

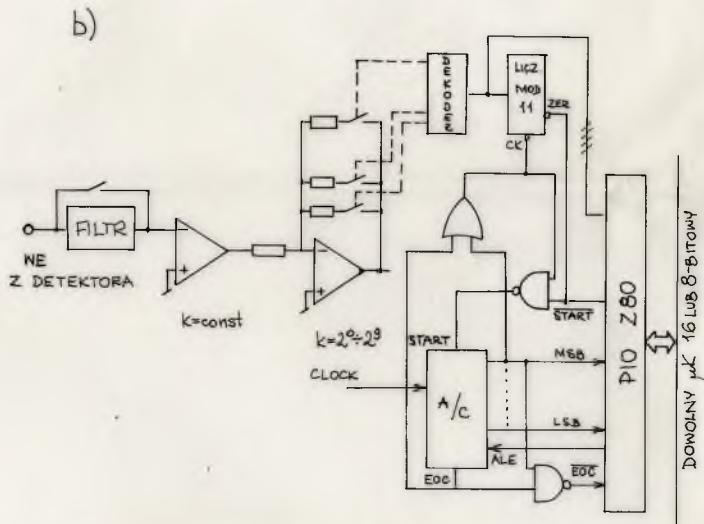
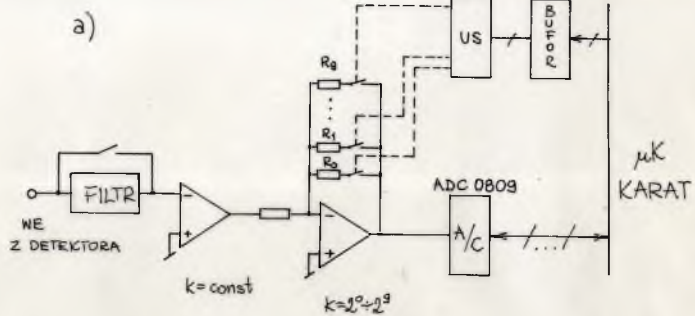
System CHOMIK?

- Prototyp -

~~materialy do sporząd.~~

~~4~~ dokument. techn.

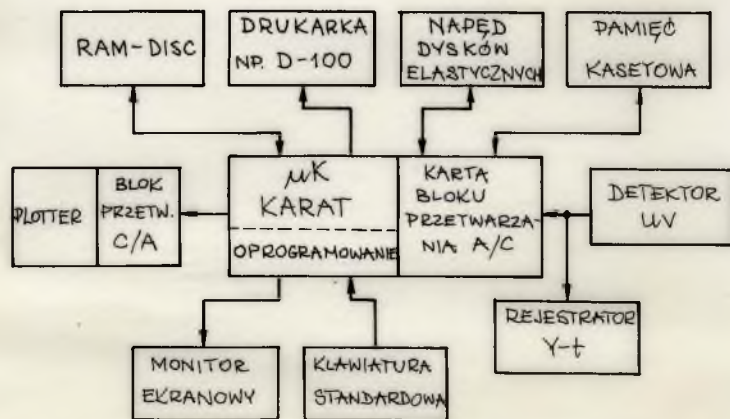
rej.



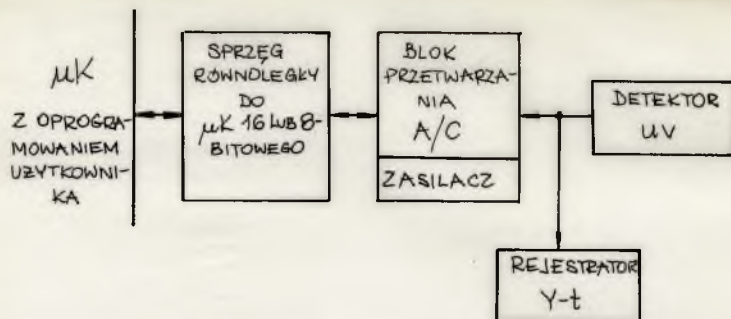
Rys.3 Blok przetwarzania a/c

a) dla wersji a urządzenia

b) dla wersji b urządzenia



Rys.1. Schemat blokowy urządzenia w/g wersji a)
- konfiguracja maksymalna



Rys.2. Schemat blokowy urządzenia w/g wersji b)

POLITECHNIKA GDAŃSKA
INSTYTUT CHEMII I TECHNOLOGII NIEORGANICZNEJ ORAZ KOROZJI
Zespół N-B Chromatografii Cieczowej

Umowa 870119
CPBP 12.3 cel 7

Część I

DOKUMENTACJA TECHNICZNA CZĘŚCI SPRZĘTOWEJ SYSTEMU
REJESTRACJI I PRZETWARZANIA DANYCH DO CHROMATOGRAFII
KARAT 2

Część II

DOKUMENTACJA OPROGRAMOWANIA SYSTEMU REJESTRACJI
I PRZETWARZANIA DANYCH DO CHROMATOGRAFII CIECZOWEJ
CHOMIK

- do użytku wewnętrznego -

Autorzy:

dr inż. Marian Kamiński
dr inż. Krystyna Gazda
mgr inż. Andrzej Adler
mgr inż. Andrzej Lech
Artur Kowalski
mgr inż. Bogdan Kandybowicz
mgr Andrzej Zapałowski
Piotr Przestrzelski

Kierownik Zespołu N-B
Chromatografii Cieczowej

Prof.dr hab.inż. J. Kowalczyk

Gdańsk 1989

SPIS TRESCI.

Część 1.

Schemat blokowy i instrukcja obsługi KARAT-2.

Część 2.

Schematy ideowe KARAT-2.

- 2.1. zespół CPU
- 2.2. zespół CPU DRAM
- 2.3. zespół CPU, CMOS RAM
- 2.4. zespół VIDEO, Układy synchronizacji
- 2.5. zespół VIDEO, Kontroler DRAM
- 2.6. zespół VIDEO, Układy dostępu DRAM
- 2.7. zespół VIDEO, Układy pamięci DRAM
- 2.8. zespół Interfejsów, Kontroler Dysku - 1
- 2.9. zespół Interfejsów, Kontroler Dysku - 2
- 2.10. zespół Interfejsów, Kontroler Magistrali
- 2.11. Kontroler Klawiatury i drukarki
- 2.12. Kontroler RS 232
- 2.13. Klawiatura
- 2.14. Interfejs JEC - 625 - arkusz 1
- 2.15. Interfejs JEC - 625 - arkusz 2

Część 3.

Wykaz elementów, Schematy montażowe KARAT-2. /Wykazy elementów oddzielnie wg numeracji elementów na schematach montażowych i oddzielnie wg asortymentu/.

- 3.1. Wykaz elementów płytki CPU i schemat montażowy.
- 3.2. Wykaz elementów płytki V-2 i schemat montażowy.
- 3.3. Wykaz elementów płytki INT-2 i schemat montażowy.
- 3.4. Wykaz elementów płytki "KLAWIATURA" i schemat montażowy.
- 3.5. Wykaz elementów płytki V-24/Mag
- 3.6. Wykaz elementów płytki IEC-625
- 3.7. Przetwornik W-1 i schemat montażowy.

Część 4.

KARAT-2 - dokumentacja mechaniczna.

1. Obudowa kompl. nr rys. B-02M-00.00
2. Płytki, nr rys. D-ML-00-16

3. Pokrywa dolna, nr rys. C-02M-01.01
4. Płytką przednia, nr rys. D-ML-00-11
5. Płytką tylną, nr rys. C-02M-01.02
6. Wspornik, nr rys. D-02M-01.03
7. Kątownik, nr rys. D-ML-00-15
8. Pokrywa górna, nr rys. C-02M-02.00
9. Ceownik, nr rys. D-ML-00-13
10. Uchwyt, nr rys. D-ML-00-14
11. Płytką ozdobną, nr rys. D-02M-03.00
12. Wspornik, nr rys. D-02M-04.00
13. Wspornik kompl., nr rys. D-02M-05.00
14. Wspornik, nr rys. C-02M-05.01
15. Ceownik, nr rys. D-02M-05.02
16. Kątownik, nr rys. D-ML-00.22
17. Kątownik, nr rys. D-02M-06.00
18. Kątownik, nr rys. D-02M-07.00
19. Klawiatura-zespół, nr rys. B-ML-01.00
20. Pokrywa górna kompl. nr rys. C-ML-01.01
21. Zetownik, nr rys. D-ML-01.07
22. Element U,C, nr rys. D-ML-01.09
23. Kątownik, nr rys. D-ML-01.08
24. Zetownik. nr rys. D-ML-01.10
25. Pokrywa dolna kompl. nr rys. C-ML-01.02
26. Ściana boczna, nr rys. D-ML-01.05

Część 5.

Instrukcja obsługi pakietów programowych systemu oprogramowania dedykowanego dla chromatografii CHOMIK.

1. Instrukcja obsługi programu DANE /10 str. tekstu + 10 str. załącznik/.
2. Program grafiki postprocesowej GR63, szczegółowy opis funkcjonalny.
3. Opis obsługi modułu czasu rzeczywistego S3.
4. Opis obsługi programów obliczeń specjalistycznych.

- 4.1. Program obliczenia danych dla analizy jakościowej AJO.
- 4.2. Program obliczenia danych dla analizy ilościowej AJO.
- 4.3. Program obliczenia kolumny i aparatury SPO.
- 4.4. Program wspomaganie analizy ilościowej i obliczeń regresji REG.
5. Opis modułu MENU-KATALOG.

Część 6.

Wydruki oprogramowania systemu rejestracji i przetwarzania danych do chromatografii cieczowej CHOMIK.

1. Pakiet programowy DANE - 59 str. + 16 strn
2. Pakiet programowy S3
3. Pakiet programowy GB63
4. Pakiet programowy REG
5. Pakiet programowy AJO
6. Pakiet programowy AJO
7. Pakiet programowy SPO
8. Pakiet programowy CHOMIK /MENU-KATALOG - CPM 2.2 m/.

POLITECHNIKA GDAŃSKA

INSTYTUT CHEMII I TECHNOLOGII NIEORGANICZNEJ ORAZ KOROZJI

Zespół N-B Chromatografii Cieczowej

Umowa 870119

CPBP 12.3 cel 7

Część I

DOKUMENTACJA TECHNICZNA CZĘŚCI SPRZĘTOWEJ SYSTEMU
REJESTRACJI I PRZETWARZANIA DANYCH DO CHROMATOGRAFII

KARAT 2

Część II

DOKUMENTACJA OPROGRAMOWANIA SYSTEMU REJESTRACJI
I PRZETWARZANIA DANYCH DO CHROMATOGRAFII CIECZOWEJ

CHOMIK

- do użytku wewnętrznego -

Autorzy:

dr inż. Marian Kamiński

dr inż. Krystyna Gazda

mgr inż. Andrzej Adler

mgr inż. Andrzej Lech

Artur Kowalski

mgr inż. Bogdan Kandybowicz

mgr Andrzej Zapałowski

Piotr Przestrzelski

Kierownik Zespołu N-B
Chromatografii Cieczowej

Prof.dr hab.inż. J. Kowalczyk

Gdańsk 1989

SPIS TRESCI.

Część 1.

Schemat blokowy i instrukcja obsługi KARAT-2.

CZĘŚĆ 2.

Schematy ideowe KARAT-2.

- 2.1. zespół CPU
- 2.2. zespół CPU DRAM
- 2.3. zespół CPU, CMOS RAM
- 2.4. zespół VIDEO, Układy synchronizacji
- 2.5. zespół VIDEO, Kontroler DRAM
- 2.6. zespół VIDEO, Układy dostępu DRAM
- 2.7. zespół VIDEO, Układy pamięci DRAM
- 2.8. zespół Interfejsów, Kontroler Dysku - 1
- 2.9. zespół Interfejsów, Kontroler Dysku - 2
- 2.10. zespół Interfejsów, Kontroler Magistrali
- 2.11. Kontroler Klawiatury i drukarki
- 2.12. Kontroler RS 232
- 2.13. Klawiatura
- 2.14. Interfejs JEC - 625 - arkusz 1
- 2.15. Interfejs JEC - 625 - arkusz 2

Część 3.

Wykaz elementów, Schematy montażowe KARAT-2. /Wykazy elementów oddzielnie wg numeracji elementów na schematach montażowych i oddzielnie wg asortymentu/.

- 3.1. Wykaz elementów płytki CPU i schemat montażowy.
- 3.2. Wykaz elementów płytki V-2 i schemat montażowy.
- 3.3. Wykaz elementów płytki INT-2 i schemat montażowy.
- 3.4. Wykaz elementów płytki "KLAWIATURA" i schemat montażowy.
- 3.5. Wykaz elementów płytki V-24/Mag
- 3.6. Wykaz elementów płytki IEC-625
- 3.7. Przetwornik W-1 i schemat montażowy.

Część 4.

KARAT-2 - dokumentacja mechaniczna.

1. Obudowa kompl. nr rys. B-02M-00.00
2. Płytki, nr rys. D-ML-00-16

3. Pokrywa dolna, nr rys. C-02M-01.01
4. Płytką przednia, nr rys. D-ML-00-11
5. Płyta tylna, nr rys. C-02M-01.02
6. Wspornik, nr rys. D-02M-01.03
7. Kątownik, nr rys. D-ML-00-15
8. Pokrywa górna, nr rys. C-02M-02.00
9. Ceownik, nr rys. D-ML-00-13
10. Uchwyt, nr rys. D-ML-00-14
11. Płyta czołowa, nr rys. D-02M-03.00
12. Wspornik, nr rys. D-02M-04.00
13. Wspornik kompl., nr rys. D-02M-05.00
14. Wspornik, nr rys. C-02M-05.01
15. Ceownik, nr rys. D-02M-05.02
16. Kątownik, nr rys. D-ML-00.22
17. Kątownik, nr rys. D-02M-06.00
18. Kątownik, nr rys. D-02M-07.00
19. Klawiatura-zespół, nr rys. B-ML-01.00
20. Pokrywa górna kompl. nr rys. C-ML-01.01
21. Zetownik, nr rys. D-ML-01.07
22. Element U,C, nr rys. D-ML-01.09
23. Kątownik, nr rys. D-ML-01.08
24. Zetownik. nr rys. D-ML-01.10
25. Pokrywa dolna kompl. nr rys. C-ML-01.02
26. Sciana boczna, nr rys. D-ML-01.05

Część 5.

Instrukcja obsługi pakietów programowych systemu oprogramowania dedykowanego dla chromatografii CHOMIK.

1. Instrukcja obsługi programu DANE /10 str. tekstu + 10 str. załącznik/.
2. Program grafiki postprocesowej GR63, szczegółowy opis funkcjonalny.
3. Opis obsługi modułu czasu rzeczywistego S3.
4. Opis obsługi programów obliczeń specjalistycznych.

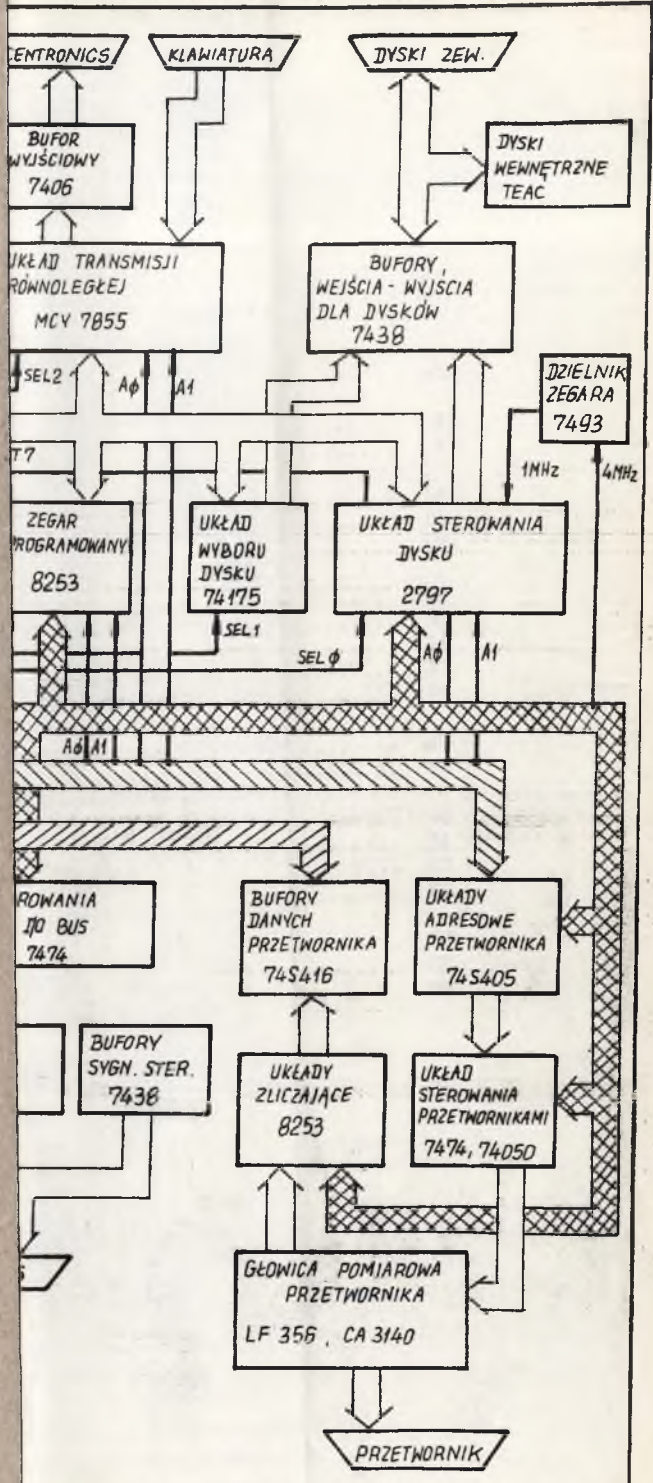
- 4.1. Program obliczenia danych dla analizy jakościowej AJO.
 - 4.2. Program obliczenia danych dla analizy ilościowej AJO.
 - 4.3. Program obliczenia kolumny i aparatury SPO.
 - 4.4. Program wspomaganie analizy ilościowej i obliczeń regresji REG.
5. Opis modułu MENU-KATALOG.

Część 6.

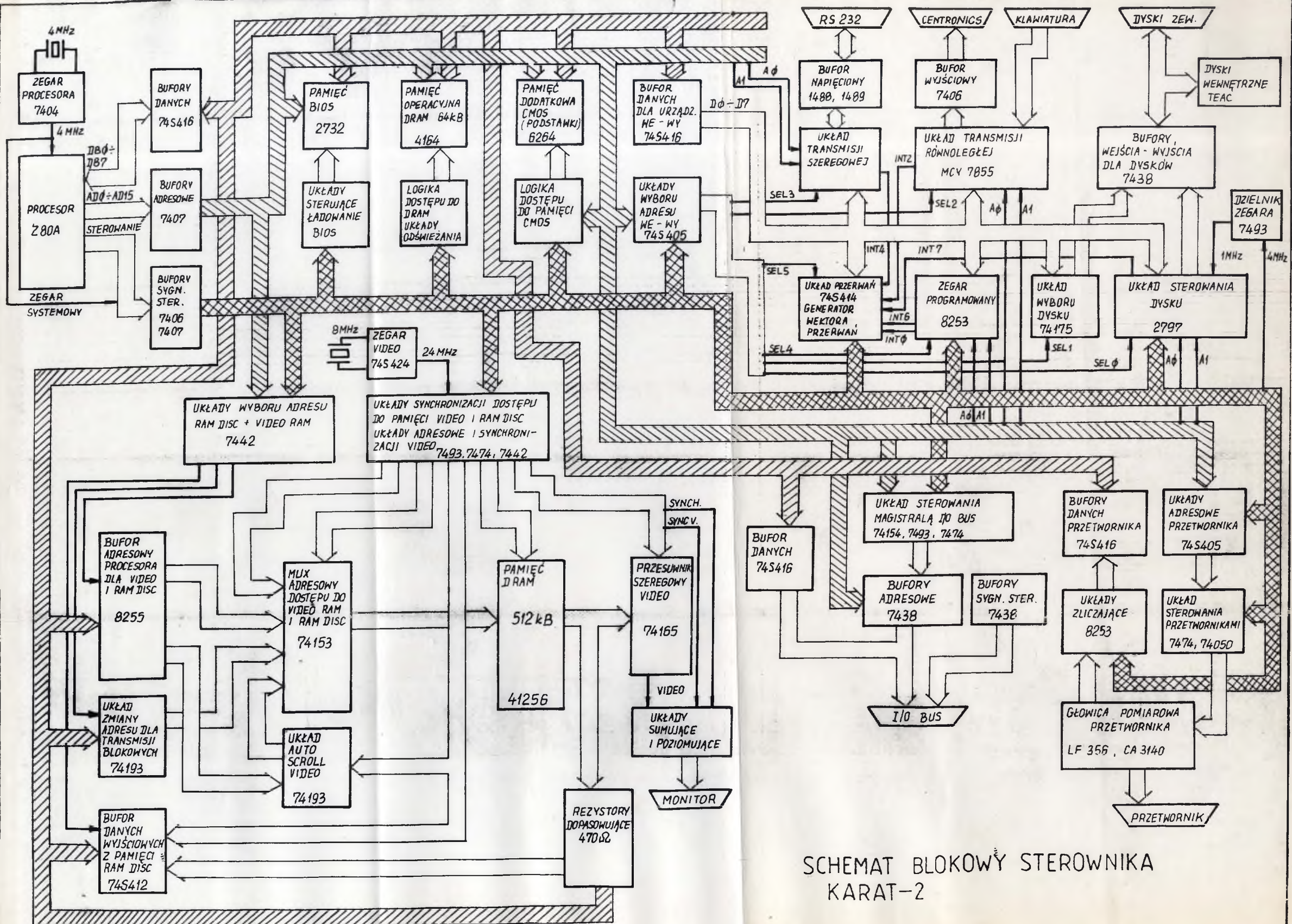
Wydruki oprogramowania systemu rejestracji i przetwarzania danych do chromatografii cieczowej CHOMIK.

1. Pakiet programowy DANE - 59 str. + 16 str.
2. Pakiet programowy S3
3. Pakiet programowy GR63
4. Pakiet programowy REG
5. Pakiet programowy AJO
6. Pakiet programowy AJO
7. Pakiet programowy SPO
8. Pakiet programowy CHOMIK /MENU-KATALOG - CPM 2.2 m/.

SCHEMAT
BLOKOWY
INSTRUKCJA
OBSŁUGI
KARAT-2



UKŁAD BLOKOWY STEROWNIKA



SCHEMAT BLOKOWY STEROWNIKA KARAT-2

Spis treści

	str
1. Opis obsługi	3
2. Opis sprzętu	4
3. <u>System operacyjny CP/M</u>	
CP/M 2.2 sterownika KARAT-2	14
- komendy rezydentne	16
- komendy nierezydentne	18
4. <u>Oprogramowanie narzędziowe</u>	
Pakiet programów assemblerowych	26
- kompilator M80	38
- program łączący L80	41
- bibliotekarz LIB80	43
- program wydruku odwołań CREF80	44
5. <u>Programy typu "debugger"</u>	
- ZDEB	45
- DET	46
6. <u>Programy "UTILITY"</u>	
- program D	48
- program FORMAT	51
- program FINDER	52
- program DIFER	52
7. <u>Edytory tekstu</u>	
- edytor EDI	53
- indeks komand WordStara	55
- kody rozkazów edytora Turbo Pascala	59

Opis obsługi

Po włączeniu sterownika KARAT na ekranie monitora pojawi się komunikat:

KARAT 2

ELKOR '87

Graphic kernel system version 3.72

Loading CP/M system from floppy drive

Jeżeli w kieszeni napędu dysków elastycznych nie zostanie włożona dyskietka z systemem CP/M na ekranie pojawi się informacja :

Insert system disk and press
any key to resume.

Po naciśnięciu dowolnego klawisza na ekranie pojawi się informacja o ładowaniu systemu :

Loading CP/M system from floppy drive

:::::::

Po poprawnym załadowaniu systemu CP/M na ekranie pojawi się:

CP/M 2.2 54 kb TPA

Created 87.09.17 Gdansk

RamDisk BIOS version 3.7

A>_

Zgłosi się system CP/M 2.2 (Control Program for Microcomputers - Program Sterujący dla Mikrokomputerów). Stanowi on powszechnie używany system operacyjny wykorzystywany na mikrokomputerach opartych o układy 8080, 8085, Z80. Zapewnia on pracę interakcyjną z systemem za pośrednictwem zestawu rozkazów. Obsługiwać może takie zadania jak ładowanie i uruchamianie programów, tworzenie i usuwanie zbiorów i.t.p. Funkcje te są opisane w dalszej części instrukcji -"System operacyjny CP/M 2.2".

Podobne zadania spełnia program narzędziowy D, którego działanie opisane jest w części instrukcji -Programy "UTILITY".

STEROWNIK KARAT-2 Opis sprzetu

```

*****
*          Z l a c z e   s y s t e m u   Z80 EC          *
*          a I          I b I          *
*****
I  1  I +5V          I  1  I +5V          I
-----
I  2  I 0V          I  2  I 0V          I
-----
I  3  I          I  3  I          I
-----
I  4  I D3          I  4  I D7          I
-----
I  5  I D2          I  5  I D6          I
-----
I  6  I D1          I  6  I D5          I
-----
I  7  I D0          I  7  I D4          I
-----
I  8  I A7          I  8  I A15          I
-----
I  9  I A6          I  9  I A14          I
-----
I 10  I A5          I 10  I A13          I
-----
I 11  I A4          I 11  I A12          I
-----
I 12  I A3          I 12  I A11          I
-----
I 13  I A2          I 13  I A10          I
-----
I 14  I A1          I 14  I A9          I
-----
I 15  I A0          I 15  I A8          I
-----
I 16  I WR.L          I 16  I RD.L          I
-----
I 17  I IORQ.L          I 17  I MEMRQ.L          I
-----
I 18  I (A16)*          I 18  I (A17)*          I
-----
I 19  I RFSH.L          I 19  I (A18)*          I
-----
I 20  I M1.L          I 20  I (A19)*          I
-----
I 21  I BUSACK.L          I 21  I BUSRQ.L          I
-----
I 22  I INTACK.L          I 22  I INT.L          I
-----
I 23  I WAIT.L          I 23  I NMI.L          I
-----
I 24  I EXRESET.L          I 24  I RESET.L          I
-----
I 25  I CLOCK.H Z80          I 25  I BOOT.H OUT          I
-----

```


STEROWNIK KARAT-A Opis Sprzetu

I 26	I EO.H	I 26	I EI.H
I 27	I	I 27	I
I 28	I	I 28	I
I 29	I	I 29	I
I 30	I	I 30	I
I 31	I	I 31	I
I 32	I	I 32	I

CROSS DYSKU

I 1	I MASA	I 1	I 8	I 1A,2A,3A
I 2	I DX2	I 2	I 3	I 7B
I 3	I DX1	I 3	I 11	I 6B
I 4	I DX3	I 4	I 13	I 16B
I 5	I RD	I 5	I 15	I 15B
I 6	I MOTOR	I 6	I 5	I 8B
I 7	I DX0	I 7	I 12	I 5B
I 8	I WE	I 8	I 2	I 12B
I 9	I READY	I 9	I 7	I 17B
I 10	I ST	I 10	I 4	I 10B
I 11	I T0	I 11	I 14	I 13B
I 12	I IX	I 12	I 9	I 4B
I 13	I WP	I 13	I 1	I 14B
I 14	I SD	I 14	I 10	I 9B
I 15	I WD	I 15	I 6	I 11B

CROSS CENTRONICS

```

*****
*      I      I      I      *
* LP  I SYGNAL  IWWP15 ICANNON*
*      I      I      I 15  *
*****
I  1  I BUSY    I  1  I 11  I
I-----I-----I-----I
I  2  I P.END   I  2  I 12  I
I-----I-----I-----I
I  3  I SELECT  I  3  I 13  I
I-----I-----I-----I
I  4  I 0V7     I  4  I19..25I
I-----I-----I-----I
I  5  I D1      I  5  I  3  I
I-----I-----I-----I
I  6  I D7      I  6  I  9  I
I-----I-----I-----I
I  7  I ACK     I  7  I 10  I
I-----I-----I-----I
I  8  I D5      I  8  I  7  I
I-----I-----I-----I
I  9  I D3      I  9  I  5  I
I-----I-----I-----I
I 10  I D0      I 10  I  2  I
I-----I-----I-----I
I 11  I D6      I 11  I  8  I
I-----I-----I-----I
I 12  I D4      I 12  I  6  I
I-----I-----I-----I
I 13  I D2      I 13  I  4  I
I-----I-----I-----I
I 14  I STROBE  I 14  I  1  I
I-----I-----I-----I
I 15  I RESET.L I 15  I 16  I
*****

```


CROSS Klawiatury

```

*****
*           I           I           I           *
* LP 3 I SYGNAL 4 I WWP15 I CANNON*
*           I           I           I 15 *
*****
I 1 I PB2 I 1 I 3 I
I-----I-----I-----I-----I
I 2 I PB3 I 2 I 4 I
I-----I-----I-----I-----I
I 3 I PB4 I 3 I 5 I
I-----I-----I-----I-----I
I 4 I PB5 I 4 I 6 I
I-----I-----I-----I-----I
I 5 I PB6 I 5 I 7 I
I-----I-----I-----I-----I
I 6 I PB7 I 6 I 8 I
I-----I-----I-----I-----I
I 7 I +5V I 7 I 12 I
I-----I-----I-----I-----I
I 8 I PB1 I 8 I 2 I
I-----I-----I-----I-----I
I 9 I PB0 I 9 I 1 I
I-----I-----I-----I-----I
I 10 I STB.L I 10 I 11 I
I-----I-----I-----I-----I
I 11 I IBF.L I 11 I I
I-----I-----I-----I-----I
I 12 I 0V I 12 I 15 I
I-----I-----I-----I-----I
I 13 I I I I I
I-----I-----I-----I-----I
I 14 I I I I I
I-----I-----I-----I-----I
I 15 I I I I I
*****
    
```

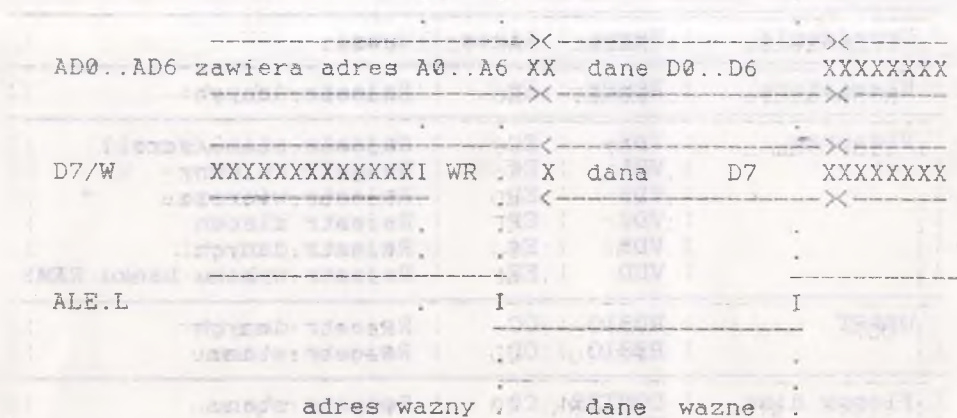

Z80 EC/IOBUS Złącze 15 stykowe

IOBUS jest multipleksowana magistrala wejścia wyjścia o dużej predkosci pracy przeznaczona do dolaczania urzadzen oddalonych do magistrali procesora Z80. Magistrala ta jest uaktywniana w czasie wykonywania przez CPU instrukcji wejścia wyjścia z adresem portu <128 (tzn 00..7FH). Znaczenie sygnalow podano ponizej.

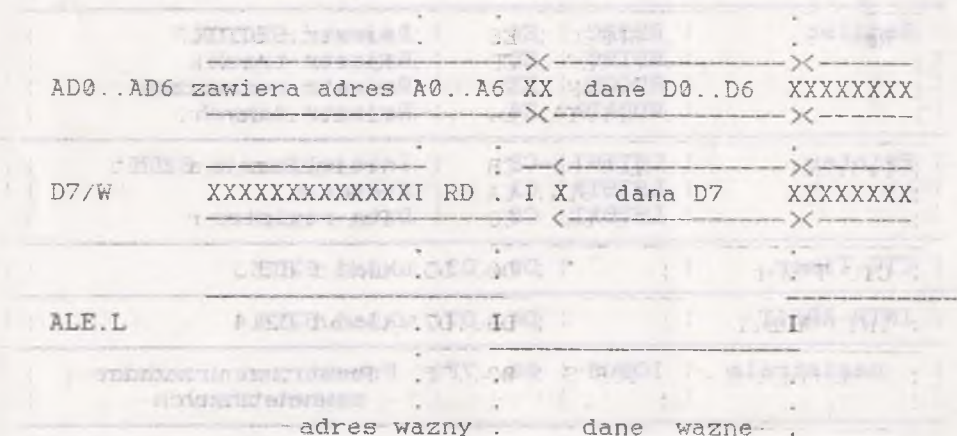
I 1	I AD0	I	AD0..AD6,D7/W
I 2	I AD1	I	Linie adresowe/dane
I 3	I AD2	I	
I 4	I AD3	I	
I 5	I AD4	I	
I 6	I AD5	I	
I 7	I AD6	I	
I 8	I D7/W H=Write L=Read	I	
I 9	I 0V Signal ground	I	Masa sygnalowa
I 10	I ALEI HL=Adres LH=Data	I	Sygnal synchronizacji danych
I 11	I Reset.H	I	Sygnal zerowania systemu
I 12	I INTACK.L in/out	I	Sygnal potw. przerwanania
I 13	I INTRQ.L	I	Sygnal przerwanania do CPU
I 14	I not used	I	
I 15	I not used	I	

STEROWNIK KARAT-290 (nie! sprzętu)

Zapis danych do urządzenia przez CPU np.: OUT (DEV1), A
(DEV1 < 80H adres urządzenia na IOBUS)



Odczyt danych z urządzenia przez CPU np.: IN A, (DEV1)
(DEV1 < 80H adres urządzenia na IOBUS)



System przerwaniowy.

Urządzenie zadające przerwania wystawia stan log 0 na linii INTRQ.L. Po przyjęciu przerwania przez procesor urządzenie otrzymuje sygnał INTACK.L W czasie stanu niskiego tego sygnału urządzenie musi wystawić na linii AD0..AD6, D7/W wektor przerwania. Zmiana stanu sygnału INTACK.L na wysoki oznacza przyjęcie przez procesor wektora.

Każde urządzenie posiada gniazdo i wtyk IOBUS. Na gnieździe sygnał INTACK.L jest sygnałem wejściowym dla urządzenia. Urządzenie jeżeli nie wystawia zadania przerwania zobowiązane jest przekazywać stan sygnału na wtyk wyjściowy.

STEROWNIK KARAT-1 opis sprzętu

Przestrzen urzadzen wejscia wyjscia komputera "KARAT-1"

Urządzenie	Nazwa	Adres	uwagi
Klawiatura	RDKEY	E8	Rejestr danych
Videoram	VDX	EC	Rejestr stanu/scroll
	VDA	E6	Rejestr kolumny
	VDB	ED	Rejestr wiersza
	VDC	EF	Rejestr zleceń
	VDR	E4	Rejestr danych
	VDD	EE	Rejestr wyboru banku RAM
USART	RDSIO	CC	Rejestr danych
	RSSIO	CD	Rejestr stanu
Floppy disc	CONTRR	C0	Rejestr stanu
	TRACKR	C1	Rejestr trk
	SECTR	C2	Rejestr sec
	DATAR	C3	Rejestr danych
	DIPORT	C4..C7	Rejestr MOTOR/seldx
Ramdisc	RDSEC	E6	Rejestr SECTOR
	RDTRC	ED	Rejestr track
	RDCOM	EE	Rejestr rozkazowy
	RDDATA	E4	Rejestr danych
Printer	LSTINI	CB	Inicjalizacja 8255
	LSTSTA	CA	Status
	LSTDAT	C8	Data register
CTC Timer		D0..D3	uklad 8253
INTR ARBIT		D4..D7	uklad 8214
magistrala	IOBUS	00..7F	Przestrzen urzadzen zewnetrznych

Wektory przerwan (wzrostajacy Priorytet)

E0	- Floppy DRQ	0
E2	- CTC 2 (MASTER)	1
E4	- SERIAL OUT	7
E6	- SERIAL IN	5
E8	- LIST ACK	6
EA	- KEYBOARD DATA READY	2
EC	- CTC 3 (SLAVE)	3
EE	- CTC 1 (BEEP)	4

Z80 pracuje w trybie IM 2 i rejestr przerwan (I) zawiera 0FAH

Oprogramowanie podstawowe

Alokacja pamięci dla systemu MTS 3.46

0D800..0EAF0 CP/M BDOS/BIOS
0EB00..0ECFF CP/M Working RAM
0ED00..0FA00 Basic Device Handling Program BDHP
0FA00..0FAFF interup vectors RAM
0FB00..0FBFF subsytem variable and working RAM
0FC00..0FFFF disk DMA SEC/buffer 512b

Po sygnale reset program bootstrap przepisuje z ROM programy CCSP i BDHP na odpowiednie adresy pamięci i przekazuje sterowanie do CCSP.

BDHP = rezydentny modul obsługi systemowych urządzeń WE-WY

BDHP jest modulem obsługującym urządzenia komputera na poziomie "sterowania elektrycznego". Modul ten jest nierozdzielny i zawiera oprogramowanie podstawowe urządzeń wejścia wyjścia komputera. W szczególności pamięci dyskowej, ram-disk, klawiatura, pamięć taśmowa, video-terminal, drukarka, łącz RS232C transmisji szeregowej. Wszystkie programy systemowe korzystają z tego modulu w zunifikowany sposób poprzez adresy łącznikowe i szereg zleceń systemowych. Komórki pamięci o adresie 0DH, 0EH zawierają adres początkowy modulu BDHP. Dostęp do zleceń systemowych możliwy jest przez użycie instrukcji Call 0CH na podobnych zasadach jak korzystanie z systemowych zleceń BDOS w systemie CP/m rejestr C zawiera numer zlecenia.

