

Wojskowa Akademia Techniczna
im. Jarosława Dąbrowskiego

Laboratorium
Architektury i organizacji komputerów I

Prowadzący mgr inż. Artur Miktus
Sprawozdanie z ćwiczenia laboratoryjnego
nr 2

Temat ćwiczenia:
Mikroprogram pobrania rozkazu.

Wykonał: Arkadiusz Ostrzyżek
Grupa: WCY22KY2S1
Data wykonania ćwiczenia: 2023-11-07

Polecenie:

Lab2 WCY22KY2S1 zima 2023

Dana jest zawartość początkowa rejestrów i pamięci operacyjnej PAO jak w poniższej tabeli:

| Rejestry | |
|----------|------------|
| A | nr |
| LR | 150+nr |
| RI | 50 |
| | |
| | |
| PAO | |
| Adres | Zawartość |
| 0 | 150+nr |
| nr | 200+nr |
| nr+5 | 255 |
| 100 | 127 |
| LR | STQ 000 nr |
| LR+1 | LLQ nr+5 |
| LR+2 | TLD 001 nr |
| LR+3 | LNG 010 1 |
| LR+4 | LRA nr+5 |
| LR+5 | ALA nr |
| | |
| 255 | 3 |

Pozostałe komórki PAO są wyzerowane.

Stopień trudności zadania:

- Na dostatecznie – pobrać pierwsze 3 rozkazy.
- Na dobrze – pobrać pierwsze 4 rozkazy.
- Na bardzo dobrze – pobrać pierwsze 5 rozkazów.

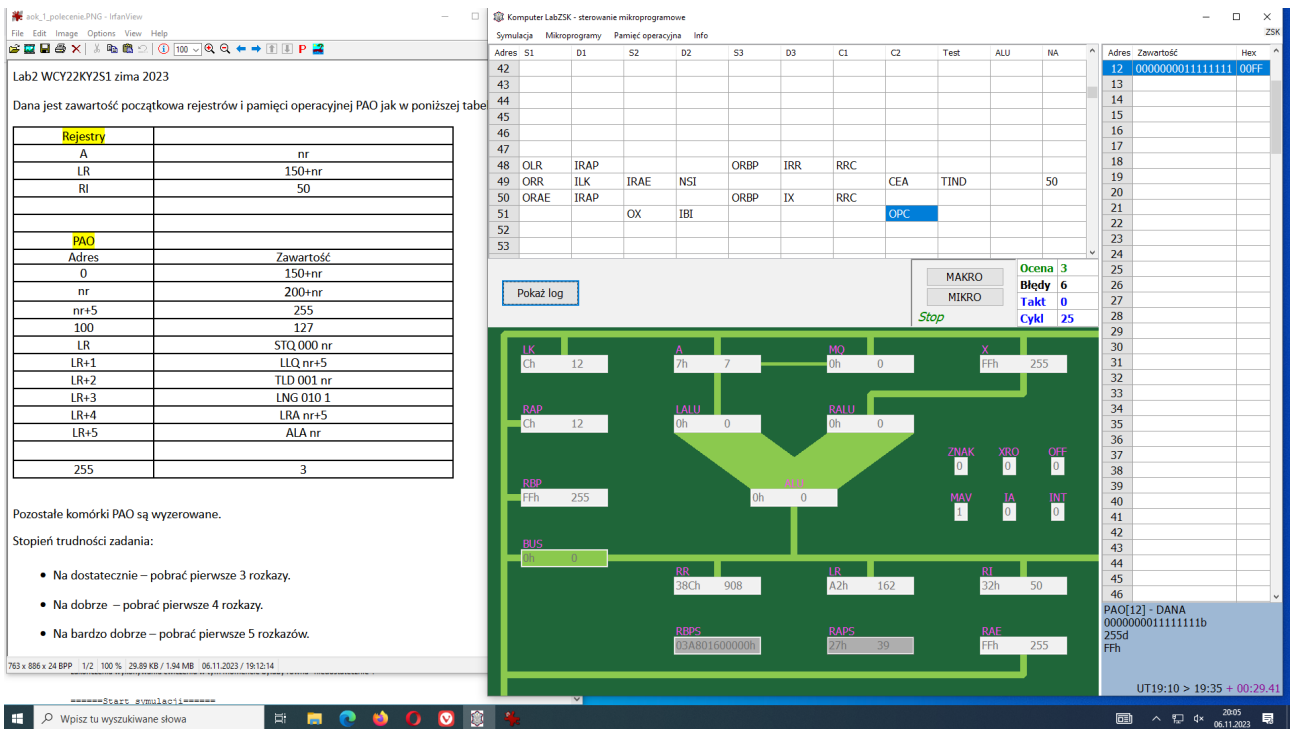
Wydruk zawartości PM:

| | | | |
|----|------|---------|------------------------|
| 0 | Test | ___TINT | Brak przzerwania |
| | NA | ___48 | |
| 1 | Test | ___TIND | Adresowanie pośrednie |
| | NA | ___52 | |
| 48 | S1 | ___OLR | LR -> BUS |
| | D1 | ___IRAP | BUS -> RAP |
| | S3 | ___ORBP | RBP -> BUS |
| | D3 | ___IRR | BUS -> RR |
| | C1 | ___RRC | Rozpoczęcie RRC |
| 49 | S1 | ___ORR | RR -> BUS |
| | D1 | ___ILK | BUS -> LK |
| | S2 | ___IRAE | SUMA -> RAE |
| | D2 | ___NSI | LR+1 -> LR |
| | C2 | ___CEA | Oblicz adres efektywny |
| | Test | ___TIND | Adresowanie pośrednie |
| | NA | ___50 | |
| 50 | S1 | ___ORAE | RAE -> BUS |
| | D1 | ___IRAP | BUS -> RAP |
| | S3 | ___ORBP | RBP -> BUS |
| | D3 | ___IX | BUS -> X |
| | C1 | ___RRC | Rozpoczęcie RRC |
| 51 | S2 | ___OX | X -> BUS |
| | D2 | ___IBI | BUS -> RAE |
| | C2 | ___OPC | OP albo AOP+32 -> RAPS |

Wydruk PAO:

| | | | | | |
|-----|-------------------|-------|-------|---------|------|
| 0 | 0000000010011101b | 009Dh | 157 | | |
| 7 | 0000000011001111b | 00CFh | 207 | | |
| 12 | 0000000011111111b | 00FFh | 255 | | |
| 100 | 0000000011111111b | 007Fh | 127 | | |
| 157 | 0010100000000111b | 2807h | OP=5 | XSI=000 | DA=7 |
| 158 | 0000001010001100b | 028Ch | AOP=5 | | N=12 |
| 159 | 1010100100000111b | A907h | OP=21 | XSI=001 | DA=7 |
| 160 | 1101101000000001b | DA01h | OP=27 | XSI=010 | DA=1 |
| 161 | 0000001110001100b | 038Ch | AOP=7 | | N=12 |
| 162 | 0000000100000111b | 0107h | AOP=2 | | N=7 |
| 255 | 0000000000000011b | 0003h | 3 | | |

Zrzut ekranu z obrazem ze stanem końcowym LabZSK:



Wydruk (plik tekstowy) logu z wykonania ćwiczenia:

Start symulatora 06.11.2023 19:10:02

Stacja "LABITC"

Zalogowano jako: "Student"

Wersja aplikacji: 1.2.3.0

Dostępne interfejsy sieciowe: 192.168.56.1

10.6.106.3

192.168.47.1

192.168.13.1

=====Start symulacji=====

19:35.52

=====Zawartość rejestrów=====

LK = 0h 0

A = 7h 7

MQ = 0h 0

$$X = 0h \quad 0$$

RAP = 0h 0

LALU = 0h 0

RALU = 0h 0

```
RBP    = 0h    0
```

ALU = 0h 0

BUS = 0h 0

RR = 0h 0

LR = 9Dh 157

RI = 32h 50

RAPS = 0h 0

RAE = 0h 0
L = 0h 0
R = 0h 0
SUMA = 0h 0

MAV = 1, IA = 0, INT = 0
ZNAK = 0, XRO = 0, OFF = 0

MAKRO

=====0=====

Takt0: RBPS=000000020030h

Takt7:

INT = 0

TEST | TINT : Brak przerwania(INT != 0)

RAPS = 48 / 30h

MAKRO

=====48=====

Takt0: RBPS=5006C4000000h

Takt1:

S1 | OLR : LR -> BUS

BUS = 157 / 9Dh

D1 | IRAP : BUS -> RAP

RAP = 157 / 9Dh

C1 | RRC : Rozpoczęcie RRC

RBP = 0 / 0h

Takt7:

S3 | ORBP : RBP -> BUS

BUS = 0 / 0h

D3 | IRR : BUS -> RR

RR = 0 / 0h

RAPS = 49 / 31h

PAO[157] = 0x0000 -zmiana-> PAO[157] = 0x2807

PAO[158] = 0x0000 -zmiana-> PAO[158] = 0x028C

PAO[159] = 0x0000 -zmiana-> PAO[159] = 0xA907

PAO[160] = 0x0000 -zmiana-> PAO[160] = 0xDA01

PAO[161] = 0x0000 -zmiana-> PAO[161] = 0x038C

PAO[162] = 0x0000 -zmiana-> PAO[162] = 0x0107

PAO[155] = 0x0107 -zmiana-> PAO[155] = 0x

PAO[154] = 0x038C -zmiana-> PAO[154] = 0x

PAO[153] = 0xDA01 -zmiana-> PAO[153] = 0x

PAO[152] = 0xA907 -zmiana-> PAO[152] = 0x

PAO[151] = 0x028C -zmiana-> PAO[151] = 0x

PAO[150] = 0x2807 -zmiana-> PAO[150] = 0x

RAP = 157 -zmiana-> RAP = 0

RAPS = 49 -zmiana-> RAPS = 0

MAKRO

=====0=====

Takt0: RBPS=000000020030h

Takt7:

INT = 0

TEST | TINT : Brak przerwania(INT ?= 0)

RAPS = 48 / 30h

MAKRO

=====48=====

Takt0: RBPS=5006C4000000h

Takt1:

S1 | OLR : LR -> BUS

BUS = 157 / 9Dh

D1 | IRAP : BUS -> RAP

RAP = 157 / 9Dh

C1 | RRC : Rozpoczęcie RRC

RBP = 10247 / 2807h

Takt7:

S3 | ORBP : RBP -> BUS

BUS = 10247 / 2807h

D3 | IRR : BUS -> RR

RR = 10247 / 2807h

RAPS = 49 / 31h

MAKRO

=====49=====

Takt0: RBPS=68C801830032h

Takt1:

S1 | ORR : RR -> BUS

BUS = 10247 / 2807h

D1 | ILK : BUS -> LK

LK = 7 / 7h

C2 | CEA : Oblicz adres efektywny

L = 7 / 7h

R = 0 / 0h

SUMA = 7 / 7h

XRO = 0

Takt6:

S2 | IRAE : SUMA -> RAE

RAE = 7 / 7h

D2 | NSI : LR+1 -> LR

LR = 158 / 9Eh

Takt7:

TEST | TIND : Adresowanie pośrednie

RAPS = 5 / 5h

RAPS = 5 -zmiana-> RAPS = 0

MAKRO

=====0=====

Takt0: RBPS=000000020030h

Takt7:

INT = 0

TEST | TINT : Brak przerwania(INT ?= 0)

RAPS = 48 / 30h

MAKRO

=====48=====

Takt0: RBPS=5006C4000000h

Takt1:

S1 | OLR : LR -> BUS

BUS = 158 / 9Eh

D1 | IRAP : BUS -> RAP

RAP = 158 / 9Eh

C1 | RRC : Rozpoczęcie RRC

RBP = 652 / 28Ch

Takt7:

S3 | ORBP : RBP -> BUS

BUS = 652 / 28Ch

D3 | IRR : BUS -> RR

RR = 652 / 28Ch

RAPS = 49 / 31h

MAKRO

=====49=====

Takt0: RBPS=68C801830032h

Takt1:

S1 | ORR : RR -> BUS

BUS = 652 / 28Ch

D1 | ILK : BUS -> LK

LK = 12 / Ch

C2 | CEA : Oblicz adres efektywny

L = 12 / Ch

R = 0 / 0h

SUMA = 12 / Ch

Takt6:

S2 | IRAE : SUMA -> RAE

RAE = 12 / Ch

D2 | NSI : LR+1 -> LR

LR = 159 / 9Fh

Takt7:

TEST | TIND : Adresowanie pośrednie

RAPS = 50 / 32h

MAKRO

=====50=====

Takt0: RBPS=900624000000h

Takt1:

S1 | ORAE : RAE -> BUS

BUS = 12 / Ch

D1 | IRAP : BUS -> RAP

RAP = 12 / Ch

C1 | RRC : Rozpoczęcie RRC

RBP = 255 / FFh

Takt7:

S3 | ORBP : RBP -> BUS

BUS = 255 / FFh

D3 | IX : BUS -> X

X = 255 / FFh

RAPS = 51 / 33h
 MAKRO
 =====51=====

Takt0: RBPS=03A801600000h
 Takt6:
 S2 | OX : X -> BUS
 BUS = 255 / FFh
 D2 | IBI : BUS -> RAE
 RAE = 255 / FFh

