

Wojskowa Akademia Techniczna im. Jarosława Dąbrowskiego

Laboratorium z przedmiotu:
Architektura i organizacja komputerów

Sprawozdanie z ćwiczenia laboratoryjnego nr 2: **Mikroprogram pobrania rozkazu**

Spis treści

| | |
|---|----|
| Treść zadania | 2 |
| Wydruk zawartości PM | 2 |
| Wydruk zawartości PAO | 3 |
| Wydruk logu z wykonania ćwiczenia | 4 |
| Opis działania mikroprogramu | 17 |
| Rozkaz dla LR = 150 | 18 |
| Rozkaz dla LR = 151 | 18 |
| Rozkaz dla LR = 152 | 19 |
| Rozkaz dla LR = 153 | 19 |
| Rozkaz dla LR = 154 | 20 |
| Rozkaz dla LR = 155 | 21 |

Treść zadania

Lab2 WCY20IY4S1 zima 2021

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

| | |
|-----------------|------------------|
| Rejestry | |
| A | nr |
| LR | 140+nr |
| RI | 3 |
| | |
| | |
| PAO | |
| Adres | Zawartość |
| 0 | 140+nr |
| nr | 40+nr |
| nr+3 | 2021 |
| 100 | 255 |
| LR | DIV 000 nr |
| LR+1 | SUB 001 nr+3 |
| LR+2 | LCA nr+3 |
| LR+3 | BAN 101 nr |
| LR+4 | LOR 111 nr+3 |
| 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 przerwania

NA ___48

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 zawartości PAO

| | | | | | | |
|-----|-------------------|-------|-------|---------|-------|--|
| 0 | 0000000010010110b | 0096h | 150 | | | |
| 10 | 0000000000110010b | 0032h | 50 | | | |
| 13 | 0000011111100101b | 07E5h | 2021 | | | |
| 100 | 0000000011111111b | 00FFh | 255 | | | |
| 150 | 0010000000001010b | 200Ah | OP=4 | XSI=000 | DA=10 | |
| 151 | 0001000100001101b | 110Dh | OP=2 | XSI=001 | DA=13 | |
| 152 | 0000010000001101b | 040Dh | AOP=8 | | N=13 | |

| | | | | | |
|-----|-------------------|-------|-------|---------|-------|
| 153 | 1100010100001010b | C50Ah | OP=24 | XSI=101 | DA=10 |
| 154 | 1100111100001101b | CF0Dh | OP=25 | XSI=111 | DA=13 |
| 155 | 0000000100001010b | 010Ah | AOP=2 | | N=10 |
| 255 | 0000000000000011b | 0003h | 3 | | |

Wydruk logu z wykonania ćwiczenia

Start symulatora 03.11.2021 18:35:04

Stacja "DESKTOP-TA2A43K"

Zalogowano jako: "Radosław"

Wersja aplikacji: 1.2.3.0

Dostępne interfejsy sieciowe: 192.168.1.72

192.168.56.1

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

18:56.53

=====Zawartość rejestrów=====

| | | |
|------|-------|-----|
| LK | = 0h | 0 |
| A | = Ah | 10 |
| MQ | = 0h | 0 |
| X | = 0h | 0 |
| RAP | = 0h | 0 |
| LALU | = 0h | 0 |
| RALU | = 0h | 0 |
| RBP | = 0h | 0 |
| ALU | = 0h | 0 |
| BUS | = 0h | 0 |
| RR | = 0h | 0 |
| LR | = 96h | 150 |
| RI | = 3h | 3 |
| 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 = 150 / 96h

D1 | IRAP : BUS -> RAP

RAP = 150 / 96h

C1 | RRC : Rozpoczęcie RRC

RBP = 8202 / 200Ah

Takt7:

S3 | ORBP : RBP -> BUS

BUS = 8202 / 200Ah

D3 | IRR : BUS -> RR

RR = 8202 / 200Ah

RAPS = 49 / 31h

MAKRO

=====49=====

Takt0: RBPS=68C801830032h

Takt1:

