

**PRZEMYSŁOWY INSTYTUT AUTOMATYKI I POMIARÓW  
MERA-PIAP**

**Al. Jerozolimskie 202 02-222 Warszawa Telefon 23-70-81**

**Ośrodek Automatyki Elektrycznej**

**Zespół Budowy Robotów i Serwomechanizmów**

**Główny wykonawca**

dr inż. Marian Wrzesień

**Wykonawcy**

mgr inż. Zbigniew Stańczak

**Konsultant**

**Nr zlecenia** RP-58.2  
Zad. 32.

Oprogramowanie użytkowe dla pakietów  
P1, P2 panelu programowania oraz MW-  
32. Weryfikacja oprogramowania użyt-  
kowego podczas testowania serii prób-  
nej układów sterowania IRp-6/60.  
Wprowadzenie programów użytkowych  
testera do pamięci EPROM. Zweryfiko-  
wana dokumentacja dla wykonan jedno-  
stkowych urządzenia. Dokumentacja  
techniczno-ruchowa.  
1. Dokumentacja techniczno-ruchowa.

**Zleceniodawca** CPBR 7.1 "Roboty przemysłowe"

**Pracę rozpoczęto dnia**

01.12.1988

**zakończono dnia**

30.03.1989

**Kierownik Zespołu**

**Kierownik Ośrodka**

dr inż. P. Jabłoński

Z-ca Dyr. d/s Automatyki

dr inż. R. Kętrzymowicz

doc. dr inż. T. Gałązka

**Praca zawiera:**

**Rozdzielnik - ilość egz:**

stron 19

**Egz. 1**

BOINTE

rysunków

**Egz. 2**

PPDiM

fotografii

**Egz. 3**

OAE

tabel

**Egz. 4**

OAE

tablic

**Egz. 5**

ZAP

załączników 2

**Egz. 6**

**Nr rejestr.** 6260

## Analiza deskryptorowa

URZADZENIA AUTOMATYCZNEJ REGULACJI  
I STEROWANIA: ROBOTY PRZEMYSŁOWE, TESTO-  
WANIE.

## Analiza dokumentacyjna

Sprawozdanie zawiera dokumentację te-  
chniczno-ruchową douruchomienia i te-  
stowania pakietów.

## Tytuły poprzednich sprawozdań

Zlecenie UR-01.03.01 K:

Etap 1: Studia wstępne oraz założenia techni-  
czno-ekonomiczne (spr. nr rej. 5494).

Etap 2: Opracowanie i uruchomienie systemu  
programowania testera (spr. nr rej.  
5549).

Zlecenie RP-58.2

Zadanie 1.1: Projekt wstępny testera (spr. nr  
rej. 5627).

Zadanie 1.2: Wykonanie dokumentacji szkicowej  
testera (spr. nr rej. 5711).

Zadanie 1.3: Wykonanie, uruchomienie i przeba-  
danie modelu użytkowego testera  
(spr. nr rej. 5911).

Zadanie 3.1: Opracowanie i wykonanie urządzeń  
dla uruchomienia i testowania  
układów sterowania IRp-6/60.  
(spr. nr rej. 6189).

UKD

338,45: 62/69]. 002.1/2  
681.3.06

Roboty przemysłowe  
Oprogramowanie

PROJEKT DOKUMENTACJI TECHNICZNO - RUCHOWEJ  
URZADZENIA DO URUCHAMIANIA I TESTOWANIA PAKIETOW  
UKŁADU STEROWANIA ROBOTOW PRZEMYSŁOWYCH IRp6/6

SPIS TRESCI

1. Przeznaczenie.
2. Opis konstrukcji.
3. Budowa i działanie.
  - 3.1. Stanowisko urządzenia testującego.
  - 3.2. Pakiet PD złącz obiektowych.
  - 3.3. Budowa i działanie pakietu interfejsu IT1.
    - 3.3.1. Przeznaczenie i opis ogólny pakietu IT1.
    - 3.3.2. Podzespół dekodera adresów.
    - 3.3.3. Układ sterujący dla bramki danych D4.
    - 3.3.4. Układ bramki danych.
    - 3.3.5. Układ bram wejściowych.
    - 3.3.6. Układ bram wyjściowych.
    - 3.3.7. Układ przerwan.
    - 3.3.8. Układ dwukierunkowych bram WE/WV.
  - 3.4. Budowa i działanie pakietu interfejsu IT2.
    - 3.4.1. Przeznaczenie i opis ogólny pakietu IT2.
    - 3.4.2. Podzespół dekodera adresów.
    - 3.4.3. Podzespół sterujący testowaniem pakietów MW-31 i MW-32.
    - 3.4.4. Podzespół sterujący testowaniem pakietu MZ-70.
    - 3.4.5. Podzespół sterujący testowaniem płytki wyświetlaczy panelu programowania.
    - 3.4.6. Podzespół sterujący testowaniem detektorów przeciążenia pakietu MC-42.
    - 3.4.7. Podzespół sterujący silnikiem wykonawczym.
  - 3.5. Budowa i działanie pakietu interfejsu IT3.
    - 3.5.1. Przeznaczenie i opis ogólny pakietu IT3.
4. Uruchamianie.
5. Opis instrukcji testowania pakietów układu sterowania robotów.
  - 5.1. Menu główne.
  - 5.2. Programy narzędziowe.
  - 5.4. Testowanie pakietu MW-31.
  - 5.5. Testowanie pakietu MW-32.
  - 5.6. Testowanie pakietu MZ-70.
  - 5.7. Testowanie pakietu ML-50.
  - 5.8. Testowanie pakietu MI-50.
  - 5.9. Testowanie pakietu MA-70.
  - 5.10. Testowanie panelu programowania.
6. Załączniki.
  1. Program Monitor MM-86.
  2. Instrukcje programów uruchomieniowych pakietów MI-50 i ML-50.

## 1. PRZEZNACZENIE

---

Urządzenie do uruchamiania i testowania (UTP) jest przeznaczone do testowania i uruchamiania następujących pakietów układu sterowania robotów przemysłowych IRp-6/60 (USRP):

- pakiet 16WE/16WY MC-42,
- pakiet kontroli MW-31,
- pakiet kontroli MW-32,
- pakiet zasilania resolwerów MZ-70,
- pakiet sterownika położenia osi MA-70,
- panel programowania,
- pakiet sprzężenia z PK-1 MI-50,
- pakiet pamięci ML-50.

Istotą rozwiązania konstrukcyjnego jest przyjęcie metody testowania funkcjonalnego, umożliwiającego testowanie pakietów od strony złącz systemowych (dolaczonych do złącz magistrali systemowej) jak i obiektowych (dolaczanych do urządzeń zewnętrznych współpracujących z systemem mikroprocesorowym układu sterowania). Wynikło stąd wymaganie podawania wszystkich sygnałów pakietu testowanego w sekwencjach odpowiadających pracy tego pakietu w pełnym zestawie USRP.

Spełnienie powyższego warunku osiągnięto poprzez:

- » zbudowanie interfejsu łączącego pakiet testowany (PT) ze złączami systemowymi UTP,
- » zbudowanie pakietu złącz obiektowych, umożliwiających dołączenie UTP do PT poprzez złącza obiektowe.

Ponadto wykorzystano niektóre pakiety wchodzące w skład USRP, które mogą pełnić rolę interfejsów dla złącz obiektowych testowanych pakietów.

## 2. OPIS KONSTRUKCJI

---

1. Urządzenie UTP jest zbudowane na bazie systemu mikroprocesorowego.
2. Elektroniczna część testera jest oparta na systemie mikroprocesorowym INTEL DIGIT-PROWAY, co zapewnia kompatybilność UTP z USRP, który bazuje na tym samym systemie mikroprocesorowym.
3. Dołączenie pakietu testowanego do urządzenia testującego umożliwia interfejsy:
  - dla złącz systemowych PT oraz
  - złącz obiektowych PT.
4. Zapewnienie komunikacji pomiędzy operatorem, a UTP osiągnięto poprzez zastosowanie monitora operatorskiego wraz z klawiaturą.
5. Konstrukcja stanowiska UTP umożliwia łatwą zmianę jego miejsca.

Z przyjętego rozwiązania konstrukcyjnego wynika następująca konfiguracja UTP:

- » Urządzenia standardowe systemu INTEL DIGIT-PROWAY:
  - kaseta MF-21,
  - plater MF-31,
  - pakiet 16-bitowej jednostki centralnej MM-86,
  - 4 pakiety rozszerzenia pamięci ML-50.
- » Urządzenia układu sterowania robotów IRp-6/60:
  - pakiet kontroli MW-31,
  - pakiet 16 WE/WY MC-42,
  - pakiet zasilania resolwerów MZ-70,
  - sterownik mocy układu napędowego TAR-1A,
  - silnik wykonawczy prądu stałego typu PZTK 88-35 TRR.

- » Stanowisko UTP (konstrukcja nosna) wraz z wymaganymi zasilaczami oraz okablowaniem wewnetrznym.
- » Pakiety opracowane wyłacznie dla UTP:
  - pakiet P0 zlacz obiektowych urzadzenia umozliwiajacy polaczenie odpowiednich zlacz obiektowych pakietow testowanych z UTP,
  - interfejs rownolegly IT1 (dla zlacz systemowych),
  - interfejs zlacz obiektowych IT2 (dla pakietow testowanych MZ-70, MW-31, MW-32 oraz plytki wyswietlaczy panelu programowania).
  - interfejs zlacz obiektowych IT3 (dla pakietow testowanych MA-70 oraz panelu programowania).
- » Zestaw kabli przeznaczonych do polaczenia odpowiednich zlacz obiektowych PT z pakietem zlacz obiektowych P0 urzadzenia.
- » Monitor ekranowy MERA 7953N z klawiatura MERA 794B.

### 3.OPIS PODZESPOLOW URZADZENIA TESTUJACEGO

---

#### 3.1.Stanowisko urzadzenia testujacego

---

Szkic konstrukcji mechanicznej urzadzenia testujacego zostal przedstawiony na rys.1.Podstawa jest metalowy stol (1), na którym znajduje sie monitor ekranowy z klawiatura (2) oraz mikroprocesorowy uk lad sterowania urzadzenia testujacego (3), do ktorego dolacza sie pakiet testowany (4) umieszczany w prowadnicy (5). Zlacza obiektowe pakietu testowanego dolacza sie za pomoca kabli do odpowiednich zlacz pakietu P0 (z lacz obiektowych) (6) umieszczonego nad kaseta uk ladu sterowania urzadzenia testujacego (3). W dolnej czesci stolu jest umieszczony blok zasilania urzadzenia (7), sterownik TAR-1A (8) oraz silnik wykonawczy PZTK88-35 TRR (9) natomiast pod blatem przeprowadzono kable (10) oraz umieszczono wentylator (11), wymuszajacy obieg powietrza w kasecie uk ladu sterowania. Z prawej strony stanowiska umieszczono dwa przyciski (12) sluzace do zalaczania i wylaczania urzadzenia. W koncowej czesci prowadnicy (5) umieszczono wylacznik (13), ktory uniemozliwia zalaczenie urzadzenia w przypadku braku pakietu poddanego testowaniu.

Obszar pamieci testera (mikroprocesor Intel 8086) :

a/. 0 - 9FFF - RAM  
 b/. A000 - CFFFF - obszar wolny,  
 c/. D0000 - EFFFF - EPROM (testy pakietow),  
 d/. F0000 - FDFFF - obszar wolny,  
 e/. FE000 - FFFFF - EPROM MM86 (program MON86).

ad. a/.

- 0 - 1FFF - RAM MM86,  
 - 2000 - 3FFF - ML50 nr 1,  
 - 4000 - 5FFF - ML50 nr 2,  
 - 6000 - 7FFF - ML50 nr 3,  
 - 8000 - 9FFF - ML50 nr 4,

ad. c/.

- D0000 - D0FFF - MENU GLOWNE,  
 - D1000 - D3FFF - MC42,  
 - D4000 - D8FFF - MW31,  
 - D9000 - DDFFF - MW32,  
 - DE000 - DFFFF - MZ70,  
 - E0000 - E0FFF - ML50,  
 - E1000 - E1FFF - MI50,  
 - E2000 - E7FFF - MA70,  
 - E8000 - EAEFF - PANEL PROGRAMOWANIA,  
 - EC000 - EDFFF - PROGRAM URUCHOMIENIOWY,

### 3.2. Pakiet PO złącz obiektowych urządzenia testującego

Zadaniem pakietu PO ((6)-rys.1) jest przekazywanie lub odbieranie sygnałów występujących na złączach obiektowych poszczególnych pakietów USRP podczas ich testowania lub uruchamiania.

Pakiet PO umożliwia dołączenie do urządzenia testującego następujących złącz obiektowych pakietów:

- » MW-31; złącza c i d,
- » MW-32; złącza e i t,
- » MZ-70; złącza 1 i 2,
- » MC-42; złącza c i d,
- » MA-70; złącza c i d,
- » płytki P3 panelu programowania; złącze Z302,
- » panel programowania; złącza Z101 i Z105.

Schemat połączeń pakietu PO z pakietami urządzenia testującego przedstawiono na rys.2. Jak wynika ze schematu, złącza pakietu PO są dołączane odpowiednio do pakietów UTP: IT2, IT3, MC-42, które pełnią rolę interfejsów dla złącz obiektowych pakietów testowanych.

### 3.3 Budowa i działanie panelu interfejsu IT1

#### 3.3.1. Przeznaczenie i opis ogólny pakietu interfejsu IT1.

Pakiet interfejsu IT1 zapewnia dołączanie pakietów testowanych do testera od strony złącz systemowych oraz ich testowanie w układzie maksymalnie zbliżonym do rzeczywistego układu pracy. Jest podzespołem wspólnym dla wszystkich testowanych pakietów i umożliwia komunikację pomiędzy testerem i testowanymi pakietami z szybkością odpowiadającą szybkości pracy pakietu testowanego.

Moduł interfejsu IT1 zawiera układy umożliwiające obsługę sygnałów wyprowadzonych na złącza A i B testowanych pakietów.

Moduł ten zbudowany jest z dwóch pakietów P1 i P2.

Pakiet P1 zawiera :

- dwa złącza systemowe A i B, przeznaczone do dołączania pakietu do magistrali testera,
- dwa złącza obiektowe AP i BP, przeznaczone do dołączania pakietu testowanego,
- złącze C umieszczone wewnątrz pakietu, przeznaczone do połączenia pakietu P1 z pakietem P2,
- układy przełączników.

Pakiet P1 zapewnia :

- bezpośrednie dołączenie do pakietu testowanego: zasilania, linii danych, adresowych, sterowania, potwierdzenia, sygnałów zegarowych, sygnałów arbitrażu dostępu do magistrali, linii INT-A0/ - INT-A3/ oraz sygnału INTA/,
- opcjonalne odłączenie linii adresowych ADR16/ - ADR19/ pakietu testowanego od magistrali testera i dołączenie powyższych linii do pakietu P2. (Funkcje te spełnia układ przełącznika czterostykowego sterowanego sygnałem STP),
- opcjonalne dołączenie linii przerwan pakietu testowanego do magistrali testera (układ przełącznika osmiopozycyjnego),
- dołączenie do pakietu P2 pozostałych linii wyprowadzonych na złącza systemowe pakietów testowanych.

Pakiet P2 zapewnia obsługę wszystkich sygnałów wyprowadzonych na złącza systemowe pakietów testowanych i nie łączonych bezpośrednio z magistralą testera. Sygnały te są dołączone do pakietu P2 poprzez jego złącze obiektowe C.

#### 3.3.2. Podzespół dekodera adresów (pakiet P2).

Dekoder adresów jest zbudowany z układów D2, D3, D5. Pamięci D2 i D3 typu Intel 3601 są zaprogramowane w ten sposób, że przy podanym adresie

odpowiadającym adresowi pakietu ich wyjścia Qi przechodzą w stan niski, a sygnały CSA/ i CSB/ uaktywniają dekodery D5 typu UCY 74LS405, wydający sygnały wyboru układu CS0/ - CS3/.

### 3.3.3. Układ sterujący dla bramki danych D4 (pakiet P2).

Układ ten jest zbudowany z układów: D1 typu 74LS04 oraz C3 i D3 typu Intel 3601. Układ D1 pełni rolę wzmacniacza dla sygnałów IORC/, IOWC/, INIT/ oraz wytwarza sygnały RD/, WR/, WR, RES, RES/ oraz T - służący do sterowania kierunkiem przepływu danych przez obwód D4. Układ C3 (częściowo wykorzystany do sterowania kierunkiem pracy dwukierunkowych układów WE/WY) zapewnia wzmocnienie i negacje sygnałów ADR1/ i ADR2/ tak, że  $A1 = (\text{ADR1/})$ ,  $A2 = (\text{ADR2/})$  oraz wytwarza sygnał STB/. Układ D3 (częściowo wykorzystany także do dekodowania adresu wraz z układem D2), wytwarza sygnały XACK/ (wyprowadzony na magistralę systemową UTP) oraz OE/.

### 3.3.4. Układ bramki danych (pakiet P2).

Układ ten zbudowany jest na obwodzie D4 typu Intel 8287 i jest sterowany sygnałami T i OE/. Bramki danych uaktywniane są wtedy, gdy odbywa się operacja czytania lub pisania do/z układu WE/WY znajdującego się na płycie P2 (sygnał OE/).

### 3.3.5. Układ bram wejściowych (pakiet P2).

Układ bram wejściowych umożliwia czytanie stanu linii wyjściowych rozmieszczonych na złączach systemowych oraz linii przerwan pakietu testowanego. Jest on zbudowany z bramy B3 typu Intel 8255 oraz układów C1 i C2 typu UCA 6474. Porty PB i PC układu B3 przyjmują sygnały PNI0 - PNI3, ERROR/, INPOS/, ROBZSYNCHR/, WRITTEN/, BUDZIK/. Przerzutniki układów C1 i C2 rejestrują narastające zbocza sygnałów SREADY/, XACK/ oraz MMAP2/.

### 3.3.6. Układ bram wyjściowych (pakiet P2).

Układ bram wyjściowych wydaje wewnętrzne sygnały sterujące dla pakietu P2 oraz sygnały na złącza systemowe pakietu testowanego, a ponadto zapewnia transformacje sygnałów adresowych ADR16/ - ADR19/. Składa się on z układów B2 i D6 typu Intel 8212 oraz bramy PA układu B3 typu Intel 8255. Układ B2 generuje:

- sygnał STP/ wykorzystywany do sterowania przekaźnikami P1 i P2 umieszczonymi na pakiecie P1 modułu IT1,
- sygnał ENO/ sterujący bramkami A6 typu SN 54125 blokującymi sygnały CP i SINREF,

Port PA układu B3 generuje sygnały: ERROR/, INPOS/, ROBZSYNCHR/, WRITTEN/, BUDZIK/. Sygnały te są połączone poprzez bramki OC z wejściami portu PB układu C3 oraz łączówkę C. Powyższe połączenie umożliwia symulowanie przez interfejs IT1 wyjść sterownika położenia typu MA70.

Brama D6 wydaje sygnały A16 - A19 podawane na linii ADR16/ - ADR19/ testera poprzez bramki exclusive-or układu D7. Umożliwia to testowanie układów pamięci o adresach takich samych jak adresy pamięci testera. Sygnał ZER/ (układ D6) zeruje przerzutniki A5.

### 3.3.7. Układ przerwan (pakiet P2).

Układ przerwan wypracowuje iloczyn logiczny sygnałów INTO - INT7/ pakietu testowanego. Zbudowany jest z układu A1 typu Intel 3601 oraz bramy PC układu B1 typu Intel 8255. Przez bramę B1 czytany jest stan linii przerwan pakietu testowanego, natomiast układ A1 umożliwia połączenie dowolnej kombinacji logicznej stanów linii przerwan pakietu badanego do linii przerwan testera.

### 3.3.8. Układ dwukierunkowych bram WE/WY (pakiet P2).

W zależności od typu pakietu testowanego układ dwukierunkowych bram

WE-WY zapewnia odczyt lub wydawanie sygnałów, które dla jednych pakietów są wejściami a dla innych wyjściami. Kierunek pracy omawianego układu sterowany jest przez układ kontroli kierunku przesyłu informacji, który realizowany jest przez obwody C2 typu UCA 6474, obwód C3 typu Intel 3601 oraz bramkę DC obwodu C4 typu UCA 6476.

Układ ten jest zbudowany z układów A2 i A3 typu Intel 8287 oraz układu B1 typu Intel 8255, pełniącego funkcję dwukierunkowej bramy WE-WY.

Układ B1 przyjmuje sygnały: INTO/ - INT7/, RESET/, MPRO/, PFIN/, PFSN/, TEST/, OPIN/, OUTON/, oraz wydaje sygnały: ZAKAZ ZAPISU/, ZEZWOL/, STOP/, SEARCH/, REDPRED/, SYNCHR/, WEØ1C/. Sygnały te są podawane na złącze C pakietu P2 modułu IT1, skąd poprzez kabel podawane są na złącze C pakietu testowanego (MA-70).

### 3.4. Budowa i działanie pakietu interfejsu IT2

#### 3.4.1. Przeznaczenie i opis ogólny pakietu interfejsu IT2.

Pakiet interfejsu IT2 jest przeznaczony do testowania następujących pakietów i płytek USB:

- pakiet kontroli MW-31,
- pakiet kontroli MW-32,
- pakiet zasilania resolverów MZ-70,
- pakiet 16 WQE/16WY MC-42,
- płytka wyświetlaczy panelu programowania,

a ponadto są w nim wytwarzane sygnały sterujące prędkością obrotową silnika symulującego ruch zespołu wykonawczego robota. Pakiet IT2 umożliwia dołączanie pakietów testowanych do urządzenia testującego od strony ich złącz obiektowych lub - w przypadku płytki wyświetlaczy panelu programowania oraz pakietu kontroli MW-32 - złącz wewnętrznych płytki.

Wymiary oraz rodzaj i rozstaw złącz pakietu IT2 umożliwiają jego umieszczenie w kasie UTP i dołączenie go bezpośrednio do magistrali MF-31 testera. Na płycie czołowej pakietu IT2 są wyprowadzone sygnały, które podczas testowania są przekazywane poprzez pakiet złącz obiektowych PO do pakietu testowanego. Na rys.3 przedstawiono schemat blokowy interfejsu IT2, na podstawie którego poniżej omówiono zasadę działania tego pakietu.

#### 3.4.2. Podzespół dekodera adresów.

Interfejs IT2 zawiera dekodér 16-bitowego adresu zbudowany na dwóch pakietach Intel 3601 (D3,D4), umożliwiający adresowanie układów WE/WY tego pakietu wewnątrz obszaru 0000H ÷ FFFFH. Komunikacje pakietu IT2 z magistralą zapewniają dwa 8-bitowe nadajniki/odbiorniki typu Intel 8287 (D5,D6), w których kierunek przepływu informacji jest określony sygnałami sterującymi I/OWC i I/ORC, natomiast uaktywnienie tego układu jest określone sygnałem wyjściowym dekodera adresów.

#### 3.4.3. Podzespół sterujący testowaniem pakietów MW-31 oraz MW-32.

Podzespół sterujący testowaniem pakietów MW-31 i MW-32 uwzględnia testowanie tych obwodów ww. pakietów, do których jest dostęp od strony złącz obiektowych lub złącza wewnętrznego (MW-32) pakietu.

Tak więc są to detektory napięć:

- zasilających stałych +5V, +5VB, +12V, +15V, -15V, -5V,
  - przemiennego zasilającego urządzenie testujące (CNS)
- oraz układy logiczne odbierające sygnały WYL.AL., WYL.ZEG., WYL.BUDZ. i PRES..

Podczas testowania detektorów napięć sprawdza się kolejno ich działanie. Wykonyje się to poprzez symulacje spadków kontrolowanych napięć za pomocą rezystorów dołączanych do dzielników stanowiących obwody wejściowe detektorów napięć pakietów MW-31 i MW-32. Odpowiednie dołączenie rezystorów umożliwia: dekodér typu 74145(B6) oraz demultiplekser analogowy

MAC 08A(B7), które są sterowane poprzez brame WE/WY 74S412N. Sygnały sterujące są podawane z szyn danych na podstawie programu testującego dla pakietów MW-31 i MW-32.

#### 3.4.4. Podzespół sterujący testowaniem pakietu MZ-70.

Podzespół sterujący testowaniem pakietu MZ-70 zawiera obwody umożliwiające przeprowadzanie testowania napięć wyjściowych tego pakietu, przeznaczonych do zasilania transformatora położenia katowego (resolvera), sprzężonego z silnikiem zespołu wykonawczego robota.