Takt7:
 C2 | OPC : OP albo AOP+32 -> RAPS

Błąd(1): RAPS = 32 / 20h (Poprawna RAPS = 37 / 25h)

MAKRO
 =====37=====

Takt0: RBPS=000000000000h

RAPS = 38 / 26h
 RAPS = 38 -zmiana-> RAPS = 0

MAKRO
 =====0=====

Takt0: RBPS=000000020030h
 Takt7:
 INT = 0
 TEST | TINT : Brak przerwania(INT != 0)

RAPS = 48 / 30h
 MAKRO
 =====48=====

Takt0: RBPS=5006C4000000h
 Takt1:
 S1 | OLR : LR -> BUS
 BUS = 159 / 9Fh
 D1 | IRAP : BUS -> RAP
 RAP = 159 / 9Fh
 C1 | RRC : Rozpoczęcie RRC
 RBP = -22265 / A907h

Takt7:
 S3 | ORBP : RBP -> BUS
 BUS = -22265 / A907h
 D3 | IRR : BUS -> RR
 RR = -22265 / A907h

RAPS = 49 / 31h
 MAKRO
 =====49=====

Takt0: RBPS=68C801830032h
 Takt1:
 S1 | ORR : RR -> BUS
 BUS = -22265 / A907h

D1 | ILK : BUS -> LK

LK = 7 / 7h

C2 | CEA : Oblicz adres efektywny

Błąd(2): L = -22265 / A907h (Poprawna L = 7 / 7h)

R = 0 / 0h

SUMA = 7 / 7h

XRO = 0

Takt6:

S2 | IRAE : SUMA -> RAE

RAE = 7 / 7h

D2 | NSI : LR+1 -> LR

LR = 160 / A0h

Takt7:

TEST | TIND : Adresowanie pośrednie

RAPS = 50 / 32h

MAKRO

=====50=====

Takt0: RBPS=900624000000h

Takt1:

S1 | ORAE : RAE -> BUS

BUS = 7 / 7h

D1 | IRAP : BUS -> RAP

RAP = 7 / 7h

C1 | RRC : Rozpoczęcie RRC

RBP = 207 / CFh

Takt7:

S3 | ORBP : RBP -> BUS

BUS = 207 / CFh

D3 | IX : BUS -> X

X = 207 / CFh

RAPS = 51 / 33h

MAKRO

=====51=====

Takt0: RBPS=03A8016000000h

Takt6:

S2 | OX : X -> BUS

BUS = 207 / CFh

D2 | IBI : BUS -> RAE

RAE = 207 / CFh

Takt7:

C2 | OPC : OP albo AOP+32 -> RAPS

Błąd(3): RAPS = 15 / Fh (Poprawna RAPS = 21 / 15h)

RAPS = 21 -zmiana-> RAPS = 0

MAKRO

=====0=====

Takt0: RBPS=000000020030h

Takt7:

INT = 0

TEST | TINT : Brak przerwania(INT != 0)

RAPS = 48 / 30h

MAKRO

=====48=====

Takt0: RBPS=5006C4000000h

Takt1:

S1 | OLR : LR -> BUS

BUS = 160 / A0h

D1 | IRAP : BUS -> RAP

RAP = 160 / A0h

C1 | RRC : Rozpoczęcie RRC

RBP = -9727 / DA01h

Takt7:

S3 | ORBP : RBP -> BUS

BUS = -9727 / DA01h

D3 | IRR : BUS -> RR

RR = -9727 / DA01h

RAPS = 49 / 31h

MAKRO

=====49=====

Takt0: RBPS=68C801830032h

Takt1:

S1 | ORR : RR -> BUS

BUS = -9727 / DA01h

D1 | ILK : BUS -> LK

LK = 1 / 1h

C2 | CEA : Oblicz adres efektywny

Błąd(4): L = -9727 / DA01h (Poprawna L = 1 / 1h)

R = 160 / A0h

SUMA = 161 / A1h

XRO = 0

Takt6:

S2 | IRAE : SUMA -> RAE

RAE = 161 / A1h

D2 | NSI : LR+1 -> LR

LR = 161 / A1h

Takt7:

TEST | TIND : Adresowanie pośrednie

RAPS = 27 / 1Bh

MAKRO

=====27=====

Takt0: RBPS=000000000000h

RAPS = 28 / 1Ch

RAPS = 28 -zmiana-> RAPS = 0

MAKRO

=====0=====

Takt0: RBPS=000000020030h

Takt7:

INT = 0

TEST | TINT : Brak przerwania(INT != 0)

RAPS = 48 / 30h

MAKRO

=====48=====

Takt0: RBPS=5006C4000000h

Takt1:

S1 | OLR : LR -> BUS

BUS = 161 / A1h

D1 | IRAP : BUS -> RAP

RAP = 161 / A1h

C1 | RRC : Rozpoczęcie RRC

RBP = 908 / 38Ch

Takt7:

S3 | ORBP : RBP -> BUS

BUS = 908 / 38Ch

D3 | IRR : BUS -> RR

RR = 908 / 38Ch

RAPS = 49 / 31h

MAKRO

=====49=====

Takt0: RBPS=68C801830032h

Takt1:

S1 | ORR : RR -> BUS

BUS = 908 / 38Ch

D1 | ILK : BUS -> LK

LK = 12 / Ch

C2 | CEA : Oblicz adres efektywny

Błąd(5): L = 908 / 38Ch (Poprawna L = 12 / Ch)

R = 0 / 0h

SUMA = 12 / Ch

Takt6:

S2 | IRAE : SUMA -> RAE

RAE = 12 / Ch

D2 | NSI : LR+1 -> LR

LR = 162 / A2h

Takt7:

TEST | TIND : Adresowanie pośrednie

RAPS = 50 / 32h

MAKRO

=====50=====

Takt0: RBPS=900624000000h

Takt1:

S1 | ORAE : RAE -> BUS

BUS = 12 / Ch

D1 | IRAP : BUS -> RAP

RAP = 12 / Ch

C1 | RRC : Rozpoczęcie RRC

Błąd(6): RBP = 252 / FCh (Poprawna RBP = 255 / FFh)

Takt7:

S3 | ORBP : RBP -> BUS

BUS = 255 / FFh

D3 | IX : BUS -> X

X = 255 / FFh

RAPS = 51 / 33h

MAKRO

=====51=====

Takt0: RBPS=03A801600000h

Takt6:

S2 | OX : X -> BUS

BUS = 255 / FFh

D2 | IBI : BUS -> RAE

RAE = 255 / FFh

Takt7:

C2 | OPC : OP albo AOP+32 -> RAPS

RAPS = 39 / 27h

20:08.17

=====Stop symulacji=====

Ocena: 3 Błędy: 6

ㅎㄷ

Opis działania programu:

Pierwszy Rozkaz:

Rozpoczynamy od fazy pobrania rozkazu, gdzie mikrorozkaz jest pobierany z komórki o adresie 0 w Pamięci Mikroprogramów. Sterowanie wyciąga zawartość komórki Pamięci Mikroprogramów o adresie, który jest określony przez RAPS (aktualnie jest to komórka o adresie zero). Następnie sprawdzamy, czy wystąpiło przerwanie. Jeśli nie, przeskakujemy do komórki 48 w Pamięci Mikroprogramów, ustawiając RAPS na 48. Teraz przechodzimy do części, która jest niewidoczna z perspektywy projektu mikroprogramu. Sterowanie musi pobrać zawartość komórki PM, wskazywanej przez bieżący RAPS, i zdekodować ją. Przekazujemy dane z LR do RAP, aby pobrać odpowiedni rozkaz. Następnie odczytujemy z komórki PAO o adresie wskazywanym przez RAP. W komórce tej znajduje się liczba binarna, którą odczytujemy i zapisujemy w RBP w systemie hexadecymalnym. Przekazujemy dane pobrane z PAO przetrzymywane w RBP i przekazujemy je do RR. Skończywszy mikrooperacje w aktualnym mikrorozkazie (RAPS = 48), zwiększamy RAPS o 1. Teraz 7 najmniej znaczących bitów BUS trafia do LK, jako efekt wykonania przesłania z RR. W tym momencie rozpoczyna się CEA. Oznacza to, że musimy rozpatrzyć typ rozkazu jaki został przez nas pobrany i zinterpretować działanie jego flag. To jest jedyny krok w trakcie pobierania różnego rodzaju rozkazów, który będzie się znacząco różnił. W tym przypadku jest to rozkaz zwykły, jego flagi wynoszą 000, a więc na a lewe wejście sumatora adresowego podajemy osiem najmniej znaczących bitów RR, a na prawe 0. Zapisujemy wynik dodawania obu wartości w polu SUMA i od razu przenosimy do rejestru RAE. Kolejnym krokiem jest obliczenie adresu następnego

rozkazu w PAO używając NSI, który zwiększa LR o 1. Następnie testujemy używając TIND, czy w rozkazie użyto trybu adresowania pośredniego. W naszym przypadku bit I = 0, co oznacza brak adresowania pośredniego, więc przekazujemy kod operacji rozkazu do RAPS, w naszym przypadku wynosi on 5. Zmieniamy RAPS manualnie na 0, aby przejść do kolejnego rozkazu.

