

Wojskowa Akademia Techniczna
im. Jarosława Dąbrowskiego

Laboratorium
Architektury i Organizacji komputerów

Prowadzący mgr inż. Artur Miktus
Sprawozdanie z ćwiczenia laboratoryjnego
nr 8

Temat ćwiczenia:

Hazardy sterowania w przetwarzaniu potokowym

Wykonał: Arkadiusz Ostrzyżek

Grupa: WCY22KY2S1

Data wykonania ćwiczenia: 2024-01-26

Zadanie wykonano na ocenę: 4.

Treść zadania:

Begin

Dane:

SKŁADNIK = 1770, UŁAMEK = 0.77, ROZMIAR = 107.

Wzór:

$$TB[i] = [7.7 * (T[i] + T[i+3] + T[i+5]) * (T[i+7] + T[i+9])] / (T[i+2] - T[i])$$

Napisać program **Lab8_nr.s** w assemblerze komputera DLX, który:

1. Zadeklaruje dwie tablice przechowujące liczby zmiennoprzecinkowe podwójnej precyzji: **T** 130- elementową oraz **TB** ROZMIAR-elementową, a także zmienną **Suma** zmiennoprzecinkową podwójnej precyzji.

2. Komórki tablicy **T** wypełni (za pomocą obliczeń, wykonanych w pętli, a nie za pomocą statycznej deklaracji z nadaniem wartości początkowych) kolejnymi liczbami o części ułamkowej równej UŁAMEK i części całkowitej rosnącej o jeden, począwszy od numeru w dzienniku studenta/ studentki, powiększonej o SKŁADNIK

(np. nr=1; UŁAMEK = 0.35; SKŁADNIK = 5; w tablicy T mają być zapisane liczby $T[1] = (1+5+0.35) = 6.35$, $T[2] = (6.35 + 1) = 7.35$ itd.).

3. Następnie dla każdego elementu tablicy **TB** wykona operację, określoną powyższym wzorem (**UWAGA: wszystkie występujące we wzorze działania mają być jawnie wykonane w programie**, nie są dopuszczalne przekształcenia wzoru (np. skrócenia), zastępowanie wykonywania działań obliczonymi stałymi. Można użyć stałych dla reprezentowania w programie wartości numeru w dzienniku, danych SKŁADNIK i UŁAMEK oraz stałych we wzorach na TB np. 1.2, 7.7 itd.

4. W zmiennej **Suma** umieści obliczoną w pętli sumę wszystkich elementów tablicy **TB**. **Uwaga** - ze względu na błąd w implementacji forwardingu ZMP w WinDLX czasem zdarza się tak, że poprawnie napisany program przy wyłączonym forwardingu "daje" poprawne wyniki, a po włączeniu forwardingu generuje złe zawartości TB albo błędną Sumę. Radzę w przypadku "niezrozumiałych" błędów wyłączyć forwarding i sprawdzić działanie programu. Szczegóły wspomnianego błędu można poznać [tutaj](#).

5. Przed rozpoczęciem tworzenia programu radzę (o ile Studentka/Student - wykonawca ćwiczenia walczy o ocenę co najmniej **db**) zaprojektować arkusz kalkulacyjny w Excelu, Calcu lub innym środowisku, wykonujący te same obliczenia w celu weryfikacji poprawności uzyskiwanych w programie wyników.

6. W treści pierwszego i ostatniego elementu TB dla danych konkretnej osoby. Obliczenia te zostaną następnie zweryfikowane na zrzucie ekranu z uruchomienia.

End

KOD:

```
.data
tablica_T:    .space 1040
tablica_TB:   .space 856
skladnik:     .double 1770.0
ulamek:       .double 0.77
nr:           .double 7.0
wartosc:      .double 7.7
jeden:        .double 1.0
suma:         .double 0
```

```
.text
ld      f2, skladnik
ld      f4, ulamek
ld      f6, nr
ld      f8, jeden
```

```
addd    f10, f10, f2
addd    f10, f10, f4
addd    f10, f10, f6
addi    r1, r0, tablica_T
addi    r2, r0, #130
```

```
fill_T:
sd      0(r1), f10
addd    f10, f10, f8
addi    r1, r1, #8
subi    r2, r2, #1
bnez    r2, fill_T
```

```
; dane dla fill_TB
addi    r1, r0, tablica_T
addi    r2, r0, tablica_TB
addi    r3, r0, #107
```

```
addd    f10, f0, f0
```

```
fill_TB:
```

```

; dodawanie1
ld f12, 0(r1)
ld f14, 24(r1)
ld f16, 40(r1)

addd f12, f12, f14 ; dodawanie1 w f12
addd f12, f12, f16

; dodawanie2
ld f18, 56(r1)
ld f20, 72(r1)

addd f18, f18, f20 ; dodawanie2 w f18

; mnozenie
ld f22, wartosc
multd f24, f22, f12
multd f24, f24, f18 ; suma mozenia f24

; odejmowanie
ld f26, 16(r1)
ld f28, 0(r1)

subd f26, f26, f28 ; odejmowanie w f26

; dzielenie
divd f30, f24, f26 ; dzielenie w f30

; suma
addd f10, f10, f30

sd 0(r2), f30
addi r1, r1, #8
addi r2, r2, #8
subi r3, r3, #1
bnez r3, fill_TB

sd suma, f10
trap 0