Podczas testu są badane następujące zgodności napięć:

- |                    |                  |
|--------------------|------------------|
| a) $AS^+ = AS^-$ ? | g) $AS^+ = AN$ ? |
| b) $AC^+ = AC^-$ ? | h) $AS^+ = AN$ ? |
| c) $AS^+ = AC^-$ ? | i) $AC^+ = AN$ ? |
| d) $AC^+ = AS^-$ ? | j) $AC^- = AN$ ? |
| e) $AS^+ = AC^+/?$ | k) $COS^* = 0$ ? |
| f) $AC^- = AS^-$ ? | l) $SIN^* = 0$ ? |

przy czym:

$AS^+, AS^-$  : wartości amplitudy sygnału SIN, odpowiednio dodatniej i ujemnej

$AC^+, AC^-$  : wartości amplitudy sygnału COS, odpowiednio dodatniej i ujemnej

$COS^*, SIN^*$  : wartości składowych stałych sygnałów COS i SIN.

znak / oznacza zmianę znaku amplitudy sygnału,

indeks N oznacza wartość znamionową amplitudy sygnałów SIN i COS.

Test według punktów a i b ma na celu porównanie ze sobą amplitud dodatniej i ujemnej każdego z napięć: SIN i COS oraz stwierdzenie czy różnica mieści się w granicach określonych parametrami technicznymi pakietu MZ-70. Test według punktów c, d, e, f ma na celu sprawdzenie, czy różnice wartości amplitud sygnałów SIN i COS mieszczą się w granicach określonych parametrami technicznymi pakietu MZ-70.

Test według punktów g, h, i, j ma na celu sprawdzenie zgodności wartości amplitud napięć SIN i COS z wartościami określonymi parametrami technicznymi pakietu MZ-70.

Test według punktów k, l ma wykazać, że składowe stałe sygnałów SIN i COS są równe zero.

W celu umożliwienia testu według wyżej opisanego zamysłu zastosowano detektory szczytowe (D14, D15, D16, D17), które umożliwiają wydzielenie amplitud badanych sygnałów SIN i COS. Ponadto zastosowano dwa inwertyory analogowe (D12, D13) wypracowujące sygnały niezbędne przy ocenie wartości amplitud.

Porównywanie napięć o wartościach równych wartościom amplitud napięć SIN i COS realizuje układ sumatora analogowego, zbudowanego na wzmacniaczu operacyjnym (D11). Wyboru porównywanych napięć dokonuje się za pomocą demultipleksera analogowego (D7, D8). Układ ten jest sterowany programowo, za pomocą sygnałów sterujących przekazywanych z magistrali danych przez bramę wyjściową do pakietu IT2 (D5, D6).

Sygnał wyjściowy z sumatora jest podawany na układ detektora okienkowego (C5, C6, C7), na którego wyjściu pojawia się 3-bitowe słowo, jednoznacznie określające wynik każdego z testów, w sposób programowy.

Podczas testowania składowych stałych sygnałów SIN i COS, napięcia te są filtrowane przez układy RC (R26, C8, C9), a następnie podawane oddzielnie do układu wyżej opisanego detektorów okienkowych.

Stała czasowa filtrów jest tak dobrana, aby napięcia na ich wyjściach, przy prawidłowych napięciach SIN i COS, nie przekraczały 10% dopuszczalnej odchyłki składowej stałej od zera, określonej parametrami technicznymi pakietu MZ-70.

Powyżej opisano testowanie napięć SIN i COS, które są wytwarzane w przetworniku C/A pakietu MZ-70. Niezależnie od wyników tego testu, sprawdzana jest część cyfrowa pakietu MZ-70. Podczas testu części cyfrowej generator wewnętrzny pakietu testowanego MZ-70 jest blokowany (sygnał BLOK), a na jego wejście zegarowe jest podany z mikroprocesora sygnał GEN. Po każdym impulsie sygnału GEN następuje zmiana stanu 8-bitowego sygnału sterującego przetwornikiem C/A pakietu MZ-70. Sygnał ten jest

porównywany z takim samym sygnałem pakietu MZ-70 stanowiącym wyposażenie UTP. Oba te sygnały są podawane poprzez bramy WE (A1, A2, C2, C3) do mikroprocesora, gdzie następuje ich porównywanie. W celu zapewnienia synchronizacji 8-bitowych sygnałów cyfrowych pakietu testowanego MZ-70 i pakietu MZ-70 stanowiącego wyposażenie UTP, do obu pakietów jest doprowadzony sygnał ZER, ustawiający dla tych sygnałów te same warunki początkowe. Sygnały sterujące BLOK, GEN i ZER są generowane programowo przez mikroprocesor UTP i przesyłane bramą WY (B1) z pakietu IT2.

#### 3.4.5. Podzespół sterujący testowaniem płytki wyświetlaczy panelu programowania.

Podzespół ten umożliwia sprawdzenie jakości wyświetlaczy oraz ich połączeń na płytce drukowanej P3 wchodzącej w skład panelu programowania USB. Pakiet IT2 umożliwia dołączenie do tej płytki:

- sygnałów informacji wejściowej wyświetlaczy,
  - napięć zasilających wyświetlacze,
  - sygnału wpisyującego informacje wejściowe do rejestrów wyświetlaczy.
- W celu sprawdzenia każdego z punktów świetlnych oraz wykrycia ewentualnych zwarcień pomiędzy tymi punktami świetlnymi, wysterowywane są kolejne wiersze, a następnie kolumny wyświetlaczy. Przy wysterowywaniu kolejnych wierszy, na wszystkie wejścia sterujące płytki P3 są podawane "1" logiczne, a napięcie zasilające jest podawane kolejno na każdy wiersz. Przy wysterowywaniu kolejnych kolumn napięcie jest podawane do wszystkich wejść zasilających jednocześnie, natomiast na wejścia informacyjne jest podawana kolejno "1" logiczna. W obu opisanych wyżej przypadkach sygnał wpisywany jest impulsowo. Powyższe testowanie ma charakter statyczny. Zbliżenie do rzeczywistych warunków pracy wyświetlaczy osiąga się za pomocą testu dynamicznego, podczas którego ww. sygnały sterujące są modulowane sygnałami, przy których elementy diodowe wyświetlaczy nie świecą. Wyborem odpowiednich wierszy i kolumn wyświetlaczy steruje mikroprocesor, sygnałami podanymi bramą WE/WY (D5, D6). Sygnały te są buforowane odpowiednio:

- dla kolumn wyświetlaczy przez tranzystory T7÷T13,
- dla wierszy wyświetlaczy przez obwód A3.

Generator sterujący pracą wyświetlaczy zrealizowano w układzie: A4 (4, 5, 6; 1, 2, 3) A3(1, 2) tranzystory T14, T15.

W przypadku testowania płytki P3, mikroprocesor wysyła te sygnały na rozkaz wydany ręcznie - przez obserwatora testowanej płytki - za pomocą klawiatury testera. Jest to podyktowane tym, że ocenę poprawności działania wyświetlaczy przeprowadza człowiek nadzorujący przebieg testu.

#### 3.4.6. Podzespół testowania detektorów przeciążenia pakietu MC-42.

Podzespół ten umożliwia sterowane programowo - dołączenie obciążenia do wyjść pakietu testowanego MC-42, za pomocą sygnałów sterujących podawanych z mikroprocesora przez brame WE/WY (D5, D6).

Testowanie odbywa się dwuetapowo. Do wyjść są dołączane kolejno:

- obciążenie powodujące przekroczenie wartości granicznej układu wyjściowego pakietu MC-42 o 5% (sygnał KOOM).
- obciążenie powodujące przepływ prądu o wartości 95% wartości granicznej (sygnał KOCD).

Powyższy sposób testowania ma potwierdzić prawidłowość doboru progów zadziałania obwodu zabezpieczającego układy wyjściowe pakietu MC-42. Z pakietu IT2 jest wyprowadzony jeden tor do testowania wszystkich 16 wyjść pakietu MC-42 (KOC). Z tego względu do tego toru dołączane będą kolejne wyjścia pakietu MC-42 oddzielnie. Wybór obciążonego wyjścia jest dokonywany programowo poprzez wysterowanie odpowiedniego wejścia pakietu testowanego.

Układ realizujący wytwarzanie sygnału KOC składa się z elementów: B5(1, 2; 3, 4), T18÷T21.

W przypadku zadziałania układu przeciążenia dowolnego wyjścia pakietu testowanego pojawia się sygnał, który poprzez złącza systemowe jest odczytywany i interpretowany przez mikroprocesor.

### 3.4.7. Podzespół sterujący silnikiem wykonawczym.

Podzespół ten (B5(5,6;8,9), T22-T25) wytwarza napięcia  $V_{s+}$  i  $V_{s-}$ , które są przeznaczone do sterowania układu napędowego TAR-1A zasilającego silnik układu wykonawczego robota. Silnik wraz z układem napędowym stanowi wyposażenie urządzenia testującego. W przypadku testowania pakietu MA-70 silnik jest uruchamiany, a sygnał wyjściowy resolwera sprzężonego z silnikiem jest wykorzystywany przy testowaniu tego pakietu. Wytworzone napięcia  $V_{s+}$  lub  $V_{s-}$  jest sterowane programowo przez mikroprocesor, który przesyła sygnały sterujące - brama WE/WY - do źródeł napięcia powodując ich uruchomienie. Prędkość obrotowa silnika jest stała, określona za pomocą potencjometrów P18 i P19.

## 3.5. BUDOWA I DZIAŁANIE PAKIETU INTERFEJSU IT3

=====

### 3.5.1. Przeznaczenie i opis ogólny pakietu IT3.

Pakiet IT3 zawiera następujące podzespoły:

- dekodery adresów (C4, D2, D3, D6, ),
- układ sterujący (B1, C1, C2, C3, D1),
- bramki danych wewnętrznych (B3, B6),
- układy wejścia/wyjścia (C5),
- przetwornik C/A (A3, A5, B4, B5),
- układ interfejsu szeregowego (A6, A7, A9, A10, T1, T2, T3),
- bramki danych (C8, D7) i adresów (D4, D5) dla testowanych: pakietu MA-70 oraz panelu programowania,
- pamięć RAM (C6, C7),
- układ kontroli pracy krokowej CPU pakietu testowanego (A2, B2).

Możliwe są trzy tryby pracy pakietu IT3 :

#### 1. Stan HOLD CPU pakietu testowanego.

Układ sterujący zezwala na pracę w tym trybie o ile  $HOLD/\bar{H} = 0$  i  $HLDA = 1$ . W trybie 1 jednostka centralna testera ma dostęp do układów wewnętrznych zespołów, do których testowania jest przeznaczony pakiet IT3, a w zależności od stanu sygnału  $PROMINH/\bar{P}$  do :

- pamięci RAM pakietu testowanego ( $PROMINH/\bar{P} = 1$ ),
- pamięci ROM pakietu IT3 ( $PROMINH/\bar{P} = 0$ ).

#### 2. Praca CPU pakietu testowanego.

W tym trybie pracy jednostka centralna testera ma dostęp tylko do układów wewnętrznych pakietu IT3. CPU pakietu testowanego wykonuje, w zależności od stanu sygnału  $PROMINH/\bar{P}$ , program zawarty w pamięci ROM pakietu testowanego lub pamięci pakietu IT3.

(Ten stan pracy nie jest obecnie wykorzystany w testerze).

#### 3. Praca krokowa CPU pakietu testowanego.

W tym trybie pracy jednostka centralna testera ma dostęp tylko do układów wewnętrznych pakietu IT3. CPU pakietu testowanego wykonuje krokowo program zawarty w pamięci pakietu IT3. Aktywna rola w tym trybie pracy spełnia układ kontroli pracy krokowej. Układ ten jest sterowany programowo sygnałami z układów wyjściowych.

(Ten stan pracy nie jest obecnie wykorzystany w testerze).

## 4. URUCHAMIANIE

-----

W urządzeniu do uruchamiania i testowania pakietów układu sterowania uruchamianiu podlega przetwornik cyfrowo-analogowy znajdujący się na pakiecie IT3. Czynności wymagane przy uruchamianiu tego podzespołu są zawarte w programie URDACIT3 UTP.

W celu wywołania tego programu należy wywołać program dla testowania pakietu MA-70, a następnie przeprowadzić działania zgodnie z instrukcją

zawarta w programie.

Przed wykonaniem uruchamiania według programu URDACIT3, należy przeprowadzić działania zgodne z instrukcją zawartą w programie URDAC t.j. należy uruchomić i podłączyć pakiet MA-70.

## 5.OPIS INSTRUKCJI TESTOWANIA PAKIETÓW UKŁADU STEROWANIA ROBOTÓW

---

### 5.1 MENU GŁÓWNE

=====

#### 5.1.1.OPIS DZIAŁANIA PROGRAMU INITMM86.ASM.

- I. Wstępne zaprogramowanie pakietu MM86
- Programowanie zegara (8293).
- Programowanie interfejsu szeregowego (8251).

#### 5.1.2.OPIS DZIAŁANIA PROGRAMU MAIN.C.

- I. Organizacja programowa systemu ITELDIT-PROWAY
- II. Ustawienie wektorów przerwan dla wszystkich testów pakietów, programu uruchomieniowego i programu narzędziowego MON86.
- III. Rozpoczęcie programu zarządzającego programami testowymi pakietów oraz wyświetlenie na monitorze ekranowym główne menu (MENUGL.C)

### 5.2.PROGRAMY NARZĘDZIOWE

=====

#### 5.2.1.OPIS DZIAŁANIA PROGRAMU MON86.ASM.

Opis w załączniku nr.1

#### 5.2.2.OPIS DZIAŁANIA PROGRAMU PR\_UR.C.

- I. Projektowanie petli programowej.
  - Wybor i zapamiętanie rodzaju operacji, dokonuje się naciskając jeden lub kilka klawiszy z posród podanych niżej(koncząc to klawiszem enter CR):
  - i : wykonywany będzie rozkaz pobrania danej (słowo/bajt), z urządzenia we/wy.
  - o : wykonywany będzie rozkaz wysłania danej (słowo/bajt), na urządzenia we/wy.
  - r : wykonywany będzie rozkaz odczytu (słowo/bajt), z pamięci.
  - w : wykonywany będzie rozkaz zapisu (słowo/bajt), do pamięci.
  - b : wykonywany rozkaz dotyczył będzie operacji na danej 8-bitowej (nie nacisnięcie tego klawisza -domyślnie- operacja na danej 16-bitowej).
  - d : w tym miejscu petli wstawione będzie opóźnienie czasowe (x100ms).
  - \* : wykonywana będzie sekwencja rozkazów tego samego typu inkrementując adres w podanych granicach i ewentualnie inkrementując dana w określonych granicach (przy każdym adresie). Inkrementacja może być zerowa.
  - c : jednokrotne wcisnięcie klawisza c powoduje ustawienie sygnału OUTON/=0 na pakiecie testowanym MC42, co umożliwi komunikację testera z pakietem testowanym.
  - a : jednokrotne wcisnięcie klawisza a powoduje wprowadzenie mikroprocesora 8080 pakietu MA70 w stan HOLD co umożliwi komunikację testera z pakietem testowanym.
- UWAGA! nie należy wciskać tego klawisza jeśli nie podłączony st

- pakiet testowany
  - p : jednokrotne wcisniecie klawisza p powoduje wprowadzenie mikroprocesora 8080 panelu programowania stan HOLD co umozliwi komunikacje testera z pakietem testowanym.  
UWAGA! nie nalezy wciskac tego klawisza jesli nie podlaczony jest pakiet testowany
  - k : zakonczenie projektu petli.
- II. Uruchomienie i sterowanie petla programowa.  
Rozpoczecie dzialania petli nastepuje po naciśnięciu dowolnego klawisza. W trakcie wykonywania petli naciśnięcie ktoregos z klawiszy: h,r,p,x,k,m,s,ESC, powoduje odpowiednie dzialanie:
  - h (help) opis dzialania opisanych tu klawiszy.
  - k (krok) zmiana trybu pracy petli z krokowej na automatyczna odwrotnie.
  - r (raport) skrocony wydruk struktury zaprogramowanej petli.
  - p pomocnicze informacje o przebiegu petli.
  - s bezwzgledny skok do poczatku petli.
  - x (XACK) dodatkowe badanie sygnalu XACK/ pakietu testowanego, w wypadku pojawienia sie tego sygnalu drukowana jest odpowiednia informacja.
  - m (monitor) wstrzymanie/wznowienie wydruku danych na monitor, w trakcie wykonywania petli (poprawa obserwowalnosci oscyloskopu, szczegolnie przy wyloczonym monitorowaniu oraz przy usunieciu pomocniczych informacji o petli).  
Dane drukowane na monitorze pojawiaja sie tylko przy instrumencie: r, br, i, bi.
  - ESC przerwanie pracy programu.

UWAGA !

Dostep do pakietow testowanych takich jak MC42, MA70, panelu programowania, za pomoca tego programu wymaga odpowiednio : ustawienia sygnalu OUTON/=0 na pakiecie MC42. W pozostalych dwu przypadkach nalezy mikroprocesory wprowadzic w stan hold .

### 5.3. TESTOWANIE PAKIETU MC-42

=====

#### 5.3.1. OPIS DZIALANIA PROGRAMU ODCZYT.C.

- I. Sprawdzenie ustawienia przełącznika AP.  
Podawane sa adresy przy ktorych, w skrajnych polozeniach przełącznika AP pakiet powinien odezwać sie sygnalem XACK/.
- II. Test poprawnosci odczytu pakietem MC42.
  - 1. Podawanie na wejscia obiektowe pakietu testowanego pelzajacej jedynki.
  - 2. Odczyt laczem systemowym wartosci zarejestrowanej przez pakiet testowany wartosc zanegowana (negacja na pakiecie).
  - 3. Porownanie wartosci odczytanej z zapisana.
- III. Ocena wynikow testu.  
W przypadku pojawienia sie niezgodnosci przedstawiana jest binarna wielkosc slowa zapisywanego i odczytywanego z pakietu.

#### 5.3.2. OPIS DZIALANIA PROGRAMU OUTON.C.

- I. Sprawdzenie ustawienia przełącznika AP.  
Podawane sa adresy przy ktorych, w skrajnych polozeniach przełącznika AP pakiet powinien odezwać sie sygnalem XACK/.
- II. Test poprawnosci blokowania sygnalem OUTON/.
  - 1. Ustawienie sygnalu OUTON/ zezwalajaco.
  - 2. Zapis do pakietu testowanego wartosci 0000 hex, wartosc zanegowana (negacja na pakiecie).
  - 3. Odczyt z lacznych obiektowych wartosci sygnalow.
  - 4. Porownanie wartosci odczytanej z wartoscia zerowa.
  - 5. Ustawienie sygnalu OUTON/ zabraniajaco.
  - 6. Zapis do pakietu testowanego wartosci 0000 hex, wartosc zanegowana (negacja na pakiecie).

7. Porównanie wartości odczytanej z wartością zerową.
- III. Ocena wyników testu.  
W przypadku pojawienia się niezgodności w pierwszym przypadku i zgodności w drugim, drukowana jest binarna wielkość słowa zapisywanego i odczytywanego z pakietu.

#### 5.3.3. OPIS DZIAŁANIA PROGRAMU PRZEC.C.

- I. Sprawdzenie ustawienia przełącznika AP.  
Podawane są adresy przy których, w skrajnych położeniach przełącznika AP pakiet powinien odezwac się sygnałem XACK/.
- II. Test poprawności zapisu do pakietu .
- III. Test poprawności zapisu do pakietu przy obciążeniu wyjść pakietu poniżej granicy przeciążenia.
- IV. Test poprawności zapisu do pakietu przy obciążeniu wyjść pakietu powyżej granicy przeciążenia.
- V. W przypadku błędu udostępniona jest możliwość regulacji detektora przeciążenia.
- VI. Ocena wyników testu tak jak przy teście zapisu do pakietu.

#### 5.3.4. OPIS DZIAŁANIA PROGRAMU ZAPIS.C.

- I. Sprawdzenie ustawienia przełącznika AP.  
Podawane są adresy przy których, w skrajnych położeniach przełącznika AP pakiet powinien odezwac się sygnałem XACK/.
- II. Test poprawności zapisu do pakietu MC42.
  1. Podawanie na wejścia systemowe pakietu testowanego pelzające; jedynki. wartość zanegowana (negacja na pakiecie).
  2. Odczyt łączem obiektowym stanu wyjść pakietu testowanego.
  3. Porównanie wartości odczytanej z zapisaną.
- III. Ocena wyników testu.  
W przypadku pojawienia się niezgodności przedstawiana jest binarna wielkość słowa zapisywanego i odczytywanego z pakietu.

#### 5.4. TESTOWANIE PAKIETU MW-31

=====

##### 5.4.1. OPIS DZIAŁANIA PROGRAMU BKM.C.

- I. Testowanie bloku kontroli przesyłu po magistrali systemowej na pakiecie MW31.
  1. Sprawdzenie prawidłowego położenia zwory D2, umożliwiające, jednoznaczne adresowanie pakietu.
  2. Inicjalizacja sygnału INIT.
  3. Podawanie w odpowiedniej kolejności sygnałów sterujących MRDC/, IORC/, MWTC/, IDWC/ na przerzutniki D9, D8 oraz C1 (reset zespołu). Dane systemowe ffff hex.
  4. Odczyt z wyjścia przerzutnika D8.
- II. Ocena wyników testu.  
Ocena wyników testu jest przeprowadzana na podstawie słowa stanu pakietu, istotne są odstępów czasowe między sygnałami sterującymi podawanymi na przerzutniki.

##### 5.4.2. OPIS DZIAŁANIA PROGRAMU DEKMW31.C.

- I. Wstępne sprawdzenie stanu sygnału XACK/.
- II. Test pozytywny dekodera adresów pakietu.  
Podawane są adresy przy których, niezależnie od stanu zwory D2, pakiet powinien odezwac się sygnałem XACK/.
- III. Test dekodera adresów (przy 4 kombinacjach stanu zwory D2), w całej przestrzeni: 1 Mbyte, (opcjonalnie).
- IV. Ocena wyników testu.  
W przypadku pojawienia się sygnału XACK/ drukowany jest adres przy jakim to nastąpiło. Jeśli dekodery odezwali się nieprawidłowo dodatkowo drukowany jest odpowiedni komunikat.

#### 5.4.3.OPIS DZIAŁANIA PROGRAMU UKZ.C.

- I. Funkcja reseton.
  - 1.Sprawdzenie sygnału RESET/ przed podaniem P\_RESET,
  - 2.Sprawdzenie sygnału RESET/ po podaniu P\_RESET,
  - 3.Sprawdzenie sygnału OUTON po podaniu sygnału P\_RESET.
- II. Funkcja detuokvb.
  - 1.Podanie LOCK/=0,
  - 2.Podanie P-RESET/=1,
  - 3.Sprawdzenie Uok5VB w stanie normalnym,
  - 4.Sprawdzenie Uok5VB przy spadku napięcia 5VB,
  - 5.Sprawdzenie detektora Uok5VB po resecie CZYT\_A.
- III. Funkcja detuok.
  - 1.Sprawdzenie detektora w stanie normalnym,
  - 2.Sprawdzenie detektora przy kolejnych spadkach kontrolowanych napięć,
  - 3.Sprawdzenie detektora po powrocie napięć do wartości początkowych.
- IV. Funkcja detcns.
  - 1.Sprawdzenie sygnałów w warunkach normalnych (po resetach), testowane sygnały: PFIN/, PFSN/, MPRO/,
  - 2.Sprawdzenie sygnałów po zaniku napięcia CNS,
  - 3.Sprawdzenie sygnałów po powrocie napięcia CNS,
  - 4.Sprawdzenie sygnału MPRO/ przy zaniku napięcia Uok.

#### 5.4.4.OPIS DZIAŁANIA PROGRAMU UP.C.

- I. Inicjacja.
  - 1.Ustawienie sygnału INIT na 1,
  - 2.Zapalenie wszystkich diod elektroluminescencyjnych.
- II. Funkcja up.
  - 1.Gaszenie kolejno każdej z diod (ocena wyniku badania na podstawie obserwacji),
  - 2.Przy wskazanym ustawieniu przełączników C10 i E14 gaszone są kolejne diody. Podczas tej operacji sprawdzane są wyjścia:
    - a.alarm (sygnał wyjściowy ALARM),
    - b.przerwanie (odczyt w słowie stanu),
    - c.sygnał stanu (odczyt przez bramy WY pakietu).
- III. Funkcja int.
  - 1.Podanie sygnału INIT/ =1,
  - 2.Sprawdzenie sygnałów przerwan systemowych INT2, INT3, INT4,
    - a.w stanie normalnym,
    - b.po podaniu odpowiednio sygnałów UST1, UST2, Przerw.1,
  - 3.Sprawdzenie sygnałów przerwan po podaniu sygnałów ZER1 i ZER2.
- IV. Ocena testu.

Komentarze drukowane w czasie testu.

#### 5.5.TESTOWANIE PAKIETU MW-32

=====

##### 5.5.1.OPIS DZIAŁANIA PROGRAMU ALPRZER.C

Program ALPRZER.C zawiera podprogramy umożliwiające testowanie obwodów:

- wejścia/wyjścia dwustanowe; podprogram WEWY.C
- tory sygnałowe INTER/,BUDZIK, RESET; podprogram INTERBUD.C
- tor sygnałowy AL.MAG; podprogram ALMAG.C
- logika przerwan; podprogram PRZER.C
- logika alarmów:ALMAG,BUDZIK,ALARM,AL.DZW,AL.SW; podprogram TORALAR.

