

PRZEMYSŁOWY INSTYTUT AUTOMATYKI I POMIARÓW
MERA-PIAP

Al. Jerozolimskie 202

02-222 Warszawa

Telefon 23-70-81

OSRODEK AUTOMATYKI ELEKTRYCZNEJ

074 ZESPÓŁ BUDOWY CYFROWYCH URZĄDZEN SYSTEMOWYCH A

Główny wykonawca mgr inż. M. Słodczyk MSR

Wykonawcy mgr inż. T. Kacprowski, mgr inż. M. Partyka

Konsultant

Nr zlecenia RP 53.2

Opracowanie pakietu MM87 jednostki centralnej z procesorem arytmetycznym. Nr zadania 1.3. Program Monitor Operatorski i testy do badań pełnych. INSTRUKCJA UŻYTKOWANIA PROGRAMU MONITOR

Zleceniodawca CPBR 7.1

Pracę rozpoczęto dnia 86.09.01
Kierownik Zespołu

zakończono dnia 86.12.15
Kierownik Ośrodka

dr inż. A. Gryczyński

prof. dr inż. T. Miśkiewicz

Praca zawiera:

Rozdzielnik - ilość egz: 5

stron 17

Egz. 1 BOINTE

rysunków -

Egz. 2 OAP

fotografii 1

Egz. 3 OAE-3

tabel -

Egz. 4 OAE-8

tablic -

Egz. 5 OAE-8

załączników -

Egz. 6

Nr rejestr. 5767

Analiza deskryptorowa OPROGRAMOWANIE * INTELDIGIT - PROWAY

Analiza dokumentacyjna Dokumentacja zawiera instrukcję użytkowania programu MONITOR OPERATORSKI pakietu MM87.

Tytuły poprzednich sprawozdań Dokumentacja pakietu MM87 nr rej. 5755

UKD

PIAP-252/53-6000

SPIS TRESCI

1.	Wstęp	2
2.	Struktura dyrektyw MONITORA	2
3.	Dyrektywy MONITORA	3
3.1.	Dyrektywa S	4
3.2.	Dyrektywa X	5
3.3.	Dyrektywa D	6
3.4.	Dyrektywa M	7
3.5.	Dyrektywa I	8
3.6.	Dyrektywa O	8
3.7.	Dyrektywa G	9
3.8.	Dyrektywa N	10
3.9.	Dyrektywa U	11
3.10.	Dyrektywa H	12
3.11.	Dyrektywa Q	12
3.12.	Dyrektywa L	13
3.13.	Dyrektywa V	15
3.14.	Dyrektywa T	15

1. WSTĘP

Program MONITOR OPERATORSKI jest przeznaczony do pierwszego etapu badań urządzeń i zestawów z pakietem jednostki centralnej 16-bitowej MM87, oraz do pierwszego etapu prac nad opracowaniem programów użytkowych i testowych. Program MONITOR jest przez operatora obsługiwany z urządzenia dołączonego do złącza C pakietu MM87. Złącze D jest wykorzystywane do komunikacji z komputerem SM-4 pracującym pod systemem RSX 11M. Zbiory są przesyłane w formacie INTELLEC-HEX. Dyrektywy komunikacji z SM-em sprawdzają założenie złącza D i jego brak sygnalizują odpowiednim komunikatem.

Program MONITOR zajmuje obszar 4K pamięci PROM pakietu jednostki centralnej o adresach od FF000H do FFFFFH. Na swoje komórki robocze wykorzystuje pamięć RAM o adresach 400H - 433H; stos programu MONITOR zaczyna się od adresu 480H i biegnie w stronę adresów młodszych, a stos użytkownika rozpoczyna się od adresu 500H również w stronę adresów młodszych.

Oprócz tego program MONITOR inicjuje trzy wektory przerwań: przerwanie nr 1 - Single Step - używane w dyrektywach N i U, przerwanie nr 3 - Breakpoint - używane w dyrektywach G i U oraz przerwanie nr 20 - BTMO - sygnalizujące przekroczenie czasu wykonywania operacji przez procesor.

Po włączeniu zasilania lub po naciśnięciu przycisku RESET program przeprowadza inicjację układów programowalnych, wpisuje do wszystkich rejestrów użytkownika zera (z wyjątkiem rejestru SP, którego wartość początkowa jest ustawiana na 500H) i oczekuje na wybór przez operatora jednej z dyrektyw.

Gdy podczas wykonywania dowolnej operacji procesor nie otrzyma w określonym czasie sygnału potwierdzenia XACK zgłaszane jest przerwanie nr 20 (BTMO). Obsługa tego przerwania polega na wyświetleniu adresu instrukcji następnej za instrukcją, dla której nie wystąpił sygnał XACK poprzedzonego znakami !@ i przekazaniu sterowania do programu MONITOR (wyświetlane jest zgłoszenie M87>).

