

# Architektura i organizacja komputerów

Sprawozdanie z laboratorium nr 5

Temat zajęć: Wprowadzenie do  
komputera DLX

Prowadzący zajęcia: Miktus Artur

Wykonawca sprawozdania:  
Borkowski Kamil WCY22IY1S1

Data wykonania: 2023.12.07

A. Rozkazy przesłań: LW

Rozkaz ten przesyła 4 bajty z komórki pamięci operacyjnej o zadanym adresie do rejestru docelowego.

Zestaw 1

Kod programu:

.data

l\_A: .word 6

.text

lw r3, l\_A

trap 0

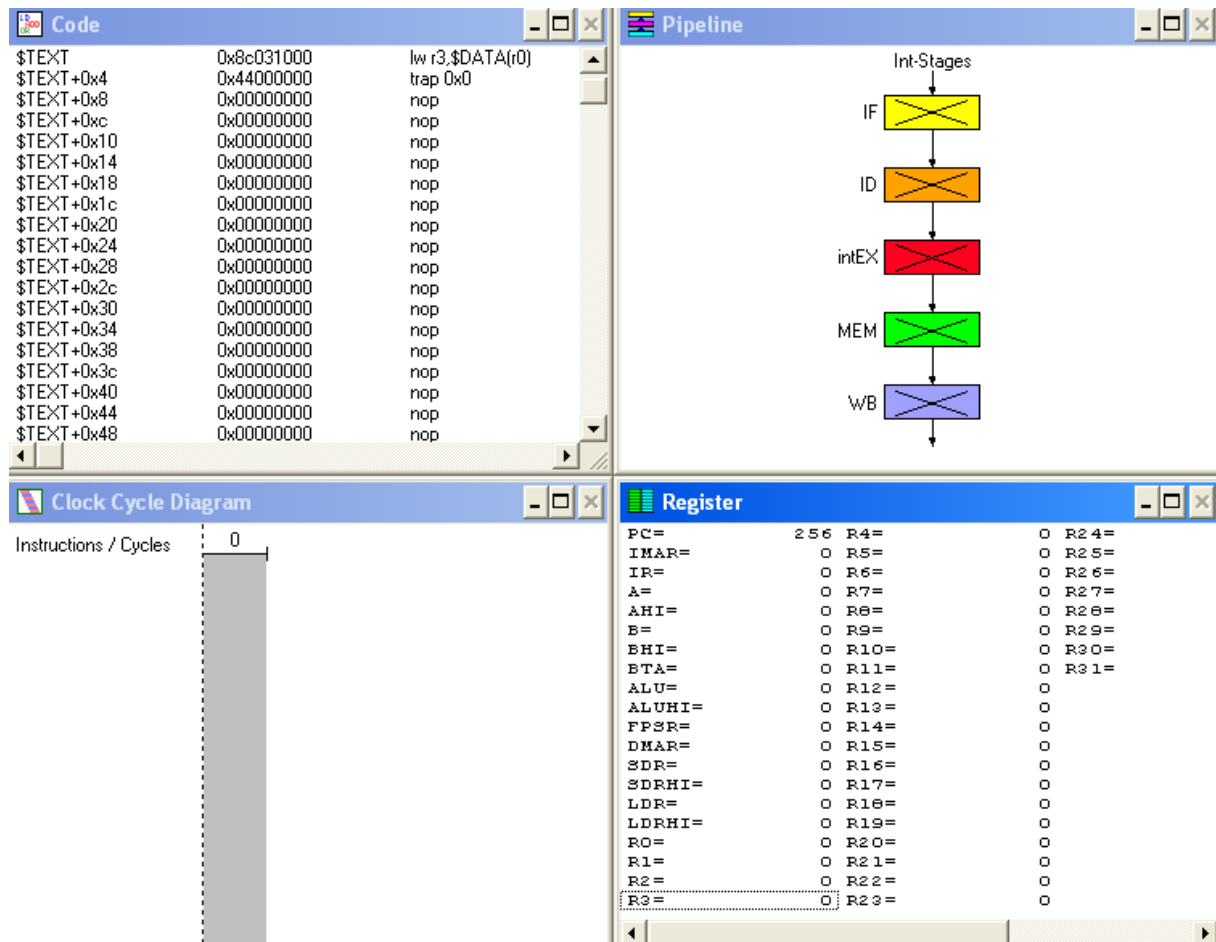
Dane wejściowe to etykieta l\_A przechowująca word 6.

Spodziewany rezultat to wartość rejestru R3 będzie równa 6.

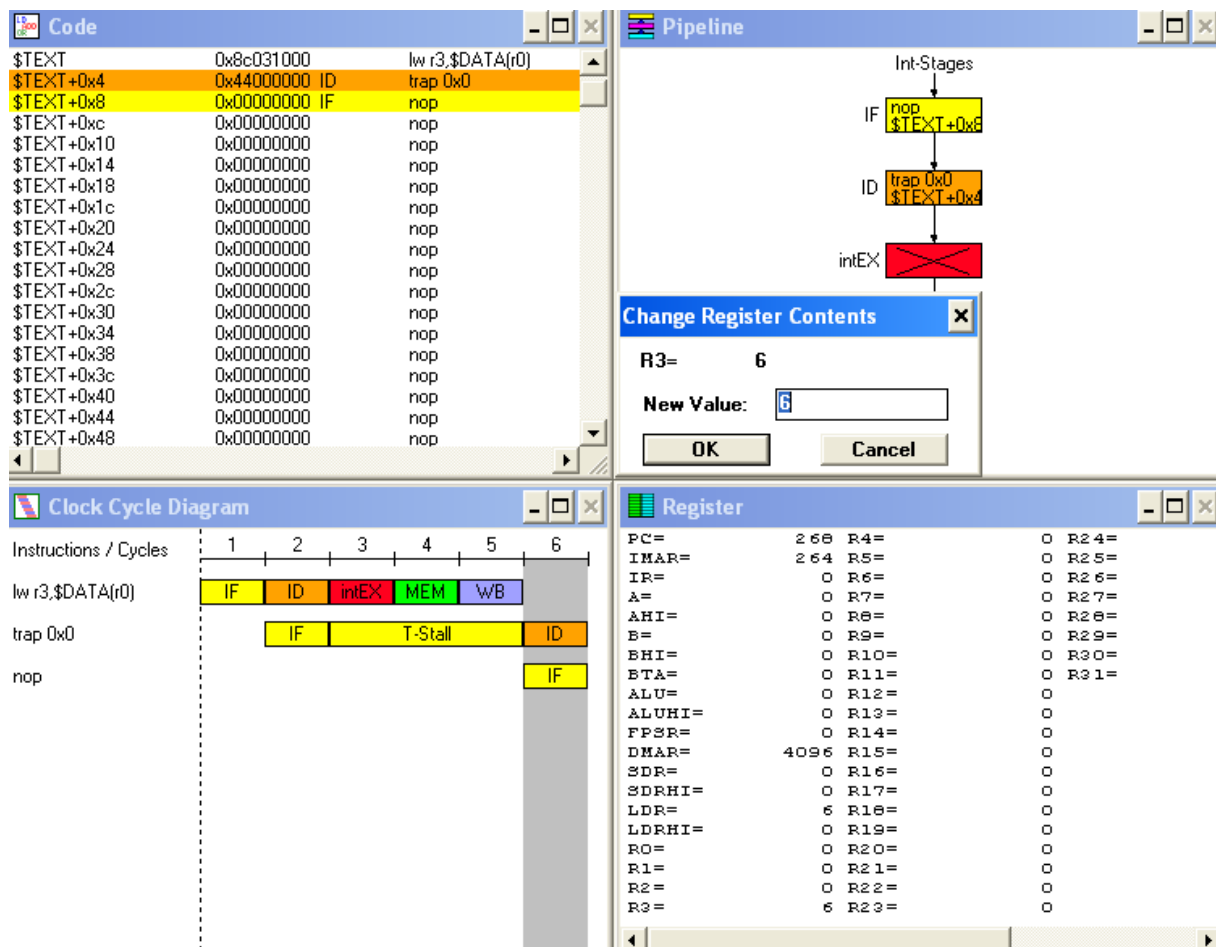
Po wykonaniu programu wartość rejestru R3 przyjęła wartość 6.

