

PRZEMYSŁOWY INSTYTUT AUTOMATYKI I POMIARÓW
MERA-PIAP
Al. Jerozolimskie 202 02-222 Warszawa Telefon 23-70-81

Ośrodek Automatyki Elektrycznej

440 Zespół Budowy Cyfrowych Urządzeń Systemowych A

Główny wykonawca

Wykonawcy mgr inż. Tadeusz Kacprowski
mgr inż. Marek Partyka
tech. Jolanta Grzegółka

Konsultant

Nr zlecenia

Instrukcja użytkowania systemu RTMT

Zleceniodawca praca własna

Pracę rozpoczęto dnia 87.04.29
Kierownik Zespołu

dr. inż. A. Syrczyński

zakończono dnia 87.12.15
Kierownik Ośrodka

prof. dr inż. T. Missala

Praca zawiera:

Rozdzielnik - ilość egz:

stron 57

Egz. 1 BOINTE

rysunków 3

Egz. 2 OAE-8

fotografii

Egz. 3 WA

tabel

Egz. 4

tablic

Egz. 5

załączników

Egz. 6

Nr rejestr. 5985

**Analiza deskryptorowa URZĄDZENIA AUTOMATYCZNEJ REGULACJI I STEROWANIA:
MIKROPROCESOR + OPROGRAMOWANIE + INTELDIGIT-PROWAY**

Analiza dokumentacyjna Instrukcja użytkowania systemu czasu rzeczywistego RTMT dla jednostki centralnej MM80. Dokumentacja zawiera opis eksploatacji systemu, opis ekstrakodów i podprogramów systemowych oraz opis programu monitora operatorskiego.

Tytuły poprzednich sprawozdań Dokumentacja ta zastępuje całkowicie dokumentację "Instrukcja użytkowania systemu RTMT" nr rej. 5167, która przestała być aktualna.

681.5 Technika sterowanie automatycznej

681.32:621.377-481.42 - Mikroprocesory

6
UKD

Spis treści

1.	Wstęp	- 3
2.	Opis i zasady działania systemu	- 4
2.1	Zarządzanie zadaniami	- 4
2.2	Budowa zadania	- 4
2.3	Włączanie oprogramowania użytkownika	- 5
2.4	Obsługa przerw	- 6
2.5	Obsługa urządzeń zewnętrznych	- 8
2.6	Warunkowe zawieszanie zadań	- 10
2.7	Zajętość pamięci	- 10
3.	Ekstrakody systemowe	- 11
3.1	Ekstrakod - uruchom zadanie - START	- 11
3.2	Ekstrakod - czytaj dane - CZYTD	- 12
3.3	Ekstrakod - pisz dane - PISZD	- 13
3.4	Ekstrakod - zwolnij urządzenie - ZWOLN	- 14
3.5	Ekstrakod - drukuj tekst w trybie alarmowym ALARM	- 15
3.6	Ekstrakod - uruchom zadanie okresowo OKRES	- 16
3.7	Ekstrakod - wyłącz zadanie okresowe - WYOKR	- 16
3.8	Ekstrakod - zawieś zadanie pod warunkiem - ZAWWA	- 17
3.9	Ekstrakod - zawieś zadanie pod warunkiem -- wykonania innego programu - ZAWPR	- 18
3.10	Ekstrakod - warunek spełniony - USTAW	- 18
3.8	Ekstrakod - zeruj warunek - ZERUJ	- 19
3.8	Ekstrakod - zawieś zadanie na czas - ZAWIE	- 19
3.8	Ekstrakod - uruchom zadanie o godzinie - URUCH	- 20
3.8	Ekstrakod - czytaj dane z czytnika - CZYTC	- 21
3.15	Ekstrakod - odłącz czytnik - ODLCT	- 22
3.16	Ekstrakod - dziurkuj dane na dziurkarce - DZIUR	- 22

3.17	Ekstrakod - pisz dane z zawieszeniem - PISZZ	- 23
3.18	Ekstrakod - odłącz dziurkarkę - ODLDT	- 24
3.19	Ekstrakod - zainstaluj zadanie - INSTZ	- 25
3.20	Ekstrakod - zainstaluj przerwanie - INSTP	- 25
3.21	Ekstrakod - zainstaluj program sprawdzania warunku INSTW	- 26
3.22	Ekstrakod - drukuj tekst w trybie alarmowym zawieszeniem - ALARZ	- 27
3.23	Ekstrakod - ustaw długość bufora wejściowego - DLBUF	- 27
3.24	Ekstrakod - maskuj przerwania - MASKA	- 28
4.	Eksploatacja programu Monitor Operatorski	- 29
4.1	Opis ogólny Monitora Operatorskiego	- 29
4.2	Opis dyrektyw Monitora Operatorskiego	- 30
5.	Podprogramy systemowe dostępne dla użytkownika	- 50
6.	Konfiguracja sprzętowa	- 54

1. Wstęp

System RTMT jest wielozadaniowym systemem operacyjnym czasu rzeczywistego, przeznaczonym dla pakietu jednostki centralnej MM80 systemu INTEL DIGIT-PROWAY. Niniejszy opis dotyczy wersji C. System umożliwia sterowanie programami - zadaniami w zależności od czasu, wzajemnych uwarunkowań oraz zdarzeń zewnętrznych. Ponadto system zapewnia komunikację oprogramowania użytkowego z monitorem ekranowym, drukarką znakowo-mozaikową, czytnikiem i dziurkarką taśmy papierowej. System posiada ekstrakody umożliwiające sterowanie zadaniami i urządzeniami zewnętrznymi.

W skład systemu wchodzi program inicjacji i restartu automatycznego oraz monitor przerw. Pod kontrolą systemu może pracować jako jedno z zadań program monitora operatorskiego MONITOR.

Dzięki modułowej budowie możliwe jest dla minimalizowania zajętości pamięci i czasu działania, generowanie wersji systemu odpowiedniej dla konkretnej aplikacji.

2. Opis i zasady działania systemu RTMT

2.1 Zarządzanie zadaniami

Zasadniczą częścią systemu jest koordynator zarządzający pracą zadań. Koordynator uruchamia zadanie zgodnie z zasadą priorytetów względnych. Zgłoszenie do wykonania zadania o wyższym priorytecie w trakcie wykonywania się zadania o priorytecie niższym, nie powoduje przerwania wykonywania się tego zadania. Następuje jedynie ustawienie odpowiednich wskaźników i zadanie zostanie uruchomione po zakończeniu wykonywania się zadania aktualnie pracującego. Istnieje możliwość zmiany zasady uruchamiania zadań przez koordynator z priorytetów względnych na priorytety bezwzględne przez odpowiednie napisanie programów obsługi przerwania użytkownika.

Koordynator może obsługiwać do 31 zadań o numerach od 1 do 31, przy czym zadanie nr 1 ma priorytet najwyższy, a zadanie 31 priorytet najniższy. Ilość zadań obsługiwanych przez koordynator może być ustalona na etapie generacji systemu.

W trakcie pracy systemu koordynator wyszukuje zadanie o najwyższym priorytecie, które jest zgłoszone do wykonania (lub do kontynuacji po zawieszeniu) i uruchamia je. Po zakończeniu pracy lub zawieszeniu tego zadania koordynator ponownie wyszukuje do wykonania zadanie o najwyższym priorytecie.

2.2 Budowa zadania

Aby dowolny program mógł pracować pod kontrolą systemu operacyjnego, konieczne jest nadanie mu formatu programu-

zadania.

Zadanie musi zaczynać się ustawieniem wskaźnika stosu na niezastrzeżony obszar RAM-u wewnętrznego pakietu MM80 lub RAM użytkownika. Długość stosu ustala użytkownik na podstawie analizy swojego oprogramowania pamiętając o tym, że zadanie może być przerwane przez przerwania zewnętrzne (do 8 przerw).

Zadanie musi kończyć się skokiem do adresu 83H określonego w systemie etykietą KOZAD.

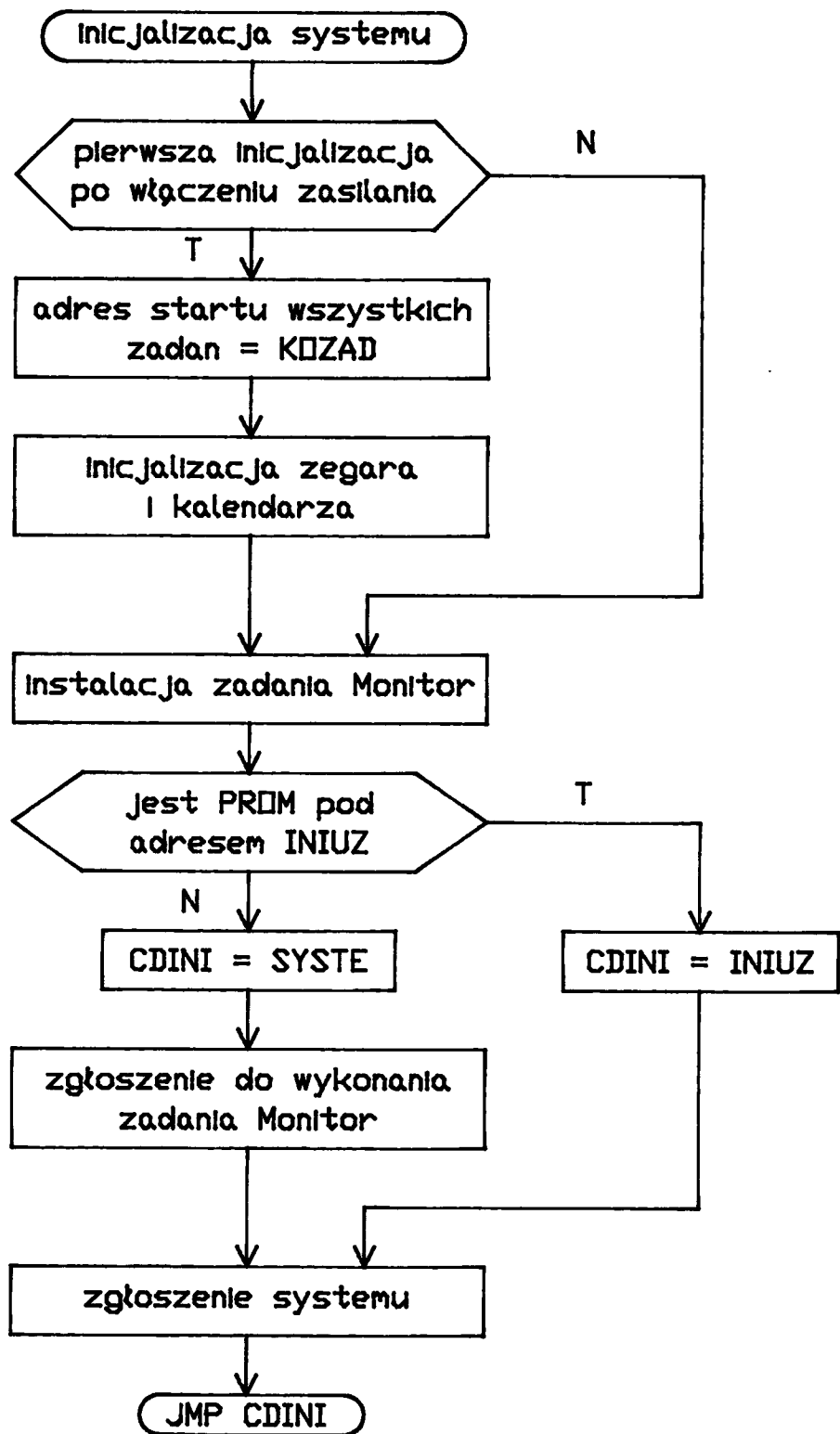
2.3 Włączanie oprogramowania użytkownika

Sposób włączania oprogramowania użytkownika zależy od zainstalowanego w systemie pakietu kontroli. Na rys.1 znajduje się schemat blokowy uruchamiania programu inicjującego oprogramowanie użytkownika dla systemu z pakietem MW30, a na rys. 2 - z pakietem MW32.