2. STRUKTURA DYREKTYW MONITORA

Program Monitor sygnalizuje gotowość do przyjmowania dyrektywy wyświetlając znaki "MM87>" na ekranie monitora. W tym momencie operator podaje nazwę dyrektywy z ewentualnymi argumentami. Kiedy wymagana liczba argumentów jest większa od 1, muszą być one rozdzielone znakiem przecinka (,). Wywołanie dyrektywy musi być zakończone, zależnie od rodzaju dyrektywy, znakiem powrotu karetki <CR>, lub przecinka (,). W danym momencie może być wykonywana tylko jedna dyrektywa, a operator może wywoływać dyrektywy po jednej w

linii.

Wszystkie argumenty liczbowe są liczbami heksadecymalnymi: dla bajtów z zakresu 00 do FF, a dla słów z zakresu 0000 do FFFF. Zera nieznaczące mogą być pomijane. Jeżeli operator podaje więcej niż dwa znaki (dla bajtu) lub niż cztery znaki (dla słowa) tylko ostatnie dwa lub cztery znaki zostaną przyjęte przez program MONITOR. Argument będący adresem składa się z wartości segmentu i przesunięcia (offset). Jeżeli segment adresu nie jest podany, jako jego wartość, przyjmowana jest zawartość rejestru CS użytkownika. Jeżeli oba składniki adresu są podawane to należy je podawać w kolejności najpierw segment potem offset oddzielone znakiem dwukropka (:). Gdy dyrektywa wymaga dwóch argumentów stanowiących adresy początku i końca obszaru pamięci to segment może być podawany tylko przy adresie początku. Wszędzie tam, gdzie argumentem jest słowo, można podać zamiast liczby heksadecymalnej nazwę rejestru. Jako argument zostanie wtedy przyjęta zawartość podanego rejestru użytkownika.

Podanie błędnej nazwy dyrektywy lub błędnego argumentu powoduje zakończenie przyjmowania dyrektywy, wyświetlenie znaku zapytania (?) i zgłoszenie gotowości do przyjęcia następnej dyrektywy znakami MMB7>.

3. DYREKTYWY MONITORA

Program MONITOR OPERATORSKI realizuje dyrektywy wymienione w tabeli 1. W tabeli tej oraz w opisie poszczególnych dyrektyw przyjęto następujące oznaczenia:

- [] - opcjonalna część dyrektywy
- []* - opcjonalna wielokrotna część dyrektywy
- < > - zmienna
- <CR> - powrót karetki
- <A> - adres
- <AP> - adres początku
- <AK> - adres końca
- <AL> - adres przeznaczenia
- <AS> - adres startu
- <D> - dane
- <PR> - przesunięcie
- <R> - nazwa rejestru
- <I/O> - adres I/O
- <O> - operator + lub -
- nazwa zbioru - nazwa pliku w systemie RSX 11M

W przykładach podanych w opisach poszczególnych dyrektyw, znaki pisane przez program MONITOR OPERATORSKI są podkreślone w odróżnieniu od znaków podawanych przez operatora.

3.1 DYREKTYWA S - odczyt i zmiana zawartości pamięci

`S[SW]<A>,[<D>],]*<CR>`

Dyrektywa S odczytu zawartości pamięci służy do odczytu bajtu (S) lub słowa (SW) z podanego obszaru pamięci. Jeżeli zawartość tej pamięci może być modyfikowana (pamięć RAM) opcjonalnie można zmienić jej zawartość.

Po wywołaniu dyrektywy literą S lub SW należy podać adres komórki pamięci, która ma być odczytana i ewentualnie zmieniona. Jeżeli adres segmentu nie zostanie podany to do obliczenia adresu efektywnego będzie użyta zawartość rejestru CS użytkownika. Wprowadzony adres należy zakończyć znakiem przecinka (,). W odpowiedzi zostanie wyświetlona zawartość podanej komórki pamięci (poprzedzona offsetem adresu), po której zostanie wyświetlony znak myślnika (-) sygnalizujący czekanie na znak z klawiatury. Jeżeli chcemy zmieniać zawartość pamięci należy w tym momencie podać jej nową wartość. Jeżeli odczytywana (i ewentualnie zmieniana) ma być tylko jedna komórka pamięci należy zakończyć dyrektywę znakiem powrotu karetki <CR>. W celu odczytania (i ewentualnej zmiany) zawartości kolejnych komórek pamięci należy kończyć wiersz dyrektywy znakiem przecinka (,). Aby odczytać zawartość wcześniejszych (o mniejszych adresach) komórek pamięci należy zamiast znaku przecinka naciskać znak minus (-).

Należy zwrócić uwagę, że przy użyciu dyrektywy SW najpierw wyświetlana jest zawartość komórki pamięci o adresie o 1 większym od podanego, a po niej zawartość komórki pamięci o podanym adresie. Analogicznie wprowadzając nowe dane (w dyrektywie SW) pierwszy wprowadzany bajt umieszczany jest w następnej komórce pamięci, a drugi bajt w bieżącej komórce.

BŁĘDY:

- próba wpisania nowej zawartości do nie istniejącej pamięci lub pamięci typu PROM - sygnalizacja znakiem wykrzyknika (!)
- podanie znaku, który nie jest cyfrą heksadecymalną w adresie lub danych - sygnalizacja znakiem zapytania (?).