W ww. programach zastosowano metodę konwersacyjną ze względu na konieczność obsługi pakietu przez użytkownika testera w czasie wykonywania testu.Ponizej zostaną omówione poszczególne testy.

##### 5.5.1.1.OPIS DZIAŁANIA PODPROGRAMU WEWY.C

- I. Poszczególne wejścia dwustanowe są wysterowywane przy pomocy pakietu MC-42 stanowiącego wyposażenie testera.

- II. Wysterowywanie wejść dwustanowych powoduje zmianę stanu rejestru C7, którego stan jest interpretowany przez mikroprocesor w celu stwierdzenia poprawności działania tego obwodu.
- III. Zbiór sterowań umożliwiający testowanie tego obwodu:  
MM86>owfbf6,  
  - 100 TEMP,
  - 1 DYM,
  - 400 WENT,
  - 2000 ZAS.OB.,
  - 200 ZAS.OB.I,
  - 2 ZAS.OB.II,
  - 800 OTW,
  - 1000 WYL.AL.

#### 5.5.1.2.OPIS DZIAŁANIA PODPROGRAMU INTERBUD.C

- I. W stanie zerowym (bez wysterowania układu we/wy) jest sprawdzany stan sygnałów RESET oraz stan rejestru C7 dla sygnałów OTW i BUDZ. i stan rejestru C6 dla sygnału INTER.
- II. Po odpowiednim wysterowaniu badanego toru sprawdza się stany ww. sygnałów oraz sygnału RESET.  
Sygnał INTER wytwarzany jest poprzez podanie sygnałów sterujących na wejścia dwustanowe ZAS.OB., ZAS.OB.I., ZAS.OB.II.
- III. Oceny testu dokonuje się na podstawie zgodności sygnałów w rejestrach C6 i C7 oraz sygnału RESET z wartościami oczekiwanymi.

#### 5.5.1.3.OPIS DZIAŁANIA PODPROGRAMU ALMAG.C

- I. Podczas testu toru AL.MAG sprawdza się zapalenie lampki kontrolnej ALMAG przy podaniu sygnałów wymuszających stan ALMAG zgodnie z dokumentacją DTR pakietu MW-32.
- II. Oceny testu dokonuje się na podstawie obserwacji lampki ALMAG.

#### 5.5.1.4.OPIS DZIAŁANIA PODPROGRAMU PRZER.C

- I. Podczas odpowiedniego wymuszania sygnałów na krosie K3 przy pomocy sygnałów: USTI, ZERI, ZERP, INTER/ sprawdza się stany sygnałów INTO÷INT7.  
W czasie testu są podawane instrukcje postępowania dla użytkownika (zwieranie odpowiedniego punktu krosu K3 ).
- II. Sygnały USTI, ZERI, ZERP wymuszane są zgodnie z DTR pakietu MW-32.
- III. Oceny testu dokonuje się na podstawie odczytu wartości sygnałów INTO÷INT7.
- IV. Instrukcja odczytu stanu sygnałów przerwan:  
MM86>ibfedc,

#### 5.5.1.5.OPIS DZIAŁANIA PODPROGRAMU TORALARM.C

- I. Przy odpowiednim podawaniu sygnałów ALARM, ALMAG i OTW sprawdza się stany sygnałów AL.SW i AL.DZW., poprzez ich odczyt z wyjść dwustanowych.
- II. Instrukcja odczytu sygnałów AL.SW i AL.DZW:  
MM86>iwfbf6, (AL.SW-&1; AL.DZW-&8)
- III. Oceny testu dokonuje się na podstawie zgodności wartości sprawdzanych sygnałów z wartościami oczekiwanymi.

#### 5.5.2.OPIS DZIAŁANIA PROGRAMU DEKADR.C

- I. Test OGRANICZONY dekodera adresów pakietu.  
Podawane są adresy, przy których pakiet powinien odezwać się sygnałem XACK/. Test jest wykonywany przy zmiennych wartościach sygnału MPRO, który może blokować obwód D5 dekodera adresów.
- II. Test PELNY.  
Podawane są adresy, przy których nie powinien pojawić się sygnał potwierdzenia XACK/. (przy stanie sygnału MPRO/=0).

### III. Testowanie sygnału MMAP2.

Podawane są adresy ze zbioru adresów, przy których pojawia się sygnał MMAP2.

### IV. Ocena wyników testu.

W przypadku nieuzasadnionego pojawienia się sygnału XACK/ lub jego nieuzasadnionego braku drukowany jest adres przy jakim to nastąpiło oraz dodatkowo drukowany jest odpowiedni komunikat.

Pojawieniu się sygnału MMAP2 towarzyszy komunikat wskazujący adres, przy którym to nastąpiło. Nieuzasadniony brak sygnału MMAP2 jest wykazany odpowiednim komunikatem.

### V. Niezależnie od przebiegu testu możliwe jest oglądanie przy pomocy oscyloskopu sygnałów wyjściowych dekodera adresów - zgodnie z informacją zawartą w komunikacie.

### VI. Wartości adresów pakietu zawarte są w DTR pakietu MW-32.

## 5.5.3.OPIS DZIAŁANIA PROGRAMU MAGISTR.C

I. Przez mikroprocesor wysyłane są w odpowiedniej kolejności dane: 0x0000, 0xffff, 0x5555, 0xaaaa. Dane te oraz towarzyszące im sygnały sterujące IORC/, IOWC/, MRDC/, MWTC/, XACK/ są zapamiętywane przez rejestry danych i rejestr rozkazów (E3, E5, E6).

### II. Ocena wyników testu.

W przypadku niezgodności pomiędzy wartościami umieszczonymi w rejestrach E3, E5, E6 z wartościami wynikającymi z aktualnego sterowania w czasie testu, drukowany jest odpowiedni komunikat o wystąpieniu błędu.

## 5.5.4.OPIS DZIAŁANIA PROGRAMU MONITOR.C

I. Przy odpowiednim programowym podawaniu sygnału OTW oraz przełączaniu przełącznika MONITOR zgodnie z instrukcjami podawanymi na bieżąco przez program testujący, jest badany stan linii DI3 rejestru C6.

### II. Ocena testu.

Wynik testu jest pozytywny, jeśli wartości sygnału odczytywanego z rejestru C6 są zgodne z wartościami oczekiwanymi.

## 5.5.5.OPIS DZIAŁANIA PROGRAMU PRESET.C

I. Po programowym ustawieniu układów pakietu MW-32 UTP oczekuje na przełączenie przycisku PRESET pakietu informując o tym odpowiednim komunikatem.

### II. Ocena wyników testu.

W przypadku braku odpowiedzi pakietu w postaci wygenerowanych sygnałów OUTON/ i RESET/ podany jest komunikat o niesprawności testowanego obwodu.

## 5.5.6.OPIS DZIAŁANIA PROGRAMU REJ9LAM.C

I. Przy programowym sterowaniu świeceniem lampek na płycie czołowej pakietu MW-32, obserwuje się poprawność ich świecenia.

### II. Ocena wyniku testu.

Ocenę przeprowadza użytkownik na podstawie obserwacji lampek.

## 5.5.7.OPIS DZIAŁANIA PROGRAMU UKZ.C

I. W stanie odpowiadającym prawidłowemu stanowi pracy sprawdzane są sygnały: STYCZNIK/, ZAKL.PAM./, MPRO/, RESET/, PFIN/, PFSN/, OUTON/.

II. Przy programowo symulowanym spadku napięcia AC badane są sygnały: RESET/, OUTON/, PFSN/, PFIN/.

III. Po zasymulowanym powrocie napięcia AC badane są stany sygnałów: RESET/, OUTON/, PFSN/, PFIN/.

IV. Przy programowo symulowanym spadku kolejno wszystkich napięć stałych dochodzących do pakietu MW-32 badane są sygnały: RESET/, OUTON/, MPRO/, STYCZNIK/, ZAKL.PAM./.

- V. Po zasymulowanym powrocie napięć stałych badane są sygnały: RESET/, OUTON/, MPRO/, STYCZNIK/, ZAKL.PAM./.
- VI. Ocena wyników testu.  
Interpretacja wyników pomiarów jest przeprowadzana przez mikroprocesor. W przypadku niezgodności wartości badanych sygnałów z wartościami oczekiwanymi drukowany jest odpowiedni komunikat.

## 5.6.TESTOWANIE PAKIETU MZ-70

=====

### 5.6.1.OPIS DZIAŁANIA PROGRAMU MZ-70.C.

#### I. Blok cyfrowy.

- 1.Sprawdzenie działania przycisków sterujących poprzez odczyt wartości max 1 min osmiobitowych słów wyjściowych bloku.
- 2.Sprawdzenie działania bloku poprzez porównanie wszystkich możliwych słów osmiobitowych z takimi samymi słowami generowanymi przez pakiet wzorcowy przy synchronicznym sterowaniu obu pakietów.

#### II. Blok analogowy.

Porównanie ze sobą lub z napięciem wzorcowym amplitud sygnału SIN i COS oraz składowych stałych tych sygnałów pakietu testowanego następująco:

|                 |          |
|-----------------|----------|
| a/.SIN+ i SIN-, | g/.SIN+, |
| b/.SIN+ i COS+, | h/.SIN-, |
| c/.SIN+ i COS-, | i/.COS+, |
| d/.COS+ i COS-, | j/.COS-, |
| e/.COS+ i SIN-, | k/.COS~, |
| f/.COS- i SIN-, | l/.SIN~, |

#### III. Ocena wyników testowania.

W przypadku wykrycia niezgodności pomiędzy sygnałami wyjściowymi a wzorcowymi jest drukowany komunikat o błędzie z wyszczególnieniem rodzaju błędu.

### 5.6.2.OPIS DZIAŁANIA PROGRAMU DAC.C.

#### I. Inicjacja.

Wprowadzenie pakietu MA70 w stan HOLD.

- II. Porównywanie napięć wyjściowych pakietu testowanego i pakietu wzorcowego dla każdej wartości słowa sterującego pracą DAC.
  - 1.Napięcie wyjściowe DAC pakietu testowanego jest stale ustawiane na poziomie przewyższającym napięcie wyjściowe DAC pakietu wzorcowego o wartość odpowiadającą dwóm schodkom napięcia DAC.  
Testuje się, czy napięcie na pakiecie nie jest zbyt niskie.
  - 2.Napięcie wyjściowe DAC pakietu testowanego jest stale ustawiane na poziomie niższym od napięcia wyjściowego DAC pakietu wzorcowego o wartość odpowiadającą dwóm schodkom napięcia DAC.  
Testuje się, czy napięcie na pakiecie nie jest zbyt wysokie.

#### III. Ocena wyników testu.

Ocena jest przeprowadzana na bieżąco; w przypadku niedopuszczalnej wartości odchyłki jest drukowany komunikat o błędzie z zaznaczeniem wartości słowa sterującego dla tego stanu.

## 5.7.TESTOWANIE PAKIETU ML-50

=====

Testy uruchomieniowe dla pakietu ML-50 zawarte są w sprawozdaniu: PIAP Nr rej.5652. Zostały one opracowane w ramach tematu: RP-58.1.

## 5.8.TESTOWANIE PAKIETU MI-50

=====

Testy uruchomieniowe dla pakietu ML-50 zawarte są w sprawozdaniu: PIAP Nr rej.5652. Zostały one opracowane w ramach tematu: RP-58.1.

Schematy polaczen krosow na pakiecie.  
 Uwaga: Krosy A1 i A2 nalezy zalozyc na czas testowania pakietu.  
 Przy pomocy krosu D6 jest ustawiany adres pakietu (tu 20 hex).



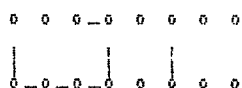
A1

A2



D1

D2



D6

## 5.9.TESTOWANIE PAKIETU MA-70

=====

### 5.9.1.OPIS DZIALANIA PROGRAMU DEKMA70.C.

- I. Inicjacja.  
Wprowadzenie w stan HOLD pakietu testowanego.
- II. Testowanie sygnalow WEOC, WE1C, WYOC, WY1C.  
1.Podawane sa adresy ze zbioru sterowan dla kazdej wartosci cztero-bitowego slowa ZA = XXXX.  
Wynika to z ogolnej postaci adresow dekodera pakietu MA70:  
WEOC,WYOC = XX(4\*ZA)  
WE1C,WY1C = XX(4\*ZA+2)  
2.W czasie testu jest obserwowany sygnal XACK/.
- III. Ocena wynikow testu.  
W przypadku nieuzasadnionego pojawienia sie sygnalu XACK/ lub jego nieuzasadnionym niewystapieniu, drukowany jest komunikat o bledzie wraz z adresem podanym do pakietu oraz wartoscia stanu ZA i nazwa sygnalu wyjsciowego dekodera.

### 5.9.2.OPIS DZIALANIA PROGRAMU DKMA70WE.C.

- I. Inicjacja.  
Wprowadzenie w stan HOLD pakietu MA-70.
- II. Testowanie sygnalow WY0-WY4 i WEO-WE5, oraz CSRAM/ i CSPROM/.  
Podawane sa kolejne wartosci adresow ze zbioru sterowan oraz obserwowany jest sygnal SREADY/.
- III. Ocena wyniku testu.  
W przypadku nieuzasadnionego pojawienia sie sygnalu SREADY/ lub jego nieuzasadnionego niewystapienia drukowany jest komunikat o bledzie wraz z adresem podanym do pakietu oraz nazwa sygnalu wyjsciowego w ktorym wystapil blad.

### 5.9.2.OPIS DZIALANIA PROGRAMU EPROM.C.

- I. Inicjacja.  
1.Wprowadzenie w stan HOLD pakietu.  
2.Podanie sygnalu PROMINH = 1.

- II. Testowanie pamięci EPROM.
  - 1. Przy adresach podawanych do pakietu ze zbioru sterowań dla pamięci EPROM tego pakietu, jest odczytywana suma zawartości pamięci.
- III. Ocena wyniku testu.
  - 1. W przypadku niezgodności odczytanej wartości z wartością wzorcową jest drukowany komunikat o błędzie.

#### 5.9.3.OPIS DZIAŁANIA PROGRAMU MP1MP2.C.

- I. Inicjacja.
  - Wprowadzenie pakietu MA70 w stan HOLD.
- II. Testowanie układu wyboru parametrów oraz przełącznika przerwan.
  - 1. Podaje się komunikaty o wymaganych położeniach segmentów testowanych przełączników.
  - 2. Sprawdza się rzeczywiste stany segmentów tych przełączników.
- III. Ocena wyników testowania.
  - W przypadku wystąpienia niezgodności stanu położenia segmentów przełączników z oczekiwanym; drukowany jest komunikat o błędzie oraz jest podawana wartość odczytana położenia segmentów każdego z przełączników.

#### 5.9.4.OPIS DZIAŁANIA PROGRAMU PWS.C.

- I. Inicjacja.
  - 1. Wprowadzenie pakietu MA70 w stan HOLD.
  - 2. Uruchomienie generowania napięć SIN i COS zasilających resolver.
- II. Testowanie zdolności do rejestrowania układu kontroli położenia wału silnika.
  - 1. Uruchomienie silnika zgodnie z ruchem wskazówek zegara.
  - 2. Dynamiczne odczytywanie zawartości rejestrów i porównywanie ich z wartościami wpisanymi poprzednio.
  - 3. Sprawdzenie wystąpienia sygnału INT oraz jego zaniku w rejestrze.
  - 4. Powtórzenie powyższych działań przy silniku obracającym się w kierunku przeciwnym do ruchu wskazówek zegara.
- III. Ocena wyników testu.
  - 1. W przypadku, gdy stan sygnału INT jest niezgodny z wartością oczekiwaną drukowany jest komunikat.
  - 2. W przypadku, gdy przyrost odczytywanych wartości z rejestru jest różny od 1 i różny od -1 i różny od 1023 i różny od -1023 jest drukowany komunikat o błędzie z podaniem wartości przyrostu oraz wartości, która powinna zostać odczytana.

#### 5.9.5.OPIS DZIAŁANIA PROGRAMU RAM.C.

- I. Inicjacja.
  - Wprowadzenie pakietu MA70 w stan HOLD.
- II. Testowanie pamięci RAM.
  - 1. Dla każdej wartości adresu ze zbioru sterowań pamięci RAM pakietu są zapisywane, a następnie odczytywane następujące wartości danych: 0x00, 0xff, 0x55, 0xaa.
- III. Ocena wyników testu.
  - W przypadku błędnego wyniku odczytu jest drukowany komunikat o błędzie wraz z wartościami adresu pamięci RAM :
    - wartością danej wpisywanej,
    - wartością danej odczytanej.

#### 5.9.6.OPIS DZIAŁANIA PROGRAMU WEWY.C.

- I. Inicjacja.
  - Wprowadzenie pakietu MA70 w stan HOLD.
- II. Testowanie rejestru położenia rzeczywistego.
  - 1. Do rejestru są wpisywane kolejno pełzająca jedynka oraz pełzające zero.
  - 2. Zawartość rejestru odczytuje się programowo i porównuje z wartością oczekiwaną.
- III. Testowanie rejestru zadanego przyrostu.

1. Do rejestru sa wpisywane kolejno pelzajaca jedynka oraz pelzajace zero.
  2. Zawartosc rejestru odczytuje sie programowo i porownuje z wartoscia oczekiwana.
- IV. Testowanie ukladow wejscia.
1. Do ukladow sa podawane kombinacje sygnalow wejsciovych: ZAK.ZAPIS, REDPRED, SEARCH, ZEZWOL, SYNCHR, WRITTEN, SYNCSW, CZUJNIK, OSZSYNCHR (uklady wejscia testuje sie przy zamontowanym i wymontowanym obwodzie D6).
  2. Jest odczytywany stan zawartosci ukladow wejscia i porownywany z wartoscia oczekiwana.
- V. Testowanie rejestru stanu sterownika polozenia.
1. Do rejestru sa wpisywane kolejno pelzajaca jedynka i pelzajace zero.
  2. Jest odczytywana zawartosc rejestru i porownywana z wartoscia oczekiwana.
  3. Test przeprowadza sie przy zamontowanym oraz wymontowanym obwodzie D5 pakietu.
- VI. Ocena wynikow testu.
- W kazdym z powyzzszych przypadkow, w razie wystapienia niezgodnosci pomiedzy wartosciami zapisywanymi i odczytywanymi jest drukowany komunikat o bledzie ze wskazaniem wpisywanej i odczytywanej danej oraz nazwa testowanego obvodu.

#### 5.10. TESTOWANIE PANELU PROGRAMOWANIA

=====

##### 5.10.1. OPIS DZIALANIA PROGRAMU EPROM.C.

- I. Inicjacja.
  1. Wprowadzenie w stan HOLD pakietu.
  2. Podanie sygnalu PROMINH = 1.
- II. Testowanie pamieci EPROM.
 

Przy adresach podawanych do pakietu ze zbioru sterowan dla pamieci EPROM tego pakietu, jest odczytywana suma zawartosci kazdej pamieci typu 2764, oraz calkowita suma zawartosci wszystkich pamieci eeprom tego pakietu.

##### 5.10.2. OPIS DZIALANIA PROGRAMU RAM.C.

- I. Inicjacja.
 

Wprowadzenie pakietu w stan HOLD.
- II. Testowanie pamieci RAM.
 

Dla kazdej wartosci adresu ze zbioru sterowan pamieci RAM pakietu sa zapisywane, a nastepnie odczytywane nastepujace wartosci danych: 0x00, 0xff, 0x55, 0xaa.
- III. Ocena wynikow testu.
 

W przypadku bladnego wyniku odczytu jest drukowany komunikat o bledzie wraz z wartosciami adresu pamieci RAM :

wartoscia danej wpisywanej,  
wartoscia danej odczytanej.

##### 5.10.3. OPIS DZIALANIA PROGRAMU PISO.C.

- I. Inicjacja.
 

Wprowadzenie pakietu w stan HOLD.

Reset interfejsow szeregowych (pakietu testowanego i testera).
- II. Testowanie USART'a (8251)
  1. Programowanie dzielnika czestotliwosci na predkosc transmisji 4800 bodow.
  2. Programowanie USART'a na 8-bitow danych, 2-stopu, dzielnik wewnetrzny = 1/16.
  3. Dla kazdej wartosci danych ze zbioru (0 hex - ff hex) nastepuja przesyly asynchroniczny w obu kierunkach. Obsluga interfejsow odbywa sie na zadanym kierunku sprawdzania rejestru stanu obu ukladow.
- III. Ocena wynikow testu.
 

W przypadku bladnego wyniku odczytu jest drukowany komunikat o bledzie

dzie wraz z wartosciami danych: odebranej i nadawanej.  
W przypadku zawieszenia transmisji drukowany jest komunikat o przyczynie bledu.

#### 5.10.4.OPIS DZIALANIA PROGRAMU LED.C.

- I. Inicjacja.  
Wprowadzenie pakietu w stan HOLD.  
Zaprogramowanie bram we/wy (8255).
- II. Testowanie diod LED.  
Ustawienie wszystkich sygnalow sterujacych naprzemian w stan wysoki i niski, co powoduje swiecenie i gasniecie LED.
- III. Ocena wynikow testu.  
Niezgodnosc informacji wydrukowanej na monitorze z zaobserwowana na panelu programowania, swiadczy o nieprawidlowym dzialaniu badanego elementu.

#### 5.10.5.OPIS DZIALANIA PROGRAMU JOI.C.

- I. Inicjacja.  
Wprowadzenie pakietu w stan HOLD.  
Zaprogramowanie bram we/wy (8255).
- II. Testowanie diod JOYSTICK'a  
1.Wystawianie trzech roznych sygnalow analogowych z testera na wejscia joystick'a  
2.Sterowanie multiplekserem analogowym i przetwornikiem a/c portem PC bramy (8255), odczyt wartosci cyfrowej brama (8255) portem PB.
- III. Ocena wynikow testu.  
Niezgodnosc wartosci analogowej, podawanej na pakiet testowany z jej interpretacja cyfrowa przedstawiona jest odpowiednim komunikatem na monitorze ekranowym.

#### 5.10.6.OPIS DZIALANIA PROGRAMU PRZYC.C.

- I. Inicjacja.  
Wprowadzenie pakietu w stan HOLD.  
Zaprogramowanie bram we/wy (8255).
- II. Testowanie zespolu PRZYCISKOW.  
1.W petli programowej wystawiane jest sterowanie ukkladu U124 (74156) brama (U122), co powoduje sekwencyjne zalaczenie zasilania na wszystkie przyciski.  
2.Odczyt stanu portu PB bramy U122 (8255) pozwala na identyfikacje nacisnietego przycisku, co przedstawione jest odpowiednim komunikatem.
- III. Ocena wynikow testu.  
Niezgodnosc wartosci wydrukowanej na monitorze ekranowym z widokiem elementow plytki P3 panelu programowania oznacza nieprawidlowe dzialanie uzytego przycisku.

#### 5.10.7 OPIS DZIALANIA PROGRAMU WYS.C.

- I. Inicjacja.  
Wprowadzenie pakietu w stan HOLD.
- II. Testowanie bloku pamieci ram wyswietlaczy i sterowania wyswietlaniem  
- ustawienie wyswietlania w stan "zakaz wyswietlania",  
- wypelnianie pamieci ram wyswietlaczy na przemian wartosciami 0 i 1,  
- ustawienie wyswietlania w stan "zezwozenie wyswietlania",  
- weryfikacja komunikatu wydrukowanego na monitorze ze stanem wyswietlaczy.
- III. Ocena wynikow testu.  
Niezgodnosc informacji wydrukowanej na monitorze z zaobserwowana na panelu programowania swiadczy o nieprawidlowym dzialaniu badanego elementu.

#### 5.10.8.OPIS DZIAŁANIA PROGRAMU BUDZ.C.

##### I. Inicjacja.

Wprowadzenie pakietu w stan HOLD.

Zaprogramowanie bram we/wy (8255).

##### II. Testowanie sygnału BUDZIK/

1.Ustawienie sygnałów sterujących świeceniem diod LED w stan wysoki.

2.Sprawdzenie poprawności działania sygnału START BUDZIK/:

3.Zalaczenie sygnału START BUDZIK/ przy braku podtrzymywania przerzutnika U105, co powoduje reset panelu i gasnienie LED.

4.Sprawdzenie poprawności działania sygnału BUDZIK/:

- zaprogramowanie bram we/wy (8255),

- ustawienie sygnałów sterujących świeceniem diod LED w stan wysoki,

- zalaczenie sygnału START BUDZIK/ z podtrzymywaniem przerzutnika sygnałem BUDZIK/ w odstępach zadanych przez operatora,

(zbyt krótkie opóźnienie powoduje reset panelu i gasnienie LED).