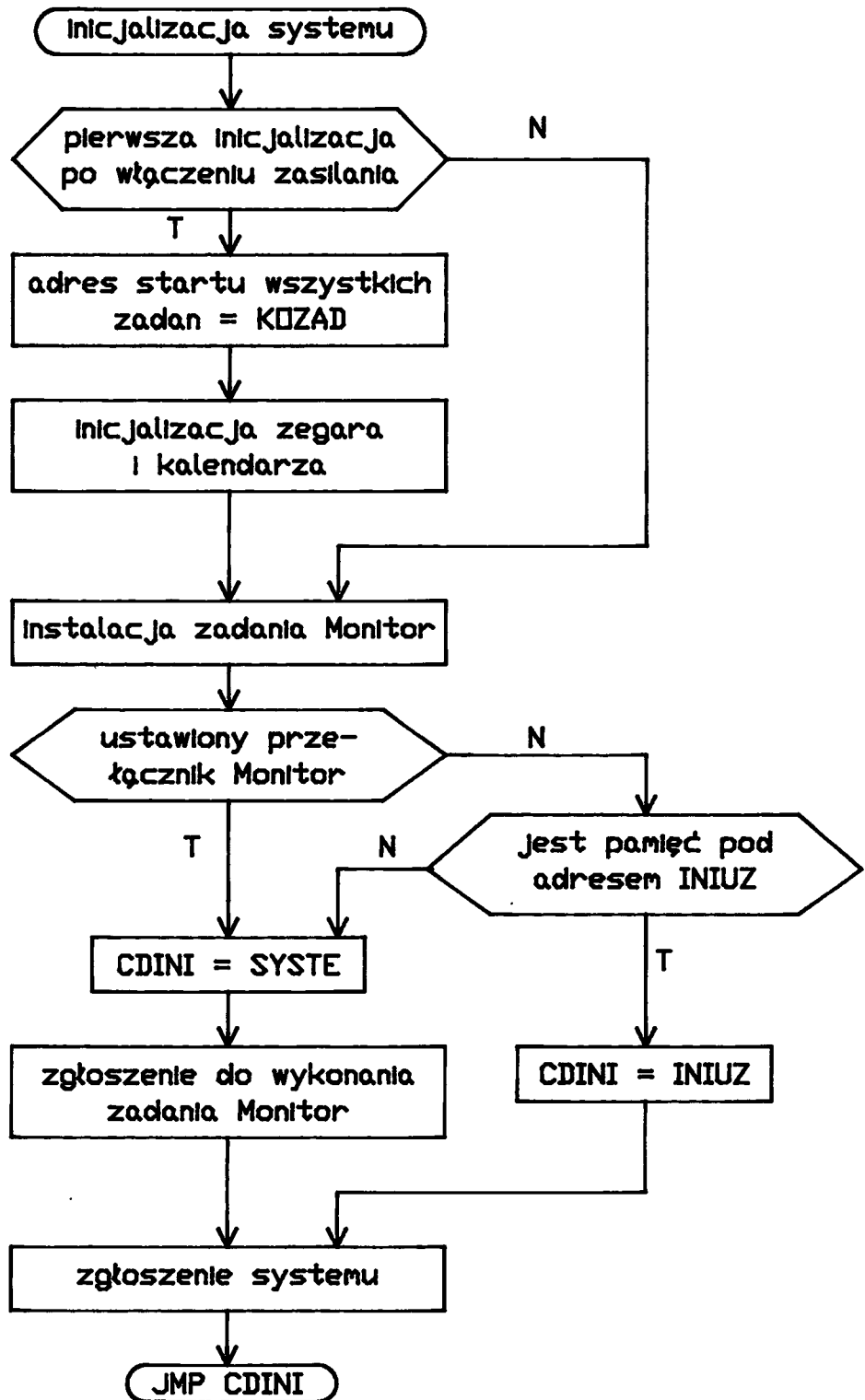
Po włączeniu zasilania lub po naciśnięciu przycisku RESET system startuje automatycznie i po wykonaniu niezbędnych działań inicjujących, uruchamia przy nie włączonych przerwaniach oprogramowanie użytkownika od adresu określonego systemową etykietą INIUZ. Ten fragment oprogramowania użytkownika uruchamiany jest automatycznie pod następującymi warunkami:

- a) dla pakietu MW30 o ile pod adresem określonym etykietą INIUZ znajduje się pamięć typu PROM
- b) dla pakietu MW32 o ile przełącznik MONITOR/PROGRAM ustawiony jest w pozycji PROGRAM, a pod adresem określonym etykietą INIUZ znajduje się pamięć typu RAM lub PROM.

Ten fragment oprogramowania zwany inicjacją użytkownika powinien być wykorzystany przez użytkownika do zainicjowania własnego oprogramowania, to znaczy wykonania takich czynności jak: zainstalowanie zadań, podanie adresów programów obsługi przerw, ustawienie własnych obszarów roboczych itp. Inicjacja użytkownika nie jest zadaniem. Ten



Rys 1. Uruchamianie programu inicjującego oprogramowanie użytkowe dla systemu z pakietem MW30



Rys 2. Uruchamianie programu inicjującego oprogramowanie użytkowe dla systemu z pakietem MW32

fragment oprogramowania nie wymaga ustawiania stosu użytkownika, natomiast musi kończyć się skokiem do adresu określonego w systemie etykietą SYSTE. Inicjacja użytkownika uruchamiana jest jednokrotnie po włączeniu zasilania lub naciśnięciu przycisku RESET. Adres w pamięci od którego ma być uruchamiana inicjacja użytkownika (etykieta INIUZ) może być określany na etapie generacji systemu.

Jeżeli pod adresem określonym etykietą INIUZ dla pakietu MW30 nie ma pamięci typu PROM, a dla pakietu MW32 nie ma pamięci typu RAM lub PROM lub przełącznik ustawiony jest w pozycji MONITOR, system nie uruchamia inicjacji użytkownika lecz po wykonaniu swojej inicjacji przechodzi przy odblokowanych (zezwoionych) przerwaniach do koordynatora zgłaszając jednocześnie zadanie MONITOR. Wówczas zadaniem inicjującym oprogramowanie użytkownika (jeśli jest niezbędne) może być dowolne zadanie uruchamiane dyrektywą monitora operatorskiego.

2.4 Obsługa przerw

Do prawidłowej pracy systemu konieczne jest doprowadzenie sygnału przerwania zegarowego do wejścia przerywającego IR1. Przerwanie to, zależnie od wersji systemu, może występować co 1 sek lub co 100 msec z tym, że obsługa systemowa wykonywana jest zawsze co 1 sekundę. Systemowa obsługa przerwania zegarowego polega na prowadzeniu zegara i kalendarza czasu astronomicznego, liczeniu time-outów czytnika i dziurkarki, obsłudze zadań okresowych, zawieszonych na czas, zgłoszonych do uruchomienia o konkretnej godzinie. Ponadto w ramach systemowej obsługi przerwania zegarowego jest wykonywana sekwencja instrukcji kontrolujących magistralę kasety (kaset dla zestawów wielokasetowych) i podtrzymywanie sygnału budzika w stanie pasywnym. W przypadku niepozytywnego wyniku kontroli magistrali odpowiednia komórka pamięci RAM o

etykiecie MAGIX jest ustawiana na wartość różną od zera (gdzie x - nr kasety: 1, 2 lub 3).

Oprócz przerwania zegarowego system wykorzystuje przerwanie od zaniku zasilania PFIN (podłączone do wejścia przerywającego IRO) oraz przerwania od zewnętrznych urządzeń dołączanych do systemu poprzez interfejsy V-24. Przerwania od urządzenia nr 0 dołączane są do wejścia przerywającego IR2, natomiast numery wejść przerywających od urządzeń 1 i 2 oraz 3 i 4, o ile urządzenia te występują w danej wersji systemu, ustalane są na etapie generacji.

Użytkownik może wykorzystywać w swoich zadaniach wszystkie przerwania nie używane przez zastosowaną przez niego wersję systemu. Ponadto istnieje możliwość kontynuacji przerwania od zaniku zasilania i przerwania zegarowego w oprogramowaniu użytkowym. W celu kontynuacji przerwania od zaniku zasilania należy w trzech komórkach pamięci RAM określonych etykietą CDPF umieścić instrukcję skoku do obsługi użytkownika. Obsługa przerwania od zaniku zasilania powinna kończyć się instrukcją HLT. Kontynuację obsługi przerwania zegarowego w oprogramowaniu użytkowym osiąga się przez wpisanie adresu tej kontynuacji do dwóch komórek systemowych określonych etykietą CDZEG (w kolejności bajtów: młodszy, starszy). Kontynuacja przerwania zegarowego uruchamiana jest co każde przerwanie, niezależnie od wykonywania obsługi systemowej i powinna kończyć się instrukcją skoku do etykiety WYODT lub WYKOD.

Po inicjacji systemu włączone są wszystkie przerwania. Użytkownik może wyłączać i włączać wszystkie przerwania nie używane przez system w swoim programie inicjacji lub w dowolnym miejscu oprogramowania za pomocą ekstrakodu MASKA. Nie powinien jednak, zmieniać bitów maski przerwania dotyczących przerwania używanych przez system.

Program obsługi przerwania użytkownika musi kończyć się skokiem do adresu 86H określonego w systemie etykietą WYODT, co zapewnia kontynuację programu przerwanoego przez to przerwanie. W ten sposób jest zrealizowana zasada

priorytetów względnych uruchamiania zadań.

Kończąc program obsługi przerwania skokiem do adresu 89H określonego etykietą systemową WYKOD można zapewnić realizację zasady priorytetów bezwzględnych uruchamiania zadań. Oznacza to, że w momencie zakończenia obsługi tego przerwania, koordynator sprawdza, czy jest zgłoszone do wykonania lub wznowienia zadanie o priorytecie wyższym, niż zadanie przerwane i ewentualnie uruchamia je.

2.5 Obsługa urządzeń zewnętrznych

Komunikacja zadań użytkownika ze wszystkimi urządzeniami zewnętrznymi dokonywana jest za pomocą ekstrakodów systemowych.

Zadanie może korzystać z urządzenia zewnętrznego pod warunkiem, że urządzenie to nie jest zajęte przez inne zadanie. Jeżeli urządzenie jest wolne, to pierwsze użycie ekstrakodu obsługującego to urządzenie przydziela je do zadania wywołującego ekstrakod. Gdy w chwili użycia ekstrakodu, urządzenie jest przydzielone do innego zadania, zadanie wywołujące ekstrakod zawiesza się w oczekiwaniu na zwolnienie urządzenia. Raz zajęte urządzenie przez dane zadanie jest do niego przydzielone, aż do momentu użycia ekstrakodu zwalnającego to urządzenie. Możliwe jest wtedy zajęcie tego urządzenia przez zadanie o najwyższym priorytecie spośród czekających na to urządzenie.

Należy pamiętać, że nie zwolnienie urządzenia uniemożliwia dostęp do niego dla innych zadań. Jedynym wyjątkiem jest podstawowe urządzenie operatorskie (urządzenie nr 0), na które można wyprowadzać komunikaty alarmowe (przy użyciu ekstrakodu ALARM lub ALARZ) niezależnie od faktu zajęcia tego urządzenia przez inne zadanie.

System może obsługiwać zależnie od wersji do pięciu urządzeń o interfejsie V-24 oraz czytnik i dziurkarkę taśmy papierowej. Urządzenia o interfejsie V-24 rozróżniane są

poprzez numery o wartościach od 0 do 4. Urządzenie o numerze 0 tzw. konsola operatorska jest obsługiwane przezsystem niezależnie od wersji i jest dołączone do złącza C pakietu MM80. Urządzenia o numerach od 1 do 4 obsługiwane są za pośrednictwem pakietów MI24. Wszystkie urządzenia o interfejsie V-24 obsługiwane są w sposób przerywaniowy. Urządzenie nr 0 dołączone jest do wejścia przerywającego IR2. Numery wejść przerywających urządzeń 1 do 4 określane są na etapie generacji systemu (po dwa urządzenia dołączone do jednego wejścia przerywającego). Dopuszczone są wejścia przerywające IR3 do IR7. Użytkownik ma możliwość kontynuacji przerwania od urządzeń 1,2 i 3,4 po odebraniu danych przychodzących (odebrany znak CR - kod ODH w ekstrakodzie CZYTD). W tym celu należy wpisać do komórek pamięci RAM oznaczonych etykietą CDU12 (dla urządzeń 1,2) lub CDU34 (dla urządzeń 3,4) adres odpowiedniej kontynuacji obsługi przerwania w kolejności bajtów: młodszy, starszy. Kontynuacja przerwania musi kończyć się skokiem do etykiety WYODT lub WYK00. W programie tym można stwierdzić, które urządzenie odebrało dane badając stan komórki oznaczonej etykietą systemową WKCR1 (dla urz. 1 i 2) lub WKCR3 (dla urz. 3 i 4). Bit D0 (najmłodszy) równy 1 sygnalizuje, że dane odebrało urządzenie nr 1 (nr 3) a bit D1 równy 1 oznacza, że dane odebrało urządzenie nr 2 (nr 4).

