

Wojskowa Akademia Techniczna
im. Jarosława Dąbrowskiego

Laboratorium
Architektury i organizacji komputerów I

Prowadzący mgr inż. Artur Miktus
Sprawozdanie z ćwiczenia laboratoryjnego
nr 4

Temat ćwiczenia:

Mikroprogram rozkazów logicznych i skoków.

Wykonał: Arkadiusz Ostrzyżek

Grupa: WCY22KY2S1

Data wykonania ćwiczenia: 2023-11-07

Polecenie:

Lab4_KY2

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.

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 któregośkolwiek rozkazu 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	
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 <= 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!=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 !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 !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 !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	0000000001010111b	0057h	87		
24	0000000011111111b	00FFh	255		
87	0000110100000111b	0D07h	OP=1	XSI=101	DA=7
88	0000001100000101b	0305h	AOP=6		N=5
89	1101001000000010b	D202h	OP=26	XSI=010	DA=2
90	1100011100000111b	C707h	OP=24	XSI=111	DA=7
91	1010101010101010b	AAAAh	-21846		
255	1000101111111100b	8BFCh	-29700		

Zrzut ekranu ze stanem końcowym LabZSK:

The screenshot displays the LabZSK simulator interface. At the top, there's a table of memory addresses (80-98) with their contents. Below this, a status bar shows 'Oscena 3', 'Bledy 6', 'Takt 0', and 'Cykl 30'. The main part of the interface is a detailed diagram of the computer's internal components, including registers (LK, A, MQ, X, RAP, LALU, RALU, RBP, ALU, BUS, RR, LR, RT, RBPS, RAPS, RAF) and their values. The diagram is color-coded with green and yellow lines representing data paths.

Wydruk logu z wykonania ćwiczenia:

Start symulatora 15.12.2023 16:48:22
 Stacja "LABITC"
 Zalogowano jako: "Student"
 Wersja aplikacji: 1.2.3.0
 Dost#pne interfejsy sieciowe: 192.168.56.1
 10.6.106.4
 192.168.47.1
 192.168.13.1

==== Start symulacji =====
 16:56.28

==== Zawarto[### rejestrów =====

LK = 0h 0
 A = 8169h -32407
 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     = 57h   87
RI     = 11h   17
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 = 87 / 57h

D1 | IRAP : BUS -> RAP

RAP = 87 / 57h

C1 | RRC : Rozpocz##cie RRC

RBP = 3335 / D07h

Takt7:

S3 | ORBP : RBP -> BUS

BUS = 3335 / D07h

D3 | IRR : BUS -> RR

RR = 3335 / D07h

RAPS = 49 / 31h

MAKRO

=====49=====

Takt0: RBPS=68C801830032h

Takt1:

S1 | ORR : RR -> BUS

BUS = 3335 / D07h

D1 | ILK : BUS -> LK

LK = 7 / 7h

C2 | CEA : Oblicz adres efektywny

L = 7 / 7h

BB###d(1): R = 0 / 0h (Poprawna R = 17 / 11h)

SUMA = 24 / 18h

XRO = 0

```

Takt6:
    S2 | IRAE : SUMA -> RAE
        RAE = 24 / 18h
    D2 |  NSI : LR+1 -> LR
        LR = 88 / 58h
Takt7:
    TEST | TIND : Adresowanie po[#rednie

        RAPS = 50 / 32h
MAKRO
=====50=====
Takt0: RBPS=900624000000h
Takt1:
    S1 | ORAE : RAE -> BUS
        BUS = 24 / 18h
    D1 | IRAP : BUS -> RAP
        RAP = 24 / 18h
    C1 |  RRC : Rozpocz##cie 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 : Rozpocz##cie RRC
        BB###d(2): RBP = -21846 / AAAAh (Poprawna RBP = -2
9700 / 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 = -32407 / 8169h
    D1 | OXE : X -> RALU
        RALU = -29700 / 8BFCh