##### III. Ocena wyników testu.

Niezgodność informacji wydrukowanej na monitorze z zaobserwowaną na panelu programowania świadczy o nieprawidłowym działaniu badanego elementu.

| PAKIET TESTERA                        | IT1                                                                                                                                                                             |
|---------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| BRAMA(8212):<br>(I/O)                 | (D6)    adres portu fed0 hex                                                                                                                                                    |
| BRAMA(8212):<br>(I/O)                 | (B2)    adres portu fec0 hex                                                                                                                                                    |
| BRAMA WEJSCIA/WYJSCIA(8255):<br>(I/O) | (B1)    adres portu kontrolnego fede hex<br>(B1)    adres portu danych PA    fed8 hex<br>(B1)    adres portu danych PB    feda hex<br>(B1)    adres portu danych PC    fedc hex |
| BRAMA WEJSCIA/WYJSCIA(8255):<br>(I/O) | (B3)    adres portu kontrolnego fec8 hex<br>(B3)    adres portu danych PA    fec8 hex<br>(B3)    adres portu danych PB    fec8 hex<br>(B3)    adres portu danych PC    fecc hex |

| PAKIET TESTERA           | IT2                                                                                                        |
|--------------------------|------------------------------------------------------------------------------------------------------------|
| BRAMA(74S412N):<br>(I/O) | (B1)    adres portu fbd0 hex (pracuje jako wyjscie)<br>(C1)    adres portu fbd0 hex (pracuje jako wyjscie) |
| BRAMA(74S412N):<br>(I/O) | (A1)    adres portu fbd0 hex (pracuje jako wejscie)<br>(C2)    adres portu fbd0 hex (pracuje jako wejscie) |
| BRAMA(74S412N):<br>(I/O) | (A2)    adres portu fbc8 hex (pracuje jako wejscie)<br>(C3)    adres portu fbc8 hex (pracuje jako wejscie) |
| BRAMA(74S412N):<br>(I/O) | (B2)    adres portu fbc0 hex (pracuje jako wejscie)<br>(B4)    adres portu fbc0 hex (pracuje jako wyjscie) |
| BRAMA(74S412N):<br>(I/O) | (B3)    adres portu fbc8 hex (pracuje jako wyjscie)<br>(C4)    adres portu fbc8 hex (pracuje jako wyjscie) |

|                                                                                                                                                                       |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| UWAGA !                                                                                                                                                               |
| <p>WSZYSTKIE ADRESY (I/O) PANELU PROGRAMOWANIA I MA70<br/> SA SUMOWANE Z WARTOSCIA F000 hex<br/> WARTOSC TA JEST DEKODOWANA NA PAKIECIE IT3 (BRAMY D5 D4)</p>         |
| <p>WSZYSTKIE ADRESY (READ/WRITE) PANELU PROGRAMOWANIA I MA70<br/> SA SUMOWANE Z WARTOSCIA 80000 hex<br/> WARTOSC TA JEST DEKODOWANA NA PAKIECIE IT3 (BRAMY D5 D4)</p> |

|                                                                                                                                                                                                                                                                                                                           |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| WPROWADZENIE MIKROPROCESORA PAKIETU TESTOWANEGO W STAN HOLD                                                                                                                                                                                                                                                               |
| <p>zapisanie na pakiet testera (IT3) do bramy B5 (port PC)<br/> wartosci 78 hex,ustawienie sygnalu HOLD:brama B3 pakietu(IT3)<br/> dana 10 hex,ustawienie sygnalu INIT/ na pakiecie (IT1):brama D6,<br/> dana 0 hex nastepnie po odpowiedniej zwloce czasowej; ustwienie<br/> sygnalu INIT(IT1) brama D6 dana 20 hex.</p> |

| PAKIET TESTERA                                                                                                                                                                                                                              | IT3 |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| <p>USART(7851):<br/> (I/O)</p> <p>(A6)    adres portu kontrolnego fdd2 hex<br/> (A6)    adres portu danych        fdd0 hex</p>                                                                                                              |     |
| <p>BRAMA(8212):<br/> (I/O)</p> <p>(B3)    adres portu: fdd8 hex</p>                                                                                                                                                                         |     |
| <p>BRAMA WEJSCIA/WYJSCIA(8255):<br/> (I/O)</p> <p>(B5)    adres portu kontrolnego fdce hex<br/> (B5)    adres portu danych PA    fdc8 hex<br/> (B5)    adres portu danych PB    fdca hex<br/> (B5)    adres portu danych PC    fdcc hex</p> |     |
| <p>BRAMA WEJSCIA/WYJSCIA(8255):<br/> (I/O)</p> <p>(B6)    adres portu kontrolnego fdc6 hex<br/> (B6)    adres portu danych PA    fdc0 hex<br/> (B6)    adres portu danych PB    fdc2 hex<br/> (B6)    adres portu danych PC    fdc4 hex</p> |     |
| <p>RAM:</p> <p>(C6)    adresy ad 80000 hex do 801ff hex<br/> (C7)    adresy ad 80200 hex do 803ff hex</p>                                                                                                                                   |     |

|                                                                                                                                                                          |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| UWAGA !                                                                                                                                                                  |
| <p>DOSTĘP DO PAKIETU MC42 PO RESECCIE MOŻLIWY JEST PO WCZESNIEJSZYM USTAWIENIU SYGNAŁU OUTON/.</p> <p>I/O:OUTON/<br/> wysłać pod port fed0 hex dana typu bajt 10 hex</p> |

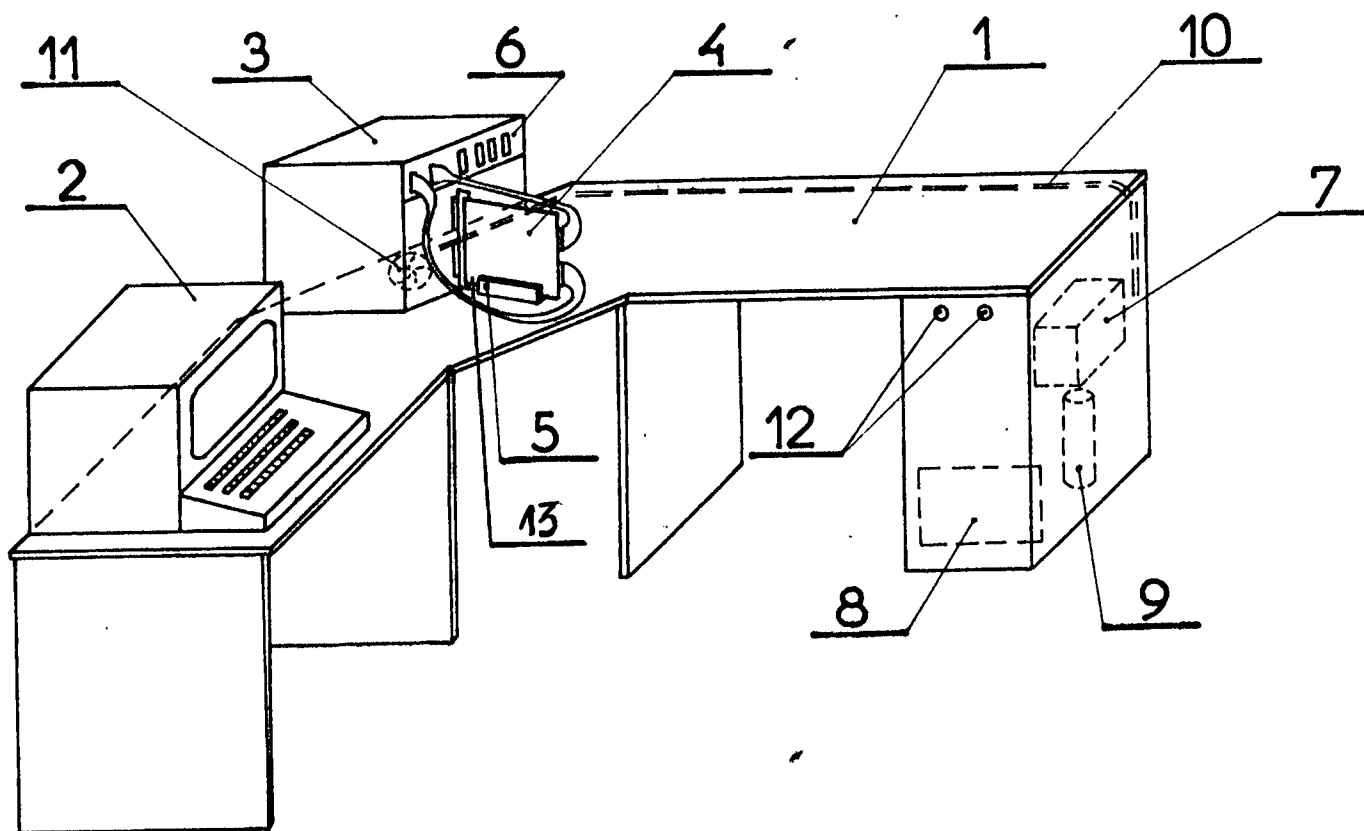
|                                                                                                                                                  |
|--------------------------------------------------------------------------------------------------------------------------------------------------|
| PAKIET MC42 - TESTOWANY!                                                                                                                         |
| <p>I/O:</p> <p>mikroprzełącznik AP0,AP1,AP2,AP3,AP4 zwarty ADRES: fb06 hex<br/> mikroprzełącznik AP0,AP1,AP2,AP3,AP4 rozarty ADRES: fbfe hex</p> |

|                                     |
|-------------------------------------|
| PAKIET MC42 - TESTERA!              |
| <p>I/O:</p> <p>ADRES : fbf6 hex</p> |

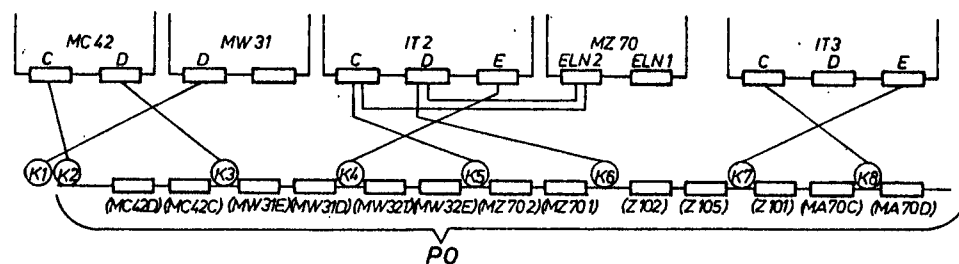
|                                                                                                                                  |
|----------------------------------------------------------------------------------------------------------------------------------|
| UWAGA !                                                                                                                          |
| <p>WSZYSTKIE ADRESY (I/O) PANELU PROGRAMOWANIA I MA70<br/> SA SUMOWANE Z WARTOŚCIĄ F000 hex (dekodowana przez tester)</p>        |
| <p>WSZYSTKIE ADRESY (READ/WRITE) PANELU PROGRAMOWANIA I MA70<br/> SA SUMOWANE Z WARTOŚCIĄ 8000 hex (dekodowana przez tester)</p> |

|                                                                                                                                                                                                                                                                                                                           |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| WPROWADZENIE MIKROPROCESORA 8080(U101) W STAN HOLD                                                                                                                                                                                                                                                                        |
| <p>zapisanie na pakiet testera (IT3) do bramy B5 (port PC)<br/> wartości 78 hex,ustawienie sygnału HOLD:brama B3 pakietu(IT3)<br/> dana 10 hex,ustawienie sygnału INIT/ na pakiecie (IT1):brama D6,<br/> dana 0 hex następnie po odpowiedniej zwłoce czasowej ustawienie<br/> sygnału INIT(IT1) brama D6 dana 20 hex.</p> |

|                                    |                                                                                                                                                                                                                                                      |
|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PANEL PROGRAMOWANIA                |                                                                                                                                                                                                                                                      |
| EPROM:                             | (U110) adresy ad 80000 hex do 81fff hex<br>(U111) adresy ad 82000 hex do 83fff hex<br>(U112) adresy ad 84000 hex do 85fff hex                                                                                                                        |
| RAM:                               | (U113) adresy ad 86000 hex do 867ff hex                                                                                                                                                                                                              |
| USART:                             | (U118) adres portu kontrolnego f08b hex<br>(U118) adres portu danych f08a hex                                                                                                                                                                        |
| LICZNIK PROGRAMOWALNY:             | (U119) adres portu kontrolnego f08f hex<br>(U119) adres portu danych licznika 0 f08c hex<br>(U117) adres portu danych licznika 0 f08d hex<br>(U119) adres portu danych licznika 0 f08e hex                                                           |
| BRAMY WEJSCIA/WYJSCIA(8255):       | (U121) adres portu kontrolnego f083 hex<br>(U121) adres portu danych PA f080 hex<br>(U121) adres portu danych PB f081 hex<br>(U121) adres portu danych PC f082 hex                                                                                   |
|                                    | (U122) adres portu kontrolnego f087 hex<br>(U122) adres portu danych PA f084 hex<br>(U122) adres portu danych PB f085 hex<br>(U122) adres portu danych PC f086 hex                                                                                   |
| RAM WYSWIETLACZY(I/O):             | (U212)(U213) adres f07c hex zakaz wyswietlania<br>(U212)(U213) adres f07e hex zezwolenie wyswietlania<br>(U212)(U213) adres f07d hex (kazdy zapis pod ten adres pcwoduje<br>inkrementacje fizycznych adresow MCY7114 modulo<br>336 zaczynajac od 0 ) |
| PRZERZUTNIK MONOSTABILNY(budzik/): | (U105) adres f0ff hex                                                                                                                                                                                                                                |
| PRZERZUTNIK D(startudzik/):        | (U104) adres f0fe hex                                                                                                                                                                                                                                |
|                                    |                                                                                                                                                                                                                                                      |



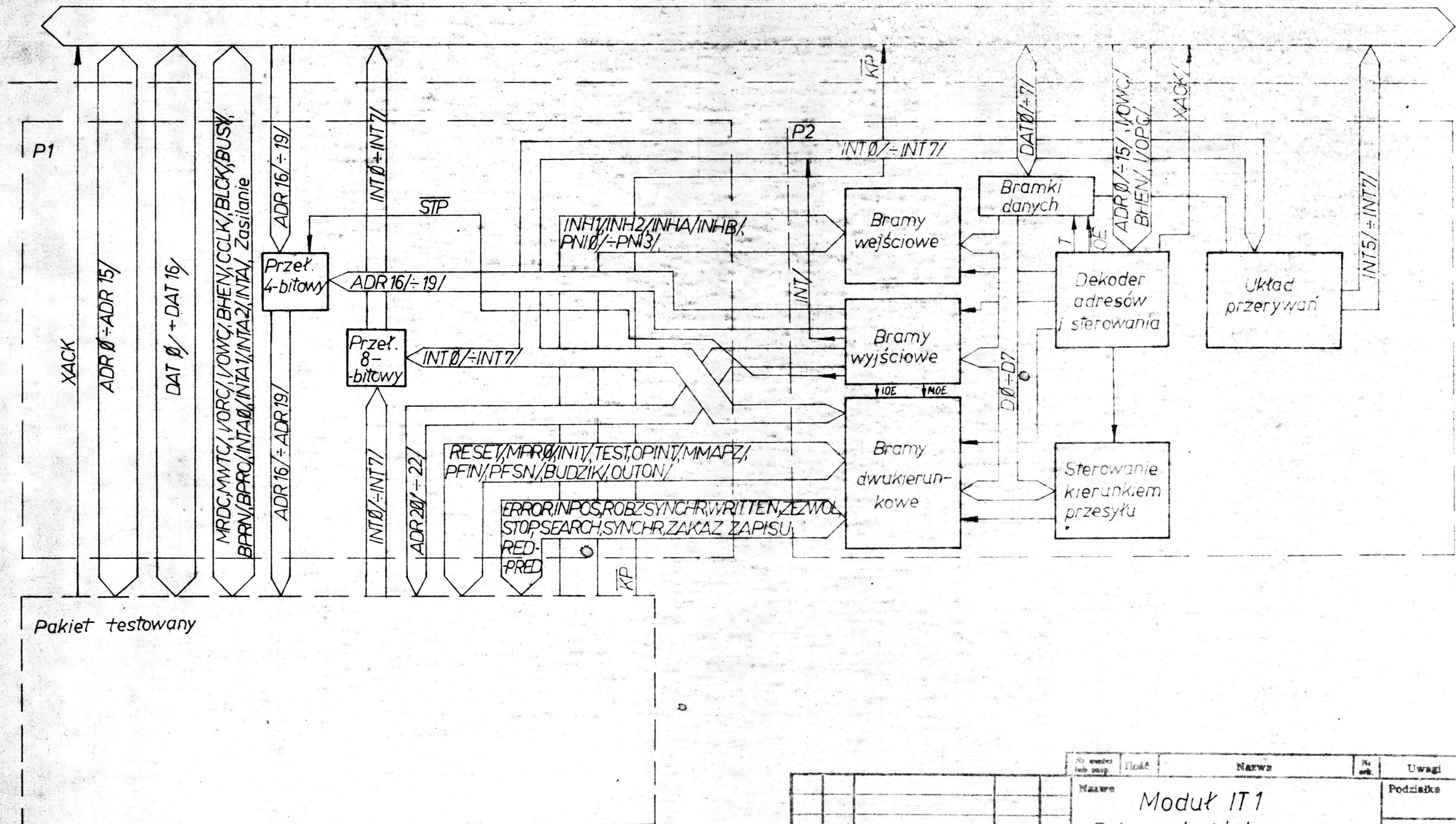
Rys. 1. Widok ogólny urządzenia testującego.



K1 ÷ K8 — kołki lutownicze

W nawiasach podano nazwy złącz dołączonych do złącz obiektowych pakietów testowanych

Rys. 2. Schemat połączeń złącz pakietu PO z pakietami urządzenia testującego



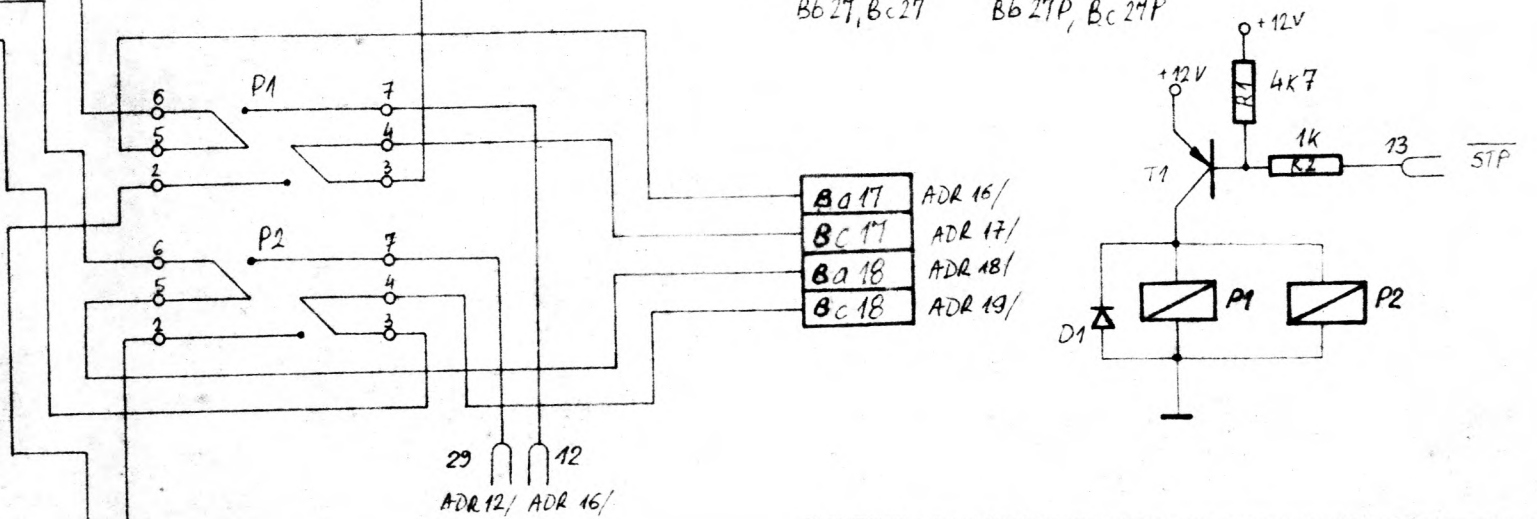
| № ewid. techn. zapis | Logo        | Nazwa                                               | № ark. | Uwagi                     |
|----------------------|-------------|-----------------------------------------------------|--------|---------------------------|
|                      |             | Nazwa                                               |        | Podpis                    |
|                      |             | Moduł IT1                                           |        |                           |
|                      |             | Schemat blokowy                                     |        | Ciężar                    |
| Chł. techn.          | Ind. techn. | Treść zmiany                                        | Podpis | Data                      |
| Projektował          |             | ABiernacki                                          | 8612   |                           |
| Konstruował          |             |                                                     |        |                           |
| Kreślił              |             | TSwat                                               | 8612   |                           |
| Sprawdził            |             | BBabicz                                             | 8612   |                           |
| Kier. Prac.          |             | MWrzesien                                           | 8612   |                           |
| Kier. Zakładu        |             | PJabłoński                                          | 8612   |                           |
|                      |             | Pracownia Inżynierii Automatyki i Pomiarów Warszawa |        |                           |
|                      |             | Zakład CAE                                          |        |                           |
|                      |             | Zastępuje rys. Nr                                   |        | Nr ark. 151               |
|                      |             | Zastąpiono przez rys. Nr                            |        | Nr rys. zast. ST-6-42-0-0 |
|                      |             | Nr rysunku                                          |        | Nr części 30              |
|                      |             |                                                     |        | Nr arch. 4668             |
|                      |             |                                                     |        |                           |

|         |       |        |
|---------|-------|--------|
| DAT 15/ | Bc 8  | Bc 8P  |
| DAT 14/ | Ba 8  | Ba 8P  |
| DAT 13/ | Bc 9  | Bc 9P  |
| DAT 12/ | Ba 9  | Ba 9P  |
| DAT 11/ | Bc 10 | Bc 10P |
| DAT 10/ | Ba 10 | Ba 10P |
| DAT 9/  | Bc 11 | Bc 11P |
| DAT 8/  | Ba 11 | Ba 11P |
| DAT 7/  | Ac 25 | Ac 25P |
| DAT 6/  | Aa 25 | Aa 25P |
| DAT 5/  | Ac 26 | Ac 26P |
| DAT 4/  | Aa 26 | Aa 26P |
| DAT 3/  | Ac 27 | Ac 27P |
| DAT 2/  | Aa 27 | Aa 27P |
| DAT 1/  | Ac 28 | Ac 28P |
| DAT 0/  | Aa 28 | Aa 28P |
| ADR 15/ | Ac 17 | Ac 17P |
| ADR 14/ | Aa 17 | Aa 17P |
| ADR 13/ | Ac 18 | Ac 18P |
| ADR 12/ | Aa 18 | Aa 18P |
| ADR 11/ | Ac 19 | Ac 19P |
| ADR 10/ | Aa 19 | Aa 19P |
| ADR 9/  | Ac 20 | Ac 20P |
| ADR 8/  | Aa 20 | Aa 20P |
| ADR 7/  | Ac 21 | Ac 21P |
| ADR 6/  | Aa 21 | Aa 21P |
| ADR 5/  | Ac 22 | Ac 22P |
| ADR 4/  | Aa 22 | Aa 22P |
| ADR 3/  | Ac 23 | Ac 23P |
| ADR 2/  | Aa 23 | Aa 23P |
| ADR 1/  | Ac 24 | Ac 24P |
| ADR 0/  | Aa 24 | Aa 24P |
| BHEN/   | Bc 14 | Bc 14P |
| MRDC/   | Aa 8  | Aa 8P  |
| MWTC/   | Ac 8  | Ac 8P  |
| I/ORC/  | Aa 9  | Aa 9P  |
| I/OWC/  | Ac 9  | Ac 9P  |
| XACK/   | Aa 10 | 49     |
| CCLK/   | Aa 12 | Aa 12P |
| BLCK    | Aa 5  | Aa 5P  |
| BUSY/   | Aa 7  | Aa 7P  |
| BPRN/   | Aa 6  | Aa 6P  |
| BPRO/   | Ac 6  | Ac 6P  |

|         |        |        |
|---------|--------|--------|
| INT-A0/ | Ba 21  | Ba 21P |
| INT-A1/ | Bc 21  | Bc 21P |
| INT-A2/ | Bc 19  | Bc 19P |
| INTA/   | Ba 19  | Ba 19P |
| INT 9/  | Ac 13  | Ac 13P |
| INT 6/  | Aa 13  | Aa 13P |
| INT 5/  | Ac 14  | Ac 14P |
| INT 4/  | Aa 14  | Aa 14P |
| INT 3/  | Ac 15  | Ac 15P |
| INT 2/  | Aa 15  | Aa 15P |
| INT 1/  | Ac 16  | Ac 16P |
| INT 0/  | Aa 16  | Aa 16P |
| ADR 16/ | Ba 17P |        |
| ADR 17/ | Bc 17P |        |
| ADR 18/ | Ba 18P |        |
| ADR 19/ | Bc 18P |        |
| XACKPT/ | 48     | Aa 10P |
| ADR 22/ | 38     | Bc 15P |
| ADR 21/ | 39     | Ba 15P |
| ADR 20/ | 40     | Ba 14P |
| PNI 0/  | 27     | Bb 8P  |
| PNI 1/  | 43     | Bb 9P  |
| PNI 2/  | 26     | Bb 10P |
| PNI 3/  | 42     | Bb 11P |
| MMAP2/  | 41     | Ba 12P |
| INH 1/  | 45     | Ac 10P |
| INH 2/  | 46     | Ac 11P |
| INH 3/  | 7      | Ba 23P |
| INH 4/  | 3      | Bc 23P |
| GND     | 47     | Bc 1P  |
|         |        | Ba 1P  |

