - Wyjścia.
- 4.1. Współpraca Kalkulatora z zewnętrzną pamięcią bębnową.
- 4.2. Współpraca Kalkulatora z urządzeniami taśmy papierowej i dalekopisem.
- 4.2.1. Perforowanie taśmy i drukowanie na dalekopisie.
- 4.2.2. Czytanie taśmy.
- 4.2.3. Wczytanie jednego znaku dalekopisowego.
- 4.3. Współpraca Kalkulatora z urządzeniami kartowymi.
- 4.3.1. Uwagi ogólne.
- 4.3.2. Współpraca z reproducerem PR80-2.
- 4.3.3. Współpraca z tabulatorem T5-M.
- 4.3.4. Organizacja logiczna współpracy Kalkulatora z urządzeniami kartowymi.
- 4.3.5. Programy wykorzystujące urządzenia kartowe.

W s t ę p.

Kalkulator Elektroniczny ODRA 1103 jest znakową maszyną cyfrową przeznaczoną głównie do współpracy z urządzeniami kartowymi. Dzięki odpowiedniej liście rozkazów i wyposażeniu w zewnętrzną pamięć bębnową. Kalkulator zastosowany w stacjach maszyn analitycznych znacznie skróci czas obliczeń przy jednoczesnym kilkakrotnym zmniejszeniu ilości kart-obiegów.

W Kalkulatorze zastosowano technikę popularnych polskich maszyn ODRA 1003 i 1013.

Również zewnętrznie Kalkulator niewiele różni się od tych maszyn. Pulpit sterowania zmontowano jako osobny zespół do ustawienia na biurku operatora, dla większej wygody przy obsłudze.

Kalkulator ODRA 1103 jest maszyną szeregową, sekwencyjną, pracującą w arytmetyce dwójkowej i dziesiętnej /dwójkowo kodowanej/.

Kalkulator wyposażony jest w układy przejścia dla sterowania reproducerem PR80-2 i tabulatorem T5-M oraz w urządzenia Wejścia - Wyjścia taśmy papierowej.

Podstawowe parametry użytkowe Kalkulatora:

Czasy wykonywania operacji dziesiętnych:

czas dodawania /8+8 cyfr/	- 430 u sek.
/16+16 cyfr/	- 770 u sek.
czas mnożenia /7 . 8 cyfr/	- 5,9 msek.
czas dzielenia /15 : 7 cyfr/	- 15 msek. /średnio/

Pamięci:

Pamięć operacyjna ferrytowa: 1024 komórek 16+1-bitowych.

Pamięć zewnętrzna bębnowa: 1 bęben o pojemności 32768 komórek 16+1 bitowych. Średni czas dostępu do informacji na bębnie - 11 msek.

Wejście:

czytnik kart reproducera lub tabulatora,

czytnik taśmy papierowej,

dalekopis.

Wyjście:

dziurkarka kart reproducera lub aparat piszący tabulatora,

perforator taśmy,

dalekopis,

1. Organizacja logiczna.

1.1. Słowo liczbowe i rozkazowe w pamięci operacyjnej.

Pamięć operacyjna jest blokiem ferrytowym o pojemności 1024 komórek technicznych 16-bitowych, lub 4096 znaków. W jednej komórce można zapamiętać:

- a/ 4 cyfry liczby dziesiętnej /cyfry zapisane są w kodzie dziesiętnym prostym/,
- b/ 16 bitów liczby zapisanej w kodzie dwójkowym,
- c/ jeden rozkaz.

Podstawową, indywidualnie adresowaną jednostką pamięci jest znak - składający się z czterech bitów, tzw. tetrada, w której pamiętana może być jedna cyfra dziesiętna, lub też jakaś inna informacja binarna.

W dalszej części opisu zamiast określenia z n a k używane będzie określenie t e t r a d a.

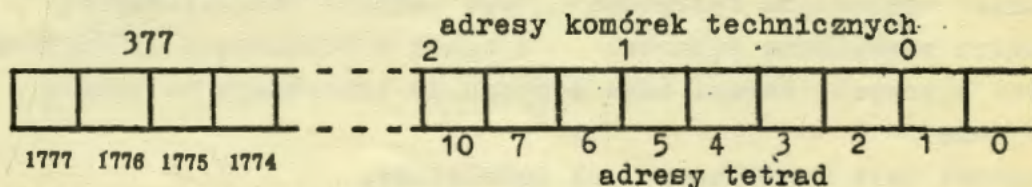
Pamięć operacyjna maszyny adresowana jest za pomocą liczb ósemkowych.

Cztery kolejne tetrydy stanowią jedną komórkę techniczną.

Pamięć maszyny posiada podwójną adresację:

- istnieją adresy "duże", które są adresami komórek technicznych,
- adresy "małe" stanowiące adresy poszczególnych tetrad.

Schemat i kierunek adresowania pamięci podaje poniższy rysunek:



Adresy w pierwszej strefie pamięci ferrytowej /B=0/

Rys. 1

Cała pamięć jest bezpośrednio dostępna przy pomocy adresów komórek technicznych, natomiast bezpośredni dostęp do tetrad istnieje w strefie 256 komórek technicznych.

Dostęp do tetrad w całej pamięci maszyny osiągany jest za pomocą dwubitowego rejestru B, lewostronnie rozszerzającego adresy "małe" /patrz pkt 3.5/.

1.1.1. Przedstawienie w pamięci liczb dziesiętnych.

Liczby dziesiętne pamiętane są w pamięci operacyjnej jako ciągi znaków - tetrad. W każdej tetradzie pamiętana jest jedna cyfra dziesiętna liczby zapisana w kodzie dziesiętnym, prostym.

Miejsce liczby w pamięci określane jest przez:

- a/ adres tetrady, w której zapisana jest najmniej znacząca cyfra liczby,
- b/ ilość cyfr liczby /długość liczby w tetradach/.

Liczby mogą tworzyć jedynie tetrady położone obok siebie tzn. posiadające kolejne adresy.

W samym słowie liczbowym nie ma żadnej informacji wskazującej na jego początek lub koniec. Programista może więc w zależności od potrzeby uważać ciąg tetrad za jedno słowo lub za zbiór kilku słów.

Przykład 1.

Liczbę 6710 można zapamiętać np. w następujący sposób:

0114	0113	0112	0111	0110	0107	0106	0105
8	1	6	7	1	0	2	5

Rys. 2

Wtedy adres jej jest równy 0107, a ilość tetrad jest równa 4. Liczba ta może stanowić część większej liczby np. 167102. Adres tej liczby jest 0106, natomiast ilość tetrad równa 6.

1.1.1.1. Długość - ilość cyfr liczby dziesiętnej.

Długość liczby podawana jest w rozkazie /patrz pkt 3.5/ wynosi ona od 1 do 8 tetrad, lub dla podwójnej dokładności od 9 do 16 tetrad.

1.1.1.2. Znak liczby dziesiętnej.

Znak liczby pamiętany jest na ogół w jej najbardziej znaczącej tetradzie /patrz przykład 2/.

Liczba uważana jest za ujemną, jeśli wszystkie bity jednej z tetrad liczby są jedynkami.

Wartością numeryczną tetrady, w której zapamiętany jest

znak "-" /tetrady znakowej/ jest zero.

W najbardziej znaczącej tetradzie liczby dodatniej pamiętana jest cyfra znacząca /w ten sam sposób jak w mniej znaczących tetradach liczby/.

Przykład 2.

Liczbę -67102 można zapamiętać np. w niżej podany sposób:

Adresy tetrad	0115	0114	0113	0112	0111	0110	0107	0106	0105
Informacja binarna	1000	0001	1111	0110	0111	0001	0000	0010	0101
Wartość numeryczna	8	1	-	6	7	1	0	2	5

Rys. 3

Adres jej jest równy 0106, a długość wynosi 6 tetrad.

U w a g a:

a/ znak "-" może być pamiętany "wewnątrz" liczby.

Wtedy liczba o adresie 0106 i długości 7 tetrad odczytana będzie jako: -1067102,

b/ wartość numeryczną 0 mają również tetrad odpowiadające naddziurce drugiej i brakowi otworu w kolumnie karty /patrz 3.6.4.2/.

1.1.1.3. Przedział liczb dziesiętnych.

Ponieważ najbardziej znacząca tetrad liczby ujemnej na ogół zajęta jest przez znak, więc liczby ujemne, na których można w maszynie wykonywać działania, są o jedną cyfrę krótsze od dodatnich. Odpowiednie przedziały wynoszą:

a/ dla pojedynczej dokładności

$$-10^7 + 1 \leq x \leq 10^8 - 1$$

b/ dla podwójnej dokładności

$$-10^{15} + 1 \leq x \leq 10^{16} - 1$$

1.1.2. Przedstawienie w pamięci liczb binarnych.

Liczba binarna zajmuje w pamięci 16 bitów jednej komórki technicznej.

Liczba binarna w pamięci traktowana jest przez maszynę zawsze jako dodatnia.

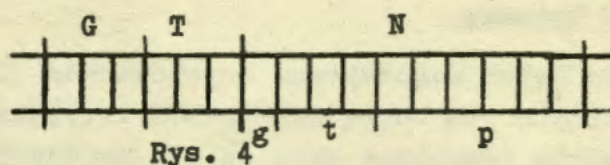
Adresem liczby binarnej jest adres komórki technicznej, w której jest ona zapisana. Liczby binarne w całej pamięci operacyjnej są bezpośrednio dostępne.

1.1.3. Przedstawienie w pamięci rozkazu.

Rozkaz zajmuje jedną komórkę techniczną, tj. 4 tetrazy, co łącznie stanowi 16 bitów.

Rozkazy w maszynie przedstawione są w postaci binarnej.

Budowę rozkazu przedstawia poniższy rysunek:



gdzie: 6-bitowa część GT oznacza kod operacji, zaś część adresowa N składa się z 10-ciu bitów.

Dla rozkazów bezadresowych w części adresowej N wyróżnia się następujące części: g, t, p. /patrz tabela rozkazów/.

Dokładne omówienie tabeli rozkazów znajduje się w dalszej części opisu.

1.2. Opis zasadniczych bloków Kalkulatora.

1.2.1. Bloki Kalkulatora.

W Kalkulatorze ODRA 1103 można pod względem funkcjonalnym wyróżnić następujące bloki:

- a/ blok Wejścia - Wyjścia
- b/ blok pamięci ferrytowej
- c/ blok sterowania
- d/ blok arytmometru.

Na rysunku 5 pokazano zasadniczą strukturę bloków oraz główne powiązania między nimi.

Informacja w Kalkulatorze jest na ogół przesyłana szeregowo tzn., że do układów maszyny przesyła się w kolejnych odstępach czasu kolejne bity.

Czas przekazywania jednego bitu nazywa się chwilą i trwa nieco ponad 4,65 usek. /tj. 1/4608 okresu obrotu bębna/.

1.2.2. Blok Wejścia - Wyjścia.

Kalkulator ODRA 1103 może współpracować z następującymi urządzeniami zewnętrznymi:

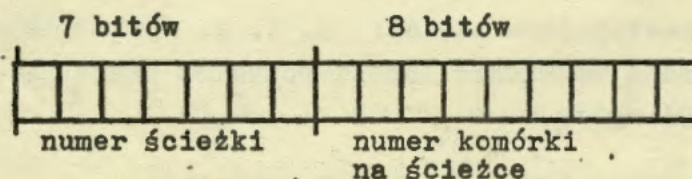
- a/ pamięć bębnowa,
- b/ reproducer PR80-2,
- c/ czytnik taśmy FC-11,
- d/ perforator taśmy FACIT,
- e/ dalekopis Siemens-100.

1.2.2.1. Pamięć bębnowa.

Jest to bęben magnetyczny o pojemności 32768 komórek 16-bitowych /co odpowiada 131072 tetradom/.

Informacja pamiętana jest na 128 ścieżkach /ponumerowanych ósemkowo od 0 do 177/ po dwieście pięćdziesiąt sześć komórek /ponumerowanych ósemkowo od 0 do 377/ każda.

Komórki na bębnie adresowane są binarnie.



Rys. 6 Adresowanie komórek na bębnie.

Na ścieżce o numerze 0 znajdują się więc komórki o adresach:

dziesiętnie:	od 0 do 255
ósemkowo:	od 0 do 377

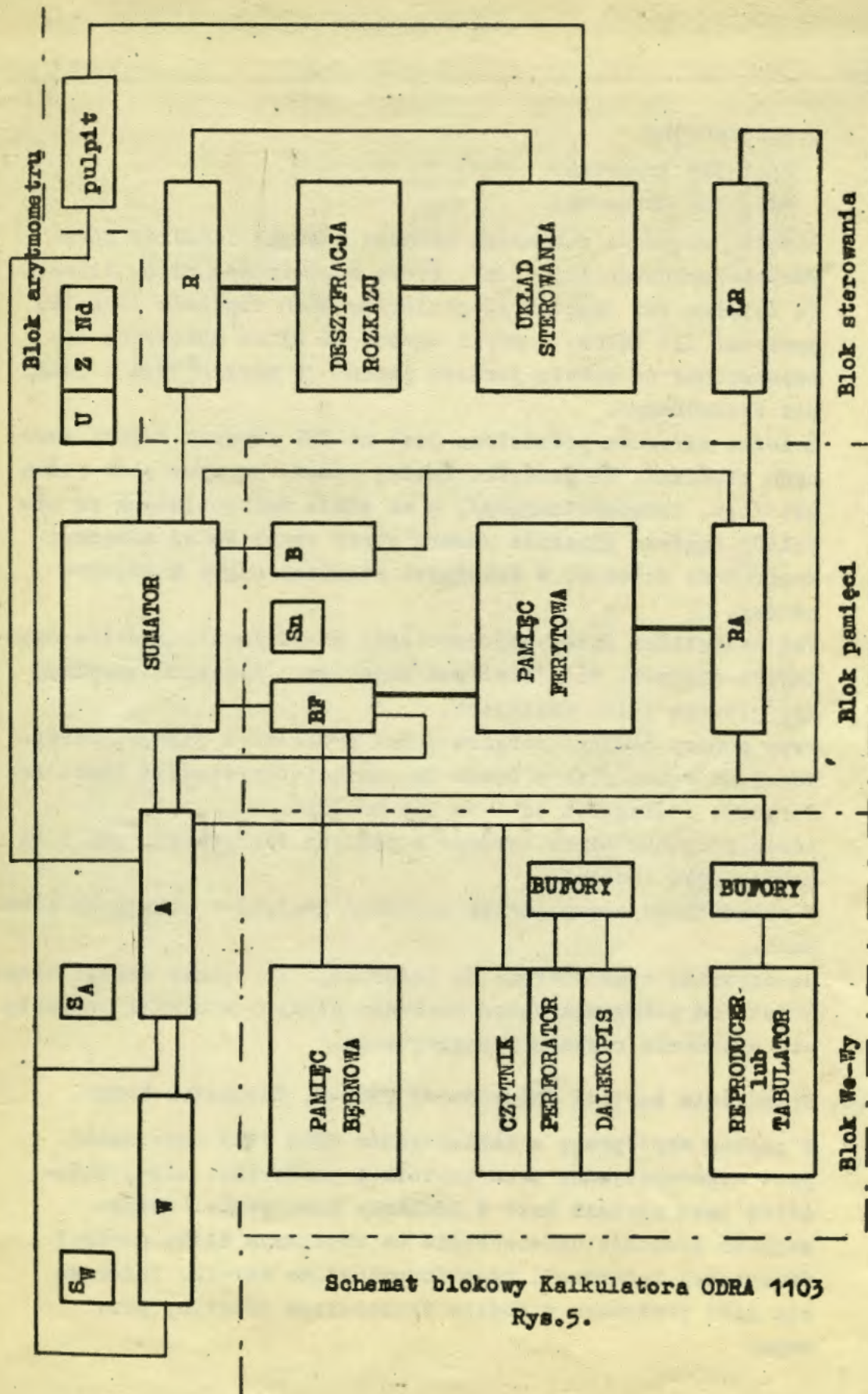
Na ścieżce pierwszej znajdują się komórki o adresach

dziesiętnie:	od 256 do 511
ósemkowo:	od 400 do 777

Na ścieżce o numerze dziesiętnym 127 /ósemkowo 177/ znajdują się komórki o adresach:

dziesiętnie:	od 32512 do 32767
ósemkowo:	od 77400 do 77777

Ścieżki bębna, na których pamięta się informację /rozkazy i liczby/ nazywa się roboczymi. Oprócz ścieżek roboczych /i zapasowych ścieżek/ znajdują się na bębnie 2 ścieżki



Schemat blokowy Kalkulatora ODRA 1103

Rys.5.

organizacyjne:

- ścieżka zegarowa,
- ścieżka adresowa.