PRZYKŁADY:

1. Sprawdzenie zawartości pamięci o adresie FF00H:

```
MMB7>SFF00:0,
0000 75-<CR>
MMB7>
```

2. Zmiana zawartości komórki pamięci RAM o adresie 100H relatywnie do rejestru DS użytkownika na wartość 11H i sprawdzenie zawartości kolejnej komórki pamięci:

```
MMB7>SDS:100,
0100 00-11,
0101 22-<CR>
MMB7>
```

3. Sprawdzenie zawartości wierzchołka stosu:

```
MMB7>SWSS:SP,  
02FE 2323-<CR>  
MMB7>
```

4. Zmiana zawartości słowa umieszczonego w pamięci RAM pod adresem 200H relatywnie do rejestru CS na wartość 1234H i odczyt dwóch poprzednich słów:

```
MMB7>SW200H,  
0200 2333-1234,  
0202 C0CC--  
0200 1234--  
01FE 3374-<CR>  
MMB7>
```

3.2 DYREKTYWA X - odczyt i zmiana zawartości rejestru

`X[<R>]I[<D>],I*<CR>`

Dyrektywa X odczytu i zmiany zawartości rejestru służy do odczytu i opcjonalnie do zmiany zawartości każdego z rejestrów procesora 8086 lub do odczytu wszystkich jego rejestrów.

W celu wyświetlenia zawartości wszystkich 14 rejestrów procesora należy po literze X wywołującej dyrektywę nacisnąć klawisz powrotu karetki <CR>. Zostanie wyświetlona zawartość rejestrów i dyrektywa kończy się.

W celu wyświetlenia i opcjonalnie zmiany zawartości jednego rejestru należy po literze X wywołującej dyrektywę napisać skrót nazwy tego rejestru jako jedną z nazw: AX, BX, CX, DX, SP, BP, SI, DI, CS, DS, SS, ES, IP, FL i zakończyć wiersz dyrektywy znakiem powrotu karetki <CR>. Na monitorze zostanie wyświetlony skrót nazwy rejestru, znak równości (=), aktualna zawartość rejestru, a po niej znak myślnika (-) sygnalizujący oczekiwanie na nowe dane.

Jesli chcemy zmieniać zawartość rejestru należy wpisać nowe dane i zakończyć dyrektywę znakiem przecinka (,) (gdy chcemy wyświetlić zawartość kolejnego rejestru) lub znakiem powrotu karetki <CR> (gdy chcemy zakończyć dyrektywę). Kolejność rejestrów jest taka, jak podana powyżej (AX...FL), a po rejestrze wskaźników procesora (FL) następuje ponownie rejestr akumulatora (AX).

Gdy nie chcemy zmieniać zawartości rejestru należy zakończyć wiersz dyrektywy znakiem powrotu karetki <CR>.

BŁĘDY:

- podanie skrótu nazwy rejestru, który nie jest jednym z wymienionych: AX, BX, CX, DX, SP, BP, SI, DI, CS, DS, SS, ES, IP, FL - sygnalizowane znakiem zapytania (?)
- podanie w danych znaku, który nie jest cyfrą heksadecymalną - sygnalizowane znakiem zapytania (?)

PRZYKŁADY:

1. Wyświetlenie zawartości wszystkich rejestrów procesora:
MMB7>X<CR>

AX=0000 BX=1111 CX=2222 DX=3333 SP=0300 BP=0000 SI=0100
DI=0200 CS=FF00 DS=0000 SS=0000 ES=1000 IP=0356 FL=F002
MMB7>

2. Zmiana zawartości rejestru CS i sprawdzenie zawartości dwóch następnych rejestrów:

MMB7>XCS<CR>
CS=FF00-0000,
DS=0000-
SS=0000-<CR>
MMB7>

3.3 DYREKTYWA D - odczyt zawartości pamięci

D<AP>[,<AK>]<CR>

Dyrektywa D odczytu zawartości pamięci służy do wyświetlania na monitorze zawartości bloku pamięci.

Operator podaje rozdzielone znakiem przecinka (,) adresy początku i końca obszaru pamięci, którego zawartość chce wyświetlić. Przy adresie końca nie należy podawać segmentu (segment adresu końca jest taki sam jak segment adresu początku). Dlatego też za pomocą dyrektywy D można wyświetlić zawartość bloku pamięci o pojemności do 64k bajtów.

Możliwe jest podanie samego adresu początku. Wyświetlona zostaje wtedy (poprzedzona offsetem) zawartość jednej komórki pamięci.

Dyrektywę kończy znak powrotu karetki <CR>. Na monitorze zostaje wyświetlony nagłówek zawierający ostatnie cyfry offsetu dla poszczególnych komórek pamięci, a pod nim dane w wierszach po 16 komórek. Każdy wiersz rozpoczyna się offsetem adresu pierwszej pozycji w danym wierszu.

Po wyświetleniu zawartości zadanego bloku pamięci MONITOR ponownie zgłasza swoją gotowość do przyjmowania dyrektyw wyświetlając znak (>).

BŁĘDY:

- podanie adresu końca mniejszego niż adres początku - sygnalizowane znakiem zapytania (?).
- użycie w adresie znaku nie będącego cyfrą heksadecymalną - sygnalizowane znakiem zapytania (?)
- użycie znaku różnego od przecinka (,) do rozdzielania adresów początku i końca obszaru pamięci - sygnalizowane znakiem zapytania (?)