Zrzut ekranu, przedstawiający załadowany program i zawartość rejestrów i komórek PAO przed i po wykonaniu:

Przed:



Po:



## Zestaw 2

Kod programu:

```
.data
```

```
l_A: .word -9000000000
```

```
.text
```

```
lw r3, l_A
```

```
trap 0
```

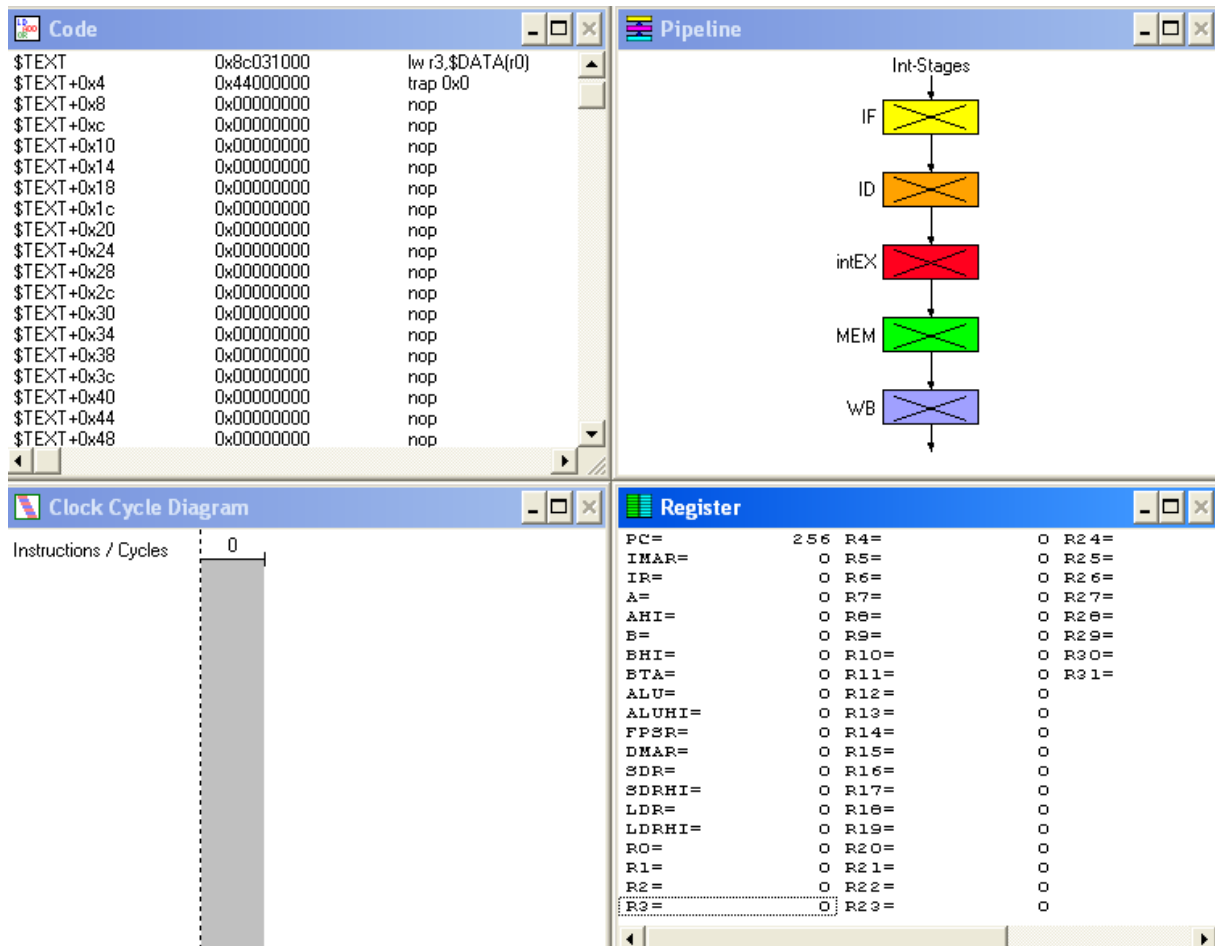
Dane wejściowe to etykieta l\_A przechowująca word -9000000000.

Spodziewany rezultat to wartość rejestru R3 będzie równa -9000000000.

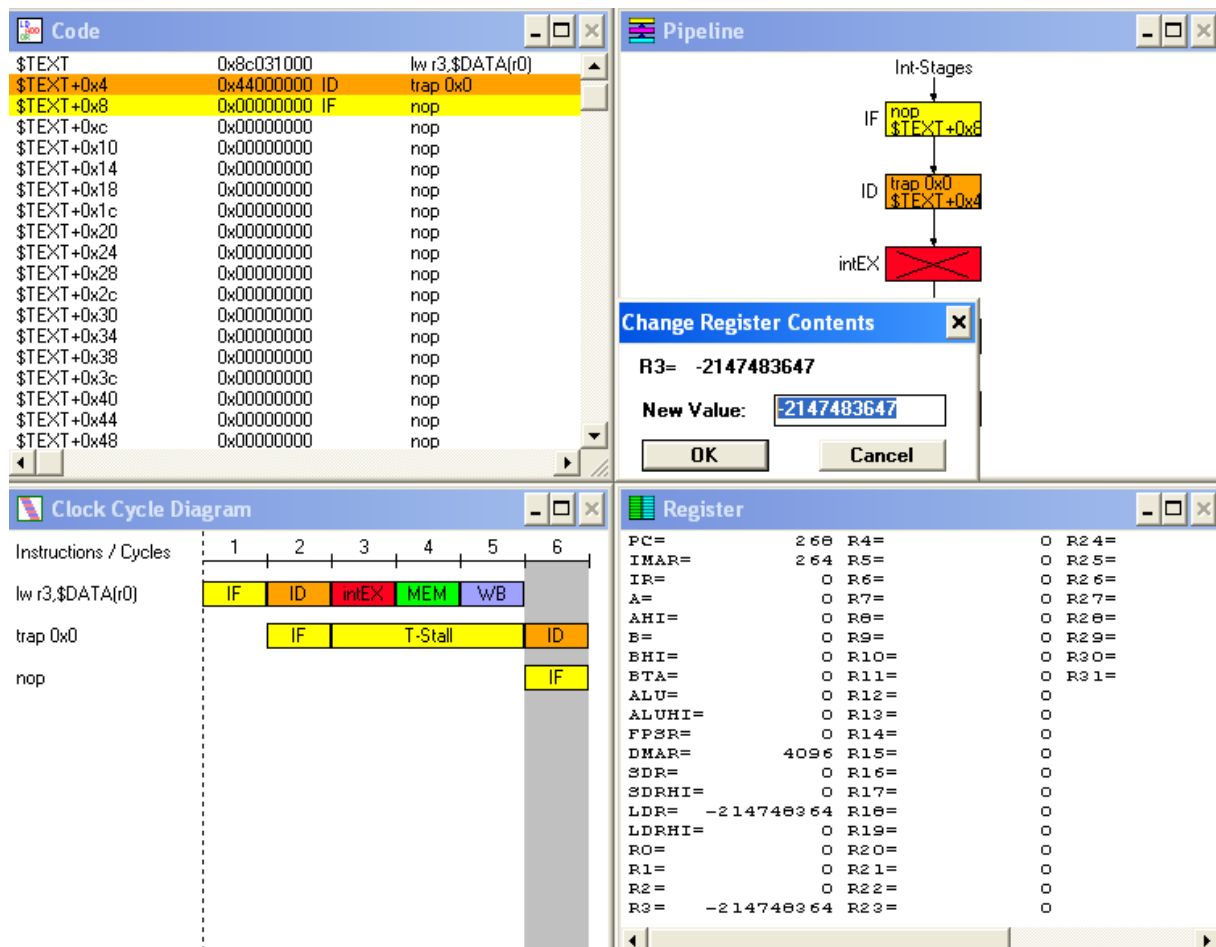
Po wykonaniu programu wartość rejestru R3 będzie równa -2147483647. Jest to minimalna wartość jaką można zapisać na 4 bajtach.

