

Wojskowa Akademia Techniczna im. Jarosława Dąbrowskiego

Laboratorium z przedmiotu:
Architektura i organizacja komputerów

Sprawozdanie z ćwiczenia laboratoryjnego nr 4: Mikroprogramy rozkazów logicznych i skoków

Spis treści

Treść zadania	2
Wydruk zawartości PM	3
Wydruk zawartości PAO	18
Wydruk logu z wykonania ćwiczenia	18
Opis działania mikroprogramu	40
Rozkaz dla LR = 90 (rozkaz ADD 101 10)	40
Rozkaz dla LR = 91 (rozkaz LLA 5)	42
Rozkaz dla LR = 92 (rozkaz LPR(LAND) 010 2)	43
Rozkaz dla LR = 93 (rozkaz BAN 111 10)	45
Rozkaz dla LR = 90 (rozkaz ADD 101 10) – ponowne wykonanie.....	46

Treść zadania

Lab4_IY4

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

Rejestry	
A	-(32400+nr)
LR	80+nr
RI	10+nr
MQ	444
PAO	
Adres	Zawartość
0	LR
10+nr+nr	255
LR	ADD 101 nr
LR+1	LLA 5
LR+2	LPR(LAND) 010 2
LR+3	BAN 111 nr
LR+4	AAAAh
255	-(29700)

Pozostałe komórki PAO są wyzerowane.

Uwaga rozkaz w formacie zwykłym o kodzie OP=26 jest opisany w Ziarko jako LPR (stary kod z poprzedniej wersji programu), ale w LabZSK przy edycji komórki kod 26 jest opisany mnemonikiem LAND czyli logiczny AND.

Stopień trudności zadania:

- Na dostatecznie – poprawnie pobrać i wykonać 3 rozkazy, począwszy od PAO[LR].
- Na dobrze – poprawnie pobrać i wykonać 4 rozkazy, począwszy od PAO[LR].
- Na bardzo dobrze – poprawnie pobrać i wykonać 5 rozkazów, począwszy od PAO[LR].

Pozostałe komórki PAO są wyzerowane.

W Pamięci Mikroprogramów mają być wpisane do wytworzenia sprawozdania (najlepiej przed zajęciami, ale niekoniecznie) mikroprogramy, realizujące **wszystkie rozkazy z grup, objętych tematyką poprzednich i dzisiejszych zajęć** (bez mnożenia i dzielenia oraz pozostałych z zestawu: MUL, DIV, SIO, LIO, BDN, CND, ENI, LDS),

Brak kompletnej PM dla bieżących grup rozkazów w sprawozdaniu **oznacza pół oceny w dół** - nie dotyczy: MUL, DIV, SIO, LIO, BDN, CND, ENI, LDS.

Uwaga: w trakcie tego ćwiczenia **nie wolno edytować RAPS na zero** po zakończeniu pobierania każdego rozkazu.

Niepoprawne (niezgodne z definicją z listy rozkazów) działanie któregoś z rozkazów z grupy na dst oznacza po wykonaniu innych wymagań ocenę ndst.

Wydruk zawartości PM

- 0 Test ___TINT Brak przerwania
 NA ___48
- 1 Test ___UNB Zawsze pozytywny
 NA ___52
- 2 Test ___UNB Zawsze pozytywny
 NA ___54
- 5 Test ___UNB Zawsze pozytywny
 NA ___56
- 6 Test ___UNB Zawsze pozytywny
 NA ___58
- 7 Test ___UNB Zawsze pozytywny
 NA ___60