Zlecenia podstawowe we-wy

0	TIOSYS	Reset system
1	INCHAR	Wczytanie znaku z klawiatury
2	BCO	Wyswietlenie znaku na ekranie
3	V24IN	Odebranie znaku z łącza RS232C (V24)
4	V24OUT	Wyslanie znaku na łącze RS232C (V24)
5	LSTOUT	Wyslanie znaku na drukarke
6	DIRIO	Besposredni dostep do konsoli
7	RDY	Odczyt sektora z dysku
8	WRDY	Zapis sektora na dysk
9	PMSDE	Wyprowadzenie lancucha znakow na konsole
A	NIL	Zarezerwowane
B	TCHIN	Testowanie stanu klawiatury
C	V24IST	Testowanie stanu odbiornika RS232C
D	V24OST	Testowanie stanu nadajnika RS232C
E	LSTST	Testowanie stanu drukarki
F	TIOTAB	Pobranie adresu tablicy zleceń systemowych
10	VFONT	Pobranie adresu wskaźnika generatora znakow
11	VSAVE	Bezposredni dostep do ekranu
12	TIOVER	Pobranie wersji systemu BDHP

Oprogramowanie podstawowe

Sekwencje sterujące wyświetlaniem znaków na ekranie:

kody jednoznakowe:

mnemonik	kod	funkcja
BEL	07	Dzwonek (ignorowany)
BS	08	cofniecie kursora o jeden znak (dla pierwszej kolumny ignorowany)
VT	09	tabulacja pozioma (następny znak jest numerem kolumny)
LF	0A	przejscie kursora o jedna linie w dol
HT	0B	tabulacja pionowa (następny znak zawiera nr wiersza)
FF	0C	czyszczenie ekranu
CR	0D	powrot kursora do pierwszej kolumny
ESC	1B	zapowiedz sekwencji ESCAPE

kody po znaku ESC

'='	3D	pozycjonowanie kursora (następne znaki to nr kolumny+31 i nr wiersza +31)
'G'	47	znaczenie jw
'O'	4F	inwersja kolorow (biale tlo)
'P'	50	normalne kolory (znaki biale) (stan z inicjalizacji)
'L'	4C	czyszczenie do konca linii z kursorem
'M'	4D	czyszczenie do konca ekranu od kursora
'J'	4A	wstawienie linii na pozycji kursora
'N'	4E	usuniecie linii z kursorem
'U'	55	następny znak bedzie wyswietlany bez interpretacji
'X'	58	następny znak zawiera szybkość migotania kursora
'@'	40	reset systemu video
'v'	76	zwijanie (kursor po dojsci do ostatniej kolumny przechodzi do pierwszej w wierszu nizej)
'w'	77	koniec zwijania (kursor po dojsci do ostatniej kolumny nie przechodzi dalej i znaki ulegaja nałożeniu na siebie) (stan z inicjalizacji)

Ekran zawiera 32 wiersze po 64 znaki.

Po przejściu kursora do ostatniego wiersza i wymuszeniu rozkazu LF następuje przesunięcie (scroll) ekranu o jeden wiersz w górę i wyczyszczenie nowego ostatniego wiersza.

System operacyjny CP/M 2.2

CP/M 2.2 STEROWNIKA KARAT-2

System operacyjny sterownika KARAT-2 dostarczany jest użytkownikowi na dyskietce systemowej. Pamięć operacyjna mikrokomputera pracującego pod nadzorem tego systemu dzielona jest na kilka obszarów, przy czym dla programów użytkowych przeznaczonych jest 60KB (tzn. 60x1024 bajty) pamięci.

System CP/M 2.2 zawiera dwa rodzaje komend:

- Komendy rezydentne, tj. takie, które po załadowaniu systemu przechowywane są w pamięci, mogą być więc realizowane natychmiast bez konieczności ładowania. Komendy rezydentne mogą zostać użyte przez napisanie nazwy rozkazu po symbolu zgłoszenia się systemu :A>
- Komendy nierezydentne, czyli programy systemowe przechowywane na dysku. Dzieje się tak dlatego, że część systemu mieszcząca się na stałe w pamięci nie może być zbyt obszerna, gdyż rozszerzenie systemu zmniejsza część pamięci dostępnej dla programów użytkownika. Dlatego też duża część funkcji spełnianych przez system realizowana jest w pamięci programów użytkowych ładowanych do pamięci jedynie na czas ich wykonania.

Programy systemowe oraz programy użytkowe identyfikowane są za pomocą nazw. Nazwa zbioru dyskowego składa się z dwóch części rozdzielonych kropką. Kropka nie może być wykorzystana w nazwie w żadnym innym kontekście. Pierwsza część, która może składać się z co najwyżej 8 znaków alfanumerycznych, to podstawowa część nazwy. Nazwa może zawierać dowolne znaki poza: "?", "*", "-", ":", ";", "=", ":", ".". Druga część to typ zbioru zawierający od 0 do 3 znaków alfanumerycznych. Typ zbioru może być dowolny, można go też pominąć. Jednak system i programy systemowe używają zbiorów lub tworzą zbiory następujących typów:

- .MAC -źródłowy tekst programu w języku makroasemblera
- .ASM -program źródłowy w języku asemblera
- .XRF -słownik asemblera w adresach bezwzględnych
- .CRF -słownik asemblera w adresach względnych
- .REL -moduł programu przemieszczalnego dla programu L80
- .COM -zbiór wyników do bezpośredniej realizacji
- .HEX -zbiór wyników w postaci szesnastkowej
- .PRN -raport z kompilacji
- .SYM -zbiór z lista symboli z kompilacji
- .BAS -program w języku BASIC
- .INT -program wyników w języku BASIC
- .IST -listing w języku BASIC
- .DAT -dane w kodzie ASCII
- .SUB -zbiór instrukcji programu SUBMIT
- .PAS -program źródłowy w języku PASCAL
- .COB -program źródłowy w języku COBOL
- .OVR -nakładka kompilatora COBOL-u
- .FOR -program źródłowy w języku FORTRAN
- .BAK -kopia zbioru pierwotnego tworzona automatycznie przez edytor
- .\$\$\$ -tymczasowe zbiory robocze

System operacyjny CP/M 2.2

Przy określaniu grup zbiorów można stosować konwencje "*" i "?". Gwiazdka oznacza dowolną nazwę lub typ, może też stanowić uzupełnienie do osmiu znaków nazwy lub trzech znaków typu. Znak zapytania zastępuje na danej pozycji dowolny dozwolony znak.

Przykłady:

- X??*.COM - zapis ten oznacza zbiory typu COM zaczynające się na literę X, po której następują dwa dowolne znaki
- X*.COM - zapis ten oznacza zbiory typu COM zaczynające się na literę X, po której następuje siedem dowolnych znaków /dopełnienie do osmiu znaków w nazwie/
- *.COM - zapis oznacza wszystkie zbiory typu COM znajdujące się na dyskietce
- *.* - zapis oznacza wszystkie zbiory o dowolnej nazwie i dowolnym typie znajdujące się na dyskietce

Przyjęto zasadę, w myśl której jednostki dyskowe oznaczane są literami od A do P. Aby wskazać jednostkę dyskową określa się ją przez podanie literowego identyfikatora i znaku ":".

Przy konwersacji z systemem mamy do dyspozycji następujące klawisze funkcyjne:

- CTRL-C - Restart systemu operacyjnego.
- CTRL-H;BS - Kasuje znak na lewo od kursora i przesuwa kursor o jedno miejsce w lewo. Znaki na prawo od kursora są przesuwane o jedno miejsce w lewo.
- CTRL-X - Kasuje wszystkie znaki na lewo od kursora i cofa kursor na początek wiersza.
- CTRL-U - Kasuje wszystkie znaki w wierszu, umieszcza znak "#" na pozycji kursora i ustawia kursor na początku następnej linii.
- TAB - Przesuwa kursor do wielokrotności 8 kolumny.
- CTRL-E - Powoduje zmianę wiersza bez przekazania rozkazu do CP/M. Umieszcza kursor na początku następnego wiersza nie kasując dotychczas wprowadzonych informacji.
- CTRL-P - kopiaj/nie kopiaj na drukarkę (przelacznik).
- CTRL-S - wstrzymaj wyprowadzanie na CRT (kontynuacja po naciśnięciu dowolnego klawisza)
- RETURN - przekazanie dyspozycji do systemu
- CTRL-J - Spełnia identyczną funkcję co klawisz RETURN.
- CTRL-M - Spełnia identyczną funkcję co klawisz RETURN lub CTRL-J.
- CTRL-R - Powtarza wiersz do kursora w następnym wierszu. Na pozycji kursora pojawia się znak "#". Ten kursor przesuwany jest o jeden wiersz w dół tej samej kolumnie.

System operacyjny CP/M 2.2

KOMENDY REZYDENTNE CP/M 2.2

ERA

ERA -Rozkaz /ERASE - wymaz/ sluzzy do usuwania jednego lub kilku zbiorow z biezacego lub podanego dysku. Wykonanie rozkazu powoduje usuniecie informacji o zbiorze z katalogu, a miejsce zajmowane przez zbior moze byc ponownie uzyte.

Przyklady:

ERA ZB1.MAC -Usun zbior o nazwie ZB1 i rozszerzeniu MAC z dysku biezacego.
ERA B:ZB1.MAC -Usun zbior o nazwie ZB1 i rozszerzeniu MAC z danego dysku.
ERA Z???.MAC -Usun zbiory o rozszerzeniu MAC i nazwach zaczynajacych sie na litere Z i majacych nie wiecej niz 3 znaki w nazwie.
ERA *.COM -Usun wszystkie zbiory typu COM z dysku biezacego.
ERA ZB1.* -Usun wszystkie zbiory o nazwie ZB1 i dowolnym rozszerzeniu z dysku biezacego.
ERA *.* -Usun wszystkie zbiory z dysku biezacego o dowolnej nazwie i dowolnym rozszerzeniu. Tego typu rozkaz powoduje, ze CP/M zgłasza sie dodatkowym komunikatem ALL (Y/N)?, zada dodatkowego potwierdzenia wykonania tego rozkazu.

DIR

DIR -Rozkaz /DIRECTORY - katalog/ sluzzy do wyswietlania na ekranie zawartosci katalogu z biezacego lub podanego dysku. Zaleznie od potrzeb mozna wyswietlic pelna zawartosc katalogu lub jego czesc.

Przyklady:

DIR -wyswietl katalog zbiorow znajdujacych sie na aktualnie przylaczonej jednostce dyskowej
DIR *.* -j.w.
DIR B: -wyswietl katalog zbiorow znajdujacych sie na jednostce dyskowej B
DIR XYZ?????.COM -wyswietl nazwy wszystkich zbiorow typu COM, ktorych nazwa podstawowa zaczyna sie na XYZ
DIR XYZ*.COM -j.w.
DIR *.COM -wyswietl wszystkie zbiory o rozszerzeniu COM
DIR ABC.* -wyswietl wszystkie zbiory o nazwie ABC i rozszerzeniu COM

REN

REN -Rozkaz /RENAME - zmien nazwe/ pozwala zmienic nazwe zbioru bez potrzeby kopiowania zbioru. Zmiana dotyczy wylacznie katalogu. Kolejno nalezy podac nowa nazwe, a po niej nazwe stara oddzielajac je znakiem "-". Operacji mozna dokonac na biezacym lub podanym dysku. W przypadku, gdy nowa nazwa istnieje w katalogu, operacja nie zostanie wykonana i CP/M zglosi sie komunikatem: FILE EXISTS

Przykłady: REN NOWA.ABC=STARA.XYZ -zbiór STARA.XYZ otrzymuje nazwę NOWA.ABC
 REN A.MAC=A.BAK -zbiór A.BAK otrzymuje nazwę A.MAC

SAVE

SAVE -Rozkaz /SAVE - przechowaj/ służy do zapisania zawartości fragmentu pamięci w postaci zbioru na dysku. Powoduje zapisanie określonej w komendzie liczby 256-bajtowych bloków pamięci /zwanych stronami/ począwszy od adresu 100H do zbioru dyskowego o podanej nazwie.

Przykład: SAVE 3 X.MEM -kopiuje pamięć od adresu 100H do adresu 3FFH do zbioru dyskowego X.MEM

TYPE

TYPE -Rozkaz /TYPE - pisz/ pozwala wyświetlić zawartość wybranego zbioru na ekranie monitora. Wyświetlany zbiór musi mieć format ASCII i być zbiorem tekstowym. W innym przypadku np. przy próbie wyświetlenia programów wynikowych, mogą powstać nieprzewidziane efekty, w skrajnym przypadku zablokowanie systemu. Zbiory muszą być podane w postaci pełnej nazwy, bez symboli "*" lub "?". Użycie funkcji kontrolnej CONTROL-S powoduje wstrzymanie wyświetlania. W ten sposób można zapobiec zbyt szybkiemu przesowaniu się tekstu na ekranie.

Przykład: TYPE C.TXT -wyprowadz na monitor zawartość zbioru C.TXT

USER

USER -Rozkaz /USER - użytkownik/ przydziela zasoby dyskowe wskazanemu użytkownikowi (oznaczonemu numerem 0 - 15)

Przykład: USER 5 -przydziela zasoby dyskowe użytkownikowi nr 5

System operacyjny CP/M 2.2

KOMENDY NIEREZYDENTNE CP/M 2.2

Komenda nierezydentna, to program systemowy ladowany do obszaru programow uzytkowych ze zbioru dyskowego o nazwie odpowiadajacej nazwie komendy i rozszerzeniu typu COM.

Program systemowy STAT

STAT -Program umożliwia uzyskanie informacji o zbiorach dyskowych, podaje i zmienia przyporządkowanie urządzeń zewnętrznych odpowiednim urządzeniom logicznym, podaje i zmienia "status" określonego zbioru lub grupy zbiorow.

Komenda o postaci: STAT

Powoduje wyprowadzenie informacji o "statusie" i wolnych obszarach na poszczególnych jednostkach dyskowych w następującej formie:

X: R/W, SPACE: nnnK

lub

X: R/O, SPACE: nnnK

gdzie

X: -jednostka dyskowa (X=A..P)

R/W -dysk, z którego dane mogą być czytane lub na który mogą być pisane

R/O -dysk, z którego dane mogą być tylko czytane

nnn -wielkość dostępnego obszaru na dysku w KB

Komenda o postaci: STAT X:

Określa wielkość wolnej pamięci na jednostce dyskowej X (X=A,B,...,P). Postać wyprowadzanej informacji jest następująca: BYTES REMAINING ON X: nnnK

gdzie: nnn-wielkość dostępnego obszaru na dysku X w KB

Komenda o postaci: STAT NAZWA.TYP

Wyprowadza rozszerzona informację o zbiorze NAZWA.TYP.

Informacja wyprowadzana ma postać:

Recs Bytes Ext Acc

rrr bbbk ee R/Y X:PPPPPPP.SSS

Bytes Remaining On X: nnnk

gdzie:

rrr -liczba 128-bajtowych rekordow

bbb -liczba KB zajmowanych przez zbior na dysku (bbb=rrrx128/1024)

ee -liczba 16 KB segmentow pamieci dyskowej zajmowanych przez zbior (ee=bbb/16)

R/Y -wskaznik dostępu do zbioru (R/Y=R/O lub R/Y=R/W)

X -nazwa jednostki dyskowej (X=A,B,...,P)

PPPPPPP -czesc podstawowa nazwy zbioru

SSS -rozszerzenie nazwy zbioru

Komenda o postaci: STAT X:NAZWA.TYP

Wyprowadza informację o zbiorze NAZWA.TYP znajdującym się na jednostce dyskowej X. Postać informacji j.w.

System operacyjny CP/M 2.2

Komenda o postaci: **STAT X:NAZWA.TYP \$R/O**

Powoduje zmianę "statusu" /wskaznika dostępu do zbioru/ zbioru NAZWA.TYP na Read Only /zbior przeznaczony tylko do czytania/.

Komenda o postaci: **STAT X:NAZWA.TYP \$R/W**

Powoduje zmianę "statusu" zbioru NAZWA.TYP na zbior przeznaczony do zapisu i odczytu.

Komenda o postaci: **STAT X:NAZWA.TYP \$SYS**

Powoduje zmianę "statusu" zbioru NAZWA.TYP na zbior systemowy. Status SYS powoduje, że nazwy zbiorów nie są przedstawiane przy wyświetlaniu katalogu dysku.

Komenda o postaci: **STAT X:NAZWA.TYP \$DIR**

Powoduje anulowanie "statusu" SYS i przywrócenie zwykłego statusu.

Komenda o postaci: **VAL:**

Tworzy przegląd dostępnych komend działających na statusach. Podaje dopuszczalne warianty przyporządkowania urządzeń fizycznych do odpowiednich typów urządzeń logicznych.

Wyświetlana jest informacja o następującej postaci:

Temp R/O Disk: d:=R/O

Set Indicator: d:filename.typ \$R/O \$R/W \$SYS \$DIR

Disk Status : DSK: d:DSK:

User Status :USR:

Iobyte Assign:

CON: = TTY: CRT: BAT: UC1:

RDR: = TTY: PTR: UR1: UR2:

PUN: = TTY: PTP: UP1: UP2:

LST: = TTY: CRT: LPT: UL1:

W systemie występują cztery urządzenia logiczne:

CON: -"konsola systemowa"

RDR: -"czytnik taśmy papierowej"

PUN: -"perforator taśmy papierowej"

LST: -"wyjściowe urządzenie listujące"

Urządzeniom fizycznym nadane zostały następujące nazwy:

TTY: -"teletype" (konsola operatorska)

CRT: -monitor telewizyjny

BAT: -urządzenie przetwarzania wsadowego (konsola jest aktualny RDR:, wyjście na aktualne LST:)

UC1: -konsola zdefiniowana przez użytkownika

PTR: -czytnik taśmy papierowej

UR1: -czytnik #1 zdefiniowany przez użytkownika

UR2: -czytnik #2 zdefiniowany przez użytkownika

PTP: -perforator taśmy papierowej

UP1: -perforator #1 zdefiniowany przez użytkownika

UP2: -perforator #2 zdefiniowany przez użytkownika

LPT: -drukarka wierszowa

UL1: -urządzenie listujące #1 zdefiniowane przez użytkownika

System operacyjny CP/M 2.2

Komenda o postaci: **STAT DEV:**

Podaje aktualnie obowiązujący sposób przyporządkowania urządzeń fizycznych urządzeniom logicznym. Przykładowo system w odpowiedzi może wyświetlić:

CON: is TTY:
RDR: is TTY:
PUN: is TTY:
LST: is TTY:

Komenda o postaci: **STAT ld1:=pd1;,ld2:=pd2;, ... ,ldn:=pdn:**
Umożliwia zmianę aktualnie obowiązującego sposobu przyporządkowania urządzeń fizycznych urządzeniom logicznym.
gdzie

ld1,ld2,...,ldn -nazwy urządzeń logicznych
pd1,pd2,...,pdn -nazwy urządzeń fizycznych

Komenda o postaci: **STAT D:DSK:**

Podaje charakterystykę dysku D:. Przykładowo system w odpowiedzi może wyświetlić:

B: Drive Characteristics
1320: 128 Byte Record Capacity
165: Kilobyte Drive Capacity
64: 32 Byte Directory Entries
64: Checked Directory Entries
128: Records/Extent
8: Records/Block
40: Sectors/Track
2: Reserved Tracks

Kolejne pozycje listy okreslaja:

- nazwe dysku
- pojemnosc dysku w 128-bajtowych rekordach
- pojemnosc dysku w KB
- liczbe pozycji katalogu
- liczbe pozycji kontrolowanych katalogu
- liczbe rekordow w "obszarze"
- liczbe rekordow w "bloku"
- liczbe sektorow na sciezce
- liczbe scizek zapasowych

Komenda o postaci: **STAT DSK:**

Podaje charakterystykę dla wszystkich przyłączanych jednostek dyskowych o postaci j.w.

Komenda o postaci: **STATUSR:**

Tworzy listę numerów użytkowników, którzy mają zbiory na aktualnie przyłączonej jednostce dyskowej. Przykładowo system w odpowiedzi może wyświetlić:

Active User :0
Active Files:0

Program systemowy PIP

PIP - Program umożliwiający ładowanie i wyprowadzanie na urządzenie zewnętrzne, kopiowanie i łączenie zbiorów dyskowych.

Komenda o postaci: PIP

Spowoduje załadowanie programu PIP do obszaru TPA i program zgłosi się znakiem "*", sygnalizującym że jest gotowy na przyjęcie komendy. Wprowadzenie linii pustej (tj. <CR> jednokrotne naciśnięcie klawisza <RETURN>) kończy działanie programu PIP. Ta forma komendy stosowana jest przy wykonywaniu większej liczby zadań.

Komenda o postaci:

PIP DESTINATION=SOURCE#1,SOURCE#2,...,SOURCE#N

Pozwala na wykonanie tylko jednej komendy. Całkowita liczba znaków w linii nie może przekraczać 255 (dla konsoli wszyskich należy użyć ^E aby wymusić fizyczny powrót karetki). Gdy w linii komendy występuje kilka zbiorów (tj. n>1) zakłada się, że zawierają znaki ASCII ze znakiem ^Z (w CP/M end-of-file) na końcu każdego zbioru.

Znaczenie składników zlecenia:

DESTINATION -zbiór lub urządzenie zewnętrzne przyjmujące dane

SOURCE#1,...,SOURCE#N -jedno lub więcej zbiorów lub urządzeń, które są kopiowane w kolejności od lewej do prawej na DESTINATION

Przykłady prawidłowych zleceń programu PIP:

A:X=B:Y -zbiór Y jednostki dyskowej B zostanie skopiowany na jednostce A pod nazwą X

X.MAC=Y.MAC,Z.MAC -stworzenie zbioru X.MAC będącego złożeniem (konkatenacją) zbiorów Y.MAC i Z.MAC na jednostce aktualnie przydzielonej do systemu.

B:W=B:X,A:Y,Z -na dysku B utworzony zostanie zbiór W będący złożeniem zbioru X z dysku B, Y z dysku A i Z z dysku bieżącego

PIP dopuszcza skróconą postać komend dla przesyłania zbiorów między napędami dyskowymi:

PIP D:=NAZWY

PIP D1:=D2:NAZWY

PIP NAZWA.TYP=D2:

PIP D1:NAZWA.TYP=D2:

Postać pierwsza kopiuje wszystkie zbiory z aktualnie przyłączonego napędu dyskowego na dysk D (NAZWY - oznacza grupę zbiorów określonych konwencjami "?" i "*"). Postać druga jest analogiczna do pierwszej z tym, że zbiory są kopiowane z dysku D2 na dysk D1. Postać trzecia kopiuje zbiór o nazwie NAZWA.TYP z dysku D2 na dysk bieżący pod tą samą nazwą. Postać czwarta jest analogiczna do pierwszej z tym, że zbiór kopiowany jest z dysku D2 na dysk D1.

System operacyjny CP/M 2.2

Przykłady:

PIP A:=B:*.* -kopiuj wszystkie zbiory z dysku B na dysk A
PIP B:=*.COM -kopiuj wszystkie zbiory z rozszerzeniem
nazwy COM z napędu dyskowego aktualnie przy-
dzie lonego do systemu na dysk B
PIP X.MAC=B: -rownowazne PIP X.MAC=B:X.MAC
PIP A:X.MAC=B: -rownowazne PIP A:X.MAC=B:X.MAC
PIP B:=Y.BAS -rownowazne PIP B:Y.BAS=Y.BAS
PIP B:=A:Y.BAS -rownowazne PIP B:Y.BAS=A:Y.BAS

PIP zezwala na bezposrednie odwolywanie sie do akceptowanych przez CP/M urzadzen logicznych i fizycznych. Nazwy tych urzadzen sa takie same jak dla komendy STAT (z pominięciem BAT). Oczywiscie urzadzenie przeznaczenia musi byc zdolne do przyjmowania danych, a urzadzenie zrodlowe do ich wysylania. Dodatkowe nazwy urzadzen logicznych, ktorzych moze uzywac PIP to:

NUL: -przeslij 40 "nulls" (0 w ASCII) do urzadzenia (np. na koncu przeslania na perforator)
EOF: -przeslij CP/M-owski end-of-file (^C w ASCII) do urzadzenia przeznaczenia (ustawiany automatycznie na koncu wszystkich przeslan znakow ASCII realizowanych przez PIP)
INP: -specjalne zrodlo wejsciu programu PIP wbudowane w sam program: PIP bierze znak po znaku poprzez odwo-
lanie sie rozkazem CALL do adresu 103H. Dane prze-
kazywane sa poprzez komorkę 109H (bit parzystosci musi byc wyzerowany).
OUT: -specjalne urzadzenie wyjsciowe programu PIP wbudo-
wane w sam program: PIP wykonuje CALL do adresu 106H z dana w rejestrze C, dla kazdego transmitowa-
nego znaku
PRN: -to samo co LST: z tym ze znaki tabulacji przyjete sa co 8-ma kolumna, wiersze sa numerowane, a nowa strona rozpoczyna sie co 60 wierszy (to samo co uzycie opcji PIP [t8np])

Przykłady:

PIP LST:=PROG.MAC -listuj zbior PROG.MAC
PIP CON:=X.MAC,Y.MAC -zlozenie zbiorow X.MAC i Y.MAC jest
wyprowadzane na konsole

Parametry komendy PIP

Uzytkownik moze okreslic jeden lub (wiecej) parametrow zawar-
tych w nawiasach kwadratowych.

- A: -Archiwum. Kopiowanie tylko tych zbiorow, ktore
zostaly zmienione od czasu ostatniego kopiowania.
- C: -Confirm-potwierdzenie. Wymaga potwierdzenia przy
kopiowaniu zbiorow.
- B: -Przeslanie blokowe: dane sa kompletowane w bufo-
rze przez program PIP az do przyjecia znaku "x-
off" (^S w ASCII) z urzadzenia zrodlowego. Pozwala
to przesylic dane do zbiorow dyskowych z urza-
dzen o ciaglym odczycie, np. pamieci kasetowej.
Po przyjeciu "x-off" PIP zeruje bufor dyskowy i
powraca do przyjmowania danych wejsciu.

System operacyjny CP/M 2.2

- Dn** -Odrzuc znaki, które "wychodzą" poza kolumnę "n" w danych przesyłanych do urządzenia przeznaczenia.
- E** -Echo. Wszystkie znaki podczas kopiowania są wysyłane również na konsolę.
- F** -Eliminuje z przesyłanego zbioru wszystkie znaki wysuwu do nowej strony (parametr P służy do wstawienia nowych "Form Feed").
- Pn** -Co każde "n" wierszy wstawić znak przejścia do nowej strony (wraz z inicjującym "Feed Form"). Jeśli n=1 lub nie występuje, znak przejścia do nowej strony jest wstawiany co 60 linii. Gdy użyto opcji F, to przed wstawieniem nowych, usuwane są istniejące znaki przejścia do nowej strony.
- Gn** -Kopiowanie do lub z obszaru użytkownika "n".
- H** -Przesłanie danych HEX: sprawdzana jest zgodność danych z formatem szesnastkowym Intela. Nieistotne znaki pomiędzy rekordami heksadecymalnymi są w trakcie kopiowania usuwane. Gdy wystąpi błąd zgłasza się konsola.
- I** -Ignorowanie rekordu ":00" w czasie przesyłania zbiorów w formacie szesnastkowym Intela (I automatycznie ustawia parametr H).
- L** -Znaki zapisane dużą literą (A-Z) są zamieniane na znaki pisane małą literą (a-z). Wszystkie inne znaki pozostają niezmiennicze.
- U** -Znaki napisane małą literą (a-z) są zamieniane na znaki pisane dużą literą (A-Z). Wszystkie inne znaki pozostają niezmiennicze.
- N** -Dodanie do każdego przesyłanego wiersza jego numeru: numerowanie rozpoczyna się od 1, kolejne numery zwiększa się o 1, nieznaczące zera są usuwane, po numerze ustawiany jest średnik. Przy wyspecyfikowaniu n2 nieznaczące zera są włączane, a po numerze wprowadzany jest znak tabulacji.
- O** -Przesłanie zbioru wyników (nie-ASCII) ignorowany jest normalny znak końca zbioru CP/M.
- Ss^Z** -Rozpocząć kopiowanie z urządzenia źródłowego po napotkaniu łańcucha s (zakonczonego ^Z).
- Qs^Z** -Zatrzymać kopiowanie z urządzenia lub zbioru źródłowego po napotkaniu łańcucha s (zakonczonego znakiem ^Z).
- Parametry Q i S mogą być używane do rozdzielania zbioru na elementy. Łańcuchy startu i zatrzymania są zawsze włączane w operacje kopiowania. Użycie dyrektywy w postaci: PIP `linia komendy` spowoduje przetłumaczenie łańcuchów po parametrach Q i S na duże litery.
- R** -Czytaj zbiory o statusie "systemowe".
- Tn** -Przesunięcie znaków tabulacji (^I) do każdej n-tej kolumny w trakcie przesyłania znaków ze źródła do miejsca przeznaczenia (po napotkaniu znaku tabulacji wstawianych jest tyle spacji by kolejny znak przypadł w kolumnie będącej wielokrotnością "n").

System Operacyjny CP/M: 2v2

- V -Sprawdz, czy dane zostały poprawnie skopiowane przez odczyt po operacji zapisu (operacja kopiowania musi odbywać się na zbiorze dyskowym)
- W -Ignoruje ochronę zapisu dla zbiorów o statusie R/O bez podawania informacji na konsolę.
- Z -Zerowanie bitu parzystości w każdym wprowadzonym znaku ASCII.

Przykłady użycia PIP z parametrami:

PIP X.MAC=B:[v]

Kopiuje X.MAC z dysku B na dysk aktualnie przyłączony, a następnie sprawdza czy dane zostały prawidłowo skopiowane.

PIP LPT:=X.MAC[nt8u]

Kopiuje X.MAC na LPT:, numerując każdą linię, przemieszczając znaki tabulacji na co 8 kolumnę i tłumacząc małe litery alfabetu na duże.

PIP PUN:=X.HEX[i],Y.ZOT[h]

Najpierw kopiuje X.HEX na PUN: ignorując kończący rekord ":00" w X.HEX; następnie kontynuując przesyłanie danych z Y.ZOT włączając każdy rekord ":00" jaki on zawiera.

PIP X.LIB=Y.MAC[sSUBRI:^Z qJP L3^Z]

Kopiuje zbiór Y.MAC zb. X.LIB. Start kopiowania po napotkaniu łańcucha "SUBRI:" i zakończenie po znalezieniu łańcucha "JP L3".

PIP PRN:=X.MAC[p50]

Przesłaj X.MAC na LST: z numeracją wierszy, tabulacją co 8 kolumnę i nową stroną co 50 linii (domyślne parametry dla PRN: [nt8p60]; p50 zmienia wiec p60)

System operacyjny CP/M 2v2

Pogram systemowy SUBMIT

SUBMIT -Program umożliwia wykonanie rozkazów CP/M zapisanych w postaci zbioru zamiast pobierania ich z klawiatury. W zbiorze z rozszerzeniem SUB mogą być zapisane różne rozkazy dostępne pod kontrolą systemu CP/M włącznie z inicjowaniem działania innych programów. Użycie programu SUBMIT pozwala na przygotowanie zbiorów zawierających pewien stały zestaw rozkazów często wykonywanych w tej samej kolejności. Działanie to stanowi rodzaj pracy wsadowej. Z zastosowaniem SUBMIT łączy się użycie programu XSUB, którego zadaniem jest przygotowanie buforów dla programów inicjowanych przez SUBMIT.

Komenda o postaci: SUBMIT nazwa parm#1..parm#n

Inicjuje realizację programu przetwarzającego grupę komend prototypowych systemu CP/M. Komendy prototypowe mogą zawierać parametry formalne, które będą zastępowane parametrami aktualnymi w trakcie wywołania. Zbiór komend prototypowych tworzony jest przy użyciu programu edytora, przy czym parametry formalne komend zapisywane są w postaci \$1 \$2 \$3 .. \$n gdzie:

nazwa -nazwa zbioru o rozszerzeniu .SUB na dysku
aktualnie przydzielonym do systemu
parm#1..parm#n-parametry aktualne

Między parametrami formalnymi a aktualnymi musi istnieć funkcjonalna zgodność. Gdy ilość parametrów w komendzie prototypowej oraz przy wywołaniu nie odpowiadają sobie, to działanie komendy SUBMIT jest przerywane. Komenda tworzy na dysku aktualnie przydzielonym do systemu zbiór o nazwie S.SUB zawierający komendy CP/M z podstawionymi przy wywołaniu parametrami aktualnymi.

Automatyczne przetwarzanie ciągu komend może być przerwane klawiszem <rubout>. Ostatnia komenda w zbiorze .SUB może inicjować inny zbiór .SUB umożliwiając tworzenie łańcucha komend przetwarzanych w trybie partiiowym.

Załóżmy że zbiór ASMBL.SUB zawiera prototypowe komendy:

```
DIR $1.*  
ERA *.BAK  
PIP $2:=$1.PRN  
ERA $1.PRN
```

i że została wprowadzona komenda

```
SUBMIT ASMBL X PRN
```

Wtedy program SUBMIT utworzy zbiór \$\$\$SUB, podstawiając w zbiorze ASMBL.SUB "X" zamiast każdego znaku \$1 oraz "PRN" zamiast każdego \$2, zawierający komendy:

```
DIR X.*  
ERA *.BAK  
PIP PRN:=X.PRN  
ERA X.PRN
```

wykonywane kolejno przez system.

Opogramowanie narzędzi

W ramach pakietu narzędzi opogramowano następujące programy:

- 1. Programy do obsługi dysków twardych.
- 2. Programy do obsługi dysków 5 1/4 cala.
- 3. Programy do obsługi dysków 8 cala.
- 4. Programy do obsługi dysków 10 cala.
- 5. Programy do obsługi dysków 12 cala.

W ramach pakietu narzędzi opogramowano następujące programy:

- 1. Programy do obsługi dysków twardych.
- 2. Programy do obsługi dysków 5 1/4 cala.
- 3. Programy do obsługi dysków 8 cala.
- 4. Programy do obsługi dysków 10 cala.
- 5. Programy do obsługi dysków 12 cala.

PAKIET PROGRAMOW ASEMBLEROWYCH

- MAKROASSEMBLER M80
- PROGRAM LACZACY L80
- PROGRAM WYDRUKU ODWOLAN CREF80
- PROGRAM LIB80

W ramach pakietu narzędzi opogramowano następujące programy:

- 1. Programy do obsługi dysków twardych.
- 2. Programy do obsługi dysków 5 1/4 cala.
- 3. Programy do obsługi dysków 8 cala.
- 4. Programy do obsługi dysków 10 cala.
- 5. Programy do obsługi dysków 12 cala.

W ramach pakietu narzędzi opogramowano następujące programy:

- 1. Programy do obsługi dysków twardych.
- 2. Programy do obsługi dysków 5 1/4 cala.
- 3. Programy do obsługi dysków 8 cala.
- 4. Programy do obsługi dysków 10 cala.
- 5. Programy do obsługi dysków 12 cala.

W ramach pakietu narzędzi opogramowano następujące programy:

- 1. Programy do obsługi dysków twardych.
- 2. Programy do obsługi dysków 5 1/4 cala.
- 3. Programy do obsługi dysków 8 cala.
- 4. Programy do obsługi dysków 10 cala.
- 5. Programy do obsługi dysków 12 cala.

W ramach pakietu narzędzi opogramowano następujące programy:

- 1. Programy do obsługi dysków twardych.
- 2. Programy do obsługi dysków 5 1/4 cala.
- 3. Programy do obsługi dysków 8 cala.
- 4. Programy do obsługi dysków 10 cala.
- 5. Programy do obsługi dysków 12 cala.

Oprogramowanie narzędziowe

Pakiet obsługi programów assemblerowych wymaga minimum 29 KB pamięci oraz co najmniej jednego napędu dyskowego.

W opisie komend występujących w niniejszym opracowaniu stosuje się następujące symbole:

- [] -wewnątrz nawiasów występuje treść opcjonalna
- < > -dane wprowadzane przez użytkownika
- { } -konieczność dokonania wyboru między dwiema (lub więcej) treściami
- ... -wskazuje, że treść może być powtarzana wielokrotnie

Pierwszym krokiem w tworzeniu oprogramowania jest napisanie zbioru zawierającego program źródłowy (zbior z rozszerzeniem nazwy .MAC). Do tego celu wykorzystuje się edytor systemu CP/M. Następnie program źródłowy należy poddać kompilacji używając M80. W rezultacie otrzymamy zbior zawierający pośredni kod wynikowy. Zbior ten należy następnie przetworzyć programem łączącym L80. Program ten może połączyć wiele zbiorów zawierających programy w postaci pośredniej dając w końcowej fazie jeden zbior z postacią wykonywalną naszego programu. Jeśli występują błędy logiczne, to ich odnalezienie ułatwia program debuggera ZDEB lub DET. Odszukanie błędów może znacznie przyspieszyć listę odwołań do użytych w programie zmiennych wydrukowana dzięki programowi CREF80. Własne biblioteki modułów skompilowanych pozwala tworzyć program LIB80.

Zbior źródłowy przetwarzany przez M80 składa się z linii napisanych w języku ASSEMBLER 8080 albo Z80. Ostatnia linia musi zawierać dyrektywę END kończącą program.

Linia programu nie może przekroczyć 132 znaków i dzieli się na pola oddzielone od siebie co najmniej jedną spacją:

- pole etykiety (symbol zakończony znakiem ":")
- pole kodu operacji
- pole argumentu
- pole komentarza (poprzedzone znakiem ";")

Wszystkie pola programu są opcjonalne, co pozwala na wprowadzenie linii pustej zwiększającej czytelność programu. Program M80 przetwarza zbior zawierający program źródłowy (NAZWA.MAC) i tworzy zbiory wyjściowe zawierające program w postaci pośredniej (NAZWA.REL) oraz raport przebiegu (NAZWA.PRN). Specjalna opcja M80 pozwala uzyskać zamiast zbioru NAZWA.PRN zbior zawierający raport przebiegu kompilacji wraz z listą odwołań do zmiennych użytych w programie (NAZWA.CRF).

Program L80 wczytuje zbiory zawierające kod relokowalny (pośredni: NAZWA.REL), które zostały wymienione na liście parametrów oraz może przeszukać bibliotekę programów standardowych, a tworzy zbior zawierający program wykonywalny (NAZWA.COM).

Program LIB80 wczytuje zbiory pośrednie (NAZWA.REL) i dopisuje je do istniejącej biblioteki lub tworzy bibliotekę wpisując wymienione przez użytkownika zbiory. W trakcie przebiegu LIB80 tworzona jest tymczasowa biblioteka, której nazwa posiada rozszerzenie .LIB.

Program CREF80 wykorzystuje zbior NAZWA.CRF i tworzy zbior przeznaczony do wydrukowania NAZWA.LST.

Oprogramowanie narzędziowe

W każdym programie, który tworzy raport przebiegu zamiast nazwy zbioru (NAZWA.PRN) można podać symbol konsoli lub drukarki. Raport przebiegu zostanie wówczas wyprowadzony na dane urządzenie (bez zapamiętania go w zbiorze).

Symbol używane w programach assemblerowych

-etykiety

-symbole ogólnodostępne (PUBLIC)

-symbole zewnętrzne (EXTERNAL)

Symbole mogą również posiadać atrybuty charakteryzujące rodzaj segmentu pamięci, w którym mają być umieszczone. Symbole mogą mieć dowolną długość, ale kompilator bierze pod uwagę:

-pierwszych 16 znaków dla etykiet

-pierwszych 6 znaków dla symboli typu PUBLIC i EXTERNAL

Symbole mogą zawierać następujące znaki:

A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z

0,1,2,3,4,5,6,7,8,9

\$....?.@

Symbol nie może zaczynać się od cyfry. Małe litery tłumaczone są na duże przez kompilator.

Etykiety

Etykieta jest nazwą, do której można odwołać się w instrukcji programu o ile operandem tej instrukcji może być adres lub wartość 16-bitowa. Etykieta musi być pierwszym tekstem użytym w linii programu. Bezpośrednio po symbolu etykiety musi wystąpić znak ":".

