Wojskowa Akademia Techniczna
im. Jarosława Dąbrowskiego

Laboratorium
Architektury i Organizacji komputerów

Prowadzący mgr inż. Artur Miktus
Sprawozdanie z ćwiczenia laboratoryjnego
nr 5

Temat ćwiczenia:
Realizacja operacji arytmetycznych w komputerze DLX

Wykonał: Arkadiusz Ostrzyżek
Grupa: WCY22KY2S1
Data wykonania ćwiczenia: 2024-01-12

Lab6 (11-12) KY2 prawdziwe

Pierwszym parametrem w tym zadaniu będzie **k = reszta z dzielenia całkowitego ostatniej cyfry numeru albumu autorki/ autora sprawozdania przez 4**. Oczywiście wartość tej reszty, czyli wynik operacji (ostatnia_cyfra mod 4) przyjmować będzie wartości ze zbioru [0,1,2,3].

Drugim parametrem będzie **nr = numer_w_dzienniku autorki/ autora sprawozdania (numer na liście grupy w USOS).**

Napisać program w asemblerze komputera WinDLX, który

1. Zadeklaruje statycznie (jak w przykładzie z mojej strony Lab5 dla zmiennej „tablica_B") **rozmiar = (10 + k)** elementowy wektor liczb całkowitych o nazwie **wektor**, pierwszy element o wartości równej **1000+nr**, każdy następny o **(k+10)** większy.
Na przykład dla osoby o numerze albumu kończącym się na 7 i numerze w dzienniku równym 5 będą to odpowiednio (k=3, nr = 5, liczba elementów wektora = 10 + k = 13): 1005, 1018, 1031, ..., 1248, 1261.
2. W pętli **policzy sumę elementów wektora dla jego wartości początkowych** do rejestru Rnr (dla osoby o numerze **5** do R**5**, dla osoby o numerze **10** do R**10** itd.) a następnie zapisze zawartość Rnr do zmiennej **suma1**.
3. Zadeklaruje stałą **stala**, równą iloczynowi (k+1) i nr, np. dla powyższego przykładu
stala = (3+1) * 5 = 4 * 5 = 20.
4. W pętli **zwiększy zawartość każdego elementu wektora o stałą stala i zapisze w miejscu dotychczasowego elementu.** Na przykład dla powyższych danych nowe zawartości wektora byłyby równe odpowiednio 1025, 1038, 1051, ..., 1268, 1281.
5. W pętli policzy **sumę elementów wektora dla jego wartości po modyfikacji** do rejestru Rnr a następnie zapisze zawartość Rnr do zmiennej **suma2**.
6. W rejestrze Rnr obliczy (**różnicę = suma2 - suma1**) i wynik zapisze do zmiennej **roznica.**
7. W rejestrze Rnr obliczy **iloczyn = rozmiar x stala** i wynik zapisze do zmiennej **iloczyn.**

W sprawozdaniu:

A. Zamieścić treść zadania z mojej strony. Jawnie podać wartości **k** i **nr**, wyniki obliczeń wartości **wektora (przed i po modyfikacji)** i wyników obliczeń: **suma1**, **suma2**, **roznica**, **iloczyn** – uzyskane dla obliczeń pisemnych.
B. Zamieścić listing napisanego przez siebie programu w postaci tekstowej, możliwej do „skopiowania" w przeglądarce typu Adobe Reader – nie zamieszczać tekstu programu w postaci obrazka. Muszę mieć możliwość skopiowania Waszego programu do mojej maszyny wirtualnej i sprawdzenia poprawności działania.
C. Zamieścić zrzutu ekranu z WinDLX z uzyskanymi wynikami, prezentowanymi jako liczby dziesiętne (prezentacja liczb heksadecymalnych nie bedzie uznana za realizacje punktu zadania)w tym
a. na jednym z obrazków ze stanem początkowym wektora i wyzerowanymi zmiennymi wynikowymi (okienko Memory/ Display, odpowiednio skonfigurowane);
b. na drugim z obrazków ze stanem wektora i wynikami obliczeń **suma1**, **suma2**, **roznica**, **iloczyn** – po zakończeniu wykonywania programu (okienko Memory/ Display, odpowiednio skonfigurowane);
c. Na trzecim z obrazków stan zmaksymalizowanego okienka Menu/ Window/ **Statistics**.
D. Zamieścić algorytm swojego programu w postaci graficznej i krótko ten algorytm opisać, ze szczególnym uwzględnieniem warunków wyjścia z każdej pętli.
E. W zależności od swojej wartości **k** wybrać **jedną** z instrukcji swojego programu, odpowiednio
a. k=0 rozkaz typu load;
b. k=1 rozkaz typu branch;
c. k=2 rozkaz typu arithmetic immediate;
d. k=3 rozkaz typu store
i opisać zmiany w rejestrach **R** i tymczasowych (**A, B, Imm** itp.) w trakcie kompletnego wykonania tego rozkazu przez poszczególne etapy komputera WinDLX, podobnie do mojego opisu na http://www.ita.wat.edu.pl/~a.miktus/AOK/Lab6/Etapy_potoku_DLX.html .
e. Opis ma być uzupełniony zrzutami ekranu z WinDLX, pokazującymi opisywane zmiany dla tej jednej, wybranej instrukcji.