8 Test ____UNB Zawsze pozytywny
NA ____62

9 Test ____UNB Zawsze pozytywny
NA ____64

10 Test ____UNB Zawsze pozytywny
NA ____66

11 Test ____UNB Zawsze pozytywny
NA ____68

12 Test ____UNB Zawsze pozytywny
NA ____69

13 Test ____UNB Zawsze pozytywny
NA ____70

16 Test ____UNB Zawsze pozytywny
NA ____72

17 Test ____UNB Zawsze pozytywny
NA ____74

18 Test ____UNB Zawsze pozytywny
NA ____76

19 Test ____UNB Zawsze pozytywny
NA ____78

20 Test ____UNB Zawsze pozytywny
NA ____80

21 Test ____UNB Zawsze pozytywny
NA ____82

22 Test ____UNB Zawsze pozytywny
NA ____86

23 Test ____UNB Zawsze pozytywny
NA ____88

24 Test ____UNB Zawsze pozytywny
NA ____90

25 Test ____UNB Zawsze pozytywny
NA ____94

26 Test ____UNB Zawsze pozytywny
NA ____96

27 Test ____UNB Zawsze pozytywny
NA ____100

28 Test ____UNB Zawsze pozytywny
NA ____102

29 Test ____UNB Zawsze pozytywny
NA ____104

33 Test ____UNB Zawsze pozytywny

NA ____106

34 Test ____UNB Zawsze pozytywny

NA ____108

35 Test ____UNB Zawsze pozytywny

NA ____112

36 Test ____UNB Zawsze pozytywny

NA ____114

37 Test ____UNB Zawsze pozytywny

NA ____116

38 Test ____UNB Zawsze pozytywny

NA ____118

39 Test ____UNB Zawsze pozytywny

NA ____120

40 Test ____UNB Zawsze pozytywny

NA ____122

41 Test ____UNB Zawsze pozytywny

NA ____124

42 Test ____UNB Zawsze pozytywny

NA ____125

43 Test ____UNB Zawsze pozytywny

NA ____126

44 Test ___UNB Zawsze pozytywny
NA ___128

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

52 S1 ___ORAE RAE -> BUS
D1 ___IRAP BUS -> RAP
S3 ___ORBP RBP -> BUS

D3 ___IX BUS -> X

C1 ___RRC Rozpoczęcie RRC

53 S1 ___IALU A -> LALU

D1 ___OXE X -> RALU

S2 ___OBE ALU -> BUS

D2 ___IA BUS -> A

C1 ___END Koniec mikroprogramu

ALU ___ADD ALU = LALU + RALU

54 S1 ___ORAE RAE -> BUS

D1 ___IRAP BUS -> RAP

S3 ___ORBP RBP -> BUS

D3 ___IX BUS -> X

C1 ___RRC Rozpoczęcie RRC

55 S1 ___IALU A -> LALU

D1 ___OXE X -> RALU

S2 ___OBE ALU -> BUS

D2 ___IA BUS -> A

C1 ___END Koniec mikroprogramu

ALU ___SUB ALU = LALU - RALU

56 S1 ___ORAE RAE -> BUS

D1 ___IRAP BUS -> RAP

S3 ___OMQ MQ -> BUS

D3 ___IRBP BUS -> RBP

C1 ___CWC Rozpoczęcie CWC

57 C1 ___END Koniec mikroprogramu

58 S1 ___ORAE RAE -> BUS
D1 ___IRAP BUS -> RAP
S3 ___OA A -> BUS
D3 ___IRBP BUS -> RBP
C1 ___CWC Rozpoczęcie CWC

59 C1 ___END Koniec mikroprogramu

60 S1 ___ORAE RAE -> BUS
D1 ___IRAP BUS -> RAP
S3 ___ORI RI -> BUS
D3 ___IRBP BUS -> RBP
C1 ___CWC Rozpoczęcie CWC

61 C1 ___END Koniec mikroprogramu

62 S1 ___ORAE RAE -> BUS
D1 ___IRAP BUS -> RAP
S3 ___ORBP RBP -> BUS
D3 ___IA BUS -> A
C1 ___RRC Rozpoczęcie RRC

63 C1 ___END Koniec mikroprogramu

64 S1 ___ORAE RAE -> BUS
D1 ___IRAP BUS -> RAP
S3 ___ORBP RBP -> BUS
D3 ___IRI BUS -> RI
C1 ___RRC Rozpoczęcie RRC

65 C1 ___END Koniec mikroprogramu

66 S1 ___ORAE RAE -> BUS
 D1 ___IRAP BUS -> RAP
 S3 ___OLR LR -> BUS
 D3 ___IRBP BUS -> RBP
 C1 ___CWC Rozpoczęcie CWC

67 C1 ___END Koniec mikroprogramu

68 S2 ___ORI RI -> BUS
 D2 ___IA BUS -> A
 C1 ___END Koniec mikroprogramu

69 S2 ___OMQ MQ -> BUS
 D2 ___IA BUS -> A
 C1 ___END Koniec mikroprogramu

70 S1 ___ORAE RAE -> BUS
 D1 ___IRAP BUS -> RAP
 S3 ___ORBP RBP -> BUS
 D3 ___IX BUS -> X
 C1 ___RRC Rozpoczęcie RRC

71 S1 ___IXRE RI -> LALU
 D1 ___OXE X -> RALU
 S2 ___OBE ALU -> BUS
 D2 ___IRI BUS -> RI
 C1 ___END Koniec mikroprogramu
 ALU ___ADD ALU = LALU + RALU

72 S3 ___ORAE RAE -> BUS

D3 ___ILR BUS -> LR

C1 ___END Koniec mikroprogramu

74 Test ___TAO OFF = 0

NA ___110

75 Test ___UNB Zawsze pozytywny

NA ___16

76 Test ___TXP RI <= 0

NA ___110

77 Test ___UNB Zawsze pozytywny

NA ___16

78 Test ___TXZ BXZ i RI != 0 || TLD i RI = 0

NA ___16

79 C1 ___END Koniec mikroprogramu

80 Test ___TXS RI >= 0

NA ___110

81 Test ___UNB Zawsze pozytywny

NA ___16

82 Test ___TXP RI <= 0

NA ___110

83 C2 ___DRI RI = RI-1

Test ___UNB Zawsze pozytywny

NA ____16

86 Test ____TAP $A \leq 0$

NA ____110

87 Test ____UNB Zawsze pozytywny

NA ____16

88 Test ____TAZ $A = 0$

NA ____16

89 C1 ____END Koniec mikroprogramu

90 Test ____TAS $A \geq 0$

NA ____110

91 Test ____UNB Zawsze pozytywny

NA ____16

94 S1 ____ORAE RAE -> BUS

D1 ____IRAP BUS -> RAP

S3 ____ORBP RBP -> BUS

D3 ____IX BUS -> X

C1 ____RRC Rozpoczęcie RRC

95 S1 ____IALU A -> LALU

D1 ____OXE X -> RALU

S2 ____OBE ALU -> BUS

D2 ____IA BUS -> A

C1 ____END Koniec mikroprogramu

ALU ____OR ALU = LALU OR RALU

96 S1 ___ORAE RAE -> BUS
 D1 ___IRAP BUS -> RAP
 S3 ___ORBP RBP -> BUS
 D3 ___IX BUS -> X
 C1 ___RRC Rozpoczęcie RRC

97 S1 ___IALU A -> LALU
 D1 ___OXE X -> RALU
 S2 ___OBE ALU -> BUS
 D2 ___IA BUS -> A
 C1 ___END Koniec mikroprogramu
 ALU ___AND ALU = LALU AND RALU

98 S3 ___OLR LR -> BUS
 D3 ___IX BUS -> X

99 S1 ___OXE X -> RALU
 S2 ___OBE ALU -> BUS
 D2 ___ILR BUS -> LR
 C1 ___END Koniec mikroprogramu
 ALU ___DECR ALU = RALU - 1

100 S1 ___IALU A -> LALU
 S2 ___OBE ALU -> BUS
 D2 ___IA BUS -> A
 C1 ___END Koniec mikroprogramu
 ALU ___NOTL ALU = NOT LALU

102 S1 ___ORAE RAE -> BUS
 D1 ___IRAP BUS -> RAP

S3 ___ORBP RBP -> BUS
D3 ___IX BUS -> X
C1 ___RRC Rozpoczęcie RRC

103 S1 ___IALU A -> LALU
 D1 ___OXE X -> RALU
 S2 ___OBE ALU -> BUS
 D2 ___IA BUS -> A
 C1 ___END Koniec mikroprogramu
 ALU ___EOR ALU = LALU XOR RALU

104 S1 ___ORAE RAE -> BUS
 D1 ___IRAP BUS -> RAP
 S3 ___OLR LR -> BUS
 D3 ___IRBP BUS -> RBP
 C1 ___CWC Rozpoczęcie CWC

105 S2 ___ORAE RAE -> BUS
 D2 ___ILR BUS -> LR
 D3 ___NSI LR+1 -> LR
 C1 ___END Koniec mikroprogramu

106 S1 ___IALU A -> LALU
 S2 ___OBE ALU -> BUS
 D2 ___IA BUS -> A
 C1 ___END Koniec mikroprogramu
 ALU ___CMA ALU = (NOT LALU)+1