Ścieżka zegarowa dostarcza układom maszyny impulsów zegarowych /synchronizujących/, które odpowiadają bitom słowa. Na ścieżce tej nagrano równomiernie 4608 impulsów /256·18, ponieważ dla każdej komórki oprócz 16 bitów roboczych zarezerwowano na bębnie jeszcze jeden bit parzystości i jeden bit techniczny/.

Ścieżka adresowa podzielona jest na 256 równych części zwanych strefami. Na początku każdej strefy nagrany jest jeden bit /tzw. synchronizujący/, a na ośmiu następujących po nim bitach nagrano binarnie numery stref zwane dalej adresami komórek na ścieżce. W kolejnych strefach nagrano kolejne adresy.

Nad wszystkimi ścieżkami roboczymi znajdują się głowice czytająco-piszące. Nad ścieżkami zegarową i adresową znajdują się głowice tylko czytające.

Przy pomocy jednego rozkazu można przesłać z pamięci ferrytowej na bęben /lub z bębna do pamięci ferrytowej/ bloki informacji o długości od 1 do 256 komórek.

Adres początku bloku zarówno w pamięci ferrytowej, jak i na bębnie jest dowolny.

W skład bloku muszą jednak wchodzić komórki o kolejnych adresach.

Rzeczywisty czas dostępu do informacji na bębnie zależy oczywiście od położenia bębna względem głowicy w chwili rozpoczęcia wykonania rozkazu przepisywania.

1.2.2.2. Urządzenie kartowe /Reproducer PR80-2, Tabulator T-5M/.

W czasie współpracy z Kalkulatorem ODRA 1103 reproducer jest wykorzystywany jako czytnik i perforator kart, tabulator jako czytnik kart i drukarka numeryczna. Rozkaz wejścia powoduje umieszczenie we wskazanym bloku pamięci ferrytowej informacji wyperforowanej na karcie. Informacja jest pamiętana w kodzie dziesiętnym prostym, przy czym:

1 - nadziurce /pozycja 11/ odpowiada kod 1111

2 - nadziurce /pozycja 12/ kod 1110. Brakowi dziurki /spacji/ odpowiada kod 1100.

Rozkaz wyjścia powoduje wyperforowanie /wydrukowanie/ informacji zapisanej we wskazanym obszarze pamięci ferrytowej.

Przy pracy z reproducerem rozkazy te powodują ponadto wczytanie kontrolne, perforowanej lub czytanej w poprzednim cyklu karty, do pamięci ferrytowej od komórki 32 do 51.

U w a g a:

Karta może być wczytana jedynie do /lub wyperforowana z/ komórek o adresach mniejszych od 512 /dziesiętnie/.

1.2.2.3. Urządzenia taśmy papierowej.

Urządzeniem wprowadzającym informację z taśmy papierowej jest czytnik taśmy "Elwro" FC-11. Informacja zawarta w jednym rzędku taśmy zostaje wprowadzona na 5 najmniej znaczących bitów akumulatora.

Informację wyprowadza się na taśmę za pomocą perforatora taśmy FACIT PE1501. Rozkaz perforowania taśmy powoduje wyperforowanie na taśmie informacji z 5-ciu najmniej znaczących bitów akumulatora.

Wprowadzanie /wyprowadzanie/ większej ilości rzędków taśmy i kompletowanie ich w słowa odbywa się za pomocą specjalnych programów wprowadzających /wyprowadzających/.

1.2.2.4. Dalekopis.

Jako urządzenie Wejścia - Wyjścia może być również użyty dalekopis. Kod binarny znaku naciśniętego na klawiaturze zostaje wprowadzony na 5 najmniej znaczących bitów akumulatora. Przy wprowadzaniu informacji wydrukowany zostaje znak, którego kod binarny odpowiada informacji z 5-ciu najmniej znaczących bitów akumulatora.

1.2.3. Blok pamięci ferrytowej.

Z logicznego punktu widzenia, można w pamięci ferrytowej wyróżnić następujące części:

- a/ blok ferrytowy wraz ze związanymi z nim bezpośrednio układami elektronicznymi,
- b/ rejestr adresowy pamięci ferrytowej RA,
- c/ rejestr buforowej pamięci ferrytowej BF i układy logiczne bitu parzystości.

W bloku z rdzeniami ferrytowymi znajduje się 1024 /ósemkowo 1777/ komórek 16+1-bitowych. Na 17 bit dopisywana zostaje jedyńska, jeśli liczba jedynek pozostałej części komórki jest nieparzysta. Bit 17 jest więc bitem parzystości.

U w a g a:

Bit ten jest również testowany w czasie przepisywania informacji na bęben oraz w czasie pobierania jej z powrotem do pamięci ferrytowej.

Po czasie jednej chwili od wygenerowania impulsu "zwrot" przy jednoczesnym ustawieniu adresu żądanej komórki w rejestrze RA, pojawia się w rejestrze buforowym pamięci ferrytowej BF informacja zapisana w tej komórce.

W Kalkulatorze ODRA 1103 wewnętrzna pętla regeneracji informacji w pamięci ferrytowej nie jest wykorzystana.

Do pamięci wpisuje się treść rejestru BF w chwili następnej po odczycie.

Postępowanie takie związane jest z faktem, że ODRA 1103 jako maszyna znakowa, często używa do obliczeń jedynie części bitów wybranej komórki. Prowadzi to również do konieczności równoległego testowania starego i tworzenia nowego bitu parzystości. Jak już zaznaczono, w czasie wykonywania działań na liczbach dziesiętnych adresowane są nie całe komórki, lecz pojedyncze tetrady /ćwiartki komórek/. Wyboru odpowiednich tetrad komórki dokonuje się w układzie sterowania, a nie w bloku pamięci. Stąd też zagadnienie to będzie omówione w następnym punkcie.

1.2.4. Blok sterowania.

Blok sterowania steruje pracą Kalkulatora.

Wypracowuje on odpowiednie serie czasowe impulsów, pobiera rozkazy z pamięci ferrytowej i ewentualnie modyfikuje je, a następnie uruchamia układy wykonujące te rozkazy. Blok sterowania kieruje również kolejnością wykonywania rozkazów oraz umożliwia dostęp do pamięci ferrytowej w wypadku zgłoszenia się urządzenia kartowego.

W bloku sterowania można wyróżnić z funkcjonalnego punktu widzenia następujące układy:

- a/ układ sterowania,
- b/ rejestr rozkazów R wraz z układem deszyfrującym,
- c/ licznik rozkazów,

Ad a/. Najważniejszą częścią układu sterowania jest układ czasu. Dla zsynchronizowania pracy kalkulatora z obrotami bębna wytwarzane są w układzie czasu impulsy tak zwanych chwil maszynowych.

W układzie czasu powstają również impulsy /powtarzające się co 4 chwile/ taktujące pracę bloku arytmometru. Iloczynując chwile maszynowe z wspomnianymi wyżej impulsami taktującymi otrzymuje się serie czasowe tzw. kroki. Kroki są seriami wykonawczymi, które wraz z impulsami matrycowymi /otrzymanymi z układu deszyfrującego rozkazy/ sterują wykonaniem rozkazu.

Rozkaz do wykonania zostaje pobrany z pamięci ferrytowej. Przebieg pobrania rozkazu zależy od tego, czy adres zapisany w jego części adresowej jest adresem komórki technicznej /adres duży/, czy też adresem tetrady /adres mały/.

Adres rozkazu zwracającego się do całej komórki technicznej zostaje wprowadzony do rejestru RA bez zmian. Jeśli jednak kod operacji wskazuje na to, że część adresowa rozkazu jest adresem tetrady, to do rejestru RA wchodzi tylko osiem bardziej znaczących bitów części adresowej rozkazu. Na dwa wolne bity RA /najbardziej znaczące/ wchodzi informacja zapisana w rejestrze rozszerzającym B.

Uzyskano w ten sposób wydłużenie efektywnego adresu o dwa bity, co pozwala na adresowanie tetrad /pojedyn-

czych znaków 4 bitowych/.

Po ewentualnym zmodyfikowaniu w sumatorze, zostaje on wprowadzony do rejestru rozkazu R. Układ deszyfrujący rozkazy jest połączony równolegle z rejestrem rozkazu. Układ ten wypracowuje na podstawie informacji zapisanej w rejestrze R odpowiednie impulsy matrycowe.

Impulsy matrycowe zostają wraz z krokami skierowane do części wykonawczej układu sterowania, gdzie powodują powstanie wykonawczych serii impulsów, koniecznych do odpowiedniej realizacji rozkazu.

Po zakończeniu wykonania rozkazu układ sterujący wykonuje następujące czynności:

sprawdza, czy w czasie wykonywania rozkazu zgłosiło się urządzenie kartowe. Jeśli tak, to udostępnia pamięć ferrytową dla wczytania /ew. wyperforowania/ jednego wiersza karty. Jeśli nie, wówczas:

jeśli ostatni wykonany rozkaz był efektywnym rozkazem skoku, to wpisuje jego adres do licznika rozkazów LR i pobiera następny rozkaz wg tego adresu. Jeśli nie, to zwiększony o 1 licznik rozkazów wprowadza do rejestru adresowego pamięci ferrytowej i pobiera następny rozkaz.

Licznik rozkazów LR jest dziesięciobitowym rejestrem, w którym pamięta się adres rozkazu, który będzie wykonany jako następny.

1.2.5. Blok arytmometru.

Blok jest zasadniczą częścią Kalkulatora. W tym bloku wykonuje się operacje arytmetyczne i logiczne. W skład bloku arytmometru wchodzi następujące układy:

- a/ sumator,
- b/ dwa rejestry arytmetyczne A i W,
- c/ układ znaków i warunków,
- d/ pulpit operatora wraz z jego układami sterowania.

Ad a/. Sumator jest układem, w którym dokonuje się operacji dodawania i odejmowania na argumentach wykonywanej operacji. W sumatorze zachodzi również modyfikacja

rozkazu pobieranego do wykonania. W Kalkulatorze zastosowano sumator szeregowy o własności dokonywania operacji na liczbach binarnych i na liczbach zapisanych w kodzie dziesiętnym prostym.

Ad b/ Kalkulator wyposażony jest w dwa 32-bitowe rejestry arytmetyczne. W każdym z nich można przechowywać ośmiocyfrową liczbę dziesiętną lub 32-bitową liczbę binarną.

W czasie operacji na liczbach "długich" /do 16-tu cyfr/ Kalkulator traktuje rejestr W jako lewe wydłużenie akumulatora /rejestru A/.

Istnieje możliwość programowego przesuwania akumulatora w lewo i w prawo.

W czasie wykonywania podprogramów mnożenia i dzielenia w rejestrach A i W przechowuje się mnożnik i mnożną /ewentualnie dzielną/ oraz niektóre wyniki pośrednie. Po zakończeniu podprogramu w A i W zapisany jest iloczyn /ewentualnie iloraz i reszta/.

Ad c/ Układ znaków i warunków utworzony jest przez 6 przerzutników, mianowicie:

- 1/ przerzutnik S_A - znak akumulatora, pali się*, jeśli liczba w akumulatorze jest ujemna,
- 2/ przerzutnik S_W - znak rejestru W, zostaje zapalony, jeśli liczba w rejestrze W jest ujemna,
- 3/ przerzutnik S_n - pali się, jeśli liczba dziesiętna pobierana z pamięci jest ujemna,
- 4/ przerzutnik U - przerzutnik warunku "ujemny" zostaje zapalony, jeśli wynik operacji jest ujemny,
- 5/ przerzutnik Z - przerzutnik warunku "zero" zostaje zapalony, jeśli wynik operacji jest równy zero,
- 6/ przerzutnik N_d - przerzutnik warunku "nadmiar" zostaje zapalony, jeśli w wyniku operacji powstał nadmiar.

Przerzutnik S_n oraz warunki są gaszone*, a potem ponownie ustawiane przez wszystkie operacje z wyjątkiem:

*Zapalić przerzutnik = przerzucić go w stan jedynki logicznej,
gasić /lub wyzerować/ przerzutnik = przerzucić go w stan zera logicznego.

- 1/ operacji Wejścia - Wyjścia,
- 2/ operacji skoków,
- 3/ operacji modyfikacji,
- 4/ operacji ZLR /zapamiętanie licznika rozkazów/.

Sposób ustawiania warunków będzie omówiony dla każdej operacji osobno;

Ad d/ pulpit sterowniczy służy do ręcznego sterowania Kalkulatorem przez operatora /ew. konserwatora/ oraz do informowania obsługi o stanie maszyny.

Informację wprowadza się do maszyny za pomocą klawiszy umieszczonych na płycie pulpitów.

Zawartość rejestrów i pewne stany maszyny wyświetlane są za pomocą lampek sygnalizacyjnych.

Rysunek 7 przedstawia płytę pulpitu sterowniczego.

Na płycie tej znajdują się od lewej:

- osiem lampek /w dwóch kolumnach po 4/ sygnalizujących stan Kalkulatora,
- lampka PA - świeci, jeśli Kalkulator wykonuje program na "pracy automatycznej",
- lampka Sn - świeci, jeśli pali się przerzutnik Sn,
- lampka MD - świeci, jeśli następny wykonany przez Kalkulator rozkaz będzie zmodyfikowany,
- lampka Nd - świeci, jeśli pali się przerzutnik "nadmiar",
- lampka WW - świeci, jeśli zajęte jest urządzenie kartowe,
- lampka Z - świeci, jeśli pali się warunek "Z",
- lampka P - świeci, jeśli pali się przerzutnik bitu parzystości. Jeśli Kalkulator przerwie obliczenia i pali się lampka P, to oznacza to, że nastąpiło przekłamanie informacji w pamięci /ferrytowej lub bębnowej/,
- lampka U - świeci, jeśli zapalony jest przerzutnik U,
- 32 lampki /osiem kolumn po 4 lampki sygnalizujące