Atrybut PUBLIC

Symbole ogólnodostępne (PUBLIC) są używane w miejscach gdzie występują operandy 16-bitowe. Można odwoływać się do nich w innych jednostkach programowych niż ta, w której symbol został zdefiniowany. Atrybut PUBLIC nadaje się symbolowi poprzez:

-nazwę i podwójny znak dwukropka BUF:: DS 1000

-użycie dyrektywy PUBLIC, ENTRY, GLOBAL (patrz:dyrektywy M80)

PUBLIC BUF

BUF: DS 1000

Atrybut EXTERNAL

Symbol zewnętrzny (EXTERNAL) jest definiowany w innej jednostce programowej niż ta, w której następuje odwołanie do niego. (W jednostce definiującej musi być typu PUBLIC) Atrybut EXTERNAL nadaje się symbolowi poprzez:

-użycie podwójnego znaku "##" bezpośrednio po nazwie:

STA BUF##

-użycie jednej z dyrektyw: EXT, EXTRN, EXTERNAL:

EXT BUF

-użycie dyrektywy dla symboli jednobajtowych:

BYTE EXT BUF

Oprogramowanie narzędziowe

Atrybuty obszarów programów i/lub danych.

- ABSOLUTE DATA MODE** - fragment posiadający ten atrybut danych musi być ładowany w określone miejsce pamięci
- DATA RELATIVE MODE** - fragment posiadający ten atrybut może ulegać zmianom, musi być ładowany do pamięci typu RAM i może podlegać relokacji na etapie łączenia
- CODE RELATIVE MODE** - fragment posiadający ten atrybut nie może ulegać zmianom, może być ładowany do pamięci typu ROM i może podlegać relokacji
- COMMON** - określa, że obszar posiadający ten atrybut może być wykorzystywany przez wiele jednostek programowych. Kompilator domyślnie przyjmuje atrybut **CODE RELATIVE MODE**.

Aby obszarowi programu lub danych nadać inny atrybut należy użyć jednej z dyrektyw:

- ASEG** - atrybut **ABSOLUTE MODE**
CSEG - atrybut **CODE RELATIVE MODE**
DSEG - atrybut **DATA RELATIVE MODE**
COMMON - atrybut **COMMON**

Operandy i wyrażenia.

Argumenty kodów operacji i dyrektyw mogą być proste lub złożone (połączone operatorami). Argumenty proste nazywa się operandami, a złożone - wyrażeniami.

Operandy

1. Liczby.

Kompilator przyjmuje domyślnie dziesiętny system liczenia. Podstawa systemu liczenia może być zmieniona przez zastosowanie dyrektywy **RADIX** lub przez określenie systemu w jakim dana liczba została zapisana, tzn.:

- nnnnB** - liczba binarna
nnnnD - liczba dziesiętna
nnnnO - liczba osemkowa
nnnnH - liczba szesnastkowa
X"nnnn" - liczba szesnastkowa

2. Znaki.

Ciągi znaków budowane są z dozwolonych w języku **ASSEMBLER** znaków ograniczonych znakami pojedynczego lub podwójnego apostrofu lub ". Jeżeli tekst ma zawierać wewnątrz znaki apostrofu należy użyć ich dwukrotnie. Np.: **DB 'TOBIAS'**. Znaki są zapamiętywane jako szesnastkowe wartości kodu ASCII.

3. Symbole.

Symbole mogą być użyte w wyrażeniach jako operandy. Dla symboli o atrybucie **EXTERNAL** można stosować następujące operatory: **+, -, *, /, MOD, HIGH, LOW**.
Rezultat wyrażenia, w którym użyto symbolu typu **EXTERNAL** będzie również typu **EXTERNAL**.

Oprogramowanie narzędziowe

- Dla symboli pozostałych typów obowiązują poniższe reguły:
- przy stosowaniu operatorów różnych od AND, OR, XOR operandy mogą być dowolnego typu
 - dla operatorów AND, OR, XOR, SHL, SHR oba operandy muszą być typu ASEG
 - jeżeli wyrażenie zawiera operand typu ASEG i operand innego niż ASEG typu, to wynik nie będzie typu ASEG
 - przy odejmowaniu operandów posiadających różne typy wynik może być typu ASEG (oczywiście może być takiego typu jakiego są wszystkie operandy - jeżeli są tego samego typu)
 - jeżeli dodajemy symbole typu ASEG i CSEG wynik będzie nieznanym dla kompilatora (problem rozwiąże program L80)
 - Jako specjalny operand może zostać użyta bieżąca wartość licznika programu. Aby odwołać się do niego należy jako operandu użyć znaku "\$". Np.: LXI H,\$
 - Jako operandu można użyć rozkazu ASSEMBLERA 8080. Ale tylko pierwszy bajt przełumaczonego rozkazu będzie użyty jako operand. Zamknięcie użytego rozkazu w nawiasach przyspieszy proces kompilacji (oznacza dla kompilatora potrzebę rozwinięcia rozkazu tylko do kodu operacji) Np.: MVI A,(RST 7)

Operatory

Dozwolone są następujące operatory logiczne i arytmetyczne:

Operator Działanie

- NUL** podaje wartość TRUE jeśli argument jest pusty. Instrukcja warunkowa: IF NUL argument jest niespełniona, jeżeli pierwszy znak argumentu jest inny niż średnik lub znak powrotu karetki CR. (Podobnie dla dyrektyw IFB i IFNB)
- TYPE** podaje wartość w postaci bajtu charakteryzującego dwie cechy argumentu:
- typ argumentu
 - czy argument jest typu EXTERNAL
- Argumentem operatora TYPE może być wyrażenie. Jeżeli wyrażenie jest błędne wówczas wartość bajtu jest równa zero. Postać bajtu odpowiada:
- x...tt -gdzie x=1 określa że argument jest EXTERNAL
 - tt -określa typ wyrażenia
- 00 -typ ASEG
 - 01 -typ CSEG
 - 10 -typ DSEG
 - 11 -typ COMMON
- Operator ten używany jest wewnątrz makroinstrukcji:
- ETY** MACRO A
- LOCAL** B
- B** SET TYPE A
- LOW** obcina wartość argumentu 16-bitowego typu ASEG podając mniej znaczący bajt
- HIGH** podaje bardziej znaczący bajt wartości argumentu 16-bitowego typu ASEG
- *** mnożenie
- /** dzielenie
- MOD** modulo (dzieli lewy argument przez prawy dając resztę z tego dzielenia) np.: MVI A,15 MOD 7

Oprogramowanie narzędziowe

SHR przesunięcie w prawo (operator przesunął lewy argument o ilość bitów podanych przez prawy argument np.:
CPI 16 SHR 5

SHL przesunięcie w lewo

- znak minus

+ dodawanie

- odejmowanie

EQ równość (daje TRUE w przypadku równości argumentów)

NE nierówność

LT mniejszy (TRUE jeśli lewy arg. < prawy arg.)

LE mniejszy lub równy (TRUE jeśli l.arg < lub = p.arg)

GT większy (TRUE jeśli l.arg > p.arg)

GE większy lub równy (TRUE jeśli l.arg > lub = p.arg)

NOT TRUE jeśli wartość logiczna l.arg jest negacją wartości logicznej zero)

AND TRUE jeśli wartość logiczna obu argumentów jest TRUE (oba argumenty muszą być typu ASEG)

OR TRUE jeśli wartość logiczna l.arg lub p.arg jest TRUE (oba argumenty muszą być typu ASEG)

XOR TRUE jeśli jeden z argumentów jest TRUE a drugi FALSE

Kolejność operatorów przy obliczaniu wartości wyrażen:

NUL, TYPE
LOW, HIGH
*, /, MOD, SHL, SHR
+,-
EQ, NE, LT, LE, GT, GE
NOT
AND
OR, XOR

Jeśli w wyrażeniach występują nawiasy, to w pierwszej kolejności są obliczane wyrażenia zawarte w nawiasach. Wszystkie operatory poza +, -, *, / muszą być oddzielone od operandów spacją.

DYREKTYWY KOMPILATORA

.8080 - oznacza wybór rozkazów mikroprocesora 8080 (jest przyjmowana domyślnie jeśli użytkownik nie użyje żadnej z powyższych dyrektyw)

Uwaga: W dalszej części przy opisie dyrektyw znakiem "*" zaznaczone są dyrektywy obowiązujące po .Z80.

DEFINE BYTE - definicja bajtu

[<ETYKIETA>] DB <WYRAZENIE>[,<WYRAZENIE>...]
* [<ETYKIETA>] DEFB <WYRAZENIE>[,<WYRAZENIE>...]
[<ETYKIETA>] DB <CIĄG ZNAKOW>[,<CIĄG ZNAKOW>...]
* [<ETYKIETA>] DEFM <CIĄG ZNAKOW>[,<CIĄG ZNAKOW>...]

Wartość wyrażenia nie może przekroczyć 255. Dyrektywa rezerwuje jednobajtowe pola za wpisaniem wartości określonej w polu argumentu. Gdy argumentem jest ciąg znaków - dyrektywa umieszcza kod ASCII każdego znaku w jednym bajcie.

Przykłady:

Definicja	Obecna wartość	Definicja	Obecna wartość
DB (1+3)*SHL 3	0003: A0		
DB 'AB'	0004: 41 42		
DB 'AB' AND 0FFH	0006: 42		

Oprogramowanie narzędziowe

DEFINE CHARACTER - definicja znaku

[<ETYKIETA>] DC <CIAG ZNAKOW>

Dyrektywa ta zapamiętuje ciąg znaków, podobnie jak DB z tym, że w bajcie kodu ostatniego znaku jest ustawiana jedynka na najbardziej znaczącym bicie. Np.:

Definicja: Po kompilacji:
DC `ABC` 0000: 41 42 C3

DEFINE SPACE - definicja obszaru

[<ETYKIETA>] DS <WYRAZENIE>[,<STALA>]

* [<ETYKIETA>] DEFS <WYRAZENIE>[,<STALA>]

Dyrektywa rezerwuje określoną przez <WYRAZENIE> ilość bajtów pamięci i może zainicjować ten obszar wartością podaną w argumentach <STALA>.

Np.: DS 100H -rezerwuje 256 bajtów pamięci

DS 100H,0 -rezerwuje 256 bajtów pamięci i zapełnia ten obszar zerami

DEFINE WORD - definicja 16-bitowego słowa

[<ETYKIETA>] DW <WYRAZENIE>[,<WYRAZENIE>...]

* [<ETYKIETA>] DEFW <WYRAZENIE>[,<WYRAZENIE>...]

Dyrektywa zapamiętuje wartość argumentu <WYRAZENIE> na dwóch bajtach pamięci (najpierw wpisywany jest mniej znaczący bajt, później bardziej znaczący).

EQUOTE - przypisanie wartości

<NAZWA> EQU <WYRAZENIE>

Dyrektywa przypisuje argumentowi NAZWA wartość WYRAZENIE, np.: <NAZWA> EQU 50H

EXTERNAL - nadanie atrybutu EXTERNAL

EXT <NAZWA>[,<NAZWA>...]

EXTRN <NAZWA>[,<NAZWA>...]

EXTERNAL <NAZWA>[,<NAZWA>...]

BYTE EXT <SYMBOL>

BYTE EXTRN <SYMBOL>

BYTE EXTERNAL <SYMBOL>

Dyrektywa informuje kompilator, że NAZWA jest nazwą zewnętrzną (musi ona być określona w innej jednostce programowej jako UBLIC). Kompilator w miejscach odwołania się do symbolu zewnętrznego ustawia zera. Program łączący uzupełnia te miejsca wstawiając odpowiednią wartość po znalezieniu miejsca definicji tego symbolu jako UBLIC.

SET - ustaw

<NAZWA> SET <WYRAZENIE>

* <NAZWA> DEFL <WYRAZENIE>

Dyrektywa działa podobnie jak EQU z tym, że tu NAZWA może być wcześniej zdefiniowana (przez inną dyrektywę SET lub EQU). Jest to szczególnie przydatne wewnątrz makroinstrukcji. Wykorzystuje się ją do ustawiania przełączników dla instrukcji kompilacji warunkowej.

Programowanie: narzędziowe

DEFINICJA I NADANIE ATRYBUTU PUBLIC

ENTRY <NAZWA>[,<NAZWA>...]

GLOBAL <NAZWA>[,<NAZWA>...]

PUBLIC <NAZWA>[,<NAZWA>...]

Dyrektywa określa, że argument NAZWA będzie dostępny dla innych jednostek programowych. Symbol o nazwie wyspecyfikowanej w tej dyrektywie określa się czasem jako pośredni.

Uwaga: Nazwa zdefiniowana tą dyrektywą nie może mieć atrybutu EXTERNAL, ani nie może być nazwą bloku COMMON.

ABSOLUTE SEGMENT - segment bezwzględny

ASEG

Segment o tym atrybucie będzie zawierał kod nierelokowalny (nieprzesuwalny).

Uwaga: Segment ASEG jest adresowany od 0000. Może to spowodować zniszczenie systemu CP/M w pamięci operacyjnej jeśli nie użyje się dyrektywy ORG.

CODE SEGMENT - segment kodu

CSEG

Dyrektywa ta przypisuje segmentowi atrybut kodu relokowalnego. Przyjmuje się, że będzie on ładowany do pamięci typu ROM. Nie może więc ulegać zmianom podczas realizacji.

Uwaga: Jeśli nie zostanie użyta inna dyrektywa (ASEG, DSEG) atrybut CSEG jest przydzielany standardowo.

DATA SEGMENT - segment danych

DSEG

Dyrektywa ta przypisuje segmentowi atrybut relokowalnych danych. Segment ten będzie ulegał zmianie w trakcie realizacji programu i jest ładowany do pamięci typu RAM.

COMMON BLOCK - obszar wspólny

COMMON <NAZWA>

Dyrektywa definiuje obszar wspólny, który może być wykorzystany przez wiele jednostek programowych (definicja obszaru wspólnego musi występować w każdym segmencie programu, który będzie używał tego obszaru. W segmencie wspólnym umieszczają się instrukcje deklaracji pamięci, np.:

Modul A

COMMON /WSPOLNY/

DANE DB 1,2,3

DS 50H

TEKST DC 'ABCD'

Modul B

COMMON /WSPOLNY/

BUF DS 53H

Uwaga 1: Nie jest wymagane definiowanie w każdej jednostce programowej obszaru o tej samej długości (ale musi być użyta ta sama nazwa bloku COMMON).

Uwaga 2: Podczas łączenia przez LINK, największy z bloków o tej samej nazwie musi być ładowany jako pierwszy.

Oprogramowanie narzędziowa

SET ORIGIN - ustaw adres początkowy segmentu
ORG <WYRAZENIE>

Dyrektywa **ORG** powoduje, że adresy rozkazów lub danych występujących po niej będą liczone względem adresu określonego w argumencie **WYRAZENIE**.

Przykład: **ORG 100H**

LD SP,300: rozkaz ten będzie zapamiętany pod adresem 100H

RELOCATE - przesunięcie

.PHASE <WYRAZENIE>

.DEPHASE

Rozkazy zawarte pomiędzy nawiasami **PHASE** i **DEPHASE** będą przeniesione w trakcie kompilacji pod adres określony przez **WYRAZENIE**.

COMMENT - komentarz

.COMMENT <OGRA NICZNIK><TEKST><OGRA NICZNIK>

Dyrektywa spowoduje, że tekst zawarty pomiędzy ogranicznikami będzie przesłany do zbioru raportu lecz nie będzie podawany kompilacji, np.: **.COMMENT +to jest komentarz+**

END OF PROGRAM - koniec programu

END [<WYRAZENIE>]

Dyrektywa **END** wskazuje kompilatorowi, że został osiągnięty koniec zbioru zawierającego program źródłowy. Argument 16-bitowy **WYRAZENIE** określa punkt wejścia do programu (adres ten będzie przekazany programowi łączącemu)

INCLUDE - dołącz

INCLUDE <NAZWA ZBIORU>

\$INCLUDE <NAZWA ZBIORU>

MACLIB <NAZWA ZBIORU>

Dyrektywa **INCLUDE** wskazuje kompilatorowi, aby dołączył do jednostki aktualnie kompilowanej tekst programu zawarty w zbiorze określonym przez argument **NAZWA ZBIORU**. Po zakończeniu kompilacji dołączonego programu następuje kompilacja aktualnej jednostki programowej.

NAME MODULE - nadanie nazwy modułowi programu

NAME <NAZWA>

Dyrektywa **NAME** wskazuje, że utworzony program przyjmuje nazwę określona przez argument **NAZWA**. (Brak tej dyrektywy powoduje utworzenie nazwy modułu z nazwy zbioru zawierającego program źródłowy).

RADIX - zmiana systemu liczbowego

.RADIX <WYRAZENIE>

Dyrektywa **.RADIX** zmienia podstawę domyślnego systemu liczbowego (10) na system określony przez argument **WYRAZENIE** (2, 8, 10 LUB 16)

Uwaga: Zmiana dotyczy tylko stałych używanych w programie (na wydruku kolumny adresu i kodu będą liczbami 16-wymi).

Oprogramowanie narzędziowe

REQUEST - zadanie dołączenia modułu skompilowanego

.REQUEST <NAZWA ZBIORU>[,<NAZWA ZBIORU>...]

Dyrektywa aktywna na etapie wykonania L80. W przypadku nierozwiązania wszystkich symboli o atrybucie **EXTERNAL** program łączący automatycznie przeszuka dyskiety w celu dołączenia modułu o nazwie podanej w dyrektywie **REQUEST** i w rozszerzeniu ***.REL**. (Wynika z tego, że dyrektywa **REQUEST** niezawsze powoduje dołączenie modułu o podanej nazwie) np.: **.REQUEST** MODULX Program L80 w przypadku wystąpienia nierozwiązanych odwołań będzie szukał zbioru **MODULX:REL**.

DYREKTYWY STERUJACE WYDRUKIEM RAPORTU KOMPILACJI

* **EJECT** <WYRAZENIE>

PAGE <WYRAZENIE>

\$EJECT

Dyrektywa powoduje wysunięcie papieru do nowej strony (lub zgaszenie całego ekranu). Używając argumentu **WYRAZENIE** definiujemy ilość wierszy na stronie (od 0 do 255). Opcjonalnie przyjmowanych jest 50 linii.

TITLE <TEKST>

Dyrektywa podaje tekst, który będzie wyprowadzony jako pierwsza linia na każdej stronie.

SUBTTL <TEKST>

\$TITLE <`<TEKST`>>

Dyrektywa powoduje, że **TEKST** będzie wyprowadzony w nagłówku strony, zaraz po tytule (jeżeli został zdefiniowany).

.PRINTX <OGRANICZNIK><TEKST><OGRANICZNIK>

Kompilator po napotkaniu tej dyrektywy w programie wyprowadzi **TEKST** na konsole.

MAKROINSTRUKCJE

DYREKTYWY MAKROINSTRUKCJI

Definiowanie makroinstrukcji

<NAZWA> **MACRO** <PARAMETRY FORMALNE>

ENDM

Argument **NAZWA** jest nazwa makroinstrukcji. Jej wywołanie w programie wygląda następująco: <NAZWA> <PARAMETRY AKTUALNE> **PARAMETRY AKTUALNE** są podstawiane w ciele makroinstrukcji w miejsca, gdzie w definicji makroinstrukcji użyto **PARAMETROW FORMALNYCH**. Ilość parametrów jest dowolna. Muszą one jednak mieścić się w jednej linii programu.

Wyjście z generowania makroinstrukcji.

EXITM

Użycie powyższej dyrektywy powoduje przerwanie rozwijania makroinstrukcji. (Stosuje się ją najczęściej w połączeniu z dyrektywami kompilacji warunkowej).

Oprogramowanie narzędziowe

Powtorzenia.

```
REPT   <WYRAZENIE>
...    CIALO
ENDM
```

Uzycie tej dyrektywy powoduje powtorzenie wygenerowania zawartosci CIALA. Ilosc powtorzen okresla wartosc argumentu WYRAZENIE.

```
IRP    <ARGUMENT>,<PARAMETRY UJETE W ZNAKI < > >
...    CIALO
ENDM
```

Uzycie IRP powoduje, ze ARGUMENT przyjmuje kolejne wartosci z listy PARAMETRY UJETE W ZNAKI < > az do ich wyczerpania. Jednoczesnie za kazdym razem nastepuje wygenerowanie CIALA. Jezeli IRP jest stosowane wewnatrz bloku MACRO, to ograniczniki < > musza byc uzyte podwojnie.

```
IRPC   <ARGUMENT>,<CIAG ZNAKOW>
...    CIALO
ENDM
```

Uzycie IRPC powoduje, ze ARGUMENT przyjmuje wartosci rowne kolejnym znakom z CIAGU ZNAKOW.

Definicja symbolu lokalnego.

```
LOCAL <SYMBOL>[,SYMBOL ...]
```

Uzycie dyrektywy LOCAL dopuszczalne jest tylko wewnatrz makroinstrukcji. Powoduje ona, ze symbole wymienione w polu argumentow beda przez kompilator traktowane jako symbole znane tylko w tej makroinstrukcji. Symbole te zostana zamienione na symbole rozpoczynajace sie od znaku .. (w tej postaci zostana zapamietane i drukowane w tablicy symboli). Symbole typu LOCAL sa uzywane miedzy innymi jako przelaczniki instrukcji warunkowych wewnatrz makroinstrukcji oraz jako etykiety, do ktorych mozna sie odwolac tylko w definicji makroinstrukcji.

Specjalne operatory makroinstrukcji.

- & -operator dolaczenia (konkatenacji) tekstu lub symbolu (parametr formalny nie moze byc uzyty wewnatrz makroinstrukcji jako ciag znakow bez udzialu tego operatora)
- :: -wewnetrzny komentarz makroinstrukcji
- ! -wskazanie, ze znak wystepujacy za tym operatorem ma byc traktowany jakby byl napisany wewnatrz parzystych znakow "< >"
- % -przypisanie wartosci w biezacym systemie liczbowym

Kompilacja warunkowa

Dyrektywy kompilacji warunkowej

Postac ogolna:

```
IFXXX [ARGUMENT] * COND [ARGUMENT]
...    CIALO IF      ...    CIALO COND
[ELSE [ELSE
...    CIALO ELSE   ...    CIALO ELSE
ENDIF             ENDC
```


~~Opis~~ ~~składnia~~ narzędziowe

IF <WYRAZENIE>

IFT <WYRAZENIE>

* COND <WYRAZENIE>

JCOND <WYRAZENIE>

Jezeli WYRAZENIE posiada wartosc logiczna 1 wówczas ciało IF/COND jest kompilowane

IFE <WYRAZENIE>

IFF <WYRAZENIE>

Jezeli WYRAZENIE posiada wartosc logiczna 0 wówczas ciało IF jest kompilowane.

IF1

Ciało IF jest kompilowane jezeli kompilator jest w 1-szym przebiegu.

IF2

Ciało IF jest kompilowane jezeli kompilator jest w 2-gim przebiegu.

IFDEF <SYMBOL>

Jezeli SYMBOL jest zdefiniowany lub jest typu EXTERNAL wówczas ciało IF jest kompilowane.

IFNDEF <SYMBOL>

Jezeli SYMBOL nie jest zdefiniowany, ani nie jest typu EXTERNAL wówczas ciało IF jest kompilowane.

IFB <ARGUMENT ZAWARTY W ZNAKACH < > >

Jezeli ARGUMENT nie jest wyszczegolniony lub jest pusty (nastepuja po sobie znaki < >) to ciało IF jest kompilowane.

IFNB <ARGUMENT ZAWARTY W ZNAKACH < > >

Jezeli ARGUMENT jest wyszczegolniony to ciało IF jest kompilowane.

IFIDN <ARGUMENT1>,<ARGUMENT2>

Jezeli ciąg znaków ARGUMENT1 jest identyczny z ciągiem ARGUMENT2 wówczas ciało IF jest kompilowane.

IFDIF <ARGUMENT1>,<ARGUMENT2>

Jezeli ciągi argumentów są różne, to ciało IF jest kompilowane. W dyrektywach IFIDN i IFDIF argumenty muszą być zawarte w znakach < >.

ELSE

Jezeli warunek zapisany w dyrektywie IF nie jest spełniony, to kompilowane jest ciało ELSE, o ile zostało zapisane w bloku IFXXX...ENDIF.

ENDIF

* ENDC

Dyrektywy te wskazują na zakończenie bloku IFXXX.

Uwaga: Dyrektywy kompilacji warunkowej mogą być zagniezdzone wielokrotnie bez ograniczeń.

Oprogramowanie narzędziowe

- 9808 **DYREKTYWY STERUJĄCE WYDRUKIEM MAKROINSTRUKCJI**
- .XLIST -zawieszenie wyprowadzania wydruku aż do odwołania
 - .LIST -odwołanie .XLIST
 - .SFCOND -zawieszenie wyprowadzania części wydruku, gdy kompilacja miała być warunkowa i warunek nie jest spełniony
 - .LFCOND -anulowanie .SFCOND
 - .TFCOND -jeżeli opcja /X kompilatora M80 została użyta, to dopuszcza się listowanie części programu kompilowanej warunkowo, gdy warunek nie jest spełniony; przy braku /X działa analogicznie jak .SFCOND
 - .XALL -zezwala na drukowanie linii rozwinięcia makroinstrukcji (bez linii nierozwiniętych). Jest przyjmowana domyślnie przez kompilator.
 - .LALL -wydruk tekstu całej makroinstrukcji
 - .SALL -zakaz drukowania rozwinięć makroinstrukcji

- DYREKTYWY STERUJĄCE WYDRUKIEM ODWOŁAŃ WEWNĘTRZNYCH**
- .XCREF -zawieszenie generowania informacji dla CREF80
 - .CREF -wznowienie generowania informacji dla CREF80
- Obie te dyrektywy są aktywne tylko w przypadku użycia opcji /C dla kompilatora M80.

KOMPILATOR M80

Program źródłowy utworzony przy pomocy edytora można poddać kompilacji przez wprowadzenie poniższych komend systemu CP/M: M80

M80 WYNIK, RAPORT=ZRODLO/OPCJE

Przy użyciu pierwszej z tych komend kompilator zgłasza gotowość do działania przez wyświetlenie znaku "***". Należy wówczas wprowadzić linię definiującą:

WYNIK, RAPORT=ZRODLO/OPCJE

Znaczenie poszczególnych elementów:

- WYNIK -nazwa zbioru, w którym znajdzie się program skompilowany (WYNIK.REL). Element ten może być pominięty.
- RAPORT -nazwa zbioru, w którym umieszczony zostanie raport kompilacji (RAPORT.PRN). Jeśli w tym miejscu umieścimy kod urządzenia zewnętrznego (LST:, TTY:), to raport nie zostanie zapamiętany. Element ten może być pominięty.
- ZRODLO -nazwa zbioru zawierającego program w postaci źródłowej
- OPCJE:
 - /O -raport przebiegu adresowany osiemkowo
 - /H -raport przebiegu adresowany szesnastkowo (opcja przyjmowana domyślnie)
 - /R -zbior wyników ma taką samą nazwę jak zbior źródłowy (z rozszerzeniem .REL)
 - /L -zbior raportu ma taką samą nazwę jak zbior źródłowy (z rozszerzeniem .PRN)
 - /C -generowany będzie zbior odwołań wewnętrznych (z rozszerzeniem .CRF); Program CREF80 może przekształcić ten zbior do postaci wydruku (.LST)
 - /Z -kompilowany będzie zbior rozkazów mikroprocesora Z80 (działanie takie jak dyrektywy .Z80)

Programowanie narzędziowe

- /I -kompilowany na bieżąco zbiór rozkazów w mikroproc. 8080 (działanie takie jak dyrektywy .8080)
- /P -przydziel kompilatorowi dodatkowo 256 bajtów pamięci na stos (na stosie zapamiętywane są wszystkie symbole oraz definicje makroinstrukcji); Należy użyć tej opcji przy powtarnej kompilacji, gdy wystąpi błąd wskazujący na przepełnienie tablicy symboli.
- /M -zeruj obszary zadeklarowane dyrektywą DS
- /X -dyrektywa sterująca wydrukiem makroinstrukcji .TCOND dopuszcza umieszczenie w raporcie przebiegu kompilacji części programu kompilowanej warunkowo przy niespełnionym warunku

Przykłady:

- 1) A>M80
*NAZWA,NAZWA=NAZWA
- 2) A>M80 NAZWA,NAZWA=NAZWA
- 3) A>M80 =NAZWA/R/L

Powyższe trzy przykłady wywołają M80 i równoważną kompilacji poddany zostanie zbiór NAZWA.MAC; raport zostanie zapamiętany w zbiorze NAZWA.PRN, a zbiór skompilowany umieszczony zostanie w zbiorze NAZWA.REL.

- 4) A>M80
*=NAZWA
- 5) A>M80 =NAZWA

Kompilacji poddany zostanie zbiór NAZWA.MAC, a zbiór skompilowany otrzyma nazwę NAZWA.REL. Raport nie zostanie wyprowadzony.

- 6) A>M80
*,=NAZWA
- 7) A>M80 ,=NAZWA

Kompilacja zbioru NAZWA.MAC bez tworzenia zbioru wyjściowego. Na konsoli zostanie wyprowadzona jedynie informacja o ilości błędów.

OPIS WYDRUKU KOMPILATORA M80

Każda strona raportu zawiera dwie następujące linie:

```
[TITLE TEKST]      M80 X.XX      PAGE Y
[SUBTTL TEKST]
```

gdzie:

- TITLE TEKST -tekst określony przez dyrektywę TITLE
- X.XXX -numer wersji kompilatora
- Y -numer strony
- SUBTTL TEKST -podtytuł określony przez dyrektywę SUBTTL

Każda linia raportu ma następującą postać:

```
[BLAD]      NNNNM      XX      XXXXM[W]      TEKST
```

gdzie:

- BLAD -jednoliterowy kod błędu (jeśli wystąpi)
- NNNM -adres przydzielony rozkazowi lub danej
- M -jeden ze znaków:
 - ' -adres zawarty jest w segmencie CSEG
 - " -adres zawarty jest w segmencie DSEG
 - ! -adres zawarty jest w bloku COMMON
 - * -symbol typu EXTERNAL
- (spacja)okresła adres bezwzględny (ASEG))
- XX -kod rozkazu po kompilacji
- XXXX -część adresowa przetłumaczonego rozkazu lub dane

Oprogramowanie narzędziowe

- W -znak okraszający pochodzenie linii programu:
C -linia została dołączana poprzez INCLUDE
+ -linia powstała w wyniku rozwinięcia makroinstrukcji
SPACE -linia pochodzi z programu źródłowego

Wydruk tablicy symboli

W tablicy symboli drukowane są w porządku alfabetycznym nazwy makroinstrukcji (MACROS:) i symbole (SYMBOLS:) użyte w programie. Po każdym symbolu drukowany jest adres i jeden ze znaków:

- I -symbol typu PUBLIC
* -symbol typu EXTERNAL
~ -symbol w segmencie CSEG
" -symbol w segmencie DSEG
! -symbol wewnątrz bloku COMMON
SPACE -adres bezwzględny (typu ASEG)
C -nazwa bloku COMMON
U -symbol niezdefiniowany

Komunikaty i kody błędów

- A -błąd argumentu (argument dyrektywy jest niewłaściwy lub przekracza zakres)
C -błąd zagłębienia dyrektyw kompilacji warunkowej (być może brak ENDIF/ENDC)
D -podwójnie zdefiniowany symbol (sygnalizowany w linii programu, w której następuje odwołanie do tego symbolu)
E -nieodwołane użycie symbolu EXTERNAL
M -powtórna definicja symbolu (sygnalizowany w linii, w której następuje próba powtórnej definicji symbolu)
N -błąd konwersji
O -błędny kod operacji (kod nie jest rozkazem Z80/8080 ani dyrektywą, ani makroinstrukcją)
P -błąd przebiegu (wartość etykiety lub nazwy w dyrektywie EQU jest różna w przebiegach 1 i 2 kompilatora)
Q -nie zakończona linia programu
R -nieodwołane użycie realokacji
U -niezdefiniowany symbol (sygnalizowany w miejscu odwołania do tego symbolu)
V -dyrektywa w pierwszym przebiegu posiada wartość nieokreśloną (w tym przebiegu)
%No END Statement
-brak linii END (może wystąpić przy braku zamknięcia ostatniego bloku kompilowanego warunkowo)
Unterminated conditional
-nie zamknięty blok IF-ENDIF (kompilator zakończył czytanie programu źródłowego po napotkaniu dyrektywy END)
Symbol table is full
-przepełnienie tablicy symboli (może wystąpić w wyniku umieszczenia zbyt dużej liczby komentarzy w makrodefinicji, są one zapamiętywane w tablicy symboli). Należy wówczas zmniejszyć liczbę komentarzy i ponownie przebieg. Jeśli błąd wystąpi ponownie należy użyć opcji /P.
[xx] [No! Fatal errors[,xx warnings]
-sygnalizator liczby błędów (pojawia się w raporcie przebiegu oraz na konsoli po zakończeniu kompilacji)

Oprogramowania narzedziowa

PROGRAM LACZACY L80

L80 łączy programy wynikowe o nazwie z rozszerzeniem .REL poddaje je preadresowaniu (realokacji). W wyniku działania programu łączaco-redagującego powstaje moduł wykonywalny o nazwie z rozszerzeniem .COM. Program L80 może wykorzystywać programy standardowe skompilowane oddzielnie, które zapamiętane są w bibliotece programów wynikowych tworzonych przez LIB80. Przy czym mogą być łączone również zbiory wynikowe utworzone przez inne kompilatory niż M80.

Wywołanie L80:

A>L80

*<ZRODLO>[,ZRODLO,...] [/OPCJA]

lub

A>L80 <ZRODLO>[,ZRODLO,...] [/OPCJA]

gdzie:

ZRODLO -nazwy programów wynikowych z rozszerzeniem .REL

OPCJA -jedna z opcji L80

Miejsce zapamiętania modułu wykonywalnego określona jest przy pomocy opcji /N. Jeżeli w trakcie łączenia ma być przeszukana biblioteka programów standardowych, należy użyć opcji /S.

Opcje L80:

- /R -restart programu L80 (możliwe jest anulowanie wcześniejszych dyspozycji)
- /E -zakonczenie pracy L80 i powrot do systemu CP/M
- /G:NAZWA -start programu utworzonego przez L80 (postac lub /G:NAZWA definiuje punkt wejscia do programu;
- /G NAZWA musi byc nazwa programu, podprogramem lub symbolem typu EXTERNAL)
- /N:NAZWA -modul wykonywalny zostanie zapamietany w zbiorze o nazwie okreslonej przed ta opcja
- /N:P -zachowanie w zbiorze tylko programu (bez obszaru danych)
- /P:<ADRES> -opcje ustalajace przesuniecie dla nastepnego programu wynikowego (P-obszar programu, D-obszar danych). Przy braku tych opcji generowany jest pierwszy adres 103H dla obszaru danych i 203H dla programu. (Oczywiscie przy dyrektywie ASEG obowiazuje ladowanie od adresow okreslonych w dyrektywach ORG).
- /S -przeszukaj biblioteke programow standardowych o nazwie wystepujacej przed /S (poszukiwanie nierozwiazanych odwoLAN)
- /U -wyprowadz niezdefiniowane symbole
- /M -wyprowadz mape aktualnej zawartosci modulu wykonywalnego
- /O -ustaw adresowanie osemkowe
- /H -ustaw adresowanie szesnastkowe
- /X -zachowaj zbior wynikowy .COM jako zbior w formacie znakowym f-my INTEL oraz nadaj mu rozszerzenie .HEX
- /Y -utworz specjalny zbior dla debuggera symbolicznego (zbior z rozszerzeniem .SYM). Obok /Y wymagane jest podanie /N i /E.

Oprogramowanie narzędziowe

Przykłady:

A>L80 A,B,STND/S,PROG/N/E/G

-połącz zbiory A.REL i B.REL, dołącz programy standardowe z biblioteki STND, zapamiętaj program wykonywalny w zbiorze o nazwie PROG.COM, a następnie uruchom go.

A>L80 NAZWA,NAZWA/N/E

-utwórz modul ladowalny NAZWA.COM z modulu NAZWA.REL

A>L80 A,B,C,D/N/E/Y

-połącz moduly A.REL, B.REL i C.REL, a następnie utwórz zbior dla debugera symbolicznego o nazwie D.SYM.

Komunikaty programu L80:

?No Start Address

-użyto opcji /G lecz nie został załadowany program główny (być może brak dyrektywy ENTRY w module głównym)

?Loading Error

-zbior nie jest w formacie wymaganym przez L80

?Out of Memory

-zbyt duży program

?Command Error

-L80 napotkał nieznaną dyspozycję (opcję)

?NAZWA Not Found

-zbior NAZWA nie został znaleziony

*2nd COMMON Larger /XXXXXXXX/

-pierwsza definicja obszaru wspólnego nie jest definicja obszaru największego. (Należy zmienić kolejność ładowania modułów przez L80 lub zmienić definicje bloku COMMON o nazwie XXXXXXXX).

*Mult. Def. Global YYYYYYYY

-wielokrotne zdefiniowanie symbolu (typu GLOBAL) YYYYYYYY

*Overlaying Program Area,Start =XXXX

,Public =<symbol>(XXXX)

,External=<symbol>(XXXX)

lub

*Overlaying Data Area,Start =XXXX

,Public =<symbol>(XXXX)

,External=<symbol>(XXXX)

-ostrzeżenie o mogącym nastąpić skasowaniu wcześniej załadowanych programów lub danych (użyto opcji /P lub /D)

?Start Symbol-NAZWA-Undefined

-symbol NAZWA jest nieznanym (użyto /G:NAZWA)

Origin Above Loader Memory,Move Anyway (Y or N)?

lub

Origin Below Loader Memory,Move Anyway (Y or N)?

-dane lub program nie mogą być załadowane w określone miejsce pamięci (użyto opcji /E lub /G), gdyż adres przekracza dopuszczalny w tej konfiguracji sprzętu adres. Po odpowiedzi Y dane lub program zostaną przesunięte tak, by zmieściły się w dostępnym obszarze pamięci.

?Nothing Loaded

-użyto opcji /E lub /G, lecz nie został załadowany program wynikowy

?Can't Save Object File

-zbior ladowalny nie może być zapisany na dysku (błąd transmisji dyskowej)

Oprogramowanie narzędziowe

BIBLIOTEKARZ LIB80

Program LIB80 umożliwia budowanie biblioteki podprogramów standardowych, które mogą być automatycznie dołączane do innych programów na etapie łączenia przez L80. Moduły, z których tworzymy bibliotekę muszą być skompilowane. Program może uaktualniać stare biblioteki wpisując na miejsce starych podprogramów nowe ich postacie. LIB80 może pracować w dwóch trybach: -budowania lub rozszerzania biblioteki
-listowania zawartości biblioteki

Wywołanie LIB80

Program LIB80 wywołuje się przez podanie komendy: LIB
Program zgłasza gotowość do działania przez wypisanie na konsoli znaku "***".

Budując lub rozszerzając bibliotekę należy posłużyć się komendą:

NAZWABIBL=NAZWAZRODLA/OPCJE

gdzie:

NAZWABIBL -zbiór, w którym będą przechowywane podprogramy (przy braku określenia program przyjmie nazwę FORLIB i doda rozszerzenie .REL)

NAZWAZRODLA -może przyjmować postać:

<NAZWA ZBIORU>[<NAZWA MODULU UJĘTA W ZNAKI <>...]