108 D2 ___ALA arytmetyczne A w lewo
 C1 ___SHT Operacja przesunięcia
 C2 ___DLK LK = [LK]-1

Test ___TLK SHT, LK=0 || !SHT, LK!=0

NA ___110

109 Test ___UNB Zawsze pozytywny

NA ___108

110 C1 ___END Koniec mikroprogramu

112 D2 ___ARA arytmetyczne A w prawo

C1 ___SHT Operacja przesunięcia

C2 ___DLK $LK = [LK] - 1$

Test ___TLK SHT, LK=0 || !SHT, LK!=0

NA ___110

113 Test ___UNB Zawsze pozytywny

NA ___112

114 D2 ___LRQ logiczne A i MQ w prawo

C1 ___SHT Operacja przesunięcia

C2 ___DLK $LK = [LK] - 1$

Test ___TLK SHT, LK=0 || !SHT, LK!=0

NA ___110

115 Test ___UNB Zawsze pozytywny

NA ___114

116 D2 ___LLQ logiczne A i MQ w lewo

C1 ___SHT Operacja przesunięcia

C2 ___DLK $LK = [LK] - 1$

Test ___TLK SHT, LK=0 || !SHT, LK!=0

NA ___110

117 Test ___UNB Zawsze pozytywny
NA ___116

118 D2 ___LLA logiczne A w lewo
C1 ___SHT Operacja przesunięcia
C2 ___DLK $LK = [LK] - 1$
Test ___TLK $SHT, LK=0 \mid \mid !SHT, LK!=0$
NA ___110

119 Test ___UNB Zawsze pozytywny
NA ___118

120 D2 ___LRA logiczne A w prawo
C1 ___SHT Operacja przesunięcia
C2 ___DLK $LK = [LK] - 1$
Test ___TLK $SHT, LK=0 \mid \mid !SHT, LK!=0$
NA ___110

121 Test ___UNB Zawsze pozytywny
NA ___120

122 D2 ___LCA cykliczne A w lewo
C1 ___SHT Operacja przesunięcia
C2 ___DLK $LK = [LK] - 1$
Test ___TLK $SHT, LK=0 \mid \mid !SHT, LK!=0$
NA ___110

123 Test ___UNB Zawsze pozytywny
NA ___122

124 S2 ___IRAE SUMA -> RAE
S3 ___ORAE RAE -> BUS
D3 ___IA BUS -> A
C1 ___END Koniec mikroprogramu

125 S2 ___IRAE SUMA -> RAE
S3 ___ORAE RAE -> BUS
D3 ___IRI BUS -> RI
C1 ___END Koniec mikroprogramu

126 S2 ___IRAE SUMA -> RAE
S3 ___ORAE RAE -> BUS
D3 ___IX BUS -> X

127 S1 ___IXRE RI -> LALU
D1 ___OXE X -> RALU
S2 ___OBE ALU -> BUS
D2 ___IRI BUS -> RI
C1 ___END Koniec mikroprogramu
ALU ___ADD $ALU = LALU + RALU$

128 S2 ___IRAE SUMA -> RAE
S3 ___ORAE RAE -> BUS
D3 ___IX BUS -> X

129 S1 ___IXRE RI -> LALU
D1 ___OXE X -> RALU
S2 ___OBE ALU -> BUS
D2 ___IRI BUS -> RI
C1 ___END Koniec mikroprogramu
ALU ___SUB $ALU = LALU - RALU$

Wydruk zawartości PAO

0	0000000001011010b	005Ah	90			
30	0000000011111111b	00FFh	255			
90	0000110100001010b	0D0Ah	OP=1	XSI=101	DA=10	
91	0000001100000101b	0305h	AOP=6		N=5	
92	1101001000000010b	D202h	OP=26	XSI=010	DA=2	
93	1100011100001010b	C70Ah	OP=24	XSI=111	DA=10	
94	1010101010101010b	AAAAh	-21846			
255	1000101111111100b	8BFCh	-29700			

Wydruk logu z wykonania ćwiczenia

Start symulatora 17.11.2021 14:19:46

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=====

14:55.23

=====Zawartość rejestrów=====

LK = 0h 0

A = 8166h -32410

MQ = 1BCh 444

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 = 5Ah 90
RI = 14h 20
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 = 90 / 5Ah

D1 | IRAP : BUS -> RAP

RAP = 90 / 5Ah

C1 | RRC : Rozpoczęcie RRC

RBP = 3338 / D0Ah

Takt7:

S3 | ORBP : RBP -> BUS

BUS = 3338 / D0Ah

D3 | IRR : BUS -> RR

RR = 3338 / D0Ah

RAPS = 49 / 31h

MAKRO

=====49=====

Takt0: RBPS=68C801830032h

Takt1:

S1 | ORR : RR -> BUS

BUS = 3338 / D0Ah

D1 | ILK : BUS -> LK

LK = 10 / Ah

C2 | CEA : Oblicz adres efektywny

L = 10 / Ah

R = 20 / 14h

SUMA = 30 / 1Eh

XRO = 0

Takt6:

S2 | IRAE : SUMA -> RAE

RAE = 30 / 1Eh

D2 | NSI : LR+1 -> LR

LR = 91 / 5Bh

Takt7:

TEST | TIND : Adresowanie pośrednie

RAPS = 50 / 32h

MAKRO

=====50=====

Takt0: RBPS=900624000000h

Takt1:

S1 | ORAE : RAE -> BUS

BUS = 30 / 1Eh

D1 | IRAP : BUS -> RAP

RAP = 30 / 1Eh

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=03A8016000000h

Takt6:

S2 | OX : X -> BUS

BUS = 255 / FFh

D2 | IBI : BUS -> RAE

RAE = 255 / FFh

Takt7:

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

RAPS = 1 / 1h

MAKRO

=====1=====

Takt0: RBPS=000000010034h

Takt7:

TEST | UNB : Zawsze pozytywny

RAPS = 52 / 34h

MAKRO

=====52=====

Takt0: RBPS=900624000000h

Takt1:

S1 | ORAE : RAE -> BUS

BUS = 255 / FFh

D1 | IRAP : BUS -> RAP

RAP = 255 / FFh

C1 | RRC : Rozpoczęcie RRC

RBP = -29700 / 8BFCh

Takt7:

S3 | ORBP : RBP -> BUS

BUS = -29700 / 8BFCh

D3 | IX : BUS -> X

X = -29700 / 8BFCh

RAPS = 53 / 35h

MAKRO

=====53=====

Takt0: RBPS=BC300E000100h

Takt1:

S1 | IALU : A -> LALU

LALU = -32410 / 8166h

D1 | OXE : X -> RALU

RALU = -29700 / 8BFCh

Takt2:

ALU | ADD : ALU = LALU + RALU

ALU = 3426 / D62h

ZNAK = 0, OFF = 1

Takt6:

S2 | OBE : ALU -> BUS

BUS = 3426 / D62h

D2 | IA : BUS -> A

A = 3426 / D62h

Takt7:

C1 | END : (Cykl 8) Koniec mikroprogramu (15:00.06)

RAPS = 0 / 0h

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 = 91 / 5Bh

D1 | IRAP : BUS -> RAP

RAP = 91 / 5Bh

C1 | RRC : Rozpoczęcie RRC

RBP = 773 / 305h

Takt7:

S3 | ORBP : RBP -> BUS

BUS = 773 / 305h

D3 | IRR : BUS -> RR

RR = 773 / 305h

RAPS = 49 / 31h

MAKRO

=====49=====

Takt0: RBPS=68C801830032h

Takt1:

S1 | ORR : RR -> BUS

BUS = 773 / 305h

D1 | ILK : BUS -> LK

LK = 5 / 5h

C2 | CEA : Oblicz adres efektywny

L = 5 / 5h

R = 0 / 0h

SUMA = 5 / 5h

Takt6:

S2 | IRAE : SUMA -> RAE

RAE = 5 / 5h

D2 | NSI : LR+1 -> LR

LR = 92 / 5Ch

Takt7:

TEST | TIND : Adresowanie pośrednie

RAPS = 50 / 32h

MAKRO

=====50=====

Takt0: RBPS=900624000000h

Takt1:

S1 | ORAE : RAE -> BUS

BUS = 5 / 5h

D1 | IRAP : BUS -> RAP

RAP = 5 / 5h

C1 | RRC : Rozpoczęcie RRC

RBP = 0 / 0h

Takt7:

S3 | ORBP : RBP -> BUS

BUS = 0 / 0h

D3 | IX : BUS -> X

X = 0 / 0h

RAPS = 51 / 33h

MAKRO

=====51=====

Takt0: RBPS=03A801600000h

Takt6:

S2 | OX : X -> BUS

BUS = 0 / 0h

D2 | IBI : BUS -> RAE

RAE = 0 / 0h

Takt7:

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

RAPS = 38 / 26h

MAKRO

=====38=====

Takt0: RBPS=000000010076h

Takt7:

TEST | UNB : Zawsze pozytywny

RAPS = 118 / 76h

MAKRO

=====118=====

Takt0: RBPS=00800A27006Eh

Takt1:

D2 | LLA : logiczne A w lewo

C1 | SHT : Operacja przesunięcia

A = 6852 / 1AC4h

Takt6:

C2 | DLK : LK = [LK]-1

LK = 4 / 4h

Takt7:

TEST | TLK : SHT, LK=0 | | !SHT, LK!=0

RAPS = 119 / 77h

MAKRO

=====119=====

Takt0: RBPS=000000010076h

Takt7:

TEST | UNB : Zawsze pozytywny

RAPS = 118 / 76h

MAKRO

=====118=====

Takt0: RBPS=00800A27006Eh

Takt1:

D2 | LLA : logiczne A w lewo

C1 | SHT : Operacja przesunięcia

A = 13704 / 3588h

Takt6:

C2 | DLK : LK = [LK]-1

LK = 3 / 3h

Takt7:

TEST | TLK : SHT, LK=0 | | !SHT, LK!=0

RAPS = 119 / 77h

MAKRO

=====119=====

Takt0: RBPS=000000010076h

Takt7:

TEST | UNB : Zawsze pozytywny

RAPS = 118 / 76h

MAKRO

=====118=====

Takt0: RBPS=00800A27006Eh

Takt1:

D2 | LLA : logiczne A w lewo

C1 | SHT : Operacja przesunięcia

A = 27408 / 6B10h

Takt6:

C2 | DLK : LK = [LK]-1

LK = 2 / 2h

Takt7:

TEST | TLK : SHT, LK=0 | | !SHT, LK!=0

RAPS = 119 / 77h

MAKRO

=====119=====

Takt0: RBPS=000000010076h

Takt7:

TEST | UNB : Zawsze pozytywny

RAPS = 118 / 76h

MAKRO

=====118=====

Takt0: RBPS=00800A27006Eh

Takt1:

D2 | LLA : logiczne A w lewo

C1 | SHT : Operacja przesunięcia

A = -10720 / D620h

Takt6:

C2 | DLK : LK = [LK]-1

LK = 1 / 1h

Takt7:

TEST | TLK : SHT, LK=0 | | !SHT, LK!=0

RAPS = 119 / 77h

MAKRO

=====119=====

Takt0: RBPS=000000010076h

Takt7:

TEST | UNB : Zawsze pozytywny

RAPS = 118 / 76h

MAKRO

=====118=====

Takt0: RBPS=00800A27006Eh

Takt1:

D2 | LLA : logiczne A w lewo

C1 | SHT : Operacja przesunięcia

A = -21440 / AC40h

Takt6:

C2 | DLK : LK = [LK]-1

LK = 0 / 0h

Takt7:

TEST | TLK : SHT, LK=0 | | !SHT, LK!=0

RAPS = 110 / 6Eh

MAKRO

=====110=====

Takt0: RBPS=00000E000000h

Takt7:

C1 | END : (Cykl 24) Koniec mikroprogramu (15:09.26)

RAPS = 0 / 0h

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 = 92 / 5Ch

D1 | IRAP : BUS -> RAP

RAP = 92 / 5Ch

C1 | RRC : Rozpoczęcie RRC

RBP = -11774 / D202h

Takt7:

S3 | ORBP : RBP -> BUS

BUS = -11774 / D202h

D3 | IRR : BUS -> RR

RR = -11774 / D202h

RAPS = 49 / 31h

MAKRO

=====49=====

Takt0: RBPS=68C801830032h

Takt1:

S1 | ORR : RR -> BUS

BUS = -11774 / D202h

D1 | ILK : BUS -> LK

LK = 2 / 2h

C2 | CEA : Oblicz adres efektywny

Błąd(1): L = 92 / 5Ch (Poprawna L = 2 / 2h)

R = 92 / 5Ch

SUMA = 94 / 5Eh

XRO = 0

Takt6:

S2 | IRAE : SUMA -> RAE

RAE = 94 / 5Eh

D2 | NSI : LR+1 -> LR

LR = 93 / 5Dh

Takt7:

TEST | TIND : Adresowanie pośrednie

RAPS = 26 / 1Ah

MAKRO

=====26=====

Takt0: RBPS=000000010060h

Takt7:

TEST | UNB : Zawsze pozytywny

RAPS = 96 / 60h

MAKRO

=====96=====

Takt0: RBPS=900624000000h

Takt1:

S1 | ORAE : RAE -> BUS

BUS = 94 / 5Eh

D1 | IRAP : BUS -> RAP

RAP = 94 / 5Eh

C1 | RRC : Rozpoczęcie RRC

RBP = -21846 / AAAAh

Takt7:

S3 | ORBP : RBP -> BUS

BUS = -21846 / AAAAh

D3 | IX : BUS -> X

X = -21846 / AAAAh

RAPS = 97 / 61h

MAKRO

=====97=====

Takt0: RBPS=BC300E000600h

Takt1:

S1 | IALU : A -> LALU

LALU = -21440 / AC40h

D1 | OXE : X -> RALU

RALU = -21846 / AAAAh

Takt2:

ALU | AND : ALU = LALU AND RALU

ALU = -22528 / A800h

ZNAK = 1, OFF = 0

Takt6:

S2 | OBE : ALU -> BUS

BUS = -22528 / A800h

D2 | IA : BUS -> A

A = -22528 / A800h

Takt7:

C1 | END : (Cykl 30) Koniec mikroprogramu (15:13.30)

RAPS = 0 / 0h

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 = 93 / 5Dh

D1 | IRAP : BUS -> RAP

RAP = 93 / 5Dh

C1 | RRC : Rozpoczęcie RRC

RBP = -14582 / C70Ah

Takt7:

S3 | ORBP : RBP -> BUS

BUS = -14582 / C70Ah

D3 | IRR : BUS -> RR

RR = -14582 / C70Ah

RAPS = 49 / 31h

MAKRO

=====49=====

Takt0: RBPS=68C801830032h

Takt1:

S1 | ORR : RR -> BUS

BUS = -14582 / C70Ah

D1 | ILK : BUS -> LK

LK = 10 / Ah

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 = 94 / 5Eh

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 = 90 / 5Ah

Takt7:

S3 | ORBP : RBP -> BUS

BUS = 90 / 5Ah

D3 | IX : BUS -> X

X = 90 / 5Ah

RAPS = 51 / 33h

MAKRO

=====51=====

Takt0: RBPS=03A801600000h

Takt6:

S2 | OX : X -> BUS

BUS = 90 / 5Ah

D2 | IBI : BUS -> RAE

RAE = 90 / 5Ah

Takt7:

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

RAPS = 24 / 18h

MAKRO

=====24=====

Takt0: RBPS=00000001005Ah

Takt7:

TEST | UNB : Zawsze pozytywny

RAPS = 90 / 5Ah

MAKRO

=====90=====

Takt0: RBPS=00000004006Eh

Takt7:

TEST | TAS : A >= 0

RAPS = 91 / 5Bh

MAKRO

=====91=====

Takt0: RBPS=000000010010h

Takt7:

TEST | UNB : Zawsze pozytywny

RAPS = 16 / 10h

MAKRO

=====16=====

Takt0: RBPS=000000010048h

Takt7:

TEST | UNB : Zawsze pozytywny

RAPS = 72 / 48h

MAKRO

=====72=====

Takt0: RBPS=00041E000000h

Takt7:

S3 | ORAE : RAE -> BUS

BUS = 90 / 5Ah

D3 | ILR : BUS -> LR

LR = 90 / 5Ah

C1 | END : (Cykl 40) Koniec mikroprogramu (15:17.00)

RAPS = 0 / 0h

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 = 90 / 5Ah

D1 | IRAP : BUS -> RAP

RAP = 90 / 5Ah

C1 | RRC : Rozpoczęcie RRC

RBP = 3338 / D0Ah

Takt7:

S3 | ORBP : RBP -> BUS

BUS = 3338 / D0Ah

D3 | IRR : BUS -> RR

RR = 3338 / D0Ah

RAPS = 49 / 31h

MAKRO

=====49=====

Takt0: RBPS=68C801830032h

Takt1:

S1 | ORR : RR -> BUS

BUS = 3338 / D0Ah

D1 | ILK : BUS -> LK

LK = 10 / Ah

C2 | CEA : Oblicz adres efektywny

L = 10 / Ah

R = 20 / 14h

SUMA = 30 / 1Eh

XRO = 0

Takt6:

S2 | IRAE : SUMA -> RAE

RAE = 30 / 1Eh

D2 | NSI : LR+1 -> LR

LR = 91 / 5Bh

Takt7:

TEST | TIND : Adresowanie pośrednie

RAPS = 50 / 32h

MAKRO

=====50=====

Takt0: RBPS=900624000000h

Takt1:

S1 | ORAE : RAE -> BUS

BUS = 30 / 1Eh

D1 | IRAP : BUS -> RAP

RAP = 30 / 1Eh

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

RAPS = 1 / 1h

MAKRO

=====1=====

Takt0: RBPS=000000010034h

Takt7:

TEST | UNB : Zawsze pozytywny

RAPS = 52 / 34h

MAKRO

=====52=====

Takt0: RBPS=900624000000h

Takt1:

S1 | ORAE : RAE -> BUS

BUS = 255 / FFh

D1 | IRAP : BUS -> RAP

RAP = 255 / FFh

C1 | RRC : Rozpoczęcie RRC

RBP = -29700 / 8BFCh

Takt7:

S3 | ORBP : RBP -> BUS

BUS = -29700 / 8BFCh

D3 | IX : BUS -> X

X = -29700 / 8BFCh

RAPS = 53 / 35h

MAKRO

=====53=====

Takt0: RBPS=BC300E000100h

Takt1:

S1 | IALU : A -> LALU

LALU = -22528 / A800h

D1 | OXE : X -> RALU

RALU = -29700 / 8BFCh

Takt2:

ALU | ADD : ALU = LALU + RALU

ALU = 13308 / 33FCh

ZNAK = 0, OFF = 1

Takt6:

S2 | OBE : ALU -> BUS

BUS = 13308 / 33FCh

D2 | IA : BUS -> A

A = 13308 / 33FCh

Takt7:

C1 | END : (Cykl 48) Koniec mikroprogramu (15:20.44)

RAPS = 0 / 0h

15:26.42

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

Ocena: 5 Błędy: 1

?