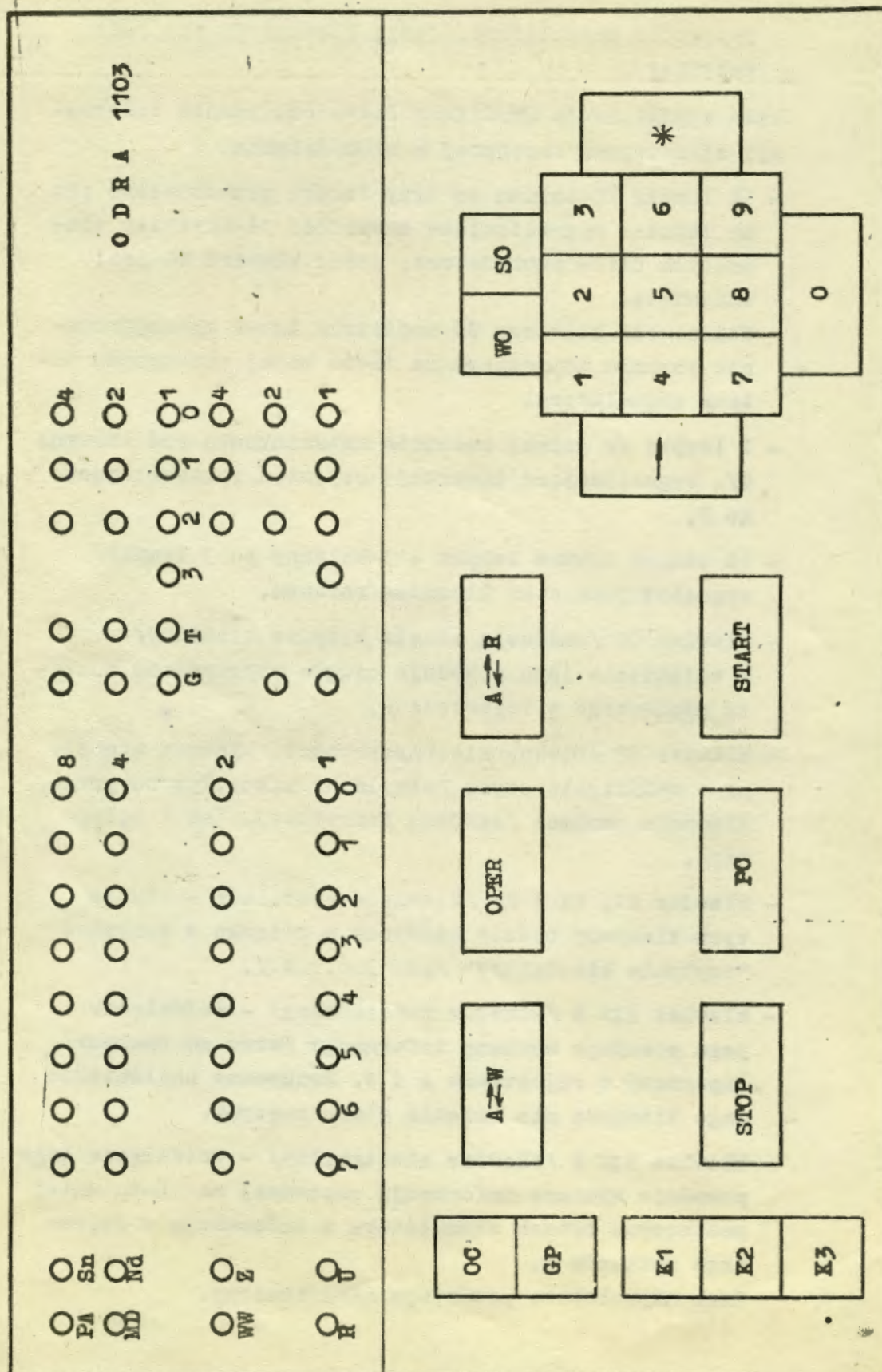
zawartość akumulatora, jeśli klawisz WO nie jest wciśnięty,

Taka sygnalizacja umożliwia łatwe odczytanie informacji dziesiętnej zapisanej w akumulatorze.

- 16 lampek /5 kolumn po trzy lampki przedzielone jedną lampką/ sygnalizujące zawartość 16-tu mniej znaczących bitów akumulatora, jeśli klawisz WO jest wciśnięty.

Wciśnięcie klawisza WO umożliwia łatwe zdeszyfrowanie rozkazu zapisanego na 16-tu mniej znaczących bitach akumulatora.

- 2 lampki /w jednej kolumnie umieszczonej pod kolumną G/, sygnalizujące zawartość rejestru rozszerzającego B,
 - 10 lampek /jedna lampka + 3 kolumny po 3 lampki/ sygnalizujące stan licznika rozkazu,
 - klawisz OC /operacja ciągła klawisz stabilny/ -
 - wciśnięcie jego powoduje ciągłe wykonywanie rozkazu zapisanego w rejestrze R,
 - klawisz BP /blokowanie parzystości, klawisz stabilny - wciśnięcie czyni Kalkulator nieczułym na przekłamania pamięci /zarówno ferrytowej, jak i bębnowej/,
 - klawisz K1, K2 i K3 /klawisze stabilne/ - użycie tych klawiszy będzie omówione w związku z rozkazem "czytanie klawiatury" /pkt 3.6.3.4./,
 - klawisz A $\rightleftharpoons$ W /klawisz niestabilny/ - wciśnięcie jego powoduje wymianę informacji /wraz ze znakami/ zapisanej w rejestrach A i W. Dwukrotne naciśnięcie tego klawisza nie zmienia stanu maszyny.
 - klawisz A $\rightleftharpoons$ R /klawisz niestabilny/ - wciśnięcie jego powoduje wymianę informacji zapisanej na 16-tu mniej znaczących bitach akumulatora z informacją w rejestrze rozkazów R.
- Znak akumulatora pozostaje niezmienny.



Pulpit sterowniocy
Rys.7.

- klawisz OPER /klawisz niestabilny/ - naciśnięcie powoduje jednokrotne wykonanie rozkazu umieszczonego w rejestrze R. Stan licznika rozkazów LR nie ulega zmianie. Jeśli w chwili naciśnięcia klawisza OPER wciśnięty jest również klawisz stabilny OC, to rozkaz zapisany w rejestrze R wykonywany będzie cyklicznie do chwili zwolnienia klawisza OC.

UWAGA: Jeśli przy pierwszym wykonaniu rozkaz w rejestrze R zostanie zmodyfikowany, to cyklicznie wykonuje się nie rozkaz "pierwotny", lecz zmodyfikowany.

- klawisz START /klawisz niestabilny/ - wciśnięcie jego powoduje wykonanie rozkazu, którego adres wskazany jest w liczniku rozkazów LR oraz dodanie jedynki do licznika rozkazów,
- klawisz PC /klawisz stabilny/ - jeśli w chwili naciśnięcia klawisza START klawisz PC nie jest wciśnięty, to Kalkulator pracuje krokowo. Jeśli w chwili naciśnięcia klawisza START klawisz PC jest wciśnięty, to Kalkulator wykonuje program automatycznie. Zapala się lampka sygnalizacyjna PA.
- klawisz STOP /klawisz niestabilny/ - wciśnięcie jego powoduje przerwanie pracy automatycznej Kalkulatora, po zakończeniu aktualnie wykonywanego rozkazu. Ten sam efekt można uzyskać zwalniając klawisz PC.

UWAGA: Jeśli w chwili przerwania pracy automatycznej Kalkulator nie zakończył rozpoczętej uprzednio operacji kartowej, to operacja ta zostanie zakończona /karta wczytana, ewentualnie wyperforowana/, mimo że wykonywanie innych operacji zostanie wstrzymane wcześniej.

- klawisz SO /klawisz stabilny/ - wciśnięcie jego powoduje wyświetlenie informacji zapisanej w 16-tu mniej znaczących bitach akumulatora /cztery prawe kolumny bloku lampek wyświetlających akumulator do odczytania jako liczbę dziesiętną/ w formie rozkazu /zapala się blok lampek w prawym górnym rogu płytki z lampkami/,

UWAGA: Wciśnięcie lub zwolnienie klawisza SO w czasie pracy Kalkulatora nie wpływa na sposób dokonywania obliczeń.

- klawisze od 0 do 9 /klawisze niestabilne/ - naciśnięcie ich powoduje:
 - a/ przesunięcie akumulatora w lewo,
 - b/ wpisanie na 4 najmniej znaczące bity akumulatora kodu dziesiętnego prostego odpowiedniej cyfry,
- klawisz WO /klawisz stabilny/ - jeśli w chwili naciśnięcia któregośkolwiek z klawiszy oznaczonych od 0 do 9 klawisz WO jest wciśnięty, to akumulator przesuwa się w lewo o 3 bity. Jeśli klawisz WO nie jest wciśnięty, to akumulator przesuwa się o 4 bity w lewo,
- klawisz → /klawisz niestabilny/ - wciśnięcie jego powoduje przesunięcie /bez zmiany znaku/ akumulatora o 2 bity w prawo,
- klawisz * /klawisz niestabilny/ - wciśnięcie jego powoduje wyzerowanie akumulatora.

1.2.6. Ręczne sterowanie Kalkulatorem.

Za pomocą pulpitu sterowniczego operator może wykonać pewne czynności wstępne celem uruchomienia programu. Może on także wpływać na pracę Kalkulatora w czasie wykonywania programu. Z pulpitu można wykonać następujące proste czynności:

- a/ Wpisanie do akumulatora liczby dziesiętnej.
Przy wciśniętym klawiszu WO naciskać odpowiednie klawisze /od 0 do 9/ począwszy od najbardziej znaczącej cyfry.
- b/ Wpisywanie do akumulatora liczby ósemkowej.
Przy wciśniętym klawiszu WO naciskać odpowiednie klawisze /od 0 do 7/ począwszy od najbardziej znaczącej cyfry.
- c/ Wpisywanie do akumulatora rozkazu.
Przy wciśniętym klawiszu WO wpisać kod operacji jak liczbę ósemkową, następnie nacisnąć 7 lub 0 zależnie od tego,

czy najbardziej znaczący bit adresu jest równy 0, czy 1.
Nacisnąć klawisz → kl. i wpisać dalsze /mniej znaczące/
cyfry adresu jak cyfry liczby ósemkowej.

UWAGA: W czasie wpisywania liczb ósemkowych i rozkazów
wciśnięcie klawisza SO /sygnalizacja ósemkowa/ ułat-
wia obserwację wpisywania.

- Wpisywanie informacji do rejestru W.
Po wpisaniu tej informacji do akumulatora nacisnąć kla-
wisz A ↔ W.
- Wpisywanie informacji do rejestru rozkazów.
Po wpisaniu tej informacji do akumulatora nacisnąć kla-
wisz A ↔ R.
- Wykonanie dowolnego rozkazu.
Po przesłaniu tego rozkazu do rejestru R nacisnąć kla-
wisz OPER.

Przykład: Ładowanie licznika rozkazów /rejestru LR/.

Wpisać żądany adres do akumulatora i przesłać go
do rejestru rozkazów, a następnie nacisnąć kla-
wisz OPER. Wykona się operacja skoku - adres zost-
anie przesłany do LR.

- Start programu od komórki o zadanym adresie.
Wprowadzić adres do rejestru LR i nacisnąć klawisz START.
Jeśli klawisz PC nie jest wciśnięty, to program wykonywa-
ny będzie krokowo. Jeśli klawisz PC jest wciśnięty, to
program wykonywany będzie automatycznie.
- Przerwanie pracy Kalkulatora.
Pracę automatyczną Kalkulatora można przerwać naciskając
klawisz STOP, albo zwalniając klawisz PC.

UWAGA: Przewidziano możliwości ingerowania z pulpitu w spo-
sób wykonywania programu /patrz opis działania roz-
kazu "CzK" - czytanie klawiatury/.

- Włączenie Kalkulatora.
Kalkulator włącza się naciskając klawisz START B. umiesz-
czony na płycie czołowej pokrywy jednostki centralnej.
Po upływie około 5 minut należy włączyć napięcia stabili-

zowane przez naciśnięcie klawisza Z.

- Wyłączenie Kalkulatora.

Kalkulator wyłącza się przez naciśnięcie klawisza STOP M, a później klawisza STQP B. Obydwa te klawisze umieszczone są na płycie czołowej pokrywy jednostki centralnej.

2. ARYTMETYKA I ALGORYTMY ZASTOSOWANE W KALKULATORZE.

2.1. Arytmetyka dla liczb dziesiętnych.

2.1.1. Operacja dodawania.

Zasadniczą arytmetyką używaną głównie do opracowania danych jest arytmetyka dziesiętna moduł - znak.

Jednym rozkazem można wykonać jedynie operację dodawania.

Możliwe jest oczywiście dodawanie liczb o równych lub o różnych znakach. Operację odejmowania należy zrealizować za pomocą dwóch rozkazów - rozkazu zmiany znaku na przeciwny i rozkazu dodawania.

Znak liczby "krótkiej" /co najwyżej 8-cyfrowej, zapisanej w akumulatorze/ pamiętany jest przez przerzutnik S_a^- /znak akumulatora/. Znak liczby "długiej" /co najwyżej 16-cyfrowej, zapisanej w połączonych rejestrach W - A/ pamiętany jest /podobnie jak znak liczby "krótkiej" zapisanej w samym tylko rejestrze W/ przez przerzutnik S_w /znak rejestru W/. Znak liczby dziesiętnej zapisanej w pamięci ferrytowej pamiętany jest na ogół w najbardziej znaczącej tetradzie tej liczby /patrz 1.1.1.2./.

Algorytm dodawania /zarówno dla liczb "krótkich", jak i dla "długich"/ realizowany jest w dwóch krokach.

W kroku pierwszym dokonuje się porównania modułów i znaków argumentów. W ten sposób ustala się znak wyniku oraz zapamiętuje fakt, czy moduły mają być dodane, czy odjęte. Znak wyniku jest równy znakowi składnika o większym module. Sumator nastawia się na odejmowanie, jeśli znaki składników są różne.

UWAGA: Jeśli składniki mają równe moduły a różne znaki, to wynik jest dodatni i równy 0.

2.1.2. Operacja mnożenia.

Operacja mnożenia zrealizowana jest w Kalkulatorze programowo.

Treść operacji mnożenia jest następująca:

mnożna /umieszczona w rejestrze W/ jest mnożona przez mnożnik /umieszczony w akumulatorze/.

Po wykonaniu podprogramu iloczyn znajduje się w rejestrach W i A, przy czym bardziej znaczące cyfry zapamiętane są w rejestrze W.

Mnożna może mieć co najwyżej 7 cyfr, a mnożnik 8 cyfr.

Algorytm mnożenia realizowany jest w dwóch krokach.

W pierwszym kroku tworzy się "tabliczkę mnożenia" tzn. do komórek 2-ej i 3-ej zostaje wysłana mnożna, do komórek 4-ej i 5-ej jej podwójna wartość, do następnych dwóch komórek jej potrojona wartość itd. do 18-ej i 19-ej komórki zostaje wysłana 9-krotna wartość mnożnej.

Drugi krok algorytmu jest właściwym krokiem wykonawczym.

Mnożnik zostaje pobrany do akumulatora. Jego cyfry /kolejno, począwszy od najmniej znaczącej/ sterują dodawaniem do rejestru W odpowiedniej wielokrotności mnożnej. Po każdym takim dodaniu rejestry W i A /zawierające sumę częściową i "niewykorzystane" cyfry mnożnika/ przesuwają się o 4 bity w prawo.

Mnożenie jest zakończone, gdy ostatnia /najbardziej znacząca/ cyfra mnożnika zostanie "wysunięta" z akumulatora. Rejestry A i W są całe zajęte przez ostatnią sumę częściową - czyli przez iloczyn. Znak iloczynu pamiętany jest przy pomocy przełącznika Sw. Jeśli iloczyn jest ujemny, zapala się warunek U.

2.1.3. Operacja dzielenia.

Operacja dzielenia /podobnie jak mnożenia/ jest w Kalkulatorze zrealizowana programowo. Treść operacji dzielenia jest następująca:

Dzielną /powstała np. jako wynik mnożenia/ znajduje się w rejestrach W i A /cyfry bardziej znaczące w W/, dzielnik zapamiętany jest w komórkach 0 i 1 pamięci ferrytowej, /cyfry bardziej znaczące w komórce 1/.

Dzielnik może być co najwyżej liczbą 7-cyfrową.

Po wykonaniu podprogramu dzielenia iloraz /co najwyżej ośmio-cyfrowy/ znajduje się w akumulatorze.

Algorytm dzielenia realizowany jest poprzez kolejne operacje odejmowania.

Jednym rozkazem programowym wykonuje się obliczenia jednej cyfry ilorazu. Rozkaz taki wykonywany jest następującym sposobem. W czasie pierwszego kroku wykonawczego zachodzi porównanie między aktualną zawartością rejestru W i dzielnikiem. Jeśli liczba zapisana w rejestrze W jest większa od dzielnika, to w drugim kroku wykonawczym:

- a/ od liczby zapisanej w W odejmuje się dzielnik, a różnicę wysyła się ponownie do W,
- b/ licznik, w którym pamiętana jest aktualnie obliczona cyfra ilorazu, zwiększany jest o jeden,
- c/ obliczona w punkcie a/ różnica jest porównywana z dzielnikiem i jeśli różnica jest większa od dzielnika, to punkty a/ do c/ zostają powtórzone.

