

Wojskowa Akademia Techniczna im. Jarosława Dąbrowskiego

Laboratorium z przedmiotu:
Architektura i organizacja komputerów

Sprawozdanie z ćwiczenia laboratoryjnego nr 3: Mikroprogramy rozkazów przesłań i arytmetycznych

Spis treści

Treść zadania	2
Wydruk zawartości PM	3
Wydruk zawartości PAO	7
Wydruk logu z wykonania ćwiczenia	7
Opis działania mikroprogramu	27
Rozkaz dla LR = 130 (rozkaz LXI 20)	28
Rozkaz dla LR = 131 (rozkaz DEX 10)	30
Rozkaz dla LR = 132 (rozkaz STX 101 10)	33
Rozkaz dla LR = 133 (rozkaz ADD 111 10)	36
Rozkaz dla LR = 134 (rozkaz ADX 010 1)	39
Rozkaz dla LR = 135 (rozkaz TXA 110 10)	42
Ze względu na dopisanie do sprawozdania elementów dokończonych po przesłaniu pliku z logiem oraz zrzutem ekranu przesyłam zaktualizowany końcowy efekt programu:	44

Treść zadania

Lab3 WCY20IY4S1 zima 2021

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

Rejestry	
A	(-2000)
LR	120+nr
RI	nr
PAO	
Adres	Zawartość
0	255
nr	100
nr+10	200+nr
LR	LXI nr+10
LR+1	DEX nr
LR+2	STX 101 nr
LR+3	ADD 111 nr
LR+4	ADX 010 1
LR+5	TXA 110 nr
255	(-2021)

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.

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) faza wykonania danego rozkazu maszynowego komputera LabZSK.

Wydruk zawartości PM

0	Test NA	____TINT ____48	Brak przerwania
1	Test NA	____UNB ____52	Zawsze pozytywny
2	Test NA	____UNB ____54	Zawsze pozytywny
5	Test NA	____UNB ____63	Zawsze pozytywny
6	Test NA	____UNB ____65	Zawsze pozytywny
7	Test NA	____UNB ____67	Zawsze pozytywny
8	Test NA	____UNB ____69	Zawsze pozytywny
9	Test NA	____UNB ____71	Zawsze pozytywny
10	Test NA	____UNB ____73	Zawsze pozytywny
11	Test NA	____UNB ____75	Zawsze pozytywny
12	Test NA	____UNB ____76	Zawsze pozytywny
13	Test NA	____UNB ____56	Zawsze pozytywny

33	Test NA	___ UNB ___ 58	Zawsze pozytywny
41	Test NA	___ UNB ___ 77	Zawsze pozytywny
42	Test NA	___ UNB ___ 78	Zawsze pozytywny
43	Test NA	___ UNB ___ 59	Zawsze pozytywny
44	Test NA	___ UNB ___ 61	Zawsze pozytywny
48	S1 D1 S3 D3 C1	___ OLR ___ IRAP ___ ORBP ___ IRR ___ RRC	LR -> BUS BUS -> RAP RBP -> BUS BUS -> RR Rozpoczęcie RRC
49	S1 D1 S2 D2 C2 Test NA	___ ORR ___ ILK ___ IRAE ___ NSI ___ CEA ___ TIND ___ 50	RR -> BUS BUS -> LK SUMA -> RAE LR+1 -> LR Oblicz adres efektywny Adresowanie pośrednie
50	S1 D1 S3 D3 C1	___ ORAE ___ IRAP ___ ORBP ___ IX ___ RRC	RAE -> BUS BUS -> RAP RBP -> BUS BUS -> X Rozpoczęcie RRC
51	S2 D2 C2	___ OX ___ IBI ___ OPC	X -> BUS BUS -> RAE OP albo AOP+32 -> RAPS
52	S1 D1 S3 D3 C1	___ ORAE ___ IRAP ___ ORBP ___ IX ___ RRC	RAE -> BUS BUS -> RAP RBP -> BUS BUS -> X Rozpoczęcie RRC
53	S1 D1 S2 D2 C1 ALU	___ IALU ___ OXE ___ OBE ___ IA ___ END ___ ADD	A -> LALU X -> RALU ALU -> BUS BUS -> A Koniec mikroprogramu ALU = LALU + RALU
54	S1 D1 S3 D3 C1	___ ORAE ___ IRAP ___ ORBP ___ IX ___ RRC	RAE -> BUS BUS -> RAP RBP -> BUS BUS -> X 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	___ORBP	RBP -> BUS
	D3	___IX	BUS -> X
	C1	___RRC	Rozpoczęcie RRC
57	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
58	S1	___IALU	A -> LALU
	S2	___OBE	ALU -> BUS
	D2	___IA	BUS -> A
	C1	___END	Koniec mikroprogramu
	ALU	___CMA	ALU = (NOT LALU)+1
59	S2	___IRAE	SUMA -> RAE
	S3	___ORAE	RAE -> BUS
	D3	___IX	BUS -> X
60	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
61	S2	___IRAE	SUMA -> RAE
	S3	___ORAE	RAE -> BUS
	D3	___IX	BUS -> X
62	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
63	S1	___ORAE	RAE -> BUS
	D1	___IRAP	BUS -> RAP
	S3	___OMQ	MQ -> BUS
	D3	___IRBP	BUS -> RBP
	C1	___CWC	Rozpoczęcie CWC
64	C1	___END	Koniec mikroprogramu
65	S1	___ORAE	RAE -> BUS
	D1	___IRAP	BUS -> RAP

	S3	___OA	A -> BUS
	D3	___IRBP	BUS -> RBP
	C1	___CWC	Rozpoczęcie CWC
66	C1	___END	Koniec mikroprogramu
67	S1	___ORAE	RAE -> BUS
	D1	___IRAP	BUS -> RAP
	S3	___ORI	RI -> BUS
	D3	___IRBP	BUS -> RBP
	C1	___CWC	Rozpoczęcie CWC
68	C1	___END	Koniec mikroprogramu
69	S1	___ORAE	RAE -> BUS
	D1	___IRAP	BUS -> RAP
	S3	___ORBP	RBP -> BUS
	D3	___IA	BUS -> A
	C1	___RRC	Rozpoczęcie RRC
70	C1	___END	Koniec mikroprogramu
71	S1	___ORAE	RAE -> BUS
	D1	___IRAP	BUS -> RAP
	S3	___ORBP	RBP -> BUS
	D3	___IRI	BUS -> RI
	C1	___RRC	Rozpoczęcie RRC
72	C1	___END	Koniec mikroprogramu
73	S1	___ORAE	RAE -> BUS
	D1	___IRAP	BUS -> RAP
	S3	___OLR	LR -> BUS
	D3	___IRBP	BUS -> RBP
	C1	___CWC	Rozpoczęcie CWC
74	C1	___END	Koniec mikroprogramu
75	S2	___ORI	RI -> BUS
	D2	___IA	BUS -> A
	C1	___END	Koniec mikroprogramu
76	S2	___OMQ	MQ -> BUS
	D2	___IA	BUS -> A
	C1	___END	Koniec mikroprogramu
77	S2	___IRAE	SUMA -> RAE
	S3	___ORAE	RAE -> BUS
	D3	___IA	BUS -> A
	C1	___END	Koniec mikroprogramu
78	S2	___IRAE	SUMA -> RAE
	S3	___ORAE	RAE -> BUS
	D3	___IRI	BUS -> RI
	C1	___END	Koniec mikroprogramu