Odebranie za pośrednictwem kanału urządzenia nr 0 znaku DC3 (CTRL\S kod 13H) w trakcie wysyłania danych (ekstrakodem PISZD, PISZZ, ALARM lub ALARZ) powoduje wstrzymanie transmisji. Wznowienie transmisji następuje po odebraniu znaku DC1 (CTRL\Q - kod 11H).

Odebranie za pośrednictwem kanału urządzenia nr 0 znaku DEL (kod ASCII - 7FH) powoduje usunięcie z bufora danych wejściowych ostatnio odebranego znaku i wysłanie jako echa znaku BS (kod ASCII - 07).

2.6 Warunkowe zawieszanie zadań

Użytkownik może przy użyciu ekstrakodów systemowych zawiesić wykonywanie bieżącego zadania w oczekiwaniu na spełnienie określonego warunku zewnętrznego. Sprawdzenie, czy dany warunek jest spełniony, może być realizowane dwojako: przez system lub za pomocą odrębnego programu użytkownika uruchamianego przez koordynator. Ilość warunków obu typów jest ustalona na etapie generacji systemu. Maksymalna ilość warunków obu typów jest ustalona na etapie generacji systemu. Maksymalna ilość warunków każdego typu wynosi 127.

2.7 Zajętość pamięci

System RTMT łącznie z programem MONITOR (w wersji maksymalnie rozbudowanej) zajmuje praktycznie cały obszar pamięci PROM pakietu jednostki centralnej MM80 to jest adresy 0-1FFFFH. Ostatni zajmowany w każdej wersji systemu adres pamięci PROM znajduje się pod etykietą SYEND. Zajętość wewnętrznej pamięci RAM pakietu MM80 zależy również od wygenerowanej wersji systemu. Ogólnie mówiąc system wykorzystuje końcowe (do adresu 3FFFFH) komórki pamięci RAM pakietu MM80, a adres pierwszej wykorzystywanej komórki pamięci znajduje się pod etykietą RAMR0. Wolny do dowolnego wykorzystania przez użytkownika jest obszar pamięci RAM od adresu 3000H do RAMR0-1.

3. Ekstrakody systemowe

Ekstrakody systemowe stanowią jedyne źródło komunikacji oprogramowania użytkowego z systemem operacyjnym i urządzeniami zewnętrznymi. Zrealizowane są one w oparciu o instrukcję mikroprocesora RST3 o kodzie heksadecymalnym ODFH. W następnej komórce pamięci po kodzie instrukcji RST3 znajduje się parametr określający typ ekstrakodu, a w kolejnych ewentualne dalsze parametry. Wszystkie ekstrakody zachowują rejestry i słowo stanu procesora niezmienione i działają przy włączonych przerwaniach.

W oprogramowaniu użytkowym wygodnie jest zdefiniować wszystkie używane ekstrakody systemowe jako makrorozkazy. W punktach opisujących poszczególne ekstrakody systemowe podano ich przykładowe makrodefinicje. Nazwy ekstrakodów nie są obligatoryjne i użytkownik definiując makroinstrukcję, może wprowadzić inne nazwy.

3.1 Ekstrakod - uruchom zadanie - START

Definicja makroinstrukcji:

```
START  MACRO  NRZAD
        RST    3
        DB     0,NRZAD
        ENDM
```

Opis:

Ekstrakod START powoduje zgłoszenie zadania o numerze określonym przez parametr NRZAD do uruchomienia. Zadanie zostanie uruchomione przez koordynator, gdy przestanie działać (skończy się lub zawiesi) aktualnie wykonywane zadanie, a zadanie zgłoszone do wykonania będzie miało najwyższy priorytet spośród zadań czekających na

uruchomienie. Zgłoszenie do uruchomienia ekstrakodem START zadania już zgłoszonego, wykonującego się lub zawieszzonego z dowolnego powodu jest ignorowane. Ekstrakod może być użyty w dowolnym zadaniu użytkownika i w programie obsługi przerwania.

Przed użyciem ekstrakodu START należy zadanie zainstalować w systemie przy pomocy ekstrakodu INSTZ lub dyrektywą J programu MONITOR.

Przykład użycia:

START 5 - zgłoszenie do uruchomienia zadania nr 5

3.2 Ekstrakod - czytaj dane - CZYTD

Definicja makroinstrukcji:

```
CZYTD  MACRO  NRURZ, BUFWE
        RST    3
        DB     1, NRURZ
        DW     BUFWE
        ENDM
```

Opis:

Ekstrakod CZYTD powoduje zawieszenie wykonywania zadania go używającego do czasu wprowadzenia z urządzenia podłączonego do interfejsu V-24 o numerze NRURZ ciągu znaków zakończonego znakiem powrotu karetki <CR> o kodach ODH. Znaki te, łącznie ze znakiem CR zostają wprowadzone do bufora zdefiniowanego przez użytkownika od adresu BUFWE. Długość bufora wejściowego każdego z urządzeń zewnętrznych może być określona przy pomocy ekstrakodu DLBUF. Jeżeli ekstrakod DLBUF nie jest użyty to system przyjmuje długość wszystkich buforów wejściowych na wartość 80. Zapełnienie bufora tekstu znakami wprowadzonymi z urządzenia zewnętrznego jest równoznaczne z zakończeniem tekstu znakiem <CR> to znaczy powoduje kontynuację działania zadania od

miejsca bezpośrednio za ekstrakodem CZTD. Pierwsze użycie ekstrakodu CZYTD lub ekstrakodu pisania na to samo urządzenie PISZD i PISZZ powoduje przydzielenie urządzenia do tego zadania aż do czasu użycia ekstrakodu ZWOLN. Jeżeli urządzenie jest przydzielone do innego zadania, to zadanie bieżące zawiesza się do czasu zwolnienia urządzenia. Ilość wczytanych znaków znajduje się w komórce systemowej o etykiecie LBWEn gdzie n=0,1,... - numer urządzenia.

Przykład użycia:

```
CZYTD 1,POLE1 - czytaj dane z urządzenia nr 1 do
                pamięci RAM, począwszy od adresu
                określonego etykietą POLE1,
```

3.3 Ekstrakod - pisz dane - PISZD

Definicja makroinstrukcji:

```
PISZD MACRO NRURZ, BUFWY
      RST 3
      DB 2,NRURZ
      DW BUFWY
      ENDM
```

Opis:

Ekstrakod PISZD powoduje dopisanie do kolejki wyjściowej urządzenia NRURZ podłączonego do interfejsu V-24 adresu tekstu rozpoczynającego się od etykiety BUFWY określonej przez użytkownika. Na urządzenie wyprowadzone są kolejne znaki bufora BUFWY aż do znaku NULL (o kodzie 0) kończącego każdy wyprowadzany tekst. Ekstrakod nie zawiesza wykonywania zadania używającego go, to znaczy tekst jest wyprowadzany równocześnie z biegnącym dalej zadaniem.

Pierwsze użycie w bieżącym zadaniu ekstrakodu PISZD lub ekstrakodów pisania z zawieszeniem PISZZ i czytania CZYTD powoduje przydzielenie urządzenia do tego zadania, aż do

czasu użycia w tym zadaniu ekstrakodu ZWOLN. Jeżeli urządzenie jest przydzielone do innego zadania, to zadanie bieżące zawieszają się na ekstrakodzie FISZD do czasu zwolnienia urządzenia.

Przykład użycia:

```
FISZD 2,POLE2 -  pisz na urządzenie nr 2 dane
                  umieszczone w pamięci począwszy
                  od adresu określonego etykietą
                  POLE2.
```

3.4 Ekstrakod - zwolnij urządzenie - ZWOLN

Definicja makroinstrukcji:

```
ZWOLN  MACRO  NRURZ
        RST    3
        DB     3,NRURZ
        ENDM
```

Opis:

Ekstrakod ZWOLN powoduje odłączenie (zwolnienie) urządzenia NRURZ od zadania, które ten ekstrakod użyło. Jeżeli zadanie nie używało poprzednio ekstrakodów FISZD, FISZZ lub CZYTD dotyczących urządzenia NRURZ, do ekstrakodu ZWOLN jest ignorowany. Jeżeli urządzenie NRURZ było używane w zadaniu przy pomocy ekstrakodów FISZD, FISZZ lub CZYTD to przed zakończeniem zadania (skokiem do systemowej dyrektywy KOZAD) należy koniecznie zastosować ekstrakod ZWOLN. W przeciwnym wypadku nie będzie można korzystać z urządzenia w innych zadaniach.

Przykład użycia:

```
ZWOLN 0 - zwolnij (odłącz ) urządzenie numer 0
        od zadania
```

3.5 Ekstrakod - drukuj tekst w trybie alarmowym - ALARM

Definicja makroinstrukcji:

```
ALARM  MACRO  BUFWY
        RST    3
        DB     4
        DW     BUFWY
        ENDM
```

Opis:

Ekstrakod ALARM powoduje dopisanie do kolejki wydruków alarmowych adresu tekstu rozpoczynającego się od adresu BUFWY określonego przez użytkownika. Ekstrakod ALARM dotyczy zawsze druku na urządzeniu podstawowym systemu o numerze 0. Na urządzenie wprowadzane są kolejne znaki bufora BUFWY aż do znaku NULL (kod ASCII 0) kończącego tekst. Ekstrakod nie zawieszania wykonywania zadania używającego go, to znaczy tekst jest wyprowadzany równocześnie z wykonującym się zadaniem. Ekstrakod ALARM nie bada zajętości urządzenia 0 przez inne zadanie. Oznacza to, że tekst drukowany tym ekstrakodem jest wyprowadzany natychmiast po zakończeniu aktualnie drukowanego tekstu i może rozdzielić wydruki pisane przez inne zadanie na urządzeniu 0 przy pomocy ekstrakodów PISZD lub PISZZ.

Przykład użycia:

```
ALARM POLE3 - pisz na urządzenie 0 poza kolejką testów
              dane umieszczone w pamięci od adresu
              określonego etykietą POLE3.
```

3.6 Ekstrakod - uruchom zadanie okresowo - OKRES

Definicja makroinstrukcji:

```
OKRES  MACRO  NRZAD,KWANT
        RST    3
        DB     5,NRZAD
        DB     KWANT
        ENDM
```

Opis:

Ekstrakod OKRES powoduje takie ustawienie programowanych liczników systemu, że koordynator będzie zgłaszał do uruchomienia zadanie NRZAD co KWANT sekund. Ekstrakod może być użyty w dowolnym zadaniu (także NRZAD) oraz w programie obsługi przerwania.

Użycie ekstrakodu OKRES w stosunku do zadania, które było już uruchomione okresowo powoduje jedynie zmianę częstotliwości wywoływania tego zadania. Zadanie będzie uruchamiane cyklicznie aż do momentu użycia w dowolnym miejscu oprogramowania użytkownika ekstrakodu WYOKR wyłączającego zadanie okresowe. Po użyciu ekstrakodu OKRES zadanie NRZAD będzie po raz pierwszy zgłoszone do uruchomienia po KWANT sekund.

Przykład użycia:

```
OKRES  6,60  - uruchom zadanie nr 6 co 60 sekund
```

3.7 Ekstrakod - wyłącz zadanie okresowe - WYOKR

Definicja makroinstrukcji:

```
WYOKR  MACRO  NRZAD
        RST    3
        DB     6,NRZAD
        ENDM
```

Opis:

Ekstrakod WYOKR powoduje zakończenie cyklicznego uruchamiania zadania NRZAD włączonego uprzednio ekstrakodem OKRES lub dyrektywą P Monitora operatorskiego. Próba zastosowania ekstrakodu WYOKR do zadania, które nie jest włączone jako zadanie okresowe, nie powoduje żadnych reakcji. Ekstrakod WYOKR może być użyty w dowolnym zadaniu (także NRZAD) lub w programie obsługi przerwania.

Przykład użycia:

WYOKR 6 - wyłącz zadanie nr 6 - jeśli było uprzednio włączone jako okresowe.