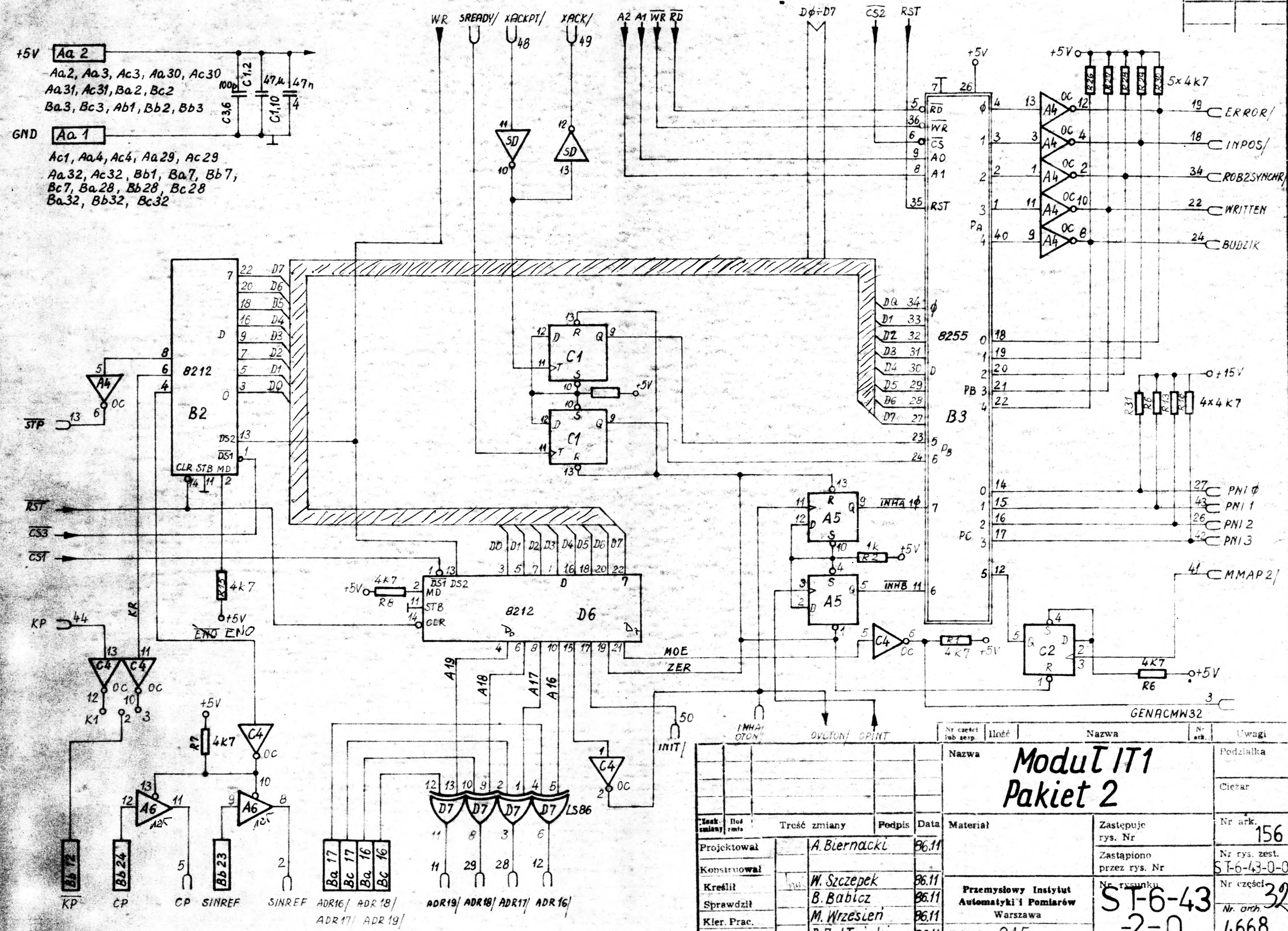
|              |    |        |
|--------------|----|--------|
| ERROR/       | 19 | Bb 20P |
| INPOS/       | 18 | Bb 21P |
| ROB2SYNCH/   | 34 | Bb 22P |
| WRITTEN/     | 22 | Bb 15P |
| WE 1Cφ       | 25 | Bb 12P |
| ZEZWOL       | 21 | Bb 16P |
| STOP         | 37 | Bb 17P |
| SEARCH       | 20 | Bb 18P |
| REDPRED      | 36 | Bb 19P |
| SYNCHR       | 7  | Bb 25P |
| ZAKAZ ZAPISU | 23 | Bb 14P |
| CP           | 5  | Bb 24P |
| SIN REF      | 2  | Bb 25P |
| RESET/       | 10 | Ba 26P |
| INIT/        | 50 | Ac 5P  |
| MPRO/        | 9  | Bc 25P |
| PFIN/        | 4  | Ba 24P |
| PFSN/        | 6  | Bc 24P |
| TEST/        | 35 | Ba 22P |
| BUDZIK       | 24 | Bb 13P |
| BTMO         | 8  | Ba 25A |

|      |                           |                                         |
|------|---------------------------|-----------------------------------------|
| +5V  | AB 1                      | AD 1P                                   |
|      | Aa 2, Ac 2, Aa 3          | Aa 2P, Ac 2P, Aa 3P, Ac 3P,             |
|      | Ac 3, Aa 30, Ac 30        | Aa 30P, Ac 30P, Aa 31P,                 |
|      | Aa 31, Ac 31, Ba 2        | Ac 31P, Ba 2P, Bb 2P, Bc 2P,            |
|      | Bb 2, Bc 2, Ba 3, Bb 3    | Ba 3P, Bb 3P, Bc 3P                     |
|      | Bc 3                      |                                         |
| GND  | Aa 1                      | Aa 1P                                   |
|      | Ac 1, Aa 4, Ac 4, Ba 29   | Ac 1P, Aa 4P, Ac 4P, Aa 29P, Ac 29P,    |
|      | Ac 29, Aa 32, Ac 32, Bb 1 | Aa 32P, Ac 32P, Bb 1P, Ba 7P, Bb 7P,    |
|      | Ba 7, Bb 7, Bc 7, Ba 28   | Ba 28P, Bb 28P, Bc 28P, Ba 32P, Bb 32P, |
|      | Bb 28, Bc 28, Ba 32       | Bc 32P                                  |
|      | Bb 32, Bc 32              |                                         |
| -5V  | Ba 31                     | Ba 31P                                  |
|      | Bb 31, Bc 31              | Bb 31P, Bc 31P                          |
| -12V | Ba 30                     | Ba 30P                                  |
|      | Bb 30, Bc 30              | Bb 30P, Bc 30P                          |
| -15V | Ba 29                     | Ba 29P                                  |
|      | Bb 29, Bc 29              | Bb 29P, Bc 29P                          |
| +15V | Ba 5                      | Ba 5P                                   |
|      | Bb 5, Bc 5                | Bb 5P, Bc 5P                            |
| +12V | Ba 4                      | Ba 4P                                   |
|      | Bb 4, Bc 4                | Bb 4P, Bc 4P                            |
| +5VB | Ba 27                     | Ba 27P                                  |
|      | Bb 27, Bc 27              | Bb 27P, Bc 27P                          |

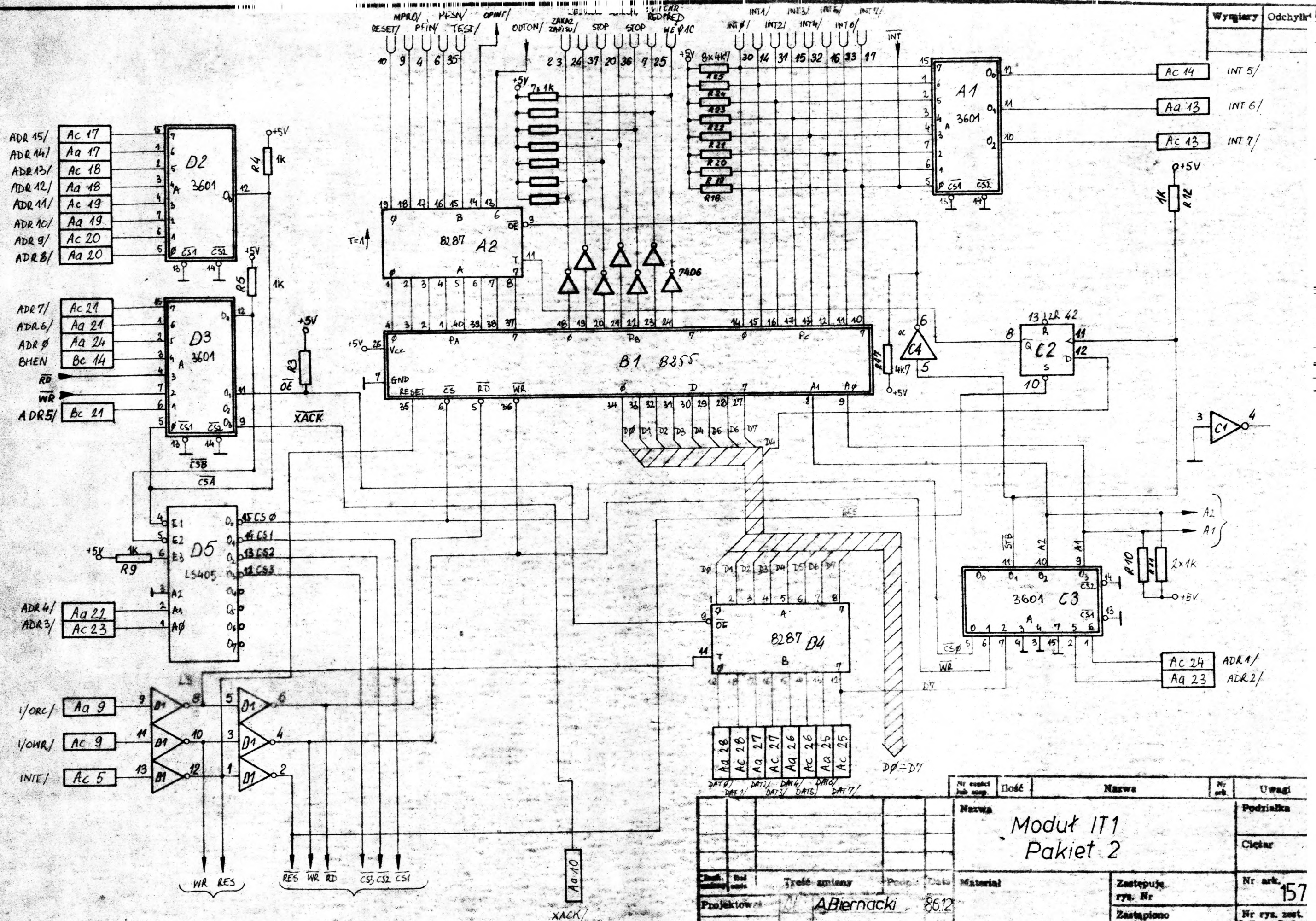


| Nr. części                                          | Flak      | Nazwa          | Nr. ark.                   | Uwagi |
|-----------------------------------------------------|-----------|----------------|----------------------------|-------|
| Moduł IT 1                                          |           |                |                            |       |
| Pakiet 1                                            |           |                |                            |       |
| Nazwa                                               | Podziałka |                |                            |       |
| Ciepota                                             |           |                |                            |       |
| Znak zmiany                                         | Flak      | Treść zmiany   | Podpis                     | Data  |
| Projektował                                         |           | ABiernacki     |                            | 8612  |
| Konstruował                                         |           | T. Swat        |                            | 8612  |
| Kreślił                                             |           | B. Babicz      |                            | 8612  |
| Sprawdził                                           |           | M. Wrzesień    |                            | 8612  |
| Kier. Prac.                                         |           | P. Jabłonowski |                            | 8612  |
| Kier. Zakładu                                       |           |                |                            |       |
| Zastępuje rys. Nr                                   |           |                | Nr. ark. 152               |       |
| Zastąpiono przez rys. Nr                            |           |                | Nr. rys. zest. ST-6-42-0-0 |       |
| Przemysłowy Instytut Automatyki i Pomiarów Warszawa |           |                | Nr. rysunku                |       |
| Zakład OAE                                          |           |                | Nr. części                 |       |
|                                                     |           |                | Nr. arch. 4668             |       |

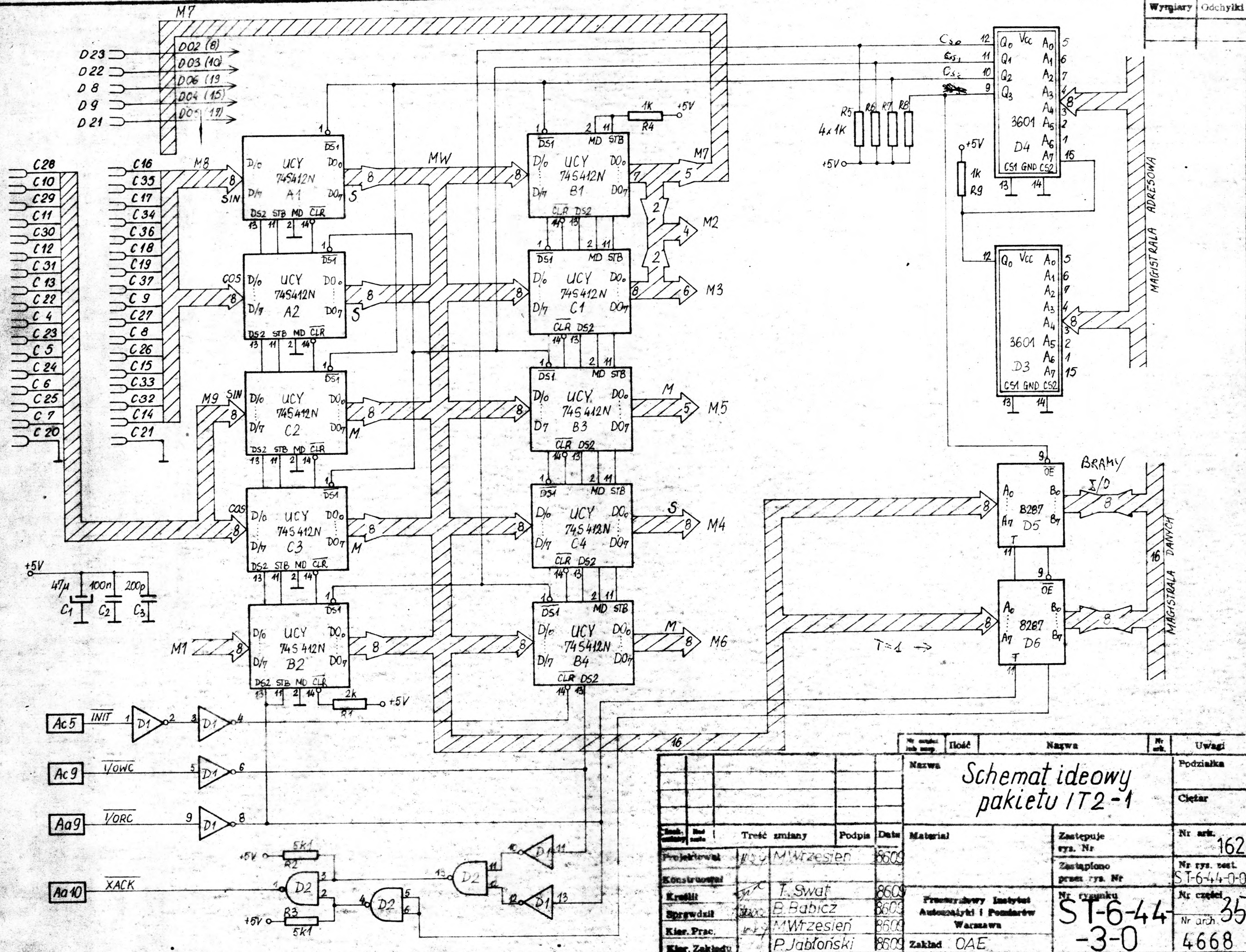
| Wymiary | Odchyłki |
|---------|----------|
|         |          |

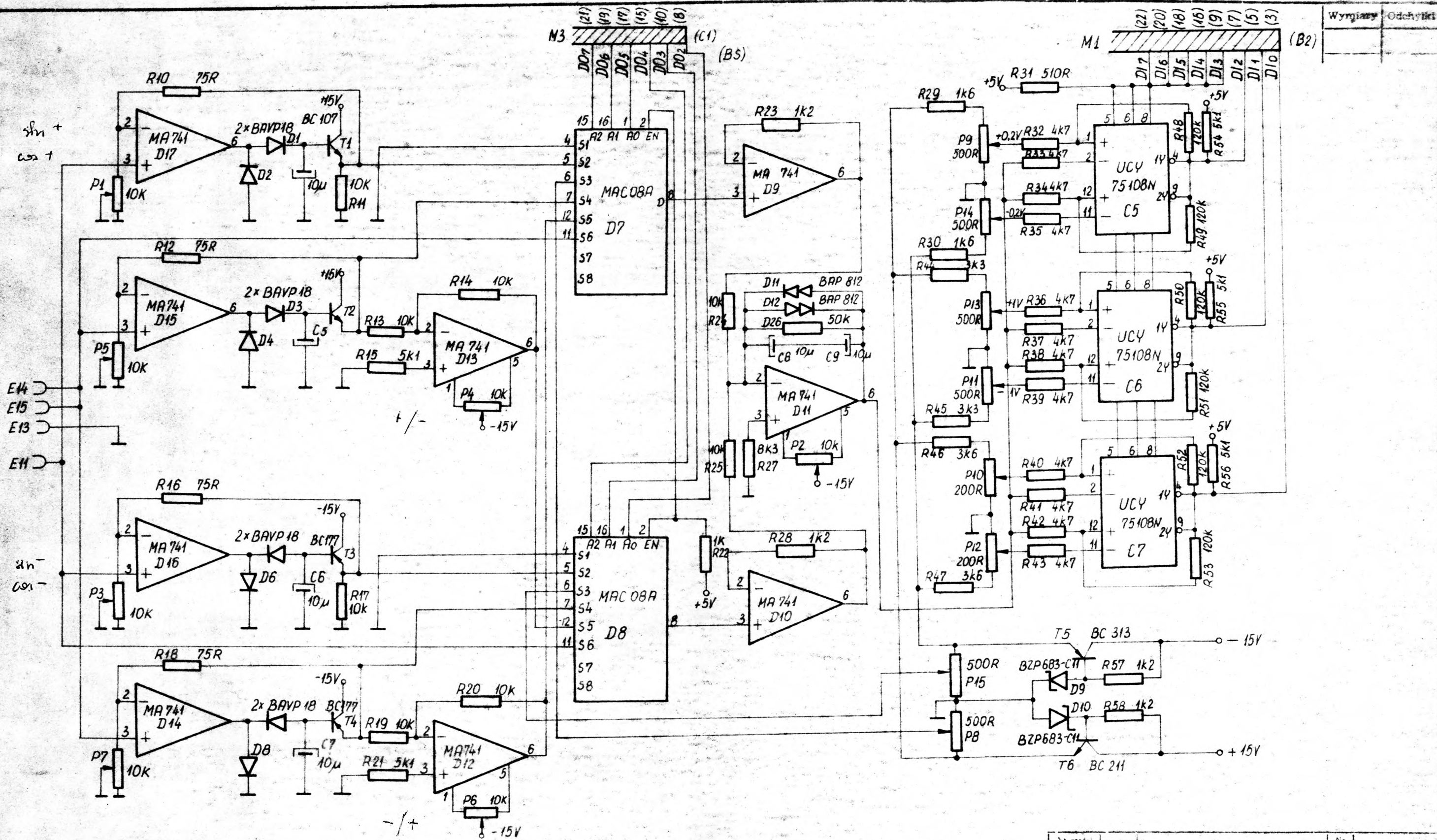


| Nazwa                                               |              | Nr. ark.                  | Uwagi |
|-----------------------------------------------------|--------------|---------------------------|-------|
| Moduł IT1                                           |              |                           |       |
| Pakiet 2                                            |              |                           |       |
| Treść zmiany                                        |              | Podpis                    | Data  |
| Projektował                                         | A. Biernacki |                           | 86.11 |
| Konstruował                                         |              |                           |       |
| Kreślił                                             | W. Szczepke  |                           | 86.11 |
| Sprawił                                             | B. Babicz    |                           | 86.11 |
| Kier. Prac.                                         | M. Wrzesień  |                           | 86.11 |
| Kier. Zakładu                                       | P. Jabłoński |                           | 86.11 |
| Zastępuje rys. Nr                                   |              | Nr. ark. 156              |       |
| Zastąpiono przez rys. Nr                            |              | Nr rys. zest. ST-6-43-0-0 |       |
| Przemysłowy Instytut Automatyki i Pomiarów Warszawa |              | Nr. rysunku ST-6-43-2-0   |       |
| Zakład OAE                                          |              | Nr. arch. 4668            |       |

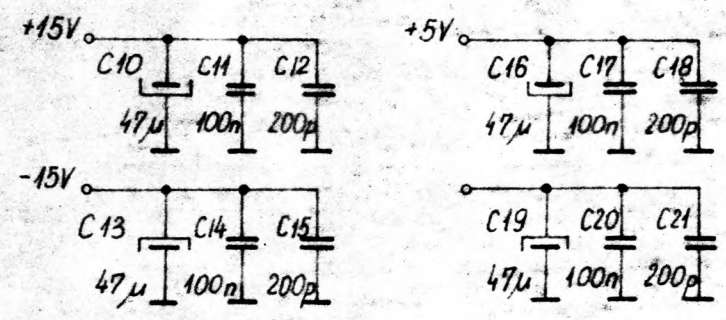


| Nr części lub ozn.    |  | Ilość | Nazwa                    | Nr ark. | Uwagi |
|-----------------------|--|-------|--------------------------|---------|-------|
| Moduł IT1<br>Pakiet 2 |  |       |                          |         |       |
| Projektant            |  |       | Zastępuje rys. Nr        |         |       |
| Konstruktor           |  |       | Zastąpiono przez rys. Nr |         |       |
| Kreślił               |  |       | Nr rysunku               |         |       |
| Sprawdził             |  |       | Nr części                |         |       |
| Kier. Prac.           |  |       | Nr arch.                 |         |       |
| Kier. Zakładu         |  |       | 4668                     |         |       |