Operacja kończy się, jeśli aktualna zawartość rejestru W /obliczona różnica/ jest mniejsza od dzielnika.

Następny kolejny rozkaz podprogramu powoduje przesunięcie "długiego" rejestru W, A o 4 bity w lewo.

Na 4 najmniej znaczące bity akumulatora zostaje wprowadzona aktualnie obliczona cyfra ilorazu.

Podprogram kończy się, gdy cały akumulator wypełniony jest cyframi ilorazu.

Jak widać z algorytmu, czas wykonania podprogramu dzielenia zależy od ilości odejmowań, tzn. od sumy cyfr ilorazu. Jeśli wszystkie cyfry ilorazu są dziewiątkami, to czas dzielenia wynosi około 20 msek. Dla ilorazu złożonego z samych zer i jedynek, dzielenie dokonuje się w czasie około 10 msek.

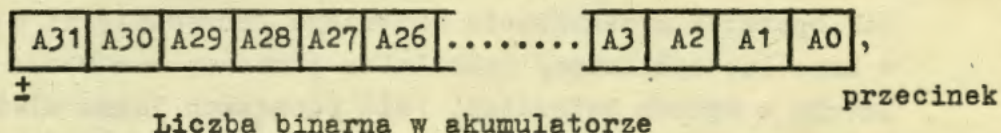
2.2. Arytmetyka dla operacji binarnych.

W dalszej części opisu operacjami binarnymi nazywane będą wszystkie te operacje arytmetyczne, które dają w wyniku liczbę dwójkową. Część adresowa tych rozkazów ma sens parametru lub

adresu komórki technicznej /nie tetrady/.

Dla wszystkich operacji binarnych zastosowano arytmetykę uzupełnieniową.

- 2.2.1. Operacje binarne, których wynik pozostaje w akumulatorze. Do przedstawienia wyniku operacji binarnej w akumulatorze użyto wszystkich 32-u pozycji akumulatora.



Rys. 8

Znak liczby zapisany jest na 31-ej pozycji akumulatora /0 plus, 1 minus/. Na pozostałych pozycjach zapisane są bity znaczące liczby, przy czym najmniej znaczący bit znajduje się na pozycji A0.

Kalkulator traktuje liczby binarne jako liczby całkowite, /przecinek znajduje się po najmniej znaczącym bicie/. Możliwa jest oczywiście konstrukcja programów, w których pozycja jednostek liczby binarnej znajduje się w dowolnym innym miejscu.

Jeśli przecinek znajduje się w miejscu pokazanym na rys. 8, to w akumulatorze można zapisać liczby binarne z przedziału:

$$-2^{32} \leq a \leq 2^{32} - 1$$

Jeśli w wyniku operacji binarnej powstanie liczba, która nie mieści się w tym przedziale, to wygenerowany zostaje warunek "nadmiar".

- 2.2.2. Operacja binarna, której wynik zostaje przesłany do komórki pamięci ferrytowej.

Aby ułatwić organizację cyklu przewidziano w kalkulatorze operację, która powoduje zmniejszenie dowolnej komórki pamięci ferrytowej o 1.

Binarna zawartość tej komórki jest zawsze traktowana jak liczba dodatnia /bez względu na to, czy 16-ty bit komórki jest równy 1 czy 0/. Operacja powoduje więc wygenerowanie

jedynie warunku "zero" /jeśli w wyniku odejmowania otrzymane 0/ lub "dodatni" /w każdym innym przypadku/.

2.2.3. Algorytm operacji przesunięć krótkich.

Operacje przesunięć krótkich powodują przesunięcie akumulatora o taką ilość pozycji, jaką podano na 5-ciu najmniej znaczących bitach części adresowej rozkazu. Bez względu na kod operacji przesunięcia krótkiego /przesunięcie w prawo, w lewo lub cykliczne/ Kalkulator przesuwa zawartość akumulatora w sposób naturalny, /dla binarnych liczb ujemnych nie zachodzi powielanie bitu znakowego przy przesunięciu w prawo/.

Operacje przesunięć krótkich nie powodują zmiany znaku akumulatora.

Przyjęcie takiego algorytmu przesunięć krótkich ułatwia mnożenie i dzielenie liczb dziesiętnych przez 10 /co jest równoważne przesuwananiu przecinka/.

Algorytm ten pozwala również na łatwe mnożenie i dzielenie liczb binarnych /dodatnich/ przez 2.

3. LISTA ROZKAZÓW KALKULATORA.

3.1. Oznaczenia i skróty.

Przy omawianiu listy rozkazów będą używane następujące oznaczenia i skróty:

- A - rejestr akumulatora
- W - rejestr W /lewe wydłużenie akumulatora/
- WA - "długi" rejestr otrzymany przez połączenie rejestrów W i A
- LR - licznik rozkazów
- B - dwubitowy rejestr, w którym pamiętany jest numer strefy pamięci ferrytowej /dla operacji dziesiętnych/
- N - adres komórki technicznej pamięci ferrytowej lub parametr zapisany w części adresowej rozkazu.

Małe litery oznaczają zawartości odpowiednich rejestrów np.:

- a - zawartość akumulatora, wa - zawartość rejestru "długiego", n - zawartość ciągu tetrad w pamięci /niekoniecznie komórek technicznych/.

Zawartość rejestru może być również oznaczona przez ujęcie w nawias nazwy rejestru np.:

a = /A/; wa = /WA/ itp.

Litery a', w' itd. oznaczają zawartość odpowiednich rejestrów po operacji.

Znak "==" oznacza operację podstawienia; wielkość po lewej stronie znaku "==" czyni się równą wielkości po prawej stronie tego znaku.

Rejestry warunków

- Z - rejestr warunku "zero"
- U - rejestr warunku "ujemny"
- Nd - rejestr warunku "nadmiar"

Skróty dodatkowe

- t - zawartość rzędu taśmy perforowanej
- d - kod binarny znaku dalekopisowego.

UWAGA: Jeśli nie będzie wyraźnie powiedziane, że dany warunek nie zeruje się przed wykonaniem omawianego rozkazu, to należy przyjąć, że warunek zeruje się i że rozkaz ten wytwarza własny warunek.

3.2. Zasady przyporządkowania kodów.

Struktura słowa rozkazowego Kalkulatora ODRA 1103 opisana była w punkcie 1.1.3. Teraz podane będą bliżej zasady przyporządkowania bitom słowa rozkazowego odpowiednich kodów operacji.

Zarówno adresy pamięci ferrytowej, jak i całe rozkazy najwygodniej przedstawić w postaci liczb ósemkowych

G				T		N									
0	1	0	1	0	0	1	0	1	1	1	0	0	1	0	1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
						g		t				p			
2			4		1		3		4				5		

Rys. 9

Rozkaz przedstawiony powyżej binarnie można zapisać w postaci ósemkowej jako 241345.

3.3. Tabele rozkazów.

Rozkazy Kalkulatora ODRA 1103 zebrane są, w zależności od wykonywanych funkcji, w trzy tabele.

W tabeli 1 podane są rozkazy, dla których pierwsza cyfra ósemkowa kodu operacji G równa jest zero lub jeden /z wyjątkiem rozkazów, których kod operacji zaczyna się od cyfr 07/. Tabele pierwszą tworzą następujące grupy rozkazów:

- a/ Grupa skoków,
- b/ Grupa operacji na parametrze N,
- c/ Grupa operacji na binarnej zawartości komórek technicznych,
- d/ Grupa utworzona z operacji pamiętania licznika rozkazów i operacji modyfikacji.

W tabeli 2 podane są rozkazy, dla których pierwsza cyfra kodu operacji $2 \leq G \leq 7$.

Są to rozkazy działające na liczbach dziesiętnych o zmiennej ilości cyfr.

Tabela 3 utworzona jest z rozkazów, których kod operacji rozpoczyna się od cyfr ósemkowych 07.

W tabeli tej zebrane są rozkazy należące do 4 grup:

- a/ grupa przesunięć,
- b/ grupa operacji pomocniczych dla mnożenia i dzielenia,
- c/ grupa operacji specjalnych,
- d/ grupa operacji wejścia - wyjścia.

Wszystkie rozkazy tabeli 3 są rozkazami bezadresowymi. Bardziej znaczące bity części adresowej mają sens kodu operacji, bity mniej znaczące zawierają parametr /np. ilość przesunięć/ bliżej precyzujący wykonanie operacji.

Rozkazy w tabelach oznaczone są 3-literowymi skrótami. Skróty te dobrano tak, by oddawały w pewnym sensie treść operacji.

- Np. SKZ - SKok przy warunku Zero
- PDL - Pobierz Długą Liczbę
- CZK - Czytaj Klawiaturę
- ODN - ODejmij parametr N

Te same trzyliterowe skróty użyto w języku PAS Kalkulatora. W dalszej części opisu omówione są szczegółowo wszystkie rozkazy Kalkulatora.

3.4. Rozkazy ujęte w tabeli pierwszej.

3.4.1. Rozkazy skoków.

Efektywne wykonanie rozkazu skokowego sprowadza się do przesłania informacji zapisanej w części adresowej rozkazu do licznika rozkazów.

Skoki warunkowe przy niespełnionym warunku skoku /skok nieefektywny/ wykonywane są jak rozkazy "nic nie rób". Wykonanie rozkazu skokowego nie zmienia warunków, ani znaków rejestrów A i W.

Czas wykonania operacji skoku /niezależnie od tego, czy skok został efektywnie wykonany, czy nie/ wynosi 36 chwil, co odpowiada 6038 operacjom/sek.

3.4.1.1. STop i Skocz STS.

Zapis: OON

funkcja: /LR/ := N i STOP

treść: wpisanie części adresowej rozkazu do licznika rozkazów i zatrzymanie programu.

Po naciśnięciu klawisza START następny rozkaz zostanie pobrany z komórki N.

3.4.1.2. SKok przy Ujemnej wartości SKU.

Zapis: 01N

funkcja: jeśli $U = 1$ to /LR/ :=N.

treść: jeśli zawartość rejestru U jest 1, to przejdź do wykonania rozkazu z komórki N.

W przeciwnym razie wykonaj rozkaz następny.

3.4.1.3. SKok przy Zerze SKZ.

Zapis: 02N

funkcja: jeśli $Z = 1$ to /LR/ :=N.

treść: jeśli zawartość rejestru Z jest 1, to przejdź do wykonania rozkazu z komórki N.

W przeciwnym razie wykonaj rozkaz następny.

3.4.1.4. SKok przy Dodatniej wartości SKD.

Zapis: 03N

funkcja: jeśli $U = 00$ i $Z = 0$, to $/LR/ := N$

treść: jeśli zawartość rejestru U jest zero i zawartość rejestru Z jest zero, to przejdź do wykonania rozkazu z komórki N.

W przeciwnym razie wykonaj rozkaz następny.

3.4.1.5. SKok przy Nadmiarze SKN.

Zapis: 04N

funkcja: jeśli $N_d = 1$, to $/LR/ := N$

treść: jeśli zawartość rejestru N_d jest 1, to przejdź do wykonania rozkazu z komórki N.

W przeciwnym razie wykonaj rozkaz następny.

3.4.1.6. SKok Bezwarunkowy SKB.

Zapis: 05N

funkcja: $/LR/ := N$

treść: przejdź do wykonania rozkazu z komórki N.

3.4.1.7. SKok przy zajętym urządzeniu kartowym Wejścia - Wyjścia SKW.

Zapis: 06N

funkcja: jeśli zajęte jest urządzenie kartowe, to $/LR/ := N$

treść: jeśli urządzenie kartowe Wejścia - Wyjścia rozpoczęło, lecz jeszcze nie zakończyło czytania /ewentualnie perforacji/ karty, to przejdź do wykonania rozkazu z komórki N.

W przeciwnym razie wykonaj rozkaz następny.

UWAGA: Treść tego rozkazu wynika z przyjętego sposobu współpracy Kalkulatora i kartowego urządzenia zewnętrznego. Współpraca ta opisana jest bliżej w punkcie 4.

Duża stosunkowo ilość rozkazów skokowych zrealizowanych w Kalkulatorze znacznie ułatwia programiście wybór różnych dróg wykonania programu. Na odgańczenie programu prawie zawsze wystarcza jeden tylko rozkaz.

3.4.2. Operacje na parametrze N.

Operacje na parametrze nie wymagają /w czasie swego wykonania/ kontaktu Kalkulatora z pamięcią operacyjną. Jednym argumentem operacji jest akumulator, a drugim parametr o długości 10 bitów zapisanych w części adresowej rozkazu. Wynik operacji zostaje przesłany do akumulatora.

Operacje na parametrze gaszą stare warunki i tworzą nowe w następujący sposób:

Warunek U zapala się, jeśli w wyniku operacji zapala się najbardziej znaczący bit akumulatora A31.

Warunek Z zapala się, jeśli w wyniku operacji zostaną wprowadzone do akumulatora same zera.

Warunek Nd zapala się, jeśli w czasie trwania operacji wystąpi różnica symetryczna przeniesień z bitu 30 na bit 31 i z bitu 31 na następny /nie istniejący już w Kalkulatorze/ bit.

Operacje na parametrze zerują przerzutnik S_a , akumulator staje się dodatni "dziesiętnie". Znak rejestru W nie ulega zmianie.

Czas wykonania wszystkich operacji na parametrze wynosi 54 chwile, co odpowiada 4025 operacjom na sek.

3.4.2.1. Odjęcie parametru N od akumulatora ODN.

Zapis: 10N

funkcja: $a' := a - N$

treść: odjęcie binarne od akumulatora informacji zapisanej na 10-ciu bitach części adresowej rozkazu i umieszczenie wyniku w akumulatorze.

3.4.2.2. Pobranie parametru N do akumulatora PON.

Zapis: 11N

funkcja: $a' := N$

treść: pobranie do akumulatora informacji zapisanej na 10-ciu bitach części adresowej rozkazu.

3.4.2.3. Dodanie parametru N do akumulatora DON.

Zapis: 12N

funkcja: $a' := a + N$

treść: dodanie binarne do akumulatora informacji zapisanej na 10-ciu bitach części adresowej rozkazu i umieszczenie wyniku w akumulatorze.

3.4.3. Operacje na binarnej zawartości komórek technicznych.

Argumentem operacji jest 16-bitowa zawartość komórki technicznej. Adres komórki podawany jest w części adresowej rozkazu. Operacje tej grupy niszczą stare warunki i ustawiają nowe:

Warunek U - jeśli po operacji pali się najbardziej znaczący bit akumulatora A31.

Warunek Z - jeśli w wyniku operacji otrzymano zero.

Warunek Nd - jeśli powstanie różnica symetryczna dwóch najbardziej znaczących przeniesień. Warunek ten może się zapalić jedynie po operacji DOT.

3.4.3.1. Odjęcie Jedynki w komórce o adresie N ODJ.

Zapis: 13N

funkcja: $/N/ := /N/ - 1$

treść: binarna zawartość komórki o adresie N zostaje zmniejszona o 1. Operacja ta powoduje ustawienie jedynie warunku Z. Warunki U i Nd zostają wyzerowane. Zawartość komórki jest zawsze traktowana przez Kalkulator jak liczba dodatnia. Czas wykonania operacji-54 chwile.

3.4.3.2. Pamiętanie Akumulatora w komórce Technicznej PAT.

Zapis: 15N

funkcja: $n' = a$

treść: 16 najmniej znaczących bitów akumulatora zostaje zapamiętana w komórce o adresie N.

Zawartość akumulatora nie zmienia się. Nadmiar nie może się zapalić. Czas wykonania operacji wynosi 72 chwile, co odpowiada 3019 operacjom na sek.

3.4.3.3. DODanie do akumulatora zawartości komórki Technicznej DOT

Zapis: 16N

funkcja: $a' := a + N$

treść: do akumulatora zostaje dodana binarnie zawartość komórki, której adres podany jest w części adresowej rozkazu.

Czas wykonania operacji wynosi 54 chwile, co odpowiada 4025 operacjom na sek.

3.4.4. Operacje ułatwiające organizowanie podprogramów.

Operacje te nie zmieniają warunków. Czas ich wykonania wynosi 36 chwile, co odpowiada 6038 operacjom/sek.