Zrzut ekranu, przedstawiający załadowany program i zawartość rejestrów i komórek PAO przed i po wykonaniu:

Przed:



Po:



## B. Arytmetyczne i logiczne:ADD

### Zestaw 1

Rozkaz ten wykonuje dodawanie wartości z podanych rejestrów i ich sumę zapisuje w podanym rejestrze.

Kod programu:

.data

l\_A: .word 10

l\_B: .word 5

.text

lw r3, l\_A

lw r2, l\_B

ADD r1, r2, r3

trap 0

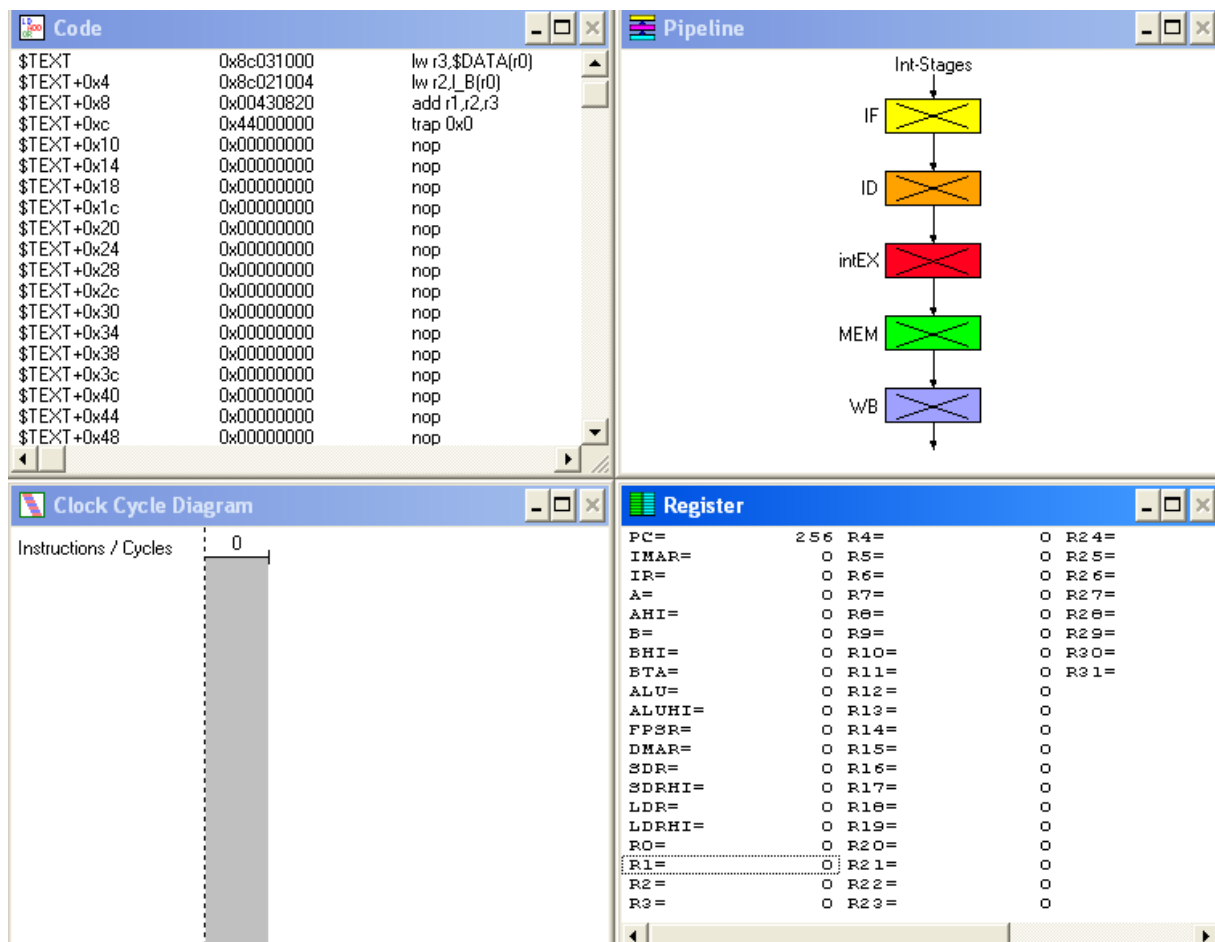
Dane wejściowe to etykieta l\_A przechowująca word 10 oraz etykieta l\_B przechowująca word 5.

Spodziewany rezultat to wartość rejestru R1 równa 15.

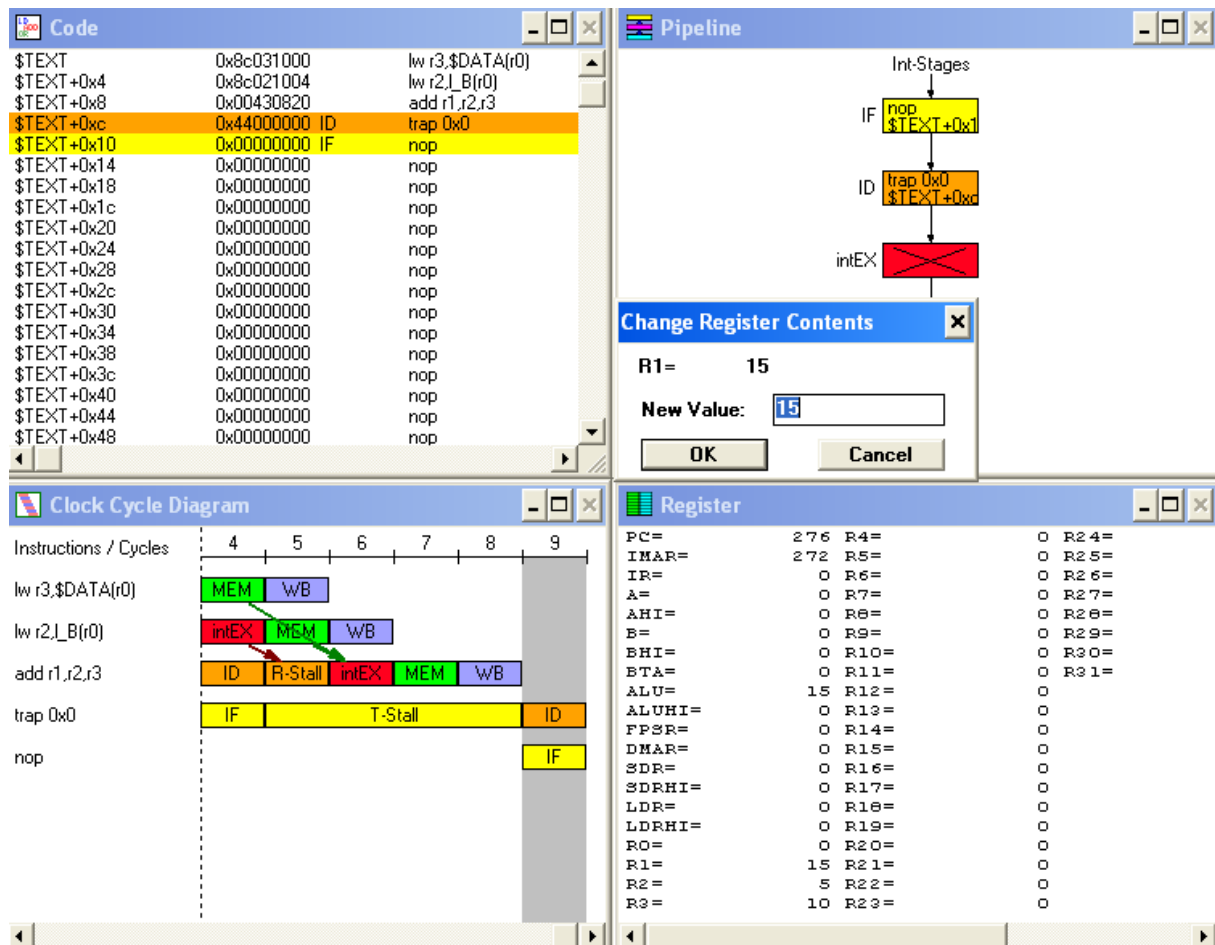
Po wykonaniu programu rejestr R1 przyjmuje wartość 15.