3.8 Ekstrakod - zawieś zadanie pod warunkiem -ZAWWA

Definicja makroinstrukcji:

```
ZAWWA  MACRO  NRWAR
          RST   3
          DB    7,NRWAR
          ENDM
```

Opis:

Ekstrakod ZAWWA powoduje zawieszenie wykonywania zadania go wywołującego do czasu stwierdzenia przez koordynator systemu, że warunek nr NRWAR jest ustawiony (spełniony). Warunek można ustawiać i zerować przy pomocy ekstrakodów USTAW i ZERUJ w innych zadaniach lub w programach obsługi przerwania.

Przykład użycia:

ZAWWA 2 - zawieś bieżące zadanie w oczekiwaniu na spełnienie warunku nr 2

3.9 Ekstrakod - zawieś zadanie pod warunkiem wykonania innego programu - ZAWPR

Definicja makroinstrukcji:

```
ZAWPR  MACRO  NRPRO
          RST   3
          DB    8,NRPRO
        ENDM
```

Opis:

Ekstrakod ZAWPR powoduje zawieszenie wykonywania zadania go wywołującego do czasu stwierdzenia przez dodatkowy program użytkownika, że można zadanie kontynuować. Program dodatkowy sprawdzający możliwość kontynuacji zadania musi być uprzednio zainstalowany przy użyciu ekstrakodu INSTW. Musi on kończyć się skokiem do adresu BCH określonego systemową etykietą KOWAR z odpowiednio ustawioną flagą CY: CY=0, gdy warunek nie jest spełniony lub CY=1, gdy zadanie można kontynuować.

Przykład użycia:

```
ZAWPR  5 - zawieś bieżące zadanie w oczekiwaniu na
          spełnienie warunku sprawdzanego przez
          program zainstalowany ekstrakodem INSTW pod
          numerem 5.
```

3.10 Ekstrakod - warunek spełniony - USTAW

Definicja makroinstrukcji:

```
USTAW  MACRO  NRWAR
          RST   3
          DB    9,NRWAR
        ENDM
```

Opis:

Ekstrakod USTAW powoduje ustąpienie warunku nr NRWAR. Jeśli warunek był wcześniej ustawiony, działanie ekstrakodu jest ignorowane. Użycie ekstrakodu USTAW powoduje zgłoszenie do kontynuacji zadania zawieszonoego pod tym warunkiem.

Przykład użycia:

USTAW 8 - ustaw warunek nr 8

3.11 Ekstrakod - zeruj warunek - ZERUJ**Definicja makroinstrukcji:**

```
ZERUJ  MACRO  NRWAR
        RST    3
        DB     10,NRWAR
        ENDM
```

Opis:

Ekstrakod ZERUJ powoduje wyzerowanie warunku nr NRWAR

Przykład użycia:

ZERUJ 2 - zeruj warunek nr 2

3.12 Ekstrakod - zawieś zadanie na czas - ZAWIE**Definicja makroinstrukcji:**

```
ZAWIE  MACRO  KWANT
        RST    3
        DB     11,KWANT
        ENDM
```

Opis:

Ekstrakod ZAWIE powoduje takie ustawienie programowych liczników systemu, że zadanie używające ten ekstrakod będzie

kontynuowane od miejsca bezpośrednio za ekstrakodem ZAWIE po czasie KWANT sekund. Użycie ekstrakodu ZAWIE w programie obsługi przerwania jest zakazane.

Przykład użycia:

ZAWIE 10 - zawieś bieżące zadanie na 10 sekund

3.13 Ekstrakod - uruchom zadanie o godzinie - URUCH

Definicja makroinstrukcji:

```
URUCH  MACRO  NRZAD,GODZ
        RST    3
        DB     12,NRZAD
        DW     GODZ
        ENDM
```

Opis:

Ekstrakod powoduje takie ustawienie programowych liczników systemowych, że zadanie NRZAD zostaje zgłoszone do wykonania o godzinie GODZ. Ekstrakod może być użyty w dowolnym zadaniu użytkownika (także NRZAD) oraz w programie obsługi przerwania. Jeżeli o godzinie GODZ zadanie NRZAD jest zawieszona lub w trakcie wykonywania ekstrakod jest ignorowany. Parametr GODZ musi być podany w formacie BCD na dwóch kolejnych bajtach, przy czym pierwszy określa minuty, a drugi godzinę żadanego czasu uruchomienia zadania. Zadanie zostanie zgłoszone do wykonania, gdy godziny i minuty czasu ustawionego w systemie osiągną parametr GODZ, a sekundy przyjmą wartość 0.

Przykład użycia:

URUCH 9,1235H - uruchom zadanie nr 9 o godzinie
12 minut 35 sekund 00

3.14 Ekstrakod - czytaj dane z czytnika - CZYTC

Definicja makroinstrukcji:

```
CZYTC  MACRO  ILZNA, BUFWE
        RST    3
        DB    13, ILZNA
        DW    BUFWE
        ENDM
```

Opis:

Ekstrakod CZYTC powoduje zawieszenie aktualnie wykonywanego zadania do czasu wczytania przez czytnik określonej przez parametr ILZNA ilości znaków. Znaki te są wprowadzane do bufora określonego przez użytkownika parametrem BUFWE. Po zakończeniu ekstrakodu CZYTC użytkownik może sprawdzić, czy dane zostały prawidłowo wczytane do bufora. W komórce określonej systemową etykietą STANC znajduje się informacja zwrotna ekstrakodu CZYTC w formacie:

- . 0 - poprawne wczytanie danych
- . 40H - czytnik nie zgłasza gotowości
- . 80H - time-out czytnika - czytnik nie otrzymał nowego znaku w okresie 2 sekund

Użycie ekstrakodu powoduje przyłączenie czytnika do zadania. Przed zakończeniem zadania (skokiem do systemowej etykiety KOZAD) należy odłączyć czytnik od zadania ekstrakodem ODLCT.

Przykład użycia:

```
CZYTC 100, POLE4 - czytaj 100 znaków z czytnika
                  do pamięci RAM począwszy od
                  adresu określonego etk. POLE4
```

3.15 Ekstrakod - odłącz czytnik - ODLCT

Definicja makroinstrukcji:

```
ODLCT  MACRO
        RST      3
        DB       14
        ENDM
```

Opis:

Ekstrakod ODLCT powoduje odłączenie (zwolnienie) czytnika taśmy papierowej od zadania, które ten ekstrakod użyło. Jeżeli zadanie nie używało poprzednio ekstrakodu czytania danych z czytnika CZYTC, to ekstrakod ODLCT jest ignorowany. Jeżeli zadanie czytało dane z czytnika ekstrakodem CZYTC, to przed zakończeniem zadania (skokiem do systemowej etykiety KOZAD) należy koniecznie użyć ekstrakodu ODLCT. W przeciwnym wypadku nie będzie można korzystać z czytnikach w innych zadaniach (także w zadaniu monitora operatorskiego).

Przykład użycia:

```
ODLCT - odłącz czytnik od aktualnie wykonującego
        się zadania.
```

3.16 Ekstrakod - dziurkuj dane na dziurkarce - DZIUR

Definicja makroinstrukcji:

```
DZIUR  MACRO  ILZNA,  BUFWY
        RST      3
        DB       15,  ILZNA
        DW       BUFWY
        ENDM
```

Opis:

Ekstrakod powoduje zawieszenie wykonywania zadania go

używającego do czasu wydziurkowania na dziurkarce ilości ILZNA znaków z bufora określonego przez użytkownika parametrem BUFWY. Po zakończeniu ekstrakodu DZIUR użytkownik może sprawdzić, czy dane zostały prawidłowo wydziurkowane. W komórce określonej systemową etykietą STAND znajduje się informacja zwrotna ekstrakodu DZIUR w formacie:

0 - poprawne wydziurkowanie danych
40H - dziurkarka nie zgłasza gotowości
80H - time-out dziurkarki - dziurkarka nie wydziurkowała nowego znaku w okresie 4 sek

Użycie ekstrakodu powoduje przyłączenie dziurkarki do zadania. Przed zakończeniem zadania (skokiem do systemowej etykiety K0ZAD) należy odłączyć dziurkarkę od zadania ekstrakodem ODLDT.

Przykład użycia:

DZIUR 50, POLES - dziurkuj 50 znaków z pamięci
począwszy od adresu określonego
etykietą POLES.

3.17 Ekstrakod - pisz dane z zawieszeniem - FISZZ

Definicja makroinstrukcji:

```
FISZZ  MACRO  NRURZ,  BUFWY
        RST    3
        DB     16, NRURZ
        DW     BUFWY
        ENDM
```

Opis:

Ekstrakod FISZZ powoduje dopisanie do kolejki wyjściowej urządzenia nr NRURZ podłączonego do interfejsu V-24 adresu testu rozpoczynającego się etykietą BUFWY określoną przez

użytkownika. Na urządzenie wyprowadzane są kolejne znaki bufora BUFWY aż do znaku NULL (o kodzie 0) kończącego tekst. Ekstrakod zawieszka wykonywanie zadania używającego go aż do czasu wyprowadzenia na urządzenie ostatniego znaku bufora BUFWY. Ekstrakod przydziela urządzenie do zadania do czasu użycia ekstrakodu ZWOLN.

Przykład użycia:

```
PISZZ 2,POLE6 - pisz na urządzenie 2 dane
                umieszczone w pamięci począwszy
                od adresu określonego etykietą
                POLE6.
```

3.18 Ekstrakod - odłącz dziurkarkę - ODLDT

Definicja makroinstrukcji:

```
ODLDT  MACRO
        RST      3
        DB       17
        ENDM
```

Opis:

Ekstrakod ODLDT powoduje odłączenie (zwolnienie) dziurkarki taśmy papierowej od zadania, które ten ekstrakod użyło. Jeżeli zadanie nie używało poprzednio ekstrakodu perforowania danych na dziurkarce DZIUR, to ekstrakod ODLDT jest ignorowany. Jeżeli jakiegokolwiek zadanie perforowało dane na dziurkarce ekstrakodem DZIUR, to przed zakończeniem zadania (skokiem do systemowej etykiety KOZAD) należy koniecznie użyć ekstrakodu ODLDT. W przeciwnym wypadku nie będzie można korzystać z dziurkarki w żadnym innym zadaniu (także w zadaniu monitora operatorskiego).

Przykład użycia:

```
ODLDT - odłącz dziurkarkę od aktualnie
        wykonującego się zadania.
```

3.19 Ekstrakod - zainstaluj zadanie - INSTZ

Definicja makroinstrukcji:

```
INSTZ  MACRO  NRZAD, ADRES
        RST    3
        DB     18, NRZAD
        DW     ADRES
        ENDM
```

Opis:

Ekstrakod INSTZ powoduje zainstalowanie w systemie programu, umieszczonego w pamięci począwszy od adresu określonego etykietą użytkownika ADRES, jako systemowego zadania o numerze NRZAD.

Przykład użycia:

```
INSTZ  5,ZAD05 - zainstaluj program zaczynający
                się etykietą ZAD05 jako zadanie
                numer 5
```

3.20 Ekstrakod - zainstaluj przerwanie - INSTP

Definicja makroinstrukcji:

```
INSTP  MACRO  NRPRZ, ADRES
        RST    3
        DB     19, NRPRZ
        DW     ADRES
        ENDM
```

Opis:

Ekstrakod INSTP powoduje zainstalowanie w systemie programu umieszczonego w pamięci począwszy od adresu określonego etykietą użytkownika ADRES jako programu obsługi

przerwania nr NRPRZ. Parametr NRPRZ=0 oznacza obsługę przerwania o najwyższym priorytecie, a odpowiednio NRPRZ=7 o najniższym priorytecie.

Przykład użycia:

```
INSTP 5, PRZER - zainstaluj program zaczynający
                 się etykietą PRZER jako program
                 obsługi przerwania nr 5
```

3.21 Ekstrakod - zainstaluj program sprawdzania warunku INSTW

Definicja makroinstrukcji:

```
INSTW  MACRO  NRWAR, ADRES
        RST    3
        DB     20, NRWAR
        DW     ADRES
        ENDM
```

Opis:

Ekstrakod INSTW powoduje zainstalowanie w systemie programu umieszczonego w pamięci począwszy od adresu określonego etykietą użytkownika ADRES jako programu nr NRWAR sprawdzania spełnienia warunku.