Wydruk zawartości PAO

0	0000000011111111b	00FFh	255		
10	0000000001100100b	0064h	100		
20	0000000011010010b	00D2h	210		
130	0000010100010100b	0514h	AOP=10	N=20	
131	0000011000001010b	060Ah	AOP=12	N=10	
132	0011110100001010b	3D0Ah	OP=7	XSI=101	DA=10
133	0000111100001010b	0F0Ah	OP=1	XSI=111	DA=10
134	0110101000000001b	6A01h	OP=13	XSI=010	DA=1
135	0101111000001010b	5E0Ah	OP=11	XSI=110	DA=10
255	1111100000011011b	F81Bh	-2021		

Wydruk logu z wykonania ćwiczenia

Start symulatora 10.11.2021 18:29:11

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=====

19:01.56

=====Zawartość rejestrów=====

LK	= 0h	0
A	= F830h	-2000
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 = 82h 130
RI = Ah 10
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 = 130 / 82h

D1 | IRAP : BUS -> RAP

RAP = 130 / 82h

C1 | RRC : Rozpoczęcie RRC

RBP = 1300 / 514h

Takt7:

S3 | ORBP : RBP -> BUS

BUS = 1300 / 514h

D3 | IRR : BUS -> RR

RR = 1300 / 514h

RAPS = 49 / 31h

MAKRO

=====49=====

Takt0: RBPS=68C801830032h

Takt1:

S1 | ORR : RR -> BUS

BUS = 1300 / 514h

D1 | ILK : BUS -> LK

LK = 20 / 14h

C2 | CEA : Oblicz adres efektywny

L = 20 / 14h

R = 0 / 0h

SUMA = 20 / 14h

Takt6:

S2 | IRAE : SUMA -> RAE

RAE = 20 / 14h

D2 | NSI : LR+1 -> LR

LR = 131 / 83h

Takt7:

TEST | TIND : Adresowanie pośrednie

RAPS = 50 / 32h

MAKRO

=====50=====

Takt0: RBPS=900624000000h

Takt1:

S1 | ORAE : RAE -> BUS

BUS = 20 / 14h

D1 | IRAP : BUS -> RAP

RAP = 20 / 14h

C1 | RRC : Rozpoczęcie RRC

RBP = 210 / D2h

Takt7:

S3 | ORBP : RBP -> BUS

BUS = 210 / D2h

D3 | IX : BUS -> X

X = 210 / D2h

RAPS = 51 / 33h

MAKRO

=====51=====

Takt0: RBPS=03A801600000h

Takt6:

S2 | OX : X -> BUS

BUS = 210 / D2h

D2 | IBI : BUS -> RAE

RAE = 210 / D2h

Takt7:

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

RAPS = 42 / 2Ah

MAKRO

=====42=====

Takt0: RBPS=00000001004Eh

Takt7:

TEST | UNB : Zawsze pozytywny

RAPS = 78 / 4Eh

MAKRO

=====78=====

Takt0: RBPS=00844E000000h

Takt6:

S2 | IRAE : SUMA -> RAE

RAE = 20 / 14h

Takt7:

S3 | ORAE : RAE -> BUS

BUS = 20 / 14h

D3 | IRI : BUS -> RI

RI = 20 / 14h

C1 | END : (Cykl 7) Koniec mikroprogramu (19:07.33)

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 = 131 / 83h

D1 | IRAP : BUS -> RAP

RAP = 131 / 83h

C1 | RRC : Rozpoczęcie RRC

RBP = 1546 / 60Ah

Takt7:

S3 | ORBP : RBP -> BUS

BUS = 1546 / 60Ah

D3 | IRR : BUS -> RR

RR = 1546 / 60Ah

RAPS = 49 / 31h

MAKRO

=====49=====

Takt0: RBPS=68C801830032h

Takt1:

S1 | ORR : RR -> BUS

BUS = 1546 / 60Ah

D1 | ILK : BUS -> LK

LK = 10 / Ah

C2 | CEA : Oblicz adres efektywny

L = 10 / Ah

R = 0 / 0h

SUMA = 10 / Ah

Takt6:

S2 | IRAE : SUMA -> RAE

RAE = 10 / Ah

D2 | NSI : LR+1 -> LR

LR = 132 / 84h

Takt7:

TEST | TIND : Adresowanie pośrednie

RAPS = 50 / 32h

MAKRO

=====50=====

Takt0: RBPS=900624000000h

Takt1:

S1 | ORAE : RAE -> BUS

BUS = 10 / Ah

D1 | IRAP : BUS -> RAP

RAP = 10 / Ah

C1 | RRC : Rozpoczęcie RRC

RBP = 100 / 64h

Takt7:

S3 | ORBP : RBP -> BUS

BUS = 100 / 64h

D3 | IX : BUS -> X

X = 100 / 64h

RAPS = 51 / 33h

MAKRO

=====51=====

Takt0: RBPS=03A8016000000h

Takt6:

S2 | OX : X -> BUS

BUS = 100 / 64h

D2 | IBI : BUS -> RAE

RAE = 100 / 64h

Takt7:

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

RAPS = 44 / 2Ch

MAKRO

=====44=====

Takt0: RBPS=00000001003Dh

Takt7:

TEST | UNB : Zawsze pozytywny

RAPS = 61 / 3Dh

MAKRO

=====61=====

Takt0: RBPS=008420000000h

Takt6:

S2 | IRAE : SUMA -> RAE

RAE = 10 / Ah

Takt7:

S3 | ORAE : RAE -> BUS

BUS = 10 / Ah

D3 | IX : BUS -> X

X = 10 / Ah

RAPS = 62 / 3Eh

MAKRO

=====62=====

Takt0: RBPS=3C200E000200h

Takt1:

S1 | IXRE : RI -> LALU

LALU = 20 / 14h

D1 | OXE : X -> RALU

RALU = 10 / Ah

Takt2:

ALU | SUB : ALU = LALU - RALU

ALU = 10 / Ah

ZNAK = 0, OFF = 0

Takt6:

S2 | OBE : ALU -> BUS

BUS = 10 / Ah

D2 | IRI : BUS -> RI

RI = 10 / Ah

Takt7:

C1 | END : (Cykl 15) Koniec mikroprogramu (19:19.50)

RAPS = 0 / 0h

MAKRO

=====0=====

Takt0: RBPS=000000020030h

Takt7:

INT = 0

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

Błąd(1): RAPS = 132 / 84h (Poprawna RAPS = 48 / 30h)

MAKRO

=====48=====

Takt0: RBPS=5006C4000000h

Takt1:

S1 | OLR : LR -> BUS

BUS = 132 / 84h

D1 | IRAP : BUS -> RAP

RAP = 132 / 84h

C1 | RRC : Rozpoczęcie RRC

RBP = 15626 / 3D0Ah

Takt7:

S3 | ORBP : RBP -> BUS

BUS = 15626 / 3D0Ah

D3 | IRR : BUS -> RR

RR = 15626 / 3D0Ah

RAPS = 49 / 31h

MAKRO

=====49=====

Takt0: RBPS=68C801830032h

Takt1:

S1 | ORR : RR -> BUS

BUS = 15626 / 3D0Ah

D1 | ILK : BUS -> LK

LK = 10 / Ah

C2 | CEA : Oblicz adres efektywny

L = 10 / Ah

R = 10 / Ah

SUMA = 20 / 14h

XRO = 0

Takt6:

S2 | IRAE : SUMA -> RAE

RAE = 20 / 14h

D2 | NSI : LR+1 -> LR

LR = 133 / 85h

Takt7:

TEST | TIND : Adresowanie pośrednie

RAPS = 50 / 32h

MAKRO

=====50=====

Takt0: RBPS=900624000000h

Takt1:

S1 | ORAE : RAE -> BUS

BUS = 20 / 14h

D1 | IRAP : BUS -> RAP

RAP = 20 / 14h

C1 | RRC : Rozpoczęcie RRC

RBP = 210 / D2h

Takt7:

S3 | ORBP : RBP -> BUS

BUS = 210 / D2h

D3 | IX : BUS -> X

X = 210 / D2h

RAPS = 51 / 33h

MAKRO

=====51=====

Takt0: RBPS=03A801600000h

Takt6:

S2 | OX : X -> BUS

BUS = 210 / D2h

D2 | IBI : BUS -> RAE

RAE = 210 / D2h

Takt7:

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

RAPS = 7 / 7h

MAKRO

=====7=====

Takt0: RBPS=000000010043h

Takt7:

TEST | UNB : Zawsze pozytywny

RAPS = 67 / 43h

MAKRO

=====67=====

Takt0: RBPS=9001D2000000h

Takt1:

S1 | ORAE : RAE -> BUS

BUS = 210 / D2h

D1 | IRAP : BUS -> RAP

RAP = 210 / D2h

Takt7:

S3 | ORI : RI -> BUS

BUS = 10 / Ah

D3 | IRBP : BUS -> RBP

RBP = 10 / Ah

C1 | CWC : Rozpoczęcie CWC

PAO[210] = 0x0000 -zmiana-> PAO[210] = 0x000A

RAPS = 68 / 44h

MAKRO

=====68=====

Takt0: RBPS=00000E000000h

Takt7:

C1 | END : (Cykl 23) Koniec mikroprogramu (19:24.55)

RAPS = 0 / 0h

MAKRO

=====0=====

Takt0: RBPS=000000020030h

Takt7:

INT = 0

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

Błąd(2): RAPS = 133 / 85h (Poprawna RAPS = 48 / 30h)

MAKRO

=====48=====

Takt0: RBPS=5006C4000000h

Takt1:

S1 | OLR : LR -> BUS

BUS = 133 / 85h

D1 | IRAP : BUS -> RAP

RAP = 133 / 85h

C1 | RRC : Rozpoczęcie RRC

RBP = 3850 / F0Ah

Takt7:

S3 | ORBP : RBP -> BUS

BUS = 3850 / F0Ah

D3 | IRR : BUS -> RR

RR = 3850 / F0Ah

RAPS = 49 / 31h

MAKRO

=====49=====

Takt0: RBPS=68C801830032h

Takt1:

S1 | ORR : RR -> BUS

BUS = 3850 / F0Ah

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 = 134 / 86h

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 = 255 / FFh

Takt7:

S3 | ORBP : RBP -> BUS

BUS = 255 / FFh

D3 | IX : BUS -> X

Błąd(3): X = 51 / 33h (Poprawna 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 = -2021 / F81Bh

Takt7:

S3 | ORBP : RBP -> BUS

BUS = -2021 / F81Bh

D3 | IX : BUS -> X

X = -2021 / F81Bh

RAPS = 53 / 35h

MAKRO

=====53=====

Takt0: RBPS=BC300E000100h

Takt1:

S1 | IALU : A -> LALU

LALU = -2000 / F830h

D1 | OXE : X -> RALU

RALU = -2021 / F81Bh

Takt2:

ALU | ADD : ALU = LALU + RALU

ALU = -4021 / F04Bh

ZNAK = 1, OFF = 0

Takt6:

S2 | OBE : ALU -> BUS

BUS = -4021 / F04Bh

D2 | IA : BUS -> A

A = -4021 / F04Bh

Takt7:

C1 | END : (Cykl 31) Koniec mikroprogramu (19:30.55)

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 = 134 / 86h

D1 | IRAP : BUS -> RAP

RAP = 134 / 86h

C1 | RRC : Rozpoczęcie RRC

RBP = 27137 / 6A01h

Takt7:

S3 | ORBP : RBP -> BUS

BUS = 27137 / 6A01h

D3 | IRR : BUS -> RR

RR = 27137 / 6A01h

RAPS = 49 / 31h

MAKRO

=====49=====

Takt0: RBPS=68C801830032h

Takt1:

S1 | ORR : RR -> BUS

BUS = 27137 / 6A01h

D1 | ILK : BUS -> LK

LK = 1 / 1h

C2 | CEA : Oblicz adres efektywny

L = 1 / 1h

R = 134 / 86h

SUMA = 135 / 87h

XRO = 0

Takt6:

S2 | IRAE : SUMA -> RAE

RAE = 135 / 87h

D2 | NSI : LR+1 -> LR

LR = 135 / 87h

Takt7:

TEST | TIND : Adresowanie pośrednie

RAPS = 13 / Dh

MAKRO

=====13=====

Takt0: RBPS=000000010038h

Takt7:

TEST | UNB : Zawsze pozytywny

RAPS = 56 / 38h

MAKRO

=====56=====

Takt0: RBPS=900624000000h

Takt1:

S1 | ORAE : RAE -> BUS

BUS = 135 / 87h

D1 | IRAP : BUS -> RAP

RAP = 135 / 87h

C1 | RRC : Rozpoczęcie RRC

RBP = 24074 / 5E0Ah

Takt7:

S3 | ORBP : RBP -> BUS

BUS = 24074 / 5E0Ah

D3 | IX : BUS -> X

X = 24074 / 5E0Ah

RAPS = 57 / 39h

MAKRO

=====57=====

Takt0: RBPS=3C200E000100h

Takt1:

S1 | IXRE : RI -> LALU

LALU = 10 / Ah

D1 | OXE : X -> RALU

RALU = 24074 / 5E0Ah

Takt2:

ALU | ADD : ALU = LALU + RALU

ALU = 24084 / 5E14h

ZNAK = 0, OFF = 0

Takt6:

S2 | OBE : ALU -> BUS

BUS = 24084 / 5E14h

D2 | IRI : BUS -> RI

RI = 24084 / 5E14h

Takt7:

C1 | END : (Cykl 37) Koniec mikroprogramu (19:39.54)

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 = 135 / 87h

D1 | IRAP : BUS -> RAP

RAP = 135 / 87h

C1 | RRC : Rozpoczęcie RRC

Błąd(4): RBP = 24064 / 5E00h (Poprawna RBP = 24074 / 5E0Ah)

Takt7:

S3 | ORBP : RBP -> BUS

BUS = 24074 / 5E0Ah

D3 | IRR : BUS -> RR

RR = 24074 / 5E0Ah

RAPS = 49 / 31h

MAKRO

=====49=====

Takt0: RBPS=68C801830032h

Takt1:

S1 | ORR : RR -> BUS

BUS = 24074 / 5E0Ah

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 = 136 / 88h

Takt7:

TEST | TIND : Adresowanie pośrednie

RAPS = 11 / Bh

MAKRO

=====11=====

Takt0: RBPS=00000001004Bh

Takt7:

TEST | UNB : Zawsze pozytywny

RAPS = 75 / 4Bh

MAKRO

=====75=====

Takt0: RBPS=01B00E000000h

Takt6:

S2 | ORI : RI -> BUS

BUS = 24084 / 5E14h

D2 | IA : BUS -> A

A = 24084 / 5E14h

Takt7:

C1 | END : (Cykl 42) Koniec mikroprogramu (20:57.09)

RAPS = 0 / 0h



Opis działania mikroprogramu

Każde działanie w ramach pobrania danego rozkazu rozpoczyna się w momencie, kiedy rejestr adresowy pamięci stałe (RAPS) wnosi 0. W związku z tym zawsze rozpoczniemy od mikrorozkazu TINT

sprawdzającego, czy nastąpiło przerwanie (symulator dla uproszczenia nigdy nie stwierdzi wystąpienia sygnału przerwania). Jeśli przerwanie nie nastąpiło, przechodzimy do 48 wiersza (taką wartość wpisujemy do RAPS). Od tego momentu rozpoczynają się opisy pobrania i wykonania każdego z rozkazów.