3.4.4.1. Zapamiętanie Licznika Rozkazów ZLR.

Zapis: 14N

funkcja: $N' := LR$

treść: zawartość licznika rozkazów zostaje zapamiętana w komórce o adresie N.

UWAGA: Jeśli rozkaz ZLR pobrany został do wykonania z komórki o adresie M, to spowoduje on zapisanie w komórce o adresie N liczby M+1 /binarnie/.

Rozkaz ZLR nie powoduje zmiany zawartości rejestrów arytmetycznych Kalkulatora.

3.4.4.2. MODyfikacja MOD.

Zapis: 17N

funkcja: modyfikacja następnego rozkazu,

treść: przed wykonaniem następnego rozkazu do jego części adresowej zostaje dodana zawartość 10-ciu najmniej znaczących bitów komórki N.

UWAGA: Ewentualne przeniesienie na część operacyjną GT jest blokowane. Jeśli następny rozkaz jest rozkazem z tabeli drugiej, to w modyfikacji bierze udział 12 najmniej znaczących bitów komórki N. Dwa najbardziej znaczące /biorące udział w modyfikacji/ bity komórki N zostają wprowadzone na odpowiednie pozycje rejestru RA po dodaniu do nich ewentualnego prze-

niesienia 1 bitów rejestru B.
/Patrz: opis rozkazów tabeli II/.

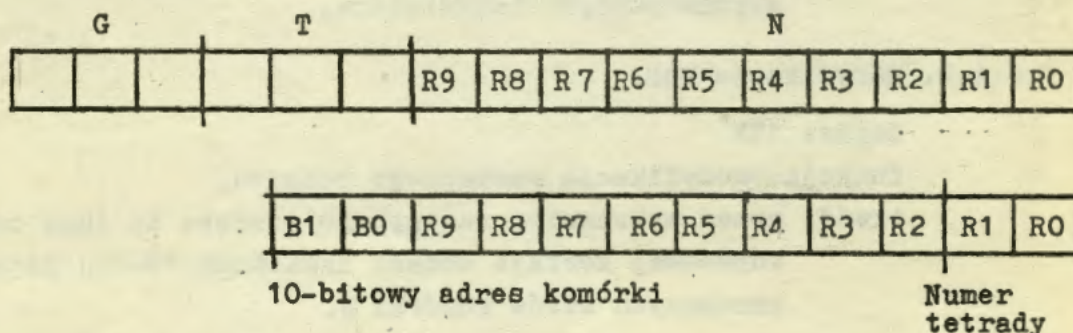
3.5. Rozkazy ujęte w tabeli drugiej.

Rozkazy ujęte w tabeli drugiej są rozkazami realizowanymi w arytmetyce dziesiętnej moduł - znak.

W tym przypadku informacja zapisana w części adresowej ma nieco inny sens niż dla rozkazów z tabeli pierwszej. Argumentem są cyfry dziesiętne /ciągi kolejnych znaków 4-bitowych - tetrad/. W części adresowej rozkazu podaje się adres tetrad, w której zapisana jest najmniej znacząca cyfra liczby /ewentualnie, do której ma być odesłana najmniej znacząca cyfra akumulatora dla operacji PAL/.

Adres tetrad jest liczbą 12-bitową - 10 bitów adresu komórki i 2 bity najmniej znaczące podające numer tetrad w komórce technicznej. W części adresowej rozkazu zapisanych jest 10 mniej znaczących bitów tej liczby tzn. 8 bitów adresu komórki i 2 bity numeru tetrad.

Dwa najbardziej znaczące bity adresu komórki tworzone są przez bity rejestru B lub /w wypadku modyfikacji/ przez dodanie do bitów B0 i B1 bitów jedenastego i dwunastego komórki modyfikującej z uwzględnieniem ewentualnego przeniesienia z pozycji poprzedniej.



R0 R9 - bity części adresowej rozkazu
B0, B1 - bity rejestru wydłużającego.

Adresowanie znaków 4-bitowych /tetrad/.

Rys. 10

Dla wszystkich rozkazów z tabeli drugiej Kalkulator sprawdza czy dwa bardziej znaczące bity - tetrady /pobieranej z pamięci/ są równe 1 /kombinacja niedziesiętna/. Taki układ bitów odpowiada naddziurkom lub brakowi dziurki na karcie oraz znakowi "-", którego kod jest identyczny z kodem pierwszej naddziurki. Tetrada, w której zapamiętany jest którykolwiek z powyższych kodów jest zerowana przed wprowadzeniem do bloku arytmometru. Za pomocą przerzutnika S_n /wyświetlonego na pulpicie/ pamiętany jest jednak znak liczby pobieranej z pamięci. W zależności od tego, czy argumentem operacji jest sam akumulator, czy też "długi" rejestr WA, operacje dziesiętne dzieli się na "krótkie" i "długie".

Warunki dla operacji obu rodzajów generowane są podobnie, mianowicie:

Warunek U - jeśli wynik operacji jest ujemny, to znaczy, gdy argument o większym module jest ujemny.

Warunek Z - jeśli wynik operacji jest równy zero.

Warunek Nd- jeśli wystąpi przeniesienie z najbardziej znaczącej tetrady /na już nie istniejącą/.

Ilość cyfr argumentu określona jest przez część T kodu operacji.

3.5.1. Operacje dziesiętne "krótkie".

Długość argumentu tj. ilość cyfr argumentu operacji dziesiętnych krótkich mieści się w granicach od 1 do 8 cyfr. Wynosi ona $T+1$, gdzie T jest cyfrą ósemkową zapisaną w części T kodu operacji.

Jeśli w wyniku operacji krótkiej zapali się warunek U, to zostanie zapalony również przerzutnik znaku akumulatora S_a /z wyjątkiem operacji porównania/.

Operacje dziesiętne krótkie nie zmieniają stanu przerzutnika znaku S_w , nie zmieniają również zawartości rejestru W.

Czas wykonania operacji nie zależy od ilości cyfr argumentu.

3.5.1.1. Pobierz Liczbę POL.

Zapis: 2TN

funkcja: $a' := n$

jedna tetrada uzerowana jest na znak liczby

zawartość a

treść: pobranie do akumulatora T+1 tetrad /począwszy od tetrad o adresie $/B/.2^{10}+N/$.

Czas wykonania operacji wynosi 72 chwile, odpowiada to około 3019 operacjom na sek.

3.5.1.2. Dodaj liczbę DOL.

Zapis: 3TN

funkcja: $a' := a + n$

treść: dodatnie dziesiętnie do akumulatora T+1 tetrad począwszy od tetrad o adresie $/B/.2^{10}+N$.

Czas wykonania operacji wynosi 90 chwil, co odpowiada około 2415 operacjom na sek.

3.5.1.3. Pamiętaj Liczbę PAL.

Zapis: 4TN

funkcja: $n' := a$

treść: pamiętanie T+1 tetrad akumulatora w pamięci, począwszy od tetrad o adresie $/B/.2^{10}+N$.

Zawartość akumulatora zostanie zapamiętana również, jeśli tetrad nie są dziesiętne.

Czas wykonania operacji wynosi 90 chwil, co odpowiada 2415 operacjom na sekundę.

UWAGA: Operacja PAL nie może spowodować zapalenia warunku Nd.

3.5.1.4. Porównanie akumulatora i obszaru pamięci POR.

Zapis: 5TN

funkcja: porównanie

treść: porównanie zawartości akumulatora z obszarem pamięci o długości T+1 tetrad rozpoczynającym się od tetrad o adresie $/B/.2^{10}+N$.

Porównanie nie powoduje zmiany zawartości akumulatora ani pamięci, zostają jedynie ustawione warunki:

Warunek U - jeśli $a - n < 0$

Warunek Z - jeśli $a = n$

Warunek Nd - nie może się zapalić

Czas wykonania operacji porównania wynosi 72 chwile, co od-

powiada 3019 operacjom/sek.

3.5.2. Operacje dziesiętne "długie".

W wykonaniu operacji dziesiętnych "długich" bierze udział rejestr długi WA.

Długość argumentu /ilość cyfr argumentu pobieranego z pamięci/ jest równa $T+9$, gdzie T jest cyfrą ósemkową zapisaną w części T kodu operacji rozkazu.

Za pomocą operacji długich można więc wykonywać działania na argumentach o długości od 9 do 16 cyfr.

Znak wyniku pamiętany jest przy pomocy przerzutnika S_w /gdy wynik jest ujemny to $S_w=1/$. Bez względu na znak wyniku przerzutnik S_a zostaje zgaszony. Wynik operacji długiej jest więc bez względu na ilość cyfr znaczących pamiętany w długim rejestrze WA.

Czas wykonania operacji długich nie zależy od wartości kolumny T - czyli od ilości cyfr argumentu.

3.5.2.1. Pobranie Długiej Liczby PDL.

Zapis: 6TN

funkcja: $wa' := n$

treść: pobranie do długiego rejestru WA $T+9$ tetrad począwszy od tetrad o adresie $B/2^{10} + N$.

Znak liczby pamiętany jest przez przerzutnik S_w .

Czas wykonania operacji wynosi 90 chwil, co odpowiada około 2415 operacjom na sekundę.

3.5.2.2. Dodanie Długiej Liczby DDL.

Zapis: 7TN

funkcja: $wa' := n + wa$

treść: dodanie do długiego rejestru WA $T+9$ tetrad począwszy od tetrad o adresie $B/2^{10} + N$.

Znak sumy pamiętany jest przez przerzutnik S_w .

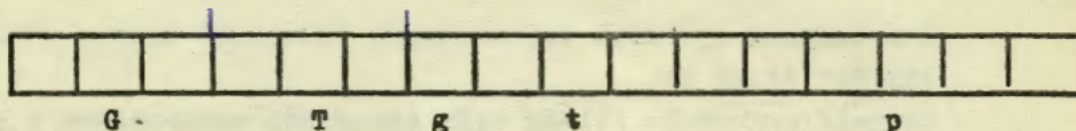
Czas wykonania operacji wynosi 162 chwile.

3.6. Rozkazy ujęte w tabeli trzeciej.

W tabeli trzeciej ujęte zostały rozkazy bezadresowe.

Kod operacji tych rozkazów ma 10 bitów 1, "zachodzi" na część

adresową. Część "g" i "t" kodu operacji może więc być zmieniona w czasie modyfikacji.



Rys. 11 Zapis rozkazów bezadresowych.

Dla wszystkich rozkazów tabeli trzeciej kod operacji zaczyna się od cyfr 07. Operacja jest bliżej precyzowana przez bity "g" i "t" rozkazu. Znaczenie części "p" rozkazu zależy od bitów "g" i "t" kodu operacji i będzie omówione szczegółowo przy odpowiednich rozkazach.

3.6.1. Operacje przesunięć krótkich.

Operacje przesunięć krótkich powodują przesuwanie 32 bitów rejestru A. Przesunięcie zachodzi w sposób naturalny tzn. bez uwzględnienia znaku liczby zapamiętanej w akumulatorze. W części "p" rozkazu podaje się ilość pozycji, o którą akumulator będzie przesunięty. Przesunięcie wykonuje się o

$$\bar{p} = p / \text{mod}.32/$$

pozycji.

Czas wykonania operacji przesunięć nie zależy od ilości przesunięć i wynosi 90 chwil, co odpowiada 2415 operacjom na sekundę. Operację przesunięć krótkich niszczą stare warunki i ustawiają nowe w następujący sposób:

Warunek U - nie może się zapalić,

Warunek Z - zapala się, jeśli po przesunięciu wszystkie bity akumulatora są równe zero.

Warunek Nd - zapala się, jeśli w czasie przesuwania w lewo zapalony zostaje bit A31 /najbardziej znaczący bit akumulatora/.

UWAGA: Operacje przesunięć krótkich nie zmieniają stanu przerzutników S_a i S_w .

3.6.1.1. Przesunięcie w LeWo Akumulatora LWA.

Zapis: 0700p

funkcja: funkcja zależy od typu liczby zapisanej w akumulatorze.

a/ $a' := a \cdot 2^{\bar{p}}$, jeśli w akumulatorze znajduje się dodatnia liczba binarna,

b/ $a' := a \cdot 10^{\bar{p}/4}$, jeśli $\bar{p} \text{ /mod.4/} \equiv 0$ i jeśli w akumulatorze zapisana jest liczba dziesiętna.

treść: przesunięcie akumulatora w lewo naturalnie o $\bar{p}$ bitów.

3.6.1.2. Przesunięcie w Prawo Akumulatora PRA.

Zapis: 0701p

funkcja: funkcja zależy od typu liczby zapisanej w akumulatorze,

a/ $a' := a \cdot 2^{\bar{p}}$, jeśli w akumulatorze znajduje się dodatnia liczba binarna,

b/ $a' := a \cdot 10^{\bar{p}/4}$, jeśli $\bar{p} \text{ /mod.4/} \equiv 0$ i jeśli w akumulatorze zapisana jest liczba dziesiętna,

treść: przesunięcie akumulatora w prawo o $\bar{p}$ bitów.

3.6.1.3. Przesunięcie akumulatora w LeWo /cyklicznie LWC/.

Zapis: 0702p

funkcja: $a' := a \cdot 2^{\bar{p}}$

treść: przesunięcie akumulatora w lewo cyklicznie o $\bar{p}$ bitów.

- UWAGA: Przesunięcie cykliczne oraz przesunięcie naturalne w lewo lub w prawo w przypadku liczb binarnych ujemnych nie mają sensu arytmetycznego. Działania przesunięć ilustrują poniższy rysunek:

a_{31}	a_{30}	a_{29}	a_{28}	a_{27}	a_{26}	$\cdot \cdot \cdot$	a_3	a_2	a_1	a_0
----------	----------	----------	----------	----------	----------	---------------------	-------	-------	-------	-------

Stan akumulatora przed przesunięciem

a_{30}	a_{29}	a_{28}	a_{27}	a_{26}	a_{25}	$\dots$	a_2	a_1	a_0	0
----------	----------	----------	----------	----------	----------	---------	-------	-------	-------	---

Stan akumulatora po wykonaniu przesunięcia w lewo o jedną pozycję /LWA/.

0	a_{31}	a_{30}	a_{29}	a_{28}	a_{27}	$\dots$	:	a_4	a_3	a_2	a_1
---	----------	----------	----------	----------	----------	---------	---	-------	-------	-------	-------

Stan akumulatora po wykonaniu przesunięcia w prawo o jedną pozycję /PRA/.

a_{30}	a_{29}	a_{28}	27	a_{26}	a_{25}	$\dots$	a_2	a_1	a_0	a_{31}
----------	----------	----------	----	----------	----------	---------	-------	-------	-------	----------

Stan akumulatora po wykonaniu przesunięcia cyklicznego w lewo o jedną pozycję /LWC/.

Rys. 12 Zawartość akumulatora po wykonaniu operacji przesunięć.

3.6.2. Operacje pomocnicze dla podprogramów mnożenia i dzielenia.

Działania mnożenia i dzielenia zrealizowane są w Kalkulatorze przy pomocy podprogramów. Dla skrócenia czasu wykonania tych podprogramów oraz zmniejszenia ilości komórek pamięci zajmowanych przez nie, wprowadzono w Kalkulatorze kilka operacji pomocniczych.

3.6.2.1. Operacja Przygotowania Mnożenia PMN.

Zapis: 0713 p

funkcja: $a' := a + w$; $/p/ = a + w$

treść: dodanie zawartości rejestru W do akumulatora i w komórce pamięci ferrytowej o adresie p.

Czas wykonania operacji wynosi 54 chwile, co odpowiada 4025 operacjom na sekundę.

Operacja przygotowania mnożenia nie zmienia warunków.

UWAGA: Operacja przygotowania mnożenia wykorzystana jest w podprogramie mnożenia do wpisania do pamięci

"tabliczki mnożenia", czyli umieszczenia w kolejnych komórkach iloczynów mnożnej /zawartości W/ przez wszystkie cyfry dziesiętne.

