

Wojskowa Akademia Techniczna  
im. Jarosława Dąbrowskiego

Laboratorium  
Architektury i organizacji komputerów I

Prowadzący mgr inż. Artur Miktus  
Sprawozdanie z ćwiczenia laboratoryjnego  
nr 3

**Temat ćwiczenia:**  
**Mikroprogramy rozkazów przesłań i arytmetycznych**

Wykonał: Arkadiusz Ostrzyżek  
Grupa: WCY22KY2S1  
Data wykonania ćwiczenia: 2023-11-22

## Treść polecenia:

Lab3 22KY2S1 zima 2023

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

Rejestry	
A	1000
LR	100+nr
RI	200
PAO	
Adres	Zawartość
0	220+nr
nr	2023
LR	ADD 001 255
LR+1	CMA 100+nr
LR+2	INX 100+nr
LR+3	STA 000 255
LR+4	LDA 111 nr
LR+5	LAI nr+100
200+nr	nr
220+nr	100
255	32000+nr

Pozostałe komórki PAO są wyzerowane.

Pozostałe komórki PAO są wyzerowane.

Stopień trudności zadania:

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

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 dzisiejszych zajęć (bez mnożenia i dzielenia oraz pozostałych z zestawu: MUL, DIV, SIO, LIO, BDN, CND, ENI, LDS)**, np. mimo że w treści przykładowych zadań nie ma odejmowania, to zarówno - pod adresem 2 w PM ma się znajdować odpowiedni skok do 54, - jak i pod adresem 54, 55 ma się znajdować się mikroprogram odejmowania.

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 - po fazie pobrania nastąpi samoczynnie (o ile została właściwie wypełniona pamięć mikroprogramów) przejście do fazy wykonania danego rozkazu maszynowego komputera LabZSK (dla dodawania RAPS = 1, a potem 52).

Wydruk 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

31    Test    \_\_\_UNB    Zawsze pozytywny  
NA    \_\_\_48

32    Test    \_\_\_UNB    Zawsze pozytywny

NA    \_\_\_98

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  $RI \leq 0$



	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 <= 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 >= 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,  $LK \neq 0$   
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,  $LK \neq 0$   
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,  $LK = 0$  || !SHT,  $LK \neq 0$   
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,  $LK = 0$  || !SHT,  $LK \neq 0$   
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 || !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 PAO:

0	0000000011100011b	00E3h	227		
7	0000011111100111b	07E7h	2023		
107	0000100111111111b	09FFh	OP=1	XSI=001	DA=255
108	0000000011101011b	00EBh	AOP=1		N=107
109	0000010111101011b	05EBh	AOP=11		N=107
110	0011000011111111b	30FFh	OP=6	XSI=000	DA=255
111	0100011100000111b	4707h	OP=8	XSI=111	DA=7
112	0000010011101011b	04EBh	AOP=9		N=107
207	0000000000000011b	0007h	7		
227	0000000000000011b	0007h	7		
255	1111010000110001b	F431h	32007		

Zawartość pliku log:

Start symulatora 22.11.2023 08:54:35

Stacja "LABITC"

Zalogowano jako: "Student"

Wersja aplikacji: 1.2.3.0

Dostępne interfejsy sieciowe: 10.6.120.15

192.168.56.1

192.168.47.1

192.168.13.1

====Start symulacji====

09:04.58

====Zawartość rejestrów====

LK	=	0h	0
A	=	3E8h	1000
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	=	6Bh	107
RI	=	C8h	200
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 = 107 / 6Bh  
D1 | IRAP : BUS -> RAP  
RAP = 107 / 6Bh  
C1 | RRC : Rozpoczęcie RRC  
RBP = 2559 / 9FFh

Takt7:

S3 | ORBP : RBP -> BUS  
BUS = 2559 / 9FFh  
D3 | IRR : BUS -> RR  
RR = 2559 / 9FFh

RAPS = 49 / 31h

MAKRO

=====49=====

Takt0: RBPS=68C801830032h

Takt1:

S1 | ORR : RR -> BUS  
BUS = 2559 / 9FFh  
D1 | ILK : BUS -> LK  
Błąd(1): LK = 255 / FFh (Poprawna LK = 127 /  
7Fh)

C2 | CEA : Oblicz adres efektywny  
Błąd(2): L = 127 / 7Fh (Poprawna L = 255 / FFh)

R = 0 / 0h

SUMA = 255 / FFh

XR0 = 0

Takt6:

S2 | IRAE : SUMA -> RAE

RAE = 255 / FFh

D2 | NSI : LR+1 -> LR

LR = 108 / 6Ch

Takt7:

TEST | TIND : Adresowanie pośrednie

RAPS = 50 / 32h

MAKRO

=====50=====

Takt0: RBPS=900624000000h

Takt1:

S1 | ORAE : RAE -> BUS

BUS = 255 / FFh

D1 | IRAP : BUS -> RAP

RAP = 255 / FFh

C1 | RRC : Rozpoczęcie RRC

RBP = 32007 / 7D07h

Takt7:

S3 | ORBP : RBP -> BUS

BUS = 32007 / 7D07h

D3 | IX : BUS -> X

X = 32007 / 7D07h

RAPS = 51 / 33h

MAKRO

=====51=====

Takt0: RBPS=03A801600000h

Takt6:

S2 | OX : X -> BUS

BUS = 32007 / 7D07h

D2 | IBI : BUS -> RAE

RAE = 32007 / 7D07h

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 = 32007 / 7D07h  
D1 | IRAP : BUS -> RAP  
    RAP = 7 / 7h  
C1 | RRC : Rozpoczęcie RRC  
    RBP = 2023 / 7E7h

Takt7:

S3 | ORBP : RBP -> BUS  
    BUS = 2023 / 7E7h  
D3 | IX : BUS -> X  
    X = 2023 / 7E7h

RAPS = 53 / 35h

MAKRO

=====53=====

Takt0: RBPS=BC300E000100h

Takt1:

S1 | IALU : A -> LALU  
    LALU = 1000 / 3E8h  
D1 | OXE : X -> RALU  
    RALU = 2023 / 7E7h

Takt2:

ALU | ADD : ALU = LALU + RALU  
    ALU = 3023 / BCFh  
    ZNAK = 0, OFF = 0

Takt6:

S2 | OBE : ALU -> BUS  
    BUS = 3023 / BCFh  
D2 | IA : BUS -> A  
    A = 3023 / BCFh

Takt7:

C1 | END : (Cykl 8) Koniec mikroprogramu  
(09:14.12)

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 = 108 / 6Ch  
D1 | IRAP : BUS -> RAP  
RAP = 108 / 6Ch  
C1 | RRC : Rozpoczęcie RRC  
RBP = 235 / EBh

Takt7:

S3 | ORBP : RBP -> BUS  
BUS = 235 / EBh  
D3 | IRR : BUS -> RR  
RR = 235 / EBh

RAPS = 49 / 31h

MAKRO

=====49=====

Takt0: RBPS=68C801830032h

Takt1:

S1 | ORR : RR -> BUS  
BUS = 235 / EBh  
D1 | ILK : BUS -> LK  
LK = 107 / 6Bh  
C2 | CEA : Oblicz adres efektywny  
L = 107 / 6Bh  
R = 0 / 0h  
SUMA = 107 / 6Bh

Takt6:

S2 | IRAE : SUMA -> RAE  
RAE = 107 / 6Bh  
D2 | NSI : LR+1 -> LR  
LR = 109 / 6Dh

Takt7:

TEST | TIND : Adresowanie pośrednie

RAPS = 50 / 32h

MAKRO

=====50=====

Takt0: RBPS=900624000000h

Takt1:

S1 | ORAE : RAE -> BUS  
    BUS = 107 / 6Bh  
D1 | IRAP : BUS -> RAP  
    RAP = 107 / 6Bh  
C1 | RRC : Rozpoczęcie RRC  
    RBP = 2559 / 9FFh

Takt7:

S3 | ORBP : RBP -> BUS  
    BUS = 2559 / 9FFh  
D3 | IX : BUS -> X  
    X = 2559 / 9FFh

RAPS = 51 / 33h

MAKRO

=====51=====

Takt0: RBPS=03A801600000h

Takt6:

S2 | OX : X -> BUS  
    BUS = 2559 / 9FFh  
D2 | IBI : BUS -> RAE  
    RAE = 2559 / 9FFh

Takt7:

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

RAPS = 33 / 21h

MAKRO

=====33=====

Takt0: RBPS=00000001006Ah

Takt7:

TEST | UNB : Zawsze pozytywny

RAPS = 106 / 6Ah

MAKRO

=====106=====

Takt0: RBPS=A4300E000400h

Takt1:

S1 | IALU : A -> LALU  
    LALU = 3023 / BCFh

Takt2:

ALU | CMA : ALU = (NOT LALU)+1  
    Błąd(3): ALU = 3023 / BCFh (Poprawna ALU = -  
3023 / F431h)

ZNAK = 1, OFF = 0

Takt6:

```
S2 | OBE : ALU -> BUS
    BUS = -3023 / F431h
D2 | IA : BUS -> A
    A = -3023 / F431h
```

Takt7:

```
C1 | END : (Cykl 15) Koniec mikroprogramu
(09:21.18)
```

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 = 109 / 6Dh
D1 | IRAP : BUS -> RAP
    RAP = 109 / 6Dh
C1 | RRC : Rozpoczęcie RRC
    RBP = 1515 / 5EBh
```

Takt7:

```
S3 | ORBP : RBP -> BUS
    BUS = 1515 / 5EBh
D3 | IRR : BUS -> RR
    RR = 1515 / 5EBh
```

RAPS = 49 / 31h

MAKRO

=====49=====

Takt0: RBPS=68C801830032h

Takt1:

```
S1 | ORR : RR -> BUS
    BUS = 1515 / 5EBh
D1 | ILK : BUS -> LK
    LK = 107 / 6Bh
C2 | CEA : Oblicz adres efektywny
    L = 107 / 6Bh
```

R = 0 / 0h  
SUMA = 107 / 6Bh

Takt6:

S2 | IRAE : SUMA -> RAE  
RAE = 107 / 6Bh  
D2 | NSI : LR+1 -> LR  
LR = 110 / 6Eh

Takt7:

TEST | TIND : Adresowanie pośrednie

RAPS = 50 / 32h

MAKRO

=====50=====

Takt0: RBPS=900624000000h

Takt1:

S1 | ORAE : RAE -> BUS  
Błąd(4): BUS = 1515 / 5EBh (Poprawna BUS =  
107 / 6Bh)

D1 | IRAP : BUS -> RAP  
RAP = 107 / 6Bh

C1 | RRC : Rozpoczęcie RRC  
Błąd(5): RBP = 12543 / 30FFh (Poprawna RBP =  
2559 / 9FFh)

Takt7:

S3 | ORBP : RBP -> BUS  
BUS = 2559 / 9FFh  
D3 | IX : BUS -> X  
X = 2559 / 9FFh

RAPS = 51 / 33h

MAKRO

=====51=====

Takt0: RBPS=03A801600000h

Takt6:

S2 | OX : X -> BUS  
BUS = 2559 / 9FFh  
D2 | IBI : BUS -> RAE  
RAE = 2559 / 9FFh

Takt7:

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

Błąd(6): RAPS = 255 / FFh (Poprawna RAPS = 43 /  
2Bh)

MAKRO

=====43=====

Takt0: RBPS=00000001007Eh

Takt7:

TEST | UNB : Zawsze pozytywny

RAPS = 126 / 7Eh

MAKRO

=====126=====

Takt0: RBPS=008420000000h

Takt6:

S2 | IRAE : SUMA -> RAE  
RAE = 107 / 6Bh

Takt7:

S3 | ORAE : RAE -> BUS  
BUS = 107 / 6Bh  
D3 | IX : BUS -> X  
X = 107 / 6Bh

RAPS = 127 / 7Fh

MAKRO

=====127=====

Takt0: RBPS=3C200E000100h

Takt1:

S1 | IXRE : RI -> LALU  
LALU = 200 / C8h  
D1 | OXE : X -> RALU  
RALU = 107 / 6Bh

Takt2:

ALU | ADD : ALU = LALU + RALU  
ALU = 307 / 133h  
ZNAK = 0, OFF = 0

Takt6:

S2 | OBE : ALU -> BUS  
BUS = 307 / 133h  
D2 | IRI : BUS -> RI  
RI = 307 / 133h

Takt7:

C1 | END : (Cykl 23) Koniec mikroprogramu  
(09:28.19)

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 = 110 / 6Eh

D1 | IRAP : BUS -> RAP

RAP = 110 / 6Eh

C1 | RRC : Rozpoczęcie RRC

RBP = 12543 / 30FFh

Takt7:

S3 | ORBP : RBP -> BUS

BUS = 12543 / 30FFh

D3 | IRR : BUS -> RR

RR = 12543 / 30FFh

RAPS = 49 / 31h

MAKRO

=====49=====

Takt0: RBPS=68C801830032h

Takt1:

S1 | ORR : RR -> BUS

BUS = 12543 / 30FFh

D1 | ILK : BUS -> LK

LK = 127 / 7Fh

C2 | CEA : Oblicz adres efektywny

L = 255 / FFh

R = 0 / 0h

SUMA = 255 / FFh

XRO = 0

Takt6:

S2 | IRAE : SUMA -> RAE

RAE = 255 / FFh

D2 | NSI : LR+1 -> LR

LR = 111 / 6Fh