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 pobieraniu 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 = 130 (rozkaz LXI 20)

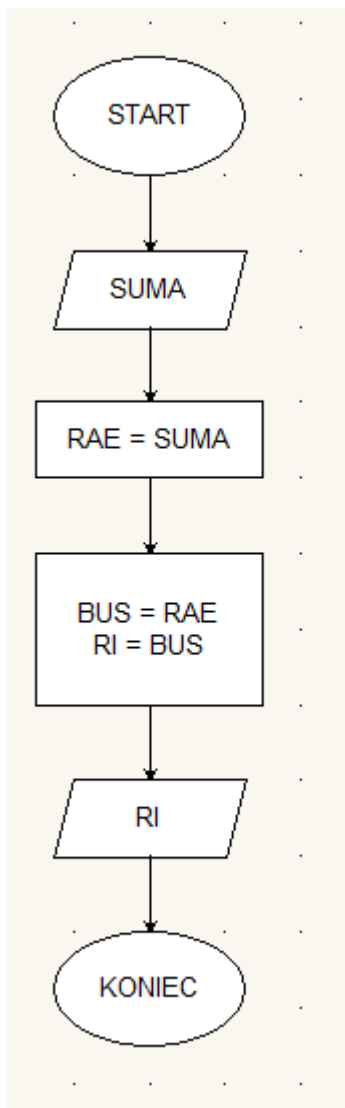
1. OLR : LR -> BUS - 130 / 82h
2. IRAP : BUS -> RAP - 130 / 82h
3. RRC : Rozpoczęcie RRC – przekazanie do RBP wartości (1300 / 514h) na podstawie indeksu z RAP (130 / 82h)
4. ORBP : RBP -> BUS - 1300 / 514h
5. IRR : BUS -> RR - 1300 / 514h
6. (koniec wiersza pamięci mikroprogramów, RAPS = 49 / 31h)
7. ORR : RR -> BUS - 1300 / 514h
8. ILK : BUS -> LK – przekazanie wartości na podstawie 7 najmniej znaczących bitów wartości magistrali (1300 / 514h). Wynika to z pojemności rejestru LK. W tym przypadku LK = 20 / 14h.
9. CEA : Oblicz adres efektywny – mając do czynienia z rozkazem rozszerzonym, do rejestru L trafia wartość 20 / 14h, rejestr R pozostaje wyzerowany. Mając te 2 wartości dodajemy je do siebie, a uzyskany wynik wpisujemy do rejestru SUMA (będzie to wartość 20 / 14h).
10. IRAE : SUMA -> RAE – 20 / 14h
11. NSI : LR+1 -> LR - 131 / 83h (zmieniamy licznik, aby być w stanie pobrać kolejny rozkaz)
12. TEST | TIND : Adresowanie pośrednie – sprawdzamy, czy wystąpiło adresowanie pośrednie. Robimy to poprzez dane dotyczące rozkazu. Jako że mamy do czynienia z rozkazem rozszerzonym, jest to adresacja pośrednia. W związku z tym w RAPS wpisujemy wartość następnej komórki pamięci mikroprogramu, czyli 50 / 32h.
13. ORAE : RAE -> BUS – 20 / 14h
14. IRAP : BUS -> RAP – 20 / 14h
15. RRC : Rozpoczęcie RRC – przekazanie do RBP wartości (210 / D2h) na podstawie indeksu z RAP (20 / 14h)

16. ORBP : RBP -> BUS - 210 / D2h
17. IX : BUS -> X – 210 / D2h
18. (koniec wiersza pamięci mikroprogramów, RAPS = 51 / 33h)
19. OX : X -> BUS – 210 / D2h
20. IBI : BUS -> RAE - 210 / D2h
21. OPC : OP albo AOP+32 -> RAPS – w zależności od typu rozkazu ustawiamy nową wartość pola RAPS. W tym przypadku mając rozkaz rozszerzony podajemy wartość AOP powiększoną o 32. AOP odczytujemy na podstawie 5, 6, 7 oraz 8 bitu zapisu 16-bitowego rozkazu. Oznacza to, że RAPS = 42 / 2Ah
22. TEST | UNB : Zawsze pozytywny – przejście na zadany wiersz pamięci mikroprogramu zawierający wykonanie rozkazu (RAPS = 78 / 4Eh)

Aktualne wartości:

- a) LR = 131 / 83h – adres komórki PAO zawierający adres następnego rozkazu.
- b) RAPS = 78 / 4Eh – adres pierwszego wiersza PM zawierającego mikrorozkazy wykonania aktualnego rozkazu.
- c) RAE = 210 / D2h – pobrana metodą pośrednią, wartość z komórki pamięci o indeksie równym obliczonego adresu efektywnego w kroku 9.

Schemat blokowy:



Opis rozkazu LXI 20 (ładuj RI bezpośrednio):

Rozkaz przekazuje do rejestru RI adresu efektywnego obliczonego i wpisanego do rejestru SUMA (wartość N podana w rozkazie).

Kolejne działania:

1. TEST | UNB : Zawsze pozytywny – nastąpiło przejście do wiersza PM z wykonaniem rozkazu
2. Wartość SUMA wynosi 20 / 14h
3. IRAE : SUMA -> RAE - 20 / 14h
4. ORAE : RAE -> BUS - 20 / 14h
5. IRI : BUS -> RI - 20 / 14h
6. Wartość RI wynosi 20 / 14h
7. END : Koniec mikroprogramu – przejście do zerowego wiersza PM

Rozkaz dla LR = 131 (rozkaz DEX 10)

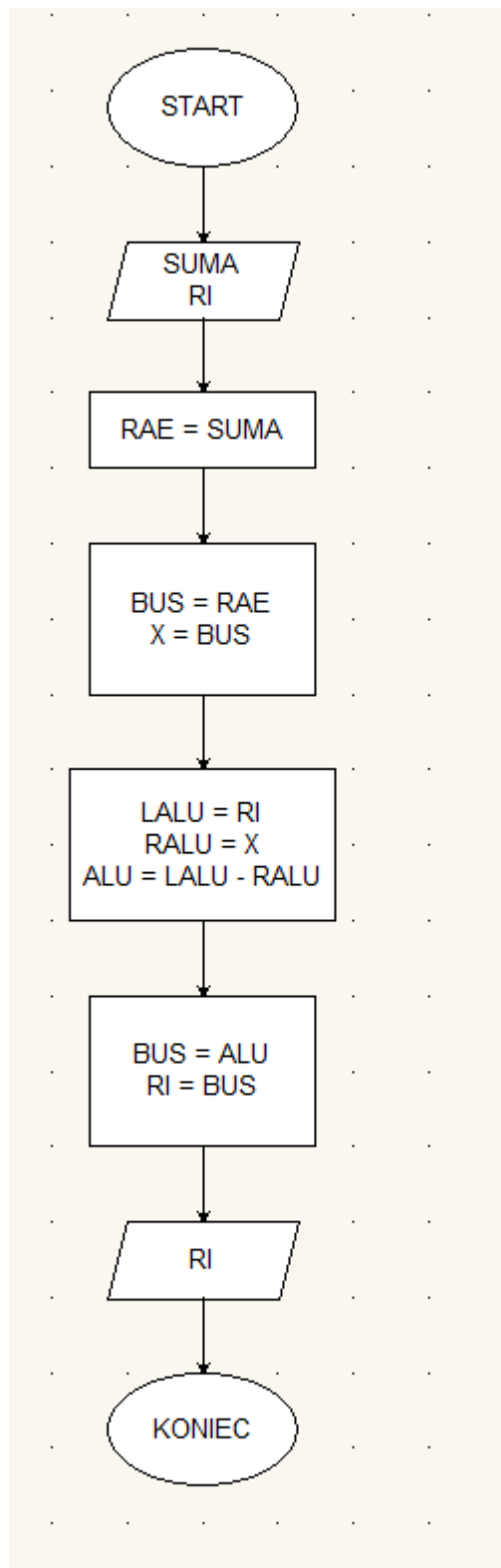
1. OLR : LR -> BUS - 131 / 83h
2. IRAP : BUS -> RAP - 131 / 83h
3. RRC : Rozpoczęcie RRC – przekazanie do RBP wartości (1546 / 60Ah) na podstawie indeksu z RAP (131 / 83h)