Jeśli $p=2$, to operacja przygotowania mnożenia ma funkcję $a'=w$; $/p/=w$.

3.6.2.2. Operacja Wykonania MNożenia WMN.

Zapis: 0714 p

funkcja: mnożenie przez jedną cyfrę mnożnika,

treść: przesunięcie rejestru WA o cztery bity w prawo i dodanie do rejestru W odpowiedniej wielokrotności mnożnej.

Czas wykonania tej operacji wynosi 54 chwile i nie zależy od cyfry sterującej mnożnika.

UWAGA: Operacja wykonania mnożenia niszczy warunki i ustawia warunek U, jeśli mnożna i mnożnik mają różne znaki.

Jeśli $p=2$, to operacja wykonania mnożenia nie bierze pod uwagę starej zawartości rejestru W.

Jeśli $p \neq 2$, to jego wartość jest obojętna.

3.6.2.3. Podprogram mnożenia.

Treść podprogramu mnożenia i zastosowany algorytm opisano w pkt 2.1.2.

UWAGA: Iloczyn zapamiętany zostaje w długim rejestrze WA, przy czym ewentualny znak "-" pamiętany jest przez przerzutnik S_w i warunek "U". Należy o tym pamiętać, jeśli iloczyn ma mniej niż 9 cyfr znaczących i jest traktowany w programie jak liczba krótka.

3.6.2.4. Operacja wykonawcza DZieLenia DZL.

Zapis: 0715 p

funkcja: obliczenie jednej cyfry ilorazu,

treść: odejmowanie cykliczne liczby z komórek zerowej i pierwszej pamięci od zawartości rejestru W do momentu, gdy następne odejmowanie dałoby w wyniku liczbę ujemną. Reszta zostaje zapamiętana w rejes-

Nr /8/	Zapis PJN	Nr /10/	Zapis PAS	U w a g i
1	2	3	4	5
0	002001	0	0	Tworzenie "tabliczki mnożenia"
1	478154	1	PAL+108,8	
2	071302	2	PMN2	
3	071304	3	PMN4	
4	071306	4	PMN6	
5	071310	5	PMN8	
6	071312	6	PMN10	
7	071314	7	PMN12	
10	071316	8	PMN14	
11	071320	9	PMN16	
12	071322	10	PMN18	Wykonanie mnożenia
13	278154	11	POL+108,8	
14	071402	12	WMN2	
15	071400	13	WMN	
16	071400	14	WMN	
17	071400	15	WMN	
20	071400	16	WMN	
21	071400	17	WMN	
22	071400	18	WMN	
23	071400	19	WMN	
24	071400	20	WMN	Komórki robocze
25	708140	21	DDL+96,1	
26	172000	22	MOD+0	
27	050001	23	SKB+1	
30	000000	24	0	
31	000000	25	0	
32	000000	26	0	
33	000000	27	0	
34	000000	28	0.	

Podprogram mnożenia

trze W, a ilość odejmowań zostaje wpisana do 4-bitowego licznika ilorazu. Operacja dzielenia niszczy stare warunki i ustawia nowe w następujący sposób:

Warunek U - jeśli dzielna i dzielnik mają różne znaki.

Warunek Z - jeśli po pierwszym odejmowaniu $/W/ = 0$.

Warunek Nd - nie może się zapalić.

Czas wykonania operacji zależy od ilości dokonywanych odejmowań i wynosi $36 /a+2/$ chwil, gdzie "a" jest ilością odejmowań.

UWAGA: Wartość p nie wpływa na wykonanie rozkazu.

3.6.2.5. Przesunięcie w LeWo Długie LWD.

Zapis: 0703 p

funkcja: wprowadzenie kolejnej cyfry ilorazu do akumulatora i przesunięcie dzielnej o jedną cyfrę dziesiętną w lewo,

treść: przesunięcie długiego rejestru WA o 4 bity w lewo i wprowadzenie na 4 najmniej znaczące bity akumulatora zawartości 4-bitowego licznika ilorazu /patrz: 3.6.2.4./. Operacja nie zamienia warunków. Czas wykonania operacji wynosi 90 chwil, co odpowiada 2415 operacjom na sek.

UWAGA: Liczba zapisana w części p rozkazu jest obojętna.

3.6.2.6. Podprogram dzielenia.

Treść podprogramu dzielenia oraz zastosowany algorytm opisano w pkt.2.1.3.

Po zakończeniu podprogramu warunki ustawione są wg akumulatora, w którym zapamiętany jest iloraz.

Ewentualny znak "-" /jeśli znaki dzielnej i dzielnika są różne/ zapamiętany jest przez przerzutnik S_a .

Wykonanie podprogramu dzielenia nie zmienia dzielnika zapamiętanego w komórkach 0 i 1 pamięci ferrytowej.

Nr /8/	Zapis PJN	Nr /10/	Zapis PAS	U w a g i
1	2	3	4	5
0	0	0	0	Wytestowanie dzielenia przez zero
1	478174	1	PAL+124,8	
2	270000	2	POL+0,8	
3	022036	3	SKZ+30	
4	278174	4	POL+124,8	
5	022032	5	SKZ+26	Wykonanie dzielenia
6	071500	6	DZL	
7	070300	7	LWD	
10	071500	8	DZL	
11	070300	9	LWD	
12	071500	10	DZL	
13	070300	11	LWD	
14	071500	12	DZL	
15	070300	13	LWD	
16	071500	14	DZL	
17	070300	15	LWD	Wykonanie dzielenia
20	071500	16	DZL	
21	070300	17	LWD	
22	071500	18	DZL	
23	070300	19	LWD	
24	071500	20	DZL	
25	070300	21	LWD	
26	071500	22	DZL	
27	070300	23	LWD	
30	172000	24	LOD=0	Wyjście z podprogramu
31	050001	25	SKB1	
32	070500	26	WYM	
33	022030	27	SKZ+24	
34	070500	28	WYM	
35	052006	29	SKB+6	
36	002030	30	STS+24	
37	000000	31	0	
40	000000	32	0	

Podprogram dzielenia

3.6.3. Operacje specjalne.

3.6.3.1. SPacjowanie Akumulatora SPA.

Operacja spacjiowania akumulatora jest operacją redakcyjną. Umożliwia ona taką redakcję karty, by nie znaczące zera wypełniające lewą część obszaru liczby /przed pierwszą cyfrą znaczącą/, nie były dziurkowane.

Zapis: 0704 p

funkcja: redakcja karty,

treść: wpisanie do akumulatora znaku odstępu /spacji/ na wszystkie tetrady, których zawartość jest równa zeru od tetrady najbardziej znaczącej do tetrady znajdującej się po lewej stronie pierwszej /od lewej/ tetrady różnej od zera włącznie.

Operacja spacjiowania nie zmienia warunków.

Czas wykonania tej operacji wynosi 54 chwile, co odpowiada 4025 operacjom na sek.

Parametr p nie wpływa na wykonanie operacji.

3.6.3.2. WYMiana akumulatora i rejestru W WYM.

Zapis: 0705 p

funkcja: $a' := w$; $w' := a$

treść: Wymiana informacji między rejestrami A i W..

Wymianie podlegają bity znaczące i znaki zapamiętane przez przerzutniki S_a i S_w .

Operacja niszczy warunki i ustawia nowe wg końcowego stanu akumulatora.

Czas wykonania operacji wymiany wynosi 54 chwile, co odpowiada 4025 operacjom na sek.

UWAGA: Parametr p nie wpływa na wykonanie rozkazu.

3.6.3.3. Umieszczenie Parametru p w rejestrze B UPB.

Zapis: 0710 p

funkcja: $/B/ := p$

treść: Umieszczenie w rejestrze B dwóch najmniej znaczących bitów parametru p rozkazu.

Operacja nie zmienia warunku. Czas wykonania opera-

cji wynosi 36 chwil, co odpowiada 6038 operacjom na sek.

UWAGA: Informacja zapisana na czterech bardziej znaczących bitach części p rozkazu jest bez znaczenia.

3.6.3.4. Czytanie Klawiatury CZK.

Operacja czytania klawiatury służy do ingerowania z pulpitu w sposób wykonania programu.

Zapis: 0711 p

funkcja: $U:=kl.1$, $Z:=kl.2$, $Nd:=kl.3$.

treść: zniszczenie starych warunków i wygenerowanie nowych w następujący sposób:

- a/ warunek U - jeśli wciśnięty jest klawisz K1
- b/ warunek Z - jeśli wciśnięty jest klawisz K2
- c/ warunek Nd - jeśli wciśnięty jest klawisz K3.

Czas wykonania operacji wynosi 36 chwil, co odpowiada 6038 operacjom na sek.

UWAGA: Nowe warunki nie zależą od informacji liczbowej zapisanych w rejestrach.

Informacja zapisana w części p rozkazu jest bez znaczenia. Stan przerzutników S_a i S_w nie zmienia się.

3.6.3.5. Zmiana Znaku Akumulatora ZZA.

Zapis: 0712 p

funkcja: $a':=-a$

treść: Zmiana stanu przerzutnika S_a na przeciwny.

Odpowiada to zmianie znaku liczby dziesiętnej zapisanej w akumulatorze. Operacja nie zmienia warunków Z i Nd, gasi warunek U i ustawia go na nowo wg stanu akumulatora po operacji. Czas wykonania operacji wynosi 36 chwil, co odpowiada 6038 operacjom na sek.

UWAGA: Parametr p nie ma znaczenia dla wykonania operacji.

3.6.4. Operacje wejścia - wyjścia.

Operacje wejścia - wyjścia umożliwiają przesyłanie informacji między Kalkulatorem oraz:

- a/ urządzeniami taśmy papierowej,
- b/ dalekopisem,
- c/ urządzeniami kartowymi,
- d/ pamięcią bębnową.

Operacjami wejścia nazywa się operacje, w wyniku których do akumulatora Kalkulatora lub do jego pamięci ferrytowej zostaje wprowadzona informacja z urządzeń zewnętrznych. Wszystkie operacje wejścia mają wspólny zapis:

0716 p

Operacjami wyjścia nazywa się operacje, w wyniku których z akumulatora Kalkulatora lub z jego pamięci ferrytowej zostaje wyprowadzona informacja do urządzeń zewnętrznych. Wszystkie operacje wyjścia mają wspólny zapis:

0717 p

Parametr p określa typ urządzenia zewnętrznego wg następującego klucza:

p = 1	pamięć bębnowa
p = 2	urządzenie kartowe
p = 4	urządzenie taśmy papierowej
p = 10	dalekopis

Operacja wejścia - wyjścia gaszą warunki i zapalają /bez względu na wczytaną ewentualnie wyprowadzoną informację/ warunek Z.

Czasy i sposób wykonania operacji zależą od rodzaju urządzenia i omówione będą później.

3.6.4.1. Operacje przesyłania bloku informacji między Kalkulatorem i pamięcią bębnową.

- a/ operacja przesyłania informacji z bębna do pamięci ferrytowej /operacja We/,

Zapis: 071601

funkcja: pobranie z bębna

treść: przepisanie bloku słów /16-bitowych + bit parzystości/ zapisanych na bębnie /począwszy od komórki o adresie podanym na bitach W14-W0 rejestru W do komórki o adresie podanym na bitach A14 - A0 akumulatora wyłącznie/ do kolejnych komórek pamięci ferrytowej począwszy od komórki o adresie podanym na bitach A25 - A16 akumulatora.

b/ Operacja przesłania bloku informacji z pamięci ferrytowej na bęben

Zapis: 071701

funkcja: zapisanie na bębnie,

treść: zapisanie na bębnie bloku słów /16-bitowych plus bit parzystości/ począwszy od komórki o adresie podanym na bitach W14 do W0 rejestru W, a skończywszy na komórce poprzedzającej komórkę o adresie podanym na bitach A14 - A0 akumulatora. Przepisywany blok informacji pamiętany jest w pamięci ferrytowej począwszy od komórki, której adres podany jest na bitach A25 - A16 akumulatora.

Maksymalna długość bloku, który można pobrać z bębna lub przesłać na bęben wynosi 256 komórek. W czasie wykonywania operacji następuje przesunięcie akumulatora o 9 bitów w prawo, a 8 najmniej znaczących bitów rejestru W staje się równe W8.

UWAGA: W ciągu 90 chwil poprzedzających wykonanie operacji bębnowej zawartości rejestru W nie może być zmieniona.

Podanie numeru ścieżki /jako części adresu komórki kończącej blok/ w akumulatorze nie jest konieczne.

Operacje bębnowe wykorzystują jedynie adres komórki na ścieżce /8 najmniej znaczących bitów całego adresu/. Niezbędne jest jednak podanie pełnego adresu w rejestrze W.

Na całkowity czas wykonania operacji bębnowej składają się następujące czasy częściowe:

- czas oczekiwania na porównanie adresów - $0 + 256$,
- czas przesyłania informacji - zależny od długości przesyłanego bloku informacji.

Jeśli przesyłanie bloku informacji wymaga kontaktu z dwoma /sąsiednimi/ ścieżkami bębna, to czas przesyłania informacji jest dłuższy i składa się z:

- czasu przesyłania informacji z pierwszej ścieżki,
- czasu potrzebnego na jeden obrót bębna, tzn. czasu użytego na zmianę adresu ścieżki i oczekiwania na zerową komórkę ścieżki następnej,
- czasu przesyłania informacji z ścieżki następnej.

Jak widać, przepisywanie bloków informacji znajdujących się na dwóch ścieżkach związane jest ze stratą jednego obrotu bębna. Blok zapisany jest na dwóch ścieżkach, jeśli adres na ścieżce pierwszej komórki bloku jest mniejszy od adresu na ścieżce ostatniej komórki bloku.

W

początek bloku na bębnie

W31				W15 W14	W8 W7	W0
-----	--	--	--	---------	-------	----

informacja zapisana na bitach
W15 do W31 jest obojętna

numer
ścieżki adres komórki
na ścieżce

A

	A25		A16 A14	A8 A7	A0
--	-----	--	---------	-------	----

adres komórki pa-
mięci ferrytowej
/początek bloku/

adres komórki
na ścieżce bęb-
na kończącej
przepisywany
blok

Rys. 13 Przygotowanie parametrów dla operacji bębnowej.

3.6.4.2. Operacje czytania i perforacji karty. I

a/ operacja wczytania karty do pamięci ferrytowej

zapis: 071602

funkcja: wczytanie karty,

treść: wpisanie do 20-tu kolejnych komórek pamięci ferrytowej /począwszy od komórki o adresie podanym na 9-ciu najmniej znaczących bitach rejestru W/ informacji zapisanej na karcie. Każdej kolumnie karty odpowiada w pamięci ferrytowej 4 bity tzn. 1 tetradą. Jeśli w kolumnie znajduje się dziurka, na którejkolwiek z cyfr, to do odpowiedniej tetrady zostaje wpisany kod dziesiętny prosty danej cyfry. Jeśli w kolumnie jest otwór w wierszu pierwszej naddziurki, to do tetrady zapisuje się kod

1111, 17

który odpowiada znakowi "-". Drugiej naddziurce odpowiada kod

1110. 16

Jeśli w kolumnie nie ma żadnego otworu, to zapamiętany zostaje kod

1100. 14

Karta zostaje wpisana do pamięci tak, że 80 kolumna pamiętana jest w najmniej znaczącej tetradzie /tetradą zerową/ komórki o adresie podanym w rejestrze W.

Kolumna 79 wpisuje się na tetradę pierwszą, 78 kolumna na tetradę drugą i 77 kolumna na tetradę trzecią tej komórki. Kolumna 76 zostaje zapamiętana w zerowej tetradzie komórki o adresie o 1 większym itd.

Sposób przedstawienia karty w pamięci ferrytowej ilustruje poniższy przykład.