LIB80 wybiera ze zbioru moduł, którego nazwa została określona (przy braku określenia wybiera wszystkie moduły)

Opcje LIB80

/E -powrót do systemu CP/M. Jeżeli nie tworzono nowej biblioteki ani nie rozszerzano starej należy zamiast /E użyć Ctrl-C

/R -zmiana nazwy biblioteki tymczasowej z .LIB na .REL

/L -wydrukuj nazwy modułów i symboli występujących w tych modułach

/U -wydrukuj symbole niezdefiniowane

/C -powrót do stanu początkowego (gdy zostanie załadowany błędny moduł)

/O -osemkowy system liczenia

/H -szesnastkowy system liczenia

Przykłady:

Tworzenie biblioteki:

A>LIB

*NEWLIB=SIN,COS,TAN,ATAN,ACOS,EXP

*LOG,SQRT

*/E

Wybieranie modułu MOD1 ze zbioru ZBIBL do biblioteki:

A>LIB

*NEWLIB=ZBIBL <MOD1>

*/E

Wybieranie modułów od MOD1 do MODN ze zbioru ZBIBL:

A>LIB

*NEWLIB=ZBIBL <MOD1..MODN>

*/E

Wybieranie modułów aż do MOD3 ze zbioru ZBIBL:

A>LIB

*NEWLIB=ZBIBL <..MOD3>

*/E

Oprogramowanie narzędziowe

Wybieranie ze zbioru ZBIBL/Modułów od poprzedzającego MOD1 do następującego po MOD2:

```
A>LIB
*NEWLIB=ZBIBL <MOD1-1..MOD2+1>
```

Listowanie zawartosci Biblioteki:

```
A>LIB
*NEWLIB.LIB/L
*Ctrl-C
```

Uwaga: Należy zachować ostrożność przy operowaniu programem LIB80 na bibliotekach, których kopii nie posiadamy. Program LIB80 podczas pracy tworzy tymczasową bibliotekę (z rozszerzeniem .LIB) i dopiero na końcu (po podaniu opcji /E) tymczasowa biblioteka jest kopiowana na miejsce starej.

Uwaga: Pracując w trybie listowania biblioteki nie należy kończyć pracy przy pomocy /E, ponieważ w bibliotece tymczasowej nie ma żadnych modułów. Szczególnie istotne jest, aby nie przystępować do pracy na bibliotekach FORLIB i PASLIB bez ich uprzedniego zabezpieczenia.

PROGRAM WYDRUKU ODWOŁAN CREF80

Wydruk odwołań do zmiennych używanych w programie zawiera listę użytych symboli, adresy definicji każdego symbolu oraz adresy wszystkich rozkazów wykorzystujących dany symbol. CREF80 tworzy w oparciu o raport przebiegu z rozszerzeniem .CRF wydruk odwołań i zapamiętuje go w zbiorze o takiej samej nazwie z rozszerzeniem .LST.

Wywołanie CREF80

```
A>CREF80
lub
```

```
A>CREF80 WYDRUK=ZRODLO
```

gdzie WYDRUK może przyjmować:

- nazwa zbioru, który będzie zawierał wydruk raportu wraz z listą odwołań
- TTY: raport zostanie wyprowadzony na konsolę
- LST: lub PRN: raport będzie wyprowadzony na drukarkę
- ZRODLO -określa nazwę zbioru utworzonego przez M80

Przykłady

```
A>CREF80 =TEST
```

```
A>CREF80
```

```
*=TEST
```

Zostanie utworzony zbiór TEST.LST zawierający raport i listę odwołań.

```
A>CREF80 PRN:=TEST
```

Raport wraz z listą odwołań zostanie wydrukowany na drukarce

PROGRAMY TYPU: "DEBUGGER"
 Programy typu "debugger" pozwalają na dynamiczne testowanie i poprawianie programów użytkowych pracujących pod systemem CP/M.

Program ZDEB

Program ten jest inicjowany przez podanie jednego ze Zleceń:
 ZDEB

ZDEB NAZWAZB.COM

ZDEB NAZWAZB.HEX

gdzie:

NAZWAZB - jest nazwa zbioru do testowania (z rozszerzeniem .COM lub .HEX)

Program testowany umieszczony jest w pamięci operacyjnej począwszy od adresu 100H.

Zlecenia programu ZDEB:

- D Dump -wyswietl na monitorze zawartosc pamieci w postaci szesnastkowej i znakow ASCII
- D100 -wyswietl 16 linii po 16 bajtow poczynajac od adresu biezacego
- D100,200 -dump pamieci od 100H do 17FH
- D100,200,200 -dump pamieci od 100H do 200H
- F Fill -wypełnij obszar pamieci bajtem wzorca
- F100,200,FF -wstaw od 100H do 200H bajt 0FFH
- G Go -wykonaj program
- G -start programu od biezacej wartosci licznika rozkazow
- G100 -start programu od adresu 100H
- G100,200 -start programu od 100H z ustawionym punktem zatrzymania na 200H
- G100,200,300 -start programu od 100H z opcjonalnymi punktami zatrzymania 200H i 300H
- G,200,300 -start od biezacej wartosci licznika rozkazow z opcjonalnymi punktami zatrzymania 200H i 300H
- I Init -wstaw nazwe zbioru do bloku sterowania dla nastepnej operacji R
- I PROG.COM -wstaw nazwe PROG.COM
- I PROG.HEX -wstaw nazwe PROG.HEX
- L List -listuj fragment programu w postaci kodu mnemotechnicznego assemblera
- L -listuj dwanaście wierszy programu poczynajac od biezacego adresu listowania (adres listowania ustawiany jest na adres nastepnej po ostatniej wylistowanej komorce pamieci)
- L100 -listuj 12 wierszy programu od 100H
- L100,200 -listuj "zdisassembled" kod od 100H do 200H (wstrzymanie listowania przez 200H)
- M Move -przesun okrelony obszar pamieci M o adresie poczatkowym do obszaru o adresie docelowym
- M100,200,400 -kopiuj obszar 100H-200H na obszar 400H-500H
- R Read -czytaj zbior typu .COM lub .HEX przygotowany zleceniem I (z dysku do obszaru programu ZDEB)

Oprogramowanie zarzedziowe

- S Show & Modify** -pokaz i zmien zawartosc komorek pamieci. Podanie "0" konczy zlecenie. Przejscie do nastepnego adresu bez zmiany zawartosci danej komorki pamieci uzyskuje sie przez <RETURN>. Modyfikacja przez wpisanie cyfry szesnastkowej, <RETURN>.
- S4000** -pokaz i zmien obszar pamieci od adresu 4000H
- T Trace** -sledzenie realizacji programu ze wskazaniem zawartosci rejestrow i wskaźników procesora
- T** -sledzenie pojedynczego kroku programu
- T100** -sledzenie wykonania 100H rozkazow
- U Untrace** -przesledzenie realizacji programu bez pokazywania zawartosci rejestrow procesora po kazdym kroku posrednim
- X eXamine** -pokaz i zmien zawartosc rejestru procesora
- X** -pokaz zawartosc rejestrow
- pokaz i zmien:
- | | | |
|----|----------------------------|----------|
| XA | akumulator A | (0-FF) |
| XB | pare rejestrow BC | (0-FFFF) |
| XD | pare rejestrow DE | (0-FFFF) |
| XH | pare rejestrow HL | (0-FFFF) |
| XS | wskaznik stosu SP | (0-FFFF) |
| XX | rejestr IX | (0-FFFF) |
| XY | rejestr IY | (0-FFFF) |
| XP | licznik rozkazow PC | (0-FFFF) |
| XC | wskaznik przeniesienia C | (0/1) |
| XZ | wskaznik zera Z | (0/1) |
| XM | wskaznik znaku (minus) M | (0/1) |
| XE | wskaznik parzystosci E | (0/1) |
| XI | wskaznik przen. pomocnicz. | (0/1) |
- Powrot do systemu uzyskuje sie standardowo przez Ctrl C.

DEBUGGER DET

DET jest to wersja "debuggera" przygotowana specjalnie dla procesora Z80

Sposoby podawania liczb:

liczby dziesietne maja dodany znak ".", a binarne znak "B".

T -adres ostatniej wolnej komorki TPA

- A Assemble** -bezposrednia assemblacja od adresu < T+1
- A 100** -assemblacja od 100H
- AT-120** -assemblacja od adresu TOP-120
- B Breakpoint** -ustawienie max. 12 breakpointow
- B120**
- B** -wypisanie tablicy breakpointow
- BX** -reset wszystkich breakpointow
- BX120** -reset breakpointa na 120H
- BR10** -nadanie breakpointowi atrybutu R
- B120:6** -ustawienie breakpointa z wartoscia poczatkowa licznika 6
- C Continue** -kontynuacja programu (stosowac z T)
- D Dump** -podanie zawartosci pamieci
- D100** -dump pamieci od 100H do 17FH
- D100 200** -dump pamieci od 100H do 200H

Programowanie narzędziowe

- D100,200 -jw.
- D100S180 -dump pamieci od 100H do 27FH
- D100,T -dump obszaru TPA
- E Examine -odczyt portu wejsciowego
input port
- E FB02
- F Fill -wypelnij obszar pamieci
F od, do, bytel, byte2,..
F od, do, "STRING"
- G Go -start programu od podanego adresu
G120
- H Help -funkcje pomocnicze sluzace do wykonywania
obliczen w roznych systemach liczenia
- H A,B -A,B-wyrazenia arytmetyczne, wyniki operacji:
A+B i A-B
- H 12*7 -wynik podany w roznych systemach
- I Init FCB -ustawienie nazwy zbioru w FCB dla nastepnej
operacji write/read
- I XDIR.COM
- L List -disasembluj podany obszar pamieci
- L -listuj nastepne 16 linii
- L180. -listuj od adresu 180
- L100,200. -listuj od adresu 100H do 200H
- L100,200 -jw.
- LS12 -listuj 12 bajtow z nastepnej strony
- LT-100
- M Move -przesun okreslony obszar pamieci M od adresu
poczkatkowego na obszar o adresie docelowym
- M80,180,200 -kopiuje przedzial 80H~180H na przedzial
200H~400H
- O Out -wyprowadz bajt na podany port
- O F900,8
- Q Find -szuka ciagu bajtow lub lancucha w podanym
obszarze pamieci
- Q100,300,CD,5,0 -szuka w obszarze od 100H do 300H ciagu bajtow
CD 05 00
- Q100,200,"STRING"
- R Read File -czytaj zbior, opcjonalnie moze byc podany
offset (domyslnie na 100)
- S Show & Modify -pokazuje zawartosc komorek pamieci i modyfi-
kuje je (podanie "." konczy komende)
- S100 23. -odpowiada poke &100,23
- S100 "STR" -jest takze mozliwe
- T Trace -sledzenie pojedynczego kroku programu ze wska-
zaniem zawartosci rejestrow i flag
- T10. -sledzenie 10 rozkazow
- TN100 -wykonanie 100H rozkazow bez pokazywania zawar-
tosci rejestrow po kazdym kroku
- TJ
- TU
- TW

Oprogramowanie narzędziowe

- V Verify -porównanie dwóch obszarów pamięci
- Vn1,n2,n3 -porównaj obszar od n1 do n2 z obszarem od n3
- Vm1Sm2,m3 -porównaj obszar od m1 o długości m2 z obszarem od m3
- W Write -zapisz do zbioru
- W100,200 -zapisz do zbioru przygotowanego zleceniem I podany obszar pamięci (od 100H do 200H)
- X -pokaz i zmien zawartosc rejestrów Z80
- X -pokaz wszystkie rejestry
- X PC 120 -ustaw PC=120H
- Y1..Y6 -zmienne pomocnicze Y1 do Y6 można tym zmiennym nadac okreslona wartosc

PROGRAMY "UTILITY"

PROGRAM D

Program D spełnia podobne funkcje jak komendy rezydentne i nierezydentne w systemie CP/M 2.2. Daje on mozliwosc szybkiej i bezposredniej manipulacji na zbiorach z biezacego lub podanego dysku. Pozwala na usuniecie jednego lub kilku zbiorow, skopiowanie jednego lub kilku zbiorow, zmiane nazwy zbioru bez potrzeby kopiowania, wyswietlenie zawartosci katalogu z biezacego lub podanego dysku, wyswietlenie zawartosci wybranego zbioru o formacie ASCII, zmiane uzytkownika, zmiane atrybutow dotyczacych wybranego zbioru, weryfikacje wybranego zbioru.

Program D posiada dodatkowa mozliwosc wyswietlenia na dolnej czesci ekranu krotkiego opisu dyrektyw. Uzyskujemy go naciskajac klawisz <SHIFT> i <?> w formie opisanej ponizej:

8-UP	2-DOWN	4-LEFT	6-RIGHT
N-SIDE +	P-SIDE	5-HOME	Q-KADRO
T-TAG	U-UNTAG	@-ERA TAG	L-DRIVE LU-USER
A-ATRYB	E-BELETE	C-COPY	R-RENAME X-TRACK 3
W-DUMP	V-TYPE	S-CZYT	D-DIR P-XCONS
C-EXIT	Y		

Uruchomienie programu D polega na nasiscieciu po sobie klawisza <D> i <ENTER>. Na ekranie ukaze sie katalog biezacego dysku (RAMDISC-oznaczony jest litera A, dysk znajdujacy sie w gornej kieszeni napedu dyskowego-oznaczony jest litera B).

Przyklad:

```
od gory ekranu
>D .COM 32 4
WS .COM 112 14
WSMSG .OVR 208 26
WSOVLY: .OVR 214 28
FORMAT .COM 96 12
```

u dolu ekranu

```
## Driver: B: Files: 5 Bytes Remaining: 1302k Atr:
```


Oprogramowanie narzędziowe

Kursor ustawiony jest na pierwszej pozycji-podświetlona jest nazwa pierwszego zbioru z katalogu. Możemy wybrać dowolny zbiór korzystając z klawiszy cyfrowych lub klawiszy funkcyjnych umieszczonych po prawej stronie klawiatury-przelanczając klawisz <NUM/LOCK> na NUM. Znaczenie klawiszy:

- <2> -przesunięcie kursora o jedna pozycję w dół
- <8> -przesunięcie kursora o jedna pozycję w górę
- <4> -przesunięcie kursora o jedna kolumnę w lewo
- <6> -przesunięcie kursora o jedna kolumnę w prawo
- <5> -powrót kursora na początek

Jeżeli ustawiliśmy kursor na wybranym zbiorze to możemy go zaznaczyć naciskając klawisz <T>. Z lewej strony nazwy zbioru ukaze się symbol "*" -w ten sposób można zaznaczyć kilka zbiorów, na których możemy wykonać operacje kopiowania lub kasowania. Jeżeli chcemy zlikwidować znak "*" -wyróżniający zbiór -należy naprowadzić kursor na ten zbiór i nacisnąć klawisz <U>. Istnieje również możliwość likwidacji wyróżnika "*" dla wszystkich zaznaczonych zbiorów -należy nacisnąć jednocześnie klawisz <SHIFT> i <@>.

Kopiowanie zbioru

Dyrektywa C-COPY programu D umożliwia skopiowanie jednego lub kilku zbiorów do RAMDISC-oznaczony litera A lub na dysk oznaczony litera B. Zmianę dysku uzyskujemy naciskając klawisz <L>, na dole ekranu ukaze się komunikat: l Drive : podajemy nazwę dysku z którym chcemy aktualnie pracować i naciskamy klawisz <ENTER>. Na ekranie wyświetlony zostanie katalog zadanego dysku opisany już wyżej. Zaznaczamy zbiory, które będziemy kopiować i naciskamy klawisz <C>, u dole ekranu pojawi się komunikat: c Drive : -podajemy nazwę dysku, na którym chcemy zapisać zbiór. Jeżeli nazwa zapisywanego zbioru istnieje już w katalogu tego dysku, to pojawi się komunikat:

```
KOZA .PAS already exists delete (Y) Y
```

Gdy chcemy skasować starsą wersję i na jej miejsce zapisać nową należy nacisnąć jednocześnie klawisz <SHIFT> i <Y> -A Po prawidłowym zapisaniu zbioru na dysk ukazze się komunikat

```
KOZA .PAS already exists-delete (Y) Y copied
```

Opis ekranu po operacji kopiowania dwóch zbiorów: WS.COM, FORMAT.COM z dysku B na dysk A (do RAMDISC-u):

```
Drive C:
D .COM 32 4
*WS .COM 112 14
WSMSG .OVR 208 26
WSOVLY1 .OVR 214 28
*FORMAT .COM 96 12
```

u dole ekranu

```
## Driver: B Files: 5 Bytes Remaining: 26K
Total of tagged files= 26K ( 26K)
c Driver : a
WS .COM already exists - delete (Y) Y copied
FORMAT .COM copied
```

Oprogramowanie narzędziowe

Kasowanie zbiorów

Przy kasowaniu zbiorów-dyrektywa E-DELETE należy wybrać dysk z którego chcemy usunąć zbiory, zaznaczyć je i nacisnąć klawisz <E>. U dołu ekranu ukaze się komunikat:

e Delete all ? (YES) :

Należy napisać słowo "YES" (dużymi literami-przycisnięty klawisz <SHIFT>). Opis ekranu po operacji kasowania dwóch zbiorów :WS.COM, FORMAT.COM z dysku B:

od góry ekranu

```
D .COM 32 4
*WS .COM 112 14
  WMSG .OVR 208 26
  WSOVLY1 .OVR 214 28
*FORMAT .COM 96 12
```

u dołu ekranu

** Driver: B Files: 5 Bytes Remaining: 302k Atr:

Total of tagged files= 26K (26K)

e Delete all ? (YES) : YES

```
WS .DOC deleted
FORMAT .DOC deleted
```

Zmiana nazwy zbioru

Dyrektywa R-RENAME pozwala na zmianę nazwy zbioru bez konieczności kopiowania zbioru. Zmiana dotyczy wyłącznie katalogu. Należy ustawić kursor na nazwę zbioru, który chcemy zmienić i nacisnąć klawisz <R>. Na dole ekranu ukaze się komunikat: r New name :TASWORD.COM

Jezeli podamy nazwę nieprawidłową pojawi się komunikat:

r New name :TAS.TEKST Wrong name

Opis ekranu po operacji zmiany nazwy zbioru WS.COM na WORDSTAR.COM:

od góry ekranu

```
D .COM 32 4
*WS .COM 112 14
  WMSG .OVR 208 26
  WSOVLY1 .OVR 214 28
  FORMAT .COM 96 12
```

u dołu ekranu

** Driver: B Files: 5 Bytes Remaining: 302k Atr:

r New name : WORDSTAR.COM

Wyswietlenie katalogu

Dyrektywa D-DIR pozwala na listing katalogu z bieżącego lub podanego dysku. Należy wybrać dysk którego katalog chcemy uzyskać i nacisnąć klawisz <D>. Na dole ekranu uzyskamy katalog wybranego dysku. Przykładowy wydruk katalogu:

d

```
NAME      TYP REC  KAM-ATTR
D .COM 32 4
WS .COM 112 14
WMSG .OVR 208 26
WSOVLY1 .OVR 214 28
FORMAT .COM 96 12
```

** Driver: B Files: 5 Bytes Remaining: 302k Atr:

Oprogramowanie naszedzi

UWAGI PRZY PROGRAM FINDER

Program FINDER służy do wyszukiwania w zbiorze określonej sekwencji znaków, tzn. adresów lub ciągów bajtów.

Opcje: IDC

A - poszukiwanie adresu równego adresowi zadanemu lub adresów mieszczących się w zadanych granicach

S - poszukiwanie ciągów bajtów równych zadanej sekwencji bajtów lub mieszczących się w zadanych granicach

P:XXXX -offset 4-znakowy; jeśli tej opcji nie ma przyjmowany jest offset 0000

Brak opcji A oznacza, że będzie przyjęta opcja S. Kolejność podania opcji jest dowolna.

Wywołanie programu FINDER: FINDER D:filename.typ/op1/op2
FINDER

Wydruk informacji wynikowej można zatrzymać naciskając dowolny znak na klawiaturze oraz uruchomić na nowo naciskając również dowolny klawisz, zakończyć wydruk można znakiem CTRL-C.

PROGRAM DIFER

Program DIFER służy do porównania dwóch zbiorów.

Opcje:

T - zbiory w postaci tekstowej

O - zbiory w postaci "object"

F - utworzony zostanie na dysku D1: zbiór różnic o nazwie "filename1.DIF"

P:XXXX -offset max. 4-znakowy; jeśli tej opcji nie ma przyjmowany jest offset 0000

Brak opcji O oznacza, że będzie przyjęta opcja T. Kolejność podania opcji jest dowolna.

Wywołanie programu DIFER: DIFER D1:filename1.typ1 D2:filename2.typ2/op1/op2/op3

PROGRAM TIX

Programowy pomiar prędkości obrotowej dysku. Prędkość obliczana jest w obrotach na minutę. Wymagane jest podanie częstotliwości CPU w kHz i numeru wybranego napędu dyskowego

PROGRAM COP

Program kopiowania zbiorów na wybranej jednostce dyskowej. Po załadowaniu programu (COP i <RETURN>) zakładamy na jednostce dyskowej dyskietkę źródłową i podajemy nazwę programu do kopiowania. Następnie należy założyć dyskietkę docelową i nacisnąć <RETURN>.

PROGRAM RDBAD

Program RDBAD testuje dyskietki. Po załadowaniu programu (RDBAD i <RETURN>) zakładamy dyskietkę do testowania i naciskamy <RETURN>. Następnie należy zdefiniować format dyskietki, tzn. podać liczbę sektorów na ścieżkę (10 i <RETURN>) oraz liczbę ścieżek testowanych (maksymalnie 35 i <RETURN>). Jeśli w trakcie testowania zostanie wykryty błąd, program RDBAD określa miejsce jego powstania (ścieżka, sektor), rodzaj błędu oraz czy jest to błąd poprawialny ("soft"), czy też niepoprawialny ("hard"). Po zakończeniu testu wyświetla informacje o sumie błędów.

Oprogramowanie narzędziowe

EDYTORY TEKSTU

Opisane dalej edytory EDI oraz EC służy do tworzenia i modyfikacji zbiorów źródłowych zapisanych w kodzie ASCII.

Edytor EDI.

Edytor EDI wywołuje się następująco: EDI PROG.MAC <RETURN> Rozkaz ten powoduje przeniesienie tekstu o nazwie PROG.MAC z jednostki aktualnie przyłączonej do systemu do bufora pamięci edytora, albo utworzenie nowego zbioru w tym buforze.

Zlecenia edytora EDI:

- IC -ustaw lancuch, kolejne znaki ASCII sa wpisywane do bufora edytora
- q -przesun kursor do wielokrotnosci 8 kolumny (tabulacja)
- +L -wswietl na monitorze +n wierszy
- +A -przesun kursor o +n linii
- +J -przesun kursor o +n znakow
- nK -kasuj n linii
- nD -kasuj n znakow
- QA -ustaw kursor na poczatku zbioru
- nS -przechowaj n linii
- U -wstaw przechowane linie (w miwjsce wskazane kursorem)
- <BS> -<BACKSPACE>, kasuj ostatni znak
- EX -zakonczenie pracy edytora w buforze pamieci z zapisaniem zbioru na jednostce dyskowej (aktualnie przydzielonej do systemu)
- EQ -jak EX, ale bez zapisywania zbioru
- <ESC> -oddziela zlecenia edytora w lancuchu zleceń
- <ESC><ESC> -dwukrotne <ESC> konczy kazde zlecenie edytora (lub lancucha zleceń)

Uwagi:

^I -oznacza CTRL-K tzn. jednoczesne naciśnięcie klawisza <CTRL> i I. Położenie kursora w buforze edytora sygnalizowane jest znakiem ". Przy zakończeniu pracy edytora przez EX stary jest automatycznie zmieniany na zbior typu .BAK.

Edytor EC

EC stanowi inną wersję edytora tekstowego. Wywołuje się go przez podanie nazwy EC i naciśnięcie <RETURN>. Następnie wpisuje się W, co pozwala na wpisanie nazwy zbioru do edycji. Wybranie E inicjuje edycję.

Zlecenia edytora EC:

ROZKAZY PRZESUWANIA KURSORA:

- CTRL-S -o znak w lewo
- CTRL-D -o znak w prawo
- CTRL-A -o slowo w lewo
- CTRL-F -o slowo w prawo
- CTRL-E -o linie w gore
- CTRL-X -o linie w dol
- CTRL-R -o strone w gore
- CTRL-C -o strone w dol

Oprogramowanie narzędziowe

CTRL-Q, CTRL-R -do początku zbioru
CTRL-Q, CTRL-C -do konca zbioru
CTRL-Q, CTRL-S -do początku linii
CTRL-Q, CTRL-D -do konca linii

ROZKAZY WSTAWIANIA I USUWANIA:

CTRL-Y -usun linie
CTRL-Q, CTRL-Y -usun do konca linii
CTRL-G -usun znak pod kursorem
DEL -usun znak z lewej

ROZKAZY DLA BLOKOW:

CTRL-K, CTRL-B -oznakuj poczatek bloku
CTRL-K, CTRL-K -oznakuj koniec bloku
CTRL-K, CTRL-T -oznakuj jedno slowo
CTRL-K, CTRL-C -kopiuj blok
CTRL-K, CTRL-V -przesun blok
CTRL-K, CTRL-Y -usun blok

POZOSTALE ROZKAZY EDYCJI:

CTRL-I, HT -tabulacja
CTRL-Q, CTRL-F -odszukaj lancuch znakow
CTRL-Q, CTRL-A -odszukaj lancuch i zastap go innym
CTRL-L -powtorz operacje szukania
CTRL-U -przerwij operacje

Przy operacjach szukania znakow mozliwe sa opcje:

B -szukaj do tylu
G -szukaj w calym zbiorze
U -ignoruj dolne/gorne znaki
W -szukaj tylko calych slow
N -zastap bez zapytania
R -dowolna liczba powtorzen

Przyklady wpisywania opcji: N10

GWU

Programowanie narzędziowe

Indeks komend WordStara

Przesuwanie kursora:

^S w lewo o znak
^D w prawo o znak
^A w lewo o słowo
^F w prawo o słowo
^E w górę o wiersz
^X w dół o wiersz

^Q^S do początku wiersza

^Q^D do końca wiersza

^Q^E do początku obszaru tekstu na ekranie

^Q^X do końca obszaru tekstu na ekranie

^Q^R do początku zbioru

^Q^C do końca zbioru

^Q^B do początku bloku

^Q^K do końca bloku

^Q0..^Q9 do podanego znacznika

^Q^P do pozycji zajmowanej przed wykonaniem ostatniej komendy

Przesuwanie tekstu na ekranie:

^Z w górę o wiersz

^W w dół o wiersz

^R w górę o ekran

^C w dół o ekran

^Q^Z ciągle przesuwanie w górę o wiersz

^Q^W ciągle przesuwanie w dół o wiersz

Wyszukiwanie i zamiana łańcuchów znaków:

^Q^F wyszukiwanie łańcuchów znaków

^Q^A zamiana łańcuchów znaków

^L powtórne wyszukiwanie lub zamiana

^Q^V przywracanie pozycji kursora sprzed ostatniej komendy wyszukiwania lub zamiany

Parametry:

n szukanie powtarzając n razy
U traktowanie małych i dużych liter identyczne
B szukanie w tekście poprzedzającym pozycję kursora
G szukanie lub zamiana do końca zbioru
N automatyczne zamienianie łańcuchów znaków

Znaki specjalne:

^A oznacza dowolny znak
^S oznacza dowolny znak specjalny
^Ox oznacza dowolny znak różny od podanego
^N oznacza <CR><LF>

Oprogramowanie narzędziowe

Podstawowe komendy wprowadzania tekstu

- ^V** ustawianie znaków ON/OFF
- RETURN** koniec akapitu
- ^N** "twardy" znak <CR>
- ^Px** wpisywanie znaków sterujących
- ^I** tabulacja
- ^O^I** ustawianie pozycji tabulacji
- ^O^N** kasowanie pozycji tabulacji
- ^O^F** ustawianie marginesów i tabulacji według dowolnego wiersza zbioru

Komendy usuwające znaki

- ** usuwanie znaku znajdującego się przed kursorem
- ^G** usuwanie znaku z pozycji kursora
- ^T** usuwanie znaków od kursora do końca słowa
- ^Q** usuwanie znaków od początku wiersza do kursora
- ^Q^Y** usuwanie znaków od kursora do końca wiersza
- ^Y** usuwanie wybranego wiersza
- ^K^Y** usuwanie bloku

Komendy zachowywania zbioru na dysku i zaniechania edycji

- ^K^S** zachowanie zbioru i powrót do jego edycji
- ^K^D** zachowanie zbioru i powrót na górnym poziomie edycji
- ^K^Q** zaniechanie edycji

Komendy redakcji dokumentu na ekranie

- ^O^C** centrowanie tekstu wiersza
- ^O^L** ustawienie lewego marginesu
- ^O^R** ustawienie prawego marginesu
- ^O^X** zezwolenie na pisanie poza ustawionymi marginesami
- ^O^G** zadanie tabulacji akapitu
- ^O^S** ustawienie odstępu międzyliniowego
- ^B** przeredagowanie akapitu

Przelaczniki dotyczące formatowania tekstu

- ^O^W** przenoszenie słów do następnego wiersza
- ^O^T** wyświetlanie "linijki"
- ^O^J** justowanie tekstu
- ^O^V** zmienna tabulacja
- ^O^H** automatyczne przenoszenie wyrazów
- ^O^E** "miękkie" znaki przeniesienia
- ^O^D** wyświetlanie znaków sterujących wydrukiem
- ^O^P** wyświetlanie granic stron

Komendy dotyczące znaczników

- ^K0...^K9** ustawianie/gaszenie znacznika o podanym numerze
- ^Q0...^Q9** przesuwanie kursora do podanego znacznika

Oprogramowanie narzędziowe

Komendy dotyczące bloków

^K^B zaznaczanie początku bloku
^K^K zaznaczanie końca bloku
^K^V przemieszczenie bloku
^K^C kopiowanie bloku
^K^Y usuwanie bloku

Komendy dotyczące bloków (cd)

^K^H gaszenie i zapalanie znaczników początku i końca bloku
^Q^B przesuwanie kursora do początku bloku
^Q^K przesuwanie kursora do końca bloku

Komendy dotyczące zbiorów dyskowych

^K^W zapisanie bloku do zbioru
^K^O skopiowanie zbioru do dysku
^K^J usunięcie zbioru z dysku
^K^F włączenie/wyłączenie opcji wyświetlania katalogu dysku
^K^R wstawienie tekstu ze zbioru dyskowego
^K^E zmiana nazwy zbioru
^K^L zmiana wybranego dysku
^K^P drukowanie zbioru

Komendy opisujące działanie WordStara

^J^H ustawienie poziomu HELP
^J^B opis redakcji akapitu
^J^P opis ustawiania znaczników
^J^D opis komend DOT
^J^M opis ustawiania marginesów i pozycji tabulacji
^J^S opis linii statusu
^J^R opis linijki
^J^F opis znaczenia flag
^J^V przenoszenie fragmentów tekstu

Komendy dodatkowe

^Q^Q powtórzenie następnej komendy
^U przerwanie komendy

Funkcje specjalne drukarki

^S podkreślenie
^B wytłuszczenie
^D podwójny druk
^X przekreślenie
^V pół odstepu międzyliniowego w dół
^T pół odstepu międzyliniowego w górę
^H nakładanie znaków
^O pojedyncza spacja
^Y zmiana koloru taśmy barwiącej

Oprogramowanie narzędziowe

Znaki sterujące wydrukiem S A X S O N Y O U Y

- ^C przerwanie drukowania
- ^K ustawienie warunkowej pozycji numeru strony
- ^L przesunięcie papieru do nowej strony
- ^A ustawienie alternatywnej szerokości czcionki
- ^N ustawienie normalnej szerokości czcionki
- ^F pisanie kodu spacji
- ^G pisanie kodu
- ^M nakładanie linii

Komendy DOT

- .LH wysokość czcionki
- .PL długość arkusza papieru
- .MT górny margines
- .MB dolny margines
- .HM margines nagłówka
- .FM margines stopki
- .PC kolumna z numerem strony
- .PO prawy margines drukarki
- .PA koniec strony
- .CP warunkowy koniec strony
- .HE nagłówek
- .FO stopka
- .OP wstrzymanie numerowania stron
- .PN numer strony
- .CW szerokość czcionki
- .SR wielkość przesunięcia pionowego frazy
- .UJ mikrojustowanie
- .BP dwukierunkowe drukowanie
- .. komentarz

Komendy MailMerge

- .DF definicja zbioru danych
- .RV wczytanie rekordu zmiennych
- .RP powtórzenie przetwarzania dokumentu
- .SV nadanie zmiennej wartości
- .AV wczytanie zmiennej z konsoli
- .DM wyświetlenie wiadomości
- .CS wymazanie ekranu
- .FI wczytanie dodatkowego tekstu
- .PF formatowanie tekstu podczas drukowania
- .RM prawy margines
- .LM lewy margines
- .LS odstęp międzyliniowy
- .OJ justowanie tekstu
- .IJ interpretacja tekstu wejściowego

Oprogramowanie narzędziowe

K O D Y R O Z K A Z O W E D Y T O R A
T U R B O P A S C A L

ROZKAZY STEROWANIA KURSOREM:

Znak w lewo :	CTRL-S	K
Znak w prawo :	CTRL-D	L
Slowo w lewo :	CTRL-A	A
Slowo w prawo :	CTRL-F	M
Linia w gore :	CTRL-E	P
Linia w dol :	CTRL-X	S
Scroll w gore :	CTRL-W	M
Scroll w dol :	CTRL-Z	
Strona w gore :	CTRL-R	
Strona w dol :	CTRL-C	
Do 1-szej linii ekranu :	CTRL-Q, CTRL-E	
Do ostatniej linii ekranu :	CTRL-Q, CTRL-X	HL
Do poczatku pliku :	CTRL-Q, CTRL-R	HL
Do konca pliku :	CTRL-Q, CTRL-C	TL
Do poczatku linii :	CTRL-Q, CTRL-S	MS
Do konca linii :	CTRL-Q, CTRL-D	ML
Do poczatku bloku :	CTRL-Q, CTRL-B	PL
Do konca bloku :	CTRL-Q, CTRL-K	PL
Do ostatniej (poprzedniej) pozycji kursora :	CTRL-Q, CTRL-P	PL

ROZKAZY WSTAWIANIA I USUWANIA:

Tryb wstawiania ON/OFF :	CTRL-V	FO
Wstaw linie :	CTRL-N	FO
Usun linie :	CTRL-Y	FW
Usun do konca linii :	CTRL-Q, CTRL-Y	FW
Usun slowo z prawej :	CTRL-T	SR
Usun znak pod kursorem :	CTRL-G	LU
Usun znak z lewej :	DEL	FR

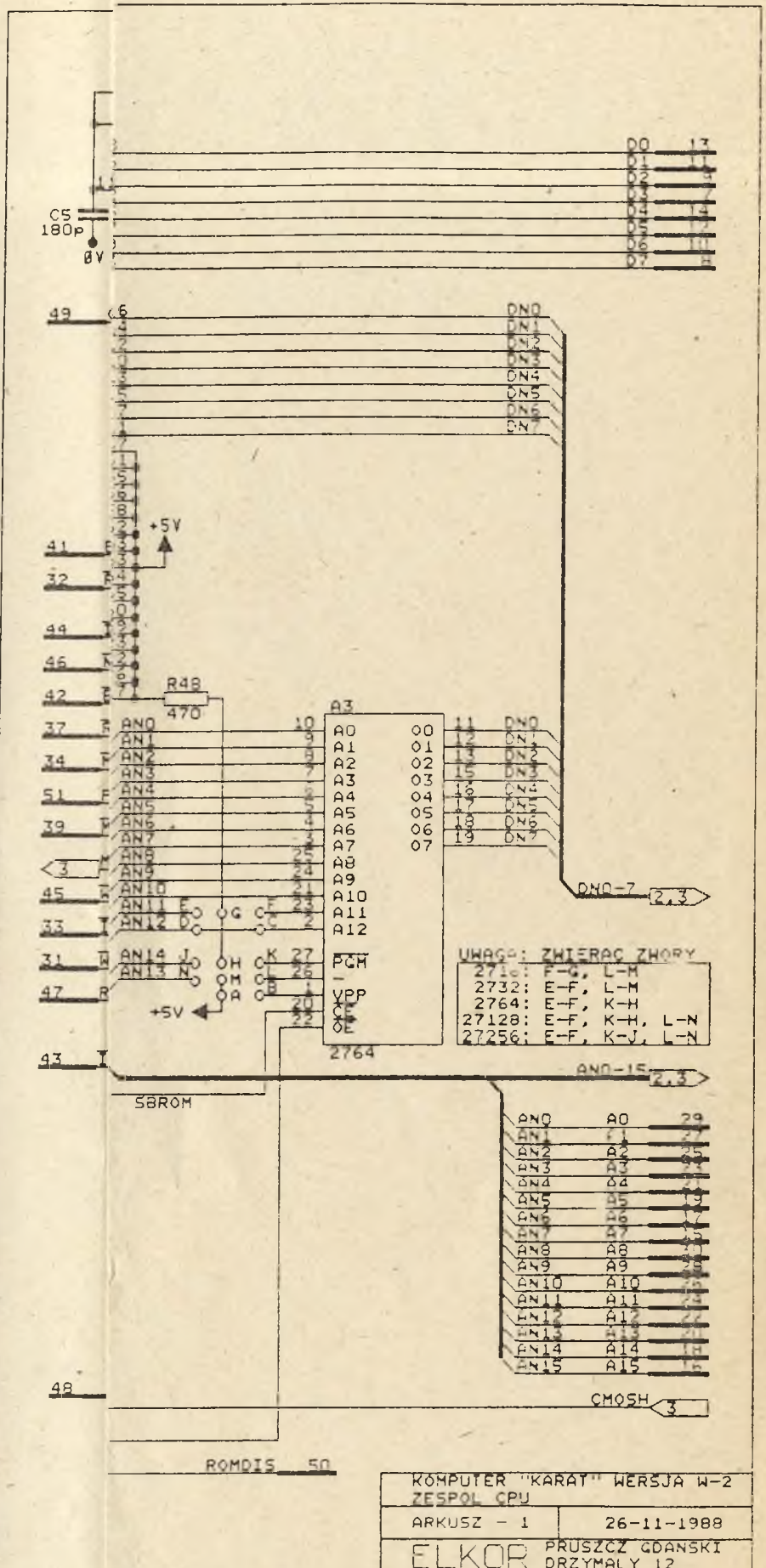
ROZKAZY DLA BLOKOW:

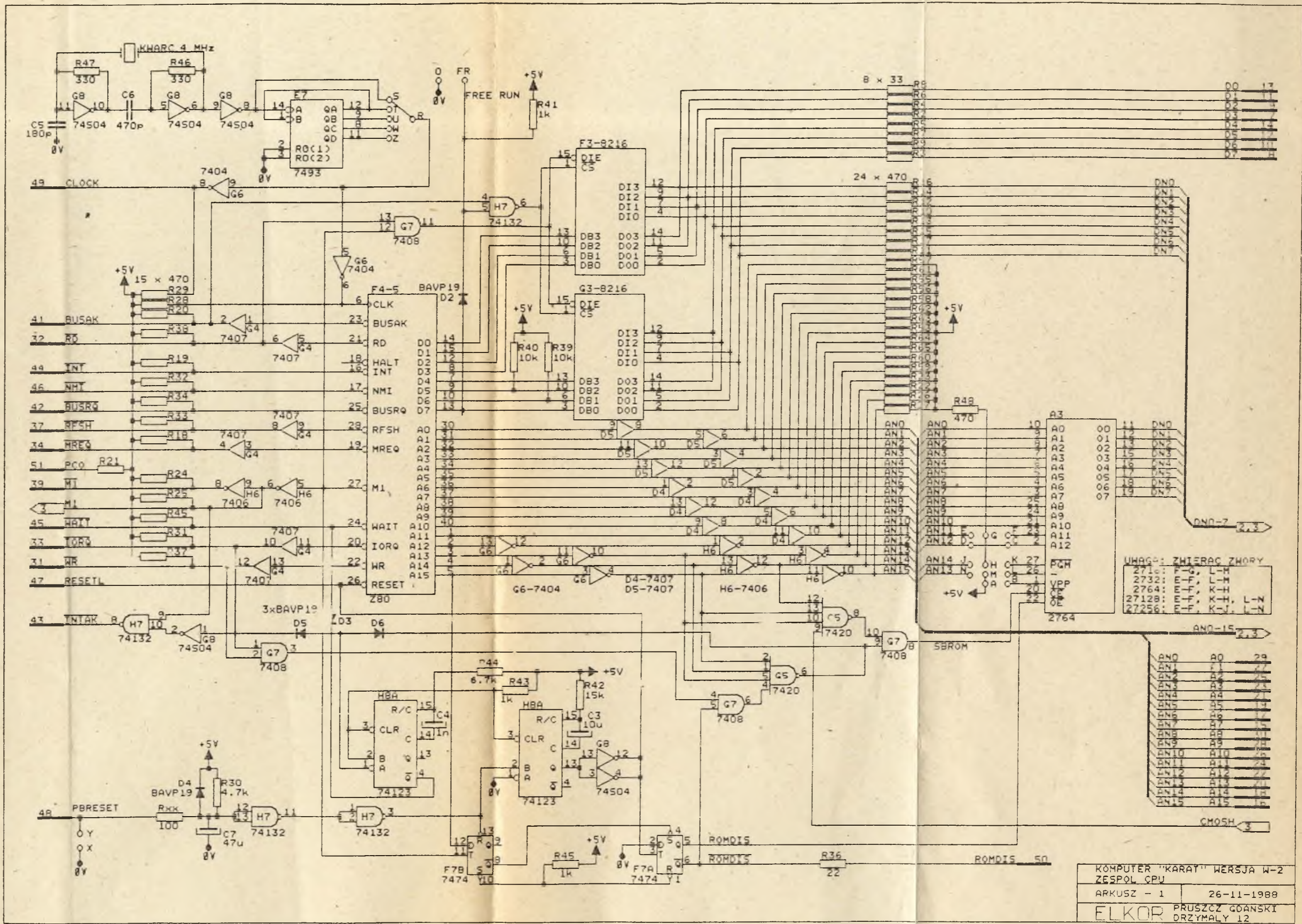
Oznakuj poczatek bloku :	CTRL-K, CTRL-B	
Oznakuj koniec bloku :	CTRL-K, CTRL-K	
Oznakuj jedno slowo :	CTRL-K, CTRL-T	
Kopiuj blok :	CTRL-K, CTRL-C	DF
Przesun blok :	CTRL-K, CTRL-V	DF
Usun blok :	CTRL-K, CTRL-Y	RF
Czytaj blok z dysku :	CTRL-K, CTRL-R	RF
Zapisz blok na dysku :	CTRL-K, CTRL-W	RF
Wykryj/wyswietl blok :	CTRL-K, CTRL-H	RF

POZOSTALE ROZKAZY EDYCJI:

Tab :	CTRL-I, HT	FF
Auto tab on/off :	CTRL-Q, CTRL-I	RM
Restore line :	CTRL-Q, CTRL-L	MI
Find :	CTRL-Q, CTRL-F	SI
Find & replace :	CTRL-Q, CTRL-A	SI
Repeat last find :	CTRL-L	LI
Abort operation :	CTRL-U	
Control character prefix :	CTRL-P do wprowadzania znakow sterujacych do pliku - musi poprzedzac wprowadzony znak. np. CTRL-P, CTRL-G dla dzwonka	

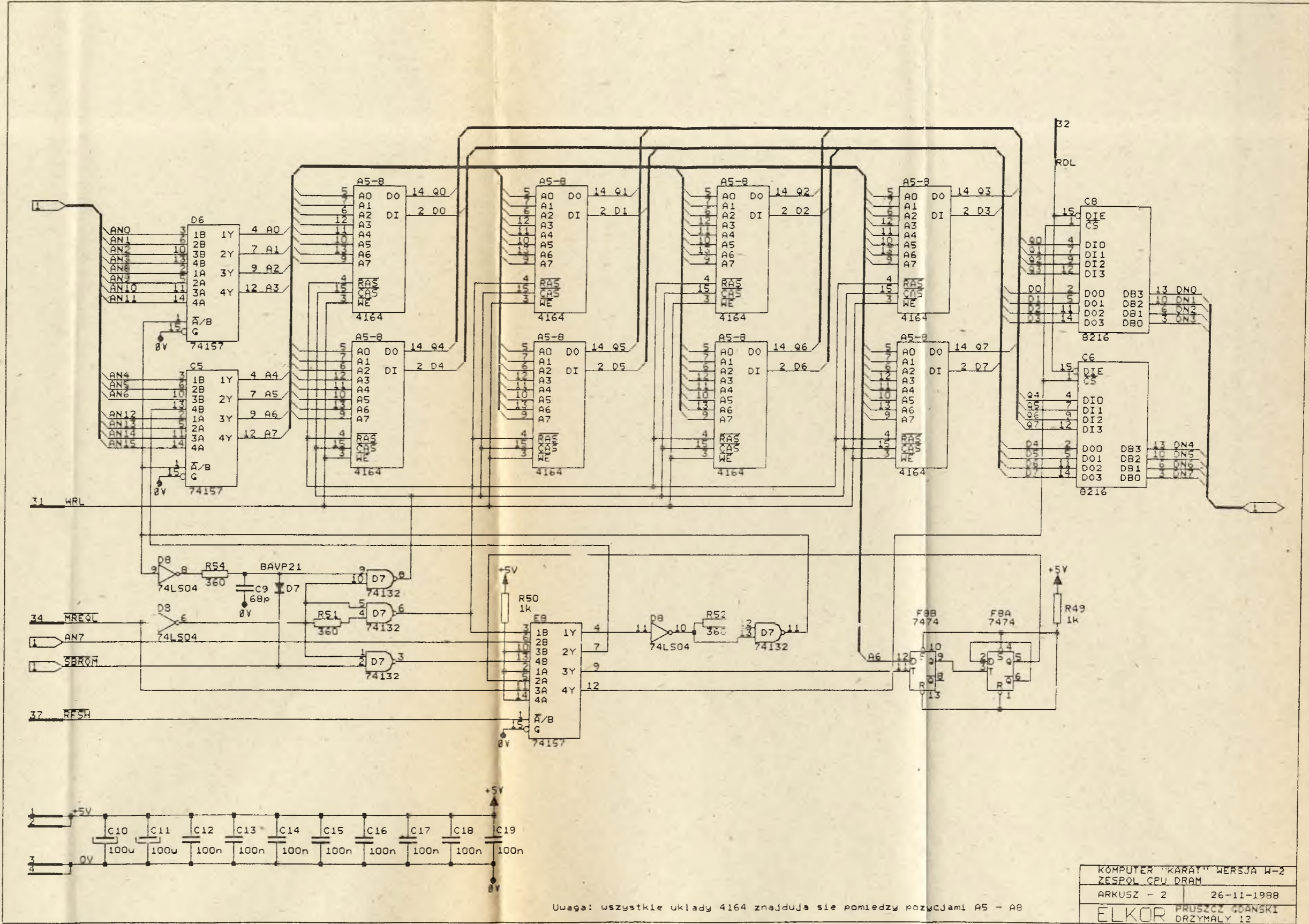
SCHEMATY IDEOWE KARAT-2





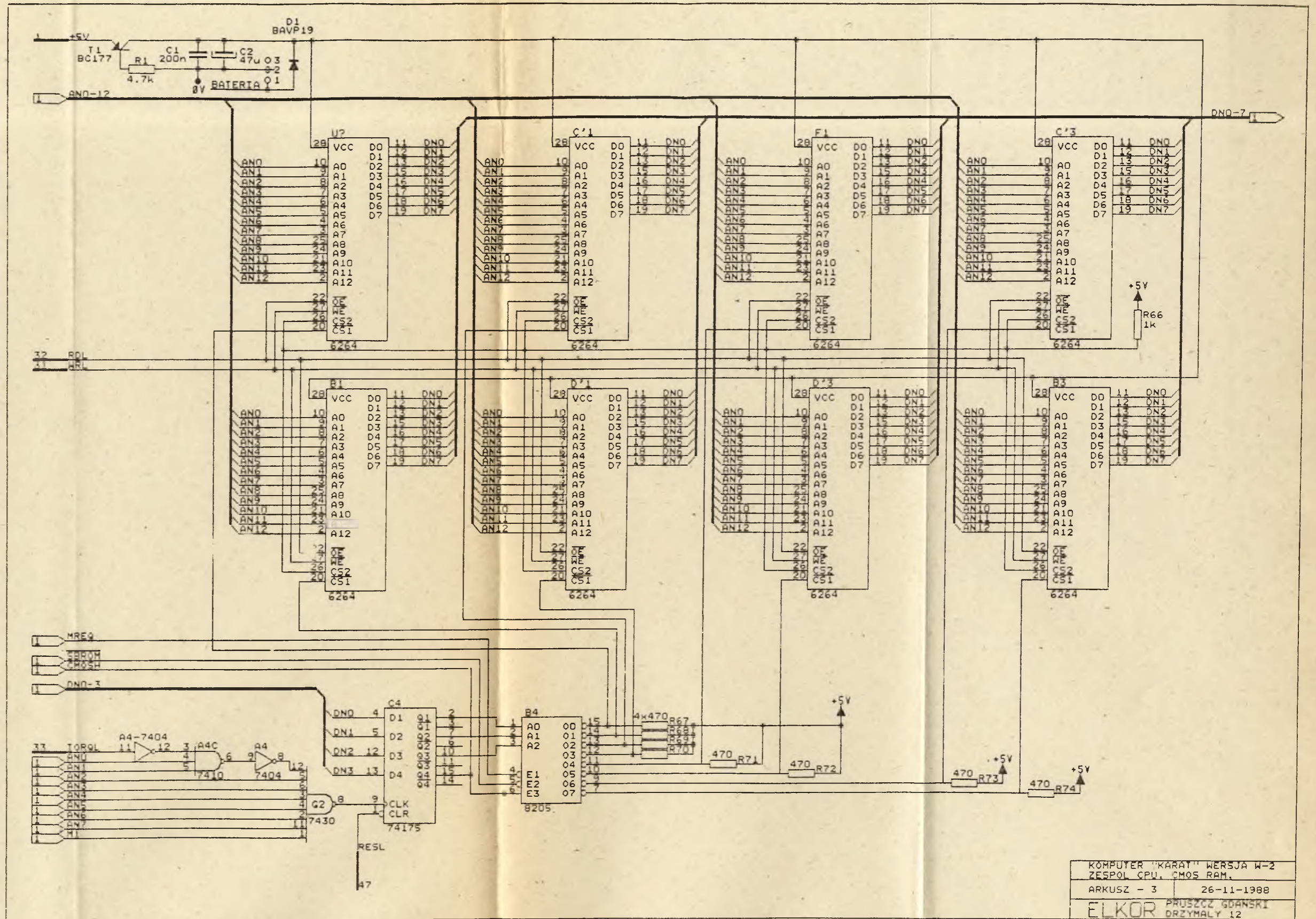
UWAGA: ZMIERAC ZMIORY
 2710: F-G, L-H
 2732: E-F, L-H
 2764: E-F, K-H
 27128: E-F, K-H, L-N
 27256: E-F, K-J, L-N

KOMPUTER "KARAT" WERSJA W-2
 ZESPOL CPU
 ARKUSZ - 1 26-11-1988
 ELKOR PRUSZCZ GDANSKI
 DRZYMALY 12

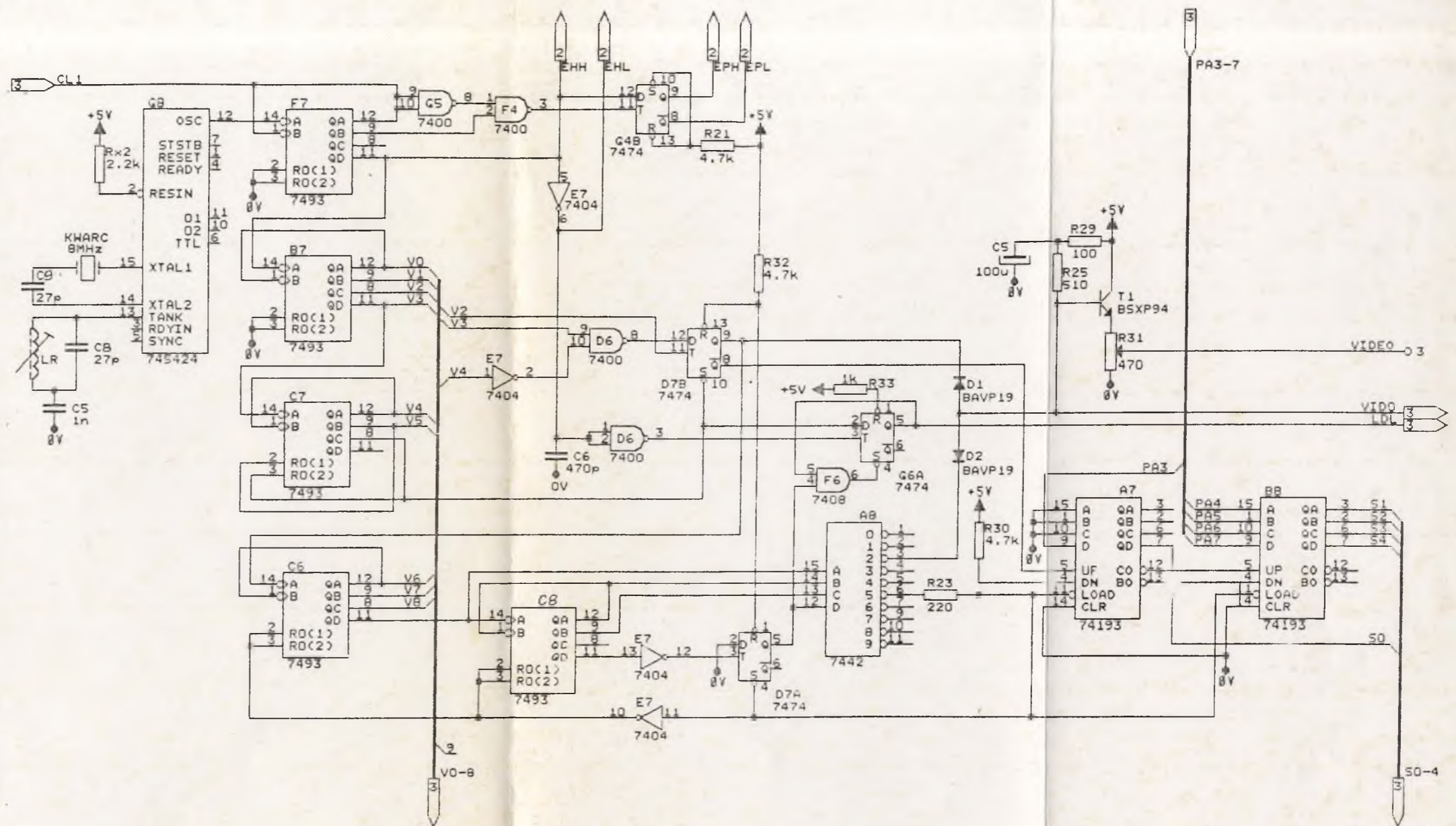


Uwaga: wszystkie układy 4164 znajdują się pomiędzy pozycjami A5 - A8

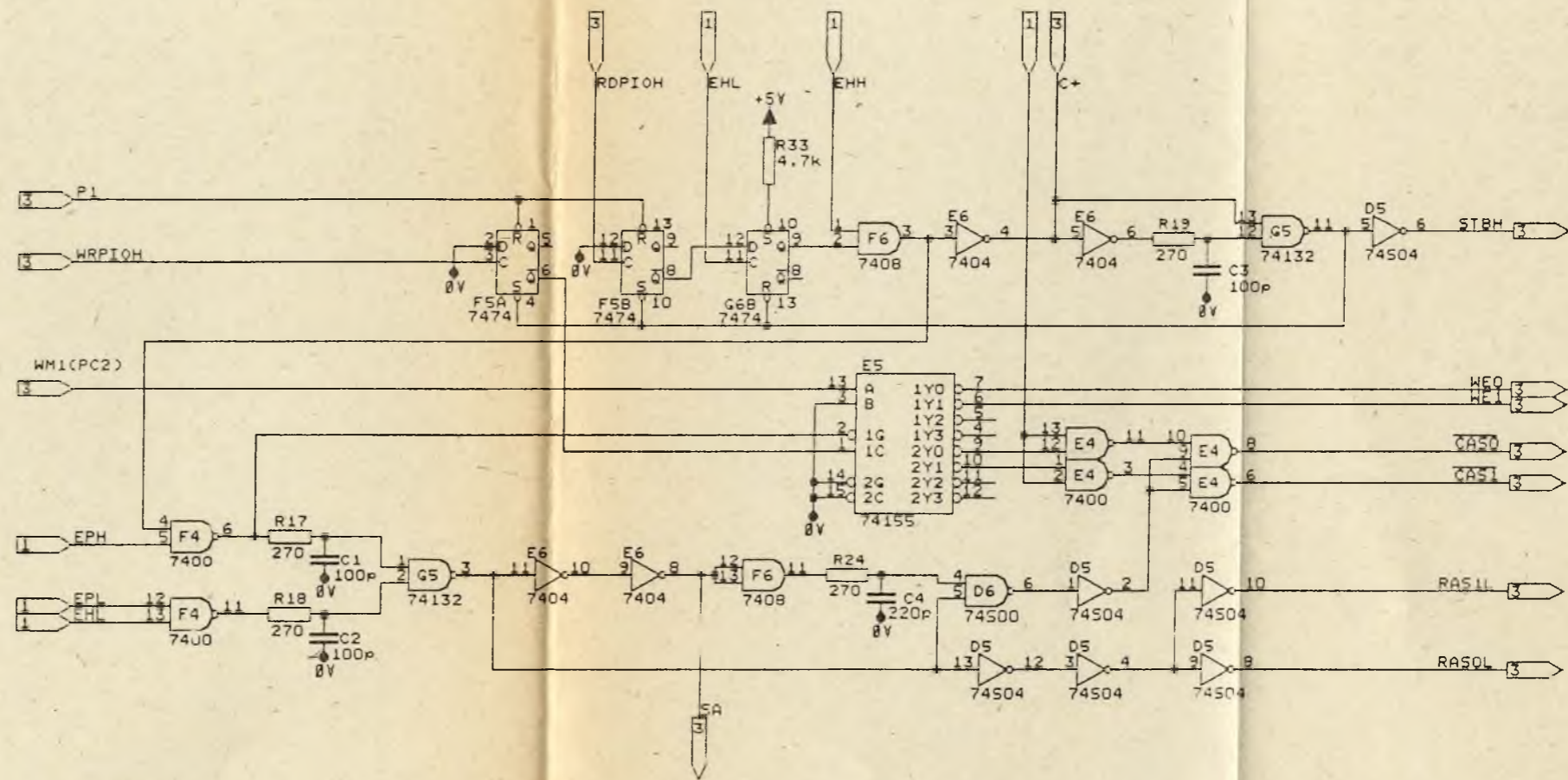
KOMPUTER "KARAT" WERSJA W-2	
ZESPÓŁ CPU DRAM	
ARKUSZ - 2	26-11-1988
ELKOR	PROSZĘ COANSKI ORZYMAŁY 12



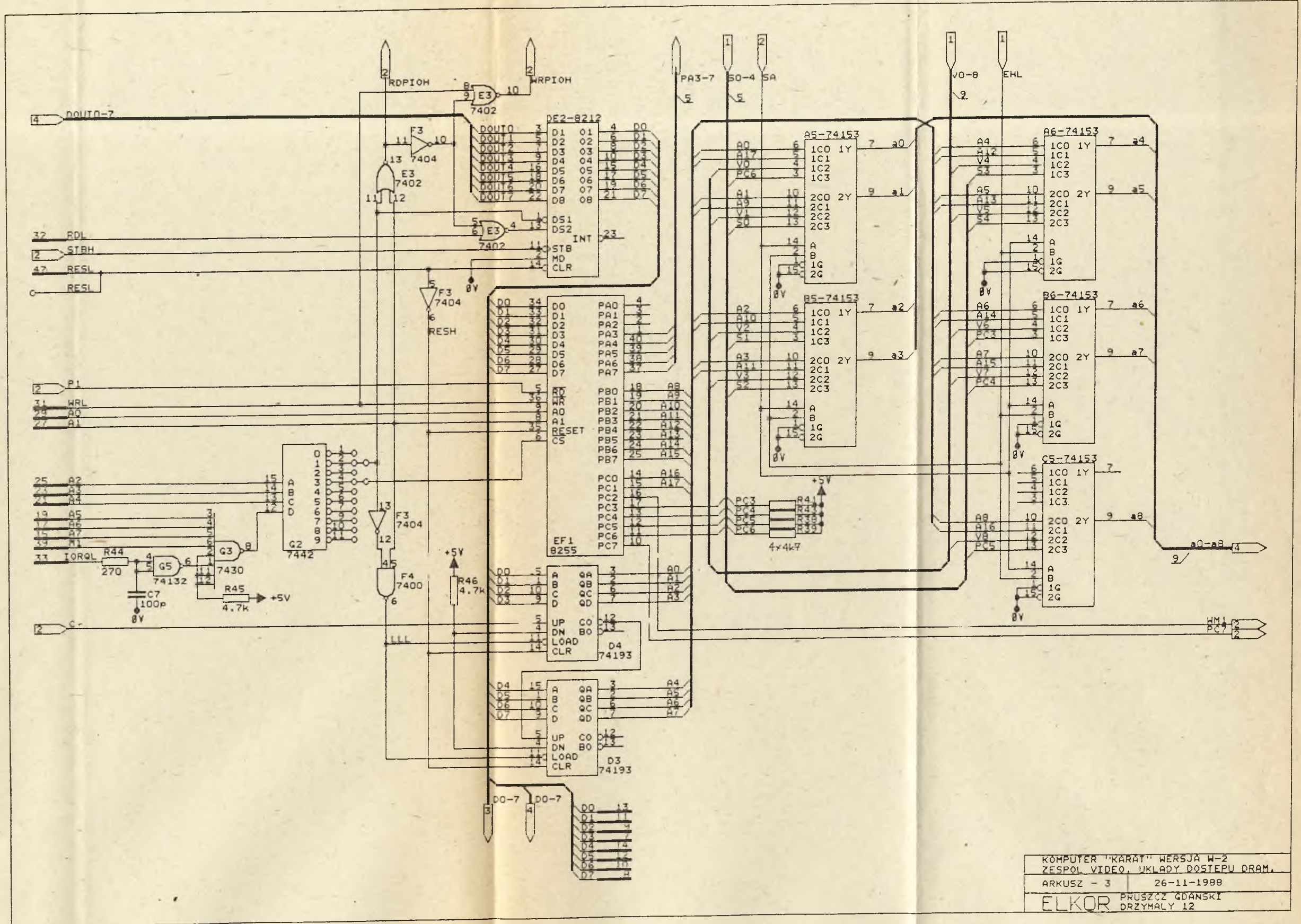
KOMPUTER "KARAT" WERSJA W-2
 ZESPOL CPU, CMOS RAM,
 ARKUSZ - 3 26-11-1988
 ELKOR PRUSZCZ GDANSKI
 DRZYMALY 12



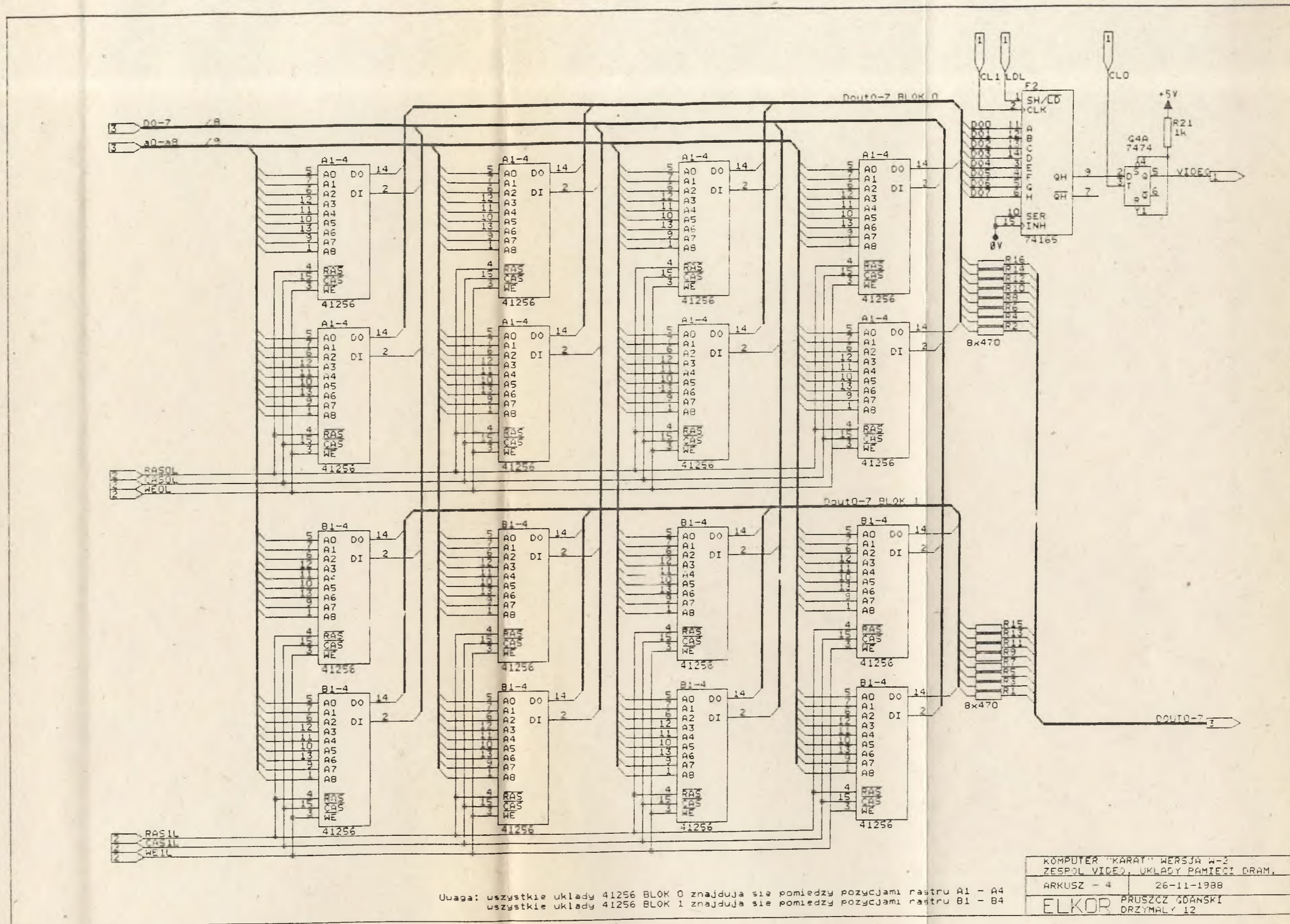
KOMPUTER "KAWAT" WERSJA W-2	
ZESPOL VIDEO, UKLADY SYNCHRONIZACJI.	
ARKUSZ - 1	26-11-1988
ELKOR	PRUSZCZ GDANSKI DRZYMALY 12



KOMPUTER "KARAT" WERSJA W-2
 ZESPÓŁ VIDEO, KONTROLER DRAM.
 ARKUSZ - 2 26-11-1988
 ELKOR PRUSZCZ GDANSKI
 DRZYMAŁY 12

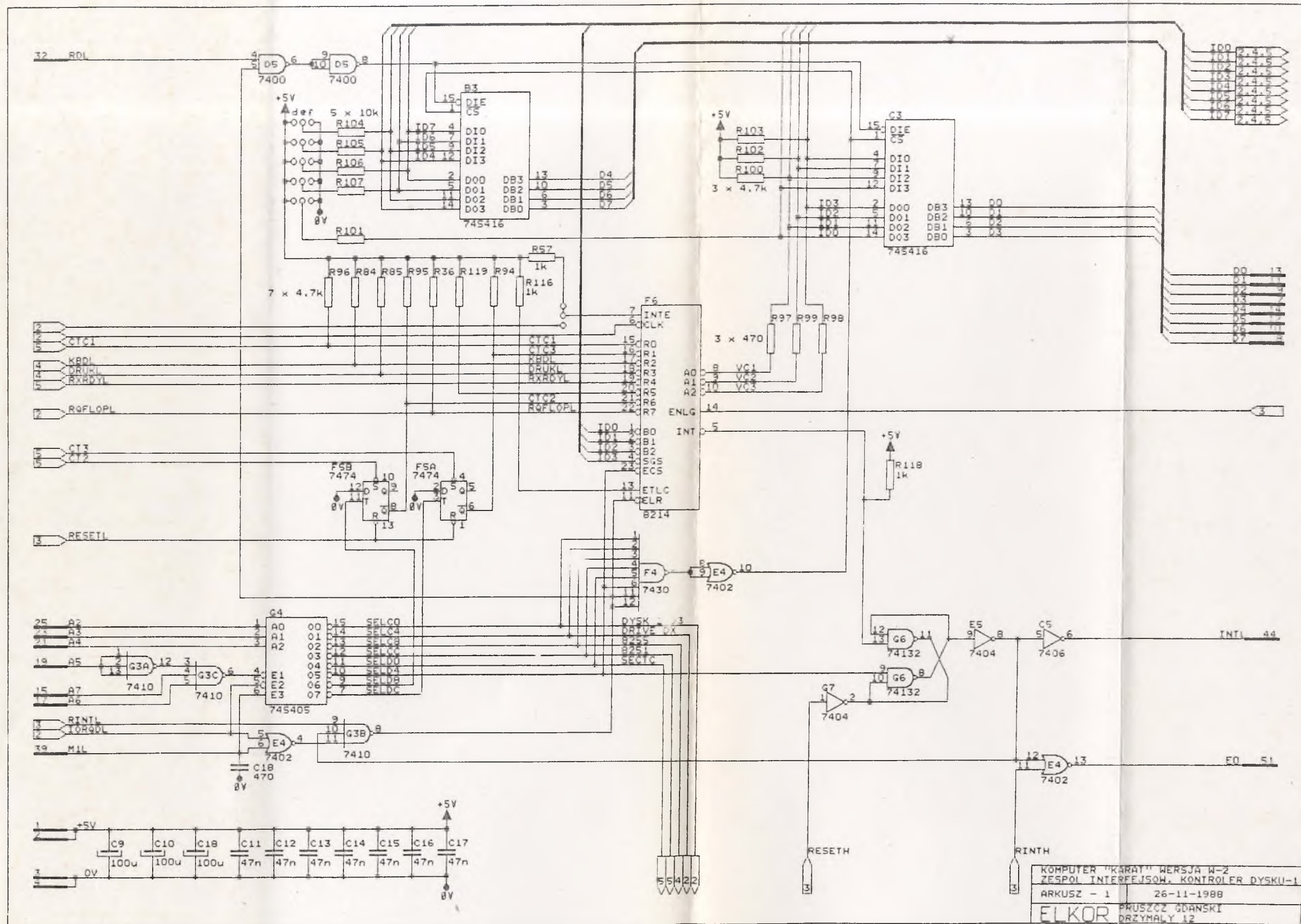


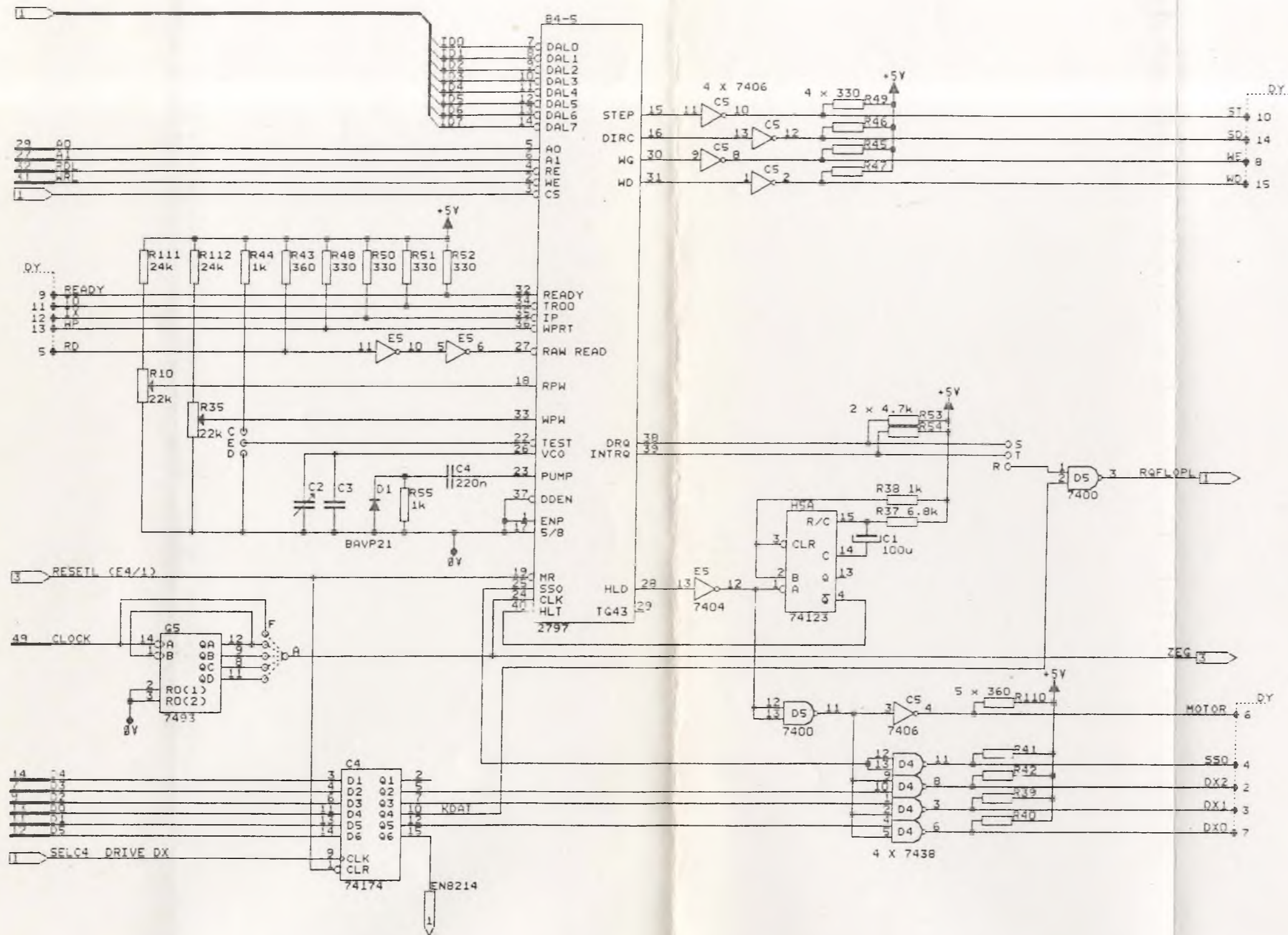
KOMPUTER "KARAT" WERSJA W-2
 ZESPÓŁ VIDEO, UKŁADY DOSTĘPU DRAM.
 ARKUSZ - 3 | 26-11-1988
 ELKOR PRUSZCZ GDAŃSKI
 DRZYMAŁY 12



Uwaga: wszystkie układy 41256 BLOK 0 znajdują się pomiędzy pozycjami rastru A1 - A4
 wszystkie układy 41256 BLOK 1 znajdują się pomiędzy pozycjami rastru B1 - B4

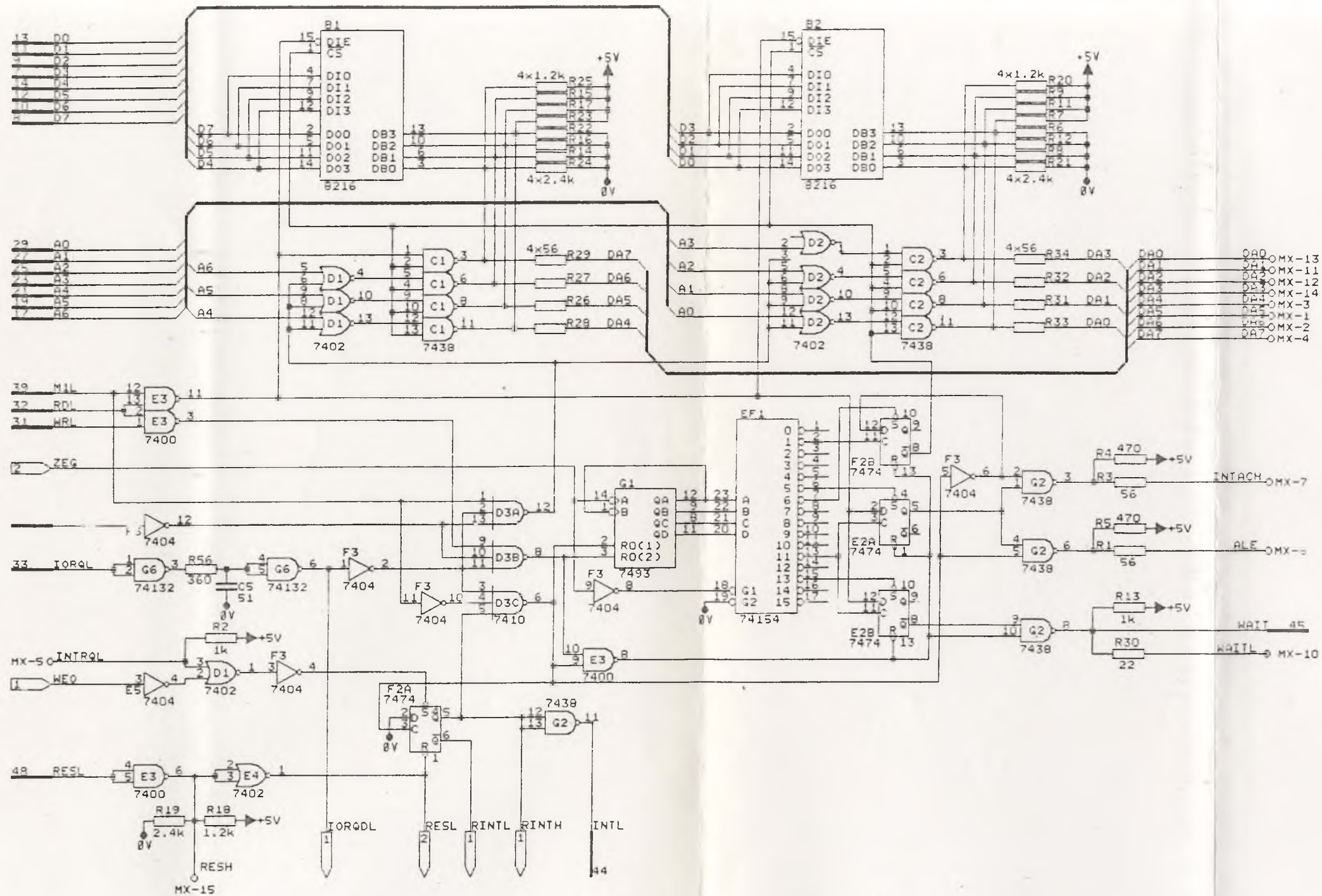
KOMPUTER "KARAT" WERSJA W-2	
ZESPÓŁ VIDEO, UKŁADY PAMIĘCI GRAM.	
ARKUSZ - 4	26-11-1988
ELKOR	PRUSZCZ GÓRSKI DRZYMAŁY 12



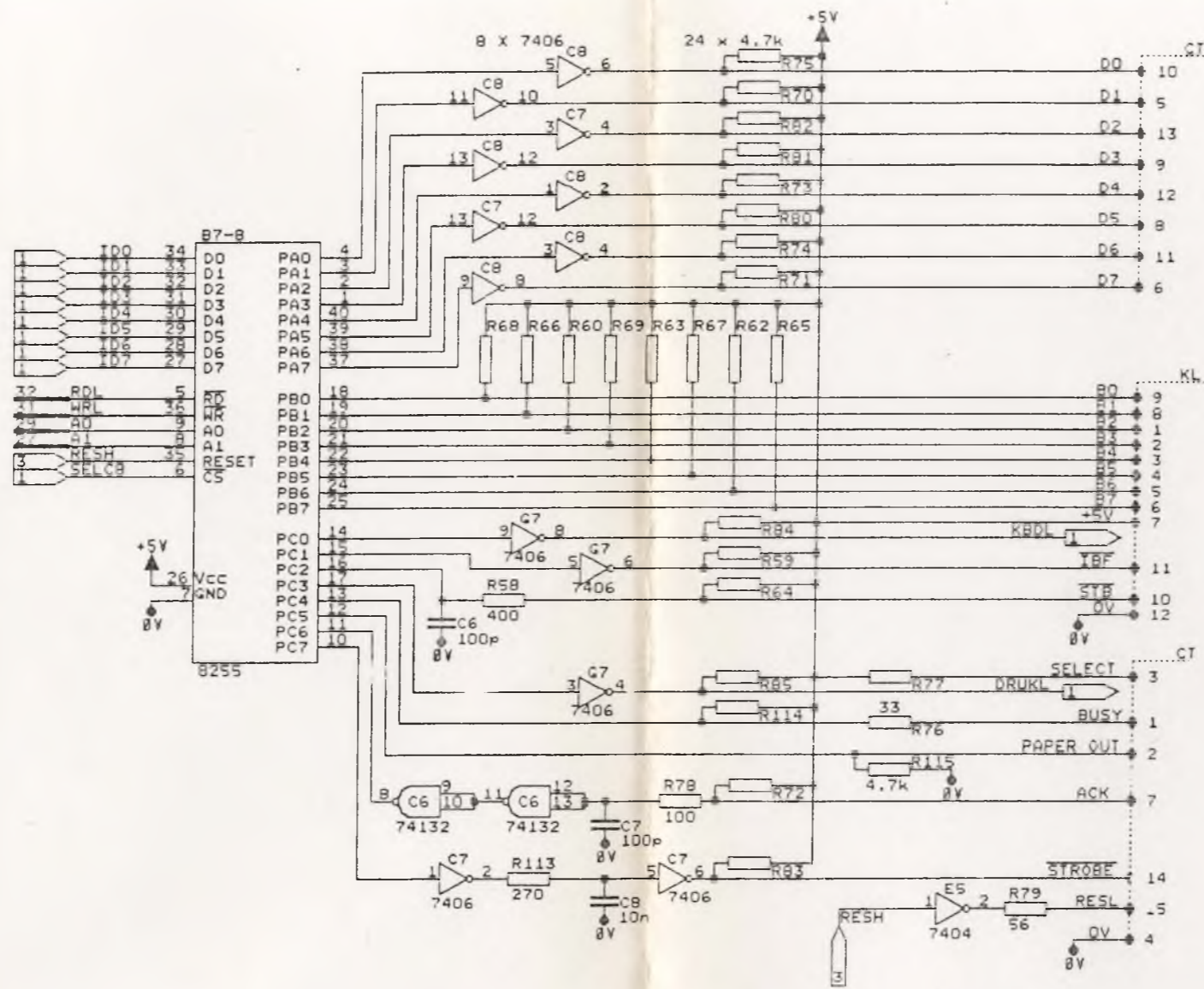


Uwaga 1: Wartość C2 zawiera się w granicach 10-40p.
 Uwaga 2: Przy stosowaniu układu 2797A, wartość C3=33p.