4. ORBP : RBP -> BUS - 1546 / 60Ah
5. IRR : BUS -> RR - 1546 / 60Ah
6. (koniec wiersza pamięci mikroprogramów, RAPS = 49 / 31h)
7. ORR : RR -> BUS - 1546 / 60Ah
8. ILK : BUS -> LK – przekazanie wartości na podstawie 7 najmniej znaczących bitów wartości magistrali (1546 / 60Ah). Wynika to z pojemności rejestru LK. W tym przypadku LK = 10 / Ah
9. CEA : Oblicz adres efektywny – mając do czynienia z rozkazem rozszerzonym, do rejestru L trafia wartość 10 / Ah, rejestr R pozostaje wyzerowany. Mając te 2 wartości dodajemy je do siebie, a uzyskany wynik wpisujemy do rejestru SUMA (będzie to wartość 10 / Ah).
10. IRAE : SUMA -> RAE - 10 / Ah
11. NSI : LR+1 -> LR - 132 / 84h (zmieniamy licznik, aby być w stanie pobrać kolejny rozkaz)
12. TEST | TIND : Adresowanie pośrednie – sprawdzamy, czy wystąpiło adresowanie pośrednie. Robimy to poprzez dane dotyczące rozkazu. Jako że mamy do czynienia z rozkazem rozszerzonym, jest to adresacja pośrednia. W związku z tym w RAPS wpisujemy wartość następnej komórki pamięci mikroprogramu, czyli 50 / 32h.
13. ORAE : RAE -> BUS - 10 / Ah
14. IRAP : BUS -> RAP - 10 / Ah
15. RRC : Rozpoczęcie RRC - przekazanie do RBP wartości (100 / 64h) na podstawie indeksu z RAP (10 / Ah)
16. ORBP : RBP -> BUS - 100 / 64h
17. IX : BUS -> X - 100 / 64h
18. (koniec wiersza pamięci mikroprogramów, RAPS = 51 / 33h)
19. OX : X -> BUS - 100 / 64h
20. IBI : BUS -> RAE - 100 / 64h
21. OPC : OP albo AOP+32 -> RAPS – w zależności od typu rozkazu ustawiamy nową wartość pola RAPS. W tym przypadku mając rozkaz rozszerzony podajemy wartość AOP powiększoną o 32. AOP odczytujemy na podstawie 5, 6, 7 oraz 8 bitu zapisu 16-bitowego rozkazu. Oznacza to, że RAPS = 44 / 2Ch
22. TEST | UNB : Zawsze pozytywny – przejście na zadany wiersz pamięci mikroprogramu zawierający wykonanie rozkazu (RAPS = 61 / 3Dh)

Aktualne wartości:

- a) LR = 132 / 84h – adres komórki PAO zawierający adres następnego rozkazu.
- b) RAPS = 61 / 3Dh – adres pierwszego wiersza PM zawierającego mikrorozkazy wykonania aktualnego rozkazu.
- c) RAE = 100 / 64h – pobrana metodą pośrednią, wartość z komórki pamięci o indeksie równym obliczonego adresu efektywnego w kroku 9.

Schemat blokowy:



Opis rozkazu DEX 10 (Zmniejsz modyfikator RI):

Rozkaz DEX zmniejsza wartość rejestru RI o podaną liczbę w rejestrze SUMA (równe liczbie N zadane w rozkazie).

Kolejne działania:

1. TEST | UNB : Zawsze pozytywny – nastąpiło przejście do wiersza PM z wykonaniem rozkazu

2. Wartość SUMA wynosi 10 / Ah, wartość RI wynosi 20 / 14 Ah
3. IRAE : SUMA -> RAE - 10 / Ah
4. ORAE : RAE -> BUS - 10 / Ah
5. IX : BUS -> X - 10 / Ah
6. IXRE : RI -> LALU - 20 / 14h
7. OXE : X -> RALU - 10 / Ah
8. SUB : ALU = LALU – RALU – wpisuje do rejestru ALU wartość rejestru LALU pomniejszoną o wartość rejestru RALU. LALU = 20 / 14h, a RALU = 10 / Ah. W związku z tym ALU = 10 / Ah. Wynik jest dodatni, więc bit znaku nie jest ustawiony, dlatego ZNAK = 0. Argumenty odejmowania są tego samego znaku, dlatego OFF = 0.
9. OBE : ALU -> BUS - 10 / Ah
10. IRI : BUS -> RI - RI = 10 / Ah
11. Wartość RI wynosi 10 / 14h
12. END : Koniec mikroprogramu – przejście do zerowego wiersza PM

Rozkaz dla LR = 132 (rozkaz STX 101 10)