```

XP_DLX [Running] - Oracle VM VirtualBox

FileMachineViewInputDevicesHelp

WINDLX

FileWindowExecuteMemoryConfigurationMemory DisplayHelp

Memory-2

suma0.33512e+090

Memory-1

tablica_TB7.34453e+074.1131e+200

Memory-3

0x000017606.24371e+070

Statistics

Total:

8258 Cycle(s) executed.
ID executed by 2912 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 disabled.

Stalls:

RAW stalls: 4999 (60.54% of all Cycles)
WAW stalls: 0 (0.00% of all Cycles)
Structural stalls: 0 (0.00% of all Cycles)
Control stalls: 235 (2.84% of all Cycles)
Trap stalls: 3 (0.04% of all Cycles)
Total: 5237 Stall(s) (63.42% of all Cycles)

Conditional Branches):

Total: 237 (8.14% of all Instructions), thereof:
taken: 235 (99.16% of all cond. Branches)
not taken: 2 (0.84% of all cond. Branches)

Load-/Store-Instructions:

Total: 1098 (37.71% of all Instructions), thereof:
Loads: 860 (78.32% of Load-/Store-Instructions)
Stores: 238 (21.68% of Load-/Store-Instructions)

Floating point stage instructions:

Total: 990 (34.00% of all Instructions), thereof:
Additions: 669 (67.58% of Floating point stage inst.)
Multiplications: 214 (21.62% of Floating point stage inst.)
Divisions: 107 (10.81% of Floating point stage inst.)

Traps:

Traps: 1 (0.03% of all Instructions)

Clock Cy...PipelineCodeRegisterBreakpoints

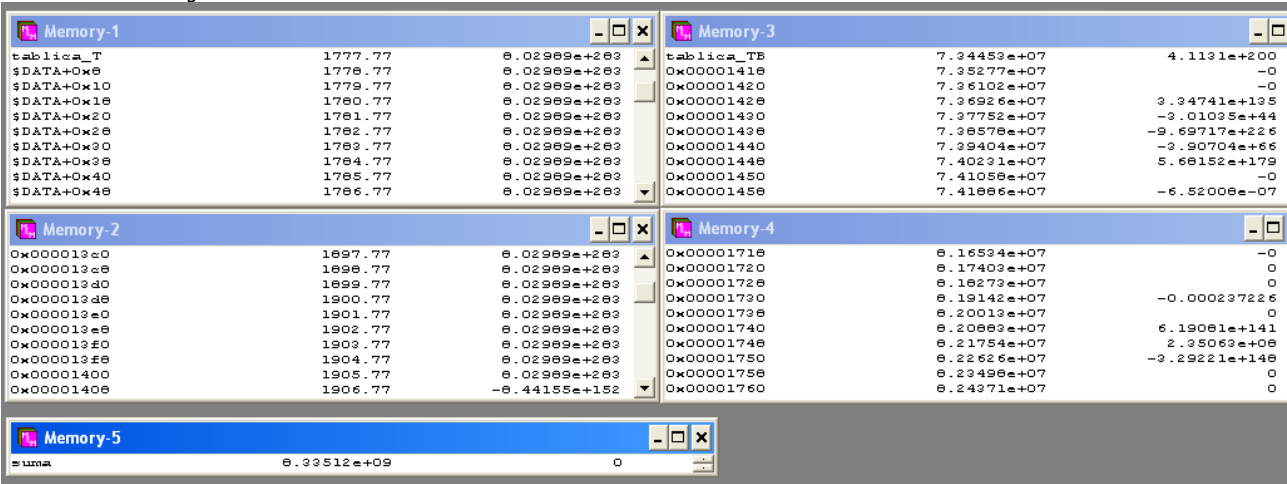
StartDLXa4.s - NotatnikWINDLX

PL10:55

Right Ctrl

10:5525/01/2024

Nowe zrzuty ekranu:



T:

TB:

Suma:

1777.77	73,445,303.92
1778.77	73,527,706.40
1779.77	73,610,155.07
1780.77	73,692,649.94
1781.77	73,775,191.02
1782.77	73,857,778.29
1783.77	73,940,411.77
1784.77	74,023,091.44
1785.77	74,105,817.31
1786.77	74,188,589.39
1897.77	81,653,451.10
1898.77	81,740,334.97
1899.77	81,827,265.05
1900.77	81,914,241.32
1901.77	82,001,263.80
1902.77	82,088,332.47
1903.77	82,175,447.34
1904.77	82,262,608.42
1905.77	82,349,815.69
1906.77	82,437,069.17

Suma	8335121957
------	------------