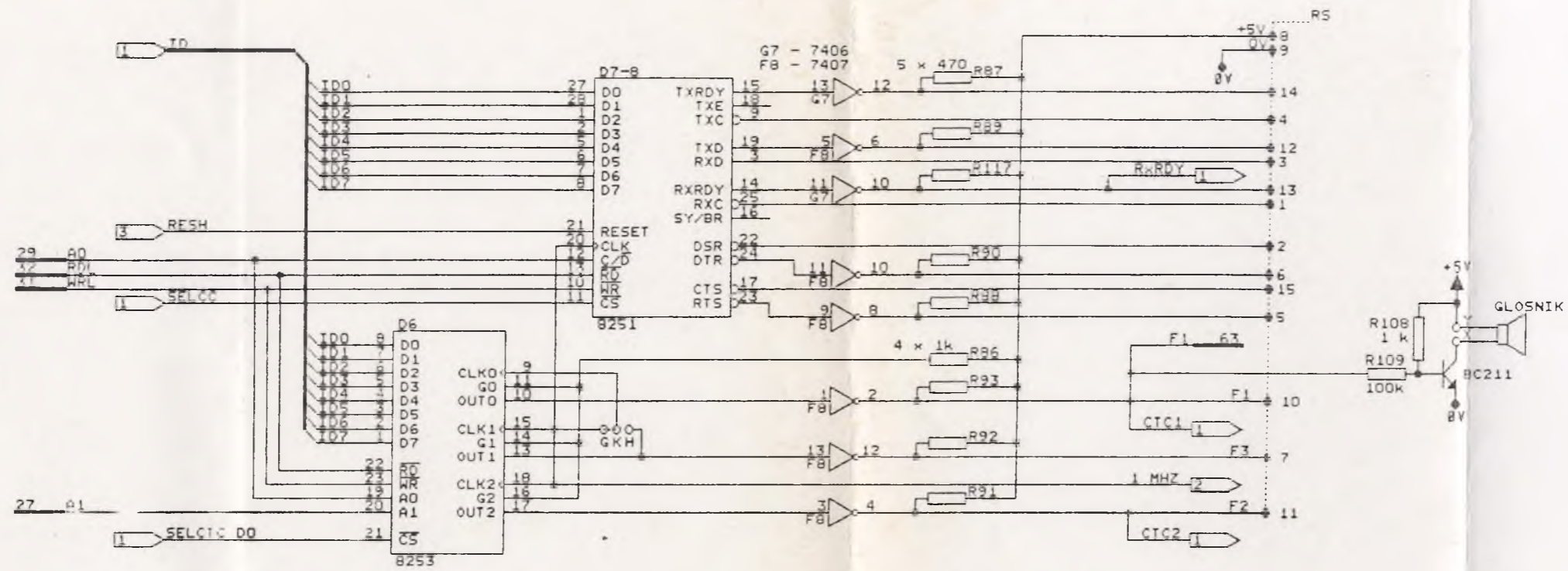
KOMPUTER "KARAT" WERSJA W-2
 ZESPÓŁ INTERFEJSÓW, KONTROLER DYSKU-2.
 ARKUSZ - 2 26-11-1988
 ELKOR PRUSZCZ GDANSKI
 DRZYMAŁY 12



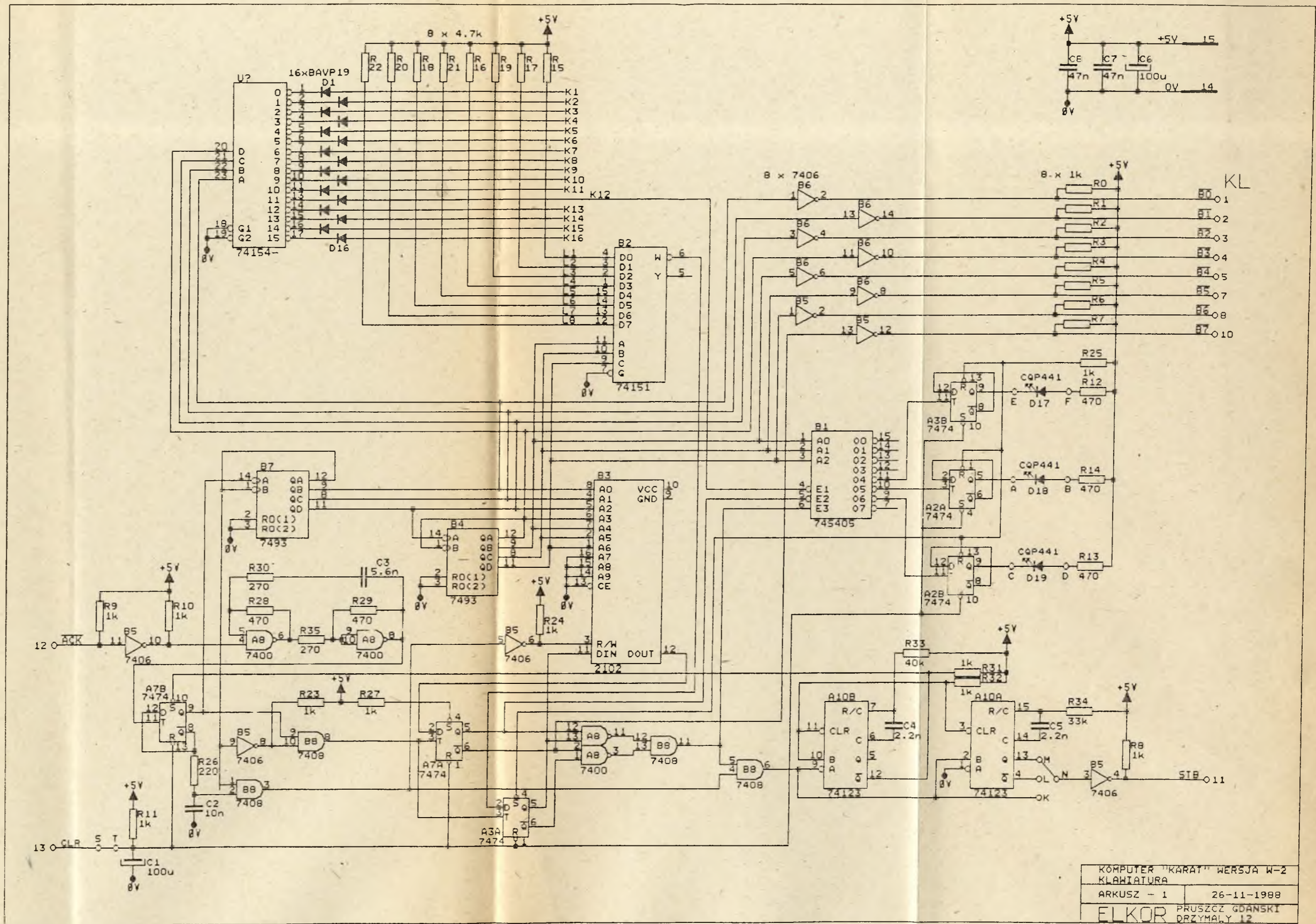
KOMPUTER "KARAT" WERSJA W-2	
ZESPÓŁ INEREFJSÓW, KONTROLER MAGISTRALI	
ARKUSZ - 3	26-11-1988
ELKOR	PRUSZCZ GDANSKI DRZYMAŁY 12



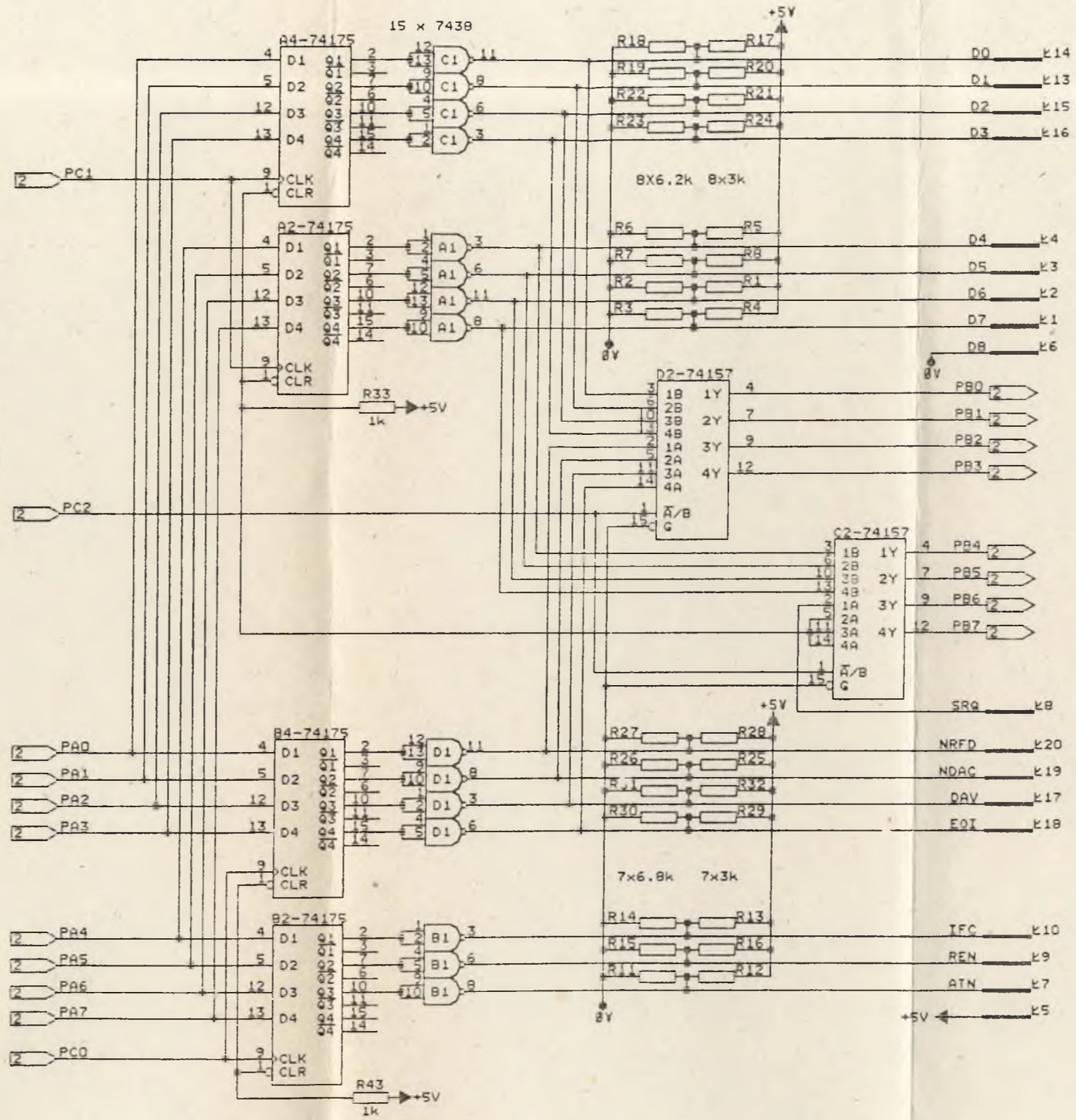
KOMPUTER "KARAT" WERSJA W-2
 KONTROLER KLAWIATURY I DRUKARKI
 ARKUSZ - 4 26-11-1988
 ELKOR PRUSZCZ GDANSKI
 DRZYMALY 12



KOMPUTER "KARAT" WERSJA W-2	
KONTROLER RS-232	
ARKUSZ - 5	26-11-1988
ELKOR	PRUSZCZ GDANSKI DRZYMAŁY 12



KOMPUTER "KARAT" WERSJA W-2
 KLAWIATURA
 ARKUSZ - 1 26-11-1988
 ELKOR PRUSZCZ GDANSKI
 DRZYMALY 12



KOMPUTER "KARAT" WERSJA W-2
 INTERFEJS IEC-625
 ARKUSZ - 1 26-11-1988
 ELKOR PRUSZCZ GDANSKI
 DRZYMALY 12

WYKAZ
ELEMENTÓW
SCHEMATY
MONTAŻOWE
KARAT-2

az elementów płytki C P U
w/g numeracji elementów

C1 - 6264	E8 - 74157	H7 - 74132
C3 - 6264	F1 - 6264	H8 - 74123
C4 - 74175	F3 - 8216	
C5 - 74157	F5 - Z-80	
C6 - 8216	F7 - 7474	
C8 - 8216	F8 - 7474	
D1 - 6264	G2 - 7430	
D3 - 6264	G3 - 8216	
D4 - 7407	G4 - 7407	
D5 - 7407	G5 - 7420	
D6 - 74157	G6 - 7404	
D7 - 74132	G7 - 7408	
D8 - 74L504	G8 - 74504	
E7 - 7493	H6 - 7406	

1 - BD138

1 - BAVP19
2 - BAVP19
3 - BAVP19
4 - BAVP19
5 - BAVP19
6 - BAVP21
7 - BAVP19

R26 - 470/0.25W	R51 - 470/0.25W
R27 - 470/0.25W	R52 - 470/0.25W
R28 - 470/0.25W	R53 - 470/0.25W
R29 - 470/0.25W	R54 - 6k7/0.25W
R30 - 470/0.25W	R55 - 1k/0.25W
R31 - 470/0.25W	R56 - 15k/0.25W
R32 - 470/0.25W	R57 - 4k7/0.25W
R33 - 470/0.25W	R58 - 100/0.25W
R34 - 470/0.25W	R59 - 1k/0.25W
R35 - 470/0.25W	R60 - 22/0.25W
R36 - 470/0.25W	R61 - 60/0.25W
R37 - 10k/0.25W	R62 - 60/0.25W
R38 - 10k/0.25W	R63 - 1k/0.25W
R39 - 470/0.25W	R64 - 360/0.25W
R40 - 470/0.25W	R65 - 1k/0.25W
R41 - 470/0.25W	R66 - 4k7/0.25W

Wykaz elementów płytki C P U
w/g numeracji elementów

1. Układy scalone :

A1 - 6264	C1 - 6264	E8 - 74157	H7 - 74132
A3 - 2764	C3 - 6264	F1 - 6264	H8 - 74123
A4 - 7410	C4 - 74175	F3 - 8216	
A5 - 4164	C5 - 74157	F5 - 2-80	
A5.5 - 4164	C6 - 8216	F7 - 7474	
A6 - 4164	C8 - 8216	F8 - 7474	
A6.5 - 4164	D1 - 6264	G2 - 7430	
A7 - 4164	D3 - 6264	G3 - 8216	
A7.5 - 4164	D4 - 7407	G4 - 7407	
A8 - 4164	D5 - 7407	G5 - 7420	
A8.5 - 4164	D6 - 74157	G6 - 7404	
B1 - 6264	D7 - 74132	G7 - 7408	
B3 - 6264	D8 - 74LS04	G8 - 74504	
B4 - 8205	E7 - 7493	H6 - 7406	

2. Tranzystory :

T1 - BD138

3. Diody :

D1 - BAUP19
D2 - BAUP19
D3 - BAUP19
D4 - BAUP19
D5 - BAUP19
D6 - BAUP21
D7 - BAUP19

2. Rezystory :

R 1 - 330/0.25W	R26 - 470/0.25W	R51 - 470/0.25W
R 2 - 330/0.25W	R27 - 470/0.25W	R52 - 470/0.25W
R 3 - 1k/0.25W	R28 - 470/0.25W	R53 - 470/0.25W
R 4 - 33/0.25W	R29 - 470/0.25W	R54 - 6k7/0.25W
R 5 - 33/0.25W	R30 - 470/0.25W	R55 - 1k/0.25W
R 6 - 33/0.25W	R31 - 470/0.25W	R56 - 15k/0.25W
R 7 - 33/0.25W	R32 - 470/0.25W	R57 - 4k7/0.25W
R 8 - 33/0.25W	R33 - 470/0.25W	R58 - 100/0.25W
R 9 - 33/0.25W	R34 - 470/0.25W	R59 - 1k/0.25W
R10 - 33/0.25W	R35 - 470/0.25W	R60 - 22/0.25W
R11 - 33/0.25W	R36 - 470/0.25W	R61 - 50/0.25W
R12 - 470/0.25W	R37 - 10k/0.25W	R62 - 360/0.25W
R13 - 470/0.25W	R38 - 10k/0.25W	R63 - 1k/0.25W
R14 - 470/0.25W	R39 - 470/0.25W	R64 - 360/0.25W
R15 - 470/0.25W	R40 - 470/0.25W	R65 - 1k/0.25W
R16 - 470/0.25W	R41 - 470/0.25W	R66 - 4k7/0.25W

R17 - 470/0.25W	R42 - 470/0.25W	R67 - 470/0.25W
R18 - 470/0.25W	R43 - 470/0.25W	R68 - 470/0.25W
R19 - 470/0.25W	R44 - 470/0.25W	R69 - 470/0.25W
R20 - 470/0.25W	R45 - 470/0.25W	R70 - 470/0.25W
R21 - 470/0.25W	R46 - 470/0.25W	R71 - 470/0.25W
R22 - 470/0.25W	R47 - 470/0.25W	R72 - 470/0.25W
R23 - 470/0.25W	R48 - 470/0.25W	R73 - 470/0.25W
R24 - 470/0.25W	R49 - 470/0.25W	R74 - 470/0.25W
R25 - 470/0.25W	R50 - 470/0.25W	R75 - 1k/0.25W

4. Kondensatory :

C 1 - 180p	C 9 - 100n	C16 - 100n
C 2 - 470p	C10 - 100n	C17 - 47n
C 3 - 10u/16V	C11 - 100n	C18 - 47n
C 4 - 1n	C12 - 100n	C19 - 47n
C 5 - 47u/16V	C13 - 100n	C20 - 47n
C 6 - 68p	C14 - 100n	C21 - 200n
C 7 - 100u/16V	C15 - 100n	C22 - 47u/16V
C 8 - 100u/16V		

6. Inne :

Rezonator kwarcowy 4 MHz
Złącze ELTRA B11 064

Wykaz elementów płytki V - 2
w/g numeracji elementów

1. Układy scalone :

A1 - 41256	D2 - 8212	F4 - 7400	G1 - 41256
A1 ² - 41256	D3 - 74193	F5 - 7474	G1 ² - 41256
A1 ³ - 41256	D4 - 74193	F6 - 7408	G1 ³ - 41256
A2 - 41256	D5 - 74804	F7 - 7493	G2 - 41256
A2 ² - 41256	D6 - 74800	G2 - 7442	G2 ² - 41256
A3 - 41256	D7 - 7474	G3 - 7430	G3 - 41256
A3 ² - 41256	D8 - 7442	G4 - 7474	G3 ² - 41256
A4 - 41256	E1 - 8255	G5 - 74132	G4 - 41256
A5 - 74153	E3 - 7402	G6 - 7474	
A6 - 74153	E4 - 7400	G7 - Nieobsadzone	
A7 - 74193	E5 - 74155	G8 - 748424	
B5 - 74153	E6 - 7404	G5 - 74153	
B6 - 74153	E7 - 7404	G6 - 7493	
B7 - 7493	F2 - 74165	G7 - 7493	
B8 - 74193	F3 - 7404	G8 - 7493	

2. Rezystory :

R 1 - 2K2/0.25W	R14 - 270/0.25W	R27 - 470/0.25W
R 2 - 4K7/0.25W	R15 - 270/0.25W	R28 - 470/0.25W
R 3 - 4K7/0.25W	R16 - 270/0.25W	R29 - 470/0.25W
R 4 - 100/0.25W	R17 - 4K7/0.25W	R30 - 470/0.25W
R 5 - 510/0.25W	R18 - 4K7/0.25W	R31 - 470/0.25W
R 6 - TVP-115 470 A	R19 - 4K7/0.25W	R32 - 470/0.25W
R 7 - 1K/0.25W	R20 - 4K7/0.25W	R33 - 470/0.25W
R 8 - 510/0.25W	R21 - 4K7/0.25W	R34 - 470/0.25W
R 9 - 4K7/0.25W	R22 - 4K7/0.25W	R35 - 470/0.25W
R10 - 220/0.25W	R23 - 1K/0.25W	R36 - 470/0.25W
R11 - 270/0.25W	R24 - 470/0.25W	R37 - 470/0.25W
R12 - 270/0.25W	R25 - 470/0.25W	R38 - 470/0.25W
R13 - 4K7/0.25W	R26 - 470/0.25W	R39 - 470/0.25W

3. Kondensatory :

Ckw - 27p	C10 - 100p	C19 - 100n
C1 - 27p	C11 - 100u/16V	C20 - 100n
C2 - 1n	C12 - 100u/16V	C21 - 100n
C3 - 470p	C13 - 100n	C22 - 100n
C4 - 100p	C14 - 100n	C23 - 100n
C5 - 100u/16V	C15 - 100n	C24 - 100n
C6 - 100p	C16 - 100n	C25 - 100n
C7 - 100p	C17 - 100n	C26 - 100n
C8 - 100p	C18 - 100n	C27 - 100n
C9 - 220p		

4. Diody :

D1 - BAVP19

D2 - BAVP19

5. Tranzystory :

T1 - BSXP94

6. Inne :

Cewka indukcyjna typ LR

Rezonator kwarcowy 8 MHz

Złącze ELTRA 811 064

Wykaz elementów płytki I N I - 2
w/g numeracji elementów

1. Układy scalone :

B1 - 8216	D1 - 7402	F3 - 7404
B2 - 8216	D2 - 7402	F4 - 7430
B3 - 8216	D3 - 7410	F5 - 7474
B4, B5 - 2797	D4 - 7438	F6 - 8214
B7, B8 - 8255	D5 - 7400	F8 - 7407
C1 - 7438	D6 - 8253	G1 - 7493
C2 - 7438	D8 - 8251	G2 - 7438
C3 - 8216	E1, F1 - 74154	G3 - 7410
C4 - 74174	E2 - 7474	G4 - 745405
C5 - 7406	E3 - 7400	G5 - 7493
C6 - 74132	E4 - 7402	G6 - 74132
C7 - 7406	E5 - 7404	G7 - 7406
C8 - 7406	F2 - 7474	H5 - 74123

2. Rezystory :

R 1 - 10k/0.25W	R 41 - 360/0.25W	R 81 - 4k7/0.25W
R 2 - 10k/0.25W	R 42 - 360/0.25W	R 82 - 4k7/0.25W
R 3 - 10k/0.25W	R 43 - 360/0.25W	R 83 - 4k7/0.25W
R 4 - 10k/0.25W	R 44 - 360/0.25W	R 84 - 4k7/0.25W
R 5 - 10k/0.25W	R 45 - 360/0.25W	R 85 - 4k7/0.25W
R 6 - 4k7/0.25W	R 46 - 1k2/0.25W	R 86 - 4k7/0.25W
R 7 - 4k7/0.25W	R 47 - 1k2/0.25W	R 87 - 4k7/0.25W
R 8 - 4k7/0.25W	R 48 - 1k2/0.25W	R 88 - 4k7/0.25W
R 9 - 4k7/0.25W	R 49 - 1k2/0.25W	R 89 - 4k7/0.25W
R 10 - 4k7/0.25W	R 50 - 2k4/0.25W	R 90 - 4k7/0.25W
R 11 - 4k7/0.25W	R 51 - 2k4/0.25W	R 91 - 4k7/0.25W
R 12 - 4k7/0.25W	R 52 - 2k4/0.25W	R 92 - 4k7/0.25W
R 13 - 1k/0.25W	R 53 - 2k4/0.25W	R 93 - 4k7/0.25W
R 14 - 1k/0.25W	R 54 - 56/0.25W	R 94 - 4k7/0.25W
R 15 - 4k7/0.25W	R 55 - 56/0.25W	R 95 - 4k7/0.25W
R 16 - 4k7/0.25W	R 56 - 56/0.25W	R 96 - 4k7/0.25W
R 17 - 4k7/0.25W	R 57 - 56/0.25W	R 97 - 4k7/0.25W
R 18 - 470/0.25W	R 58 - 360/0.25W	R 98 - 4k7/0.25W
R 19 - 470/0.25W	R 59 - 1k/0.25W	R 99 - 4k7/0.25W
R 20 - 470/0.25W	R 60 - 2k4/0.25W	R100 - 4k7/0.25W
R 21 - 1k/0.25W	R 61 - 1k2/0.25W	R101 - 4k7/0.25W
R 22 - 24k/0.25W	R 62 - 1k2/0.25W	R102 - 4k7/0.25W
R 23 - 24k/0.25W	R 63 - 1k2/0.25W	R103 - 33/0.25W
R 24 - 1k/0.25W	R 64 - 1k2/0.25W	R104 - 4k7/0.25W
R 25 - 360/0.25W	R 65 - 1k2/0.25W	R105 - 4k7/0.25W
R 26 - 330/0.25W	R 66 - 2k4/0.25W	R106 - 100/0.25W
R 27 - 330/0.25W	R 67 - 2k4/0.25W	R107 - 4k7/0.25W

R 28 - 330/0.25W	R 68 - 2k4/0.25W	R108 - 270/0.25W
R 29 - 330/0.25W	R 69 - 2k4/0.25W	R109 - 56/0.25W
R 30 - 330/0.25W	R 70 - 56/0.25W	R110 - 470/0.25W
R 31 - 330/0.25W	R 71 - 56/0.25W	R111 - 470/0.25W
R 32 - 330/0.25W	R 72 - 56/0.25W	R112 - 470/0.25W
R 33 - 330/0.25W	R 73 - 56/0.25W	R113 - 470/0.25W
R 34 - TVP-115 22k A	R 74 - 470/0.25W	R114 - 470/0.25W
R 35 - TVP-115 22k A	R 75 - 56/0.25W	R115 - 1k/0.25W
R 36 - 1k/0.25W	R 76 - 470/0.25W	R116 - 1k/0.25W
R 37 - 4k7/0.25W	R 77 - 56/0.25W	R117 - 1k/0.25W
R 38 - 4k7/0.25W	R 78 - 1k/0.25W	R118 - 1k/0.25W
R 39 - 1k/0.25W	R 79 - 22/0.25W	R119 - 1k/0.25W
R 40 - 6k8/0.25W	R 80 - 4k7/0.25W	R120 - 100k/0.25W

3. Kondensatory :

C 1 - 470p	C 8 - 10n	C15 - 47n
C 2 - TCF-10d 10/40p	C 9 - 100p	C16 - 47n
C 3 - 33p	C10 - 100u/16V	C17 - 47n
C 4 - 220n	C11 - 100u/16V	C18 - 47n
C 5 - 100u/16V	C12 - 100u/16V	C19 - 47n
C 6 - 51p	C13 - 47n	
C 7 - 100p	C14 - 47n	

4. Diody :

D1 - BAVP21

5. Transystory :

T1 - BC211

6. Inne :

Złącze ELTRA 811 064
 Złącze WMP15.1/2.5
 Złącze WMP15.1/2.5
 Złącze WMP15.1/2.5

Wykaz elementów płytki "KLAWIATURA"
w/g numeracji elementów

1. Układy scalone :

A 2 - 7474	B1 - 74S405	B6 - 7406
A 3 - 7474	B2 - 74151	B7 - 7493
A 7 - 7474	B3 - 2102	B8 - 7408
A 8 - 7400	B4 - 7493	B9 - 74154
A10 - 74121	B5 - 7406	

2. Rezystory :

R 1 - 4k7/0.25W	R17 - 1k/0.25W	R33 - 1k/0.25W
R 2 - 4k7/0.25W	R18 - 1k/0.25W	R34 - 33k/0.25W
R 3 - 4k7/0.25W	R19 - 1k/0.25W	R35 - 1k/0.25W
R 4 - 4k7/0.25W	R20 - 1k/0.25W	R36 - 40k/0.25W
R 5 - 4k7/0.25W	R21 - 1k/0.25W	
R 6 - 4k7/0.25W	R22 - 1k/0.25W	
R 7 - 4k7/0.25W	R23 - 1k/0.25W	
R 8 - 4k7/0.25W	R24 - 1k/0.25W	
R 9 - 1k/0.25W	R25 - 1k/0.25W	
R10 - 1k/0.25W	R26 - 1k/0.25W	
R11 - 1k/0.25W	R27 - 1k/0.25W	
R12 - 220/0.25W	R28 - 1k/0.25W	
R13 - 270/0.25W	R29 - 470/0.25W	
R14 - 470/0.25W	R30 - 470/0.25W	
R15 - 270/0.25W	R31 - 470/0.25W	
R16 - 470/0.25W	R32 - 1k/0.25W	

3. Kondensatory :

C1 - 100u/16V	C5 - 2.2n
C2 - 10n	C6 - 100u/16V
C3 - 5.6n	C7 - 47n
C4 - 2.2n	C8 - 47n

4. Diody :

D 1 D16 - BAV9 17
D17, D18, D19 - CQP 441a

5. inne :

Kontaktowy typ n - 24 -112 /DULAN-UNIPRA/

Wykaz elementów płytki V - 24/Mag
w/g numeracji elementów

1. Układy scalone :

A1 - 1489	B2 - 7493
A2 - 7474	B3 - 7404
A3 - 7400	B4 - 1488
A4 - 74157	C4 - 7486
B1 - 74121	D4 - 74069

2. Rezystory :

R 1 - 22k/0.25W	R14 - 4k7/0.25W
R 2 - 1k/0.25W	R15 - 4k7/0.25W
R 3 - 1k/0.25W	R16 - TOP-115 1k A
R 4 - 1k/0.25W	R17 - 9k1/0.25W
R 5 - 2k/0.25W	R18 - 1k/0.25W
R 6 - 22k/0.25W	R19 - 200/0.25W
R 7 - 10k/0.25W	R20 - 10k/0.25W
R 8 - 100k/0.25W	R21 - 1k/0.25W
R 9 - 10k/0.25W	R22 - 1k/0.25W
R10 - 200k/0.25W	R23 - 2k7/0.25W
R11 - 1k/0.25W	R24 - 2k4/0.25W
R12 - 4k7/0.25W	R25 - 360/0.25W
R13 - 4k7/0.25W	R26 - 2k7/0.25W

3. Kondensatory :

C1 - 0.1u	C6 - 1n
C2 - 47u/16V	C7 - 0.1u
C3 - 220n	C8 - 47u/16V
C4 - 20p	C9 - 47n
C5 - 220n	

4. Diody :

D1 - BAVP19	D5 - BAVP19
D2 - BAVP19	D6 - BAVP19
D3 - BAVP19	D7 - BAVP21
D4 - BAVP19	

5. Tranzystory :

T1 - BC107
T2 - BC178

6. Inne :

K1 - kontaktron K-B/406-1

Złącze WWP3/2.5

Złącze WWP5/2.5

Złącze WWP10/2.5

Złącze WWP15/2.5

Wykaz elementów płytki IEC - 625
w/g numeracji elementów

1. Układy scalone :

A1 - 7438	B1 - 7438	C1 - 7438	D1 - 7438
A2 - 74175	B2 - 74175	C2 - 74157	D2 - 74157
A4 - 74175	B4 - 74175	C4 - 74175	D4 - 7474
A5 - 7400	B5 - 8216	C5 - 8216	D5 - 7485
A3, B3, C3 - 8255			

2. Rezystory :

R 1 - 6k2/0.25W	R17 - 1k/0.25W	R33 - 1k/0.25W
R 2 - 3k/0.25W	R18 - 6k8/0.25W	R34 - 22/0.25W
R 3 - 6k2/0.25W	R19 - 3k/0.25W	R35 - 22/0.25W
R 4 - 3k/0.25W	R20 - 6k8/0.25W	R36 - 22/0.25W
R 5 - 6k2/0.25W	R21 - 3k/0.25W	R37 - 22/0.25W
R 6 - 3k/0.25W	R22 - 6k8/0.25W	R38 - 22/0.25W
R 7 - 6k2/0.25W	R23 - 3k/0.25W	R39 - 22/0.25W
R 8 - 3k/0.25W	R24 - 6k8/0.25W	R40 - 22/0.25W
R 9 - 6k2/0.25W	R25 - 3k/0.25W	R41 - 22/0.25W
R10 - 3k/0.25W	R26 - 6k8/0.25W	R42 - 200/0.25W
R11 - 6k2/0.25W	R27 - 3k/0.25W	R43 - 22/0.25W
R12 - 3k/0.25W	R28 - 6k8/0.25W	R44 - 22/0.25W
R13 - 6k2/0.25W	R29 - 3k/0.25W	R45 - 22/0.25W
R14 - 3k/0.25W	R30 - 6k8/0.25W	R46 - 22/0.25W
R15 - 6k2/0.25W	R31 - 3k/0.25W	R47 - 1k/0.25W
R16 - 3k/0.25W	R32 - 1k/0.25W	

3. Kondensatory :

C1 - 100p
C2 - 47u/16V
C3 - 47n
C4 - 47n

4. Inne :

Złącze WNF15.1/2.5
Złącze WNF20.1/2.5

Wykaz elementów płytki C P U
w/g asortymentu

1. Układy scalone :

6264	- 8 szt	7406	- 1 szt	7493	- 1 szt
2764	- 1 szt	7407	- 3 szt	74123	- 1 szt
4164	- 8 szt	7408	- 1 szt	74132	- 2 szt
Z-80	- 1 szt	7410	- 1 szt	74157	- 3 szt
8216	- 4 szt	7420	- 1 szt	74175	- 1 szt
7404	- 1 szt	7430	- 1 szt	8205	- 1 szt
74LS04	- 2 szt	7474	- 2 szt		

2. Tranzystory :

BD138 - 1 szt

3. Diody :

BAVP19 - 6 szt
BAVP21 - 1 szt

4. Rezystory :

22/0.25W - 1 szt
33/0.25W - 8 szt
330/0.25W - 2 szt
360/0.25W - 3 szt
100/0.25W - 1 szt
470/0.25W - 47 szt
1k/0.25W - 6 szt
4k7/0.25W - 2 szt
6k7/0.25W - 1 szt
10k/0.25W - 2 szt
15k/0.25W - 1 szt