S1 | ORR : RR -> BUS

BUS = 8202 / 200Ah

D1 | ILK : BUS -> LK

LK = 10 / Ah

C2 | CEA : Oblicz adres efektywny

L = 10 / Ah

R = 0 / 0h

SUMA = 10 / Ah

XRO = 0

Takt6:

S2 | IRAE : SUMA -> RAE

RAE = 10 / Ah

D2 | NSI : LR+1 -> LR

LR = 151 / 97h

Takt7:

TEST | TIND : Adresowanie pośrednie

RAPS = 4 / 4h

RAPS = 4 -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 = 151 / 97h

D1 | IRAP : BUS -> RAP

RAP = 151 / 97h

C1 | RRC : Rozpoczęcie RRC

RBP = 4365 / 110Dh

Takt7:

S3 | ORBP : RBP -> BUS

BUS = 4365 / 110Dh

D3 | IRR : BUS -> RR

RR = 4365 / 110Dh

RAPS = 49 / 31h

MAKRO

=====49=====

Takt0: RBPS=68C801830032h

Takt1:

S1 | ORR : RR -> BUS

BUS = 4365 / 110Dh

D1 | ILK : BUS -> LK

LK = 13 / Dh

C2 | CEA : Oblicz adres efektywny

L = 13 / Dh

R = 0 / 0h

SUMA = 13 / Dh

XRO = 0

Takt6:

S2 | IRAE : SUMA -> RAE

RAE = 13 / Dh

D2 | NSI : LR+1 -> LR

LR = 152 / 98h

Takt7:

TEST | TIND : Adresowanie pośrednie

RAPS = 50 / 32h

MAKRO

=====50=====

Takt0: RBPS=900624000000h

Takt1:

S1 | ORAE : RAE -> BUS

BUS = 13 / Dh

D1 | IRAP : BUS -> RAP

RAP = 13 / Dh

C1 | RRC : Rozpoczęcie RRC

RBP = 2021 / 7E5h

Takt7:

S3 | ORBP : RBP -> BUS

BUS = 2021 / 7E5h

D3 | IX : BUS -> X

X = 2021 / 7E5h

RAPS = 51 / 33h

MAKRO

=====51=====

Takt0: RBPS=03A801600000h

Takt6:

S2 | OX : X -> BUS

BUS = 2021 / 7E5h

D2 | IBI : BUS -> RAE

RAE = 2021 / 7E5h

Takt7:

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

RAPS = 2 / 2h

RAPS = 2 -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 = 152 / 98h

D1 | IRAP : BUS -> RAP

RAP = 152 / 98h

C1 | RRC : Rozpoczęcie RRC

RBP = 1037 / 40Dh

Takt7:

S3 | ORBP : RBP -> BUS

BUS = 1037 / 40Dh

D3 | IRR : BUS -> RR

RR = 1037 / 40Dh

RAPS = 49 / 31h

MAKRO

=====49=====

Takt0: RBPS=68C801830032h

Takt1:

S1 | ORR : RR -> BUS

BUS = 1037 / 40Dh

D1 | ILK : BUS -> LK

LK = 13 / Dh

C2 | CEA : Oblicz adres efektywny

L = 13 / Dh

R = 0 / 0h

SUMA = 13 / Dh

Takt6:

S2 | IRAE : SUMA -> RAE

RAE = 13 / Dh

D2 | NSI : LR+1 -> LR

LR = 153 / 99h

Takt7:

TEST | TIND : Adresowanie pośrednie

RAPS = 50 / 32h

MAKRO

=====50=====

Takt0: RBPS=900624000000h

Takt1:

S1 | ORAE : RAE -> BUS

BUS = 13 / Dh

D1 | IRAP : BUS -> RAP

RAP = 13 / Dh

C1 | RRC : Rozpoczęcie RRC

RBP = 2021 / 7E5h

Takt7:

S3 | ORBP : RBP -> BUS

BUS = 2021 / 7E5h

D3 | IX : BUS -> X

X = 2021 / 7E5h

RAPS = 51 / 33h

MAKRO

=====51=====

Takt0: RBPS=03A801600000h

Takt6:

S2 | OX : X -> BUS

BUS = 2021 / 7E5h

D2 | IBI : BUS -> RAE

RAE = 2021 / 7E5h

Takt7:

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

RAPS = 40 / 28h

MAKRO

=====40=====

Takt0: RBPS=000000000000h

RAPS = 41 / 29h

RAPS = 41 -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 = 153 / 99h

D1 | IRAP : BUS -> RAP

RAP = 153 / 99h

C1 | RRC : Rozpoczęcie RRC

RBP = -15094 / C50Ah

Takt7:

S3 | ORBP : RBP -> BUS

BUS = -15094 / C50Ah

D3 | IRR : BUS -> RR

RR = -15094 / C50Ah

RAPS = 49 / 31h

MAKRO

=====49=====

Takt0: RBPS=68C801830032h

Takt1:

S1 | ORR : RR -> BUS

BUS = -15094 / C50Ah

D1 | ILK : BUS -> LK

LK = 10 / Ah

C2 | CEA : Oblicz adres efektywny

L = 10 / Ah

R = 3 / 3h

SUMA = 13 / Dh

XRO = 0

Takt6:

S2 | IRAE : SUMA -> RAE

RAE = 13 / Dh

D2 | NSI : LR+1 -> LR

LR = 154 / 9Ah

Takt7:

TEST | TIND : Adresowanie pośrednie

RAPS = 50 / 32h

RAPS = 50 -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 = 154 / 9Ah

D1 | IRAP : BUS -> RAP

RAP = 154 / 9Ah

C1 | RRC : Rozpoczęcie RRC

RBP = -12531 / CF0Dh

Takt7:

S3 | ORBP : RBP -> BUS

BUS = -12531 / CF0Dh

D3 | IRR : BUS -> RR

RR = -12531 / CF0Dh

RAPS = 49 / 31h

MAKRO

=====49=====

Takt0: RBPS=68C801830032h

Takt1:

S1 | ORR : RR -> BUS

BUS = -12531 / CF0Dh

D1 | ILK : BUS -> LK

LK = 13 / Dh

C2 | CEA : Oblicz adres efektywny

L = 0 / 0h

R = 0 / 0h

SUMA = 0 / 0h

XRO = 1

Takt6:

S2 | IRAE : SUMA -> RAE

RAE = 0 / 0h

D2 | NSI : LR+1 -> LR

LR = 155 / 9Bh

Takt7:

TEST | TIND : Adresowanie pośrednie

RAPS = 50 / 32h

MAKRO

=====50=====

Takt0: RBPS=900624000000h

Takt1:

S1 | ORAE : RAE -> BUS

BUS = 0 / 0h

D1 | IRAP : BUS -> RAP

RAP = 0 / 0h

C1 | RRC : Rozpoczęcie RRC

RBP = 150 / 96h

Takt7:

S3 | ORBP : RBP -> BUS

BUS = 150 / 96h

D3 | IX : BUS -> X

X = 150 / 96h

RAPS = 51 / 33h

MAKRO

=====51=====

Takt0: RBPS=03A801600000h

Takt6:

S2 | OX : X -> BUS

BUS = 150 / 96h

D2 | IBI : BUS -> RAE

RAE = 150 / 96h

Takt7:

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

RAPS = 25 / 19h

RAPS = 25 -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 = 155 / 9Bh

D1 | IRAP : BUS -> RAP

RAP = 155 / 9Bh

C1 | RRC : Rozpoczęcie RRC

RBP = 266 / 10Ah

Takt7:

S3 | ORBP : RBP -> BUS

BUS = 266 / 10Ah

D3 | IRR : BUS -> RR

RR = 266 / 10Ah

RAPS = 49 / 31h

MAKRO

=====49=====

Takt0: RBPS=68C801830032h

Takt1:

S1 | ORR : RR -> BUS

BUS = 266 / 10Ah

D1 | ILK : BUS -> LK

LK = 10 / Ah

C2 | CEA : Oblicz adres efektywny

L = 10 / Ah

$R = 0 / 0h$

$SUMA = 10 / Ah$

Takt6:

S2 | IRAE : SUMA -> RAE

$RAE = 10 / Ah$

D2 | NSI : LR+1 -> LR

$LR = 156 / 9Ch$

Takt7:

TEST | TIND : Adresowanie pośrednie

Błąd(1): RAPS = 51 / 33h (Poprawna RAPS = 50 / 32h)

$RAPS = 50$ -zmiana-> $RAPS = 0$

❓

Opis działania mikroprogramu

Każde działanie w ramach pobrania danego rozkazu rozpoczyna się w momencie, kiedy rejestr adresowy pamięci stałe (RAPS) wnosi 0. W związku z tym zawsze rozpoczniemy od mikrorozkazu TINT sprawdzającego, czy nastąpiło przerwanie (symulator dla uproszczenia nigdy nie stwierdzi wystąpienia sygnału przerwania). Jeśli przerwanie nie nastąpiło, przechodzimy do 48 wiersza (taką wartość wpisujemy do RAPS). Od tego momentu rozpoczynają się opisy pobrania każdego z rozkazów. Po pobraniu każdego rozkazu wartość RAPS będzie ustawiana na 0, aby móc pobrać kolejny rozkaz.

Większość wartości będzie przekazywana poprzez magistralę (BUS). Operację przekazania wartości z jednego rejestru do drugiego za pośrednictwem magistrali będę opisywał jako jeden zestaw.

Rejestry wykorzystywane w ćwiczeniu (ich mnemoniki wraz z nazwą):

- A – Akumulator (przechowuje wyniki operacji)
- LK – Licznik Kroków
- LR – Licznik Rozkazów
- RAE – Rejestr Adresu Efektywnego
- RAP – Rejestr Adresowy pamięci
- RAPS – Rejestr Adresowy Pamięci Stałej
- RBP – Rejestr Buforowy Pamięci (na podstawie indeksu z RAP)
- RI – Rejestr Indeksowy (wartość modyfikująca adres rozkazu)
- RR – Rejestr rozkazów
- SUMA – Rejestr sumy adresowej (suma wartości rejestrów L oraz R)
- X – Rejestr danych (pomocniczy)

Kolejne działania przy pobieraniu rozkazu będą opisane w takiej konstrukcji:

Nr. Sygnał Sterujący: Definicja – działanie lub po prostu przekazywana wartość

Rozkaz dla LR = 150

1. OLR : LR -> BUS - 150 / 96h
2. IRAP : BUS -> RAP - 150 / 96h
3. RRC : Rozpoczęcie RRC – przekazanie do RBP wartości (8202 / 200Ah) na podstawie indeksu z RAP (150 / 96h)
4. ORBP : RBP -> BUS - 8202 / 200Ah
5. IRR : BUS -> RR - 8202 / 200Ah
6. (koniec komórki pamięci mikroprogramów, RAPS = 49 / 31h)
7. ORR : RR -> BUS - 8202 / 200Ah
8. ILK : BUS -> LK – przekazanie wartości na podstawie 7 najmniej znaczących bitów wartości magistrali (8202 / 200Ah). Wynika to z pojemności rejestru LK. W tym przypadku LK = 10 / Ah
9. CEA : Oblicz adres efektywny – mając do czynienia z rozkazem zwykłym o wartościach XSI odpowiednio 000, do rejestru L trafia wartość 10 / Ah, rejestr R pozostaje wyzerowany. Mając te 2 wartości dodajemy je do siebie, a uzyskaną wartość wpisujemy do rejestru SUMA (będzie to wartość 10 / Ah).
10. IRAE : SUMA -> RAE - 10 / Ah
11. NSI : LR+1 -> LR - 151 / 97h (zmieniamy licznik, aby być w stanie pobrać kolejny rozkaz)
12. TEST | TIND : Adresowanie pośrednie – sprawdzamy, czy wystąpiło adresowanie pośrednie. Robimy to poprzez dane dotyczące rozkazu. Jako, że jest to rozkaz zwykły, a bit I jest ustawiony na 0, oznacza to, że nie adresacja nie jest pośrednia. W związku z tym w RAPS wpisujemy wartość pola OP rozkazu, czyli 4 / 4h.