Opis działania mikroprogramu

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

- A – Akumulator (przechowuje wyniki operacji)
- LK – Licznik Kroków
- LR – Licznik Rozkazów
- MQ – Rejestr Iloczynu/Ilorazu
- 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 wykonywaniu rozkazu będą opisane w takiej konstrukcji:

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

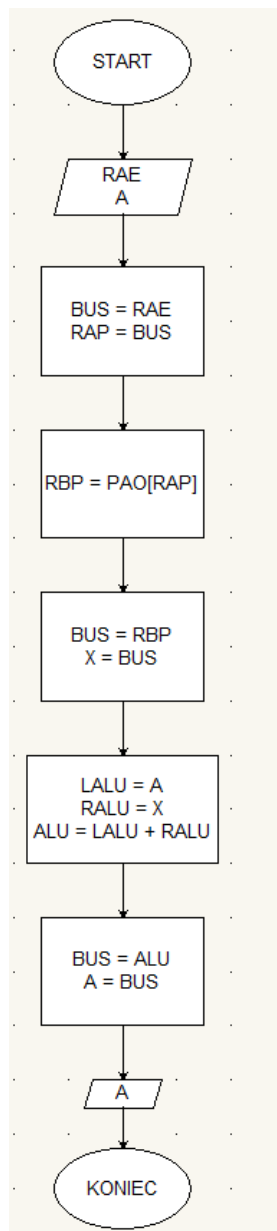
Do każdego rozkazu załączona jest graficzna prezentacja algorytmu w postaci schematu blokowego.

Rozkaz dla LR = 90 (rozkaz ADD 101 10)

Wartości po pobraniu rozkazu:

- LR = 91 / 5Bh – adres komórki PAO zawierający adres następnego rozkazu.
- RAPS = 52 / 34h – adres pierwszego wiersza PM zawierającego mikrorozkazy wykonania aktualnego rozkazu.
- RAE = 255 / FFh – pobrana metodą pośrednią (rozkaz zwykły, l=1), wartość z komórki pamięci o indeksie równym obliczonego adresu efektywnego.
- LK = 10 / Ah – liczba kroków wyznaczona na podstawie 7 najmniej znaczących bitów wartości komórki PAO o indeksie równym aktualnego licznika rozkazów
- XRO = 0 – brak ustawionej flagi

Schemat blokowy:



Opis rozkazu ADD 101 10 (Dodawanie):

Rozkaz ADD dodaje do wartości rejestru A wartość komórki pamięci o indeksie równym wartości rejestru RAP.

Kolejne działania:

1. TEST | UNB : Zawsze pozytywny – nastąpiło przejście do wiersza PM z wykonaniem rozkazu
2. Wartość RAE wynosi 255 / FFh, wartość A wynosi -32410 / 8166h
3. ORAE : RAE -> BUS - 255 / FFh
4. IRAP : BUS -> RAP - 255 / FFh
5. RRC : Rozpoczęcie RRC – przekazanie do RBP wartości (-29700 / 8BFCh) na podstawie indeksu z RAP (255 / FFh)
6. ORBP : RBP -> BUS – (-29700 / 8BFCh)
7. IX : BUS -> X – (-29700 / 8BFCh)
8. IALU : A -> LALU - (-32410 / 8166h)
9. OXE : X -> RALU – (-29700 / 8BFCh)

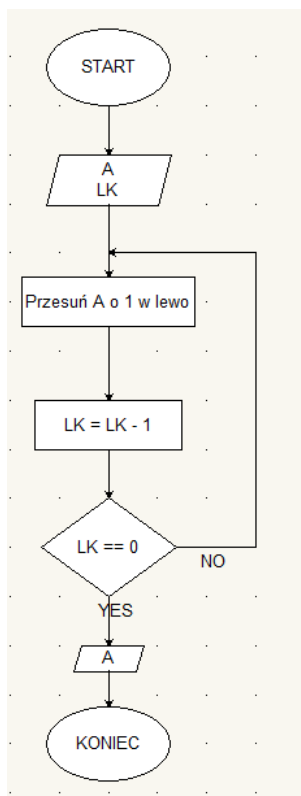
10. ADD : $ALU = LALU + RALU$ - wpisuje do rejestru ALU wartość sumy wartości w rejestrach LALU oraz RALU. $LALU = -32410 / 8166h$, a $RALU = -29700 / 8BFCh$. W związku z tym $ALU = 3426 / D62h$. Wynik jest dodatni, $ZNAK = 0$. Argumenty dodawania są ujemnego znaku, a wynik jest dodatni, dlatego $OFF = 1$.
11. OBE : $ALU \rightarrow BUS - 3426 / D62h$
12. IA : $BUS \rightarrow A - 3426 / D62h$
13. Wartość A wynosi $3426 / D62h$
14. END : Koniec mikroprogramu – przejście do zerowego wiersza PM

Rozkaz dla LR = 91 (rozkaz LLA 5)

Wartości po pobraniu rozkazu:

- a) LR = 92 / 5Ch – adres komórki PAO zawierający adres następnego rozkazu.
- b) RAPS = 118 / 76h – adres pierwszego wiersza PM zawierającego mikrorozkazy wykonania aktualnego rozkazu.
- c) RAE = 0 / 0h – pobrana metodą pośrednią (rozkaz rozszerzony), wartość z komórki pamięci o indeksie równym obliczonego adresu efektywnego.
- d) LK = 5 / 5h – liczba kroków wyznaczona na podstawie 7 najmniej znaczących bitów wartości komórki PAO o indeksie równym aktualnego licznika rozkazów
- e) XRO = 0 – brak ustawionej flagi

Schemat blokowy:



Opis rozkazu LLA 5 (Przesunięcie logiczne w lewo A):

Rozkaz LLA przesuwa wszystkie cyfry zapisu binarnego liczby A w lewo o N miejsc (N najbardziej znaczących znaków pierwotnej liczby jest tracona, N znaków na początku liczby to zera).