Stopień trudności zadania:
1. Na ocenę **dst** punkty **1** – **2** zadania i **A** – **D** sprawozdania.
2. Na ocenę **db** punkty **1** – **5** zadania i **A** – **D** sprawozdania.
3. Na ocenę **bdb** punkty **1** – **7** zadania i **A** – **E** sprawozdania (czyli wszystko).

W przypadku stwierdzenia niesamodzielnej pracy = nieuczciwości studentów osoby oszukujące (dawca i biorcy) za zadanie otrzymują ocenę **zero do średniej**. To samo w przypadku niewykonania zadania i nieprzysłania sprawozdania w terminie.

A. Wyliczenia wartości

Wartość k: 83744 / 4 = 20936 reszta 0
Wartość nr: 7

Wartości wektora przed:
1007, 1024, 1041, 1058, 1075, 1092,1109, 1126, 1143, 1160

Wartości wektora po:
1014, 1031, 1048, 1065, 1082, 1099, 1116, 1133, 1150, 1167

suma1: 10635
suma2: 10905
roznica: 70
iloczyn: 70

B. kod programu

```
.data
stala:          .word 7
rozmiar:        .word 10
tablica_B:      .word 1007, 1024, 1041, 1058, 1075, 1092,1109, 1126, 1143, 1160
suma1:          .word 0
suma2:          .word 0
roznica:        .word 0
iloczyn:        .word 0

.text
lw r1, stala
addi r10, r0, tablica_B
```

```
lw r12, rozmiar
addi r13, r0, suma1
addi r14, r0, suma2
addi r15, r0, roznica
addi r16, r0, iloczyn

p1:
lw r2, 0(r10)
add r7, r7, r2
add r11, r1, r2
sw 0(r10), r11

addi r10, r10, #4
subi r12, r12, #1
bnez r12, p1

sw suma1, r7

addi r10, r0, tablica_B
lw r12, rozmiar
addi r7, r0, #0

p2:
lw r2, 0(r10)
add r7, r7, r2

addi r10, r10, #4
subi r12, r12, #1
bnez r12, p2

sw suma2, r7

lw r26, suma1
lw r27, suma2
sub r28, r27, r26
sw roznica, r28

lw r29, stala
lw r30, rozmiar
mult r31, r29, r30
sw iloczyn, r31

trap 0
```

C. Zrzuty pamięci

a. Pamięć po wczytaniu

```
zmienna                 0
stala                   7
rozmiar                10
wektor               1007
wektor+0x4           1024
wektor+0x8           1041
wektor+0xc           1058
0x0000101c           1075
0x00001020           1092
0x00001024           1109
0x00001028           1126
0x0000102c           1143
0x00001030           1160
suma1                   0
suma2                   0
roznica                 0
iloczyn                 0
```