Zrzut ekranu, przedstawiający załadowany program i zawartość rejestrów i komórek PAO przed i po wykonaniu:

Przed:



Po:



## Zestaw 2

Kod programu:

.data

l\_A: .word 1000000000

l\_B: .word 2000000000

.text

lw r3, l\_A

lw r2, l\_B

ADD r1, r2, r3



trap 0

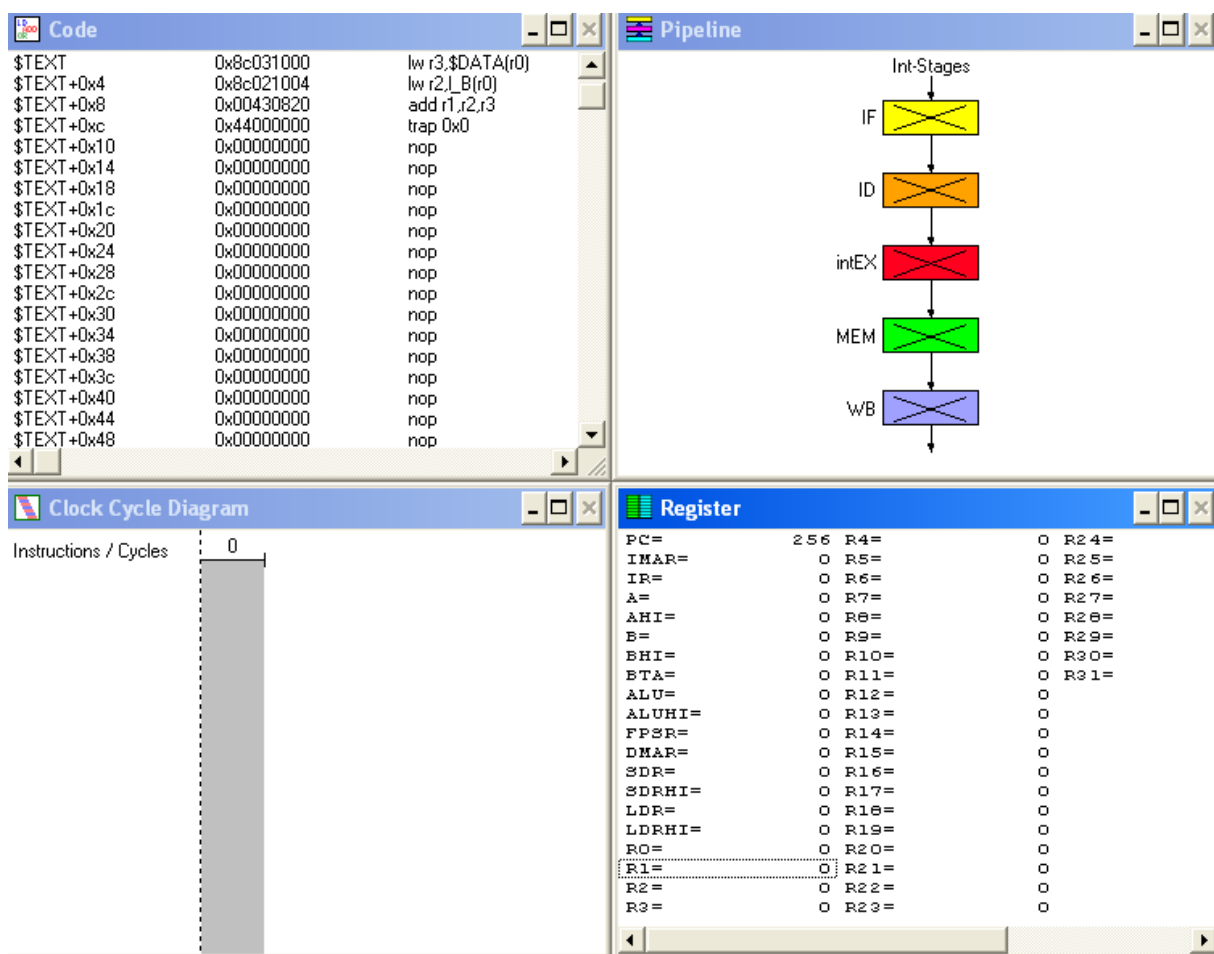
Dane wejściowe to etykieta l\_A przechowująca word 1000000000 oraz etykieta l\_B przechowująca word 2000000000.

Spodziewany rezultat to wartość rejestru R1 równa 3000000000.

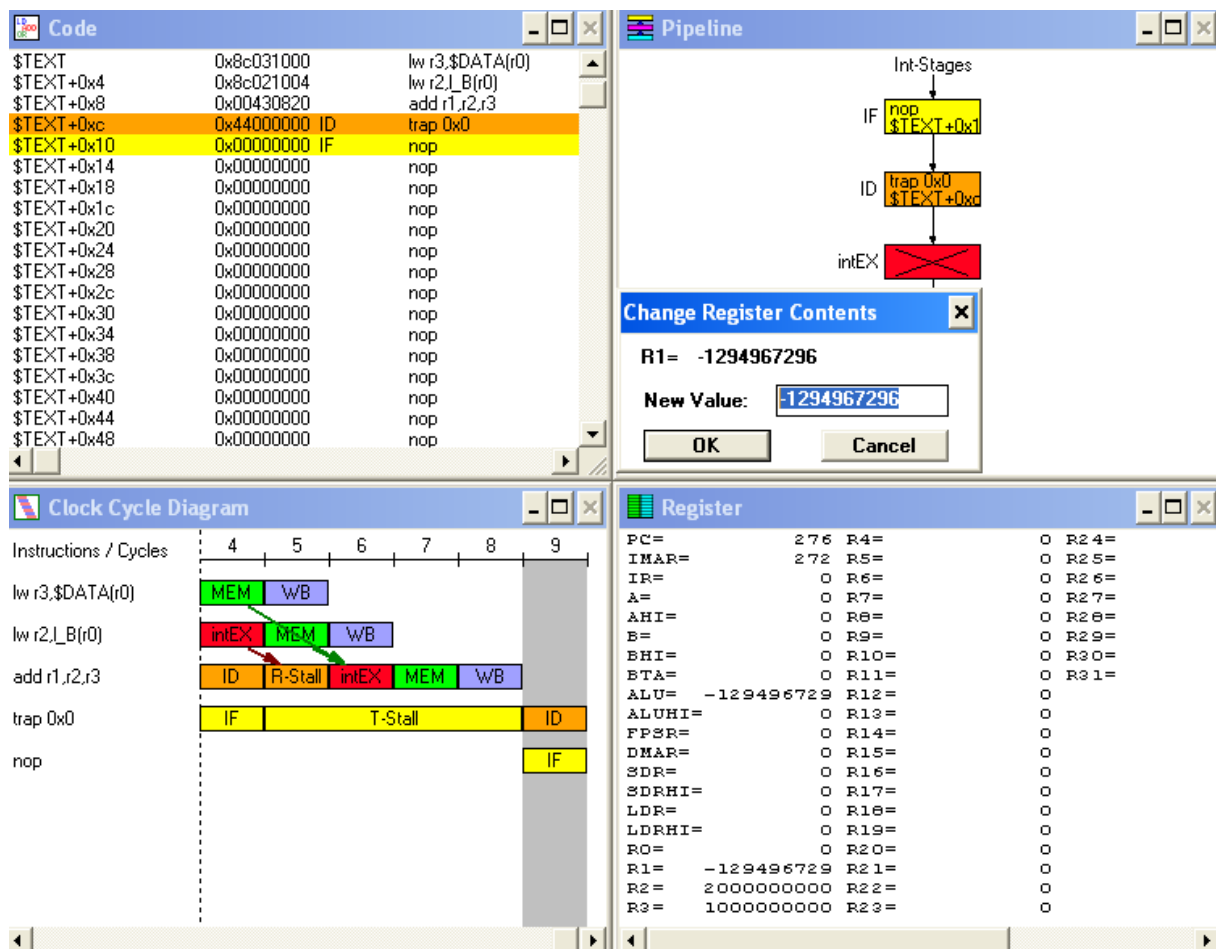
Po wykonaniu programu rejestr R1 przyjmuje wartość -1294967296.