Kolejne działania:

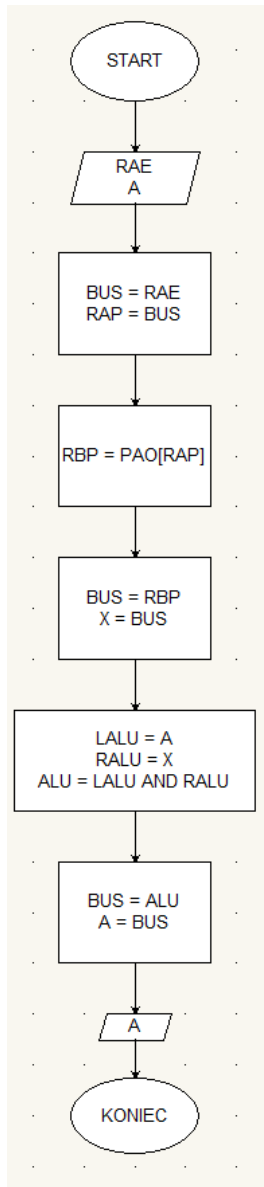
1. TEST | UNB : Zawsze pozytywny – nastąpiło przejście do wiersza PM z wykonaniem rozkazu
2. Wartość A wynosi 6852 / 1AC4h, wartość LK wynosi 5 / 5h
3. LLA : logiczne A w lewo
 - a. SHT : Operacja przesunięcia:
 - 1) A = 3426 / D62h = 0000 1101 0110 0010 binarnie
 - 2) Nowe A: 0 0001 1010 1100 0100 – „0” jest tracone, „0” jest dostawiane
 - 3) A = 0001 1010 1100 0100 binarnie = 6852 / 1AC4h
 - b. DLK : LK = [LK]-1 - 4 / 4h
4. TLK : SHT, LK=0 || !SHT, LK!=0 – ponieważ LK jest różne od zera, wracamy do 119 wiersza PM, a następnie z powrotem do 118 wiersza PM, żeby ponownie wykonać operację przesunięcia.
5. LLA : logiczne A w lewo
 - a. SHT : Operacja przesunięcia:
 - 1) A = 6852 / 1AC4h = 0001 1010 1100 0100 binarnie
 - 2) Nowe A: 0 0011 0101 1000 1000 – „0” jest tracone, „0” jest dostawiane
 - 3) A = 0011 0101 1000 1000 binarnie = 13704 / 3588h
 - b. DLK : LK = [LK]-1 - 3 / 3h
6. TLK : SHT, LK=0 || !SHT, LK!=0 – ponieważ LK jest różne od zera, wracamy do 119 wiersza PM, a następnie z powrotem do 118 wiersza PM, żeby ponownie wykonać operację przesunięcia.
7. LLA : logiczne A w lewo
 - a. SHT : Operacja przesunięcia:
 - 1) A = 13704 / 3588h = 0011 0101 1000 1000 binarnie
 - 2) Nowe A: 0 0110 1011 0001 0000 – „0” jest tracone, „0” jest dostawiane
 - 3) A = 0110 1011 0001 0000 binarnie = 27408 / 6B10h
 - b. DLK : LK = [LK]-1 - 2 / 2h
8. TLK : SHT, LK=0 || !SHT, LK!=0 – ponieważ LK jest różne od zera, wracamy do 119 wiersza PM, a następnie z powrotem do 118 wiersza PM, żeby ponownie wykonać operację przesunięcia.
9. LLA : logiczne A w lewo
 - a. SHT : Operacja przesunięcia:
 - 1) A = 27408 / 6B10h = 0110 1011 0001 0000 binarnie
 - 2) Nowe A: 0 1101 0110 0010 0000 – „0” jest tracone, „0” jest dostawiane
 - 3) A = 0110 1011 0001 0000 binarnie = -10720 / D620h (overflow)
 - b. DLK : LK = [LK]-1 - 1 / 1h
10. TLK : SHT, LK=0 || !SHT, LK!=0 – ponieważ LK jest różne od zera, wracamy do 119 wiersza PM, a następnie z powrotem do 118 wiersza PM, żeby ponownie wykonać operację przesunięcia.
11. LLA : logiczne A w lewo
 - a. SHT : Operacja przesunięcia:
 - 1) A = -10720 / D620h = 0110 1011 0001 0000 binarnie
 - 2) Nowe A: 0 1101 0110 0010 0000 – „0” jest tracone, „0” jest dostawiane
 - 3) A = 1101 0110 0010 0000 binarnie = -21440 / AC40h
 - b. DLK : LK = [LK]-1 - 0 / 0h
12. TLK : SHT, LK=0 || !SHT, LK!=0 – ponieważ LK poprzez dekrementację zrównało się z zerem to znaczy, że przesunięcie zostało wykonane, więc przechodzimy na adres PM kończący wykonanie rozkazu.
13. Wartość A wynosi -21440 / AC40h
14. END : Koniec mikroprogramu – przejście do zerowego wiersza PM

Rozkaz dla LR = 92 (rozkaz LPR(LAND) 010 2)

Wartości po pobraniu rozkazu:

- a) $LR = 93 / 5Dh$ – adres komórki PAO zawierający adres następnego rozkazu.
- b) $RAPS = 96 / 60h$ – adres pierwszego wiersza PM zawierającego mikrorozkazy wykonania aktualnego rozkazu.
- c) $RAE = 94 / 5Eh$ – pobrana metodą bezpośrednią (rozkaz zwykły, $I=0$), wartość obliczonego adresu efektywnego.
- d) $LK = 2 / 2h$ – liczba kroków wyznaczona na podstawie 7 najmniej znaczących bitów wartości komórki PAO o indeksie równym aktualnego licznika rozkazów
- e) $XRO = 0$ – brak ustawionej flagi

Schemat blokowy:



Opis rozkazu LAND 010 2 (Iloczyn logiczny A i komórki):

Rozkaz LAND zwraca na rejestr A iloczyn logiczny rejestru A i danej komórki PAO

Kolejne działania:

1. TEST | UNB : Zawsze pozytywny – nastąpiło przejście do wiersza PM z wykonaniem rozkazu
2. Wartość RAE wynosi $94 / 5Eh$, wartość A wynosi $-21440 / AC40h$