Przykład użycia:

```
INSTW 2, SPRAW - zainstaluj program zaczynający
                 się etykietą SPRAW jako program
                 nr 2 sprawdzający spełnienie
                 warunku.
```

3.22 Ekstrakod - drukuj tekst w trybie alarmowym z zawieszeniem - ALARZ

Definicja makroinstrukcji:

```
ALARZ  MACRO  BUFWY
        RST    3
        DB     21
        DW     BUFWY
        ENDM
```

Opis:

Ekstrakod ALARZ powoduje dopisanie do kolejki wydruków alarmowych tekstu rozpoczynającego się od adresu określonego przez użytkownika. Ekstrakod ALARZ dotyczy druku na podstawowym urządzeniu systemu o nr 0. Na urządzenie wyprowadzane są kolejne znaki z bufora BUFWY aż do znaku NULL (kod ASCII 0) kończącego tekst. Ekstrakod zawiesza wykonywanie zadania używającego go aż do czasu wyprowadzenia na urządzenie ostatniego znaku bufora BUFWY. Ekstrakod ALARZ nie bada zajętości urządzenia 0 przez inne zadanie. Oznacza to, że tekst drukowany tym ekstrakodem jest wyprowadzany natychmiast po zakończeniu aktualnie drukowanego tekstu.

Przykład użycia:

```
ALARZ  TEKST -  pisz na urządzenie 0 poza kolejką
                tekstów dane umieszczone w
                pamięci od określenia etykieta
                TEKST.
```

3.23 Ekstrakod - ustaw długość bufora wejściowego - DLBUF

Definicja makroinstrukcji:

```
DLBUF  MACRO  NRURZ, ILZNA
        RST    3
        DB     22, NRURZ, ILZNA
        ENDM
```

Opis:

Ekstrakod DLBUF ustala maksymalną ilość znaków (ILZNA) wprowadzonych z urządzenia nr NRURZ ekstrakodem CZYTD.

3.24 Ekstrakod - maskuj przerwania - MASKA**Definicja makroinstrukcji:**

```
MASKA  MACRO  DANE
        RST    3
        DB     23, DANE
        ENDM
```

Opis:

Ekstrakod MASKA służy do blokowania/odblokowywania poszczególnych wejść przerywających układu PIC8259. Parametr DANE określa maskę przerwań wpisywaną do układu 8259. Najstarszy bit (D7) parametru DANE dotyczy wejścia IR7 (o najniższym priorytecie), a bit D0 - wejścia IRO o najwyższym priorytecie. Kolejne bity parametru DANE blokują (jeśli bit jest jedynką) lub odblokowują (jeśli bit jest zerem) odpowiadające im zgłoszenia przerwań.

Przykład użycia:

```
MASKA  01100000B  - blokuj przerwanie IR6 i IR5
```


4. Eksploatacja programu monitor operatorski

4.1 Opis ogólny monitora operatorskiego

Program MONITOR jest to monitor operatorski opracowany dla pakietu jednostki centralnej MM80 przeznaczony do pracy pod systemem operacyjnym RTMT. Program ten jest przeznaczony do przygotowywania i uruchamiania zadań użytkowych. Program MONITOR wykonuje typowe dla takich programów polecenia operatora dotyczące pamięci operacyjnej, monitora ekranowego oraz czytnika i dziurkarki.

Program MONITOR pracuje pod kontrolą systemu RTMT jako jedno z zadań użytkownika. Numer zadania MONITOR jest określany na etapie generacji systemu - domyślnie przyjmowany jest numer 2. Sterowanie programem monitora możliwe jest tylko z podstawowego urządzenia operatorskiego (o numerze 0) i może odbywać się w czasie rzeczywistym podczas pracy na obiekcie. Aby program MONITOR mógł rozpocząć pracę, urządzenie nr 0 nie może być zajęte przez żadne z innych zadań. Gdy urządzenie to jest wolne, program monitor można uruchomić znakiem CTRL/C (kod 03). Należy pamiętać, że przez cały czas pracy zadania MONITOR urządzenie to jest zajęte i żadne inne zadanie nie może korzystać z tego urządzenia. Dopuszczone będą tylko wydruki alarmowe (wywoływane ekstrakodem ALARM lub ALARZ).

Po wywołaniu monitor zgłasza się wyświetlając urządzeniu nr 0 tekst:

MONITOR MM-80

>

Znak ">" oznacza oczekiwanie programu na wybranie przez operatora jednej z dyrektyw wraz z jej parametrami. Operator wybiera jedną z dyrektyw pisząc jej jedno- lub dwuliterową nazwę, podaje (jeśli są wymagane) parametry, po czym kończy wprowadzanie danych znakiem powrotu karetki <CR>. Użycie zamiast nazwy dyrektywy znaku kropki <.> zakończonego

znakiem <CR> powoduje zwolnienie urządzenia nr 0, zakończenie programu MONITOR i powrót do koordynatora systemu.

Wykrycie błędu w dyrektywie (np. podanie nazwy nie oznaczającej żadnej z dyrektyw lub podanie w adresie cyfry nie hex) sygnalizowane jest znakiem "?". Natomiast, gdyby operator chciał wpisywać dane lub ustawić pułapkę do obszaru nieistniejącej pamięci albo do pamięci typu FROM następuje wtedy sygnalizacja znakiem "!". Po zasygnalizowaniu błędu program MONITOR ponownie zgłasza swoją gotowość znakiem ">".

Operator może zawsze, zanim zakończy wprowadzać dane znakiem powrotu karetki <CR> wycofać ostatnio wprowadzony znak naciskając klawisz DEL (RUBOUT). Usunięcie znaku sygnalizowane jest cofnięciem kursora. Kolejne naciskanie znaku znaku DEL powoduje usuwanie następnych znaków w kolejności odwrotnej do ich wprowadzania.

4.2 Opis dyrektyw monitora operatorskiego

W podanym niżej opisie poszczególnych dyrektyw programu MONITOR przyjęto następujące oznaczenia.

- spacja, która koniecznie musi wystąpić między parametrami dyrektywy

<CR> - znak powrotu karetki

aaaa - cztery znaki hex określające adres

pppp - cztery znaki hex określające adres początku obszaru pamięci

kkkk - cztery znaki hex określające adres końca obszaru pamięci

ww - dwa znaki hex określające wartość komórki

dd - dwie cyfry dziesiętne

dddd - cztery cyfry dziesiętne

Ponieważ jako podstawowe urządzenie operatorskie może

być zastosowany zarówno monitor ekranowy jak i drukarka dlatego też używane w tekście terminy "drukuję" i "wyświetla" należy traktować zamiennie.

4.2.1 DYREKTYWA O - odczyt i ewentualna zmiana zawartości komórki pamięci

Wywołanie dyrektywy:

```
Oaaaa<CR>
```

gdzie: aaaa - adres komórki.

W odpowiedzi program MONITOR wyświetla zawartość tej komórki poprzedzonej jej adresem i oczekuje (bez sygnalizacji znakiem ">") na dalsze dyspozycje operatora. Naciśnięcie znaku powrotu karetki <CR> spowoduje wyświetlenie zawartości komórki o adresie o 1 większym od poprzedniej, naciśnięcie znaku minus <-> i <CR> spowoduje wyświetlenie zawartości komórki o adresie o 1 mniejszym niż poprzednio. Aby zakończyć dyrektywę należy podać znaki <.> i <CR>, zgłosi się wtedy ponownie program MONITOR znakiem ">".

Chcąc zmienić zawartość aktualnie wyświetlanej komórki, należy podać, bezpośrednio po jej dotychczasowej zawarości, dwa znaki hex, tworzące jej nową zawartość. Następnie należy wybrać dalsze działanie dyrektywy w sposób identyczny jak podano wyżej tzn. odczyt następnej lub poprzedniej komórki albo zakończenie dyrektywy.

Program sprawdza, czy podana nowa zawartość komórki pamięci została poprawnie wpisana do pamięci i każdą niezgodność sygnalizuje znakiem <!>. Użycie znaku nie będącego cyfrą hex w adresie lub w nowej zawartości komórki albo podanie znaku różnego od <->, <.> lub <CR> powoduje sygnalizację błędu znakiem <?>, po czym następuje ponownie oczekiwanie na wprowadzenie danych.

4.2.2 DYREKTYWA M - odczyt zawartości obszaru pamięci

Wywołanie dyrektywy:

Mpppp kkkk<CR>

W odpowiedzi program wyświetla nagłówek, który stanowią ostatnie cyfry adresu komórek pamięci, a następnie w wierszach po 16 komórek w jednym wierszu ich zawartości. Każdy wiersz jest poprzedzony adresem pierwszej komórki w danym wierszu. Błędne wywołanie dyrektywy tj. użycie cyfry nie hex w którymkolwiek z adresów lub podanie adresu końca mniejszego od adresu początku powoduje sygnalizację błędu znakiem <?>, po czym następuje zgłoszenie MONITORA znakiem ">". Przy programowym wywołaniu dyrektywy na ekranie wyświetlana jest zawartość całego zadeklarowanego obszaru pamięci z komórkami o adresach pppp i kkkk włącznie jako dwucyfrowe liczby heksadecymalne rozdzielone znakami spacji.

4.2.3 DYREKTYWA V - zapis stałej wartości do obszaru pamięci

Wywołanie dyrektywy:

Vpppp kkkk ww<CR>

gdzie ww - wartość wpisywana do wszystkich komórek pamięci od adresu pppp do kkkk włącznie. Po wypełnieniu podaną wartością komórek pamięci dyrektywa kończy swoje działanie i powraca do programu MONITOR, który zgłasza swoją gotowość znakiem ">". Gdy w którymkolwiek z adresów lub w wartości wpisywanej do komórek użyta zostanie cyfra nie hex, lub adres początku będzie większy od adresu końca, nastąpi sygnalizacja błędu znakiem <?>, po czym zgłosi się program MONITOR.

Dyrektywa sprawdza przed zapisaniem każdej komórki, czy jest to pamięć typu RAM. Gdy tak nie jest, sygnalizuje błąd znakiem <!> i adresem wykrytej niepoprawnej komórki, po czym przechodzi do programu MONITOR.

4.2.4 DYREKTYWA Z - zapis obszaru pamięci

Wywołanie dyrektywy:

Zaaaa<CR>

gdzie aaaa - adres pierwszej zapisywanej komórki. Jeżeli w adresie zostanie użyty znak nie będący cyfrą heksadecymalną wyświetlony będzie znak <?> i zgłosi się program MONITOR. Gdy adres jest poprawny, następuje sprawdzenie, czy komórka o podanym w wywołaniu adresie należy do pamięci RAM. Gdy nie, to następuje sygnalizacja błędu znakiem <!> i powrót do MONITORA.

Jeżeli nie zostanie stwierdzony błąd, drukowany jest nagłówek podobnie jak w dyrektywie M. Dane należy wpisywać w kolejnych wierszach pod nagłówkiem. Każdy wiersz jest poprzedzony adresem pierwszej występującej w nim komórki. Poszczególne dane wpisywane do kolejnych komórek należy oddzielać znakiem spacji. Analiza wprowadzonych danych i zapis do komórek dokonywane są dopiero po naciśnięciu znaku <CR>. Dlatego zaleca się, po wypełnieniu całego wiersza, kończyć go znakiem <CR>, mimo że dalsze wpisywanie poprawnych danych (tzn. grup po dwie cyfry hex rozdzielonych spacją) nie będzie uważane za błąd.

Jeżeli podczas podawania danych popełniony zostanie błąd (zostanie użyta cyfra nie hex), to podczas analizy wprowadzonego tekstu będą zignorowane wszystkie przeznaczone do wpisania wartości, począwszy od pierwszej komórki, dla której wystąpił błąd aż do znaku <CR>. Następnie wydrukowany będzie w nowej linii adres pierwszej komórki wiersza, w którym został popełniony błąd. Jeżeli operator przystąpi do dalszego wprowadzania danych, to będą one zapisywane począwszy od komórki błędnie zapisanej.

Przed wpisaniem nowej wartości do komórki, sprawdzone jest, czy ta komórka należy do pamięci typu RAM. Jeżeli nie

to sygnalizowany jest błąd komunikatem:

aaaa !

gdzie aaaa - adres pierwszej, dla której stwierdzono błąd, po czym zgłasza się program MONITOR znakiem ">".