```

Takt2:

```

    ALU | ADD : ALU = LALU + RALU
        ALU = 3429 / D65h
        ZNAK = 0, OFF = 1

```

Takt6:

```

    S2 | OBE : ALU -> BUS
        BUS = 3429 / D65h
    D2 | IA : BUS -> A
        A = 3429 / D65h

```

Takt7:

C1 | END : (Cykl 8) Koniec mikroprogramu (17:09.52)

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 = 88 / 58h
    D1 | IRAP : BUS -> RAP
        RAP = 88 / 58h
    C1 | RRC : Rozpocz##cie 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 = 89 / 59h
Takt7:
    TEST | TIND : Adresowanie po[#rednie

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 : Rozpocz##cie 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

BB###d(3): RAPS = 0 / 0h (Poprawna 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 przesuni##cia

BB###d(4): A = 202 / CAh (Poprawna A = 6858 / 1ACA
h)

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 przesuni##cia

A = 13716 / 3594h

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 przesuni##cia
        A = 27432 / 6B28h
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 przesuni##cia
        A = -10672 / D650h
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 przesuni##cia

```

```

        A = -21344 / ACA0h
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 (17:28.44)

        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 = 89 / 59h
    D1 | IRAP : BUS -> RAP
        RAP = 89 / 59h
    C1 | RRC : Rozpocz##cie RRC
        BB###d(5): RBP = 514 / 202h (Poprawna RBP = -1177
4 / 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
        L = 2 / 2h

```

R = 89 / 59h
SUMA = 91 / 5Bh
XRO = 0

Takt6:

S2 | IRAE : SUMA -> RAE
RAE = 91 / 5Bh
D2 | NSI : LR+1 -> LR
LR = 90 / 5Ah

Takt7:

TEST | TIND : Adresowanie po[#rednie

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 = 91 / 5Bh
D1 | IRAP : BUS -> RAP
RAP = 91 / 5Bh
C1 | RRC : Rozpocz##cie 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 = -21344 / ACA0h
D1 | OXE : X -> RALU
RALU = -21846 / AAAAh

Takt2:

ALU | AND : ALU = LALU AND RALU

BB###d(6): ALU = 2236 / 8BCh (Poprawna ALU = -2236
8 / A8A0h)

ZNAK = 1, OFF = 0

Takt6:

S2 | OBE : ALU -> BUS

```

        BUS = -22368 / A8A0h
D2 |   IA : BUS -> A
        A = -22368 / A8A0h
Takt7:
C1 |   END : (Cykl 30) Koniec mikroprogramu (17:39.16)