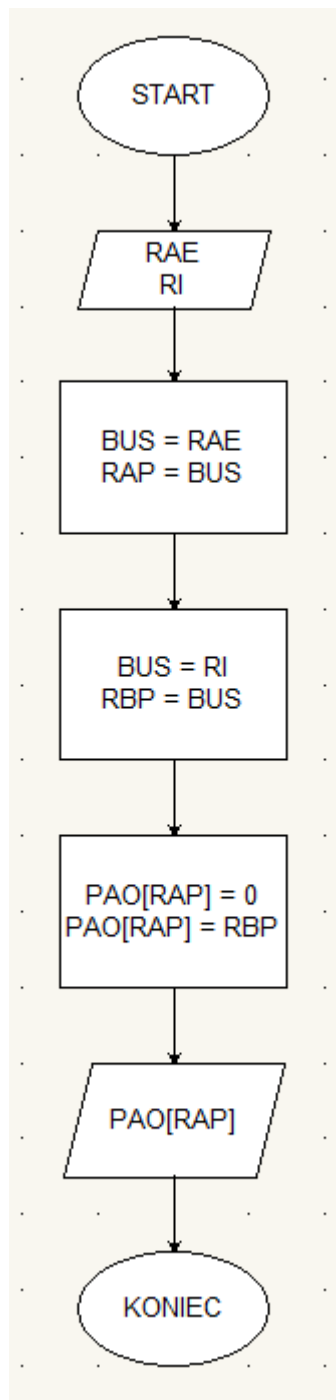
1. OLR : LR -> BUS - 132 / 84h
2. IRAP : BUS -> RAP - 132 / 84h
3. RRC : Rozpoczęcie RRC – przekazanie do RBP wartości (15626 / 3D0Ah) na podstawie indeksu z RAP (132 / 84h)
4. ORBP : RBP -> BUS - 15626 / 3D0Ah
5. IRR : BUS -> RR – 15626 / 3D0Ah
6. (koniec wiersza pamięci mikroprogramów, RAPS = 49 / 31h)
7. ORR : RR -> BUS - 15626 / 3D0Ah
8. ILK : BUS -> LK – przekazanie wartości na podstawie 7 najmniej znaczących bitów wartości magistrali (15626 / 3D0Ah). Wynika to z pojemności rejestru LK. W tym przypadku LK = 10 / Ah.
9. CEA : Oblicz adres efektywny – CEA : Oblicz adres efektywny – mając do czynienia z rozkazem zwykłym o wartościach XSI odpowiednio 101, do rejestru L trafia wartość 10 / Ah, a do rejestru R wartość rejestru RI, czyli 10 / Ah. Mając te 2 wartości dodajemy je do siebie, a uzyskany wynik wpisujemy do rejestru SUMA (będzie to wartość 20 / 14h).
10. IRAE : SUMA -> RAE – 20 / 14h
11. NSI : LR+1 -> LR - 133 / 85h (zmieniamy licznik, aby być w stanie pobrać kolejny rozkaz)
12. TEST | TIND : Adresowanie pośrednie – sprawdzamy, czy wystąpiło adresowanie pośrednie. Robimy to poprzez dane dotyczące rozkazu. Jako że wartość I wynosi 1, mamy do czynienia z adresacją pośrednią. W związku z tym w RAPS wpisujemy wartość następnej komórki pamięci mikroprogramu, czyli 50 / 32h.
13. ORAE : RAE -> BUS - 20 / 14h
14. IRAP : BUS -> RAP – 20 / 14h
15. RRC : Rozpoczęcie RRC – przekazanie do RBP wartości (210 / D2h) na podstawie indeksu z RAP (20 / 14h)
16. ORBP : RBP -> BUS - 210 / D2h
17. IX : BUS -> X - 210 / D2h
18. (koniec wiersza pamięci mikroprogramów, RAPS = 51 / 33h)
19. OX : X -> BUS – 210 / D2h
20. IBI : BUS -> RAE – 210 / D2h

21. OPC : OP albo AOP+32 -> RAPS – w zależności od typu rozkazu ustawiamy nową wartość pola RAPS. W tym przypadku mając rozkaz zwykły podajemy wartość OP, którą określa 5 pierwszych bitów w 16-bitowym zapisie rozkazu. Oznacza to, że $RAPS = 7 / 7h$.
22. TEST | UNB : Zawsze pozytywny – przejście na zadany wiersz pamięci mikroprogramu zawierający wykonanie rozkazu ($RAPS = 67 / 43h$)

Aktualne wartości:

- a) $LR = 133 / 85h$ – adres komórki PAO zawierający adres następnego rozkazu.
- b) $RAPS = 67 / 43h$ – adres pierwszego wiersza PM zawierającego mikrorozkazy wykonania aktualnego rozkazu.
- c) $RAE = 210 / D2h$ – pobrana metodą pośrednią, wartość z komórki pamięci o indeksie równym obliczonego adresu efektywnego w kroku 9.

Schemat blokowy:



Opis rozkazu STX 101 10 (Zapamiętaj modyfikator RI):

Rozkaz STX wpisuje wartość rejestru RI (na podstawie wartości N rozkazu) do komórki pamięci z rejestru RAE.

Kolejne działania:

- 23. TEST | UNB : Zawsze pozytywny – nastąpiło przejście do wiersza PM z wykonaniem rozkazu
- 24. Wartość RAE wynosi 210 / D2h, wartość RI wynosi 10 / Ah
- 25. ORAE : RAE -> BUS - 210 / D2h
- 26. IRAP : BUS -> RAP - 210 / D2h
- 27. ORI : RI -> BUS - 10 / Ah

28. IRBP : BUS -> RBP - 10 / Ah
29. CWC : Rozpoczęcie CWC – wartość PAO w wierszu z numerem o wartości równej wartości rejestru RAP (210) jest zerowana, a następnie uzupełniana wartością rejestru RBP (0x000A)
30. Wartość PAO[210] wynosi 10 / Ah
31. END : Koniec mikroprogramu – przejście do zerowego wiersza PM

Rozkaz dla LR = 133 (rozkaz ADD 111 10)

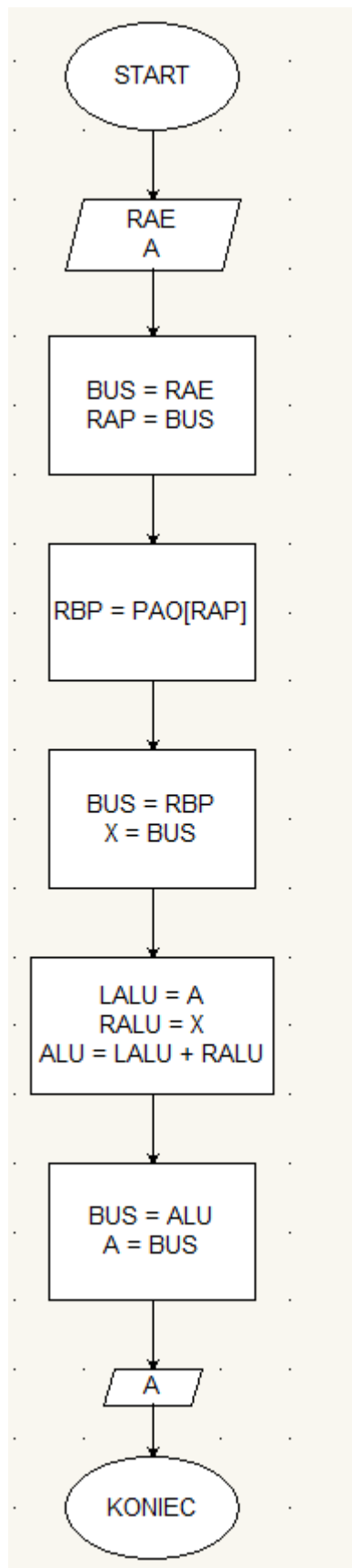
1. OLR : LR -> BUS - 133 / 85h
2. IRAP : BUS -> RAP - 133 / 85h
3. RRC : Rozpoczęcie RRC – przekazanie do RBP wartości (3850 / F0Ah) na podstawie indeksu z RAP (133 / 85h)
4. ORBP : RBP -> BUS - 3850 / F0Ah
5. IRR : BUS -> RR - 3850 / F0Ah
6. (koniec wiersza pamięci mikroprogramów, RAPS = 49 / 31h)
7. ORR : RR -> BUS - 3850 / F0Ah
8. ILK : BUS -> LK – przekazanie wartości na podstawie 7 najmniej znaczących bitów wartości magistrali (3850 / F0Ah). Wynika to z pojemności rejestru LK. W tym przypadku LK = 10 / Ah.
9. CEA : Oblicz adres efektywny – mając do czynienia z rozkazem zwykłym o wartościach XSI odpowiednio 111, rejestry L oraz R są wyzerowane. Dodając je do siebie, SUMA będzie miała wartość 0 / 0h. Jest to niedozwolona metoda adresacji. Program chce przenieść 2 różne wartości w jeden rejestr, co jest nie może mieć miejsca. Jest to określone przez zmianę wartości rejestru XRO na 1.
10. IRAE : SUMA -> RAE – 0 / 0h
11. NSI : LR+1 -> LR - 134 / 86h (zmieniamy licznik, aby być w stanie pobrać kolejny rozkaz)
12. TEST | TIND : Adresowanie pośrednie – sprawdzamy, czy wystąpiło adresowanie pośrednie. Robimy to poprzez dane dotyczące rozkazu. Jako że wartość I wynosi 1, mamy do czynienia z adresacją pośrednią. W związku z tym w RAPS wpisujemy wartość następnej komórki pamięci mikroprogramu, czyli 50 / 32h.
13. ORAE : RAE -> BUS – 0 / 0h
14. IRAP : BUS -> RAP – 0 / 0h
15. RRC : Rozpoczęcie RRC – przekazanie do RBP wartości (255 / FFh) na podstawie indeksu z RAP (0 / 0h)
16. ORBP : RBP -> BUS - 255 / FFh
17. IX : BUS -> X – 255 / FFh
18. (koniec wiersza pamięci mikroprogramów, RAPS = 51 / 33h)
19. OX : X -> BUS – 255 / FFh
20. IBI : BUS -> RAE - 255 / FFh
32. OPC : OP albo AOP+32 -> RAPS – OPC : OP albo AOP+32 -> RAPS – w zależności od typu rozkazu ustawiamy nową wartość pola RAPS. W tym przypadku mając rozkaz zwykły podajemy wartość OP, którą określa 5 pierwszych bitów w 16-bitowym zapisie rozkazu. Oznacza to, że RAPS = 1 / 1h.
21. TEST | UNB : Zawsze pozytywny – przejście na zadany wiersz pamięci mikroprogramu zawierający wykonanie rozkazu (RAPS = 52 / 34h)