Rozkaz dla LR = 151

1. OLR : LR -> BUS - 151 / 97h
2. IRAP : BUS -> RAP - 151 / 97h
3. RRC : Rozpoczęcie RRC – przekazanie do RBP wartości (4365 / 110Dh) na podstawie indeksu z RAP (151 / 97h)
4. ORBP : RBP -> BUS - 4365 / 110Dh
5. IRR : BUS -> RR - 4365 / 110Dh
6. (koniec komórki pamięci mikroprogramów, RAPS = 49 / 31h)
7. ORR : RR -> BUS - 4365 / 110Dh
8. ILK : BUS -> LK – przekazanie wartości na podstawie 7 najmniej znaczących bitów wartości magistrali (4365 / 110Dh). Wynika to z pojemności rejestru LK. W tym przypadku LK = 13 / Dh
9. CEA : Oblicz adres efektywny – mając do czynienia z rozkazem zwykłym o wartościach XSI odpowiednio 001, do rejestru L trafia wartość 13 / Dh, rejestr R pozostaje wyzerowany. Mając te 2 wartości dodajemy je do siebie, a uzyskaną wartość wpisujemy do rejestru SUMA (będzie to wartość 13 / Dh).
10. IRAE : SUMA -> RAE - 13 / Dh
13. NSI : LR+1 -> LR - 152 / 98h (zmieniamy licznik, aby być w stanie pobrać kolejny rozkaz)
11. TEST | TIND : Adresowanie pośrednie – sprawdzamy, czy wystąpiło adresowanie pośrednie. Robimy to poprzez dane dotyczące rozkazu. Jako, że wartość I wynosi 1, mamy do czynienia z adresacją pośrednią. W związku z tym w RAPS wpisujemy wartość następnej komórki pamięci mikroprogramu, czyli 50 / 32h.
12. ORAE : RAE -> BUS - 13 / Dh
13. IRAP : BUS -> RAP - 13 / Dh

14. RRC : Rozpoczęcie RRC - przekazanie do RBP wartości (2021 / 7E5h) na podstawie indeksu z RAP (13 / Dh)
15. (koniec komórki pamięci mikroprogramów, RAPS = 51 / 33h)
16. OX : X -> BUS - 2021 / 7E5h
17. IBI : BUS -> RAE - 2021 / 7E5h
18. OPC : OP albo AOP+32 -> RAPS – w zależności od typu rozkazu ustawiamy nową wartość pola RAPS. W tym przypadku mając rozkaz zwykły podajemy wartość OP, którą określa 5 pierwszych bitów w 16-bitowym zapisie rozkazu. Oznacza to, że RAPS = 2 / 2h