Takt7:

TEST | TIND : Adresowanie pośrednie

RAPS = 6 / 6h

MAKRO

=====6=====

Takt0: RBPS=00000001003Ah

Takt7:

TEST | UNB : Zawsze pozytywny

RAPS = 58 / 3Ah

MAKRO

=====58=====

Takt0: RBPS=9003D2000000h

Takt1:

S1 | ORAE : RAE -> BUS

BUS = 255 / FFh

D1 | IRAP : BUS -> RAP

RAP = 255 / FFh

Takt7:

S3 | OA : A -> BUS

BUS = -3023 / F431h

D3 | IRBP : BUS -> RBP

Błąd(7): RBP = 12543 / 30FFh (Poprawna RBP = -3023 / F431h)

C1 | CWC : Rozpoczęcie CWC

PA0[255] = 0x7D07 -zmiana-> PA0[255] = 0xF431

RAPS = 59 / 3Bh

MAKRO

=====59=====

Takt0: RBPS=00000E000000h

Takt7:

C1 | END : (Cykl 29) Koniec mikroprogramu  
(09:31.56)

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 = 111 / 6Fh  
D1 | IRAP : BUS -> RAP  
RAP = 111 / 6Fh  
C1 | RRC : Rozpoczęcie RRC  
RBP = 18183 / 4707h

Takt7:

S3 | ORBP : RBP -> BUS  
BUS = 18183 / 4707h  
D3 | IRR : BUS -> RR  
RR = 18183 / 4707h

RAPS = 49 / 31h

MAKRO

=====49=====

Takt0: RBPS=68C801830032h

Takt1:

S1 | ORR : RR -> BUS  
BUS = 18183 / 4707h  
D1 | ILK : BUS -> LK  
LK = 7 / 7h  
C2 | CEA : Oblicz adres efektywny  
Błąd(8): L = 7 / 7h (Poprawna 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 = 112 / 70h

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 = 227 / E3h

Takt7:

S3 | ORBP : RBP -> BUS  
BUS = 227 / E3h  
D3 | IX : BUS -> X  
X = 227 / E3h

RAPS = 51 / 33h

MAKRO

=====51=====

Takt0: RBPS=03A801600000h

Takt6:

S2 | OX : X -> BUS  
BUS = 227 / E3h  
D2 | IBI : BUS -> RAE  
RAE = 227 / E3h

Takt7:

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

Błąd(9): RAPS = 41 / 29h (Poprawna RAPS = 8 / 8h)

MAKRO

=====8=====

Takt0: RBPS=00000001003Eh

Takt7:

TEST | UNB : Zawsze pozytywny

RAPS = 62 / 3Eh

MAKRO

=====62=====

Takt0: RBPS=900664000000h

Takt1:

S1 | ORAE : RAE -> BUS  
BUS = 227 / E3h  
D1 | IRAP : BUS -> RAP  
RAP = 227 / E3h  
C1 | RRC : Rozpoczęcie RRC  
RBP = 7 / 7h

Takt7:

S3 | ORBP : RBP -> BUS  
BUS = 7 / 7h  
D3 | IA : BUS -> A  
A = 7 / 7h

RAPS = 63 / 3Fh

MAKRO

=====63=====

Takt0: RBPS=00000E000000h

Takt7:

C1 | END : (Cykl 37) Koniec mikroprogramu  
(09:37.18)

RAPS = 0 / 0h

09:51.08

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

Ocena: 3 Błędy: 9

뎡뎡

Zrzut ekranu:

Komputer LabZSK - sterowanie mikroprogramowe

Symulacja Mikroprogramy Pamięć operacyjna Info

Adres	S1	D1	S2	D2	S3	D3	C1	C2	Test	ALU	NA
54	ORAE	IRAP			ORBP	IX	RRC				
55	IALU	OXE	OBE	IA			END			SUB	
56	ORAE	IRAP			OMQ	IRBP	CWC				
57							END				
58	ORAE	IRAP			OA	IRBP	CWC				
59							END				
60	ORAE	IRAP			ORI	IRBP	CWC				
61							END				
62	ORAE	IRAP			ORBP	IA	RRC				
63							END				
64	ORAE	IRAP			ORBP	IRI	RRC				
65							END				
66	ORAE	IRAP			OLR	IRBP	CWC				

Pokaż log

MAKRO Ocena: 3  
MIKRO Błędy: 9  
Stop Takt: 0  
Cykl: 37

LK 7h 7

RAP E3h 227

RBP 7h 7

BUS 0h 0

A 7h 7

LALU 0h 0

RR 4707h 18183

RBPS 00000E000000h

MO 0h 0

RALU 0h 0

LR 70h 112

RAPS 0h 0

X E3h 227

ALU 133h 307

RI 133h 307

RAE E3h 227

ZNAK 0

MAV 1

XRO 1

IA 0

OFF 0

INT 0

Adres

Zawartość

Hex

221

222

223

224

225

226

227 000000000000111 0007

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255 1111010000110001 F431

PAO[227] - DANA

000000000000111b

7d

7h

UT08:54 > 09:04 + 00:42:42

Ziarko(instr).pdf

Lab3 KY2

https://www.ita.wat.edu.pl/~a.miki/

140%

LK

ADD U01 255

LR+1	CMA 100+nr
LR+2	INX 100+nr
LR+3	STA 000 255
LR+4	LDA 111 nr
LR+5	LAI nr+100
200+nr	nr
220+nr	100
255	32000+nr

Pozostałe komórki PAO są wyzerowane.

Stopień trudności zadania:

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

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 dzisiejszych zajęć (bez mnożenia i dzielenia oraz pozostałych z zestawu: MUL, DIV, SIO, LIO, BDN, CND, ENI, LDS), np. mimo że w treści przykładowych zadań nie ma odejmowania, to zarówno - pod adresem 2 w PM ma się znajdować odpowiedni skok do 54, - jak i pod adresem 54, 55 ma znajdować się mikroprogram odejmowania.

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

Uzasadnienie końcowej zawartości LR, RAPS, RAE na koniec mikroprogramu pobrania rozkazu dla każdego wykoannego rozkazu:

#### 1. Rozkaz ADD

0:

TINT nie wykazuje przerwania, RAPS ustawiany jest na 48.

48:

Wartość LR przenoszona jest do RAP (107).

RRC ustawia RBP na wartość rozkazu z wiersza PAO równego wartości RAP (2559).

Wartość RBP przenoszona jest do RR (2559).

RAPS zwiększany jest o jeden. (49).

49:

Wartość rozkazu z RR przenoszona jest do LK (255) . Informacja z DA mieści się na 8 bitach, a LK rozpatruje tylko 7 ostatnich bitów. Z wiąztu z tym jego wartość ustawiana jest na 1111111b (127).

Wykonujemy CEA, a więc DA z RR jest przenoszone do L (255). Jest to rozkaz zwykły z flagą I = 1 oznaczającą, że zachodzi adresowanie pośrednie, a więc R = 0.

Suma L (255) oraz R (0) wpisywana jest do RAE (255).

NSI zwiększa LR o 1 (108).

Test TIND wykrywa adresowanie pośrednie, a więc RAPS jest ustawiany na wartość NA (50).

50:

Wartość RAE przenoszona jest do RAP (255).

RRC ustawia RBP na wartość rozkazu z wiersza PAO równego wartości RAP (32007).

Wartość RBP przenoszona jest do X (255).

RAPS zwiększany jest o 1 (51).

51:

Wartość X przenoszona jest do RAE (32007).

OPC ustawia wartość RAPS na wartość równą OP code aktualnie wykonywanego rozkazu (1).

1:

UNB jest zawsze pozytywny, RAPS ustawiany jest na wartość NA (52).

52:

Do RAP przenoszone jest 7 najmniej znaczących bitów z RAE (7).

RRC ustawia RBP na wartość rozkazu z wiersza PAO równego wartości RAP (2023).

Wartość RBP przenoszona jest do X (2023).

RAPS zwiększany jest o 1 (52).

53:

Wartość A przenoszona jest do LALU (1000).

Wartość X przenoszona jest do RALU (2023).

Wykonywany jest rozkaz ADD, a więc ALU ustawiamy na LALU + RALU (3023).

Wartość ALU przenoszona jest do A (3023).

END ustawia RAPS na 0.

## 2. Rozkaz CMA

0:

TINT nie wykazuje przerwania, RAPS ustawiany jest na 48.

48:

Wartość LR przenoszona jest do RAP (108).

RRC ustawia RBP na wartość rozkazu z wiersza PAO równego wartości RAP (235).

Wartość RBP przenoszona jest do RR (235).

RAPS zwiększany jest o jeden. (49).

49:

Wartość rozkazu z RR przenoszona jest do LK (235) . Całość DA mieści się w LK (107).

Wykonujemy CEA, a więc DA z RR jest przenoszona do L (107). Jest to rozkaz rozszerzony, a więc R = 0.

Suma L (107) oraz R (0) wpisywana jest do RAE (107).

NSI zwiększa LR o 1 (108).

Test TIND wykrywa adresowanie pośrednie, a więc RAPS jest ustawiany na wartość NA (50).

50:

Wartość RAE przenoszona jest do RAP (107).

RRC ustawia RBP na wartość rozkazu z wiersza PAO równego wartości RAP (2559).

Wartość RBP przenoszona jest do X (2559).

RAPS zwiększany jest o 1 (51).

51:

Wartość X przenoszona jest do RAE (2559).