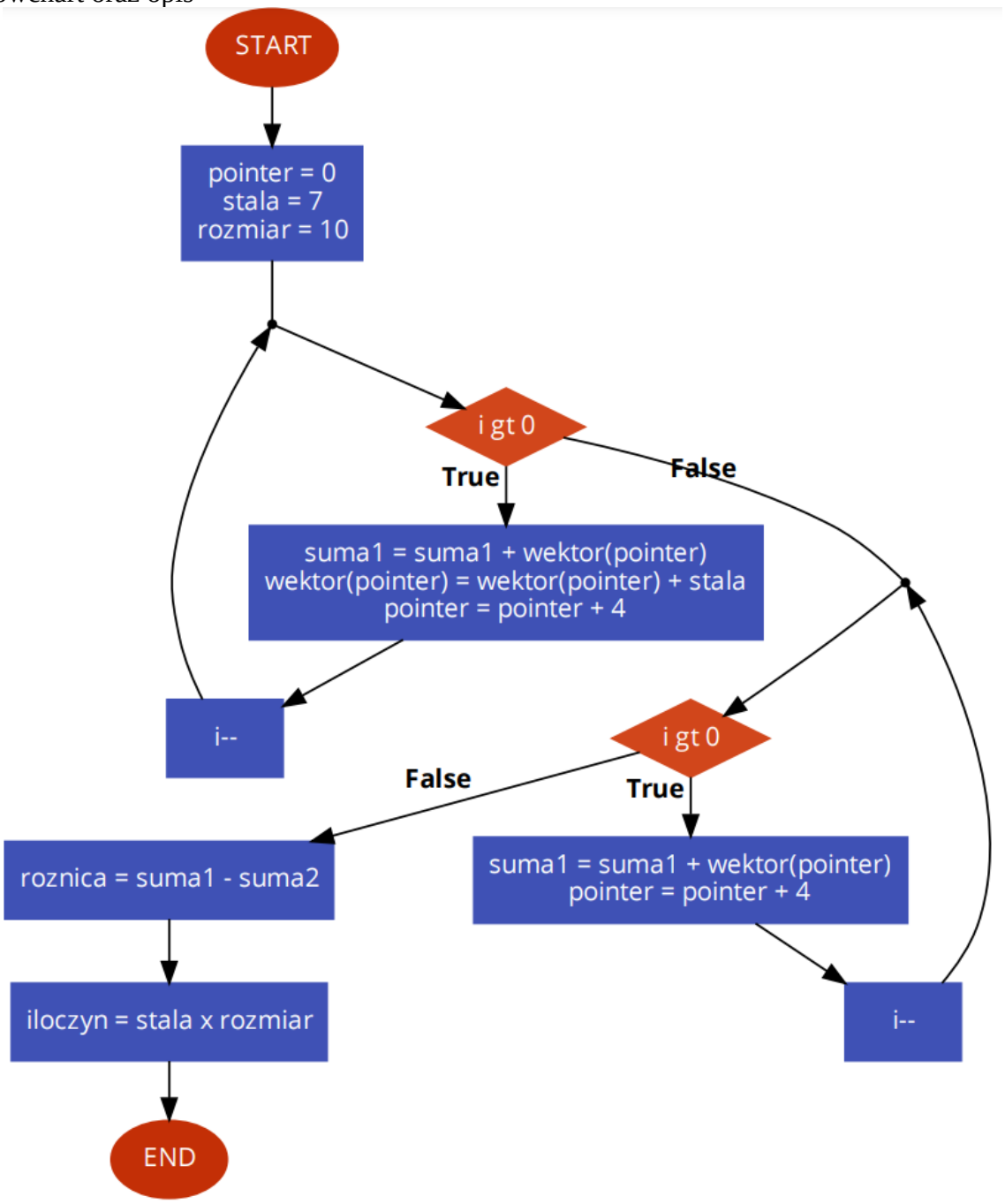
b. Pamięć po uruchomieniu

```
zmienna                 0
stala                   7
rozmiar                10
wektor               1014
wektor+0x4           1031
wektor+0x8           1048
wektor+0xc           1065
0x0000101c           1082
0x00001020           1099
0x00001024           1116
0x00001028           1133
0x0000102c           1150
0x00001030           1167
suma1               10835
suma2               10905
roznica                70
iloczyn                70
```

c. Statistics

```
Total:
    209 Cycle(s) executed.
    ID executed by 141 Instruction(s).
    2 Instruction(s) currently in Pipeline.

Hardware configuration:
    Memory size: 32768 Bytes
    faddEX-Stages: 1, required Cycles: 2
    fmulEX-Stages: 1, required Cycles: 5
    fdivEX-Stages: 1, required Cycles: 19
    Forwarding enabled.

Stalls:
    RAW stalls: 46 (22.01% of all Cycles), thereof:
        LD stalls: 22 (47.83% of RAW stalls)
        Branch/Jump stalls: 20 (43.48% of RAW stalls)
        Floating point stalls: 4 (8.70% of RAW stalls)
    WAW stalls: 0 (0.00% of all Cycles)
    Structural stalls: 0 (0.00% of all Cycles)
    Control stalls: 18 (8.61% of all Cycles)
    Trap stalls: 7 (3.35% of all Cycles)
    Total: 71 Stall(s) (33.97% of all Cycles)

Conditional Branches):
    Total: 20 (14.18% of all Instructions), thereof:
        taken: 18 (90.00% of all cond. Branches)
        not taken: 2 (10.00% of all cond. Branches)

Load-/Store-Instructions:
    Total: 41 (29.08% of all Instructions), thereof:
        Loads: 27 (65.85% of Load-/Store-Instructions)
        Stores: 14 (34.15% of Load-/Store-Instructions)

Floating point stage instructions:
    Total: 1 (0.71% of all Instructions), thereof:
        Additions: 0 (0.00% of Floating point stage inst.)
        Multiplications: 1 (100.00% of Floating point stage inst.)
        Divisions: 0 (0.00% of Floating point stage inst.)

Traps:
    Traps: 1 (0.71% of all Instructions)
```

D. Flowchart oraz opis



Program wykonuje dwie pętle. Przy pierwszej pętli sumuje liczby, a następnie zwiększa ich wartości o stałą. Pętla druga tylko sumuje liczby. Na koniec wylicza różnicę sum oraz iloczyn stałej i rozmiaru.

E. Opis Load

Load word powoduje zapisanie danego słowa w miejscu w pamięci przez nas wyznaczonym.