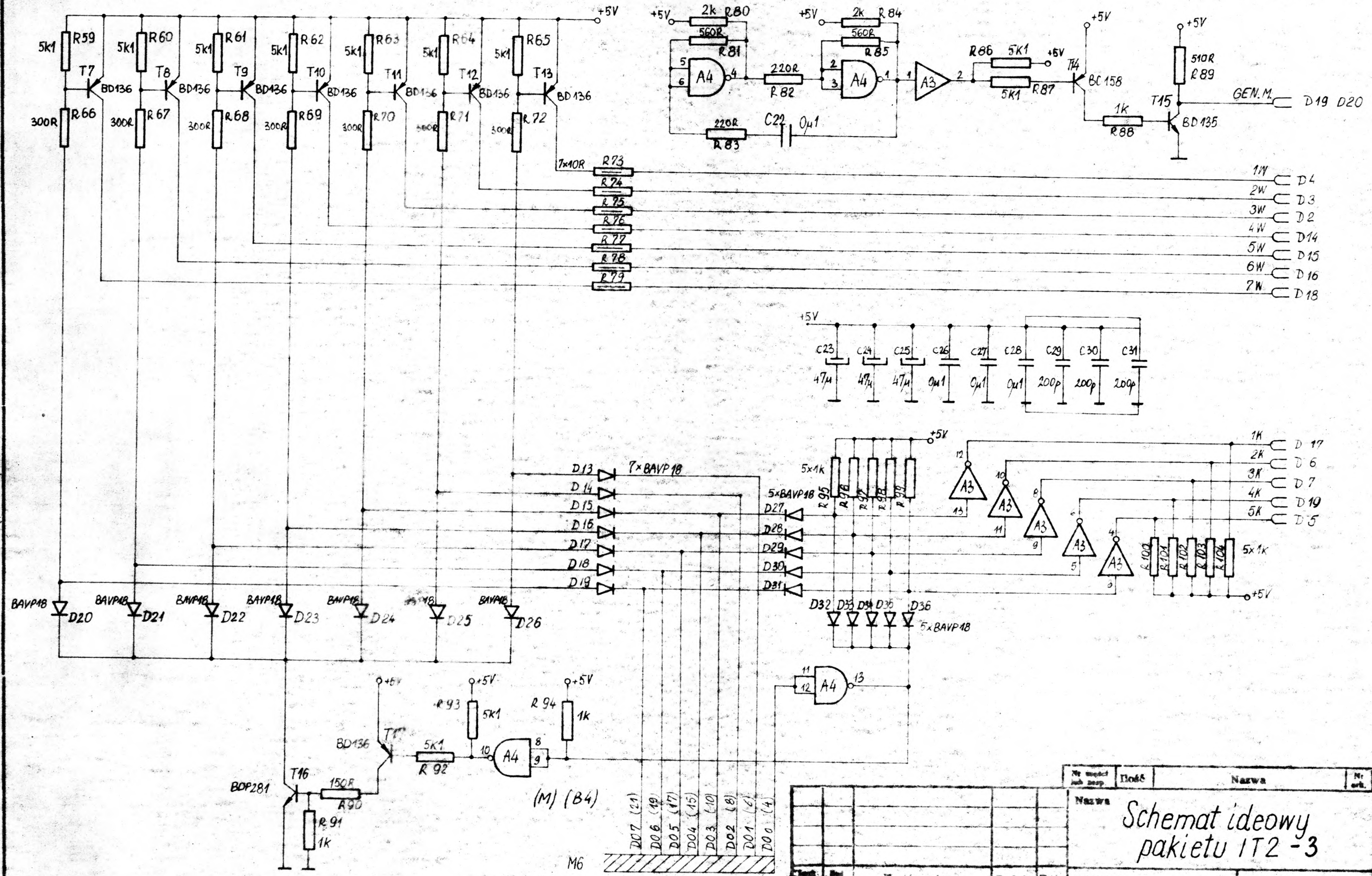




| Wygląd | Odczyt |
|--------|--------|
|        |        |

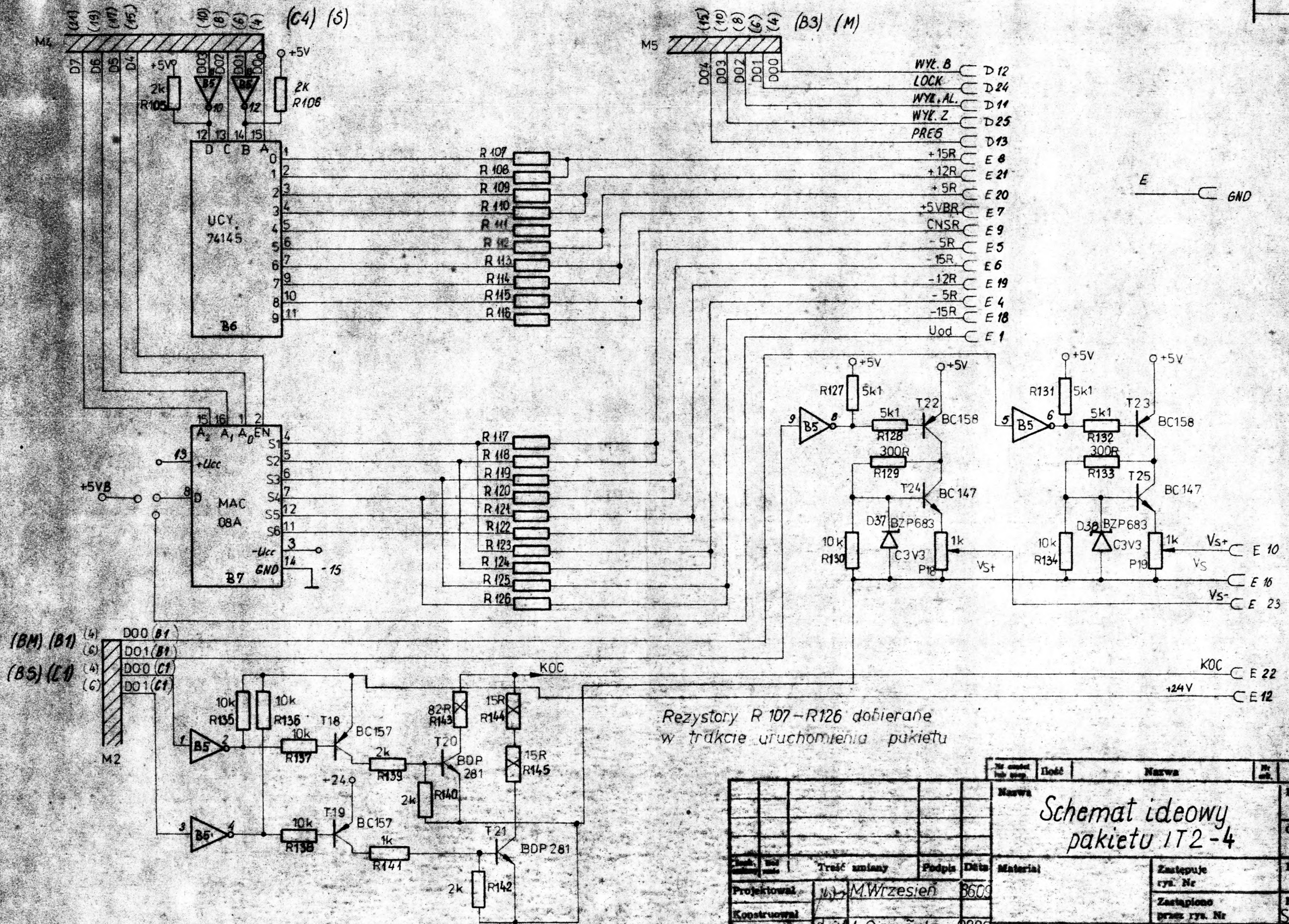


| Nr części lub zespołu |  | Ilość       |  | Nazwa        |  | Nr ark.                         |  | Uwagi                                               |  |
|-----------------------|--|-------------|--|--------------|--|---------------------------------|--|-----------------------------------------------------|--|
|                       |  |             |  |              |  | Nazwa                           |  | Podziałka                                           |  |
|                       |  |             |  |              |  | Schemat ideowy<br>pakietu IT2-2 |  | Ciężar                                              |  |
| Zest. zmiany          |  | Ilość zmian |  | Treść zmiany |  | Podpis                          |  | Data                                                |  |
| Projektował           |  |             |  | M. Wrzesień  |  | 88.10                           |  | Materiał                                            |  |
| Konstruował           |  |             |  |              |  |                                 |  | Zastępuje rys. Nr                                   |  |
| Kreślił               |  | 31/10       |  | B. Lipiecka  |  | 88.10                           |  | Zastąpiono przez rys. Nr                            |  |
| Sprawdził             |  |             |  |              |  |                                 |  | Nr rysunku                                          |  |
| Kier. Prac.           |  |             |  | M. Wrzesień  |  | 88.10                           |  | Przemysłowy Instytut Automatyki i Pomiarów Warszawa |  |
| Kier. Zakładu         |  |             |  | P. Jablonski |  | 88.10                           |  | Zakład OAE                                          |  |
|                       |  |             |  |              |  |                                 |  | ST-6-44-4-0                                         |  |
|                       |  |             |  |              |  |                                 |  | Nr części 36                                        |  |
|                       |  |             |  |              |  |                                 |  | Nr arch. 4668                                       |  |



| Nr. uwag      |  | Ilość        |  | Nazwa          |  | Nr. ark.                 |  | Uwagi         |  |
|---------------|--|--------------|--|----------------|--|--------------------------|--|---------------|--|
|               |  |              |  | Nazwa          |  | Podziałka                |  |               |  |
|               |  |              |  | Schemat ideowy |  | Ciepła                   |  |               |  |
|               |  |              |  | pakietu IT2-3  |  |                          |  |               |  |
| Projektował   |  | M. Wrzesień  |  | 8609           |  | Zastępuje rys. Nr        |  | Nr ark.       |  |
| Konstruował   |  | T. Swat      |  | 8609           |  | Zastąpiono przez rys. Nr |  | 164           |  |
| Kreślił       |  | M. Wrzesień  |  | 8609           |  | Nr rysunku               |  | Nr rys. zost. |  |
| Sprawdził     |  | M. Wrzesień  |  | 8609           |  | ST-6-44                  |  | Nr części     |  |
| Kier. Prac.   |  | P. Jabłoński |  | 8609           |  | -5-0                     |  | Nr arch.      |  |
| Kier. Zakładu |  |              |  |                |  | Zakład OAE               |  | 4668          |  |

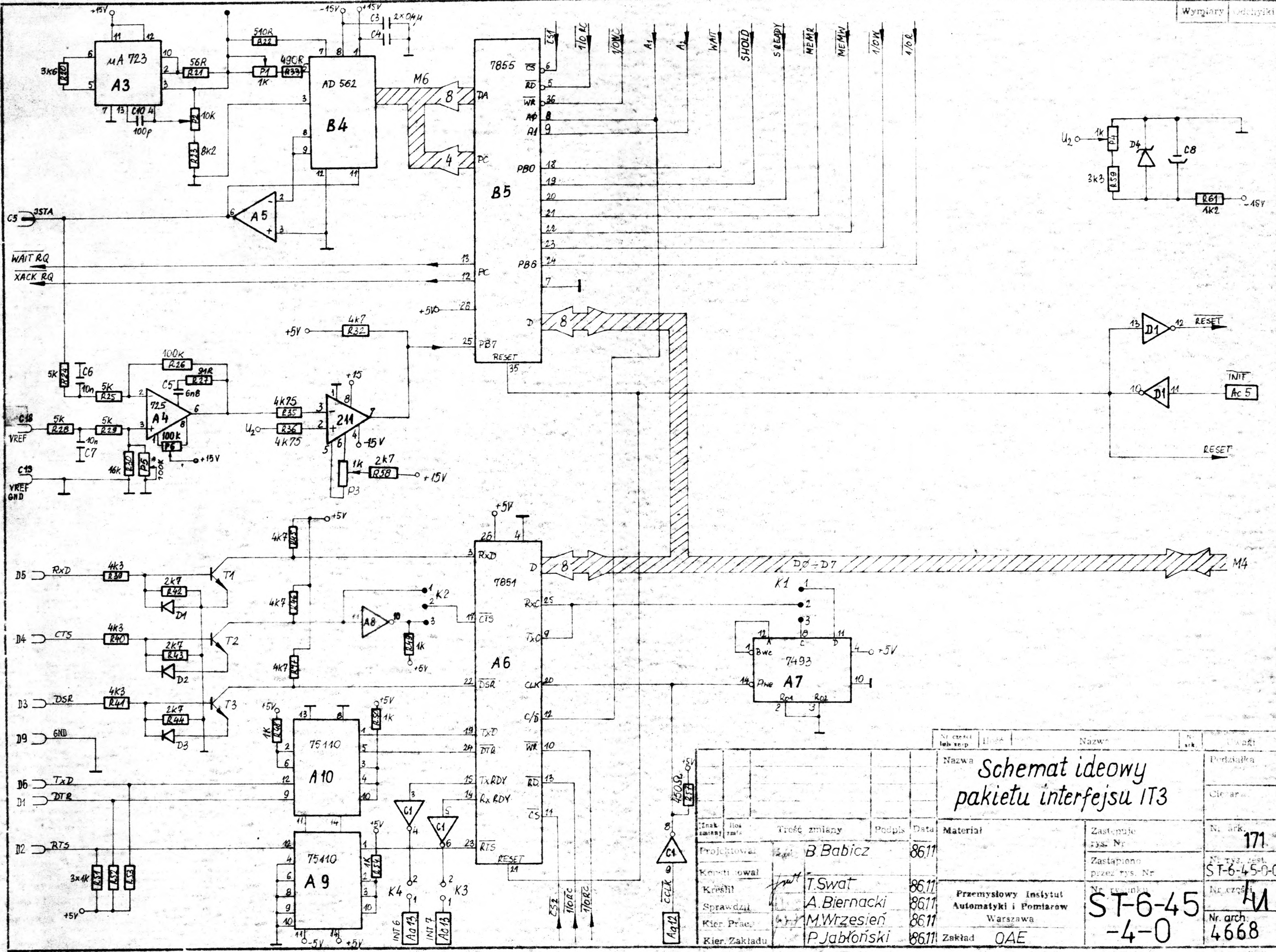
| Wymiary | Odchył |
|---------|--------|
|         |        |



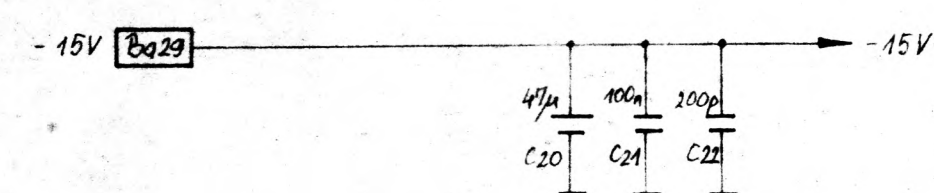
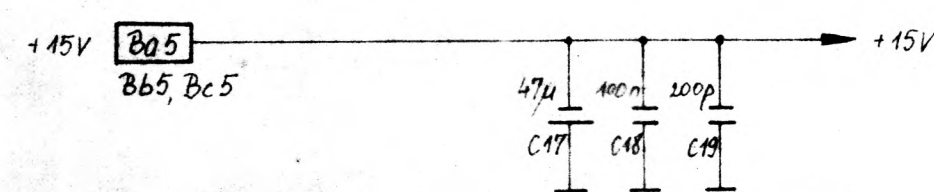
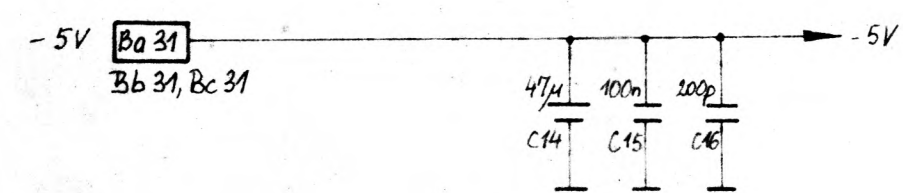
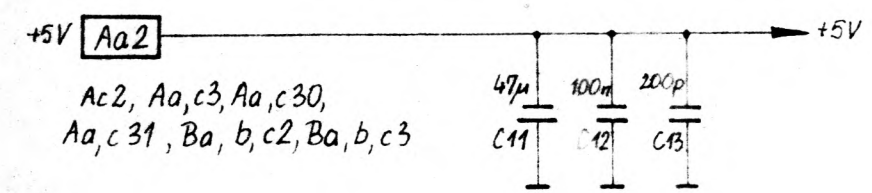
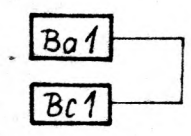
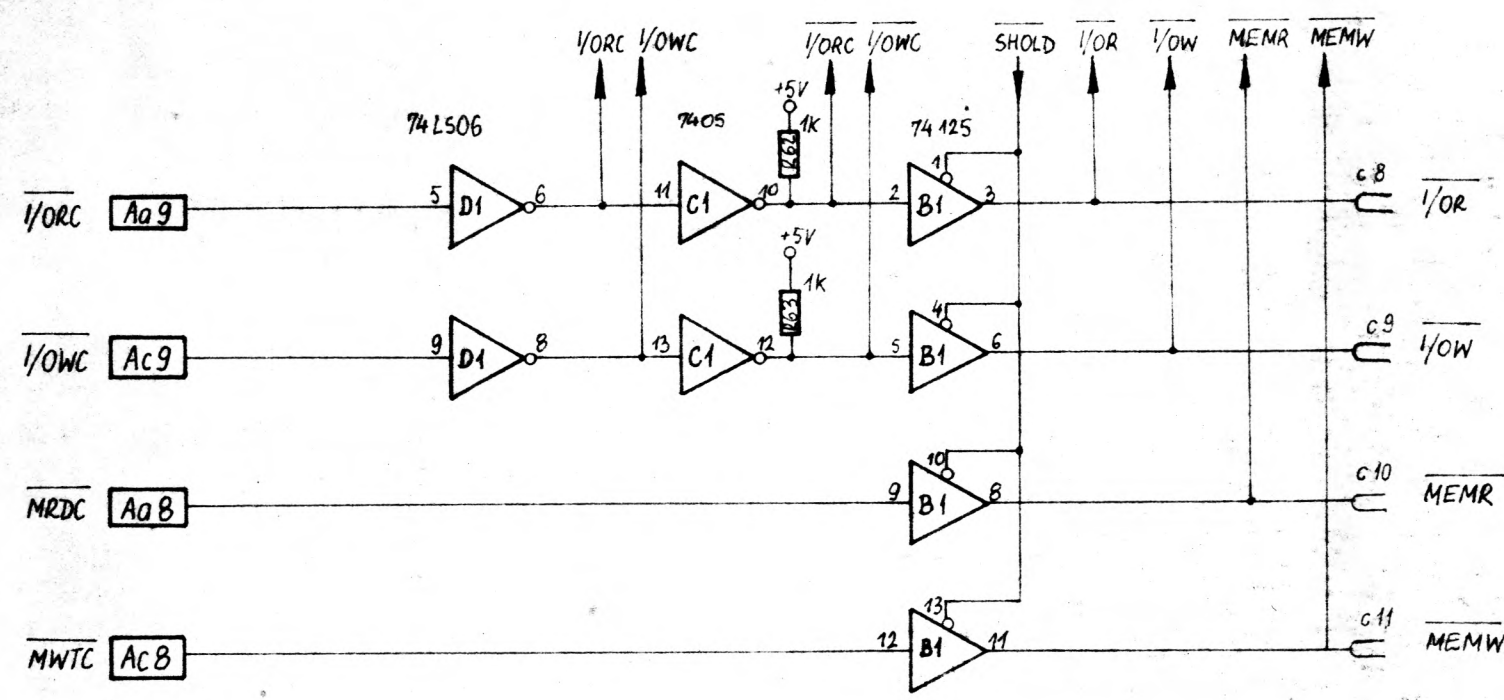
Rezystory R 107-R126 dobierane w trakcie uruchomienia pakietu

| Nazwa                        |              |      |  | Uwagi        |            |
|------------------------------|--------------|------|--|--------------|------------|
| Nazwa                        |              |      |  | Podziałka    |            |
| Schemat ideowy pakietu IT2-4 |              |      |  | Ciepota      |            |
| Projektował                  | M. Wrzesień  | 3609 |  | Nr ark.      | 165        |
| Konstruował                  |              |      |  | Nr rys. aut. | ST-6-44-0- |
| Kreślił                      | L. Orzyńska  | 0886 |  | Nr rysunku   | 32         |
| Sprawił                      | B. Babicz    | 8600 |  | Nr arch.     | 4668       |
| Kier. Prac.                  | M. Wrzesień  | 8600 |  |              |            |
| Kier. Zakładu                | P. Jabłoński | 8609 |  |              |            |
| Zakład OAE                   |              |      |  | ST-6-44-6-0  |            |





| Nazwa                    |  | Nazwa                  |  | Nazwa          |  |
|--------------------------|--|------------------------|--|----------------|--|
| Schemat ideowy           |  | pakietu interfejsu IT3 |  |                |  |
| Projektował              |  | B. Babicz              |  | 86.11          |  |
| Kreślił                  |  | T. Swat                |  | 86.11          |  |
| Sprawdził                |  | A. Biernacki           |  | 86.11          |  |
| Kier. Prac.              |  | M. Wrzesień            |  | 86.11          |  |
| Kier. Zakładu            |  | P. Jabłoński           |  | 86.11          |  |
| Zakład                   |  | OAE                    |  |                |  |
| Zastępuje rys. Nr        |  |                        |  | Nr. ark.       |  |
| Zastąpiono przez rys. Nr |  |                        |  | Nr. rys. test. |  |
| Nr. rysunku              |  | ST-6-45                |  | Nr. części     |  |
| -4-0                     |  |                        |  | Nr. arch.      |  |
|                          |  | 4668                   |  |                |  |



| Nr. części lub zast. |        | Nazwa                                 |        | Nr. ark. | Liczba                                              |                |
|----------------------|--------|---------------------------------------|--------|----------|-----------------------------------------------------|----------------|
|                      |        | Nazwa                                 |        |          | Podziałka                                           |                |
|                      |        | Schemat ideowy pakietu interfejsu IT3 |        |          | Ciepota                                             |                |
| Wzrost               | Wzrost | Treść zmiany                          | Podpis | Data     | Material                                            | Nr. ark.       |
| Projektował          |        | B. Babicz                             |        | 86.11    |                                                     | 172            |
| Konstruował          |        |                                       |        |          | Zastępca rys. Nr                                    | Nr. rys. zest. |
| Kreślił              |        | T. Swat                               |        | 86.11    | Zastąpiono przez rys. Nr                            | ST-6-45-0      |
| Sprawdził            |        | A. Biernacki                          |        | 86.11    |                                                     | Nr. części     |
| Kier. Prac.          |        | M. Wrzesień                           |        | 86.11    |                                                     | Nr. arch.      |
| Kier. Zakładu        |        | P. Jabłoński                          |        | 86.11    | Zakład OAE                                          | 4668           |
|                      |        |                                       |        |          | Przemysłowy Instytut Automatyki i Pomiarów Warszawa | ST-6-45-0-5-0  |

ZALACZNIK 1

PROGRAM MONITOR PAKIETU MM86

## SPIS TRESCI

|       |                             |    |
|-------|-----------------------------|----|
| 1.    | Wstęp                       | 2  |
| 2.    | Struktura dyrektów MONITORA | 2  |
| 3.    | Dyrektory MONITORA          | 3  |
| 3.1.  | Dyrektora S                 | 3  |
| 3.2.  | Dyrektora X                 | 4  |
| 3.3.  | Dyrektora D                 | 5  |
| 3.4.  | Dyrektora M                 | 6  |
| 3.5.  | Dyrektora I                 | 7  |
| 3.6.  | Dyrektora O                 | 7  |
| 3.7.  | Dyrektora W                 | 2  |
| 3.8.  | Dyrektora R                 | 9  |
| 3.9.  | Dyrektora V                 | 10 |
| 3.10. | Dyrektora K                 | 11 |
| 3.11. | Dyrektora F                 | 11 |
| 3.12. | Dyrektora G                 | 12 |
| 3.13. | Dyrektora N                 | 13 |
| 3.14. | Dyrektora U                 | 14 |
| 3.15. | Dyrektora H                 | 15 |
| 3.16. | Dyrektora L                 | 16 |

## 1. WSTEP

Program MONITOR OPERATORSKI jest przeznaczony do pierwszego etapu badan urzadzen i zestawow z pakietem Jednostki centralnej 16-bitowej MM86, oraz do pierwszego etapu prac nad opracowaniem programow uzytkowych i testowych.

Program MONITOR jest przez operatora obsluziwany z urzadzenia dolaczonego do kanalu V-24 pakietu Jednostki centralnej. Urzadzeniem tym moze byc monitor ekranowy, drukarka DZM 180-KSR lub minikomputer SM-4 (po opracowaniu dla tego minikomputera programu sprzesajacego z pakietem MM86).

Program MONITOR zajmuje obszar 4K pamieci PROM pakietu Jednostki centralnej o adresach od FF000H do FFFFFH. Ponadto program wykorzystuje na komorki robocze i stos 256 bajtow pamieci RAM od 00100H do 001FFH. Oprócz tego program MONITOR inicjuje dwa wektory przerwan:

przerwanie nr 1 - Single Step - uzywane w dyrektywach N i U

przerwanie nr 3 - Breakpoint - uzywane w dyrektywach G i U

Po wlaczeniu zasilania lub po naciśnięciu przycisku RESET program przeprowadza inicjację układów programowalnych, wpisuje do wszystkich rejestrów użytkownika zera (z wyjątkiem rejestru SP, którego wartość początkowa jest ustawiana na 300H) i oczekuje na wybór przez operatora jednej z dyrektyw.

## 2. STRUKTURA DYREKTYW MONITORA

Program MONITOR sygnalizuje gotowosc do przyjecia dyrektywy wyswietlajac znak (>) na ekranie monitora ekranowego. W tym momencie operator podaje nazwe dyrektywy z ewentualnymi argumentami. Kiedy wymagana liczba argumentow jest wieksza od 1, musza one byc rozdzielone przecinkami. Wywołanie dyrektywy musi byc zakonczone, zaleznie od rodzaju dyrektywy, znakiem powrotu-karetki (CR) lub przecinka (,). W danym momencie moze byc wykonywana tylko jedna dyrektywa, a operator moze wywolowac dyrektywy po jednej w danej linii.