PRZYKŁADY:

1. Wyświetlenie zawartości pamięci od adresu 101H do

adresu 115H relatywnie do rejestru DS:

MMB7>DDS:101,115<CR>

	<u>0</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	<u>8</u>	<u>9</u>	<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>	<u>E</u>	<u>F</u>
<u>0100</u>		<u>11</u>	<u>22</u>	<u>33</u>	<u>44</u>	<u>55</u>	<u>66</u>	<u>77</u>	<u>88</u>	<u>99</u>	<u>AA</u>	<u>BB</u>	<u>CC</u>	<u>DD</u>	<u>EE</u>	<u>FF</u>
<u>0110</u>	<u>00</u>	<u>11</u>	<u>22</u>	<u>33</u>	<u>44</u>	<u>55</u>										

MMB7>

3.4 DYREKTYWA M - przesunięcie zawartości pamięci

M<AP>,<AK>,<AL><CR>

Dyrektywa M służy do przesuwania zawartości pamięci z jednego obszaru na inny. Dyrektywa ta może też służyć do wpisywania stałej wartości do obszaru pamięci.

Obszar pamięci, z którego dane są przesuwane jest deklarowany poprzez adresy początku i końca rozdzielone znakiem przecinka (,). Podobnie jak w dyrektywie D adres końca przesuwanego obszaru jest ustalany relatywnie do tego samego segmentu jak adres początku. Zatem największy przesuwany obszar pamięci może mieć 64k bajtów. Adres przeznaczenia (oddzielony od adresu końca znakiem przecinka (,)) może zawierać dowolną wartość segmentu. Gdy segment adresu przeznaczenia nie będzie podany jego wartość zostaje przyjęta taka jak dla adresu początku.

Jeżeli adres przeznaczenia jest o jeden większy od adresu początku to pamięć ograniczona adresem początku i adresem końca +1 zostanie wypełniona zawartością komórki określonej przez adres początku.

BŁĘDY:

- użycie w adresie znaku nie będącego cyfrą heksadecymalną - sygnalizowane znakiem zapytania (?)
- podanie adresu końca mniejszego od adresu początku - sygnalizowane znakiem zapytania (?)
- użycie znaku różnego od przecinka (,) dla rozdzielenia poszczególnych adresów - sygnalizowane znakiem zapytania (?)
- próba przesunięcia danych do pamięci typu PROM lub do pamięci nieistniejącej - sygnalizowane znakiem wykrzyknika (!) poprzedzonym offsetem adresu pierwszej komórki pamięci, dla której stwierdzono błąd.

PRZYKŁADY:

1. Przesunięcie zawartości pamięci od adresu 100H do 130H relatywnie do rejestru CS na obszar pamięci poczynsz od adresu 500H relatywnie do rejestru DS:

MMB7>M100,130,DS:500<CR>

MMB7>

2. Wpis wartości OFFH do obszaru pamięci od adresu 300H do 3FFH relatywnie do zawartości rejestru BX:

MMB7>SBX:300,

0300 10-FF<CR>

MMB7>MBX:300,3FE,301<CR>

MMB7>

3.5. DYREKTYWA I - czytanie bramy wejściowej

I[W]<I/O>[,]*<CR>

Dyrektywa I czytania bramy wejściowej służy do odczytu bajtu (dyrektywa I) lub słowa (dyrektywa IW) z podanej bramy wejściowej i wyświetlenia wartości tych danych na monitorze. Ponieważ przestrzeń adresowa bram wejściowych i wyjściowych jest ograniczona do 64K, adres bramy podaje się jako liczbę heksadecymalną z zakresu 0-0FFFFH bez podawania wartości segmentu.

Po podaniu adresu bramy wejściowej należy nacisnąć znak przecinka (,) w celu wyświetlenia na monitorze bajtu lub słowa danych z bramy wejściowej. Każdy następny znak przecinka (,) powoduje wyświetlenie, w nowej linii, aktualnych danych z bramy wejściowej. Znak powrotu karetki <CR> kończy dyrektywę.

BŁĘDY:

- podanie adresu bramy wejściowej nie będącego liczbą heksadecymalną - sygnalizacja znakiem zapytania (?).

PRZYKŁADY:

1. Odczyt słowa z bramy o adresie 1717H:

MM87>IW1717,

2200<CR>

MM87>

2. Trzykrotny odczyt bajtu z bramy o adresie 0DAH:

MM87>IDA,

FF,

F5,

FF<CR>

MM87>

3.6. DYREKTYWA O - wysłanie danych do bramy wyjściowej

O[W]<I/O>,<D>[, <D>]*<CR>

Dyrektywa O wysyłania danych do bramy wyjściowej służy do wysłania bajtu (dyrektywa O) lub słowa (dyrektywa OW) z klawiatury monitora na podaną bramę wyjściową. Ponieważ przestrzeń adresowa bram wyjściowych jest ograniczona do 64K, adres bramy podaje się jako liczbę heksadecymalną z zakresu 0-0FFFFH bez podawania wartości segmentu. Po podaniu adresu bramy wyjściowej należy nacisnąć znak przecinka (,), a po nim podać dane do wysłania. Następnie należy podać znak powrotu karetki <CR> w celu wysłania danych i zakończenia dyrektywy lub znak przecinka (,) w celu wysłania danych i oczekiwania na podanie z klawiatury nowych danych do tej samej bramy wyjściowej. Program oczekuje na nowe dane w nowym wierszu

po znaku myślnika (-). Kolejne dane można wysłać do bramy wyjściowej przez podanie tych danych zakończonych znakiem przecinka (,). Znak powrotu karetki <CR> po nowych danych kończy dyrektywę.