        RAPS = 0 / 0h

        17:45.13
===== Stop  symulacji =====
Ocena: 3    BB###dy: 6
ps↵

```

Opis działania programu:

0: TINT nie wykazuje przerwania, RAPS ustawiany jest na 48.

48: Wartość LR przenoszona jest do RAP (87).

RRC ustawia RBP na wartość rozkazu z wiersza PAO równego wartości RAP (3335). Wartość RBP przenoszona jest do RR (3335). RAPS zwiększany jest o jeden. (49).

49: Wartość rozkazu z RR przenoszona jest do LK (3335) . LK rozpatruje tylko 7 ostatnich bitów. Z wiąztu z tym jego wartość ustawiana jest na (7). Wykonujemy CEA, a więc DA z RR jest przenoszone do L (7). Jest to rozkaz zwykły z flagami $X = 1$, $S = 0$, $I = 1$, a więc R będzie równe wartości z RI (17). Suma L (7) oraz R (17) wpisywana jest do RAE (24). NSI zwiększa LR o 1 (88). Test TIND wykrywa adresowanie pośrednie, a więc RAPS jest ustawiany na wartość NA (50).

50: Wartość RAE przenoszona jest do RAP (24). RRC ustawia RBP na wartość rozkazu z wiersza PAO równego wartości RAP (255). Wartość RBP przenoszona jest do X (255). RAPS zwiększany jest o 1 (51).

51: Wartość X przenoszona jest do RAE (255). OPC ustawia RAPS na wartość OP (1).

1: Wartość RAPS jest ustawiana na NA (52).

52: Wartość RAE przenoszona jest do RAP (255). RRC ustawia RBP na wartość rozkazu z wiersza PAO równego wartości RAP (-29700). Wartość RBP przenoszona jest do X (-29700). RAPS zwiększany jest o 1 (53).

53: Wartość A przenoszona jest do LALU (-32407), a wartość X przenoszona do RALU (-29700). Wykonywana jest operacja ADD. Ze względu na to, że suma LALU i RALU przekracza -32768, wartość zostanie zmieniona na dodatnią, „przerzucona” na drugą stronę. Wartość ALU zostanie ustawiona na ich sumę. (3429) ($-32407 - 29700 = -62107$; $-62107 + 32768 = -29339$; $32768 - 29339 = 3429$). Wartość ALU przenoszona jest do A (3429). END ustawia wartość RAPS na 0.

0: TINT nie wykazuje przerwania, RAPS ustawiany jest na 48.

48: Wartość LR przenoszona jest do RAP (88). RRC ustawia RBP na wartość rozkazu z wiersza PAO równego wartości RAP (773). Wartość RBP przenoszona jest do RR (773). RAPS zwiększany jest o jeden. (49).

49: Wartość rozkazu z RR przenoszona jest do LK (773) . LK rozpatruje tylko 7 ostatnich bitów. Z wiązki z tym jego wartość ustawiana jest na (5). Wykonujemy CEA, a więc DA z RR jest przenoszone do L (5). Jest to rozkaz rozszerzony, a więc R będzie równe 0. Suma L (5) oraz R (0) wpisywana jest do RAE (5). NSI zwiększa LR o 1 (89). Test TIND wykrywa adresowanie pośrednie, a więc RAPS jest ustawiany na wartość NA (50).