Zrzut ekranu, przedstawiający załadowany program i zawartość rejestrów i komórek PAO przed i po wykonaniu:

Przed:



Po:



### C. Rozkazy skoków: BEQZ

Rozkaz ten wykonuje skok pod warunkiem, że sprawdzana przez niego wartość jest równa 0.

Zestaw 1

Kod programu:

.data

l\_A: .word 0

.text

lw r1, l\_A

lw r2, 0(r1)

p1:

ADDI r2, r2, 1

BEQZ r2, p1

trap 0

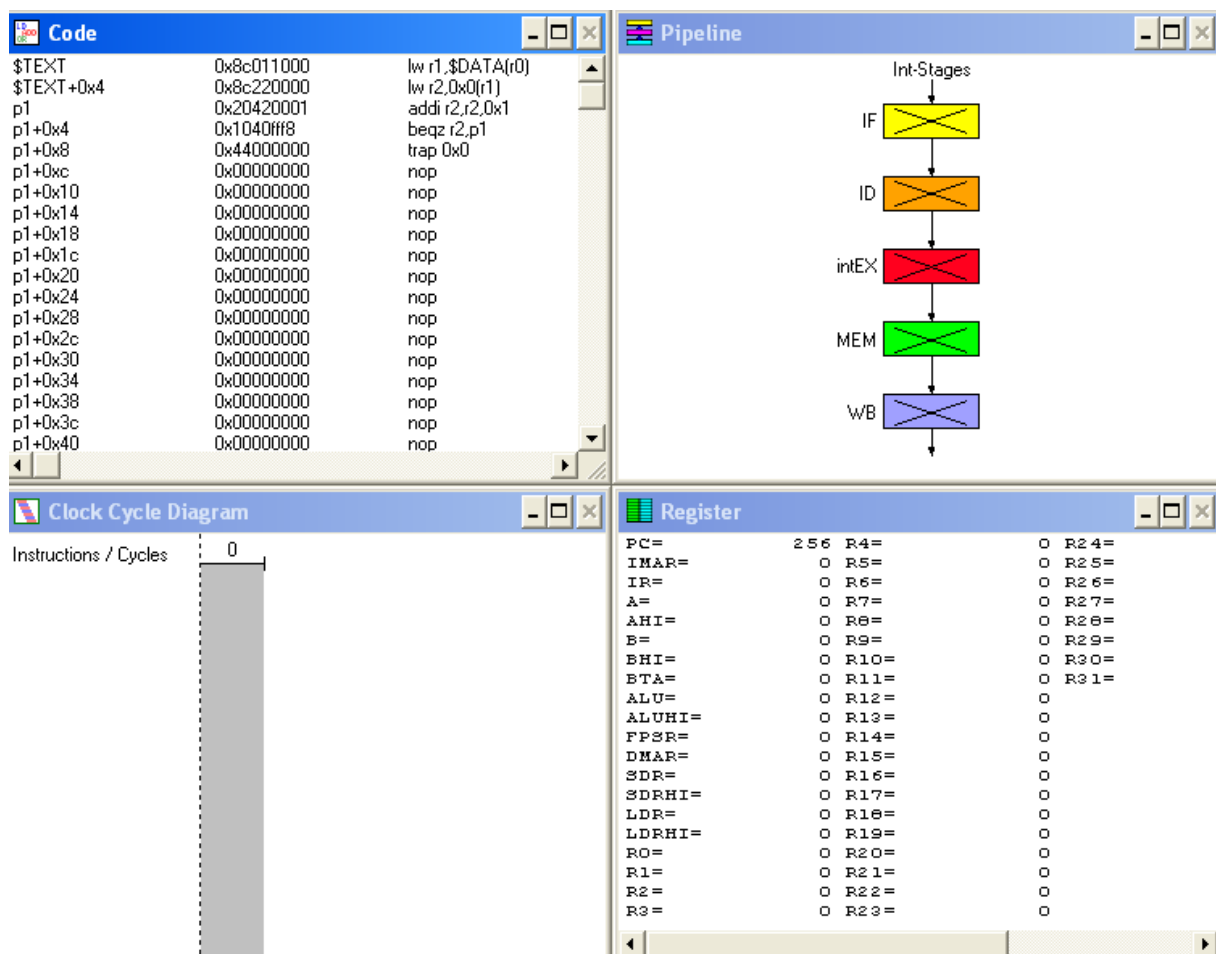
Dane wejściowe to etykieta l\_A przechowująca word 0.

Spodziewany rezultat to wartość rejestru R2 będzie równa 1.

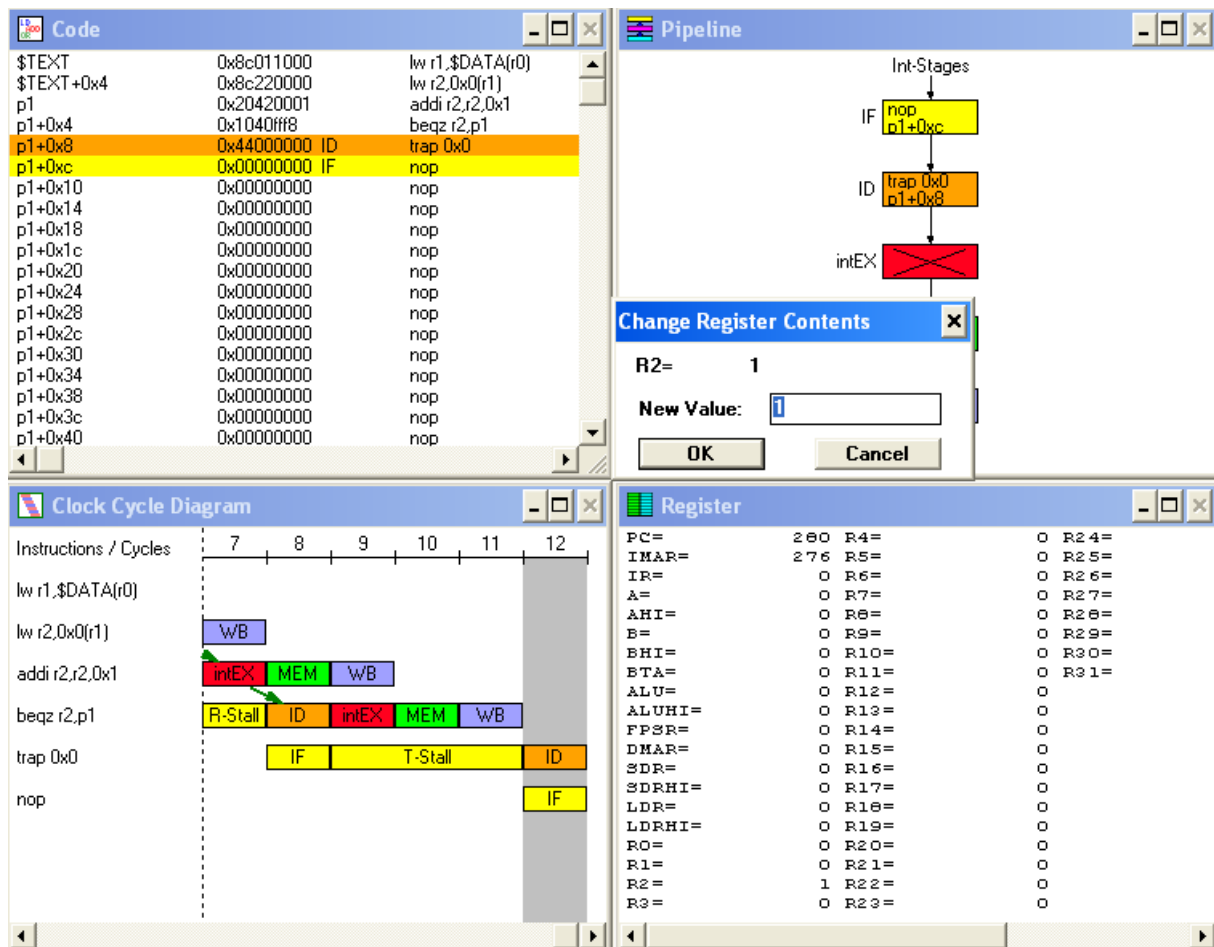
Po wykonaniu programu wartość rejestru R2 przyjęła wartość 1.