Wprowadzanie danych kończy się naciskając w dowolnym momencie znak kropki <.>, a po nim znak powrotu karetki <CR>. Nastąpi wtedy powrót do programu MONITOR.

4.2.5 DYREKTYWA R - przesun blok danych

Wywołanie dyrektywy:

Rpppp kkkk aaaa<CR>

Dyrektywa pobiera zawartość komórek z obszaru pamięci określonego adresami pppp i kkkk włącznie i ładuje je do klejnych komórek pamięci począwszy od adresu aaaa. Zakończenie dyrektywy powoduje powrót do programu MONITOR i ogłoszenie jego gotowości znakiem ">".

Sygnalizacja błędów jest taka sama jak w dyrektywie V.

4.2.6 DYREKTYWA L - czytanie taśmy papierowej

Wywołanie dyrektywy:

L<CR>

Dyrektywa służy do czytania taśmy papierowej i zapisu odczytanej informacji do pamięci RAM. Dyrektywa bada czy znaki są heksadecymalne, czy jest poprawna suma kontrolna w rekordach, czy dane na taśmie papierowej mają właściwy format oraz czy informacja została poprawnie wpisana do pamięci. Gdy cała taśma zostanie poprawnie wczytana, drukowany jest komunikat:

OK

po czym zgłosi się program MONITOR znakiem ">".

W przypadku, gdy zostanie stwierdzony jakikolwiek błąd nastąpi sygnalizacja komunikatem

n BŁAD CT

gdzie n cyfra od 1 do 6.

Jedynie gdy stwierdzona zostanie niezgodność informacji wpisanej do komórki pamięci z poprawnie odczytaną, błąd sygnalizowany jest komunikatem:

aaaa !

gdzie aaaa oznacza adres pierwszej komórki, dla której wystąpił błąd.

Po zasygnalizowaniu błędu dyrektywa kończy swoje działanie i zgłasza się znakiem ">" program MONITOR.

W komunikacie o błędzie

- n = 1 oznacza, że wystąpił koniec taśmy zanim nastąpił ostatni rekord informacji lub czytnik został wyłączony z sieci w trakcie czytania informacji
- n = 2 oznacza, że odczytany został znak o nieparzystej liczbie bitów
- n = 3 oznacza, że wystąpiła zła suma kontrolna w rekordzie
- n = 4 oznacza, że w rekordzie wystąpił znak nie będący cyfrą hex
- n = 5 oznacza, że stwierdzony zły typ rekordu (różny od 0 dla rekordów informacyjnych)
- n = 6 oznacza, że nastąpiło przekroczenie czasu, tzn. czytnik w ciągu 2 sek. nie odczytał żadnego znaku. Może to być spowodowane np. zacięciem się taśmy lub nie włączeniem zasilania czytnika.

Dane na wczytywanej taśmie powinny mieć format zgodny z formatem Intellec Hex.

Istnieje możliwość wczytywania danych z taśmy papierowej do innego obszaru pamięci niż w rekordach na taśmie. W tym celu należy wywołać dyrektywę za pomocą komunikatu :

Laaaa <CR>

gdzie aaaa oznacza przesunięcie danych, tzn. jest to wartość dodawana do każdego adresu ładowanie informacji odczytanej z taśmy. Komunikaty są identyczne jak w dyrektywie L bez przesunięcia.

4.2.7 DYREKTYWA S - sprawdzanie taśmy papierowej

Wywołanie dyrektywy:

S <CR>

Dyrektywa porównuje informację zapisaną na taśmie papierowej z zawartością pamięci. Diagnostyka taśmy papierowej jest identyczna jak w dyrektywie L. Jedynie zamiast wpisywania informacji do pamięci porównywana jest wartość odczytana z taśmy z zawartością komórki pamięci i w przypadku niezgodności drukowany jest komunikat:

aaaa !

po czym zgłasza się znakiem ">" program MONITOR. aaaa oznacza adres pierwszej komórki dla, której stwierdzono błąd. Pozostałe komunikaty o błędach są takie same jak w dyrektywie L. Gdy dla całej taśmy nie zostanie stwierdzony żaden błąd drukowany jest komunikat

OK

po czym zgłasza się program MONITOR.

Podobnie jak w dyrektywie L istnieje możliwość porównywania zawartości taśmy z pamięcią o adresach przesuniętych o wartość aaaa w stosunku do adresów ładowania na taśmie papierowej. W tym przypadku dyrektywę należy uruchomić podając tekst:

Saaaa <CR>

4.2.8 DYREKTYWA D - dziurkowanie taśmy papierowej

Wywołanie dyrektywy:

- a) Dpppp kkkk.<CR>
- b) Dpppp kkkk <CR>

Dyrektywa służy do wprowadzania zawartości pamięci na taśmę papierową. Informacja na taśmie jest dziurkowana zgodnie z formatem Intellec Hex.

Używana w wywołaniu dyrektywy, w którymkolwiek z adresów, cyfry nie hex lub podanie adresu początku wyprowadzanego obszaru większego od adresu końca powoduje sygnalizację błędu znakiem "?" i powrót do programu MONITOR, który zgłasza się znakiem ">". Podobne działanie nastąpi, gdy po ostatnim znaku określającym adres końca wprowadzanego obszaru podany będzie znak różny od "." lub <CR>.

Jeżeli wywołanie dyrektywy nastąpiło poprzez podanie tekstu a) to po wyprowadzeniu na taśmę zawartości obszaru określonego adresami pppp i kkkk, zostanie wydziurkowany rekord końca, a następnie pewien odcinek pustej taśmy, po czym zgłosi się znakiem ">" program MONITOR.

Jeżeli dyrektywa była wywołana tekstem b), to po wydziurkowaniu zawartości zadeklarowanego obszaru dyrektywa przechodzi do oczekiwania na podanie przez operatora, w nowej linii, dalszych dyspozycji. Operator ma do wyboru następujące możliwości:

- c) pppp kkkk <CR>
- d) pppp kkkk.<CR
- e) . CR

Podanie tekstu c) spowoduje wyprowadzenie na taśmę papierową kolejnego określonego adresami pppp i kkkk obszaru pamięci i ponowne oczekiwanie na dalsze dyspozycje operatora. Gdy został podany tekst d), to po wydziurkowaniu zawartości zadeklarowanego adresami pppp i kkkk obszaru pamięci zostanie wyprowadzony rekord kończący dane na taśmie, następnie pewien odcinek pustej taśmy, po czym nastąpi zgłoszenie MONITORA znakiem ">". Podanie tekstu e) spowoduje

wyprowadzenie tylko rekordu końca, odcinka pustej taśmy i zakończenie dyrektywy.

Jeżeli, w którymkolwiek z tekstów c), d), e) zostanie stwierdzony błąd, to nastąpi jego sygnalizacja znakiem "?", po czym dyrektywa oczekiwac będzie na poprawne dyspozycje operatora.

Jeżeli w trakcie dziurkowania taśmy zostanie stwierdzona niepoprawna praca dziurkarki, wyświetlany jest komunikat:

n BŁAD DT

gdzie n = 1 oznacza, że dziurkarka jest nie włączona do sieci lub, że skończyła się taśma

n = 2 oznacza przekroczenie czasu tzn. , że dziurkarka nie zgłosiła gotowości przez około 3 sekundy

Po wyświetleniu powyższego komunikatu następuje zakończenie dyrektywy i zgłasza się program MONITOR.

4.2.9 DYREKTYWA K - kopiowanie taśmy

Wywołanie dyrektywy:

K<CR>

Dyrektywa służy do kopiowania taśmy o dowolnym formacie. Po wywołaniu dyrektywy taśma czytana przez czytnik będzie kopiowana na dziurkarce aż do końca taśmy w czytniku, a następnie zgłosi się znakiem ">" program MONITOR. W razie stwierdzenia niepoprawnej pracy czytnika lub dziurkarki zostanie wydrukowany komunikat:

 n BŁAD CT gdzie n = 1 lub 6
lub n BŁAD DT gdzie n = 1 lub 2

po czym wykonywanie dyrektywy zostanie zakończone i zgłosi się program MONITOR. Znaczenie błędów jest takie samo jak podane w opisie dyrektyw L i D.

4.2.10 DYREKTYWA W - uruchomienie programu

Wywołanie dyrektywy:

```
Waaaa<CR>
```

gdzie aaaa jest adresem startu uruchamianego programu. Jeżeli w adresie będzie użyty znak nie będący cyfrą heksadecymalną, to wyświetlony zostanie znak <?> i nastąpi powrót do MONITORA. Użycie dyrektywy powoduje, że system pracuje w modzie jednozadaniowym tzn. wykonywany jest tylko uruchomiony przez tę dyrektywę program.

Zadne inne zadania nie będą uruchamiane, ani wznowione jeżeli nie zostanie zakończony uruchomiony program. W programie tym mogą być używane wszystkie ekstrakody systemowe i musi się on kończyć skokiem do adresu określonego etykietą systemową KOZAD.

4.2.11 DYREKTYWA J - instalowanie zadania

Wywołanie dyrektywy:

```
Jdd aaaa<CR>
```

Dyrektywa powoduje przypisanie zadaniu o numerze określonym parametrem dd adresu startu aaaa. Błąd jest sygnalizowany znakiem <?> gdy w numerze instalowanego zadania zostanie użyty znak nie będący cyfrą dziesiętną lub w adresie startu zadania użyty będzie znak nie będący cyfrą heksadecymalną. Po zasygnalizowaniu błędu nastąpi powrót do programu MONITOR. Nieznaczące zero w numerze instalowanego zadania można pominąć.

Po wykonaniu dyrektywy zgłasza się znakiem ">" program MONITOR.

4.2.12 DYREKTYWA F - usunięcie zadania

Wywołanie dyrektywy:

Fdd<CR>

gdzie dd - dwie cyfry dziesiętne określające nr zadania. Nieznaczące zero można pominąć. Użycie znaku różnego od cyfry dziesiętnej powoduje sygnalizację błędu znakiem <?> i powrót do MONITORA.

Dyrektywa powoduje usunięcie zadania ze wszystkich tablic i kolejek systemowych, zwalnia urządzenia zajęte przez to zadanie. Dyrektywa nie usuwa zadania z tabeli startów zadań. Chcąc usunąć zadanie z tej tabeli należy je zainstalować z adresem startu określonym dyrektywą KOZAD (0083H).

Jeżeli usuwane zadanie aktualnie jest zawieszona, to nie zostanie ono już wznowione. Po wykonaniu dyrektywy zgłasza się znakiem ">" program MONITOR.

4.2.13 DYRETYWA Y - wyświetlenie informacji o systemie

Wywołanie dyrektywy:

Y <CR>

Użycie dyrektywy powoduje wyświetlenie na ekranie monitora informacji o zainstalowanych w systemie zadaniach i dostępnej pamięci zewnętrznej. W kolejnych wierszach wyświetlane są adresy zadań zainstalowanych w systemie oraz adresy grupowe (z dokładnością do 4k pamięci) wykrytej w systemie pamięci RAM, a potem PROM. Nie są wyświetlane (jako, że zawsze istnieją) adresy 0 i 1 (PROM pakietu MM80), 3 (RAM pakietu MM80) oraz E (adresy urządzeń obiektowych). Po zakończeniu działania następuje powrót do programu MONITOR.

4.2.14 DYREKTYWA G - uruchomienie zadania

Wywołanie dyrektywy:

Gdd <CR>

Dyrektywa powoduje zgłoszenia do wykonania zadania o numerze określonym parametrem dd. Muszą to być dwie cyfry dziesiętne (nieznaczące zero można pominąć). Jeżeli, któryś ze znaków nie jest cyfrą dziesiętną następuje sygnalizacja błędu znakiem "?" i powrót do programu MONITOR.

W przypadku poprawnego wywołania, po dopisaniu zadania do kolejki zadań oczekujących na uruchomienie następuje zwolnienie urządzenia nr 0 przez zadanie MONITOR i zadanie MONITOR kończy się. Zadanie MONITOR można ponownie uruchomić (pod warunkiem, że urządzenie nr 0 nie jest zajęte) znakiem CTRL/C.

