

# Architektura i organizacja komputerów

Sprawozdanie z laboratorium nr 7

Temat zajęć: Hazardy danych w  
przetwarzaniu potokowym

Borkowski Kamil WCY22IY1S1

Data wykonania: 2024.01.11

**Treść zadania:**

Begin

Napisać program L7\_nr w asemblerze komputera WinDLX, który

1. Zadeklaruje dwie tablice przechowujące liczby całkowite:

TA 120- elementową oraz

TB 110-elementową

a także zmienną zmiennoprzecinkową podwójnej precyzji .double o nazwie Suma i nada jej wartość początkową zero.

2. Komórki tablicy TA wypełni rosnąco kolejnymi liczbami całkowitymi począwszy od  $(100 + nr)$  – dla  $nr = 5$  będą to odpowiednio 105, 106, 107 itd.

3. Następnie dla każdego elementu tablicy TB wykona operację:

$$TB[i] = (TA[i+4] + TA[i+9]) * 32$$

4. W pętli obliczy do rejestru Rnr (osoba o  $nr = 5$  do R5) sumę elementów tablicy TB.

5. Zawartość Rnr zapisze po niezbędnej konwersji do zmiennej Suma.

6. W treści programu asemblerowego .s, na koniec programu, w komentarzu (symbol średnika ; na początku linii) proszę pisemnie obliczyć na podstawie niezbędnych (ale łatwo wyznaczalnych) elementów TA i wzoru na TB, jaka będzie wartość pierwszego i ostatniego elementu TB dla danych konkretnej osoby. Obliczenia te zostaną następnie zweryfikowane na zrzucie ekranu z uruchomienia.

End

**Treść napisanego programu:**

;nr 7

.data

TA: .space 480

TB: .space 440

Suma: .double 0

.text

addi r7, r0, #0 ; suma

addi r1, r0, #107 ; pierwszy element TA

addi r20, r0, TA ; i dla TA

sw 0(r20), r1 ;przypisanie pierwszego elementu

addi r30, r0, #119 ; i dla petli1

petla1:

lw r2, 0(r20)

addi r15, r2, #1

subi r30, r30, #1

addi r20, r20, #4

sw 0(r20), r15

bnez r30, petla1

addi r30, r0, #110 ; nowe i dla petli2

addi r20, r0, TA ; i dla TA

addi r21, r0, TB ; i dla TB

petla2:

addi r22, r20, #16 ; i+4

lw r22, 0(r22) ; TA[i+4]

addi r23, r20, #36 ; i+9

lw r23, 0(r23) ; TA[i+9]

add r24, r22, r23 ;TA[i+4] +TA[i+9]

addi r25, r0, #32

multu r25, r25, r24

sw 0(r21), r25

addi r20, r20, #4

addi r21, r21, #4

subi r30, r30, #1

bnez r30,petla2

addi r30, r0, #110 ; nowe i dla petli2

addi r21, r0, TB ; i dla TB

petla3:

lw r2, 0(r21)