Zrzut ekranu, przedstawiający załadowany program i zawartość rejestrów i komórek PAO przed i po wykonaniu:

Przed:



Po:



## Zestaw 2

Kod programu:

.data

l\_A: .word 5

.text

lw r1, l\_A

lw r2, 0(r1)

p1:

ADDI r2, r2, 1

BEQZ r2, p1

trap 0

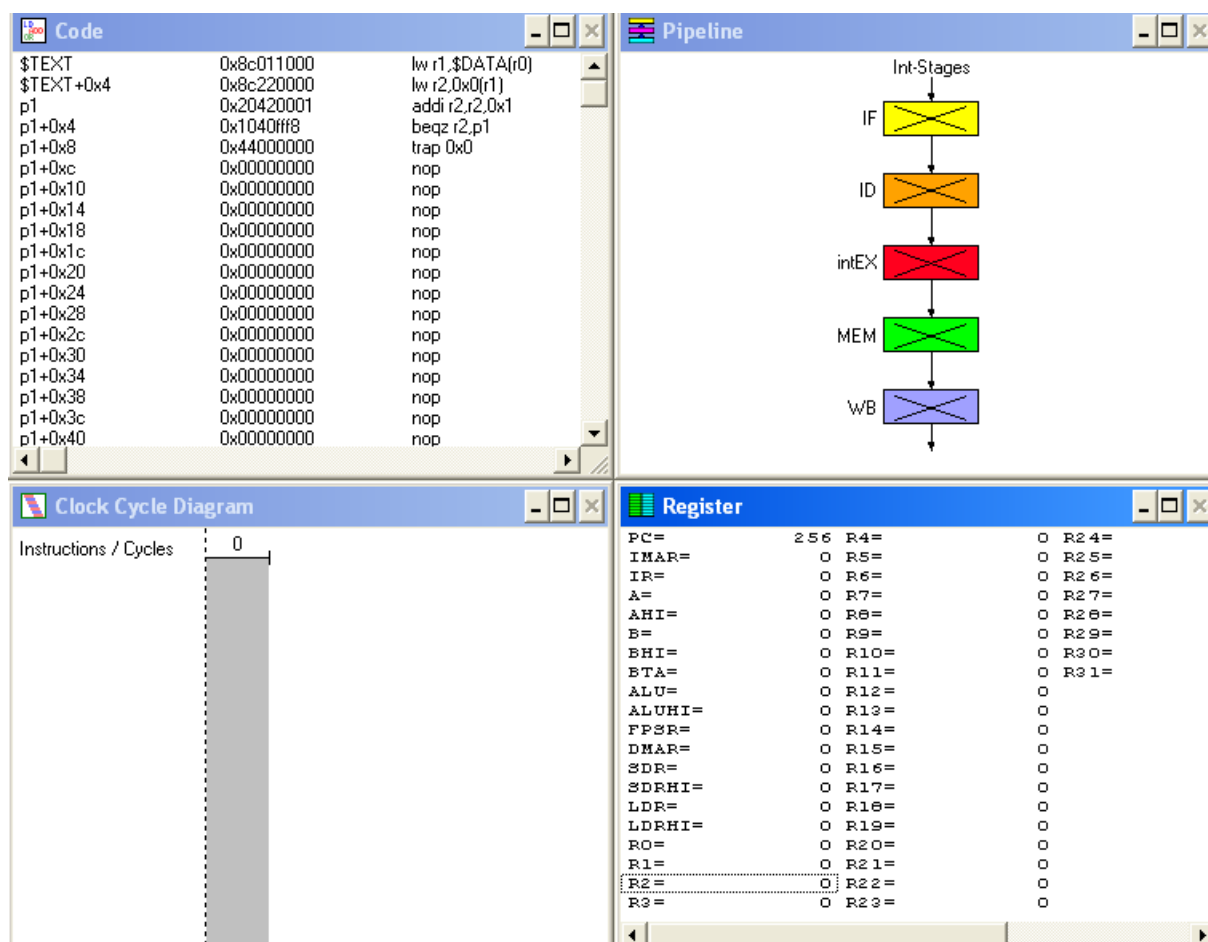
Dane wejściowe to etykieta l\_A przechowująca word 5.

Spodziewany rezultat to wartość rejestru R1 będzie równa 5 oraz R2 będzie równy 0.

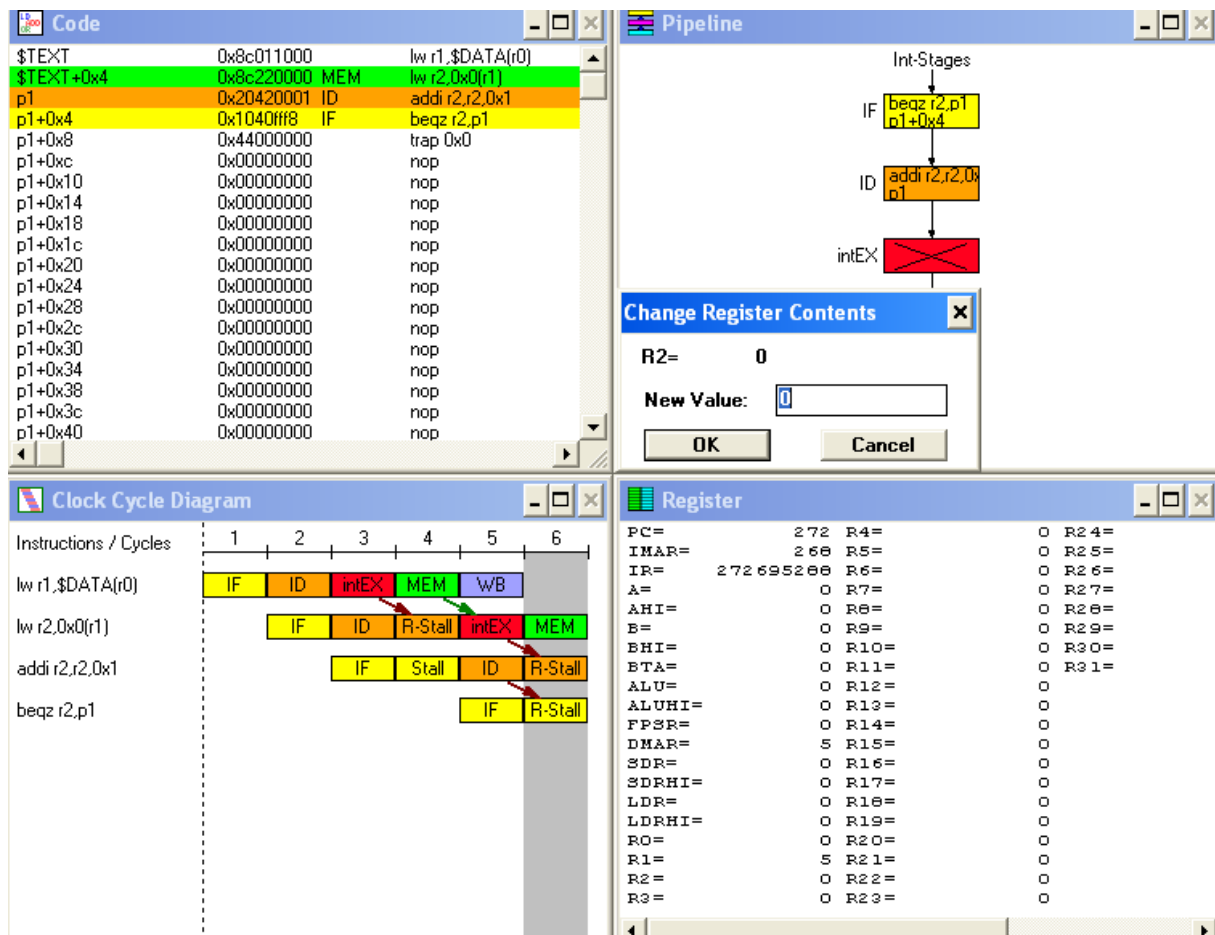
Po wykonaniu programu wartość rejestru R1 przyjęła wartość 5 oraz R2 równa się 0.