4.2.15 DYREKTYWA B - ustawienie pułapki (break-point)

Wywołanie dyrektywy:

B1 aaaa<CR>

lub B2 aaaa<CR>

Dyrektywa B służy do ustawiania pułapki w programie użytkownika umieszczonym w pamięci RAM. Program MONITOR umożliwia jednoczesne operowanie dwoma pułapkami (o numerach 1 i 2). Po ustawieniu jednej lub obu pułapek dyrektywą B należy spowodować uruchomienie programu dyrektywą G. Jeżeli w trakcie wykonywania program dojdzie do adresu, pod którym została uprzednio ustawiona jedna z pułapek jego działanie zostanie wstrzymane i na ekranie monitora zostanie wyświetlony numer pułapki, kod instrukcji na której został zatrzymany program oraz zawartość wszystkich rejestrów i

wskaźników procesora. Kod instrukcji oraz zawartość rejestrów jest wyświetlana w postaci liczb heksadecymalnych. Następnie MONITOR zgłasza gotowość znakiem >. W tym momencie operator ma do swojej dyspozycji wszystkie ogólnie dostępne dyrektywy MONITOR-a oraz dodatkowo dyrektywy C,U,X oraz N, które mogą być używane tylko wtedy, gdy program użytkownika jest zatrzymany w pułapce. Zmiana adresu ustawienia tej pułapki, na której zatrzymany jest program (ponowne użycie dyrektywy B) powoduje usunięcie pułapki ustawionej uprzednio, ponowne ustawienie pułapki w nowym miejscu i wznowienie programu użytkownika od przerwane go miejsca. Zmiana adresu ustawienia pułapki, na której program nie jest zatrzymany (lub ustawienie nowej pułapki gdy poprzednio nie była ona używana) nie powoduje kontynuacji zatrzymanego programu.

Próba ustawienia pułapki dyrektywą B w pamięci PROM sygnalizowana jest znakiem <!>, po czym ponownie zgłasza się program MONITOR. Użycie w wywołaniu dyrektywy znaku nie będącego cyfrą heksadecymalną powoduje sygnalizację błędu znakiem <?> i ponownie zgłasza się program MONITOR.

Użycie przycisku RESET powoduje usunięcie obu pułapek.

4.2.16 DYREKTYWA X - odczyt i ewentualna zmiana zawartość rejestru

Wywołanie dyrektywy:

- a) X<CR>
- b) Xr<CR>

gdzie r - nazwa rejestru.

Dyrektywa X może być używana jed uruchamiany program zatrzymany jest w pułapce (użyta była dyrektywa B a potem dyrektywa G. Dyrektywa X umożliwia wyświetlenie zawartości wszystkich rejestrów procesora w zatrzymanym pułapką programie użytkownika (wywołanie a), lub wyświetlenie i ewentualną zmianę zawartości pojedynczych rejestrów przerwane go programu (wywołanie b).

Wyświetlanie zawartości wszystkich rejestrów przerwane go programu jest powtórzeniem komunikatu, jaki wyświetla program MONITOR gdy uruchamiany program zatrzymuje się w pułapce ustawionej przez operatora dyrektywą B. Ta opcja dyrektywy X nie umożliwia zmiany zawartości poszczególnych rejestrów.

W wywołaniu b) dyrektywy X jako nazwę rejestru należy podać jedną z liter A,B,C,D,E,H lub L dla odczytu i ewentualnej zamiany zawartości pojedynczego rejestru, literę F dla flag procesora, literę P dla licznika rozkazów PC lub literę S dla wskaźnika stosu SP. Po podaniu jednoliterowej nazwy rejestru na ekranie wyświetla się w nowym wierszu zawartość tego rejestru (dwa znaki heksadecymalne dla rejestru i flag procesora lub cztery znaki heksadecymalne dla rejestrów PC i SP). W tym momencie operator może zakończyć dyrektywę znakiem kropki <.> i znakiem <CR> lub wpisać nową wartość (dwa znaki heksadecymalne dla rejestrów i flag procesora lub cztery znaki heksadecymalne dla rejestrów PC i SP) i zakończyć ją znakiem powrotu karetki <CR>. Zakończenie dyrektywy powoduje powrót do programu MONITOR i zgłoszenie jego gotowości znakiem >.

4.2.17 DYREKTYWA U - usunięcie pułapki

Wywołanie dyrektywy:

U<CR>
lub U1<CR>
lub U2<CR>

Dyrektywa U umożliwia usunięcie wybranej pułapki (wywołanie U1 lub U2) lub obu pułapek (wywołanie U) i kontynuację zatrzymanego programu o ile stał on w aktualnie usuniętej pułapce. Dyrektywa ta może być wywołana jedynie wtedy, gdy program stoi w ustawionej wcześniej pułapce. Usunięcie pułapki, w której program aktualnie nie jest

zatrzymany nie powoduje jego kontynuacji.

4.2.18 DYREKTYWA C - kontynuacja programu bez przesunięcia pułapki

Wywołanie dyrektywy:

C<CR>

Dyrektywa C służy do kontynuacji programu zatrzymanego w pułapce bez zmiany miejsca ustawienia pułapki. Może ona być użyta jedynie wtedy, gdy program stoi na ustawionej wcześniej pułapce.

4.2.19 DYREKTYWA N - praca krokowa

Wywołanie dyrektywy:

N<CR>

Dyrektywa N umożliwia automatyczne przesuwania ustawionej wcześniej pułapki o jeden rozkaz. Użycie dyrektywy N jest możliwe jedynie w przypadku, gdy program użytkownika po uruchomieniu dyrektywą G doszedł do poprzednio ustawionej pułapki. Uruchomienie programu użytkownika od adresu, w którym była umieszczona pułapka następuje automatycznie bezpośrednio po naciśnięciu litery N i znaku <CR>. Jeżeli w wyniku użycia dyrektywy N pułapka ma być przesunięta na obszar PROM jest sygnalizowany błąd wyświetleniem znaku zapytania '<?>' i dyrektywa N jest ignorowana. Dyrektywa N nie rozróżnia numerów pułapek - jedynie w przypadku gdy przesuwana dyrektywą N pułapka dochodzi do wcześniej ustawionej drygiej pułapki dyrektywa jest ignorowana i sygnalizowany jest błąd znakiem zapytania '<?>'.
Dyrektywa N przesuwa ustawioną pułapkę zgodnie z

wykonaniem się rzeczywistego programu za każdym razem o jeden rozkaz (o 1,2 lub 3 komórki w zależności od długości następnego rozkazu). Dla rozkazów zmieniających sterowanie (skoki, skoki do podprogramów, PCHL itd) pułapka przesuwana jest do następnego rozkazu zgodnie z obliczonym na podstawie wskaźników i rejestrów procesora sterowaniem. Jedynie rozkaz RSTn jest zawsze traktowany jak rozkaz nie zmieniający sterowania to znaczy nowa pułapka jest ustawiona za rozkazem RST, a nie w komórkach 0,8,10H itd. Rozkaz ekstrakodu systemowego RST3 jest w dyrektywie N traktowany jak jeden rozkaz niezależnie od typu ekstrakodu tzn. następna pułapka ustawiona jest za ekstrakodem.

4.2.20 DYREKTYWA P - włączenie/wyłączenia zadania okresowego

Wywołanie dyrektywy:

Pnr dd <CR>
lub Pnr<CR>

Dyrektywa powoduje zgłoszenie zadania o numerze nr do wykonywania okresowo co dd sekund. Pierwsze uruchomienie zadania nastąpi po upływie dd sekund od momentu zakończenia dyrektywy. Jeżeli zadanie było wcześniej zgłoszone do wykonywania okresowo, to nowe użycie dyrektywy P z parametrem dd spowoduje zmianę okresu wykonywania zadania. Użycie dyrektywy P bez parametru dd powoduje wyłączenie zadania okresowego. Gdy w momencie użycia dyrektywy zadanie jest w trakcie wykonywania (zawieszona), to wykona się ono do końca.

Użycie w wywołaniu dyrektywy znaku nie będącego cyfrą dziesiętną powoduje sygnalizację błędu znakiem "?" i powrót do programu MONITOR.

Nieznaczące zero w numerze uruchamianego zadania można pominąć.

4.2.21 DYREKTYWA Q - czytanie programu przez kanał V-24

Wywołanie dyrektywy:

Qx [dddd]<CR>

gdzie x numerem kanału V-24, a dddd jest opcjonalnym przesunięciem wczytywanego programu. Dyrektywa Q umożliwia wprowadzanie do pamięci RAM systemu programu zgodnego z formatem Intellec Hex poprzez dowolny z zainstalowanych kanałów V-24. Numer kanału podaje operator bezpośrednio po literze Q jako jedną cyfrę z zakresu 0-4. Kanał 0 oznacza kanał V-24 pakietu jednostki centralnej MM80. Kanały 1-4 oznaczają kolejne kanały V-24 na pakietach MI24. Parametr dddd oznacza opcjonalne przesunięcie ładowanego programu w stosunku do adresów ładowania znajdujących się w przesyłanych rekordach.

Przesyłanie danych poprzez kanał V24 polega na uruchomieniu dyrektywy Q, a dopiero potem rozpoczęciu wysyłania danych przez inny komputer. Użycie do czytania danych kanału 0 z definicji wymaga zmiany złącza C pakietu MM80 przed rozpoczęciem nadawania i powtórnego przełożenia tego złącza po jego zakończeniu.

Dyrektywa kontroluje, czy odbierane dane są znakami heksadecymalnymi i sprawdza sumę kontrolną, odbieranych rekordów, a ewentualne błędy sygnalizuje odpowiednimi komunikatami. Poprawne odczytanie danych sygnalizowane jest wyświetleniem komunikatu OK (dla kanału 0 dopiero po ponownej zamianie złącza C) po czym zgłasza się program MONITOR.

4.2.22 DYREKTYWA H - włączenie zadania o godzinie

Wywołanie dyrektywy:

Hnr gg.mm

Dyrektywa powoduje zgłoszenie do wykonania zadania o numerze nr, gdy w licznikach systemowych czasu rzeczywistego pojawi się godzina gg.mm (pierwsze dwa znaki określają godzinę, następne minuty). Wszystkie cyfry muszą być dziesiętne, a godziny i minuty muszą być rozdzielone znakiem kropki ".". Jeżeli tak nie jest, to następuje sygnalizacja błędu znakiem "?" i powrót do MONITORA. Sygnalizacja błędu znakiem "?" nastąpi również, gdy podana zostanie godzina większa niż 23 i minuty większe niż 59.

Nieznaczące zero w numerze włączanego zadania można pominąć. Po wykonaniu dyrektywy nastąpi powrót do MONITORA.

4.2.23 DYREKTYWA T - zmiana czasu lub daty w systemie

Wywołanie dyrektywy:

a) TTgg.mm<CR>

b) TDdd.mm.rr<CR>

Dyrektywa T umożliwia odczytanie i ewentualną zmianę czasu w systemie (wywołanie a) lub odczyt i ewentualną zmianę daty w systemie (wywołanie b). Odczyt czasu i daty bez możliwości zmiany wykonuje się wywołując dyrektywę bez podania parametrów: TT<CR> lub TD<CR>. Czas musi być podawany w formacie dwie cyfry dla godzin, znak rozdzielający i dwie cyfry dla minut. Natomiast data musi być podawana w formacie dwie cyfry dla dnia, znak rozdzielający, dwie cyfry dla miesiąca, znak rozdzielający i dwie cyfry dla roku. W wywołaniu dyrektywy mogą być używane tylko cyfry dziesiętne. Każdy błąd sygnalizowany jest

znakiem "?" i powrotem do programu MONITOR.

Prawidłowe działanie dyrektywy kończy się zgłoszeniem przez program MONITOR gotowości znakiem ">".

4.2.23 DYREKTYWA I - odczyt/zapis pakietu obiektowego lub bramy I/O

Wywołanie dyrektywy:

- a) INaa[.]<CR>
- b) IOaa dd[.]<CR>
- c) IRaaaa[.]<CR>
- d) IWaaa dddd[.]<CR>

Dyrektywa I umożliwia odczyt (wywołanie a), zapis (wywołanie b) bramy I/O lub odczyt (wywołanie c) albo zapis (wywołanie d) pakietu obiektowego 16-bitowego adresowanego jako pamięć.

W dyrektywach odczytu, jako parametr, podaje się adres bramy (2 cyfry heksadecymalne) lub adres pakietu (4 cyfry heksadecymalne) zakończony znakiem kropki <.> i znakiem <CR> lub samym znakiem <CR>. Wywołanie z kropką powoduje jednorazowy odczyt bramy/pakietu i wyświetlenie odczytanych danych. Wywołanie bez kropki powoduje odczyt bramy/pakietu, wyświetlenie danych i oczekiwanie na decyzję operatora: podanie znaku <CR> powtarza komunikację z tym samym adresem, natomiast kropka i znak <CR> kończą dyrektywę.