OPC ustawia wartość RAPS na wartość równą OP code aktualnie wykonywanego rozkazu (33).

33:

UNB jest zawsze pozytywny, RAPS ustawiany jest na wartość NA (106).

106:

Wartość A przenoszona jest do LALU (3023).

RALU ustawiany jest na 0, ze względu na rozkaz CMA.

Wykonywany jest rozkaz CMA, a więc ALU ustawiamy na odwrotność RALU (-3023).

Wartość ALU przenoszona jest do A (-3023).

END ustawia RAPS na 0.

### 3. INX

0:

TINT nie wykazuje przerwania, RAPS ustawiany jest na 48.

48:

Wartość LR przenoszona jest do RAP (109).

RRC ustawia RBP na wartość rozkazu z wiersza PAO równego wartości RAP (1515).

Wartość RBP przenoszona jest do RR (1515).

RAPS zwiększany jest o jeden. (49).

49:

Wartość rozkazu z RR przenoszona jest do LK (255) . Całość DA mieści się w LK (107).

Wykonujemy CEA, a więc DA z RR jest przenoszone do L (107). Jest to rozkaz rozszerzony, a więc  $R = 0$ .

Suma L (255) oraz R (0) wpisywana jest do RAE (107).

NSI zwiększa LR o 1 (110).

Test TIND wykrywa adresowanie pośrednie, a więc RAPS jest ustawiany na wartość NA (50).

50:

Wartość RAE przenoszona jest do RAP (107).

RRC ustawia RBP na wartość rozkazu z wiersza PAO równego wartości RAP (2559).

Wartość RBP przenoszona jest do X (2559).

RAPS zwiększany jest o 1 (51).

51:

Wartość X przenoszona jest do RAE (2559).

OPC ustawia wartość RAPS na wartość równą OP code aktualnie wykonywanego rozkazu (43).

43:

UNB jest zawsze pozytywny, RAPS ustawiany jest na wartość NA (126).

126:

Wartość SUMY przenoszona jest do RAE (107).

Wartość RAE przenoszona jest do X (107).

RAPS zwiększany jest o 1 (127).

Wartość RI przenoszona jest do LALU (200).

Wartość X przenoszona jest do RALU (107).

127:

ALU wynosi  $RALU + LALU$  (307).

Wartość ALU przenoszona jest do RI (307).

END ustawia RAPS na 0.

#### 4. STA

0:

TINT nie wykazuje przerwania, RAPS ustawiany jest na 48.

48:

Wartość LR przenoszona jest do RAP (110).

RRC ustawia RBP na wartość rozkazu z wiersza PAO równego wartości RAP (12543).

Wartość RBP przenoszona jest do RR (12543).

RAPS zwiększany jest o jeden. (49).

49:

Wartość rozkazu z RR przenoszona jest do LK (12543) . Informacja z DA mieści się na 8 bitach, a LK rozpatruje tylko 7 ostatnich bitów. Z wiązty z tym jego wartość ustawiana jest na 1111111b (127).