Zrzut ekranu, przedstawiający załadowany program i zawartość rejestrów i komórek PAO przed i po wykonaniu:

Przed:



Po:



## D. Rozkazy zmiennoprzecinkowe: ADDF

Rozkaz ten wykonuje dodawanie wartości zmiennoprzecinkowych z podanych rejestrów i ich sumę zapisuje w rejestrze docelowym.

Zestaw 1

Kod programu:

`.data`

`l_A: .float 10.1`

`l_B: .float 5.4`

`.text`

`lf f1, l_A`

`lf f2, l_B`

ADDF f3, f1, f2

trap 0

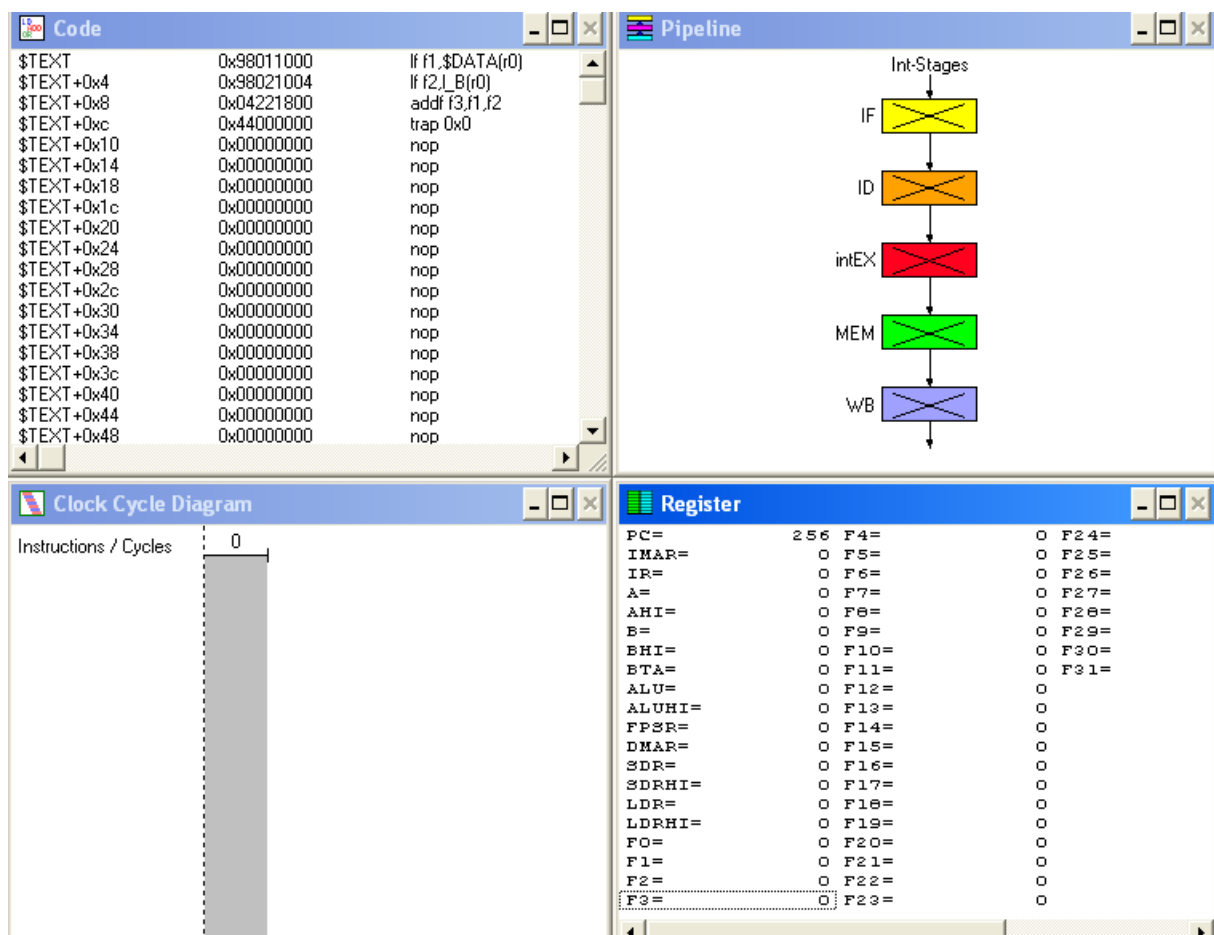
Dane wejściowe to etykieta l\_A przechowująca float 10.1 oraz l\_B przechowująca 5.4.

Spodziewany rezultat to wartość rejestru F3 będzie równa 15.5.

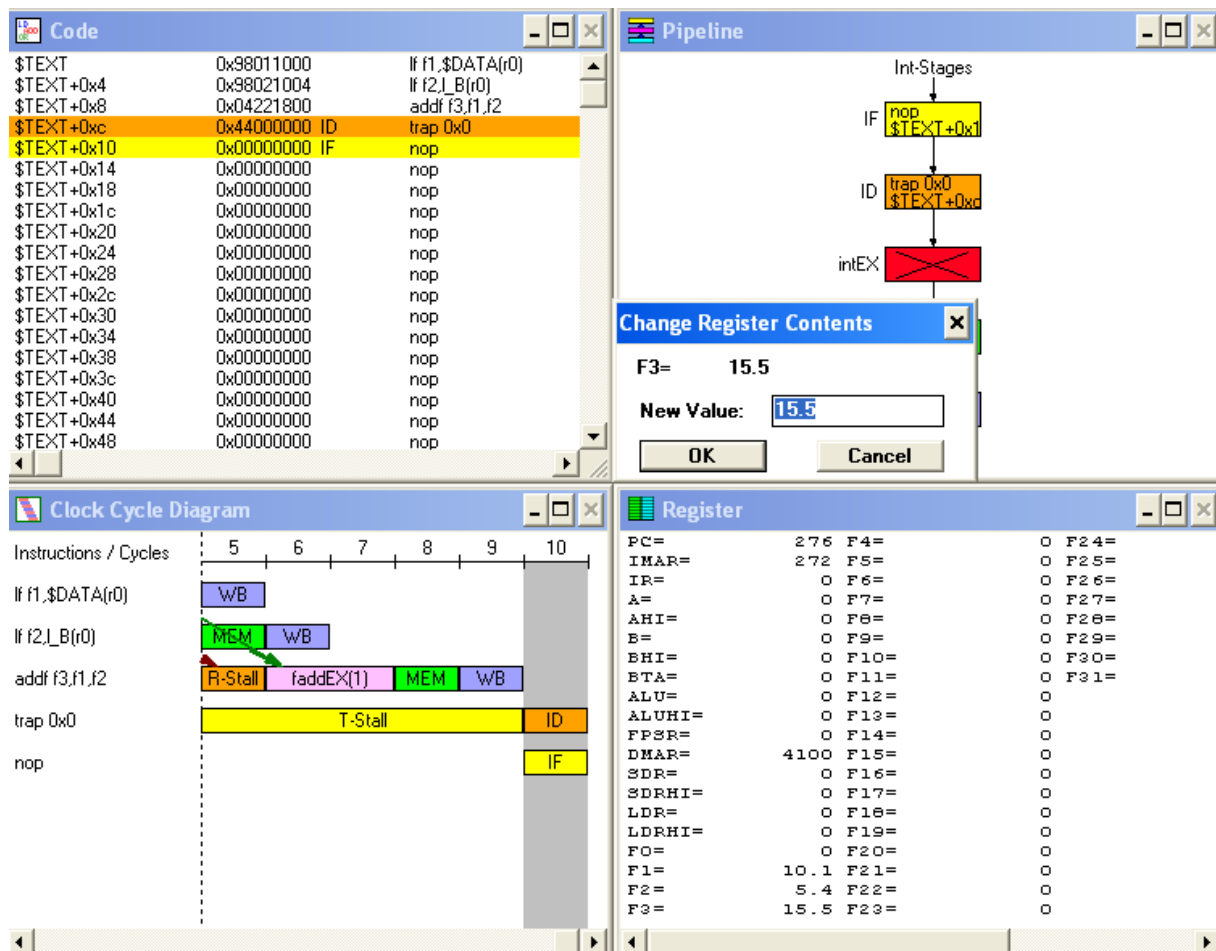
Po wykonaniu programu wartość rejestru F3 przyjęła wartość 15.5.