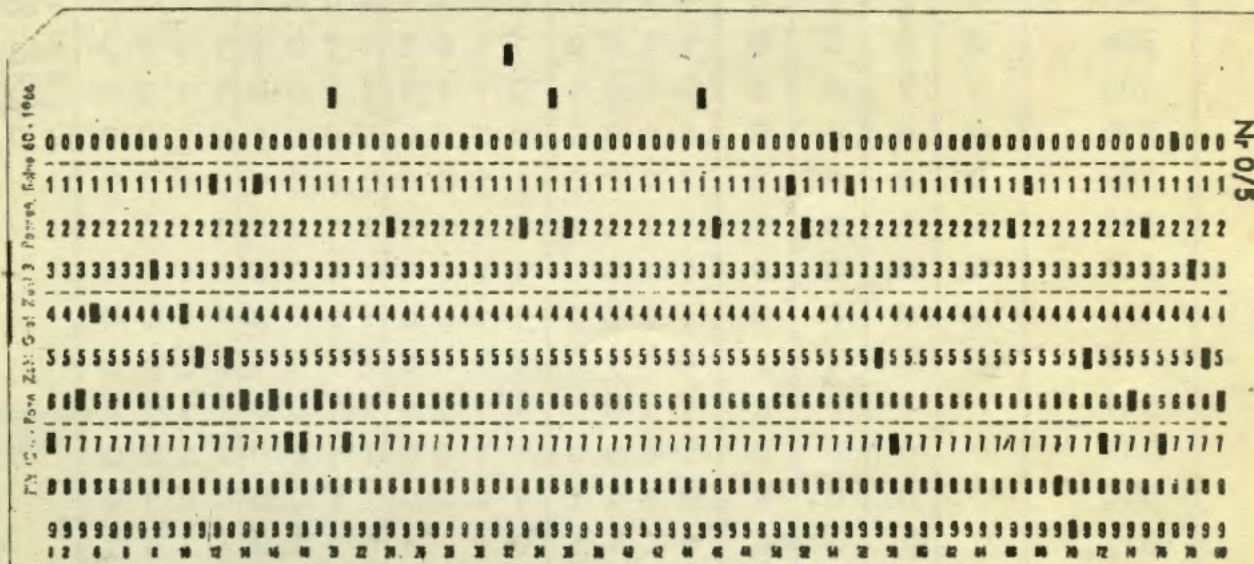
Przykład 3.

Należy wczytać kartę do obszaru pamięci od komórki 300 do 323 /ósemkowo/.

Pierwszą czynnością jest ustawienie adresu 300 w rejestrze W.

W8 W7 W6 W5 W4 W3 W2 W1 W0
0 1 1 0 0 0 0 0 0

Następnie należy wykonać rozkaz 071602. Załóżmy, że przed szczotkami odczytu przeszła karta przedstawiona na rys. 14.



Rys. 14

Adresy dla operacji dziesiętnych mają sens adresów tetrad. Możliwe jest więc bezpośrednie opracowanie wczytanej z karty informacji bez uprzedniego stosowania rozkazów redagujących.

- b/ Operacja perforowania karty
zapis: 071702
funkcja: perforowanie karty,

Adres ko- mórki /ósemk./	Liczba zapisana				Binarne przedstawienie komórki																Uwagi
	tet 3	tet 2	tet 1	tet 0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
1	2	3	4	5									6								7
300	0	3	5	6	0	0	0	0	0	0	1	1	0	1	0	1	0	1	1	0	Kody 1111 1110 1100 mają wartość cyfrową = 0 2-ga nad- dziurka
301	0	6	2	7	1	1	0	0	0	1	1	0	0	0	1	0	0	1	1	1	
302	8	9	5	7	1	0	0	0	1	0	0	1	0	1	0	1	0	1	1	1	
303	0	2	1	0	1	1	0	0	0	0	1	0	0	0	0	1	1	1	0	0	
304	0	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	
305	5	7	0	0	0	1	0	1	0	1	1	1	1	1	0	0	1	1	0	0	
306	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	1	1	1	0	0	
307	0	0	1	2	1	1	0	0	1	1	0	0	0	0	0	1	0	0	1	0	
310	-	2	0	0	1	1	1	1	0	0	1	0	1	1	0	0	1	1	0	0	
311	0	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	
312	0	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	
313	2	0	-	2	0	0	1	0	1	1	0	0	1	1	1	1	0	0	1	0	
314	0	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	1	0	
315	0	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	
316	7	0	0	2	0	1	1	1	1	1	0	0	1	1	0	0	0	0	1	0	
317	7	7	6	-	0	1	1	1	0	1	1	1	0	1	1	0	1	1	1	1	
320	5	6	1	6	0	1	0	1	0	1	1	0	0	0	0	1	0	1	1	0	
321	0	4	5	1	1	1	0	0	1	1	0	0	1	1	0	0	0	0	1	1	
322	0	0	0	3	1	1	0	0	1	1	0	0	1	1	0	0	0	0	1	1	
323	7	0	6	4	0	1	1	1	1	1	0	0	0	1	1	0	0	1	0	0	

Przedstawienie w pamięci operacyjnej informacji
wczytanej z karty/rys. 14/.

treść: wyperforowanie na karcie informacji zapamiętanej
w 20 kolejnych komórkach pamięci ferrytowej począ-
wszy od komórki, której adres podany jest na 9-ciu
najmniej znaczących bitach akumulatora.
Każdej tetradzie pamięci ferrytowej /tzn. 4 bitom/
odpowiada jedna kolumna karty. Jeśli w tetradzie
zapisany jest kod którejkolwiek z cyfr, to w odpo-
wiednim wierszu cyfrowym odpowiedniej kolumny wy-
perforowany zostanie otwór. Otworowi w wierszu

pierwszych naddziurek odpowiada kod

1111.

A otworowi w wierszu drugich naddziurek kod

1110.

Jeśli w tetradzie zapisana jest kombinacja bitów

1100.

/lub jakakolwiek inna kombinacja nie będąca kodem cyfry lub kodem pierwszej, ewentualnie drugiej naddziurki/, to w kolumnie karty nie zostanie wyperforowany żaden otwór.

Przygotowanie do perforowania informacji w obszarze pamięci może wymagać wykonania pewnej ilości operacji redakcyjnych SPA. Unika się w ten sposób osłabienia wyperforowanej karty przez perforowanie otworów odpowiadających nieznaczającym zerom na lewo od pierwszej znaczącej tetrady liczby. Przygotowanie adresu w akumulatorze jest analogiczne jak przygotowanie adresu w rejestrze W dla operacji czytania karty.

UWAGA: Obszary czytania ewentualnie perforowania karty mogą znajdować się jedynie w pierwszej połowie pamięci ferrytowej tzn. w komórkach 0 do 512.

W czasie perforacji karty, karta perforowana w poprzednim cyklu zostaje /w celu skontrolowania prawidłowości perforacji/ wczytana na pole od komórki 40 do 63 /ósemkowo/ czyli 32 do 51 /dziesiętnie/.

3.6.4.3. Operacje czytania i perforacji karty II.

Wykonanie rozkazu 071622 lub 071722 różni się tylko tym od wykonania rozkazów 071602 i 071702, że:

- a/ informacja strefy numerycznej wczytywana jest do /ew. wyprowadzana z/ pola o adresie wskazanym w rejestrze A lub W, przy czym bity 3-ci i 5-ty adresu muszą być równe jeden,
- b/ informacja strefy naddziurek wczytywana jest do /ew. wyprowadzana z/ pola o adresie zmniejszonym logicznie o 20 przez wyzerowanie dwóch bitów.
- c/ czytanie kontrolne odbywa się:
 - informacja numeryczna - od komórki 32

- informacja strefy naddziurek - od komórki 64.

Uwaga: Jeśli w kolumnie nie ma dziurki w strefie naddziurek, to do odpowiedniej tetrady wprowadzane zostaje zero.

3.6.4.4. Operacje czytania i perforowania taśmy papierowej.

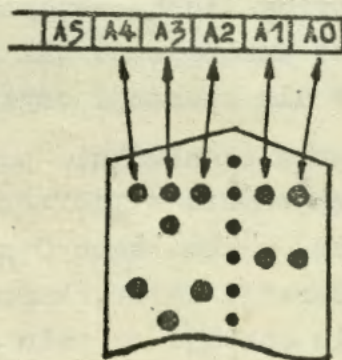
Operacje czytania i perforowania taśmy komunikują się z pięcioma najmniej znaczącymi bitami akumulatora.

p/ Operacja czytania jednego rządka taśmy.

zapis: 071604

funkcja: $a' := t$

treść: wyzerowanie pięciu najmniej znaczących bitów akumulatora i wpisanie na nie informacji binarnej z przeczytanego rządka taśmy.



Rys. 15. Wczytanie /wyperforowanie/ rządka taśmy do akumulatora.

b/ Operacja perforowania jednego rządka taśmy

zapis: 071704

funkcja: $t' := a$

treść: wyperforowanie na taśmie zawartości pięciu najmniej znaczących pozycji akumulatora.

3.6.4.5. Operacje wejścia - wyjścia z dalekopisu.

a/ Operacja czytania z dalekopisu

zapis: 071610

funkcja: $a' := d$

treść: wyzerowanie pięciu najmniej znaczących bitów akumulatora i wpisanie na nie kodu dalekopisowego znaku naciśniętego na klawiaturze dalekopisu.

UWAGA: Po zdeszyfrowaniu rozkazu czytania z dalekopisu Kalkulator zatrzymuje się i czeka na naciśnięcie przez programistę któregośkolwiek z klawiszy. Kod binarny znaku odpowiadającego naciśniętemu klawiszowi zostaje wprowadzony do akumulatora /w sposób analogiczny do pokazanego na rys. 15/ i Kalkulator przechodzi do dalszych obliczeń.

b/ Operacja pisania na dalekopis

zapis: 071710

funkcja: $a' := a$

treść: wydrukowanie na dalekopisie jednego znaku, którego kod binarny odpowiada informacji zapisanej na pięciu najmniej znaczących bitach akumulatora.

UWAGA: Operacja perforowania rzędka taśmy i drukowania na dalekopis nie zmieniają informacji zapisanej w akumulatorze.

Taśma	Wartość		Po znaku	
	dziesiętna	binarna	liter	cyfr
.	0	00000	pusta	
. 0	1	00001	e	3.
.0	2	00010	zmiana wiersza /lf/	
.00	3	00011	a	-
0.	4	00100	odstęp /sp/	✓
0. 0	5	00101	s	
0.0	6	00110	i	8
0.00	7	00111	u	7
0 .	8	01000	cofacz /cr/	
0 . 0	9	01001	d	
0 .0	10	01010	r	4
0 .00	11	01011	j	
00.	12	01100	n	,
00. 0	13	01101	f	
00.0	14	01110	c	:
00.00	15	01111	k	
0 .	16	10000	t	5
0 . 0	17	10001	z	+
0 .0	18	10010	l	
0 .00	19	10011	w	2
0 0.	20	10100	h	
0 0. 0	21	10101	y	6
0 0.0	22	10110	p	0
0 0.00	23	10111	q	1
00 .	24	11000	o	9
00 . 0	25	11001	b	?
00 .0	26	11010	g	
00 .00	27	11011	znak cyfr /fs/	
000.	28	11100	m	
000. 0	29	11101	x	/
000.0	30	11110	v	=
000.00	31	11111	znak liter /ls/	

Rys. 16 Kod zewnętrzny.

TABELE ROZKAZÓW O D R A 1103

T a b e l a I

G	T	0	1	2	3	4	5	6	7
0	STS	SKU	SKZ	SKD	SKN	SKB	S·K·W	TAB.III	
1	ODN	PON	DON	ODJ	ZLR	PAT	DOT	MOD	

Tabela II

Rozsze- rzenie adresu	T	G
B	2	P·O·L
	3	D·O·L
	4	P·A·L
	5	P·O·R
	6	P·D·L
	7	D·D·L
	0+7	

T a b e l a III

GT	t	0	1	2	3	4	5	6	7
07	0	LWA	PRA	LWC	LWD	SPA	WYM		
07	1	UPD	CZK	ZZA	PMN	WMN	DZL	WE	WY

N

T

G

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
											g				
												t			
													p		

Rys. 17

4. WSPÓŁPRACA KALKULATORA Z URZĄDZENIAMI WEJŚCIA - WYJŚCIA.

- 4.1. Współpraca Kalkulatora z zewnętrzną pamięcią bębnową. Wykonanie operacji przepisywania bloków informacji do pamięci bębnowej i z powrotem nie różni się pod względem organizacyjnym od wykonania operacji arytmetycznej czy organizacyjnej.

Po zeszyfrowaniu rozkazu i odczekaniu na odpowiednie ustawienie się bębna /ustawienie pierwszej komórki przepisywanego bloku pod głowicą/ następuje przesłanie informacji. Kalkulator przechodzi do wykonania rozkazu następnego dopiero wtedy, gdy przesłanie z bębna jest w pełni zakończone, tzn., gdy przesłana informacja jest w całości zapamiętana w pamięci ferrytowej lub na bębnie /zależnie od tego, czy wykonana operacja bębnowa była operacją wejścia czy wyjścia/.

Maksymalny czas wykonania operacji bębnowej jest jednak znacznie dłuższy od czasu wykonania operacji arytmetycznych i wynosi w przybliżeniu 65 msek /czyli czas trzech obrotów bębna/.

- 4.2. Współpraca Kalkulatora z urządzeniami taśmy papierowej z dalekopisem.

Współpraca Kalkulatora z czytnikiem taśmy, perforatorem taśmy i dalekopisem ułożona jest na nieco innych zasadach niż współpraca z bębniem.

Kalkulator może mianowicie przejść do wykonania następnego rozkazu mimo, że odpowiednia operacja wejścia-wyjścia nie została jeszcze zakończona /w tym sensie, że części mechaniczne urządzenia zewnętrznego nie powróciły jeszcze do położenia wyjściowego/.

- 4.2.1. Perforowanie taśmy i drukowanie na dalekopisie.

Jeśli Kalkulator "zwraca" się do perforatora taśmy lub dalekopisu po raz pierwszy po dłuższym odcinku czasu /po 6,6 msek do perforatora lub 0,1 sek. do dalekopisu/, to operacja wykonuje się natychmiast. Pobudzone zostają odpowiednie sterujące układy logiczne a informacja /która ma

zostać wyprowadzona/ zostaje przesłana do przejściowych rejestrów buforowych. Perforator wyposażony jest we własny rejestr buforowy, a rejestr buforowy dalekopisu znajduje się w Kalkulatorze.

Dalsze sterowanie urządzeniem zewnętrznym zachodzi równocześnie z pobraniem i wykonywaniem następnych rozkazów.

Jeśli jednak Kalkulator "zwraca" się po raz wtóry do tego samego urządzenia zewnętrznego po czasie krótszym od podanego wyżej, to konieczne jest odczekanie na sygnał gotowości urządzenia. Po otrzymaniu tego sygnału, Kalkulator wykonuje operację w sposób opisany powyżej.

Jeśli dwa rozkazy wyjścia wykonywane są bezpośrednio po sobie, to czas oczekiwania na "uspokojenie" się układów elektrycznych i mechanicznych może być znaczny i zbliża się do 6,6 msek dla perforatora i 0,1 sek. dla dalekopisu.

Czas czynnego wykonania operacji można pominąć w stosunku do czasu oczekiwania.

4.2.2. Czytanie taśmy.

Operacja czytania jednego rządka taśmy wykonywana jest podobnie jak operacje opisane w pkt. 4.2.1. Informacja, która wydziarkowana jest w rządku taśmy znajdującym się aktualnie nad fotodiodami czytnika zapamiętana jest w rejestrze buforowym czytnika. Rozkaz czytania taśmy powoduje przesłanie tej informacji do akumulatora i wygenerowanie impulsu START czytnika. Czas zajętości czytnika tzn. czas między impulsem START i pojawieniem się sygnału gotowości czytnika do przeczytania następnego rządka taśmy wynosi /dla czytnika FC-11/ około 3,0 msek.