Aktualne wartości:

- d) LR = 134 / 86h – adres komórki PAO zawierający adres następnego rozkazu.
- e) RAPS = 52 / 34h – adres pierwszego wiersza PM zawierającego mikrorozkazy wykonania aktualnego rozkazu.

- f) $RAE = 255 / FFh$ – pobrana metodą pośrednią, wartość z komórki pamięci o indeksie równym obliczonego adresu efektywnego w kroku 9.

Schemat blokowy:



Opis rozkazu ADD 111 10 (Dodawanie):

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

Kolejne działania:

33. TEST | UNB : Zawsze pozytywny – nastąpiło przejście do wiersza PM z wykonaniem rozkazu
34. Wartość RAE wynosi 255 / FFh, wartość A wynosi -2000 / F830h.
35. ORAE : RAE -> BUS - 255 / FFh
36. IRAP : BUS -> RAP - 255 / FFh
37. RRC : Rozpoczęcie RRC – przekazanie do RBP wartości (-2021 / F81Bh) na podstawie indeksu z RAP (255 / FFh)
38. ORBP : RBP -> BUS - -2021 / F81Bh
39. IX : BUS -> X – -2021 / F81Bh
40. (koniec wiersza pamięci mikroprogramów, RAPS = 53 / 35h)
41. IALU : A -> LALU - -2000 / F830h
42. OXE : X -> RALU - -2021 / F81Bh
43. ADD : ALU = LALU + RALU – wpisuje do rejestru ALU wartość sumy wartości w rejestrach LALU oraz RALU. W LALU znajduje się wartość -2000 / F830h, natomiast w RALU -2021 / F81Bh. W związku z tym ALU = -4021 / F04Bh. Wynik jest ujemny, ZNAK = 1. Argumenty dodawania są tego samego znaku, dlatego OFF = 0.
44. OBE : ALU -> BUS - -4021 / F04Bh
45. IA : BUS -> A – -4021 / F04Bh
46. Wartość A wynosi -4021 / F04Bh
47. END : Koniec mikroprogramu – przejście do zerowego wiersza PM

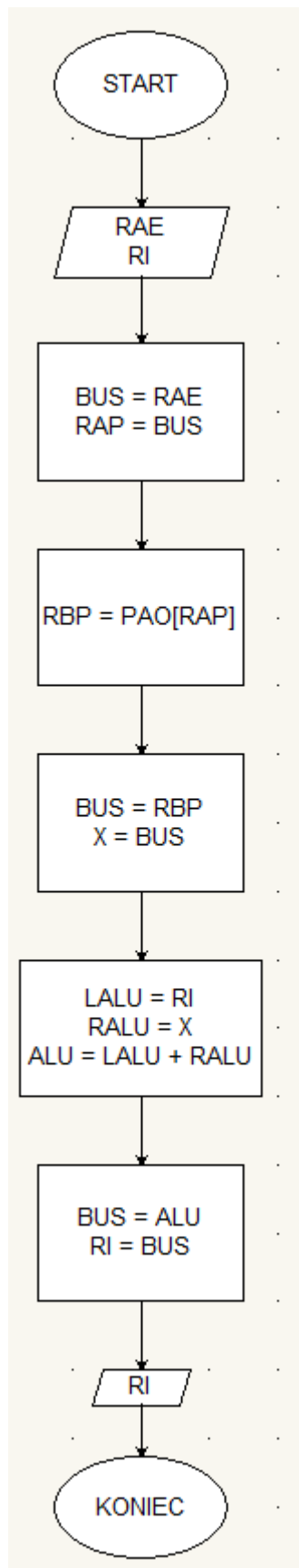
Rozkaz dla LR = 134 (rozkaz ADX 010 1)

1. OLR : LR -> BUS - 134 / 86h
2. IRAP : BUS -> RAP - 134 / 86h
3. RRC : Rozpoczęcie RRC – przekazanie do RBP wartości (27137 / 6A01h) na podstawie indeksu z RAP (134 / 86h)
4. ORBP : RBP -> BUS - 27137 / 6A01h
5. IRR : BUS -> RR - 27137 / 6A01h
6. (koniec wiersza pamięci mikroprogramów, RAPS = 49 / 31h)
7. ORR : RR -> BUS - 27137 / 6A01h
8. ILK : BUS -> LK – przekazanie wartości na podstawie 7 najmniej znaczących bitów wartości magistrali (27137 / 6A01h). Wynika to z pojemności rejestru LK. W tym przypadku LK = 1 / 1h.
1. CEA : Oblicz adres efektywny – mając do czynienia z rozkazem zwykłym o wartościach XSI odpowiednio 010, do rejestru L trafia wartość DA rozkazu (1 / 1h), a do rejestru R wartość rejestru LR, czyli 134 / 86h. Mając te 2 wartości dodajemy je do siebie, a uzyskany wynik wpisujemy do rejestru SUMA (będzie to wartość 135 / 87h).
9. IRAE : SUMA -> RAE – 135 / 87h
10. NSI : LR+1 -> LR - 135 / 87h (zmieniamy licznik, aby być w stanie pobrać kolejny rozkaz)
11. TEST | TIND : Adresowanie pośrednie – sprawdzamy, czy wystąpiło adresowanie pośrednie. Robimy to poprzez dane dotyczące rozkazu. Jako że jest to rozkaz zwykły, a bit I jest ustawiony na 0, oznacza to, że nie adresacja nie jest pośrednia. W związku z tym w RAPS wpisujemy wartość pola OP rozkazu, czyli 13 / Dh.
12. TEST | UNB : Zawsze pozytywny – przejście na zadany wiersz pamięci mikroprogramu zawierający wykonanie rozkazu (RAPS = 56 / 38h)

Aktualne wartości:

- g) $LR = 135 / 87h$ – adres komórki PAO zawierający adres następnego rozkazu.
- h) $RAPS = 56 / 38h$ – adres pierwszego wiersza PM zawierającego mikrorozkazy wykonania aktualnego rozkazu.
- i) $RAE = 135 / 87h$ – pobrana metodą bezpośrednią, wartość równa adresowi efektywnemu obliczonemu w kroku 9.