4. kondensatory :

58p - 1 szt
180p - 1 szt
470p - 1 szt
1n - 1 szt
4/n - 4 szt
100n - 8 szt
220n - 1 szt
10u/16V - 1 szt
47u/16V - 2 szt
100u/16V - 2 szt

6. Inne :

Rezonator kwarcowy 40kHz - 1 szt
Złącze ELTRA 811064 - 1 szt

Wykaz elementów płytki V - 2
w/g asortymentu

1. Układy scalone :

7400 - 2 szt	7474 - 4 szt	74165 - 1 szt
74S00 - 1 szt	7493 - 5 szt	74193 - 4 szt
7402 - 1 szt	7442 - 2 szt	74S424 - 1 szt
7404 - 3 szt	74132 - 1 szt	41256 - 16 szt
74S04 - 1 szt	74153 - 5 szt	8212 - 1 szt
7408 - 1 szt	74155 - 1 szt	8255 - 1 szt
7430 - 1 szt		

2. Rezystory :

100/0.25W - 1 szt	510/0.25W - 2 szt
220/0.25W - 1 szt	1k/0.25W - 2 szt
270/0.25W - 5 szt	2k2/0.25W - 1 szt
470/0.25W - 16 szt	4k7/0.25W - 10 szt
TVP-115 470 - 1 szt	

3. Kondensatory :

27p - 2 szt
100p - 9 szt
220p - 1 szt
470p - 1 szt
1n - 1 szt
100n - 14 szt
100u/16V - 3 szt

4. Diody :

BAVP19 - 2 szt

4. Tranzystory :

BSXP94 - 1 szt

4. Inne :

Cewka indukcyjna typ LR	- 1 szt
Rezonator kwarcowy 8 MHz	- 1 szt
Złącze ELTRA B11 064	- 1 szt

Wykaz elementów płytki I N T - 2
w/g asortymentu

1. Układy scalone :

7400	- 2 szt	7438	- 4 szt	8216	- 4 szt
7402	- 3 szt	7474	- 3 szt	8214	- 1 szt
7404	- 2 szt	7493	- 2 szt	749405	- 1 szt
7406	- 4 szt	74132	- 2 szt	8251	- 1 szt
7407	- 1 szt	74154	- 1 szt	8253	- 1 szt
7410	- 2 szt	74174	- 1 szt	8255	- 1 szt
7430	- 1 szt	74123	- 1 szt	2797A	- 1 szt

2. Rezystory :

22/0.25W	- 1 szt
33/0.25W	- 1 szt
56/0.25W	- 11 szt
100/0.25W	- 1 szt
270/0.25W	- 1 szt
330/0.25W	- 8 szt
360/0.25W	- 7 szt
470/0.25W	- 10 szt
1k/0.25W	- 13 szt
1k2/0.25W	- 9 szt
2k4/0.25W	- 9 szt
4k7/0.25W	- 38 szt
6k8/0.25W	- 1 szt
10k/0.25W	- 5 szt
24k/0.25W	- 2 szt
100k/0.25W	- 1 szt
pot. TVP-115 22k A	- 2 szt

3. Kondensatory :

33p	- 1 szt	10n	- 1 szt
10P-10d 10/40p	- 1 szt	47n	- 7 szt
51p	- 1 szt	220n	- 1 szt
100p	- 2 szt	100u/16V	- 4 szt
470p	- 1 szt		

4. Diody :

BAVP21 - 1 szt

5. Tranzystory :

BC211 - 1 szt

6. Inne :

Złącze ELTRA 811 064 - 1 szt

Złącze WWP15.1/2.5 - 3 szt

Wykaz elementów płytki "KLAWIATURA"
w/g asortymentu

1. Układy scalone :

7400 - 1 szt
7406 - 2 szt
7408 - 1 szt
7474 - 3 szt
7493 - 2 szt
74121 - 1 szt
74151 - 1 szt
74154 - 1 szt
74S405 - 1 szt
2102 - 1 szt

2. Rezystory :

220/0.25W - 1 szt
270/0.25W - 2 szt
470/0.25W - 5 szt
1k/0.25W - 18 szt
4k7/0.25W - 8 szt
33k/0.25W - 1 szt
40k/0.25W - 1 szt

3. Kondensatory :

2.2n - 2 szt
5.6n - 1 szt
10n - 1 szt
47n - 2 szt
100u/16V - 2 szt

4. Diody :

BAVP19 - 16 szt
DQP 441 - 3 szt

5. Inne :

Kontaktrony H - 24 - 112 - 85 szt

Wykaz elementów płytki V - 24/Mag
w/g asortymentu

1. Układy scalone :

7400 - 1 szt	74121 - 1 szt
7404 - 1 szt	74157 - 1 szt
7474 - 1 szt	74069 - 1 szt
7486 - 1 szt	1488 - 1 szt
7493 - 1 szt	1489 - 1 szt

2. Rezystory :

200/0.25W - 1 szt
360/0.25W - 1 szt
1k/0.25W - 7 szt
TVP-115 1k A - 1 szt
2k/0.25W - 1 szt
2k4/0.25W - 1 szt
2k7/0.25W - 2 szt
4k7/0.25W - 4 szt
9k1/0.25W - 1 szt
10k/0.25W - 3 szt
22k/0.25W - 2 szt
100k/0.25W - 1 szt
200k/0.25W - 1 szt

3. Kondensatory :

20p - 1 szt
47n - 1 szt
100n - 2 szt
220n - 2 szt
47u/16V - 2 szt

4. Diody :

BAVP19 - 6 szt
BAVP21 - 1 szt

5. Transystory :

BC107 - 1 szt
BC178 - 1 szt

6. Inne :

Kontaktron K-8/406-1 - 1 szt
Złącze WWP3/2.5 - 1 szt
Złącze WWP5/2.5 - 1 szt
Złącze WWP10/2.5 - 1 szt
Złącze WWP15/2.5 - 1 szt

Wykaz elementów płytki IEC - 625
w/g asortymentu

1. UKłady scalone :

7400	-	1 szt
7438	-	4 szt
7474	-	1 szt
74157	-	2 szt
74175	-	5 szt
7485	-	1 szt
8216	-	2 szt
8255	-	1 szt

2. Rezystory :

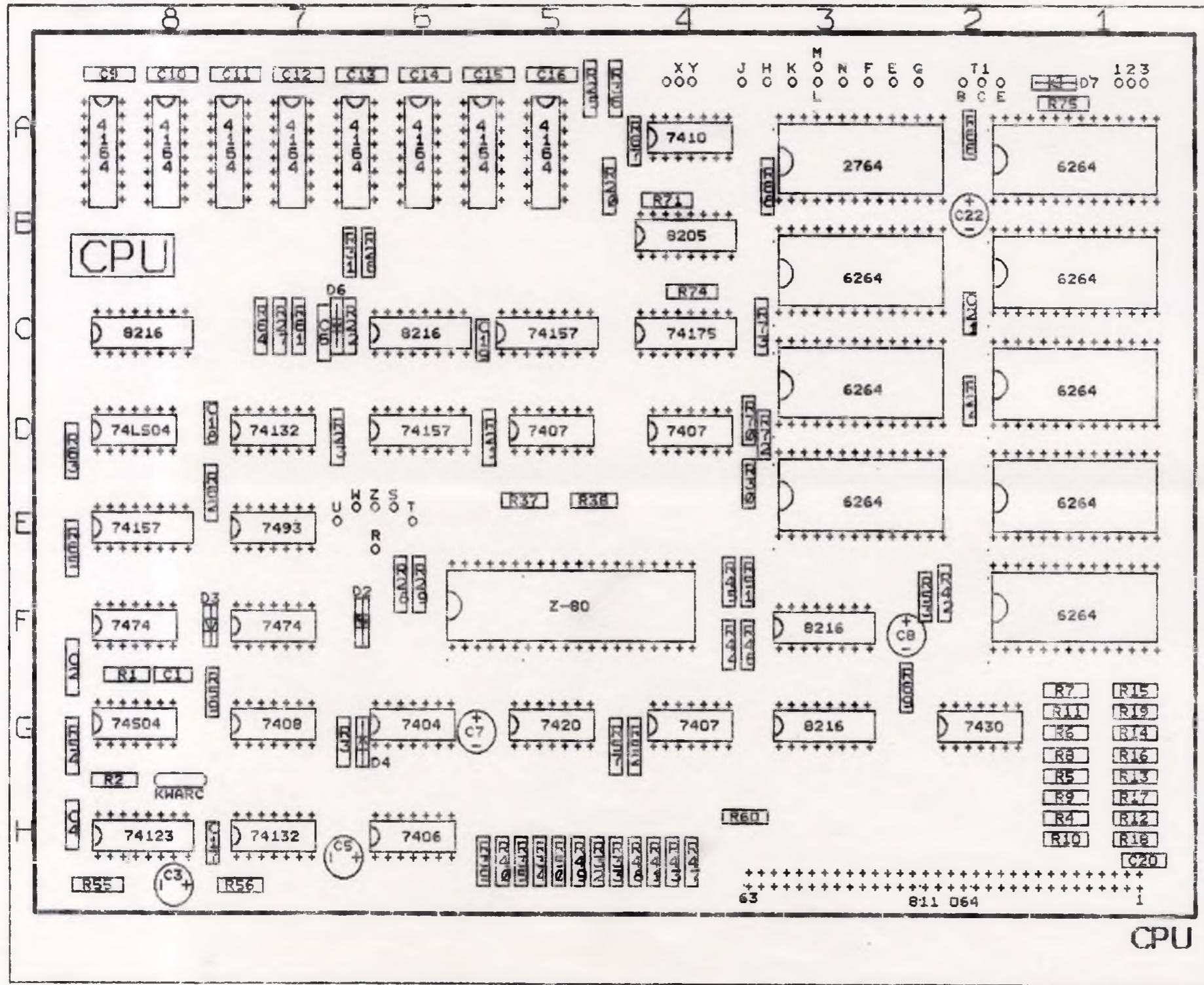
22/0.25W	-	12 szt
200/0.25W	-	1 szt
1k/0.25W	-	4 szt
3k/0.25W	-	15 szt
6k2/0.25W	-	8 szt
6k8/0.25W	-	7 szt

3. Kondensatory :

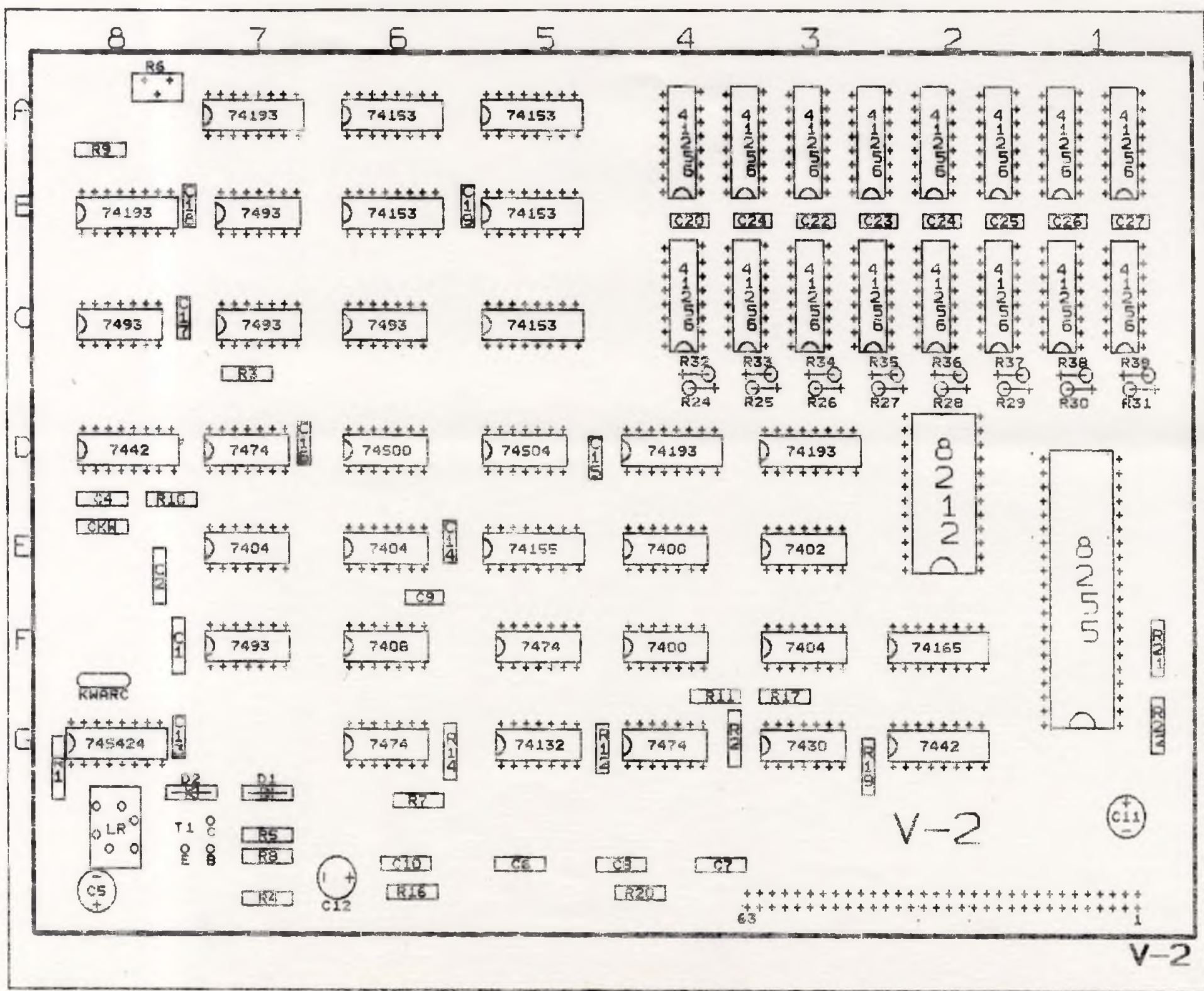
100p	-	1 szt
47n	-	2 szt
47u/16V	-	1 szt

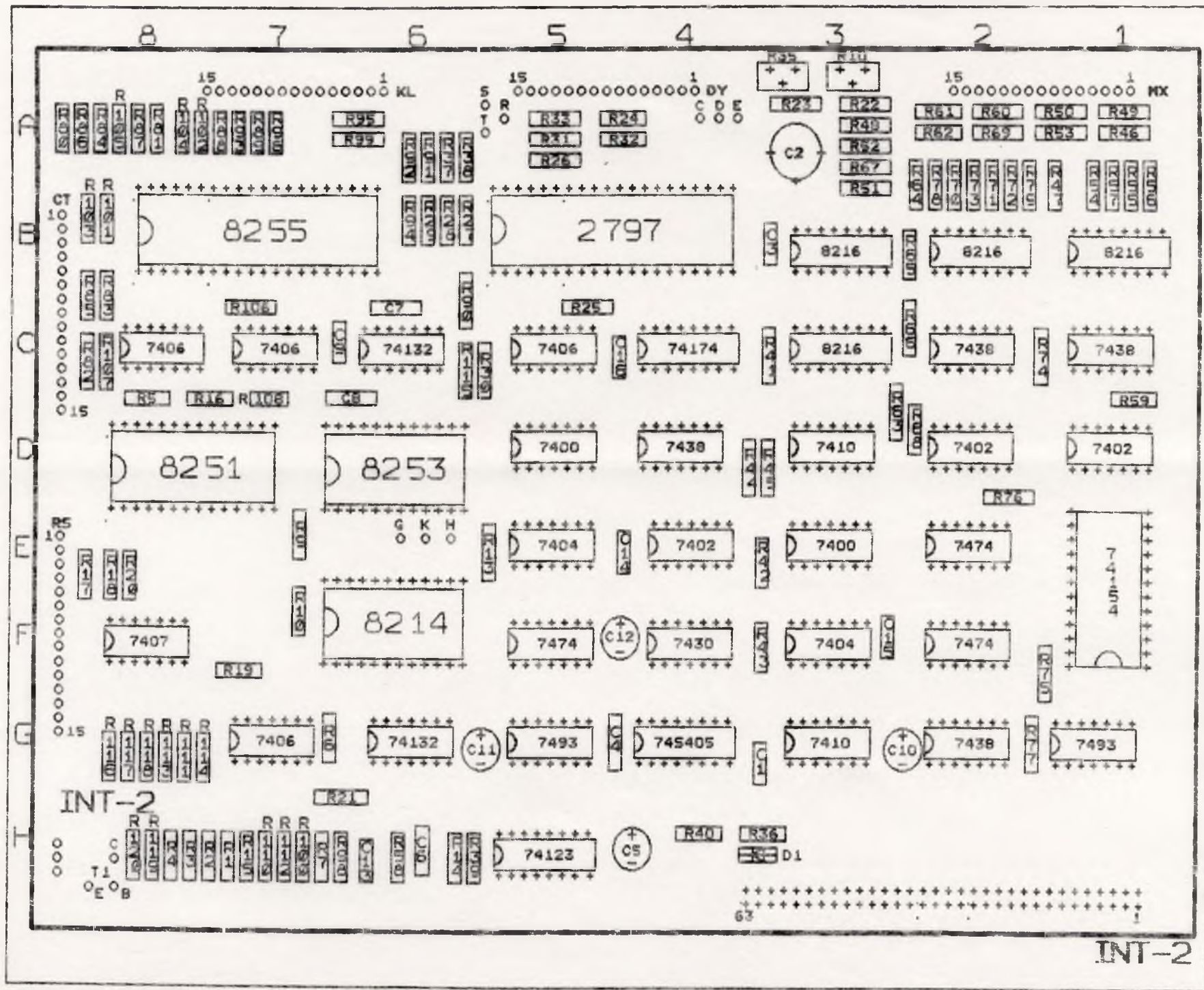
4. Inne :

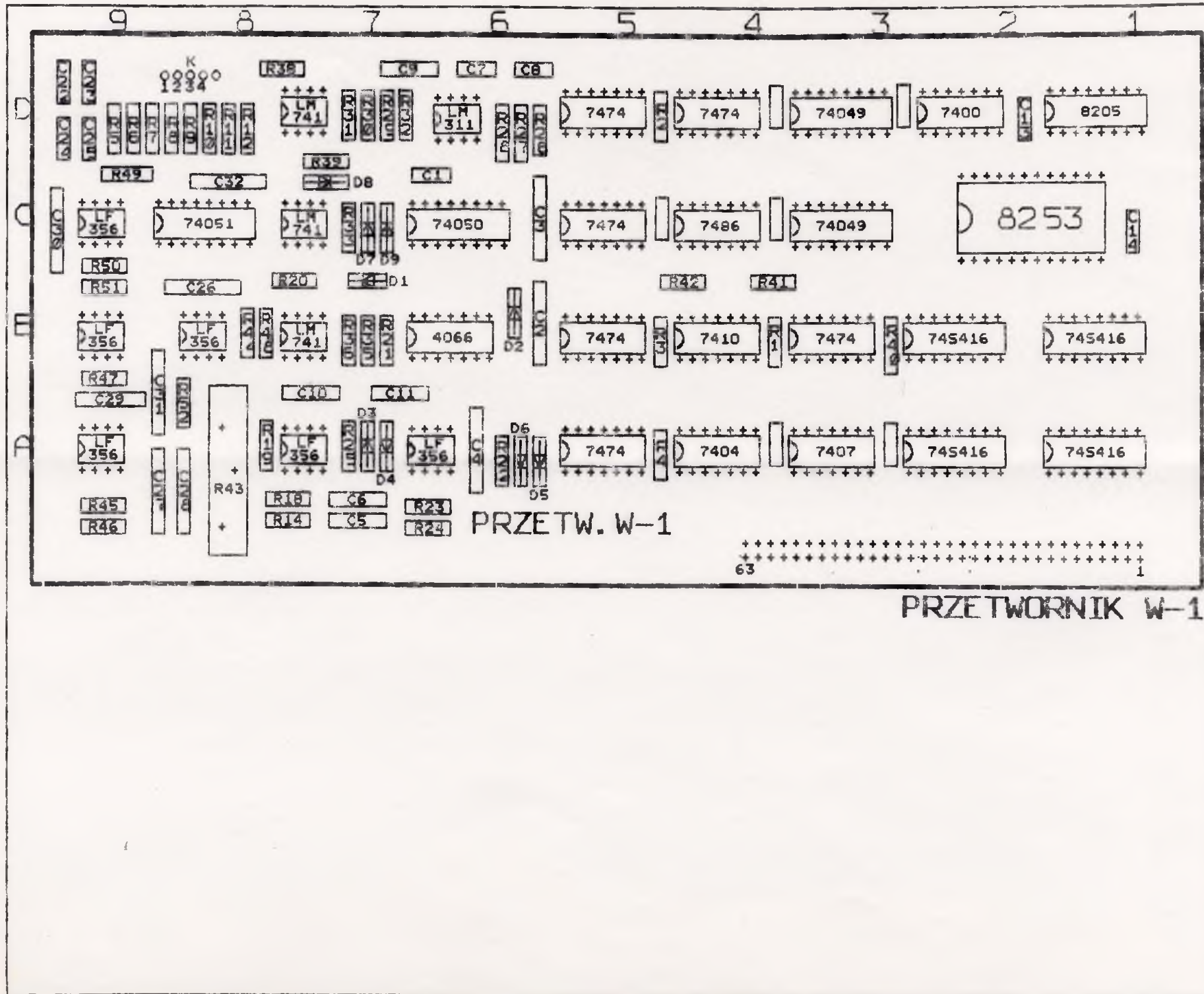
Złącze	WNP15.1/2.5	-	1 szt
Złącze	WNP20.1/2.5	-	1 szt



CPU







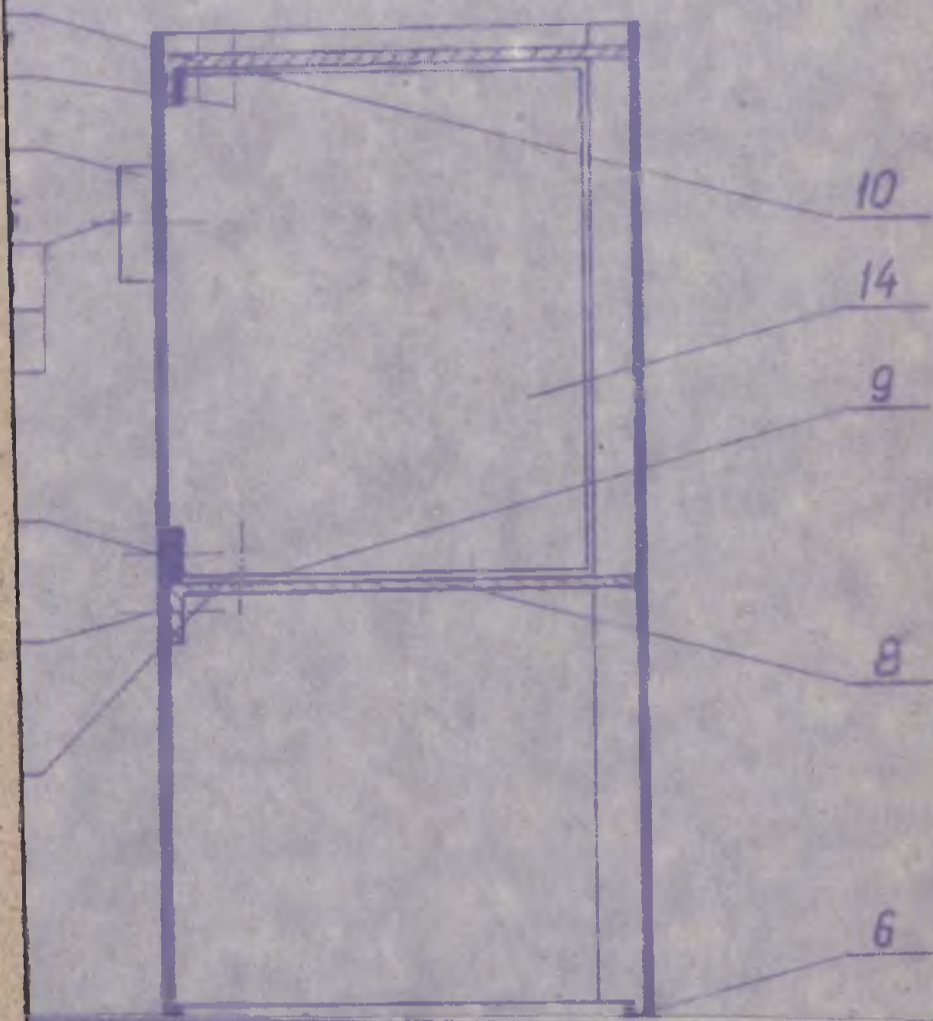
PRZETWORNİK W-1

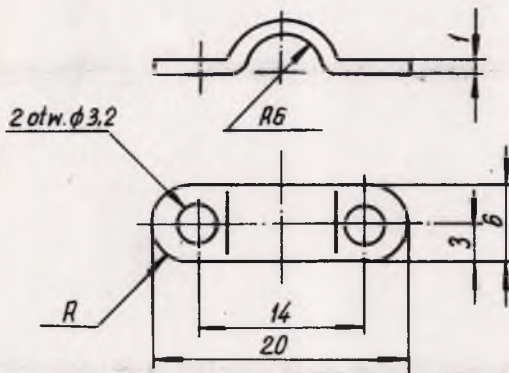
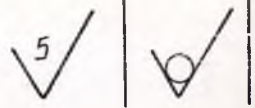
DOKUMENTACJA

MECH.

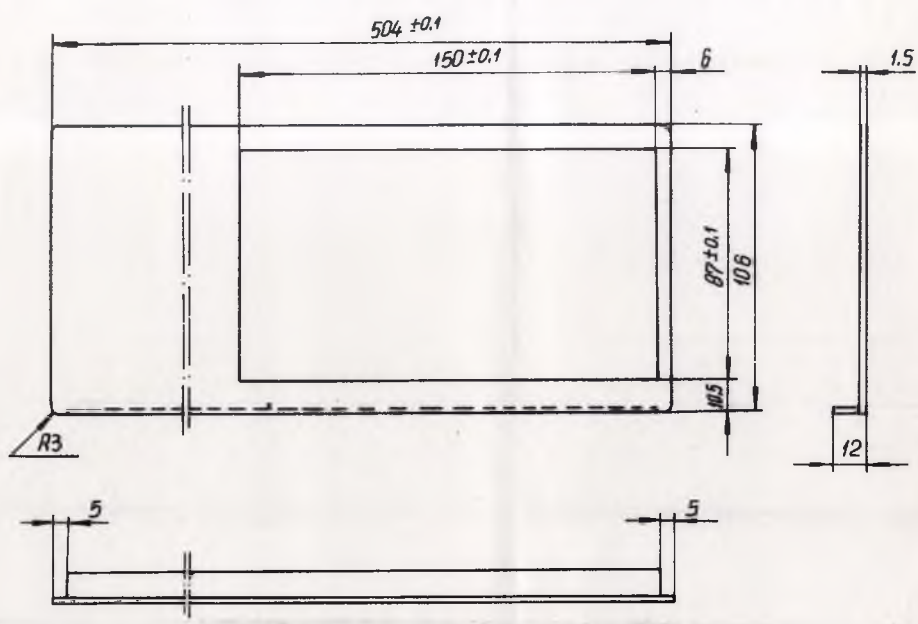
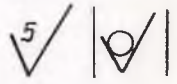
KARAT-2

B-B

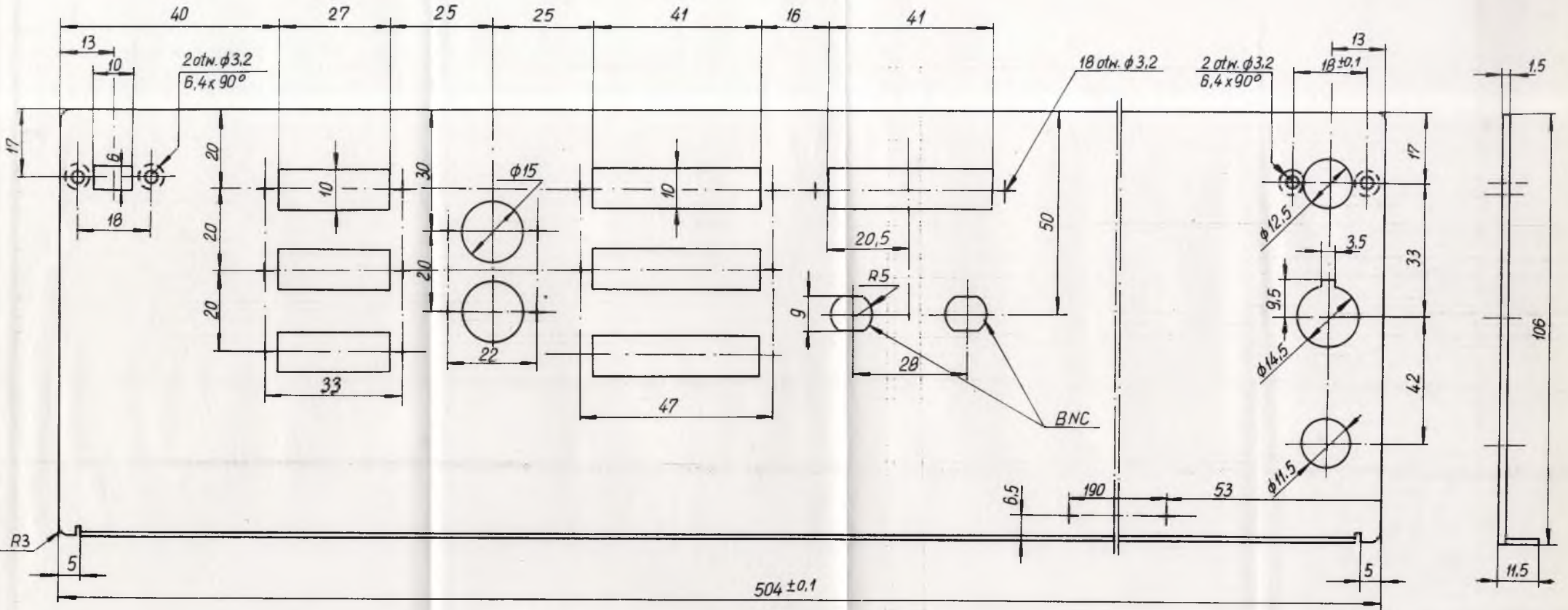




B-ML-00-00								
Wchodzi do rys.		Wyk.	Nr zm.	Il. zm.	Tresc zmiany lub nr karty zmian		Podpis	Data
Podz. 2:1	Kreslil	Podpis	Data	Material: Blacha Z-II-T 1.0 PN81/H-92121				
	Konstr.			Powloka ochronna: OxFe PN-73/H-04652				
	Sprawdz.			Zastepuje rys. nr				
	Zatwierdz.			Zastapiony przez rys. nr				
ELKOR	Nazwa				Nr rysunku			
	Płytką				D-ML-00-16			
				Arkuszy:		Arkusz:		



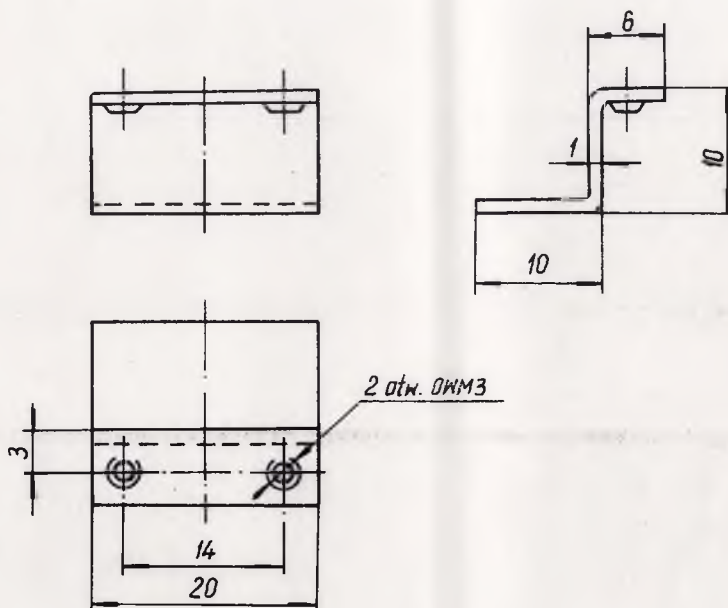
C-02M-01.00								
Wchodzi do rys.		Wyk.	Nr zm.	Il. zm	Tresc zmiany lub nr karty zmian		Podpis	Data
Podz. 1:2	Kreslil	Podpis	Data	Material: Blacha Z-II-1 1,5 PN-81/H-92121				
	Konstr.			Powloka ochronna:				
	Sprawdz.			Zastepuje rys. nr				
	Zatwierdz.			Zastapiony przez rys. nr				
ELKOR	Nazwa Płyta przednia				Nr rysunku D-ML-00-11			
				Ankieta: 25		Ankieta: 1		



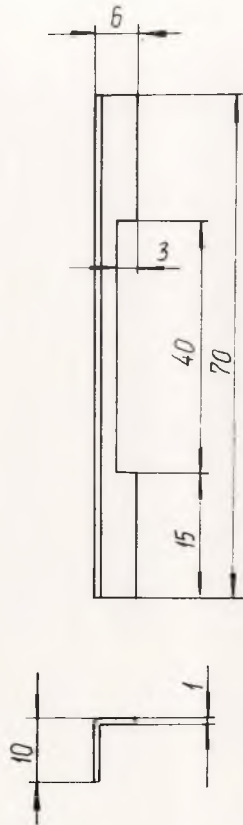
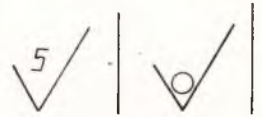
Uwaga!