Drugi Rozkaz:

Przy pobieraniu drugiego rozkazu cały proces zachodzi bez zmian, jednak przy operacji CEA musimy rozpatrzyć rozkaz rozszerzony. Oznacza to, że w prawej części sumatora będzie zero, a w lewej wartość hexadecymalna z RR. Kolejna różnica jest w momencie analizy rozkazu testu TIND. Mamy do czynienia z rozkazem adresowanym pośrednio, a więc LR zmieni się na 50. Następnie pobieramy informacje z RAE i przekazujemy ją do RAP. Rozpoczynamy ponownie RRC, a więc wartość RBP jest zmieniana na wartość hexadecymalną adresu PAO o numerze zamieszczonym w RAP. Przekazujemy zawartość RBP do X, po czym zwiększamy RAPS o 1 (51), ponieważ dotarliśmy do końca wiersza i nie ma na nim żadnych testów. Przekazujemy dane z X do RAE. Finalnie wykonujemy OPC, a więc ustawiamy LR równy wartości OP-kodu. W przypadku rozkazów rozszerzonych dodajemy 32 do danej wartości, w tym przypadku wynosi on 37. Jest to odpowiednik nie zdanej operacji TIND z wiersza 49.

Trzeci Rozkaz:

Trzeci rozkaz jest rozkazem zwykłym, z flagami 001. Oznacza to, że jest on rozkazem adresowanym pośrednio. Oznacza to, że przebędziemy taką samą drogę co rozkaz drugi. W pierwszej części RRC ustawi nam wartość na zawartość komórki PAO 159, wynoszącą A907h. Przy operacji CEA, lewa komórka będzie mieć wartość 7, a prawa 0 (ze względu na flage). RAE będzie równe 7. NSI zwiększy wartość do 160, a test TIND ustawi wartość RAPS na 50, przenosząc nas do „drugiej części”. RRC z 50 wiersza ustawi RBP na 207h. OPC ustawia RAPS na 21, po czym musimy ustawić ręcznie RAPS na 0, aby rozpocząć proces od nowa i pobrać kolejny rozkaz.

Czwarty Rozkaz:

Czwarty rozkaz jest rozkazem zwykłym, z flagami 010. Oznacza to, że jego pobieranie zakończy się po 2 wierszach, tak jak rozkazu pierwszego. RRC będzie pobrane z komórki 160, wartość pobrana to będzie DA01h. Do LK zostanie przekazana wartość 1. Po rozpoczęciu operacji CEA, do lewej komórki damy wartość 1. Do prawej przenosimy wartość z LR, która wynosi 160, a więc RAE będzie wynosić 161. NSI zwiększa zwiększy LR do 160. Test TIND sprawdzi, że nie posiadamy rozkazu rozszerzonego, a więc RAPS zostanie ustawiony na wartość OP kodu, a więc w tym przypadku 27.

Piąty Rozkaz:

Piąty rozkaz jest rozkazem rozszerzonym. Oznacza to, że jego pobieranie zakończy się w tym samym miejscu co rozkazu drugiego. RRC pobierze dane z komórki 161, będzie to wartość 908. W LK będziemy mieć wartość 12. Rozkaz CEA w lewej części będzie mieć wartość 12, a prawy 0, a więc RRC wynosi 12. NSI zwiększy wartość LR do 162. Test TIND będzie zdany, więc RAPS wynosi 50. RRC pobierze wartość z komórki 12, wyniesie ona 00FFh. OPC wyniesie 39 i tym samym zakończymy pobieranie 5 rozkazów.

Wyniki po pobraniu każdego z mikrorozkazów:

1. LR : 158, RAPS : 5 , RAE : 7
2. LR : 159, RAPS : 37, RAE : 255
3. LR : 160, RAPS : 21, RAE : 50
4. LR : 161, RAPS : 27, RAE : 161
5. LR : 162, RAPS : 39, RAE : 255