Po wykonaniu operacji czytania /tj. po wpisaniu rządka taśmy do akumulatora i wypracowaniu impulsu START/ Kalkulator przechodzi do wykonywania następnych rozkazów programu nie czekając na pojawienie się sygnału gotowości. Czekanie na ten sygnał konieczne jest jedynie wtedy, gdy w programie przewidziane jest wczytanie dwóch /lub więcej/ rządków taśmy i jeśli rozkazy czytania pojawiają się w odstępach krótszych od 3,0 msek. Czytnik pracuje wtedy z maksymalną szybkością; taki reżim pracy jest z punktu widzenia jego eksploatacji najbardziej korzystny.

4.2.3. Wczytanie jednego znaku z dalekopisu.

Po zeszyfrowaniu rozkazu czytania jednego znaku dalekopisowego Kalkulator wstrzymuje dalsze wykonywanie programu i czeka na naciśnięcie któregokolwiek klawisza dalekopisu /lub wczytanie znaku przez czytnik dalekopisu/. Czas oczekiwania nie jest ograniczony z góry.

Po naciśnięciu klawisza do akumulatora wprowadzony zostaje kod binarny odpowiedniego znaku i Kalkulator kontynuuje wykonywanie programu.

UWAGA: Jak wynika z powyższego opisu czas czynnego wykonywania operacji wejścia - wyjścia /czas wyprowadzenia lub wprowadzenia informacji jest krótki w stosunku do czasu zajętości urządzeń zewnętrznych.

Pisząc programy tak, by zwracały się one do urządzeń zewnętrznych w odpowiednich odstępach czasu można uzyskać jednoczesną pracę z maksymalną prędkością tych urządzeń, a jednocześnie nie tracić czasu pracy Kalkulatora na oczekiwanie.

Przez jednoczesną pracę należy rozumieć, że Kalkulator zwraca się do jednych urządzeń w czasie, gdy inne są zajęte tzn., że czasy oczekiwania wykorzystane są na obliczenia.

4.3. Współpraca Kalkulatora z urządzeniami kartowymi.

4.3.1. Uwagi ogólne.

Kalkulator może współpracować z:

reproducerem PR 80-2

tabulatorem T5-M

Nie wykorzystuje on jednak pełnej możliwości tych maszyn, co szczególnie ostro widoczne jest przy pracy z tabulatorem. Kalkulator traktuje maszyny analityczne jedynie jako urządzenie pozwalające na wczytanie karty i wyperforowanie wyników na karcie /reproducer/ lub wydrukowanie tych wyników przy pomocy aparatu piszącego /tabulator/.

Programy zewnętrzne maszyn analitycznych nie mają więc zastosowania, nie są również używane liczniki tabulatora ani

układy porównywania kontrolnego reproducera.

Maszyny analityczne współpracują z Kalkulatorem START -

- STOP-owo. Impuls startowy z Kalkulatora powoduje włączenie silnika na czas przejścia jednej karty. Po przejściu karty maszyna analityczna zatrzyma się, jeśli nie zostanie wypracowany następny impuls START.

4.3.2. Współpraca z reproducerem PR80-2.

Odczytu kart dokonuje się na drodze czytania reproducera przy pomocy pierwszych szczotek. Szczotki kontrolne nie są na ogół wykorzystywane /patrz uwaga/. Droga reprodukcji kart jest przerwana.

Karty perforowane są na drodze perforacji. Karta wyperforowana w pierwszym cyklu wczytana zostaje /przy pomocy szczotek kontrolnych drogi perforacji/ na stałe miejsce pamięci ferrytowej. Umożliwia to programową kontrolę prawidłowości perforacji.

UWAGA: Możliwe jest przełączenie układów wczytania kontrolnego na szczotki czytania kontrolnego na drodze odczytu. Kalkulator będzie wtedy sprawdzał prawidłowości czytania kart. Perforacja nie będzie testowana.

Zrealizowane technicznie czytanie i perforacja pozwalają jedynie na opracowywanie kart, które mają co najwyżej jedną dziurkę w kolumnie.

4.3.3. Współpraca z tabulatorem T5-M.

Odczytu karty dokonuje się przy pomocy szczotek dolnych.

Dzięki dobudowaniu trzech dodatkowych krzywek możliwe jest wczytywanie otworów z wszystkich dwunastu wierszy karty.

Do drukowania przy pomocy aparatu piszącego nie wykorzystuje się liczników.

Drukowaniem steruje się bezpośrednio z Kalkulatora przy pomocy 80-ciu równoległych kanałów nie jest więc możliwe drukowanie znaków we wszystkich 83 kolumnach aparatu piszącego. Trzy kolumny pozostaną nie wykorzystane.

4.3.4. Organizacja logiczna współpracy Kalkulatora z urządzeniami kartowymi.

Sposób wykonania operacji kartowej różni się zasadniczo od sposobu wykonania wszystkich omówionych operacji.

Wykonanie rozkazów nie związanych z urządzeniami kartowymi charakteryzowało się tym, że Kalkulator przechodził do wykonania rozkazu następnego dopiero wtedy, gdy opracowanie informacji związane z operacją poprzednią była zakończona /tzn., że operacja po jej zakończeniu nie zmieniała już stanu rejestrów i pamięci Kalkulatora/.

Inaczej jest w przypadku operacji kartowej. Rozkaz wczytania przy perforacji karty powoduje jedynie zapamiętanie adresu początku pola pamięci ferrytowej i wypracowanie sygnału startu dla odpowiedniego urządzenia. Kalkulator zdąży wykonać kilka tysięcy rozkazów zanim wczytanie /ewentualnie perforacja/ karty efektywnie się rozpocznie.

Gdy pierwszy wiersz karty /wiersz dziewiątek/ znajdzie się pod szczotkami odczytu /ewentualnie pod nożami perforującymi/, to obliczenia zostają wstrzymane na czas około 400 μ sek, które zużywa Kalkulator na sterowanie odczytem /perforacją/ dziurek znajdujących się w tym wierszu.

Wstrzymanie takie ma miejsce dla każdego wiersza, czyli dwanaście razy przy opracowaniu jednej karty.

Czas między wstrzymaniami wykorzystywany jest przez Kalkulator do normalnych obliczeń.

4.3.5. Programy wykorzystujące urządzenia kartowe.

Programy wykorzystujące urządzenia kartowe nie różnią się w zasadzie od programów, które z tych urządzeń nie korzystają. Należy jednak pamiętać o pewnych ograniczeniach wynikających z konstrukcji i sposobu pracy maszyn analitycznych:

- a/ nie można używać do obliczeń informacji wczytywanej z karty przed ostatecznym zakończeniem wczytywania,
- b/ w czasie perforacji nie można zmieniać informacji zapisanej w polu pamięci, z którego perforuje się kartę,

- c/ jeśli w czasie jednego cyklu kartowego karty mają być zarówno wczytywane /karty z drogi odczytu/, jak i perforowane /karty z drogi perforacji/, to rozkazy zapoczątkowujące wczytanie i perforację muszą być wykonane w krótkim odstępie czasu jeden po drugim. Rozkazy muszą być już wykonane, gdy pojawią się odpowiednie sygnały wstrzymań z urządzeń kartowych,
- d/ programy, które wykorzystują jedynie drogę czytania reprodúcera, powodują przesuwanie się kart również na drodze perforacji. Własność ta jest istotna, jeśli ilość kart czytanych w programie przekracza ilość kart perforowanych.

154220

071000	671314	071601	671320	071601	671324	071601	000400
110400	150000	110000	170000	151377	130000	030233	110177
150000	110000	160000	070010	070500	111400	070020	071701
130000	030241	071000	671310	071701	671314	071701	671320
071701	001135	000000	000000	000400	001000	001400	010000
000000	470100	110007	150022	170022	200100	150028	070003
160023	160023	150023	170022	200077	160023	130022	030276
170270	050001	000000	000000	000000	000000	001001	010003
011005	020007	040011	041021	050023	051025	060027	100031
101041	110043	111045	120047	140051	141061	150063	151065
160067	200071	201101	210103	211105	220107	240111	241121
250123	251125	260127	300131	301141	310143	311145	320147
340151	341161	350163	351165	360167	400171	401201	410203
411205	420207	440211	441221	450223	451225	460227	000231
071000	110267	150000	110000	170000	150000	130000	030404
110174	070500	071622	110124	071702	110024	150000	110000
070400	170000	150123	130000	030421	050537	110024	150000
150001	110000	170000	160037	170000	150245	130000	030431
110000	170001	160077	170001	150217	130001	030440	050560
071100	040453	050410	270370	000410	000000	150312	170312
660030	140507	050510	470120	070500	470130	170312	210047
020500	070500	170312	270030	140507	050510	700127	470127
640123	030505	070400	070500	070400	170455	050001	000000
470000	071302	071304	071306	071310	071312	071314	071316
071320	071322	270000	071402	071400	071400	071400	071400
071400	071400	071400	071400	170507	050001	000000	071100
010557	671060	020557	600777	030557	110760	140455	050456
470537	070500	400547	600537	020557	110016	400536	050426
671320	020450	110001	370370	470370	050626	000000	111220
140455	050456	571237	020575	000575	070500	501247	020601
000601	120000	020604	000604	071100	020450	110016	150000
110000	170000	160251	150001	071200	330004	020623	070020
170000	120000	000623	130000	030610	050644	110000	160225
030567	050604	201305	100006	021131	100003	021131	001126
110000	051116	110000	051120	231321	030651	201325	030651
000651	201325	020664	100001	020664	100001	020664	100005
020664	100001	020664	000664	201270	020667	000667	201257
020672	000672	110000	161413	511330	020715	211330	030701
000450	060701	140270	050271	070010	070500	111400	070020
071601	110000	161413	030715	000450	211326	141376	051410
201305	141376	051411	201320	501470	020733	501500	010741
501504	010744	050736	201325	070037	020742	201210	010743
000743	201210	010740	050750	201325	070037	020736	050741
201320	100002	020761	100003	020767	100002	020761	051032
000000	201305	100005	020773	100004	020772	051032	201305
100011	020773	000773	171412	110000	150760	110000	170760
160001	070020	170760	160000	021024	541252	011027	110002
160760	150760	110000	170760	160001	070020	170760	160000
021032	551277	021027	051007	110000	160237	031034	201176
011034	001034	201176	011031	201320	150024	110000	161777
021131	100001	151125	021047	211275	100220	011223	110000
170024	051052	051121	051064	051101	051071	051131	051131
051116	051101	051075	051131	201316	000002	021117	110000
051120	201325	070037	021120	050640	201325	070037	021116
050642	211315	100103	021116	100001	021116	100002	021116
100022	021116	100001	021116	110000	051117	120002	120003
120002	161125	501272	021126	050632	000000	211273	031131
001131	211315	141376	170024	051377	071000	110371	410724
411006	411012	110361	410752	110400	150000	110000	170000
151377	130000	031147	111413	150313	110010	150314	141326
051327	050000	141312	051313	170313	110000	161161	170314
151401	141326	051327	511466	140270	050271	070001	161173
141312	051313	110000	161212	141312	051313	141326	051334
051214	051174	021372	051374	110000	161213	141312	051313
130314	031164	051225	131125	051047	170313	110000	151412
141326	051327	051233	070220	150000	070120	141312	051313
230000	141312	051313	141326	051327	051257	070220	150000
070120	141312	051313	230000	141312	051313	051334	141326
051327	051262	151777	110000	161402	151400	110000	161403
151401	110000	161407	151402	110000	161410	151403	110000
161413	140270	050271	070010	070500	111400	070020	071701
051144	000000	000000	170313	150000	110001	160313	070026
031322	001321	070126	150313	171312	050001	000000	110000
071604	100033	021334	051327	110000	470000	110000	071604
070001	150002	170002	210724	011364	370000	070004	470000
110000	071604	070001	150002	170002	210724	011360	051345
260001	021370	171326	050002	021370	311470	011336	000400
171326	050001	171376	050001	171376	000001	000000	050450
111160	151416	110000	171416	160217	131416	031403	100761
070220	100161	070210	100000	020230	000020	000000	055400

EKD ODRA 1103
OPIS OGÓLNY 1103-I-1

Lista zmian NrO

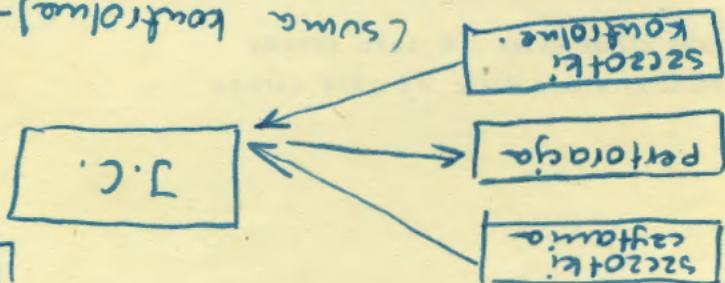
luty, marzec 1968

Lp.	Str.	Wiersz	Powinno być
1	19	- 8 [*]	skreślić: (wraz ze znakami).
2	52	+ 1 [*]	3.6.4.2. Operacje czytania i perforacji karty I.

^{*} - 8 ósmy wiersz licząc od dołu strony

^{*} + 1 pierwszy wiersz licząc od góry strony

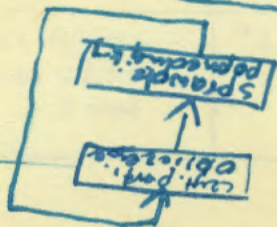
karta cyfrowa



(suma kontrolna) - PAL

01301

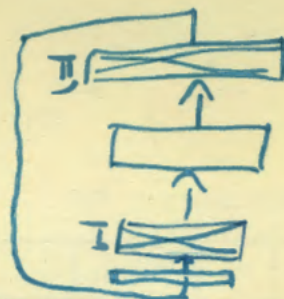
(suma kontrolna) - PAL



suma k. I

99999
10213
110212

99999
10313
110312



11111
10313
21424
11111
10213
11324

155000							
071001	110527	070500	071622	110020	150400	110005	150405
110000	150375	110050	150374	110135	150373	110205	150372
110527	070500	071622	110577	070500	071622	210652	510000
022032	002000	210650	410004	110664	070500	071622	
110050	150404	110240	142167	052170	110734	070500	071622
170374	110527	071722	110135	150404	110564	142167	052170
110527	070500	071622	170373	110527	071722	110205	150404
111024	142167	052170	110577	070500	071622	170372	110527
071722	110000	150404	142167	052170	110240	142373	052374
110664	070500	071622	170375	110527	071722	110050	150404
110240	142167	052170	110564	142373	052374	110734	070500
071622	170374	110527	071722	110135	150404	110564	142167
052170	111024	142373	052374	110527	070500	071622	170373
110527	071722	110205	150404	111024	142167	052170	110000
142373	052374	052072	142321	052322	110527	071722	002000

002170	052470	110010	170377	400651	071100	022250	170377
260543	460010	110000	070400	052205	150527	170404	150530
170404	150531	170404	150532	170404	150533	170404	150534
170404	150535	170404	150536	170404	150537	170404	150540
170404	150541	210004	070500	260010	142344	052345	070120
300024	070104	070010	170377	440566	110002	170377	400633
052317	110000	070400	052254	150527	170404	150530	170404
150531	170404	150532	170404	150533	170404	150534	170404
150535	170404	150536	170404	150537	170404	150540	170404
150541	170377	260624	070500	210004	142344	052345	070114
300024	070104	070004	170377	440566	110003	170377	400633
172167	050001						

002322	110000	160377	070107	150367	110000	170367	162337
142373	052374	110527	071722	172321	050001	000240	000564
001024	000000	000000					

002345	470030	071302	071304	071306	071310	071312	071314
071316	071320	071322	270030	071402	071400	071400	071400
071400	071400	071400	071400	071400	172344	050001	

002374	150366	071100	022466	052434	260207	070500	071001
210004	142344	052345	070120	300024	070104	070010	071000
570232	022434	002463	260270	070500	071001	210004	142344
052345	070114	300024	070104	070004	071000	570232	022434
002463	110120	150365	071001	170366	270644	071000	170365
570170	022446	002457	110000	162440	100010	152440	110000
160365	100010	150365	032436	110027	070012	120644	152440
071001	172373	050001	071000	052416			

150377	170377	200651	022162	052171	006000		
--------	--------	--------	--------	--------	--------	--	--