Wszystkie argumenty (z wyjątkiem argumentow dyrektywy X) sa liczbami heksadecymalnymi: dla bajtow z zakresu 00 do FF, a dla slow z zakresu 0000 do FFFF. Zera nieznaczace moga byc pomijane. Jezeli operator poda wiecej niz dwa znaki (dla bajtu) lub wiecej niz cztery (dla slowa) tylko ostatnie dwa lub cztery znaki zostana przyjete przez program MONITOR. Argument bedacy adresem sklada sie z wartosci segmentu i przesuniecie (offset). Jezeli segment adresu nie jest podawany jako jego wartosc, przyjmowana jest zawartosc rejestru CS uzytkownika. Jezeli oba skladniki adresu sa podawane to nalezy je podac w kolejnosci najpierw segment pozniej przesuniecie oddzielone znakiem dwukropka (:). Gdy dyrektywa wymaga dwuch argumentow stanowiacych adres poczatku i konca obszaru pamieci to segment mozna podac tylko przy adresie poczatku.

Wszedzie tam, gdzie argumentem jest slowo, mozna podac,

zamiast liczby heksadecymalnej, nazwę rejestru. Jako argument zostanie wtedy przyjęta zawartość podanego rejestru użytkownika.

Podanie błędnej nazwy dyrektywy lub błędnego argumentu powoduje zakończenie przyjmowania dyrektywy, wyświetlenie znaku zapytania (?) i zgłoszenie gotowości do przyjmowania dyrektyw znakiem ()).

### 3. DYREKTYWY MONITORA

-----

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

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

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

#### 3.1 DYREKTYWA S - odczyt i zmiana zawartości pamięci

-----

SIWJ<A>,[<D>],J\*<CR>

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

Po wywołaniu dyrektywy litera S lub SW należy podać adres komórki pamięci, która ma być odczytana. Jeżeli adres segmentu nie zostanie podany to do obliczenia adresu efektywnego będzie użyta zawartość rejestru CS użytkownika. Wprowadzany adres należy zakończyć znakiem przecinka. W odpowiedzi zostanie wyświetlona zawartość podanego obszaru pamięci (poprzedzona offsetem adresu), po której zostaje wyświetlony znak myślnika (-) sygnalizujący oczekiwanie na znak z klawiatury. Jeśli chcemy zmieniać zawartość pamięci należy podać nową jej wartość.

Jeżeli odczytywana (i ewentualnie zmieniana) ma być tylko jedna komórka pamięci należy zakończyć dyrektywę znakiem powrotu karetki <CR>. Jeżeli odczytywane (i ewentualnie zmieniane) mają być kolejne komórki pamięci należy kończyć wiersz dyrektywy znakiem przecinka (,) w celu wyświetlenia zawartości kolejnej komórki pamięci (w dyrektywie S) lub zawartości dwóch kolejnych komórek pamięci (w dyrektywie SW). Podobnie zakończenie wiersza dyrektywy znakiem minusa (-)

powoduje wyświetlenie zawartości poprzedniej komórki pamięci (w dyrektywie S) lub zawartości dwóch poprzednich komórek pamięci (w dyrektywie SW) z możliwością zmiany ich zawartości. Znak powrotu karetki <CR> kończy dyrektywę.

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

#### BLEDY:

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

#### PRZYKŁADY:

1. Sprawdzenie zawartości pamięci o adresie FF000H:

```
M86>SFF00:0,
0000 75-<CR>
M86>
```

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

```
M86>SDS:100,
0100 00-11,
0101 22-<CR>
M86>
```

3. Sprawdzenie zawartości wierzchołka stosu:

```
M86>SWSS:SP,
02FE 2323-<CR>
M86>
```

4. Zmiana zawartości słowa umieszczonego w pamięci RAM pod adresem 200H relatywnie do rejestru CS na wartość 1234H i dwóch poprzednich słów na wartości 0FFFFH:

```
M86>SW200H,
0200 2333-1234-
01FE C0CC-FFFF-
01FC 3373-FFFF<CR>
M86>
```

### 3.2 DYREKTYWA X - odczyt i zmiana zawartości rejestru

---

**XE<R>III<D>I,I\*<CR>**

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

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

W celu wyświetlenia i opcjonalnie zmiany zawartości jednego rejestru należy po literze X wywołującej dyrektywę napisać skróty nazwy tego rejestru jako jedną z nazw: AX, BX, CX, DX, SP, BP, SI, DI, CS, DS, SS, ES, IP, FL i zakończyć wiersz dyrektywy znakiem powrotu karetki <CR>. Na monitorze zostanie

AM

wyświetlony skrótu nazwy rejestru, znak równości (=), aktualna zawartość rejestru, a po niej znak myślnika (-) sygnalizujący oczekiwanie na nowe dane.

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

Gdy nie chcemy zmieniać zawartości rejestru należy zakończyć wiersz dyrektywy znakiem powrotu karetki (CR).

#### BŁĘDY:

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

#### PRZYKŁADY:

1. Wyświetlenie zawartości wszystkich rejestrów procesora:

```
M86>X<CR>
AX=0000 BX=1111 CX=2222 DX=3333 SP=0300 BP=0000 SI=0100
DI=0200 CS=FF00 DS=0000 SS=0000 ES=1000 IP=0356 FL=F002
M86>
```

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

```
M86>XCS<CR>
CS=FF00-0000,
DS=0000,
SS=0000<CR>
M86>
```

### 3.3 DYREKTYWA D - odczyt zawartości pamięci

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

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

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

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

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

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

#### BLEDY:

- podanie adresu konca mniejszego niz adres poczatku - sygnalizowane znakiem zapytania (?)
- uzycie w adresie znaku nie bedacego cyfra heksadecymalna - sygnalizowane znakiem zapytania (?)
- uzycie znaku roznego od przecinka (,) do rozdzielienia adresow poczatku i konca obszaru pamieci - sygnalizowane znakiem zapytania (?)

#### PRZYKLADY:

1. Wswietlenie zawartosci pamieci od adresu 101H do adresu 115H relatywnie do rejestru DS:

```
M86>DDS:101,115<CR>
```

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

```
M86>
```

2. Wswietlenie zawartosci komorki pamieci o adresie 1000H relatywnie do rejestru CS:

```
M86>D1000<CR>
```

```
1000 15
```

```
M86>
```

#### 3.4 DYREKTYWA M - przesuniecie zawartosci pamieci

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

Dyrektywa M sluzi do przesuwania zawartosci pamieci z jednego obszaru na inny. Dyrektywa ta moze tez sluzyc do wpisywania stalej wartosci do obszaru pamieci.

Obszar pamieci, z ktorego dane sa przesuwane jest deklarowany poprzez adresy poczatku i konca rozdzielone znakiem przecinka (,).

Podobnie jak w dyrektywie D adres konca przesuwanego obszaru jest ustalany relatywnie do tego samego segmentu jak adres poczatku. Zatem najwiekszy przesuwany obszar pamieci moze miec 64k bajtow. Adres przeznaczenia (oddzielony od adresu konca znakiem przecinka (,)) moze zawierac dowolna wartosc segmentu. Gdy segment adresu przeznaczenia nie bedzie podany jego wartosc zostaje przyjeta taka jak dla adresu poczatku.

Jezeli adres przeznaczenia jest o jeden wiekszy od adresu poczatku to pamiec osraniczona adresem poczatku i adresem konca +1 zostanie zapelniona zawartoscia komorki okreslonej przez adres poczatku.

#### BLEDY:

- uzycie w adresie znaku nie bedacego cyfra heksadecymalna - sygnalizowane znakiem zapytania (?)
- podanie adresu konca mniejszego od adresu poczatku - sygnalizowane znakiem zapytania (?)
- uzycie znaku roznego od przecinka (,) dla rozdzielenia poszczegolnych adresow - sygnalizowane znakiem zapytania (?)
- proba przesuniecie danych do pamieci typu PROM lub do pamieci nieistniejacej - sygnalizowane znakiem wykzyknika (!) poprzedzonym offsetem adresu pierwszej komorki pamieci, dla ktorej stwierdzono blad.

#### PRZYKŁADY:

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

```
M86>M100,130,DS:500<CR>
```

```
M86>
```

2. Wpis wartości 0FFH do obszaru pamięci od adresu 300H do 3FFH relatywnie do zawartości rejestru BX:

```
M86>SBX:300,
```

```
0300 10-FF<CR>
```

```
M86>MBX:300,3FE,301<CR>
```

```
M86>
```

#### 3.5. DYREKTYWA I - czytanie bramy wejściowej

---

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

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

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

#### BLEDY:

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

#### PRZYKŁADY:

1. Odczyt słowa z bramy o adresie 1717H:

```
M86>IW1717,
```

```
2200<CR>
```

```
M86>
```

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

```
M86>IDA,
```

```
FF,
```

```
F5,
```

```
FF<CR>
```

```
M86>
```

#### 3.6. DYREKTYWA O - wysłanie danych do bramy wyjściowej

---

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

Dyrektywa O wysyłania danych do bramy wyjściowej służy do wysłania bajtu (dyrektywa O) lub słowa (dyrektywa OW) z klawiatury monitora na podaną bramę wyjściową. Ponieważ przestrzeń adresowa bram wyjściowych jest ograniczona do 64K,

- adres bramy podaje się jako liczbę heksadecymalną z zakresu 0-0FFFFH bez podawania wartości segmentu.

Po podaniu adresu bramy wyjściowej należy nacisnąć znak przecinka (,), a po nim podać dane do wysłania. Następnie należy podać znak powrotu karetki (CR) w celu wysłania danych i zakończenia dyrektywy lub znak przecinka (,) w celu wysłania danych i oczekiwania na podanie z klawiatury nowych danych do tej samej bramy wyjściowej. Program oczekuje na nowe dane w nowym wierszu po znaku myślnika (-). Kolejne dane można wysyłać do bramy wyjściowej przez podanie tych danych zakończonych znakiem przecinka (,). Znak powrotu karetki (CR) po nowych danych kończy dyrektywę.

#### BLEDY:

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

#### PRZYKŁADY:

1. Wysłanie bajtu do bramy o adresie 0D0H:  
M86)0D0,80(CR)  
M86)
2. Wysłanie kolejnych słów do bramy o adresie 207H:  
M86)0W207,BAEC,  
-FF00,  
-00FF(CR)  
M86)

### 3.7. DYREKTYWA W - wyprowadzenie zawartości pamięci na

tasme papierowa

WEX] (AP), (AK) [, (AS)] (CR)

Dyrektywa W służy do wyprowadzenia zawartości zadeklarowanego obszaru pamięci na taśmie papierową w formacie 8086 lub 8080 (zgodnie z Hexadecimal Object File Format).

Po wywołaniu dyrektywy (litera W dla formatu 8086 lub WX dla formatu 8080) operator podaje adres początku i końca (oddzielone przecinkiem) obszaru pamięci, którego zawartość ma być wyprowadzona na taśmę papierową. W adresie końca nie można podać segmentu (segment adresu końca jest taki sam jak segment adresu początku). Dlatego też za pomocą dyrektywy W można wyperforować zawartość bloku pamięci o pojemności do 64K bajtów.

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

Dyrektywa zostaje zakończona znakiem powrotu karetki (CR) i na taśmie papierowej jest dziurkowana:

- 100 znaków pustych.
- rekord rozszerzenia adresu (extended address record) - tylko w formacie 8086
- rekordy danych zawierające zawartość bloku pamięci od adresu początku do adresu końca włącznie
- rekord adresu startu (start address record) - tylko w formacie 8086, jeżeli był podany adres startu

- rekord konca zbioru (end of file record)
- 100 znaków pustych

Gdy operator zazadał dziurkowania taśmy w formacie 8086 (W), wartość segmentu podanego przez niego w adresie początku (lub zawartość rejestru CS gdy segment w adresie startu nie został podany) jest wyprowadzana na taśmie w rekordzie rozszerzenia adresu. Offset adresu początku jest umieszczany jako adres ładowania w pierwszym rekordzie danych. Wyprowadzany rekord adresu startu (o ile został podany adres startu) zawiera segment i offset, które zostają umieszczone w rejestrach CS i IP przy czytaniu taśmy dyrektywa R.

Gdy operator zazadał dziurkowania taśmy w formacie 8080 (WX), offset adresu początku jest umieszczany jako adres ładowania w pierwszym rekordzie danych; natomiast adres startu programu (jeśli został podany) jest umieszczany w rekordzie konca zbioru. Segment adresu początku jest w tym przypadku używany tylko do definicji wyprowadzanego na taśmie bloku pamięci i nie jest dziurkowany na taśmie.

Wszystkie znaki wyprowadzane na taśmie są znakami parzystymi.

#### BLEDY:

- podanie adresu konca mniejszego niż adres początku - sygnalizacja znakiem zapytania (?)
- użycie w którymkolwiek z adresów znaku nie będącego cyfrą heksadecymalną - sygnalizacja znakiem zapytania (?)
- dziurkarka nie włączona do sieci lub brak w niej taśmy - sygnalizacja komunikatem: BLAD DT

#### PRZYKŁADY:

1. Wyprowadzenie na taśmie zawartości pamięci od adresu 521H do 638H relatywnie do rejestru CS w formacie 8086:

MB6)W521,638(CR)  
MB6)

2. Wyprowadzenie na taśmie zawartości pamięci od adresu FF100H do FF330H z adresem startu 100H w formacie 8080:

MB6)WxFF00:100,330,100(CR)  
MB6)

### 3.8. DYREKTYWA R - czytanie danych z taśmy papierowej

RE<PR>J<CR>

Dyrektywa R umożliwia czytanie danych z taśmy papierowej w formacie 8086 lub 8080 (zgodnie z Hexadecimal Object File Format) i umieszczanie ich w pamięci RAM.

Jeśli taśma jest w formacie 8080 to dane są umieszczane w pamięci zgodnie z adresami ładowania znajdującymi się w rekordach danych relatywnie do segmentu o wartości 0. Adres startu w rekordzie konca zbioru (jeśli jest różny od 0000) umieszczany jest w rejestrze IP, a zawartość rejestru CS pozostaje bez zmiany.

Jeżeli taśma jest w formacie 8086 i zawiera rekord adresu startu programu to rejestry CS i IP przyjmują wartości zgodne z odczytanymi w tym rekordzie.

Gdy operator użyje w wywołaniu dyrektywy opcjonalnego przesunięcia to jego wartość jest dodawana do każdego adresu ładowania danych odczytanego z taśmy. Przesunięcie jest liczbą

• heksadecymalna z zakresu 0-FFFFH.

Gdy cała tasma zostanie poprawnie wczytana to jest to sygnalizowane komunikatem: OK. —

#### BLEDY:

- czytnik nie włączony do sieci lub brak tasm w czytniku - sygnalizacja komunikatem: BLAD CT
- odczytany został znak nieparzysty - sygnalizacja komunikatem: PARZYSTOSC
- wykryta zła suma kontrolna rekordu - sygnalizacja komunikatem: ZLA SUMA
- niezdefiniowany typ rekordu - sygnalizacja komunikatem: ZLY TYP
- zablokowana tasma papierowa - sygnalizacja komunikatem: BLAD CT
- odczytany w rekordzie znak nie stanowi cyfry heksadecymalnej - sygnalizacja znakiem zapytania (?)
- podanie w przesunięciu znaku, który nie jest cyfra heksadecymalna - sygnalizacja znakiem zapytania (?)
- adres ładowania na tasmie (plus ewentualne przesunięcie) wyznacza pamięć typu PROM lub pamięć nieistniejąca - sygnalizacja znakiem wykrzyknika (!) poprzedzonym offsetem adresu dla którego stwierdzono ten błąd.

#### PRZYKŁADY:

1. Wczytanie tasm papierowej:

M86)R<CR>

OK

M86)

2. Wczytanie tasm papierowej do pamięci o adresie o 100H większym od adresu ładowania znajdującego się na tasmie:

M86)R100<CR>

OK

M86)

### 3.9. DYREKTYWA V - porównanie zawartości pamięci

z tasm papierowa

VE<PR>J<CR>

Dyrektywa V umożliwia porównanie zawartości pamięci z danymi na tasmie papierowej w formacie 8086 lub 8080 (zgodnie z Hexadecimal Object File Format).

Działanie dyrektywy V jest analogiczne do działania dyrektywy R z tym, że dane z tasm nie są wpisywane do pamięci a jedynie z nią porównywane. Adres startu programu na tasmie (jeżeli jest) jest ignorowany. Jeżeli dla całej tasm zostanie stwierdzona zgodność zawartości pamięci z danymi na tasmie jest wyświetlany komunikat: OK.

#### BLEDY:

- sygnalizacja błędów analogiczna jak w dyrektywie R z tym, że znak wykrzyknika (!) poprzedzony offsetem jest wyświetlany dla pierwszej komórki pamięci, dla której stwierdzono brak zgodności z danymi na tasmie.

#### PRZYKLADY:

1. Porównanie zawartości pamięci z danymi na taśmie papierowej:  
M86)V(CR)  
OK  
M86>
2. Porównanie danych z taśmy z zawartością pamięci o adresie o 200H większym od adresu ładowania znajdującego się na taśmie:  
M86)V200(CR)  
OK  
M86>

### 3.10. DYREKTYWA K - kopiowanie taśmy papierowej

KEPJ(CR)

Dyrektywa K umożliwia kopiowanie dowolnej taśmy papierowej.

Dyrektywa pozwala kopiować taśmę bez kontroli parzystości (K) lub z kontrolą parzystości (KP). Po naciśnięciu klawisza powrotu karetki rozpoczyna się kopiowanie taśmy, które trwa aż do wykrycia końca taśmy kopiowanej.

#### BLEDY:

- odczytanie z czytnika znaku nieparzystego w trybie kopiowania taśmy z kontrolą parzystości - sygnalizacja komunikatem: PARZYSTOSC
- czytnik taśmy nie włączony do sieci, taśma nie podłożona lub zablokowana - sygnalizacja komunikatem: BLAD CT
- dziurkarka taśmy nie włączona do sieci lub brak w niej taśmy - sygnalizacja komunikatem: BLAD DT

#### PRZYKLADY:

1. Kopiowanie dowolnej taśmy papierowej:  
M86)K(CR)  
M86>
2. Kopiowanie taśmy papierowej z kontrolą parzystości:  
M86)KP(CR)  
M86>

### 3.11. DYREKTYWA P - dziurkowanie taśmy dla programatora

P(AP), (AK)E, (PR)I(CR)

Dyrektywa P służy do wyprowadzania obszaru pamięci na taśmie papierowej w formacie 8080 z przeznaczeniem dla programatora pamięci EPROM typów 2708 i 2716. Ponieważ pamięci te są 8-bitowe, dyrektywa P wyprowadza kolejno dwie taśmy, pierwsza zawierająca parzyste bajty zadeklarowanego obszaru i druga zawierająca bajty nieparzyste. Każda taśma ma adres ładowania pierwszego rekordu ustawiony na wartość 0000 i niezależnie od deklarowanych adresów zawiera 1K bajtów.

Operator podaje adres początku i końca wyprowadzanego na taśmę obszaru pamięci oddzielone znakiem przecinka (;). W adresie końca nie można podawać sesmentu (sesment jest taki sam

- Jak w adresie początku). Ponieważ obie wyprowadzane taśmy zawierają po 1K bajtów, dlatego zadeklarowany obszar pamięci nie może być dłuższy od 2K bajtów. Gdy zadeklarowany obszar jest krótszy od 2K bajtów to obie taśmy są uzupełniane (na końcu) odpowiednią ilością bajtów zawierających same jedynki (OFFH). Spowoduje to że dalsza część przepalanego PROM-u pozostanie niezmieniona.

Operator może podać po adresie końca przesunięcie (poprzedzone znakiem przecinka (,)) jako liczbę heksadecymalną z zakresu 0-7FFH. Określa ono ilość bajtów jakie mają pozostać nieprzypalane na początku wyprowadzanego dyrektywa P obszaru pamięci (ilość bajtów zawierających same jedynki (OFFH) jaka znajduje się na obu taśmach razem przed pierwszym bajtem danych z pamięci).

Taśma zawierająca bajty parzyste (młodsze) jest poprzedzona na taśmie litera L, a taśma zawierająca bajty nieparzyste (starsze) jest poprzedzona na taśmie litera H.

#### BLEDY:

- podanie adresu końca mniejszego od adresu początku - sygnalizacja znakiem zapytania (?)
- zadeklarowany obszar pamięci wyprowadzany na taśmie większy od 2K bajtów - sygnalizacja znakiem zapytania (?)
- przesunięcie większe od 7FFH - sygnalizacja znakiem zapytania (?)
- zadeklarowany obszar pamięci wyprowadzany na taśmie plus przesunięcie większy od 2K bajtów - sygnalizacja znakiem zapytania (?)
- użycie w którymkolwiek z adresów lub w przesunięciu znaku nie będącego cyfrą heksadecymalną - sygnalizacja znakiem zapytania (?)
- dziurkarka nie włączona do sieci lub brak w niej taśmy - sygnalizacja komunikatem: BLAD DT

#### PRZYKŁADY:

1. Wyprowadzenie na taśmy dla programatora zawartości pamięci od 1000H do 1FFFFH relatywnie do rejestru ES:  
M86>PES:1000,17FF<CR>  
M86>PES:1800,1FFF<CR>  
M86>
2. Wyprowadzenie na taśmy dla programatora zawartości pamięci od 250H do 305H relatywnie do rejestru CS z przesunięciem 1K:  
M86>P250,305,4000<CR>  
M86>

### 3.12. DYREKTYWA G - uruchomienie programu

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

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

Po wywołaniu dyrektywy litera G na monitorze jest wyświetlana aktualna zawartość rejestru IP oraz zawartość komórki pamięci, której adres jest wyznaczony przez zawartość rejestrow CS i IP (komórka pamięci, od której byłby wykonywany program gdyby nie był podany nowy adres). Podanie w tym momencie znaku powrotu karetki <CR> kończy dyrektywę i

uruchamia program od adresu określonego przez rejestry CS i IP. W tym momencie można także uruchomić program od innego adresu podając ten adres i kończąc go znakiem powrotu karetki (CR).

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

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

```
BR @aaaa:bbbb  
M86)
```

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

Po zatrzymaniu programu w miejscu ustawienia pulapki możliwe jest korzystanie ze wszystkich dyrektyw MONITORA. W szczególności można użyć dyrektywy G z ponownym ustawieniem pulapki (z wyjątkiem ustawienia pulapki w miejscu zatrzymania programu - do tego celu służy dyrektywa U).

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

```
@aaaa:bbbb  
M86)
```

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

#### BLEDY:

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

#### PRZYKŁADY:

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

```
M86)G 1234-50 (CR)
```

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

```
M86)G 0000-00 1000:0,40 (CR)
```

```
BR @1000:0040
```

```
M86)G 0040-58 ,47 (CR)
```

```
BR @1000:0047
```

```
M86)
```

### 3.13. DYREKTYWA N - praca krokowa (single step)

NE<AS>],[[<AS>]],]\*<CR>

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

Podczas wyświetlania dyrektywy litera N na monitorze wyświetlana jest aktualna zawartość rejestru IP i zawartość komórki pamięci, której adres jest wyznaczony przez rejestry CS i IP (komórka pamięci od której byłby wykonywany program gdyby nie był podany nowy adres). Gdy operator chce wykonać instrukcję o innym adresie niż określony przez rejestry CS i IP może podać ten adres w tym momencie i zakończyć go znakiem przecinka (;).

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

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

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

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

#### BLEDY:

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

#### PRZYKŁADY:

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

M86>N 0000-00 1A5E,

1A5F-50,

1A60-B8,

1A63-8E,

1A65-B9 <CR>

M86>

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

M86>N 0031-03,

0034-40,

0035-50 34,

0035-50 <CR>

### 3.14. DYREKTYWA U - utrzymanie pulapki

U<CR>

- sygnalizacja komunikatem: ZLA TRANSMISJA - ECHO
- wysłana zła suma kontrolna rekordu - sygnalizacja komunikatem: ZLA TRANSMISJA - BŁĄD SUMY
- niezdefiniowany typ rekordu - sygnalizacja komunikatem: ZLA TRANSMISJA - ZŁY TYP
- adres ładowania w rekordzie (plus ewentualne przesunięcie) nie wolny w pamięci typu RAM - sygnalizacja komunikatem: ZLA TRANSMISJA - BRAK RAM

PRZYKŁADY:

1. Czytanie pliku o nazwie PROBNY.TB6 z dysku systemowego:  
M86 L:PROBNY.TB6 CR)  
TRANSMISJA OK  
M86)
2. Czytanie zb. op. o nazwie TEST.TB6 z dyskietki nr1 z konta [25,10] z przesunięciem 1000H:  
M86 L1000 DX1:[25,10]TEST.TB6(CR)  
TRANSMISJA OK  
M86)

Dyrektywa U służy do uruchomienia programu od adresu określonego przez rejestry CS i IP z ustawieniem pulapki w miejscu adresu startu programu.