50: Wartość RAE przenoszona jest do RAP (5). RRC ustawia RBP na wartość rozkazu z wiersza PAO równego wartości RAP (0). Wartość RBP przenoszona jest do X (0). RAPS zwiększany jest o 1 (51).

51: Wartość X przenoszona jest do RAE (0). OPC ustawia RAPS na wartość OP (38).

38: Wartość RAPS jest ustawiana na NA (118).

118 oraz 119: Wykonywana jest operacja LLA. Jest to logiczne przesunięcie w lewo. Aby obliczyć LLA, musimy przedstawić liczbę z A w systemie binarnym. Następnie wszystkie bity, poza bitem znajdującym się najbardziej na lewo, przesuwane są w lewo. Operacja TLK zmniejsza LK o 1, po czym sprawdza czy LK jest równe 0. Jeśli jest to prawda, RAPS ustawiany jest na NA (110). W przeciwnym razie, RAPS zwiększany jest o 1 (119). Wiersz 119 zmienia zawsze wartość RAPS na NA (118). Ze względu na to, że LK jest równe 5 na początku, LLA zostanie wykonane 5 razy.

0D65h	0000 1101 0110 0101
1ACAh	0001 1010 1100 1010
3594h	0011 0101 1001 0100
6B28h	0110 1011 0010 1000
D650h	1101 0110 0101 0000
ACA0h	1010 1100 1010 0000

Finalna wartość A wynosi ACA0h i wartość RAPS ustawiona zostaje na 110.

110: END ustawia wartość RAPS na 0.

0: TINT nie wykazuje przerwania, RAPS ustawiany jest na 48.

48: Wartość LR przenoszona jest do RAP (89).

RRC ustawia RBP na wartość rozkazu z wiersza PAO równego wartości RAP (-11774). Wartość RBP przenoszona jest do RR (-11774). RAPS zwiększany jest o jeden. (49).

49: Wartość rozkazu z RR przenoszona jest do LK (-11774). LK rozpatruje tylko 7 ostatnich bitów. Z wiązki z tym jego wartość ustawiana jest na (2). Wykonujemy CEA, a więc DA z RR jest przenoszone do L (2). Jest to rozkaz zwykły z flagą S = 1, a więc R będzie równe LR (89). Suma L (2) oraz R (89) wpisywana jest do RAE (91). NSI zwiększa LR o 1 (90). Test TIND nie wykrywa adresowania pośrednie, a więc RAPS jest ustawiany na wartość OP (26).

26: Wartość RAPS jest ustawiana na NA (96).

96: Wartość RAE przenoszona jest do RAP (91). RRC ustawia RBP na wartość rozkazu z wiersza PAO równego wartości RAP (-21846). Wartość RBP przenoszona jest do X (-21846). RAPS zwiększany jest o 1 (97).

97: Wartość A przenoszona jest do LALU (-21344), a wartość X przenoszona do RALU (-21846). Wykonywana jest operacja AND. Musimy więc wykonać iloczyn logiczny komórek RALU i LALU.

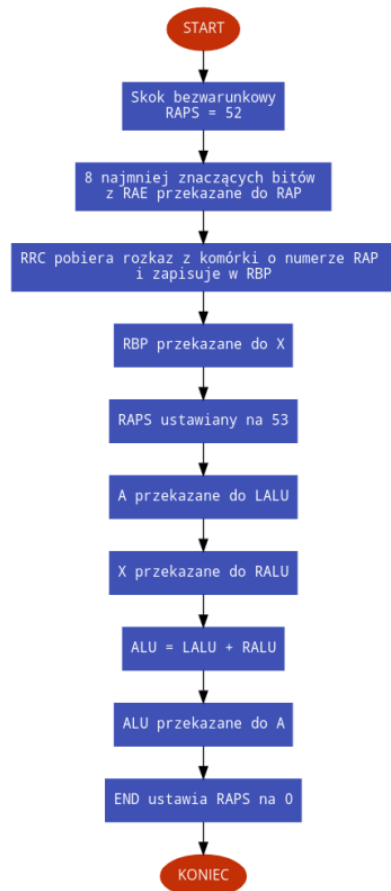
Operacja AND:

1010110010100000	ACA0h
1010101010101010	AAAAh
1010100010100000	A8A0h

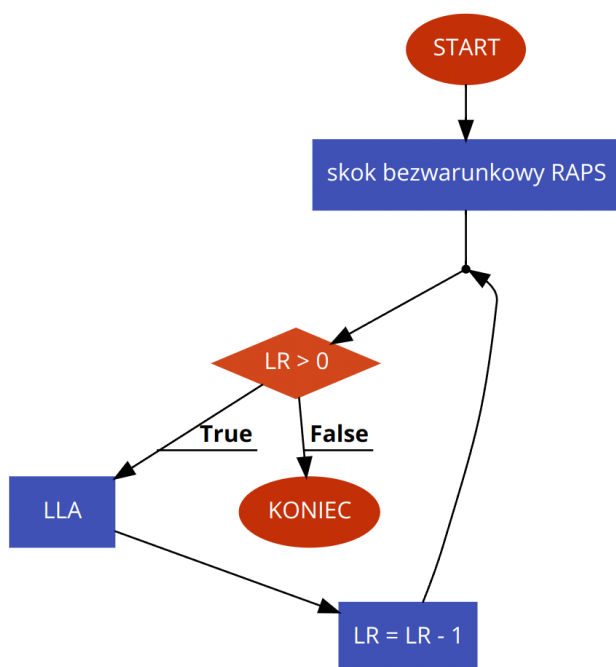
Wartość z ALU przenoszona jest do A (-22368).
END ustawia wartość RAPS na 0.

Schematy blokowe:

1. ADD



2. LLA



3. AND