Wykonujemy CEA, a więc DA z RR jest przenoszone do L (255). Jest to rozkaz rozszerzony, a więc R = 0.

Suma L (255) oraz R (0) wpisywana jest do RAE (255).

NSI zwiększa LR o 1 (111).

Test TIND jest pozytywny, a więc RAPS jest ustawiany na wartość rozkazu (6).

6:

UNB jest zawsze pozytywny, RAPS ustawiany jest na wartość NA (58).

58:

Wartość RAE przenoszona jest do RAP (255).

Wartość A przenoszona jest do RBP (255).

CWC zapisuje wartość RBP (-3023) w komórce o numerze z RAP (255).

RAPS zwiększany jest o 1 (59).

END ustawia RAPS na 0.

#### 5. LDA

0:

TINT nie wykazuje przerwania, RAPS ustawiany jest na 48.



48:

Wartość LR przenoszona jest do RAP (111).

RRC ustawia RBP na wartość rozkazu z wiersza PAO równego wartości RAP (18183).

Wartość RBP przenoszona jest do RR (18183).

RAPS zwiększany jest o jeden. (49).

49:

Wartość rozkazu z RR przenoszona jest do LK (18183) . Całość DA mieści się w LK (7).

Wykonujemy CEA. Flagi XSI są równe I. Oznacza to, że ktoś popełnił błąd w trakcie programowania procesora. Oznacza to, że błąd zostanie wykazany poprzez ustawienie L, R na 0 oraz XRO na 1.

Suma L (0) oraz R (0) wpisywana jest do RAE (0).

NSI zwiększa LR o 1 (112).

Test TIND wykrywa adresowanie pośrednie, a więc RAPS jest ustawiany na wartość NA (50).

50:

Wartość RAE przenoszona jest do RAP (0).

RRC ustawia RBP na wartość rozkazu z wiersza PAO równego wartości RAP (227).

Wartość RBP przenoszona jest do X (227).

RAPS zwiększany jest o 1 (51).

51:

Wartość X przenoszona jest do RAE (227).

OPC ustawia wartość RAPS na wartość równą OP code aktualnie wykonywanego rozkazu (8).

8:

UNB jest zawsze pozytywny, RAPS ustawiany jest na wartość NA (62).

62:

Wartość RAE przenoszona jest do RAP (227).

RRC ustawia RBP na wartość rozkazu z wiersza PAO równego wartości RAP (7).

Wartość RBP przenoszona jest do A (7).

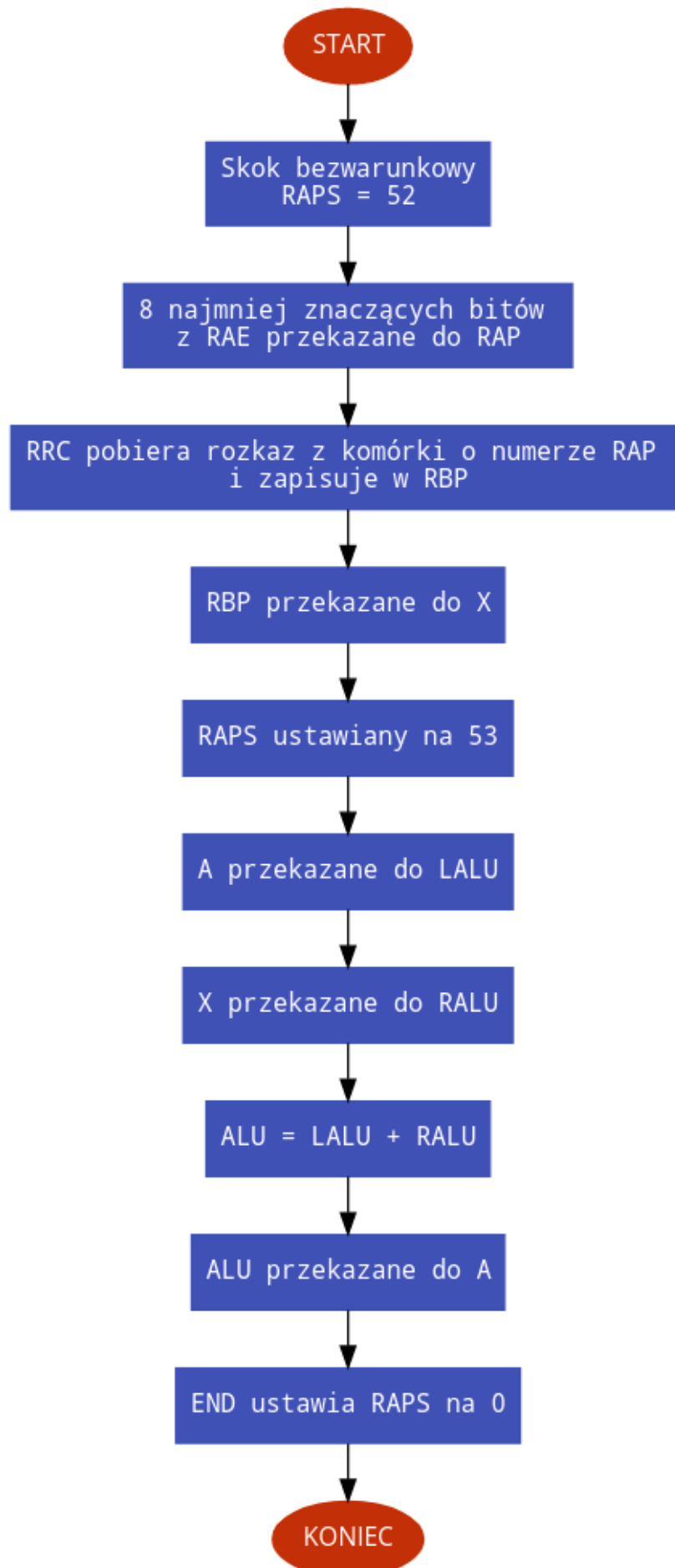
RAPS zwiększany jest o 1. (63)

63:

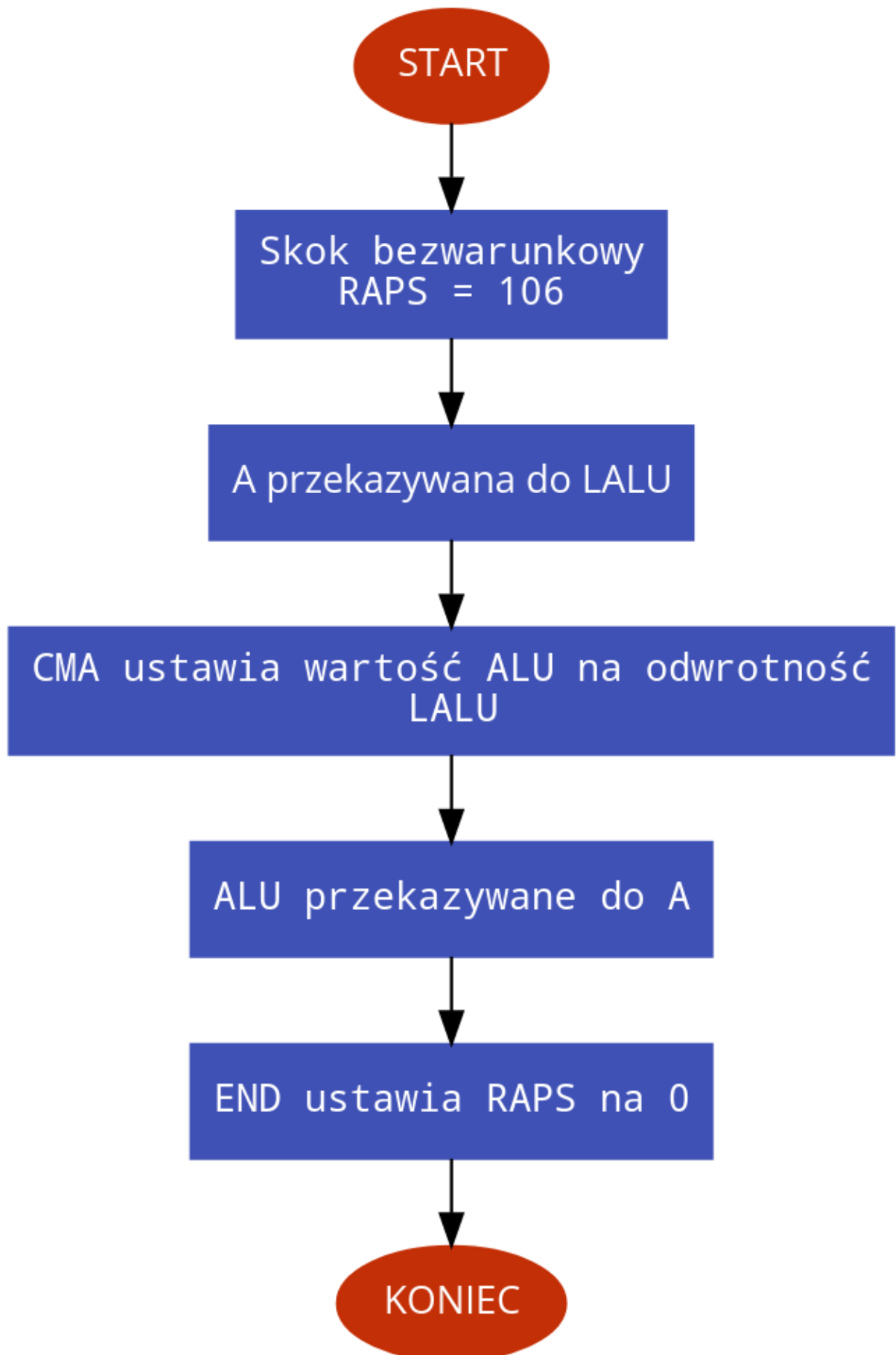
END ustawia RAPS na 0.

## Schematy blokowe:

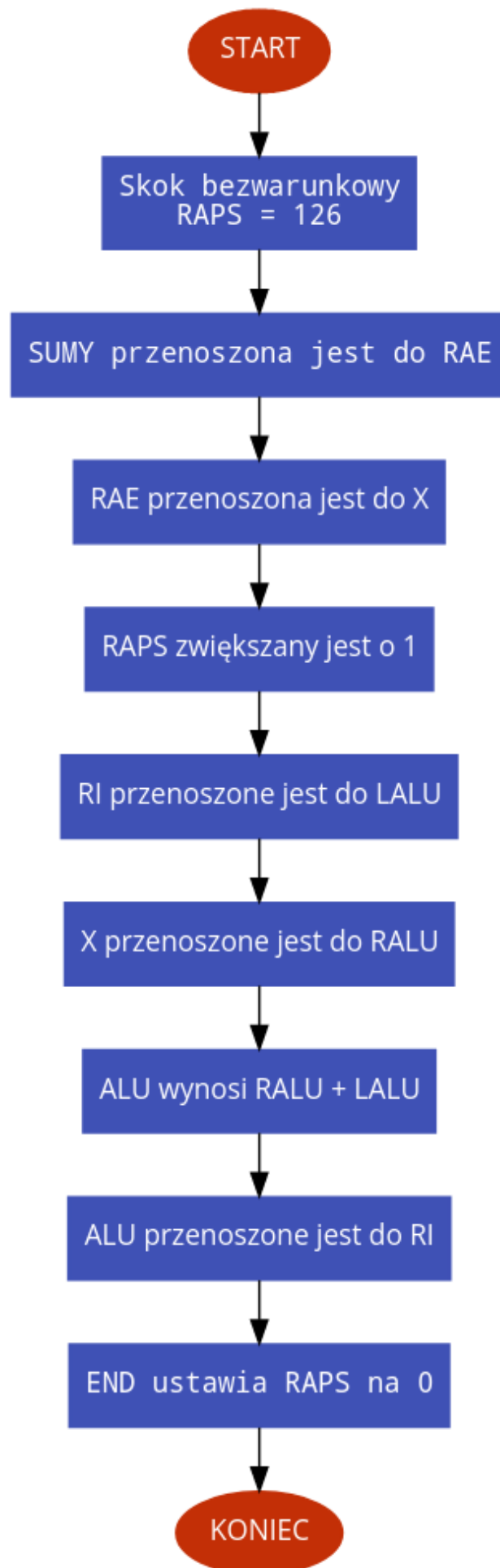
### 1. ADD



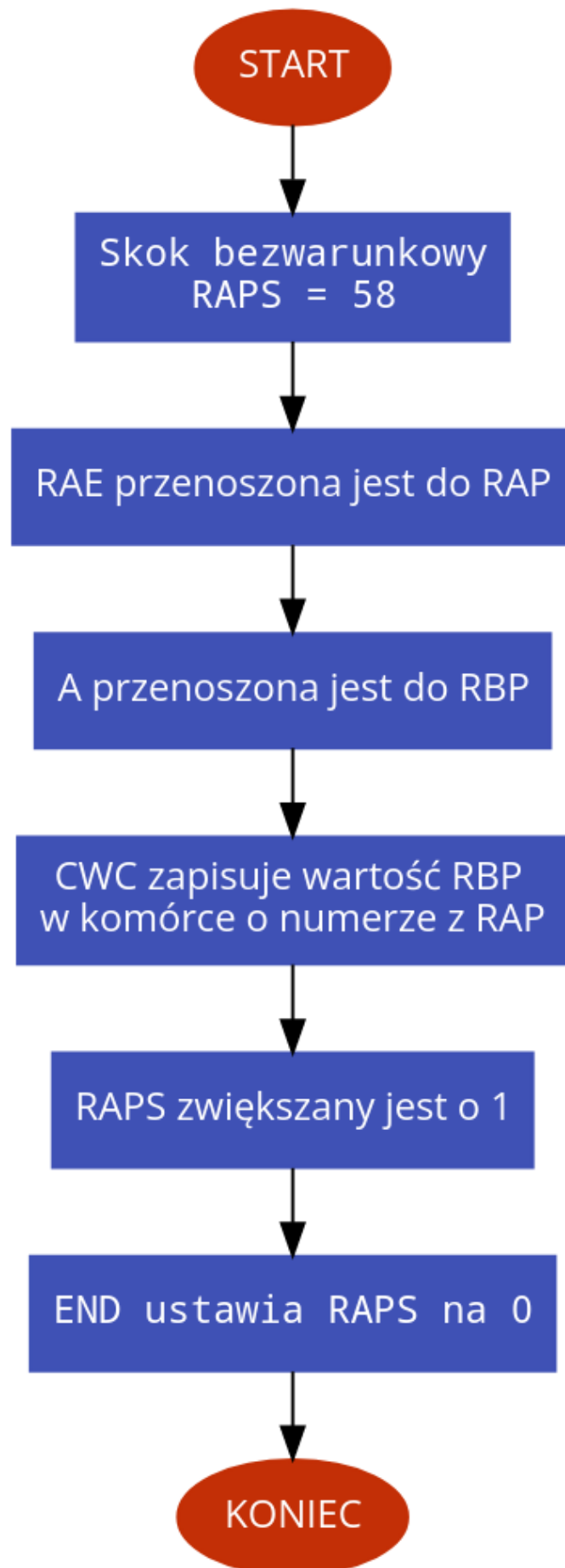
## 2. CMA



### 3. INX



#### 4. STA



## 5. LDA