BŁĘDY:

- podanie adresu bramy nie będącego liczbą heksadecymalną - sygnalizacja znakiem zapytania (?)

PRZYKŁADY:

1. Wysłanie bajtu do bramy o adresie 0DOH:

MM87>0D0,80<CR>

MM87>

3.7. DYREKTYWA G - uruchomienie programu

G[<AS>],[,<A>]<CR>

Dyrektywa G służy do uruchomienia programu użytkownika od podanego adresu z ewentualnym ustawieniem pułapki (breakpoint).

Po wywołaniu dyrektywy literą G na ekranie wyświetlana jest aktualna zawartość rejestru IP oraz zawartość komórki pamięci, której adres jest wyznaczony przez zawartość rejestrów CS i IP (komórką pamięci, od której byłby wykonywany program gdyby nie był podany nowy adres). Podanie w tym momencie znaku powrotu karetki <CR> kończy dyrektywę i uruchamia program od adresu określonego przez rejestry CS i IP. W tym momencie można także uruchomić program od innego adresu podając ten adres i kończąc go znakiem powrotu karetki <CR>.

Dyrektywa G umożliwia także opcjonalne ustawienie pułapki. Adres pułapki (sam offset) musi być podany po znaku przecinka (,), za adresem startu programu (o ile był podany).

Gdy operator ustawia pułapkę, kod instrukcji o podanym adresie jest pamiętany przez program MONITOR i zastąpiony kodem instrukcji przerwania programowego INT3. Kiedy program użytkownika natrafi na tę instrukcję jego wykonywanie zostanie przerwane i sterowanie zostaje przekazane do programu MONITOR. Sygnalizowane jest to wyświetleniem komunikatu:

BR @aaaa:bbbb

M87>

W komunikacie tym aaaa oznacza bieżącą zawartość rejestru CS, a bbbb - zawartość rejestru IP (rejestry te wyznaczają adres, na którym program został przerwany). Jeżeli ponownie zostanie użyta dyrektywa G bez podania adresu startu to program zostanie wznowiony od instrukcji zastąpionej przez pułapkę.

Po zatrzymaniu programu w miejscu ustawienia pułapki

możliwe jest korzystanie ze wszystkich dyrektyw MONITORA. W szczególności można użyć dyrektywę G z ponownym ustawieniem pułapki (z wyjątkiem ustawienia pułapki w miejscu zatrzymania programu – do tego celu służy dyrektywa U).

Do obsługi pułapki program MONITOR wykorzystuje przerwanie INT3 – użytkownik nie może zmieniać wektora tego przerwania. Użytkownik może korzystać z obsługi tego przerwania umieszczając w swoim programie instrukcję przerwania programowego INT3. Po wykonaniu tej instrukcji sterowanie zostaje przekazane do programu MONITOR przy zachowaniu wszystkich rejestrów procesora. Sygnalizowane jest to komunikatem:

```
aaaa:bbbb
M87>
```

Obsługę przerwania INT3 można także wykorzystać do obsługi przerwania operatora. W tym celu w wektor obsługi tego przerwania należy wpisać adres obsługi przerwania INT3 (segment= i offset=). Działanie w przypadku zgłoszenia przerwania operatora będzie identyczne jak dla wykonania instrukcji INT3 w programie użytkownika.

BŁĘDY:

- użycie w adresie znaku nie będącego cyfrą heksadecymalną – sygnalizacja znakiem zapytania (?)
- próba ustawienia pułapki na obszarze pamięci typu PROM lub pamięci nieistniejącej – sygnalizacja znakiem wykrzyknika (!)

PRZYKŁADY:

1. Uruchomienie programu od adresu wyznaczonego przez bieżącą zawartość rejestrów CS i IP:

```
MM87>G 1234-50 <CR>
```

2. Uruchomienie programu od adresu 10000H z ustawieniem pułapki na adresie 10040H i wznowienie programu z przesunięciem pułapki na adres 10047H:

```
MM87>G 0000-00 1000:0,40<CR>
```

```
BR @1000:0040
```

```
MM87>G 0040-58 ,47<CR>
```

3.8. DYREKTYWA N – praca krokowa (single step)

```
NI<AS>],[[<AS>],]*<CR>
```

Dyrektywa N służy do wykonania pojedynczej instrukcji programu. Po wykonaniu tej instrukcji sterowanie jest ponownie przekazywane do programu MONITOR.