Rozkaz dla LR = 152

1. OLR : LR -> BUS - 152 / 98h
2. IRAP : BUS -> RAP - 152 / 98h
3. RRC : Rozpoczęcie RRC – przekazanie do RBP wartości (1037 / 40Dh) na podstawie indeksu z RAP (152 / 98h)
4. ORBP : RBP -> BUS - 1037 / 40Dh
5. IRR : BUS -> RR - 1037 / 40Dh
6. (koniec komórki pamięci mikroprogramów, RAPS = 49 / 31h)
7. ORR : RR -> BUS - 1037 / 40Dh
8. ILK : BUS -> LK – przekazanie wartości na podstawie 7 najmniej znaczących bitów wartości magistrali (1037 / 40Dh). Wynika to z pojemności rejestru LK. W tym przypadku LK = 13 / Dh
9. CEA : Oblicz adres efektywny – mając do czynienia z rozkazem rozszerzonym, do rejestru L trafia wartość 13 / Dh, rejestr R pozostaje wyzerowany. Mając te 2 wartości dodajemy je do siebie, a uzyskaną wartość wpisujemy do rejestru SUMA (będzie to wartość 13 / Dh).
10. IRAE : SUMA -> RAE - 13 / Dh
11. NSI : LR+1 -> LR - 153 / 99h (zmieniamy licznik, aby być w stanie pobrać kolejny rozkaz)
12. TEST | TIND : Adresowanie pośrednie – sprawdzamy, czy wystąpiło adresowanie pośrednie. Robimy to poprzez dane dotyczące rozkazu. Jako, że mamy do czynienia z rozkazem rozszerzonym, jest to adresacja pośrednia. W związku z tym w RAPS wpisujemy wartość następnej komórki pamięci mikroprogramu, czyli 50 / 32h.
13. ORAE : RAE -> BUS - 13 / Dh
14. IRAP : BUS -> RAP - 13 / Dh
15. RRC : Rozpoczęcie RRC - przekazanie do RBP wartości (2021 / 7E5h) na podstawie indeksu z RAP (13 / Dh)
16. ORBP : RBP -> BUS - 2021 / 7E5h
17. IX : BUS -> X - 2021 / 7E5h
18. (koniec komórki pamięci mikroprogramów, RAPS = 51 / 33h)
19. OX : X -> BUS - 2021 / 7E5h
20. IBI : BUS -> RAE - 2021 / 7E5h
21. OPC : OP albo AOP+32 -> RAPS – w zależności od typu rozkazu ustawiamy nową wartość pola RAPS. W tym przypadku mając rozkaz rozszerzony podajemy wartość AOP powiększoną o 32. AOP odczytujemy na podstawie 5,6,7 oraz 8 bitu zapisu 16-bitowego rozkazu. Oznacza to, że RAPS = 40 / 28h

Rozkaz dla LR = 153

1. OLR : LR -> BUS - 153 / 99h
2. IRAP : BUS -> RAP - 153 / 99h
3. RRC : Rozpoczęcie RRC – przekazanie do RBP wartości (-15094 / C50Ah) na podstawie indeksu z RAP (153 / 99h)
4. ORBP : RBP -> BUS - -15094 / C50Ah

5. IRR : BUS -> RR - -15094 / C50Ah
6. (koniec komórki pamięci mikroprogramów, RAPS = 49 / 31h)
7. ORR : RR -> BUS - -15094 / C50Ah
8. ILK : BUS -> LK – przekazanie wartości na podstawie 7 najmniej znaczących bitów wartości magistrali (-15094 / C50Ah). Wynika to z pojemności rejestru licznika kroków. W tym przypadku LK = 10 / Ah.
9. CEA : Oblicz adres efektywny – mając do czynienia z rozkazem zwykłym o wartościach XSI odpowiednio 101, do rejestru L trafia wartość 10 / Ah, a do rejestru R wartość rejestru RI, czyli 3 / 3h. Mając te 2 wartości dodajemy je do siebie, a uzyskaną wartość wpisujemy do rejestru SUMA (będzie to wartość 13 / Dh).
10. IRAE : SUMA -> RAE - 13 / Dh
11. NSI : LR+1 -> LR - 154 / 9Ah (zmieniamy licznik, aby być w stanie pobrać kolejny rozkaz)
12. TEST | TIND : Adresowanie pośrednie – sprawdzamy, czy wystąpiło adresowanie pośrednie. Robimy to poprzez dane dotyczące rozkazu. Jako, że wartość I wynosi 1, mamy do czynienia z adresacją pośrednią. W związku z tym w RAPS wpisujemy wartość następnej komórki pamięci mikroprogramu, czyli 50 / 32h.