Po wywołaniu dyrektywy litera U na monitorze wyświetlana jest aktualna wartość rejestru IP i zawartość komórki pamięci, której adres jest wyznaczony przez zawartość rejestrów CS i IP (komórka pamięci, od której będzie wykonywany program). Naciśnięcie znaku powrotu karetki (CR) powoduje wykonanie pierwszego instrukcji programu z trybem single step, ustawienie pulapki w miejscu pierwszego instrukcji i kontynuacja wykonywania programu. Dyrektywa U służy do wykonywania petli programu z strażnikiem pulapki w tym samym miejscu.

**BŁĘDY:**

- podanie znaku różnego od powrotu karetki (CR) w celu zakończenia dyrektywy - sygnalizacja znakiem zapytania (?).

**PRZYKŁADY:**

1. Dwukrotne wykonanie petli programowej:

MS6>U 0151-90 (CR)

BR @1000:0151

MS6>U 0151-90 (CR)

BR @1000:0151

MS6>

### 3.15. DYREKTYWA H - arytmetyka heksadecymalna

**H(D)<O>(D)=**

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

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

**BŁĘDY:**

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

**PRZYKŁADY:**

MS6>H1+2=0003

MS6>HFFFF+02=0001

MS6>H0001-3=FFFE

MS6>

### 3.16. DYREKTYWA L - czytanie danych z SM-4

## LE<PR> nazwa zbioru<CR>

Dyrektyna L umożliwia wczytanie danych w formacie taśmy papierowej (3000 lub 8086) przez interfejs V-24 pakietu MM86 Złącze C. Mikrokomputer MM86 zastępuje w czasie wykonywania operacji z terminala SM-4 wysłania do SM-a znaków danych wysłaniem znaków PIF (PERIPHERIAL INTERCHANGE PROGRAM) z komputera przesłanie zawartości pamięci do terminala zbioru na ekran. Przez wywołaniem funkcji LMM86 w terminalu zbioru na ekranie pojawia się komunikat o błędach. Jeśli komunikat PIF nie może być aktywny, komunikat LMM86 nie może być aktywny. Jeśli komunikat LMM86 nie może być aktywny, komunikat PIF nie może być aktywny.

Po dokonaniu poprawek i zmian w terminalu SM-4 podanie komunikatu LMM86 i nastąpił błąd w terminalu MM86 zamienia na komunikat MM86 złącze do monitora ekranowego na złącze od mikrokomputera SM. Po dokonaniu tej zmiany mikro wysła do SM-a tekst.

## FIP TI: nazwa zbioru<CR>

sprawdzając po wysłaniu każdego znaku otrzymane od SM-a echo, a następnie zaczyna odbierać nadawane przez SM-4 znaki. W trakcie odbioru kontrolowana jest zgodność otrzymywanego zbioru z "Hexadecimal Object File Format" i sprawdzana jest prawidłowość zapisu danych do pamięci. Po zakończeniu transmisji operator ponownie dołącza do MM86 monitor ekranowy, na którym zostaje wypisany komunikat o poprawności zakończonej transmisji. Transmisja zakończona prawidłowo sygnalizowana jest komunikatem:

### TRANSMISJA OK

Jeżeli MM86 stwierdzi błąd w odbieranym echo w trakcie wysyłania do SM-a komunikatu z nazwą zbioru sygnalizuje to (po zamianie złącz) komunikatem:

### ZŁA TRANSMISJA - ÉCHO

Błędy w czasie transmisji danych sygnalizowane są po zamianie złącz komunikatem analogicznym jak w dyrektywie R czytania taśmy papierowej poprzedzone tekstem: ZŁA TRANSMISJA.

Jeżeli przesyłany zbiór jest w formacie 8080 to dane umieszczane są w pamięci RAM zgodnie z adresami ładowania znajdującymi się w rekordach relatywnie do segmentu o wartości 0. Adres startu w rekordzie końca zbioru (jeżeli jest różny od 0000) umieszczony jest w rejestrze IP użytkownika, a zawartość rejestru CS pozostaje bez zmiany.

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

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

### BŁĘDY:

- podanie w przesunięciu znaku nie będącego cyfrą heksadecymalną - sygnalizacja znakiem zapytania (?)
- błędne echo od SM-a w odpowiedzi na inicjację transmisji -

• signalizacja komunikatem: ZLA TRANSMISJA - ECHO

... ..  
... ..  
... ..  
... ..

... ..

... ..  
... ..  
... ..  
... ..  
... ..  
... ..  
... ..  
... ..  
... ..  
... ..

63

64

Tabela 1.  
Zestawienie dyrektyw MONITORA OPERATORSKIEGO

| I | I                                  | I | I                         |
|---|------------------------------------|---|---------------------------|
| I | Dyrektywa - Funkcja                | I | Wywołanie                 |
| I | I                                  | I | I                         |
| I | S Odczyt i zmiana zawartosci ko-   | I | SCWJ<A>,[<D>],]*<CR>      |
| I | morki pamieci                      | I | I                         |
| I | I                                  | I | I                         |
| I | X Odczyt i zmiana zawartosci re-   | I | XC<R>[[<D>],]*<CR>        |
| I | jestru                             | I | I                         |
| I | I                                  | I | I                         |
| I | D Odczyt zawartosci pamieci        | I | D<AP>[,<AK>]]<CR>         |
| I | I                                  | I | I                         |
| I | M Przesuniecie zawartosci pamieci  | I | M<AP>,<AK>,<AL><CR>       |
| I | I                                  | I | I                         |
| I | I Odczyt danych z bramy wejsciowej | I | ICWJ<I/O>,[,]*<CR>        |
| I | I                                  | I | I                         |
| I | O Wpis danych do bramy wyjsciowej  | I | OCWJ<I/O>,<D>[,<D>]]*<CR> |
| I | I                                  | I | I                         |
| I | W Wyprowadzenie zawartosci pamie-  | I | WCXJ<AP>,<AK>[,<AS>]]<CR> |
| I | ci na tasme papierowa              | I | I                         |
| I | I                                  | I | I                         |
| I | R Wprowadzenie danych z tasmy pa-  | I | RC<PR>]]<CR>              |
| I | pierowej do pamieci                | I | I                         |
| I | I                                  | I | I                         |
| I | V Porownanie zawartosci pamieci    | I | VC<PR>]]<CR>              |
| I | z tasma papierowa                  | I | I                         |
| I | I                                  | I | I                         |
| I | K Kopiowanie taśmy papierowej      | I | K[FP]]<CR>                |
| I | I                                  | I | I                         |
| I | P Wyprowadzenie zawartosci pamie-  | I | P<AP>,<AK>[,<PR>]]<CR>    |
| I | ci na tasme papierowa dla pro-     | I | I                         |
| I | gramatora                          | I | I                         |
| I | I                                  | I | I                         |
| I | G Uruchomienie programu            | I | GC<AS>]][,<A>]]<CR>       |
| I | I                                  | I | I                         |
| I | N Wykonanie pojedynczej instru-    | I | NC<AS>]][[<AS>]],]*<CR>   |
| I | kcji programu uzytkownika          | I | I                         |
| I | I                                  | I | I                         |
| I | U Utrzymanie pulapki               | I | U<CR>                     |
| I | I                                  | I | I                         |
| I | H Arytmetyka heksadecymalna        | I | H<D><O><D>=               |
| I | I                                  | I | I                         |
| I | L Wprowadzanie danych z SM do      | I | LC<PR>]],nazwa zbioru<CR> |
| I | pamieci                            | I | I                         |
| I | I                                  | I | I                         |

## SPIS TRESCI

|                                            |      |
|--------------------------------------------|------|
| 1. Testy uruchomieniowe pakietu            |      |
| pamięci RAM/PROM - ML50                    | - 2  |
| 1.1 Testy repetycyjne pakietu ML50         | - 2  |
| 1.2 Testy diagnostyczne pakietu ML50       | - 3  |
| 2. Testy uruchomieniowe pakietu sprzężenia |      |
| z pamięcią kasetową - MI50                 | - 7  |
| 2.1 Testy rejestrów pakietu                | - 8  |
| 2.2 Testy współpracy z PK                  | - 10 |

## 1. TESTY URUCHOMIENIOWE PAKIETU PAMIĘCI RAM/PROM - ML50

Testy pakietu ML50 składają się z dwóch części zawierających programy repetycyjne oraz diagnostyczne. Testy repetycyjne umożliwiają oscyloskopową obserwację i kontrolę wszelkich sygnałów podczas pracy pakietu. Testy diagnostyczne sprawdzają poprawność działania pakietu, a wykryte błędy sygnalizują odpowiednimi komunikatami, na podstawie których operator może lokalizować uszkodzenie.

Testy napisane są w języku assemblera 8086 jako procedura typu FAR. Po wywołaniu testy pakietu ML50 zgłaszają się komunikatem:

### TESTY URUCHOMIENIOWE PAKIETU ML50

- 1 - TESTY REPETYCYJNE
- 2 - TESTY DIAGNOSTYCZNE
- . - KONIEC TESTOW
- :

Po znaku dwukropka ":" operator powinien wybrać żadaną grupę testów. Naciśnięcie znaku kropki "." kończy testy i powoduje powrót z podprogramu.

#### 1.1 TESTY REPETYCYJNE PAKIETU ML50

Testy repetycyjne polegają na ciągłym odczycie/zapisie danych z/do pakietu ML50. Operacje odczytu/zapisu mogą być 8-mio lub 16-to bitowe. Przerwanie testów jest możliwe przez podanie znaku CTRL/Z (jednocześnie klawisz CTRL oraz klawisz Z). Następuje wtedy powrót do części głównej testów.

Po uruchomieniu testów repetycyjnych na ekranie wyświetlane jest menu w następującej postaci:

- 1 - ODCZYT 8-mio BITOWY
- 2 - ODCZYT 16-to BITOWY
- 3 - ZAPIS 8-mio BITOWY
- 4 - ZAPIS 16-to BITOWY
- . - KONIEC TESTOW
- :

Po wybraniu jednej z wersji testu operator podaje adres z jakim ma następować komunikacja w formacie SEGMENT:OFFSET zakończony znakiem CR oraz dla testów zapisu także dane, które mają być wpisywane do pakietu.

Komunikacje zapisu/odczytu wykonywane są co około 22 mikrosekundy.

## 1.2 TESTY DIAGNOSTYCZNE PAKIETU ML50

Po uruchomieniu testów diagnostycznych na ekranie wyświetlany jest spis dostępnych programów:

- 1 - TEST PROM (8-mio bitowy)
- 2 - TEST PROM ( 16-bitowy)
- 3 - TEST UPŁYWNOSCI RAM
- 4 - TEST INFORMACJI RAM
- . - KONIEC TESTOW
- :

Operator powinien naciskając odpowiedni klawisz (cyfry od 1 do 4) wybrać właściwy program.

Testy diagnostyczne po każdej komunikacji z pamięcią sprawdzają sygnał XACK. Brak XACK powoduje wyświetlenie komunikatu:

BRAK XACK DLA ADRESU: XXXX

i program czeka na decyzję operatora. Wciśnięcie znaku kropki "." kończy test, natomiast każdy inny znak powoduje kontynuację programu.

#### 1.2.1 Testy FROM

Przed uruchomieniem testów FROM należy umieścić w podstawkach kostki EPROM ze wzorcową zawartością. Test jest przystosowany do obsadzenia pakietu kostkami o pojemności 2K (2716) i testuje cały obszar FROM -32K. Program porównuje informacje odczytywaną z kolejnych komórek pamięci FROM z informacją wygenerowaną z generatora pseudolosowego. Każdy błąd sygnalizowany jest komunikatem o formacie:

| adres | dane poprawne | dane odczytane |
|-------|---------------|----------------|
|-------|---------------|----------------|

Adres wyświetlony jest w postaci heksadecymalnej (sam OFFSET) natomiast dane w postaci binarnej.

Po wydruku komunikatu o błędzie program oczekuje na reakcję operatora. Wciśnięcie znaku kropki "." kończy test i program przechodzi do części głównej natomiast wciśnięcie dowolnego innego znaku powoduje kontynuację testu.

Test 1 wykonuje przekazy 8-mio bitowe a test 2 przekazy 16-to bitowe. Test kończy się komunikatem:

KONIEC TESTU

i w przypadku braku błędów trwa około 6 sek.

#### 1.2.2 Test upływności RAM

Test upływności polega na zapisaniu całej pamięci RAM

(8K) pakietu jedynekami (z jednoczesnym sprawdzeniem poprawności zapisu) odczekaniu zadeklarowanego przez operatora czasu. sprawdzeniu zawartości pamięci, a następnie powtórzeniu tej samej operacji dla informacji składającej się z zer. Poszczególne fazy testu sygnalizowane są odpowiednimi komunikatami. Komunikaty o wykrytych błędach są identyczne jak w testach FROM i decyzja o dalszej pracy testu należy do operatora (koniec testu lub kontynuacja testowania). Przy uruchomieniu testu operator podaje opóźnienie w sekundach między zapisem a powtornym odczytem danych jako liczbę heksadecymalną maksymalnie o 4 znakach oraz adres początku obszaru RAM na pakiecie.

Wszystkie przekazy informacji w teście są 8-mio bitowe.

### 1.2.3 Test informacji RAM

Test sprawdza całą pamięć RAM pakietu (8k) w blokach po 2k. Test składa się z 2 faz sygnalizowanych odpowiednimi komunikatami na ekranie monitora. Najpierw wykonywany jest test typu "pływające zero" tzn. cała pamięć wypełniona jest jedynekami a następnie poszczególne bity są zerowe. Po wpisaniu każdego zera sprawdzany jest cały obszar bloku czy nie nastąpiło przekłamanie wpisanej poprzednio informacji. Po fazie "pływające zero" następuje "pływająca jedynka".

Po uruchomieniu na ekranie wyświetlany jest komunikat informujący o aktualnie wykonywanym fragmencie testu:

pływająca jedynka

lub

pływające zero

a także obszar aktualnie testowanej pamięci wewnętrznej RAM jako:

BLOK 1

BLOK 2

BLOK 3

BLOK 4

Nieudana próba wpisania pływającej jedynki (pływającego zera) do komórki pamięci sygnalizowana jest wyświetleniem na ekranie komunikatu w postaci:

Błąd dla adresu YYYY zapisano XXXX odczytano QQQQ

Informacja o wykrytym błędzie wyświetlana jest na ekranie w postaci:

Po wpisaniu XXXX do komórki YYYY odczytano QQQQ z komórki ZZZZ

W teście pływającej jedynki informacja wpisywana XXXX zawiera zawsze tylko jedną jedynkę na kolejnych 16-bitach, a informacja odczytana QQQQ powinna być zerowa.

W teście pływającego zera informacja wpisywana XXXX zawiera zawsze tylko jedno zero na kolejnych 16-bitach, a informacja odczytana QQQQ powinna mieć wartość FFFF.

Po wyświetleniu komunikatu o błędzie program czeka na decyzję operatora: kropka kończy test, każdy inny znak powoduje kontynuację. W przypadku braku błędów test trwa około 8 minut. Zakończenie sygnalizowane jest komunikatem:

KONIEC TESTU

## 2. TESTY URUCHOMIENIOWE PAKIETU SPRZĘZENIA Z PAMIĘCIĄ KASETOWĄ - MI50

Testy uruchomieniowe pakietu MI50 składają się z dwóch części zawierających programy wymagające podłączenia pamięci kasetowej i nie wymagające takiego podłączenia. W obu grupach testów znajdują się programy o charakterze repetytywnym umożliwiające oscyloskopową obserwację wybranych sygnałów na pakiecie oraz programy o charakterze diagnostycznym sprawdzające określone funkcje pakietu.

Testy pakietu MI50 są napisane w języku assemblera jako procedura typu FAR. Po uruchomieniu zgłaszają one się komunikatem:

PODAJ ADRES PAKIETU - 2 CYFRY HEX [20]:

Operator powinien podać adres testowanego pakietu MI50 jako dwie cyfry heksadecymalne zakończone znakiem powrotu karetki <CR>. Jeżeli adres testowanego pakietu jest równy 20H wystarczy wciśnięcie samego znaku <CR>. Podanie przez operatora znaku kropki "." powoduje zakończenie testów i powrót z podprogramu.

Po podaniu przez operatora adresu pakietu na ekranie wyświetlany jest tekst:

TESTY URUCHOMIENIOWE PAKIETU MI50

1 - TESTY REJESTROW PAKIETU

2 - TESTY WSPÓŁPRACY Z PK

. - KONIEC TESTOW

:

Po znaku dwukropka ":" operator powinien wybrać żadaną grupę testów. Naciśnięcie znaku kropki "." kończy testy i powoduje powrót z podprogramu.

## 2.1 TESTY REJESTROW PAKIETU

Testy te działają przy niepodłączonej pamięci kasetowej do pakietu.

W skład testów rejestrów pakietu wchodzi oscyloskopowe programy repetycyjnego odczytu/zapisu wszystkich rejestrów pakietu oraz program sprawdzenia działania licznika pomiaru czasu. Testy repetycyjne mogą być przerwane w dowolnym momencie przez podanie znaku CTRL/Z (jednocześnie klawisz CTRL i klawisz Z). Następuje wtedy powrót do części głównej testów.

Po uruchomieniu testów rejestrów pakietu na ekranie wyświetlane jest menu w następującej postaci:

- 1 - ODCZYT REJESTRU
- 2 - ZAPIS DO REJESTRU INFORMACJI STAŁEJ
- 3 - ZAPIS DO REJESTRU INFORMACJI NAPRZEMIENNEJ
- 4 - SPRAWDZENIE LICZNIKA CZASU
- :

Zapis możliwy jest do wszystkich rejestrów pakietu, natomiast odczyt tylko rejestru danych USART'A, rejestru słowa stanu USART'a i rejestru stanu pakietu. Dla testów zapisu operator musi podać informację jaka ma być przesyłana do wybranego rejestru. Operacja odczytu rejestru wykonywana jest co około 20  $\mu$ s a operacja zapisu co około 23  $\mu$ s.

W teście 3 do rejestru wpisywana jest informacja podana przez operatora na przemian z informacją będącą jej negacją. W tym teście komunikacje są wykonywane co około 24 mikrosekundy.

### 2.1.1 Testy zapisu/odczytu rejestrów

Po uruchomieniu programu 1,2 lub 3 operator w odpowiedzi na komunikat:

- 1 - REJESTR DANYCH
- 2 - S.S. USART'A
- 3 - S.S. PK
- 4 - REJESTR LICZNIKA
- 5 - REJESTR ZEZWOLEN
- :

powinien wybrać rejestr, którego ma dotyczyć test.

Przerwanie każdego z tych testów możliwe jest w dowolnym momencie poprzez podanie z klawiatury znaku CTRL/Z.

### 2.1.2 Sprawdzenie licznika czasu

Operator musi podać z klawiatury dwucyfrową liczbę hexadecymalną określającą ilość taktów zegara 10 mikrosekund jakie ma zliczyć licznik czasu na pakiecie MI50. Test sprawdza czy w słowie stanu pakietu ustawiony jest bit sygnalizujący przepełnienie licznika po zadeklarowanym czasie z dokładnością plus/minus jeden impuls. Za wczesne lub za późne (albo brak) pojawienie się sygnału o przepełnieniu licznika sygnalizowane jest odpowiednim komunikatem.

Po wykonaniu każdego sprawdzenia działania licznika program przechodzi ponownie do zapytania o opóźnienie. Operator może podać nową wartość (dwie cyfry hexadecymalne) lub znak kropki "." w celu zakończenia wybranego testu i przejścia do części głównej testów pakietu MI50.

## 2.2 TESTY WSPÓŁPRACY Z PK

Testy te wymagają podłączenia do pakietu MI50 jednostki pamięci kasetowej. W skład testów współpracy z PK wchodzi testy sprawdzające poprawność wszystkich funkcji realizowanych przez pamięć kasetową: przewijania w przód i do tyłu, ustawiania taśmy w położeniu początkowym, zapisu i odczytu bloków danych oraz wyszukiwania bloków.

Po uruchomieniu testów współpracy z PK na ekranie wyświetlane jest zapytanie:

PODAJ NUMER PRZEWIJAKA(1/2):

Operator podaje numer testowanego przewijaka jako cyfrę 1 lub 2. Następnie wyświetlana jest lista dostępnych testów w postaci:

- 1 - REPETYCYJNY ZAPIS BŁOKÓW
- 2 - REPETYCYJNY ODCZYT BŁOKÓW
- 3 - ZAPIS KOLEJNYCH BŁOKÓW DANYCH
- 4 - ODCZYT KOLEJNYCH BŁOKÓW DANYCH
- 5 - PRZEWIJANIE TASMY
- 6 - USTAWIANIE TASMY W POŁOŻENIE POCZĄTKOWE
- 7 - ZLICZANIE BŁOKÓW
- :

Operator podaje numer wybranego testu jako liczbę z zakresu 1-7 lub znak kropki "." w celu zakończenia testów współpracy z PK. Testy o numerach 1-5 i 7 można przerwać w dowolnym momencie podając znak CTRL/Z z klawiatury monitora ekranowego. Testy te kończą się też po wykryciu znacznika końca taśmy - sygnalizowane jest to komunikatem:

KONIEC TASMY

### 2.2.1 Repetycyjny zapis bloków

Test ten polega na zapisywaniu na taśmie bloków danych o długości 1 bajt w postaci: SYNC, DANE, SYNC w sposób nieskończony bez przerwy międzyblokowej. Test ten umożliwia oscyloskopową obserwację zapisu danych na taśmie. W trakcie zapisu włączona jest także głowica czytająca, dane odbierane nie są czytane przez program ale można je obserwować na oscyloskopie.

W odpowiedzi na zapytanie:

PODAJ INFORMACJE (2 CYFRY HEX):

operator podaje informację, która ma być zapisywana na taśmie (jako DANE).

Test sygnalizuje odpowiednimi komunikatami sytuację gdy pamięć kasetowa nie jest przydzielona i gdy zapis na taśmie jest niedozwolony. Test można przerwać w dowolnym momencie znakiem CTRL/Z.

### 2.2.2 Repetycyjny odczyt bloków

Test ten polega na odczytywaniu z taśmy bloków zapisanych testem 1 (repetycyjny zapis bloków) czyli bloków o długości 1 bajt w postaci: SYNC, DANE, SYNC. Umożliwia on oscyloskopową obserwację odczytu danych z taśmy. Odczytywane dane nie są wyświetlane na ekranie monitora. Test można przerwać w dowolnym momencie znakiem CTRL/Z.

### 2.2.3 Zapis kolejnych bloków danych

Test zapisuje na taśmie bloki danych o długości 256 jednakowych bajtów z przerwą międzyblokową ok. 1 sek.. Do kolejnych bloków zapisywane są kolejne (począwszy od 1)

cyfry heksadecymalne pominawszy OAAH (znak synchronizacji). Pierwszy blok składa się z 256 jedynek, drugi z 256 dwójek itd.. W trakcie zapisu włączona jest także głowica czytająca: dane odbierane nie są czytane przez program ale można je obserwować na oscyloskopie.

Test ma charakter nieskończony i może być przerwany w dowolnym momencie znakiem CTRL/Z.

#### 2.2.4 Odczyt kolejnych bloków

Test ten polega na odczytywaniu z taśmy kolejnych bloków danych zapisanych testem 3. Program sprawdza poprawność odebranych danych. Jeżeli cały blok odczytany został poprawnie, tzn. odebrano dokładnie 256 jednakowych bajtów, na monitorze wyświetlany jest bajt stanowiący zawartość bloku. W przypadku gdy program odczyta ilość bajtów różną od 256 to na ekranie wyświetlany jest komunikat:

ZŁA DŁUGOSC

natomiast gdy kolejne bajty w bloku są niejednakowe to wyświetlany jest tekst:

BŁĄD DANYCH

Niezależnie od poprawności odczytywanych bloków test będzie dalej i można go przerwać znakiem CTRL/Z.

#### 2.2.5 Przewijanie taśmy

Test zgłasza się komunikatem:

PODAJ KIERUNEK PRZEWIJANIA (P/T):

Operator powinien nacisnąć literę F gdy chce aby przewijanie następowało do przodu lub literę T dla przewijania do tyłu.

Test można przerwać znakiem CTRL/Z lub kończy się sam po wykryciu znacznika końca taśmy.

#### 2.2.6 Ustawianie taśmy w położeniu początkowym

Test ustawia taśmę na pierwszym znaczniku końca taśmy. Program wysyła do pamięci kasetowej sygnał REV i oczekuje aż taśma przewinie się do fizycznego początku co jest sygnalizowane pojawieniem się sygnału RDY. Następnie przy powolnym ruchu w przód wyszukiwany jest znacznik końca taśmy.

#### 2.2.7 Zliczanie bloków

Operator po podaniu kierunku przewijania (analogicznie jak w teście 5) wprowadza ilość przerw międzyblokowych (dwucyfrową liczbę heksadecymalną), która ma zliczyć pakiet.

Test kończy się po zasygnalizowaniu przez pakiet zliczenia zadanej liczby bloków: możliwe jest wcześniejsze przerwanie testu znakiem CTRL/Z.