Schemat blokowy:



Opis rozkazu ADX 010 1 (Dodaj do RI):

Rozkaz ADX dodaje do rejestru RI wartość pamięci o adresie równym wartości rejestru RAE.

Kolejne działania:

48. TEST | UNB : Zawsze pozytywny – nastąpiło przejście do wiersza PM z wykonaniem rozkazu
49. Wartość RAE wynosi 135 / 87h, wartość RI wynosi 10 / Ah.
50. ORAE : RAE -> BUS – 135 / 87h
51. IRAP : BUS -> RAP – 135 / 87h
52. RRC : Rozpoczęcie RRC – przekazanie do RBP wartości (24074 / 5E0Ah) na podstawie indeksu z RAP (135 / 87h)
53. ORBP : RBP -> BUS - 24074 / 5E0Ah
54. IX : BUS -> X – 24074 / 5E0Ah
55. IXRE : RI -> LALU - 10 / Ah
56. OXE : X -> RALU - 24074 / 5E0Ah
57. ADD : ALU = LALU + RALU – wpisuje do rejestru ALU wartość sumy wartości w rejestrach LALU oraz RALU. LALU = 10 / Ah, a RALU = 24074 / 5E0Ah. W związku z tym ALU = 24084 / 5E14h. Wynik jest dodatni, więc bit znaku nie jest ustawiony, dlatego ZNAK = 0. Argumenty dodawania są tego samego znaku, dlatego OFF = 0.
58. OBE : ALU -> BUS - 24084 / 5E14h
59. IRI : BUS -> RI - 24084 / 5E14h
60. Wartość RI wynosi -- 24084 / 5E14h
61. END : Koniec mikroprogramu – przejście do zerowego wiersza PM

Rozkaz dla LR = 135 (rozkaz TXA 110 10)

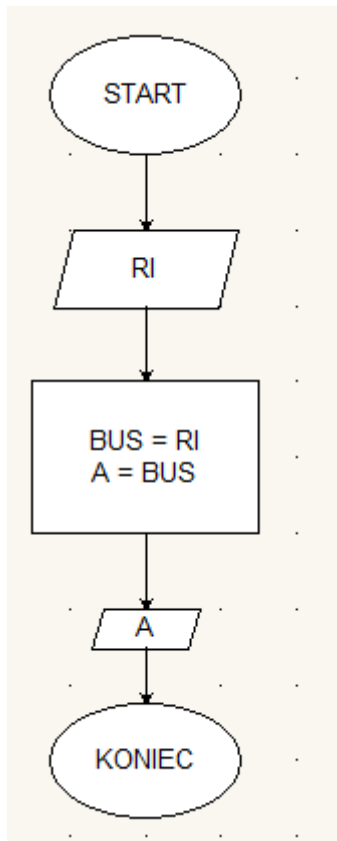
1. OLR : LR -> BUS - 135 / 87h
2. IRAP : BUS -> RAP - 135 / 87h
3. RRC : Rozpoczęcie RRC – przekazanie do RBP wartości (24074 / 5E0Ah) na podstawie indeksu z RAP (135 / 87h)
4. ORBP : RBP -> BUS - 24074 / 5E0Ah
5. IRR : BUS -> RR - 24074 / 5E0Ah
6. (koniec wiersza pamięci mikroprogramów, RAPS = 49 / 31h)
7. ORR : RR -> BUS - 24074 / 5E0Ah
8. ILK : BUS -> LK – przekazanie wartości na podstawie 7 najmniej znaczących bitów wartości magistrali (24074 / 5E0Ah). Wynika to z pojemności rejestru LK. W tym przypadku LK = 10 / Ah.
9. CEA : Oblicz adres efektywny – mając do czynienia z rozkazem zwykłym o wartościach XSI odpowiednio 110, rejestry L oraz R są wyzerowane. Dodając je do siebie, SUMA będzie miała wartość 0 / 0h. Jest to niedozwolona metoda adresacji. Program chce przenieść 2 różne wartości w jeden rejestr, co jest nie może mieć miejsca. Jest to określone przez zmianę wartości rejestru XRO na 1.
10. IRAE : SUMA -> RAE – 0 / 0h
11. NSI : LR+1 -> LR - 136 / 88h (zmieniamy licznik, aby być w stanie pobrać kolejny rozkaz)
12. TEST | TIND : Adresowanie pośrednie – sprawdzamy, czy wystąpiło adresowanie pośrednie. Robimy to poprzez dane dotyczące rozkazu. Jako że jest to rozkaz zwykły, a bit I jest ustawiony na 0, oznacza to, że nie adresacja nie jest pośrednia. W związku z tym w RAPS wpisujemy wartość pola OP rozkazu, czyli 11 / Bh.
13. TEST | UNB : Zawsze pozytywny – przejście na zadany wiersz pamięci mikroprogramu zawierający wykonanie rozkazu (RAPS = 75 / 4Bh).

Aktualne wartości:

- a) LR = 136 / 88h – adres komórki PAO zawierający adres następnego rozkazu.

- b) RAPS = 75 / 4Bh – adres pierwszego wiersza PM zawierającego mikrorozkazy wykonania aktualnego rozkazu.
- c) RAE = 0 / 0h – pobrana metodą bezpośrednią, wartość równa adresowi efektywnemu obliczonemu w kroku 9 (niedozwolona metoda adresacji).

Schemat blokowy:



Opis rozkazu TXA 110 10 (Prześlij RI do A):

Rozkaz TXA przekazuje wartość rejestru RI do rejestru A.

Kolejne działania:

- 62. TEST | UNB : Zawsze pozytywny – nastąpiło przejście do wiersza PM z wykonaniem rozkazu
- 63. Wartość RI wynosi 24084 / 5E14h.
- 64. ORI : RI -> BUS - 24084 / 5E14h
- 65. IA : BUS -> A - 24084 / 5E14h
- 66. Wartość A wynosi -- 24084 / 5E14h.
- 67. END : Koniec mikroprogramu – przejście do zerowego wiersza PM

Ze względu na dopisanie do sprawozdania elementów dokończonych po przestaniu pliku z logiem oraz zrzutem ekranu przesyłam zaktualizowany końcowy efekt programu:

Komputer LabZSK - sterowanie mikroprogramowe

Simulacja Mikroprogramy Pamięć operacyjna Info

Adres	S1	D1	S2	D2	S3	D3	C1	C2	Test	ALU	NA
71	ORAE	IRAP			ORBP	IRI	RRC				
72							END				
73	ORAE	IRAP			OLR	IRBP	CWC				
74							END				
75			ORI	IA			END				
76			OMO	IA			END				

Pokaż log

MAKRO
MIKRO

Stop

Ocena 4
Błędy 4
Takt 0
Cykl 42

Adres	Zawartość	Hex
135	0101111000001010	5E0A
136		
137		
138		
139		
140		
141		
142		
143		
144		
145		
146		
147		
148		
149		
150		
151		
152		
153		
154		
155		
156		
157		
158		

PAO[135] - ROZKAZ ZWYKŁY
OP = Bh (TXA)
X = 1
S = 1
I = 0
DA = Ah
UT18:29 > 19:01 + 02:13.03

21:14
10.11.2021