Rozkaz dla LR = 154

1. OLR : LR -> BUS - 154 / 9Ah
2. IRAP : BUS -> RAP - 154 / 9Ah
3. RRC : Rozpoczęcie RRC – przekazanie do RBP wartości (-12531 / CF0Dh) na podstawie indeksu z RAP (154 / 9Ah)
4. ORBP : RBP -> BUS - -12531 / CF0Dh
5. IRR : BUS -> RR - -12531 / CF0Dh
6. (koniec komórki pamięci mikroprogramów, RAPS = 49 / 31h)
7. ORR : RR -> BUS - -12531 / CF0Dh
8. ILK : BUS -> LK – przekazanie wartości na podstawie 7 najmniej znaczących bitów wartości magistrali (-12531 / CF0Dh). Wynika to z pojemności rejestru licznika kroków. W tym przypadku LK = 13 / Dh.
9. CEA : Oblicz adres efektywny – mając do czynienia z rozkazem zwykłym o wartościach XSI odpowiednio 111, rejestry L oraz R są wyzerowane. Dodając je do siebie, SUMA będzie miała wartość 0 / 0h.
10. IRAE : SUMA -> RAE - 0 / 0h
11. NSI : LR+1 -> LR - 155 / 9Bh (zmieniamy licznik, aby być w stanie pobrać kolejny rozkaz)
12. TEST | TIND : Adresowanie pośrednie – sprawdzamy, czy wystąpiło adresowanie pośrednie. Robimy to poprzez dane dotyczące rozkazu. Jako, że wartość I wynosi 1, mamy do czynienia z adresacją pośrednią. W związku z tym w RAPS wpisujemy wartość następnej komórki pamięci mikroprogramu, czyli 50 / 32h.
13. ORAE : RAE -> BUS - 0 / 0h
14. IRAP : BUS -> RAP - 0 / 0h
15. RRC : Rozpoczęcie RRC - przekazanie do RBP wartości (150 / 96h) na podstawie indeksu z RAP (0 / 0h)
16. ORBP : RBP -> BUS - 150 / 96h
17. IX : BUS -> X - 150 / 96h
18. (koniec komórki pamięci mikroprogramów, RAPS = 51 / 33h)
19. OX : X -> BUS - 150 / 96h
20. IBI : BUS -> RAE - 150 / 96h

21. OPC : OP albo AOP+32 -> RAPS – w zależności od typu rozkazu ustawiamy nową wartość pola RAPS. W tym przypadku mając rozkaz zwykły podajemy wartość OP, którą określa 5 pierwszych bitów w 16-bitowym zapisie rozkazu. Oznacza to, że RAPS = 25 / 19h

Rozkaz dla LR = 155

1. OLR : LR -> BUS - 155 / 9Bh
2. IRAP : BUS -> RAP - 155 / 9Bh
3. RRC : Rozpoczęcie RRC – przekazanie do RBP wartości (266 / 10Ah) na podstawie indeksu z RAP (155 / 9Bh)
4. ORBP : RBP -> BUS - 266 / 10Ah
5. IRR : BUS -> RR - 266 / 10Ah
6. (koniec komórki pamięci mikroprogramów, RAPS = 49 / 31h)
7. ORR : RR -> BUS - 266 / 10Ah
8. ILK : BUS -> LK – przekazanie wartości na podstawie 7 najmniej znaczących bitów wartości magistrali (266 / 10Ah). Wynika to z pojemności rejestru LK. W tym przypadku LK = 13 / Dh
9. CEA : Oblicz adres efektywny – mając do czynienia z rozkazem rozszerzonym, do rejestru L trafia wartość 10 / Ah, rejestr R pozostaje wyzerowany. Mając te 2 wartości dodajemy je do siebie, a uzyskaną wartość wpisujemy do rejestru SUMA (będzie to wartość 10 / Ah).
10. IRAE : SUMA -> RAE - 10 / Ah
11. NSI : LR+1 -> LR - 156 / 9Ch (wpisujemy pomimo faktu, że nie będziemy nic z tym robić)
12. TEST | TIND : Adresowanie pośrednie – sprawdzamy, czy wystąpiło adresowanie pośrednie. Robimy to poprzez dane dotyczące rozkazu. Jako, że mamy do czynienia z rozkazem rozszerzonym, jest to adresacja pośrednia. W związku z tym w RAPS wpisujemy wartość następnej komórki pamięci mikroprogramu, czyli 50 / 32h.