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:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, wyświetlacz

Opis wygenerowany automatycznie

Po:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, wyświetlacz

Opis wygenerowany automatycznie

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:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, wyświetlacz

Opis wygenerowany automatycznie

Po:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Ikona komputerowa

Opis wygenerowany automatycznie

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:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, wyświetlacz

Opis wygenerowany automatycznie

Po:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, wyświetlacz

Opis wygenerowany automatycznie

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:

Obraz zawierający tekst, elektronika, zrzut ekranu, oprogramowanie

Opis wygenerowany automatycznie

Po:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Ikona komputerowa

Opis wygenerowany automatycznie

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:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, wyświetlacz

Opis wygenerowany automatycznie

Po:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Ikona komputerowa

Opis wygenerowany automatycznie

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:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, wyświetlacz

Opis wygenerowany automatycznie

Po:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, wyświetlacz

Opis wygenerowany automatycznie

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:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, numer

Opis wygenerowany automatycznie

Po:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, wyświetlacz

Opis wygenerowany automatycznie

Zestaw 2

Kod programu:

.data

l\_A: .float 100000.0

l\_B: .float 0.4

.text

lf f1, l\_A

lf 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 zaokrągla wartości dziesiętne przy dużych liczbach.

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

Przed:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Ikona komputerowa

Opis wygenerowany automatycznie

Po:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Ikona komputerowa

Opis wygenerowany automatycznie