Po wywołaniu dyrektywy literą N na monitorze wyświetlana jest aktualna zawartość rejestru IP i zawartość komórki pamięci, której adres jest wyznaczony

przez rejestry CS i IP (komórka pamięci od której byłby wykonywany program gdyby nie był podany nowy adres). Gdy operator chce wykonać instrukcję o innym adresie niż określony przez rejestry CS i IP może podać ten adres i zakończyć go znakiem przecinka (,).

Podanie znaku przecinka (,) powoduje wykonanie jednej instrukcji programu i powrót do dyrektywy N. W nowym wierszu wyświetlany jest adres (offset) i pierwszy bajt następnej instrukcji do wykonania. W tym momencie operator może podać:

- a) - znak powrotu karetki <CR> w celu zakończenia dyrektywy i przejścia do programu MONITOR;
- b) - znak przecinka (,) w celu wykonania kolejnej instrukcji programu;
- c) - adres następnej instrukcji do wykonania zakończony znakiem przecinka (,);

Jeżeli operator wybierze działania wg punktów b) lub c), po wykonaniu kolejnej instrukcji, nastąpi powrót do dyrektywy N.

Dyrektywa wykorzystuje przerwanie nr 1 - single step - w związku z tym użytkownik nie może zmieniać wektora tego przerwania. Nie wolno wykonywać dyrektywy N dla wykonywania programu obsługi przerwania po jego wystąpieniu i w sekwencjach programu zmieniających rejestry stosu (SS i SP).

BŁĘDY:

- podanie w adresie znaku nie będącego cyfrą heksadecymalną - sygnalizacja znakiem zapytania (?)

PRZYKŁADY:

1. Wykonanie kilku instrukcji poczynawszy od adresu 1A5EH relatywnie do rejestru CS:

```
MMB7>N 0000-00 1A5E,  
1A5F-50 ,  
1A60-B8 ,  
1A63-8E ,  
1A65-B9 <CR>  
MMB7>
```

2. Wykonanie dwóch kolejnych instrukcji i ponowne wykonanie drugiej instrukcji:

```
MMB7>N 0031-03 ,  
0034-40 ,  
0035-50 34,  
0035-50 <CR>
```

3.9. DYREKTYWA U - utrzymanie pułapki

U<CR>

Dyrektywa U służy do uruchomienia programu od adresu określonego przez rejestry CS i IP z ustawieniem pułapki w

miejscu adresu startu programu.

PRZYKŁADY:

1. Dwukrotne wykonanie pętli programowej:

MMB7>U 0151-90 <CR>

BR @1000:0151

MMB7>U 0151-90 <CR>

BR @1000:0151

MMB7>

3.10. DYREKTYWA H - arytmetyka heksadecymalna

H<D><O><D>=

Dyrektywa H arytmetyki heksadecymalnej służy do obliczania sumy i różnicy liczb heksadecymalnych z zakresu 0-FFFFH.

Po wywołaniu dyrektywy literą H należy podać dwie liczby heksadecymalne rozdzielone znakiem operatora arytmetycznego: plusem (+) lub minusem (-) i zakończone znakiem równości (=). W odpowiedzi na monitorze jest wyświetlany bezpośrednio za znakiem równości wynik operacji arytmetycznej w postaci czterech cyfr heksadecymalnych i dyrektywa kończy się.

BŁĘDY:

- podanie danych nie będących liczbą heksadecymalną - sygnalizacja znakiem zapytania (?)
- podanie jako operatora arytmetycznego znaku różnego od plusa (+) lub minusa (-) - sygnalizacja znakiem zapytania (?)

PRZYKŁADY:

MMB7>H1+2=0003

MMB7>HFFFF+02=0001

3.11. DYREKTYWA Q - bezpośrednia komunikacja z SM-em

Q

Dyrektywa Q powoduje, że monitor ekranowy dołączony do pakietu MMB7 staje się zwykłym terminalem minikomputera SM (linia od SM-a musi być dołączona do złącza D pakietu). Od momentu wywołania dyrektywy aż do jej zakończenia znakiem CTRL/J (kod 1DH) wszystkie znaki podane przez operatora z klawiatury są przesyłane do SM-a, a znaki odbierane od SM-a są wyświetlane na ekranie.

14

3.12. DYREKTYWA L - czytanie danych z SM-a

LI<PR>],nazwa zbioru<CR>

Dyrektywa L umożliwia wczytanie danych w formacie taśmy papierowej (8080 lub 8086) przez złącze D pakietu MM87. Mikrokomputer MM87 zastępuje, w czasie wykonywania dyrektywy L, jeden z terminali SM-a. MM87 wysyła do SM-a sekwencję znaków wywołującą program PIP i powodującą przesłanie zawartości podanego przez operatora zbioru na terminal. Przed wywołaniem dyrektywy L komputer SM musi znajdować się w stanie oczekiwania na przyjęcie rozkazu MCR-a, a zadanie PIP nie może być aktywne. Urządzenie zewnętrzne, z którego ma być przesyłany zbiór musi być uprzednio przyłączone do systemu rozkazem MOU.