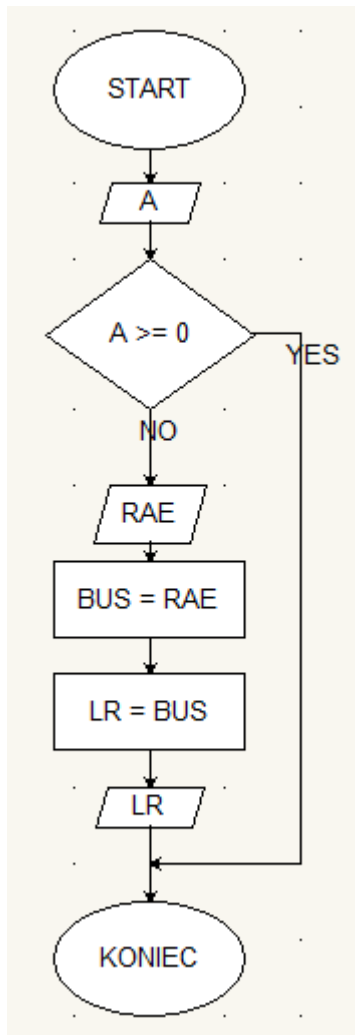
3. ORAE : RAE -> BUS - 94 / 5Eh
4. IRAP : BUS -> RAP - 94 / 5Eh
5. RRC : Rozpoczęcie RRC - przekazanie do RBP wartości (-21846 / AAAAh) na podstawie indeksu z RAP (94 / 5Eh)
6. ORBP : RBP -> BUS - (-21846 / AAAAh)
7. IX : BUS -> X - (-21846 / AAAAh)
8. IALU : A -> LALU - (-21440 / AC40h)
9. OXE : X -> RALU - (-21846 / AAAAh)
10. AND : ALU = LALU AND RALU – iloczyn logiczny zwraca najmniejszą wartość z podanych argumentów. Dlatego do rejestru ALU zwracana jest wartość RALU, czyli -21846 / AAAAh. Wynik jest ujemny, ZNAK = 1. Argumenty dodawania są tego samego znaku, dlatego OFF = 0.
11. OBE : ALU -> BUS - (-22528 / A800h)
12. IA : BUS -> A - (-22528 / A800h)
13. Wartość A wynosi -22528 / A800h
14. END : Koniec mikroprogramu – przejście do zerowego wiersza PM

Rozkaz dla LR = 93 (rozkaz BAN 111 10)

Wartości po pobraniu rozkazu:

- a) LR = 94 / 5Eh – adres komórki PAO zawierający adres następnego rozkazu.
- b) RAPS = 90 / 5Ah – adres pierwszego wiersza PM zawierającego mikrorozkazy wykonania aktualnego rozkazu.
- c) RAE = 94 / 5Eh – pobrana metodą pośrednią (rozkaz zwykły, l=1), wartość z komórki pamięci o indeksie równym obliczonego adresu efektywnego.
- d) LK = 10 / Ah – liczba kroków wyznaczona na podstawie 7 najmniej znaczących bitów wartości komórki PAO o indeksie równym aktualnego licznika rozkazów
- e) XRO = 1 – wystąpiła niedozwolona metoda adresacji. Program chce przenieść 2 różne wartości w jeden rejestr, co jest nie może mieć miejsca. Stąd flaga XRO została zmieniona na wartość 1.

Schemat blokowy:



Opis rozkazu BAN 111 10 (Skocz, jeśli A ujemny):

Rozkaz BAN wykonuje skok licznika rozkazu do wartości rejestru RAE pod warunkiem, że A jest ujemne.

Kolejne działania:

1. TEST | UNB : Zawsze pozytywny – nastąpiło przejście do wiersza PM z wykonaniem rozkazu
2. Wartość A wynosi -22528 / A800h
3. TEST | TAS : $A \geq 0$ – wynik testu to fałsz, więc A jest ujemne, dlatego RAPS przechodzi kolejno na wiersz 91, potem 16, a na końcu na 72, gdzie wykonywany jest skok licznika rozkazów.
4. Wartość RAE wynosi 90 / 5Ah
5. ORAE : $RAE \rightarrow BUS$ – 90 / 5Ah
6. ILR : $BUS \rightarrow LR$ – 90 / 5Ah
7. Wartość LR wynosi 90 / 5Ah
8. END : Koniec mikroprogramu – przejście do zerowego wiersza PM

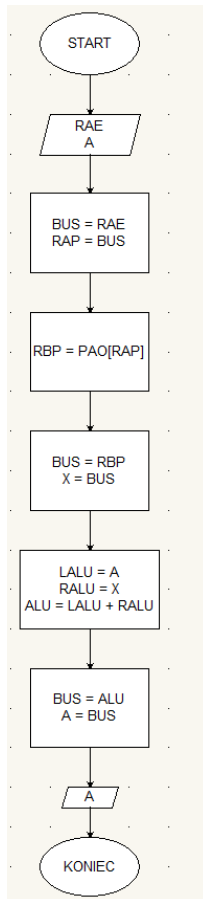
Rozkaz dla LR = 90 (rozkaz ADD 101 10) – ponowne wykonanie

Wartości po pobraniu rozkazu:

- a) LR = 91 / 5Bh – adres komórki PAO zawierający adres następnego rozkazu.

- b) $RAPS = 52 / 34h$ – adres pierwszego wiersza PM zawierającego mikrorozkazy wykonania aktualnego rozkazu.
- c) $RAE = 255 / FFh$ – pobrana metodą pośrednią (rozkaz zwykły, $I=1$), wartość z komórki pamięci o indeksie równym obliczonego adresu efektywnego.
- d) $LK = 10 / Ah$ – liczba kroków wyznaczona na podstawie 7 najmniej znaczących bitów wartości komórki PAO o indeksie równym aktualnego licznika rozkazów
- e) $XRO = 0$ – brak ustawionej flagi

Schemat blokowy:



Opis rozkazu ADD 101 10 (Dodawanie):

Rozkaz ADD dodaje do wartości rejestru A wartość komórki pamięci o indeksie równym wartości rejestru RAP.

Kolejne działania:

1. TEST | UNB : Zawsze pozytywny – nastąpiło przejście do wiersza PM z wykonaniem rozkazu
2. Wartość RAE wynosi 255 / FFh, wartość A wynosi -32410 / 8166h
3. ORAE : RAE -> BUS - 255 / FFh
4. IRAP : BUS -> RAP - 255 / FFh
5. RRC : Rozpoczęcie RRC – przekazanie do RBP wartości (-29700 / 8BFCh) na podstawie indeksu z RAP (255 / FFh)
6. ORBP : RBP -> BUS – (-29700 / 8BFCh)
7. IX : BUS -> X – (-29700 / 8BFCh)
8. IALU : A -> LALU – (-22528 / A800h)

9. OXE : X -> RALU - (-29700 / 8BFCh)
10. ADD : ALU = LALU + RALU - wpisuje do rejestru ALU wartość sumy wartości w rejestrach LALU oraz RALU. LALU = -22528 / A800h, a RALU = -29700 / 8BFCh. W związku z tym ALU = 13308 / 33FCh. Wynik jest dodatni, ZNAK = 0. Argumenty dodawania są ujemnego znaku, a wynik jest dodatni, dlatego OFF = 1.
11. OBE : ALU -> BUS - 13308 / 33FCh
12. IA : BUS -> A - 13308 / 33FCh
13. Wartość A wynosi 13308 / 33FCh
14. END : Koniec mikroprogramu – przejście do zerowego wiersza PM