W dyrektywach zapisu, jako parametr, podaje się adres bramy (2 cyfry heksadecymalne) lub adres pakietu (4 cyfry heksadecymalne), znak spacji i dane do wysłania (2 cyfry heksadecymalne dla dyrektywy IO, 4 cyfry heksadecymalne dla dyrektywy IW) zakończone znakiem kropki <.> i znakiem <CR> lub samym znakiem <CR>. Wywołanie z kropką powoduje jednorazowy zapis danych do bramy/pakietu. Wywołanie bez kropki powoduje wysłanie danych do bramy/pakietu i

oczekiwanie na decyzję operatora. Operator może po znaku myślnika podać:

- znak kropki <.> i znak <CR> w celu zakończenia dyrektywy;
- sam znak <CR> w celu wysłania tych samych danych pod ten sam adres;
- nowe dane zakończone znakiem <CR>.

5. Podprogramy systemowe dostępne dla użytkownika

W opisie omawianych podprogramów, jako adresy startów podawane są etykiety systemowe. Wszystkie podprogramy (z wyjątkiem instrukcji mikroprocesora RST 2, RST 4, RST 5, RST6 i RST 7) wywoływane są instrukcją CALL.

RST 2 - podprogram czytania bramy I/O.

Przed wywołaniem podprogramu należy umieścić w rejestrze L adres bramy I/O. Dane są czytane do rejestru A (akumulatora). Podprogram nie zmienia zawartości rejestrów (oprócz A).

RST 4 - podprogram opóźnienia programowego.

Przed wejściem do podprogramu należy umieścić w rejestrze B jednostkę opóźnienia (1 dla jednostki 1 ms i wartość różną od 1 dla jednostki 100 ms), a w rejestrze C ilość jednostek. Wyjście z RST4 następuje po zadanych w rejestrach B,C czasie. Podprogram nie zmienia zawartości rejestrów D, E, H, L.

Przykład: wykonanie ciągu instrukcji

MVI B,80H ; jednostka 100m

MVI C,1AH ; 26 jednostek

RST4

powoduje opóźnienie programowe 2,6 sekundy

RST 5 - podprogram czytania pakietu 16-bitowego.

Przed wywołaniem podprogramu w parze rejestrów H, L należy umieścić adres pakietu. Po wyjściu z podprogramu informacja odczytana z pakietu znajduje się w parze rejestrów D, E. Podprogram nie zmienia zawartości rejestrów B, C, H, L.

RST6 - podprogram pisania do pakietu 16-bitowego.

Przed wywołaniem podprogramu w parze rejestrów H, L należy umieścić adres pakietu, a w parze rejestrów D, E - wpisywaną informację. Podprogram nie zmienia zawartości rejestrów B, C, D, E, H, L.

RST Z - podprogram pisania do bramy I/O.

Przed wywołaniem podprogramu należy umieścić w rejestrze L adres bramy I/O a w rejestrze A (akumulatorze) dane do wysłania. Podprogram nie zmienia zawartości rejestrów.

SUMA - dodawanie zawartości akumulatora do pary rejestrów H,L.

Podprogram nie zmienia zawartości B, C, D, E

HXAS 1 - zamiana 4 młodszych bitów akumulatora na kod ASCII cyfry heksadecymalnej w akumulatorze. Rejestry niezmienniane B, C, D, E, H, L.

HXAS 2 - zamiana zawartości akumulatora na kody ASCII dwóch cyfr heksadecymalnych w pamięci.

Przed wywołaniem podprogramu, w parze rejestrów B,C należy umieścić adres pamięci RAM, do której mają być wpisane kody ASCII. Do pierwszej komórki pamięci wpisywany jest kod ASCII czterech starszych bitów akumulatora, a do następnej komórki wpisywany jest kod ASCII czterech młodszych bitów. Na wyjściu z podprogramem zawartość pary rejestrów B, C jest zwiększona o 2. Rejestry niezmienniane D, E, H, L.

HXAS4 - zamiana zawartości pary rejestrów HL na kody ASCII czterech cyfr heksadecymalnych w pamięci.

Przed wywołaniem podprogramu w parze rejestrów BC należy umieścić adres pamięci RAM, do której mają być wpisane kody ASCII. Do pierwszej komórki pamięci wpisywany jest kod ASCII czterech starszych bitów rejestru H. Do komórki pamięci o adresie BC+3 wpisywany jest kod ASCII czterech młodszych

bitów rejestru L. Na wyjściu z podprogramu zawartość pary rejestrów BC jest zwiększana o 4. Rejestry niezmienniane: D, E, H, L.

ASHX - zamiana cyfry heksadecymalnej w kodzie ASCII z akumulatora na wartość binarną w akumulatorze.

Jeżeli na wejściu do podprogramu nie znajduje się w akumulatorze kod ASCII cyfry heksadecymalnej, to sygnalizowany jest błąd poprzez ustawienie flagi CY=1 w słowie stanu procesora. W przeciwnym przypadku CY=0. Podprogram nie zmienia zawartości rejestrów B, C, D, E, H, L

DWHX - zamiana dwóch cyfr heksadecymalnych w kodach ASCII z kolejnych komórek pamięci na liczbę binarną w akumulatorze.

Na wejściu podprogramu, w parze rejestrów B,C należy ustawić adres pierwszej komórki pamięci. Jeżeli, któryś ze znaków ASCII nie reprezentuje cyfry heksadecymalnej, to CY=1. W przeciwnym przypadku CY=0, a zawartość pary rejestrów B,C jest zwiększona o 2. Rejestry niezmienniane: D, H, L.

CZHX - zamiana czterech cyfr heksadecymalnych o kodach ASCII z kolejnych komórek pamięci na dwie liczby binarne w rejestrach D, E.

Na wejściu podprogramu, w parze rejestrów B, C należy ustawić adres pierwszej komórki pamięci. Jeżeli którykolwiek ze znaków ASCII nie reprezentuje cyfry heksadecymalnej, to CY=1. W przeciwnym przypadku CY=0, a zawartość pary rejestrów B, C jest zwiększona o 4. Rejestry niezmienniane: H, L.

DZHEX - zamiana dziesiętnej cyfry w kodzie ASCII z komórki pamięci na wartość binarną w akumulatorze.

Przed wywołaniem podprogramu w parze rejestrów B, C należy ustawić adres komórki pamięci, w której znajduje się kod ASCII zamienianej cyfry. Jeżeli zawartość komórki pamięci

nie reprezentuje cyfry dziesiętnej, to następuje sygnalizacja poprzez ustawienie flagi CY=1 w słowie stanu procesora. Rejestry niezmienniane: B, C, D, E, H, L.

DWDZ - zamiana jednej lub dwóch cyfr dziesiętnych w kodzie ASCII z kolejnych komórek pamięci, na wartość binarną w akumulatorze.

Przed wywołaniem podprogramu w parze rejestrów B, C należy ustawić adres pierwszej komórki pamięci. Jeżeli jej zawartość nie reprezentuje cyfry dziesiętnej, to CY=1, a zawartość B,C jest zwiększona o 1. Jeżeli zawartość drugiej komórki pamięci nie reprezentuje cyfry dziesiętnej, to CY=0, a B, C jest zwiększona o 1 (w stosunku do wartości na wejściu podprogramu). Jeżeli w obu komórkach pamięci są cyfry dziesiętne, to CY=0, a B, C jest zwiększane o 2. Rejestry niezmienniane: H, L.

DZBCD - zamiana dwóch cyfr dziesiętnych w kodzie ASCII z kolejnych komórek pamięci na liczbę binarną w kodzie BCD w akumulatorze.

Działanie podprogramu jest analogiczne do DWHX. Rejestry niezmienniane: H, L.

POLI - porównanie zawartości pary rejestrów D,E z zawartością pary rejestrów H, L.

Zawartość par rejestrów DE i HL traktowana jest jako liczby całkowite bez znaku. Wynik porównania sygnalizowany jest na flagach CY i Z w słowie stanu procesora:

gdy $D,E = H,L$ to CY=0 Z=1

gdy $D,E < H,L$ to CY=1 Z=0

gdy $D,E > H,L$ to CY=0 Z=0

Rejestry niezmienniane: B, C, D, E, H, L.

6. Konfiguracja sprzętowa

System RTMT przeznaczony jest do pracy na mikrokomputerze MM80 i znajduje się w wewnętrznej pamięci EPROM pakietu jednostki centralnej.

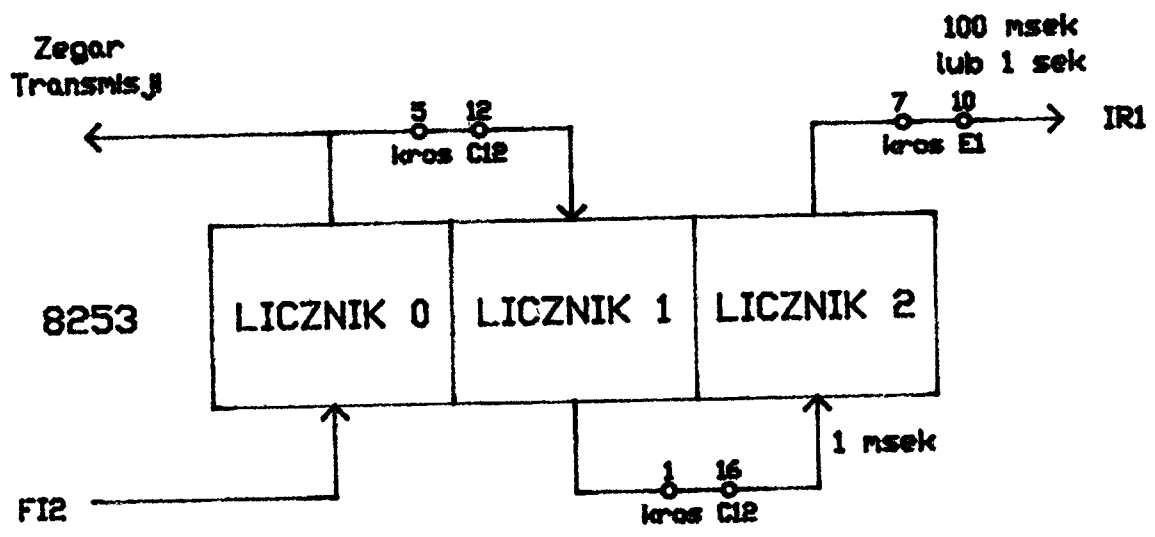
Do pakietu procesora można dołączyć:

- do złącza C dowolne urządzenie o interfejsie V24 stanowiące podstawowe urządzenie operatorskie systemu. Urządzeniu temu przyporządkowany jest zawsze numer logiczny 0;
- do złącza D czytnik taśmy papierowej CT2100;
- do złącza E dziurkarka taśmy papierowej DT1055;

Dodatkowo system może obsługiwać inne urządzenia o interfejsie V-24 podłączone do zestawu mikroprocesorowego poprzez pakiety MI24. System może obsługiwać, zależnie od wygenerowanej wersji, do dwóch pakietów MI24 (4 dodatkowe kanały V-24).

Ponadto system może współpracować z pakietami kontroli kasety MW30 lub MW32. Zależnie od wersji może on obsługiwać 1, 2 lub 3 kasety.

System programuje wszystkie trzy liczniki układu 8253 znajdującego się na pakiecie MM80. Na wejście licznika nr 0 podawany jest sygnał Φ_2 . Wyjście tego licznika jest połączone z wejściem licznika nr 1 oraz z wejściami RxC i TxC układu 8251 (zegar transmisji). Licznik nr 1 programowany jest tak aby na jego wyjściu uzyskać przebieg prostokątny o okresie 1 msek. Sygnał ten jest podawany na wejście licznika nr 2. Na wyjściu licznika nr 2, zależnie od wersji systemu otrzymywany jest przebieg o okresie 100 msek lub 1 sek podawany na wejście przerywające IR1 jako przerwanie zegarowe. Schemat połączeń sygnałów zegarowych na pakiecie MM80 przedstawia rysunek 3.



Rys 3. Schemat połączeń sygnałów zegarowych