Po zakończeniu dyrektywy L znakiem <CR> (po podaniu opcjonalnego przesunięcia i nazwy zbioru) mikrokomputer MM87 wysyła do SM-a tekst:

PIP TI:=nazwa zbioru<CR>

sprawdzając po wysłaniu każdego znaku otrzymane od SM-a echo, a następnie zaczyna odbierać nadawane przez SM znaki. W trakcie odbioru kontrolowana jest zgodność otrzymywanego zbioru z "Hexadecimal Object File Format" i sprawdzana jest prawidłowość zapisu danych do pamięci. Wszystkie znaki otrzymywane poprzez kanał transmisji (złącze D) są na bieżąco wyświetlane na ekranie monitora.

Transmisja zakończona prawidłowo sygnalizowana jest komunikatem:

TRANSMISJA OK

Jeżeli MM87 stwierdzi błąd w odbieranym echu w trakcie wysyłania do SM-a komunikatu z nazwą zbioru sygnalizuje to komunikatem:

ECHO

i na ekranie wyświetlane są wszystkie znaki otrzymywane od SM-a aż do znaku zgłoszenia MCR-a.

Błąd w trakcie transmisji danych sygnalizowany jest w momencie jego wykrycia jednym z następujących komunikatów:

BLAD SUMY
ZLY TYP
BLAD W TRANSMISJI

i MM87 ignoruje od tej chwili wszystkie znaki przychodzące od SM-a aż do otrzymania znaku ">".

Jeżeli przesyłany zbiór jest w formacie 8080 to dane umieszczane są w pamięci RAM zgodnie z adresami ładowania znajdującymi się w rekordach relatywnie do segmentu o wartości 0. Adres startu w rekordzie końca zbioru (jeżeli

jest różny od 0000)- umieszczany jest w rejestrze IP użytkownika, a zawartość rejestru CS pozostaje bez zmiany.

Jeżeli przesyłany zbiór jest w formacie 8086 i zawiera rekord adresu startu programu to rejestry CS i IP użytkownika przyjmują wartości zgodnie z odczytanymi w tym rekordzie.

Gdy operator użyje w wywołaniu dyrektywy opcjonalnego przesunięcia, to jego wartość (liczba heksadecymalna z zakresu 0-FFFF) jest dodawana do każdego adresu ładowania (offsetu) odczytanego z rekordu danych.

BŁĘDY:

- podanie w przesunięciu znaku nie będącego cyfrą heksadecymalną - sygnalizowane znakiem zapytania (?)
- błędne echo od SM-a w trakcie inicjacji transmisji - sygnalizacja komunikatem: ECHO
- wykryta błędna suma kontrolna w rekordzie danych otrzymanym z SM-a - sygnalizacja komunikatem: BLAD SUMY
- niezdefiniowany typ rekordu - sygnalizacja komunikatem: ZLY TYP
- otrzymanie od SM-a w rekordzie danych znaku nie będącego cyfrą heksadecymalną - sygnalizacja komunikatem: BLAD W TRANSMISJI
- adres ładowania w rekordzie (plus ewentualne przesunięcie) nie wyznacza pamięci typu RAM lub informacja została źle wpisana do pamięci - sygnalizacja znakiem wykrzyknika poprzedzonym offsetem adresu, dla którego wystąpił błąd.

PRZYKŁADY:

1. Czytanie zbioru o nazwie PROBNY.T86 z dysku systemowego:
MB7>L,PROBNY.T86<CR>

```

|
|
|  zawartość zbioru
|
|

```

≥
TRANSMISJA OK

MB7>

2. Czytanie zbioru o nazwie TEST.T86 z dyskietki DX1: z konta [25,10] z przesunięciem 1000H:

MB7>L1000,DX1:[25,10]TEST.T86<CR>

```

|
|
|  zawartość zbioru
|
|

```

≥
TRANSMISJA OK

MB7>

3.13. DYREKTYWA V - porównanie zawartości pamięci ze zbiorem z SM-a

V[<PR>],nazwa zbioru<CR>

Dyrektywa V umożliwia porównanie zawartości pamięci (RAM lub PROM) ze zbiorem w formacie INTELLEC-HEX znajdującym się w komputerze SM. Działanie dyrektywy i jej obsługa operatorska są analogiczne jak dyrektywy L z tą różnicą, że dane transmitowane z SM-a nie są wpisywane do pamięci mikrokomputera a jedynie z nią porównywane. Niezgodność danych sygnalizowana jest znakiem wykrzyknika poprzedzonym offsetem adresu komórki, dla której stwierdzono błąd.

3.14. DYREKTYWA T - prześlij dane do SM-a

T[X]<AP>,<AK>,[<AS>]<CR>
nazwa zbioru<CR>

Dyrektywa T służy do przesyłania zawartości zadeklarowanego obszaru pamięci do minikomputera SM jako zbioru w formacie INTELLEC-HEX o podanej przez operatora nazwie. Dane mogą być przesyłane w formacie 8086 (dyrektywa T) lub w formacie 8080 (opcja TX).

Po wywołaniu dyrektywy operator podaje adres początku i końca (oddzielone znakiem przecinka) obszaru pamięci, którego zawartość ma być przesyłana. W adresie końca nie można podać segmentu (segment adresu końca jest taki sam jak segment adresu początku). Dlatego też za pomocą dyrektywy T można przysyłać zawartość bloku pamięci o pojemności do 64k bajtów.