W wyk. 1 nie wykonywać 2 otw. z gniazdami BNC

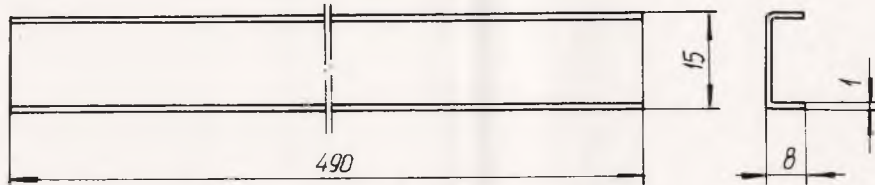
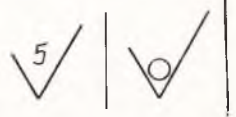
C-02M-01.00							
Nchadzi do rys		Wyk	Nr zm.	Il. zmian	Treść zmiany lub nr karty zmian		Podpis
Podziatka	Kreslit	Podpis	Data	Materiał Blacha Z-II-T#15			Ciepzar
1:1	Konstruował			Powłoka ochronna Emalia czarna (baranek)			
	Sprawił			Zastępuje rys nr			
	Zatwierdził			Zastąpiłony przez rys			
ELKOR	Nazwa Płyta tylna						Nr rys. C-02M-01.02
						Arkuszy: 26	Arkusze: 5



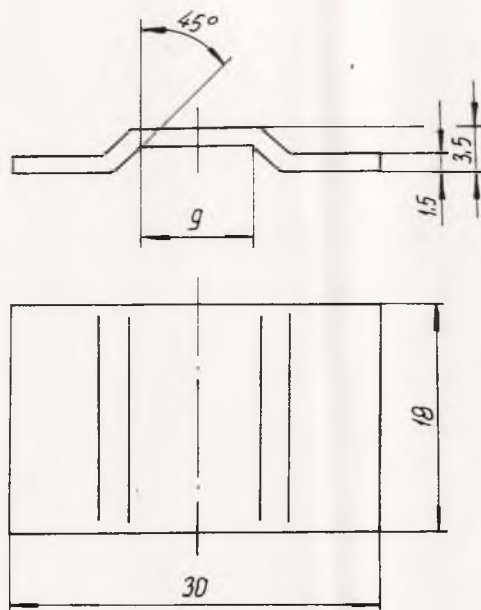
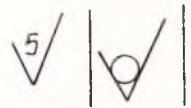
C-02M-01.00						
Wchodzi do rys.		Wyk.	Nr zm.	Il. zm	Treść zmiany lub nr karty zmian	Podpis
2:1	Kreslil	Podpis		Data	Material: Blacha Z-II-T#1,0	
	Konstr.				Powłoka ochronna:	
	Sprawdz.				Zastępuje rys. nr	
	Zatwierdz.				Zastąpiony przez rys. nr	
ELKOR	Nazwa					Nr rysunku
	Wspornik					D-02M-01.03
					Arkuszy: 26	Arkusz: 6



C-02M-01.00								
Wchodzi do rys.		Wyk.	Nr zm.	Il.zm	Tresc zmiany lub nr karty zmian		Podpis	Data
Podz. 1:1	Kreslil	Podpis	Data	Material: Blacha Z-II-T 1.0 PN-81/H-92121				
	Konstr.			Powloka ochronna:				
	Sprawdz.			Zastepuje rys. nr				
	Zatwierdz.			Zastapiony przez rys. nr				
ELKOR	Nazwa					Nr rysunku		
	Kątownik					D-ML-00-15		
					Arkusz: 26		Arkusz: 7	

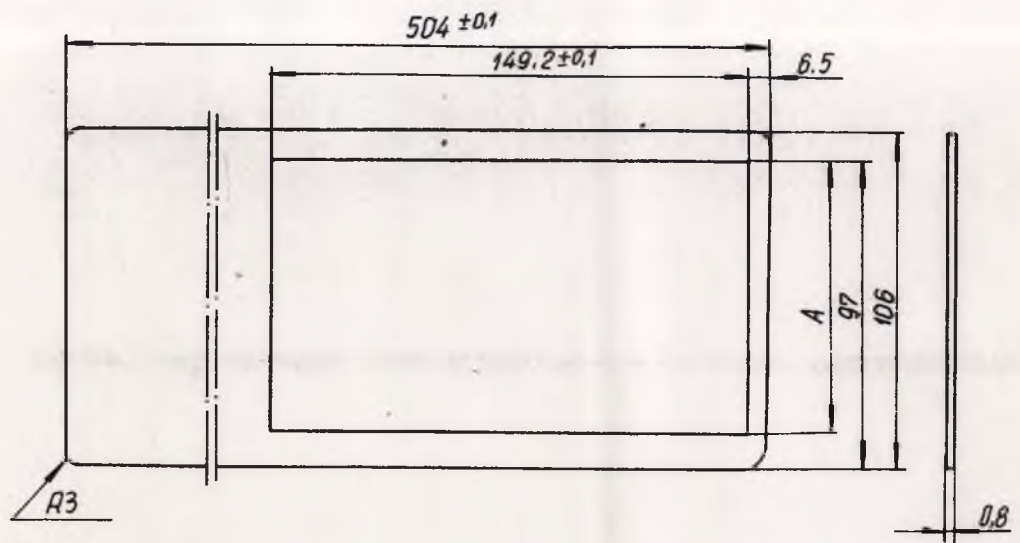


C-02M-02.00								
Wchodzi do rys.		Wyk.	Nr zm.	Il. zm	Tresc zmiany lub nr karty zmian		Podpis	Data
1:1	Kreslil	Podpis		Data	Material: Blacha Z-II-T 1.0 PN-81/H-92121			
	Konstr.				Powloka ochronna:			
	Sprawdz.				Zastepuje rys. nr			
	Zatwierdz.				Zastapiony przez rys. nr			
ELKOR	Nazwa				Nr rysunku			
	Ceownik				D-ML-00-13			



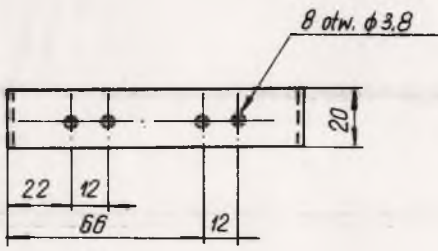
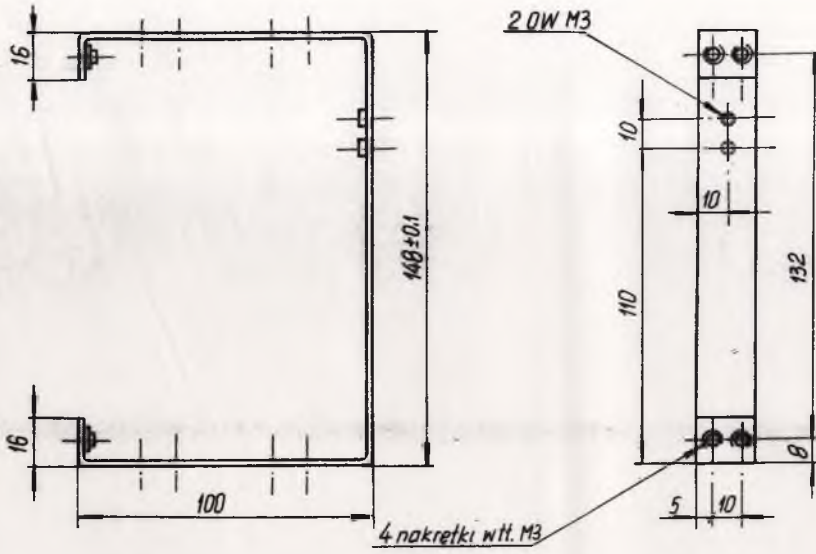
C-02M-02.00								
Wchodzi do rys.		Wyk.	Nr zm.	Il. zm.	Tresc zmiany lub nr karty zmian		Podpis	Data
Podz. 2:1	Kreslil	Podpis		Data	Material: Blacha Z-II-T 1,5 PN-81/H-92121			
	Konstr.				Powloka ochronna:			
	Sprawdz.				Zastepuje rys. nr			
	Zatwierdz.				Zastapiony przez rys. nr			
ELKOR	Nazwa Uchwyt						Nr rysunku D-ML-00-14	
							Arkusz: 26	Arkusz: 10

1,25



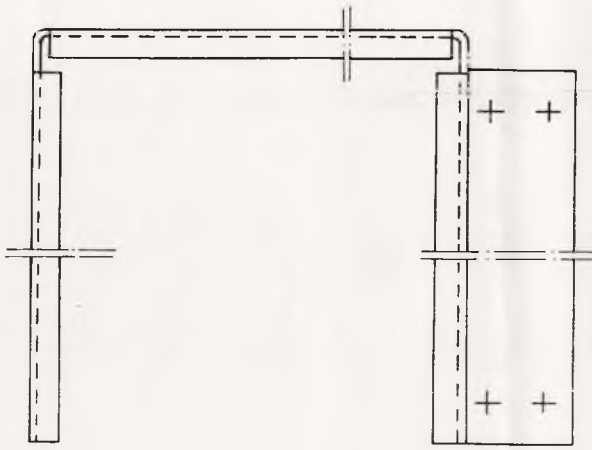
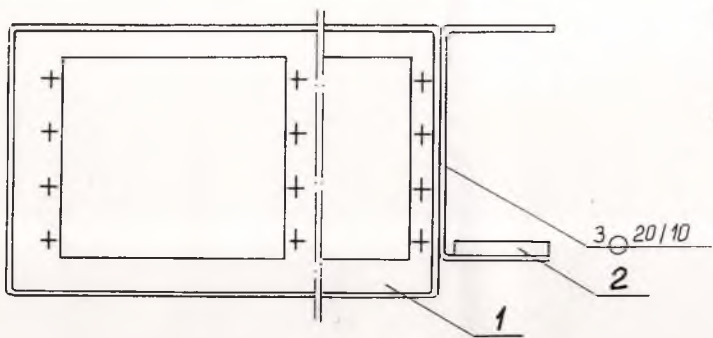
Wyk./num	A
1	42,5
2	86,3

B-02M-00.00								
Wchodzi do rys.		Wyk.	Nr zm.	Il.zm	Tresc zmiany lub nr karty zmian		Podpis	Data
Podz. 1:2	Kreslil	Podpis	Data	Material: Blacha PA4 = 08				
	Konstr.			Powloka ochronna: Al / An 10u PN-80/H-97023				
	Sprawdz.			Zastepuje rys. nr				
	Zatwierdz.			Zastapiony przez rys. nr				
ELKOR	Nazwa				Nr rysunku			
	Płyta czokowa				D-02M-03.00			
				Arkuszy: 26		Arkusz: 11		



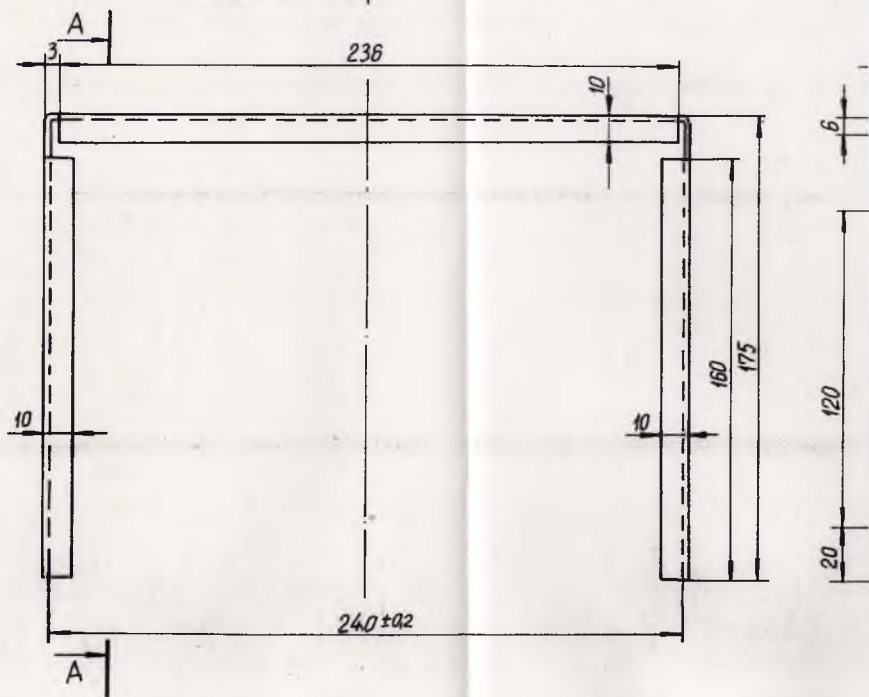
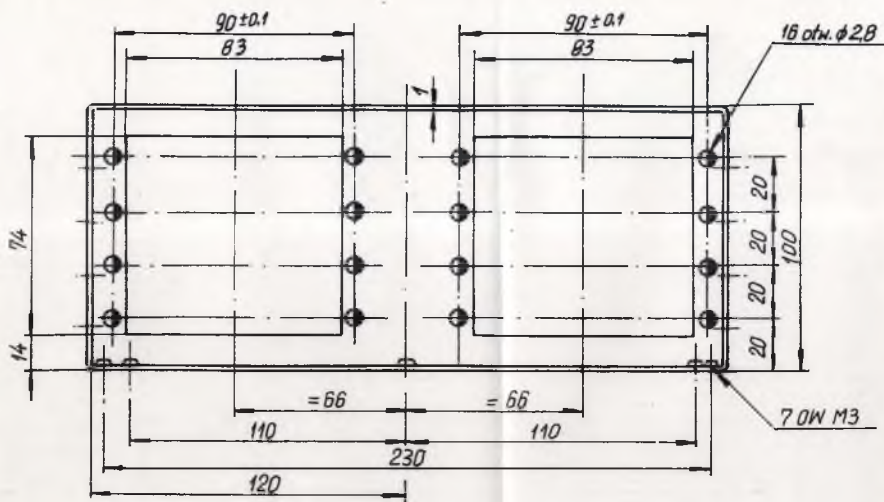
B-02M-00.00							
Wchodzi do rys.		Wyk.	Nr zm.	Il.zm	Treść zmiany lub nr karty zmian	Podpis	Data
1:2	Kreslil	Podpis	Data	Materiał: Blacha Z-II-T# 1,5			
	Konstr.			Powłoka ochronna: Farba podkładowa c.szara			
	Sprawdz.			Zastępuje rys. nr			
	Zatwierdz.			Zastąpiony przez rys. nr			
ELKOR	Nazwa					Nr rysunku	
	Wspornik					D-02M-04.00	
					Arkuszy: 26	Arkusz: 12	

2.5

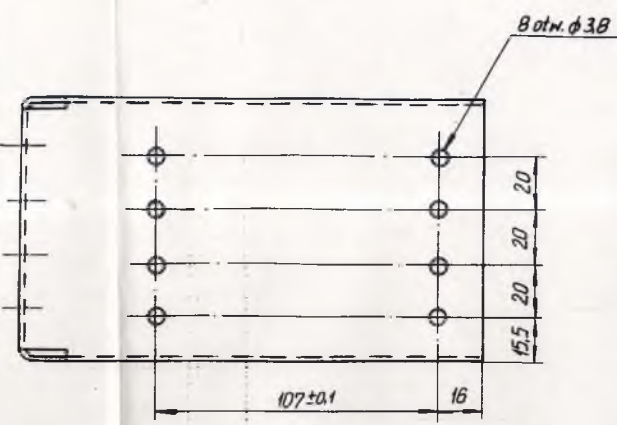


Uwaga!
 Detal malować emalią
 ftalową karbomidową
 czarną 3461-364-990

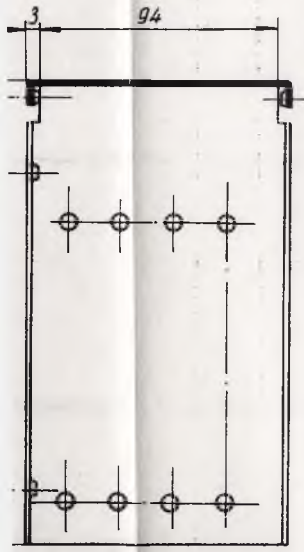
			1	2	Łeownik		D-02M-05.02		
			1	1	Wspornik		C-02M-05.01		
4	3	2	1	Poz.	Nazwa przedmiotu i oznacz. materiału		Nr rys. lub normy	Uwagi	
			B-02M-00.00	Wchodzi do rys	Zastępuje rys nr				
				Podz	Zastąpiony przez rys. nr				
				1:2	Kreślił				
					Konstruował				
					Sprawdził				
				Zatwierdził	Podpis	Data	Nr zmiany	Il. zmian	
							Treść zm. lub karty zm.	Podpis	Data
ELKOR	NOZWA						Nr rys.		
	Wspornik kompl.						D-02M-05.00		
							Arkuszy: 26		Arkusz: 13



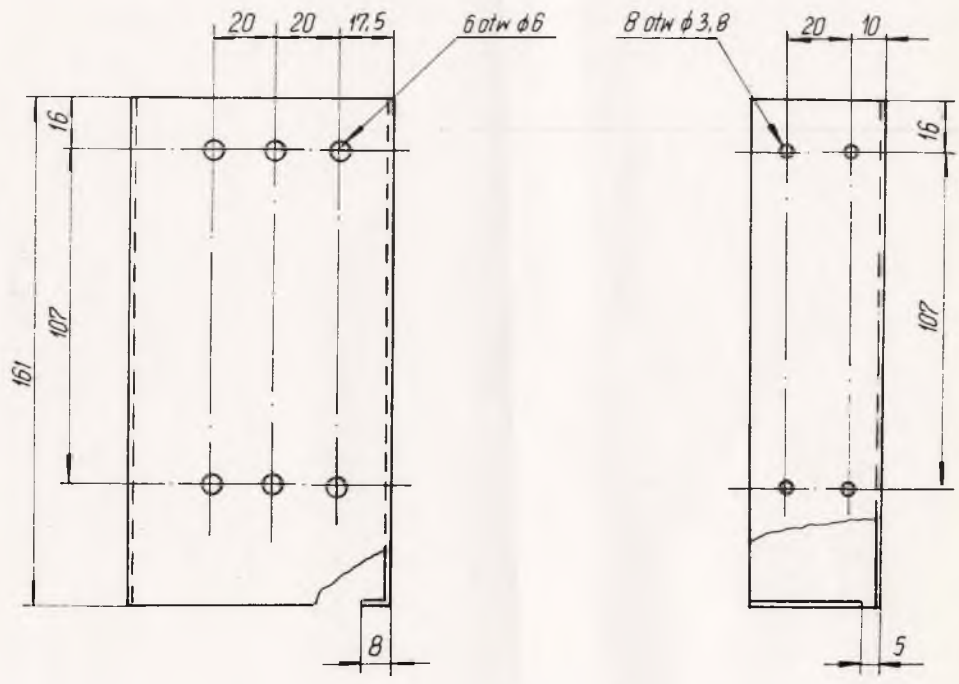
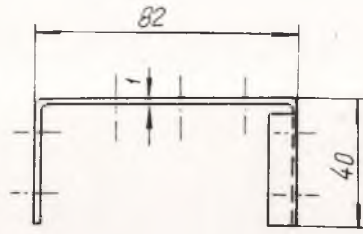
2.5



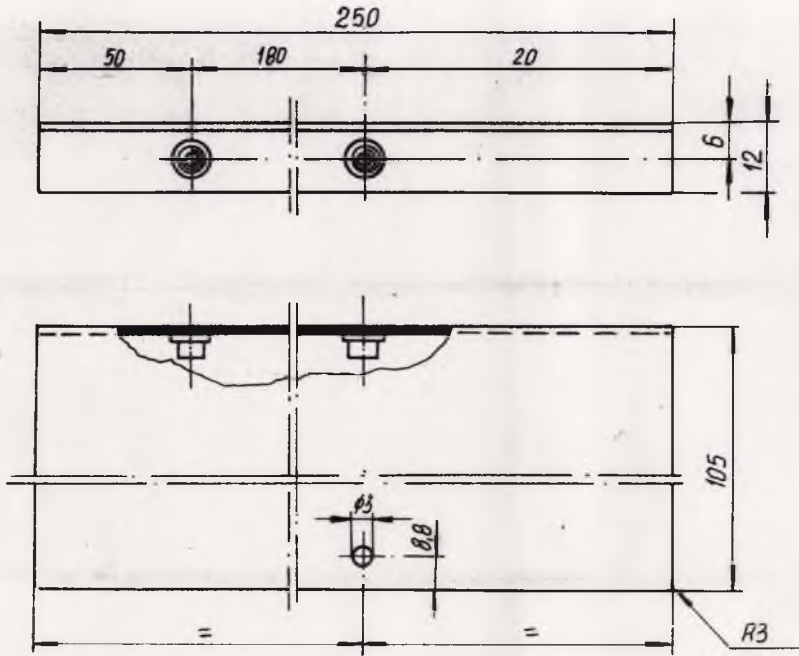
A-A



D-02M-05.00						
Należy do rysunku		Wyk.	Nr zmiany	Uz. zmian	Treść zmiany lub nr karty zmian	Podpis
Podziałka	Kreślił	Podpis	Data	Materiał Blacha Z-II-T 1.0		Data
1:2	Konstruował			Powłoka ochronna		
	Sprawdził			Zastępuje rys nr		
	Zatwierdził			Zastąpiony przez rys nr		
ELKOR	NAZWA Wspornik				Nr rys. C-02M-05.01	
				Arkusz: 26	Arkusz: 14	

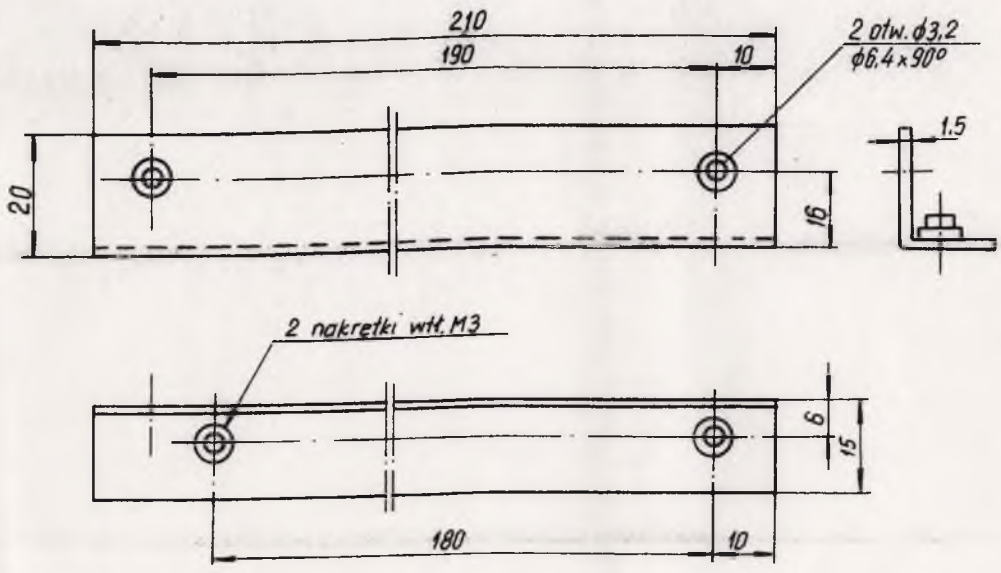


D-02M-05.00						
Wchodzi do rys.		Wyk.	Nr zm.	Il.zm	Treść zmiany lub nr karty zmian	Podpis
Podz.	Kreslil	Podpis	Data	Material: Blacha Z-II-T ≠ 1		
1:2	Konstr.			Powłoka ochronna:		
	Sprawdz.			Zastępuje rys. nr		
	Zatwierdz.			Zastąpiony przez rys. nr		
ELKOR	Nazwa				Nr rysunku	
	Ceownik				D-02M-05.02	
				Arkuszy: 26		Arkusz: 15



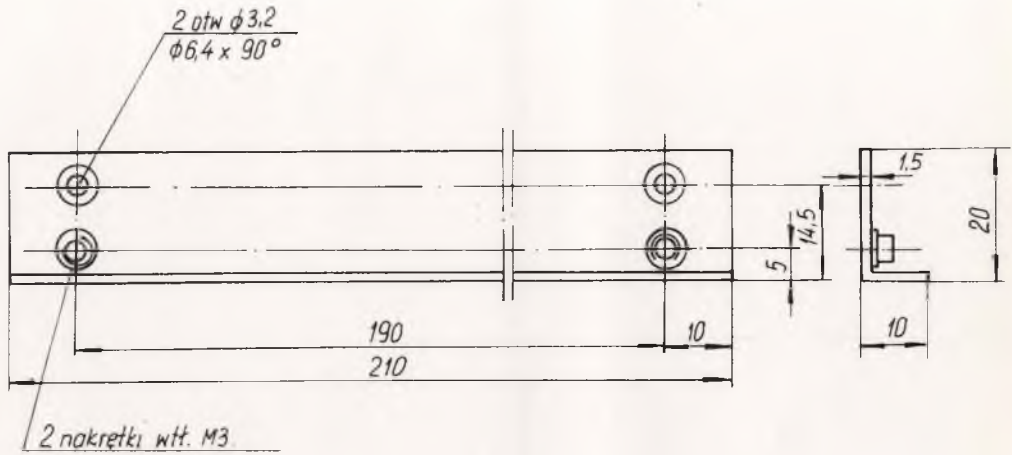
B-02M-00.00								
Wchodzi do rys.		Wyk.	Nr zm.	Il.zm	Tresc zmiany lub nr karty zmian		Podpis	Data
Podz. 1:1	Kreslil	Podpis		Data	Material: Blacha Z-IT 1,5 PN-81/H-92121			
	Konstr.				Powloka ochronna: emalia stal. karbon 3164-364-990 czarna			
	Sprawdz.				Zastepuje rys. nr			
	Zatwierdz.				Zastapiony przez rys. nr			
ELKOR	Nazwa				Nr rysunku			
	Kątownik				D-ML-00-22			
				Arkuszy: 22		Arkusz: 16		

2,5



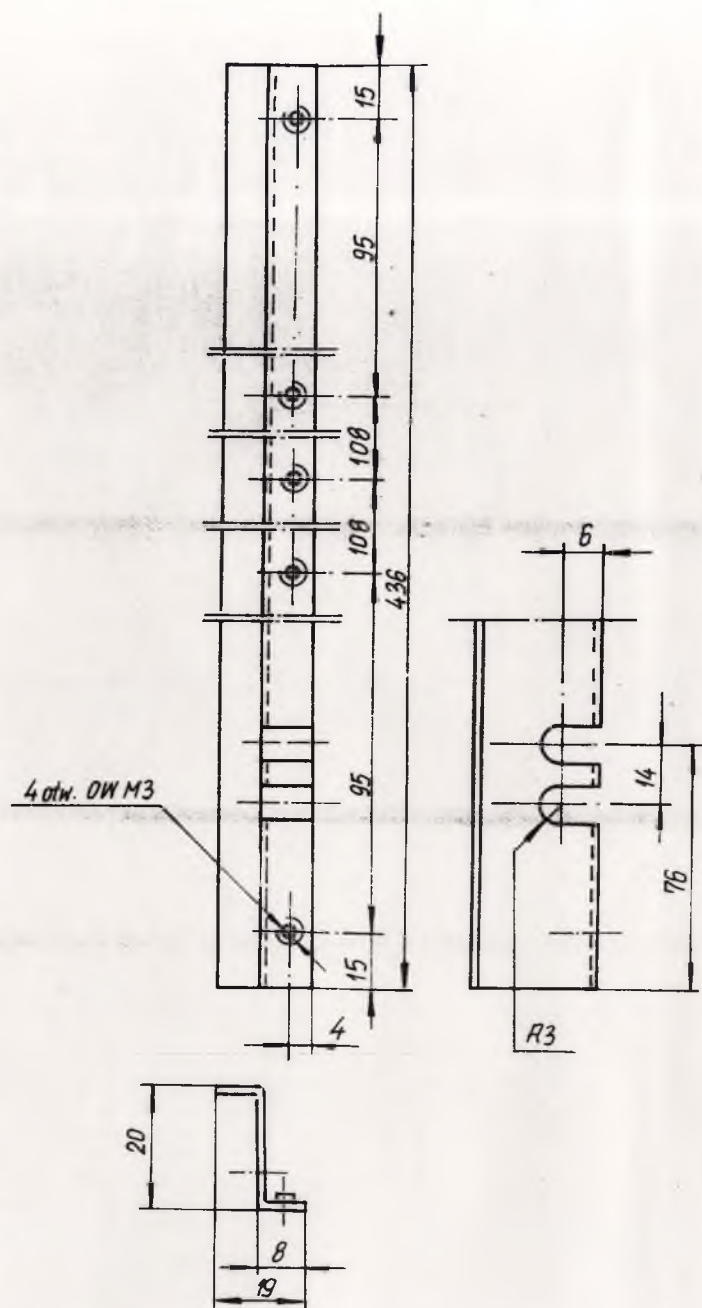
B-02M-00.00								
Wchodzi do rys.		Myk.	Nr zm.	Il. zm.	Tresc zmiany lub nr karty zmian		Podpis	Data
Podz. 1:1	Kreslil	Podpis		Data	Material: Blacha Z-II-T # 1.5			
	Konstr.				Powloka ochronna: Farba podkladowa czarna			
	Sprawdz.				Zastepuje rys. nr			
	Zatwierdz.				Zastapiony przez rys. nr			
ELKOR	Nazwa				Nr rysunku			
	Kątownik				D-02M-06.00			
						Arkuszy: 26	Arkuszy: 17	

2.5

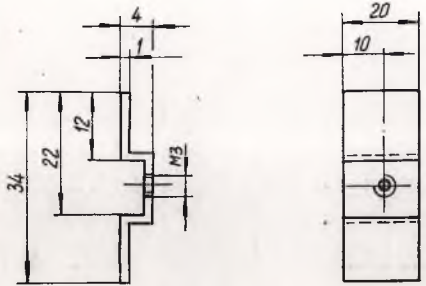
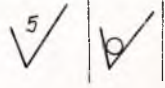


B-02M-00.00								
Wchodzi do rys.		Wyk.	Nr zm.	Il. zm	Tresc zmiany lub nr karty zmian		Podpis	Data
Podz.	Kreslil	Podpis	Data	Material: Blacha Z-II-T#1,5				
1:1	Konstr.			Powloka ochronna: Farba podkładowa c.szara				
	Sprawdz.			Zastępuje rys. nr				
	Zatwierdz.			Zastąpiony przez rys. nr				
ELKOR	Nazwa				Nr rysunku			
	Kątownik				D-02M-07.00			
					Arkusz: 26 Arkusz: 18			

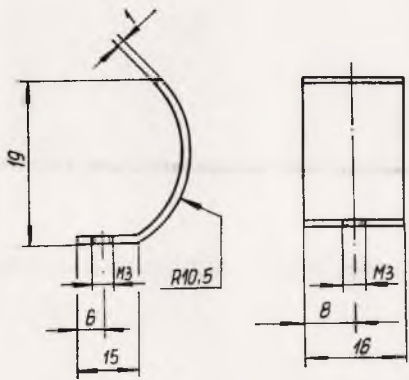
5 ✓



C-ML-01-01		3					
Wchodzi do rys.		Wyk.	Nr zm.	Il. zm	Treść zmiany lub nr karty zmian	Podpis	Data
Podz.	Kreslil	Podpis	Data	Material: Blacha Z-II-T 1.0 PN-81/H-92121			
1:1	Konstr.			Powłoka ochronna:			
	Sprawdz.			Zastępuje rys. nr			
	Zatwierdz.			Zastąpiony przez rys. nr			
ELKOR	Nazwa					Nr rysunku	
	Zetownik					D-ML-01-07	
						Arkuszy: 26	Arkusz: 21



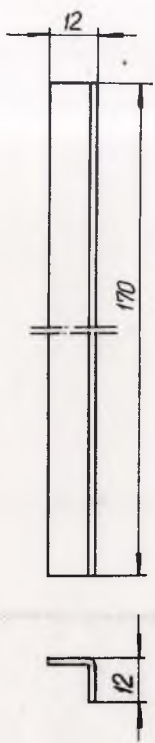
U



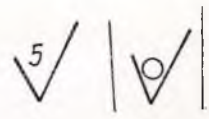
C

4	3	2	1	Podr.	Nazwa przedmiotu i ozn. materiału	Nr rys. lub normy	Uwagi
		C-ML-01-01	B-ML-01-00	Wprowadzi do rys.	Zastępuje rys. nr.		
				Podr.	Zastąpiony przez rys. nr.		
				2:1	Kreślił		
					Konstruował		
					Sprawił		
					Zatwierdził		
					Prof. pisa		
					Zrobił		
					Nr zmiany / L. zmian	Treść zmiany lub nr karty zm.	Podpis Data
ELKOR		Nazwa Element U, C				Nr rys. D-ML-01-09	
		Arkuszy: 26		Arkusz: 22			

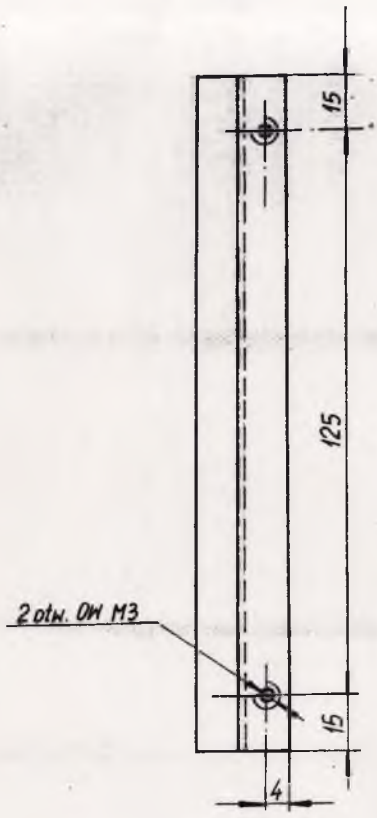
5 ✓ | ✓



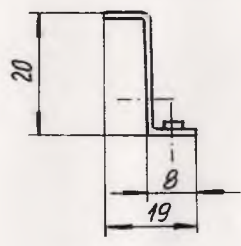
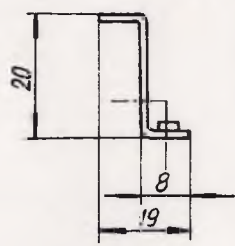
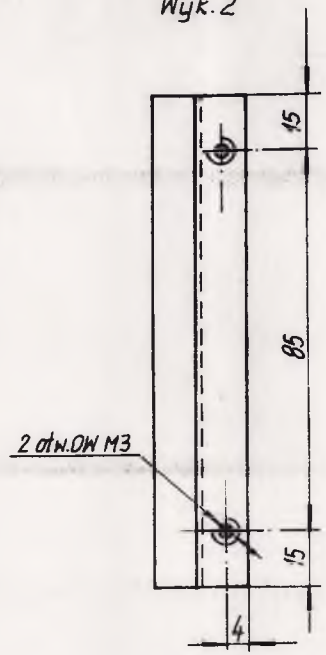
4	3	2	1	Poz	Nazwa przedmiotu i oznaczenie materiału	Nr rys. lub zmiany	Uwagi
		C-ML-01-01	C-ML-01-02	Wchodzi do nr 1	Zastępuje rys. nr.		
				Podz. 1:1	Zastęp przez rys. nr.		
				Kreślił			
				Konstruował			
				Sprawił			
				Zatwierdził			
				Podpis			
				Data			
					Nr zmiany / R. zmian	Treść zmiany lub nr karty zm.	Podpis Data
ELKOR		NAZWA Kątownik				Nr rys. D-ML-01-08	
						Arkuszy: 26 Arkusz: 23	



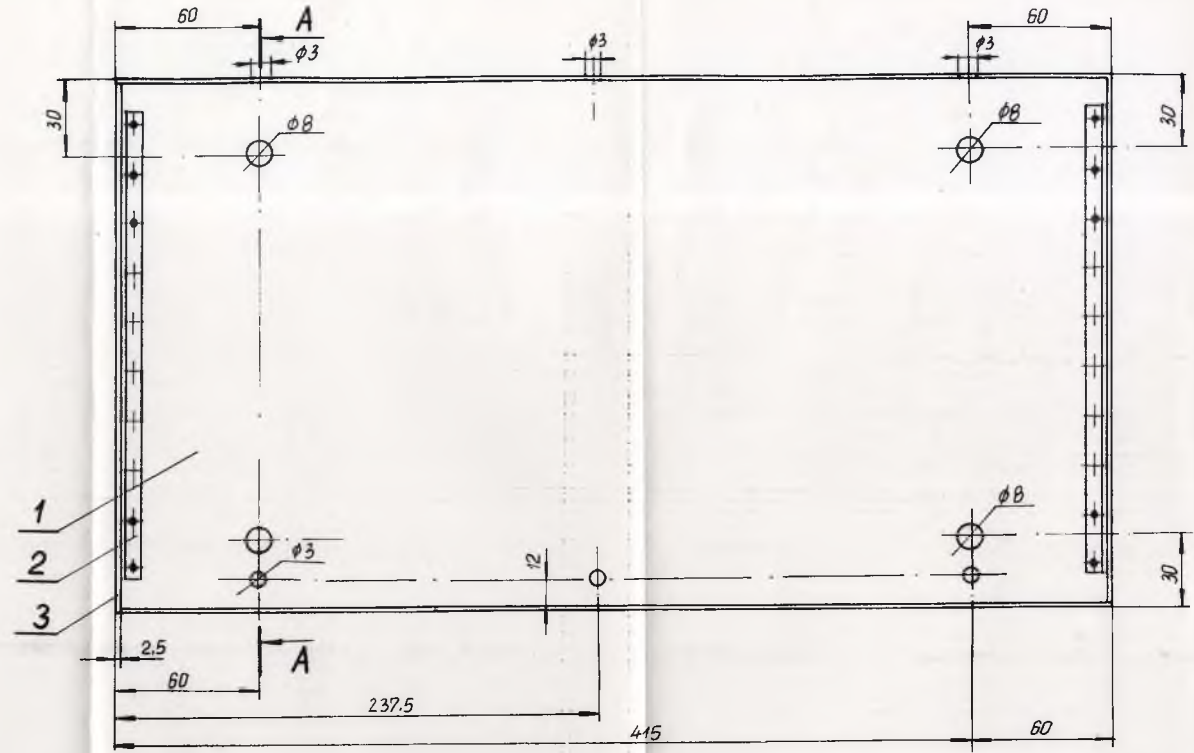
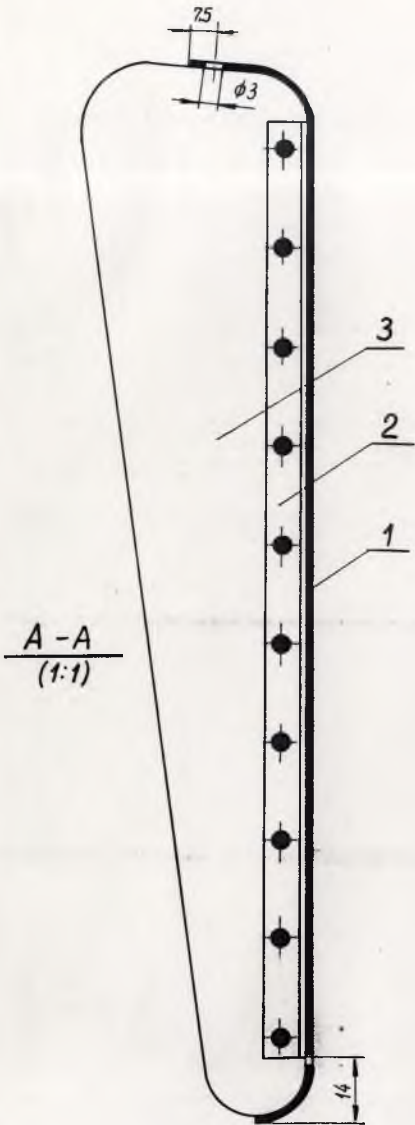
Wyk. 1



Wyk. 2

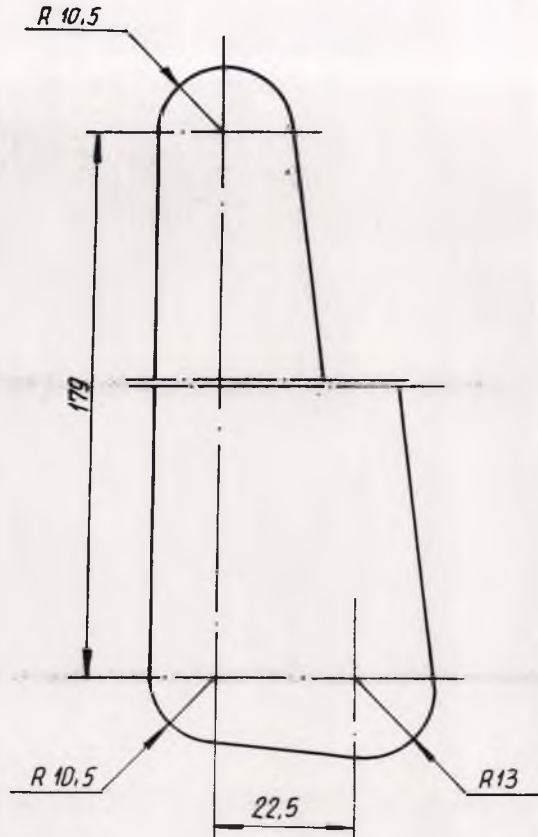
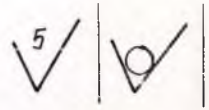


C-ML-01-01							
Wchodzi do rys.		Wyk.	Nr zm.	Il. zm	Treść zmiany lub nr karty zmian	Podpis	Data
Podz. 2:1	Kreslil	Podpis	Data	Material: Płyta Tc Fe 3.0 PN-73/E-29080			
	Konstr.			Powłoka ochronna:			
	Sprawdz.			Zastępuje rys. nr			
	atwierdź			Zastąpiony przez rys. nr			
ELKOR	Nazwa					Nr rysunku	
	Zetownik					D-ML-01-10	
						Arkuszy: 26	Arkusz: 24



Uwaga! Detal malować emalią ftalową korbomidową 3461-364-990 czarna.

		2	3	Ścianka boczna	D-ML-01-05	
		2	2	Kątownik	D-ML-01-08	
		1	1	Pokrywa dolna		
4	3	2	1	Poz. Nazwa przedmiotu i ozn. materiału	Nr rys. lub normy	Uwagi
				Zastęp rys. nr.		
				Zastęp przez rys. nr.		
				Kreślił		
				Konstruował		
				Podzi 1:2 Sprawdził		
				Zatwierdził		
				Nr zmiany	Il. zmian	Trasć zmiany lub nr karty zm.
				Podpis		Data
ELKOR				Nazwa	Pokrywa dolna kompl.	
				Nr rys.	C-ML-01-02	
				Arkusz: 26	Arkusz: 25	

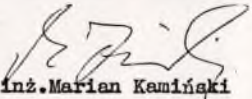


C-ML-01-02						
Wchodzi do rys.		Wyk.	Nr zm.	Il. zm	Treść zmiany lub nr karty zmian	Podpis
1:1	Kreslil	Podpis		Data	Material: Blacha Z-II-T 1.0 PN-81/H-92121	
	Konstr.				Powłoka ochronna:	
	Sprawdz.				Zastępuje rys. nr	
	Zatwierdz.				Zastąpiony przez rys. nr	
ELKOR	Nazwa				Nr rysunku	
	Ścianka boczna				D-ML-01-05	
				Arkuszy: 26		Arkusz: 26

PROTOKOŁ ZAKOŃCZENIA
realizacji umowy nr 892070

Ze względu na "zestarzenie się" konstrukcji mikrokomputera KARAT, na którym oparta została część sprzętowa systemu rejestracji i przetwarzania danych CHOMIK oraz przy braku kolejnych zamówień, strony postanawiają zakończyć realizację w/w umowy.wdrożeniowej z dniem 91-05-30 po wykonaniu /przy wzajemnej współpracy/ 1 szt. systemu w ramach przygotowań serii informacyjnej oraz wpłacić na rzecz Politechniki Gdańskiej sumę umowną w wysokości 3.800.000,-zł za przekazaną dokumentację zużyte materiały oraz pomoc merytoryczną w dotychczasowych pracach.

Przedstawiciel Wykonawcy:


dr inż. Marian Kamiński

Przedstawiciel Wdrażającego:

mgr inż. Marek Antosz

Z-ca PRZEDSA
mgr inż. Marek Antosz