Zrzut ekranu, przedstawiający załadowany program i zawartość rejestrów i komórek PAO przed i po wykonaniu:

Przed:



Po:



## Zestaw 2

Kod programu:

.data

l\_A: .float 100000.0

l\_B: .float 0.4

.text

If f1, l\_A

If f2, l\_B

ADDF f3, f1, f2



trap 0

Dane wejściowe to etykieta l\_A przechowująca float 100000.0 oraz l\_B przechowująca 0.4.

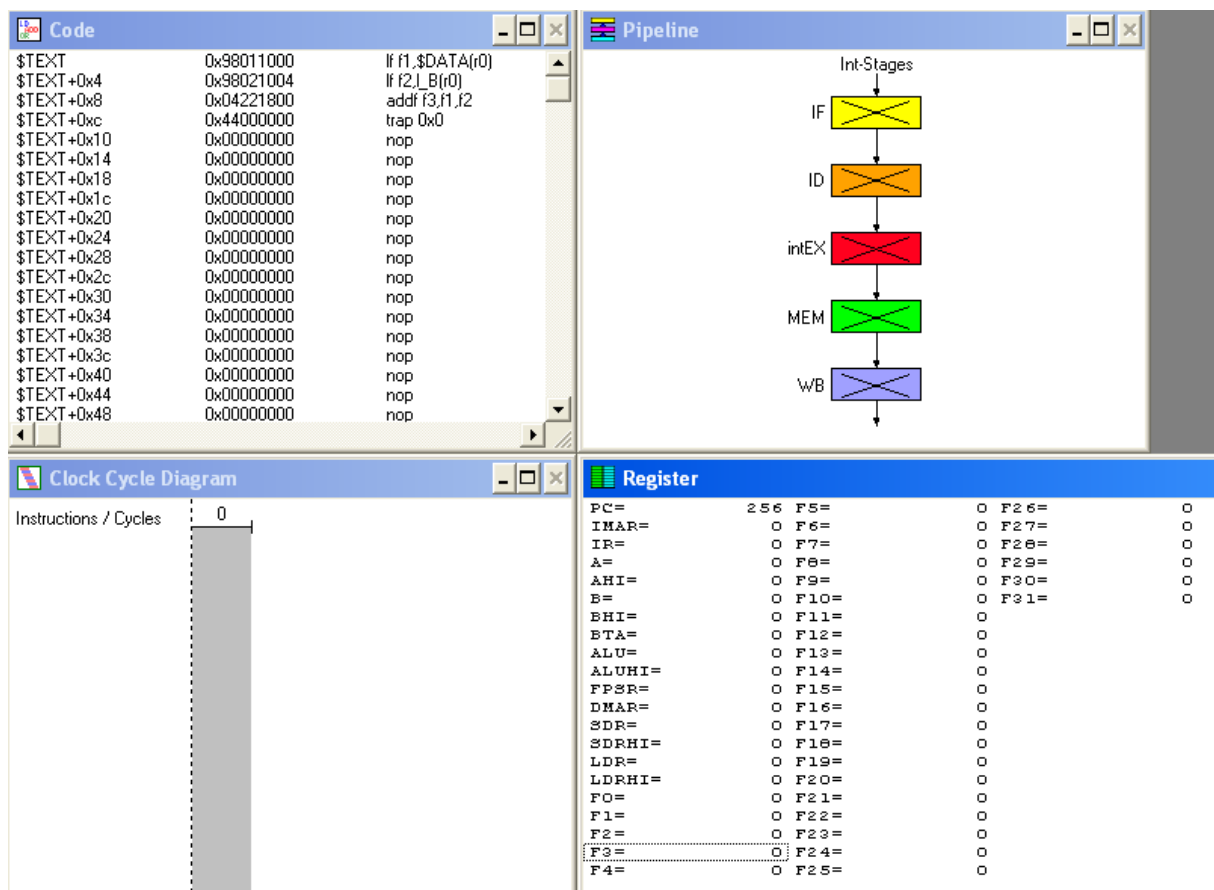
Spodziewany rezultat to wartość rejestru F3 będzie równa 100000.4.

Po wykonaniu programu wartość rejestru F3 przyjęła wartość 100000.0.

Wygląda na to, że program zaokrągliła wartości dziesiętne przy dużych liczbach.

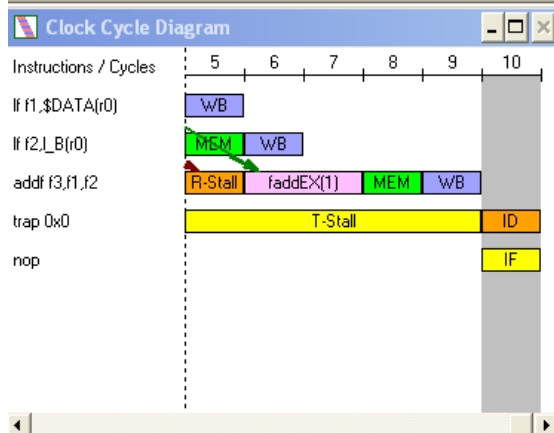
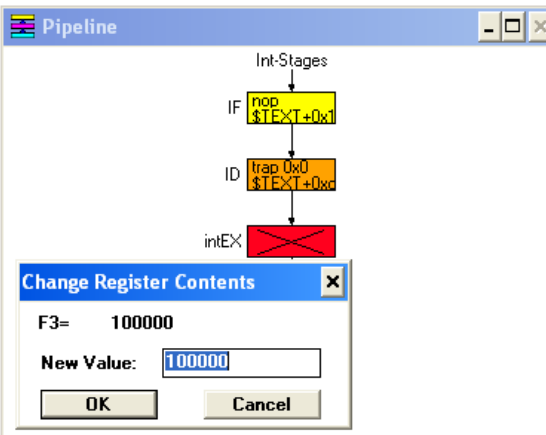
Zrzut ekranu, przedstawiający załadowany program i zawartość rejestrów i komórek PAO przed i po wykonaniu:

Przed:



Po:

Code		
\$TEXT	0x98011000	If f1,\$DATA(r0)
\$TEXT+0x4	0x98021004	If f2,_B(r0)
\$TEXT+0x8	0x04221800	addf f3,f1,f2
\$TEXT+0xc	0x44000000	ID trap 0x0
\$TEXT+0x10	0x00000000	IF nop
\$TEXT+0x14	0x00000000	nop
\$TEXT+0x18	0x00000000	nop
\$TEXT+0x1c	0x00000000	nop
\$TEXT+0x20	0x00000000	nop
\$TEXT+0x24	0x00000000	nop
\$TEXT+0x28	0x00000000	nop
\$TEXT+0x2c	0x00000000	nop
\$TEXT+0x30	0x00000000	nop
\$TEXT+0x34	0x00000000	nop
\$TEXT+0x38	0x00000000	nop
\$TEXT+0x3c	0x00000000	nop
\$TEXT+0x40	0x00000000	nop
\$TEXT+0x44	0x00000000	nop
\$TEXT+0x48	0x00000000	nop



Register			
PC=	276	F5=	0
IMAR=	272	F6=	0
IR=	0	F7=	0
A=	0	F8=	0
AHI=	0	F9=	0
B=	0	F10=	0
BHI=	0	F11=	0
BTA=	0	F12=	0
ALU=	0	F13=	0
ALUHI=	0	F14=	0
FP3R=	0	F15=	0
DMAR=	4100	F16=	0
SDR=	0	F17=	0
SDRHI=	0	F18=	0
LDR=	0	F19=	0
LDRHI=	0	F20=	0
FO=	0	F21=	0
F1=	100000	F22=	0
F2=	0.4	F23=	0
F3=	100000	F24=	0
F4=	0	F25=	0