Po adresie końca opcjonalnie może być podany (poprzedzony znakiem przecinka) offset adresu startu programu (segment adresu startu jest taki sam jak segment wyprowadzanego obszaru pamięci).

Po zakończeniu znakiem powrotu karetki <CR> pierwszego wiersza wywołania dyrektywy w odpowiedzi na tekst:

PODAJ NAZWE FLIKU:

operator podaje nazwę pliku jaki ma zostać utworzony w SM-ie zakończoną znakiem <CR>.

Pakiet MM87 wysyła do SM-a tekst:

PIP nazwa zbioru=TI:

sprawdzając po wysłaniu każdego znaku otrzymane od SM-a echo, a następnie zaczyna nadawać zawartość pamięci. Po wysłaniu każdego znaku otrzymane w odpowiedzi echo jest wyświetlane na ekranie monitora.

Jeżeli MM87 stwierdzi błąd w odbieranym echu przerywa transmisję i sygnalizuje to komunikatem:

ECHO

a na ekranie wyświetlane są wszystkie znaki otrzymywane od SM-a aż do znaku zgłoszenia MCR-a.

Po przesłaniu wszystkich danych MM87 wysyła znak CTRL/Z, oczekuje na zgłoszenie MCR-a po czym powraca do programu MONITOR.

Gdy operator przesyła zbiór w formacie 8086, wartość segmentu podanego w adresie początku (lub zawartość rejestru CS gdy segment nie został podany) jest przesyłany w rekordzie rozszerzenia adresu. Offset adresu początku jest umieszczany jako adres ładowania w pierwszym rekordzie danych. Transmitowany rekord adresu startu (o ile został podany adres startu) zawiera segment i offset, które zostaną umieszczone w rejestrach CS i IP użytkownika przy ściąganiu danych dyrektywą L.

Gdy operator wyprowadza dane w formacie 8080 offset adresu początku jest umieszczany jako adres ładowania w pierwszym rekordzie danych, natomiast adres startu programu (jeżeli został podany) jest umieszczany w rekordzie końca zbioru. Segment adresu początku jest w tym przypadku używany tylko do definicji wyprowadzanego obszaru pamięci i nie jest transmitowany do SM-a.

BŁĘDY:

- podanie adresu końca mniejszego niż adres początku - sygnalizowane znakiem zapytania
- podanie w przesunięciu znaku nie będącego cyfrą heksadecymalną - sygnalizowane znakiem zapytania (?)
- błędne echo od SM-a w trakcie transmisji - sygnalizacja komunikatem: ECHO

PRZYKŁADY:

1. Przesłanie zawartości pamięci w formacie 8086 od adresu 1000H do adresu 11FFH relatywnie do rejestru CS jako zbiór PROBA.T86 na dysku systemowym:

```
MM87>T1000,11FF,1000<CR>
PODAJ NAZWE PLIKU:PROBA.T86<CR>
```

```
|
|
|przesyłany zbiór
|
|
```

```
^Z
^
```

```
MM87>
```

2. Przesłanie zawartości pamięci od adresu 100:0 do adresu 3FFH w formacie 8080 na dyskietkę DX1 jako zbioru TEST.T86:

MB7>TX100:0,3FF<CR>

PODAJ NAZWE PLIKU:DX1:TEST.T86<CR>

|
|
transmitowany zbiór
|
|

^Z

>

MB7>

Tabela 1
Zestawienie dyrektyw MONITORA OPERATORSKIEGO

I=====I	
I Dyrektywa - Funkcja I	I Wywołanie I
I=====I	
I S Odczyt i zamiana komórki pamięci	I S[W]<A>,[<D>],]*<CR>
I-----I	I-----I
I X Odczyt i zamiana zawartości	I X<R>[[<D>],]*<CR>
I rejestru	I
I-----I	I-----I
I D Odczyt zawartości pamięci	I D<AP>[,<AK>]<CR>
I-----I	I-----I
I M Przesunięcie zawartości pamięci	I M<AP>,<AK>,<AL><CR>
I-----I	I-----I
I I Odczyt danych z bramy wejściowej	I I[W]<I/O>[,]*<CR>
I-----I	I-----I
I O Wpis danych do bramy wyjściowej	I O[W]<I/O>,<D>[,<D>]*<CR>
I-----I	I-----I
I G Uruchomienie programu z ewentualną	I G<AS>[,<A>]<CR>
I pułapką	I
I-----I	I-----I
I N Wykonanie pojedynczej instrukcji	I N<AS>,[<AS>],]*<CR>
I programu użytkownika	I
I-----I	I-----I
I U Utrzymanie pułapki	I U<CR>
I-----I	I-----I
I H Arytmetyka heksadecymalna	I H<D><O><D>=
I-----I	I-----I
I Q Komunikacja z SM-em	I Q
I-----I	I-----I
I L Czytanie danych z SM-a	I L<PR>[,nazwa<CR>
I-----I	I-----I
I V Porównanie danych z SM-a	I V<PR>[,nazwa<CR>
I-----I	I-----I
I T Przesyłanie danych do SM-a	I T[X]<AP>,<AK>[,<AS>]<CR>
I	I nazwa<CR>
I-----I	I-----I