add r7, r7, r2

addi r21, r21, #4

subi r30, r30, #1

bnez r30, petla3

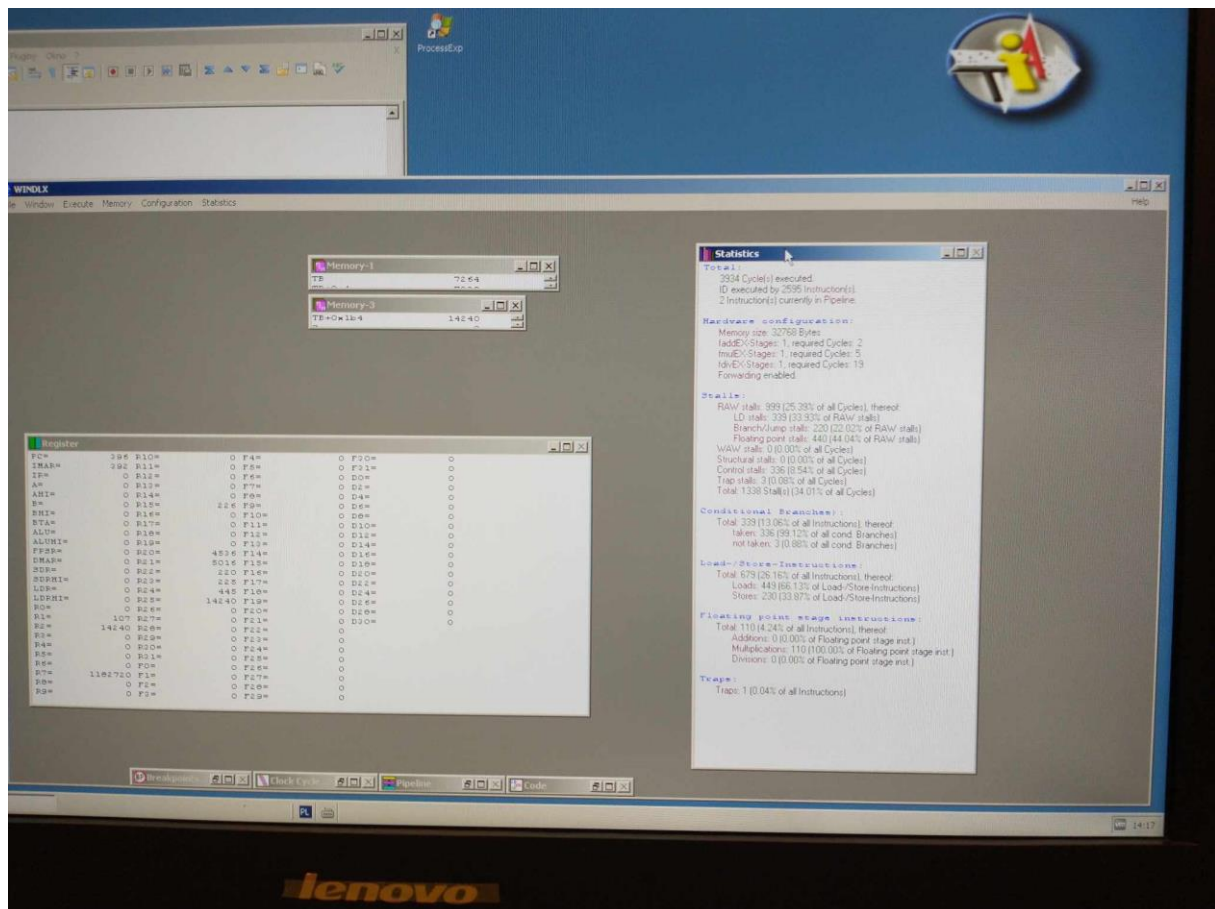
;movi2fp f7, r7

;cvti2d f10, f7

;sd Suma, f10

trap 0

Zdjęcie ekranu z wynikami:

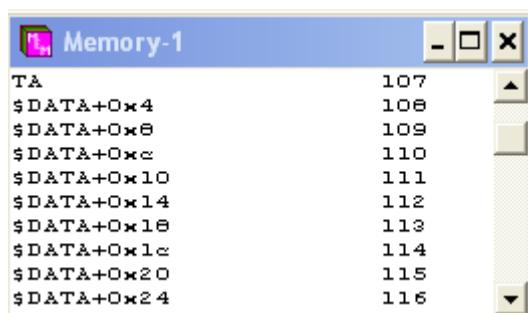


Nowe okienka, nieobecne na zrzutach ekranu, przesłanych na koniec zajęć:

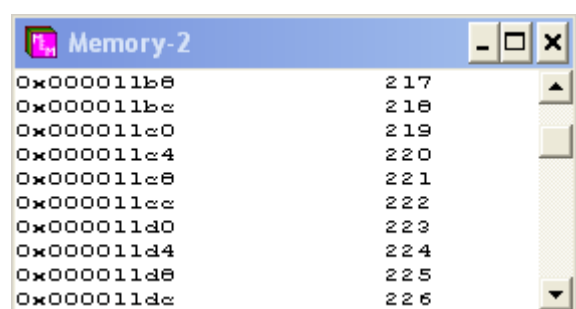
Zmienna Suma:



Pierwsze 10 elementów tablicy TA:



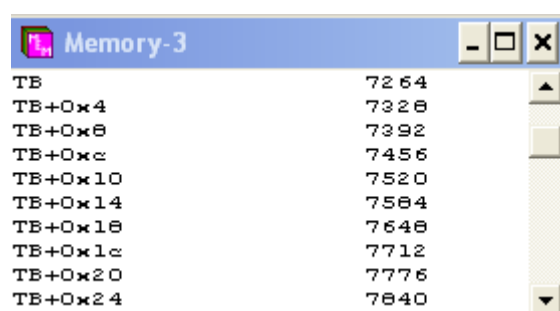
Ostatnie 10 elementów tablicy TA:



A screenshot of a debugger window titled "Memory-2". It displays a list of memory addresses and their corresponding values. The addresses are in hexadecimal, and the values are in decimal. The list shows the last 10 elements of array TA.

0x000011b8	217
0x000011bc	218
0x000011c0	219
0x000011c4	220
0x000011c8	221
0x000011cc	222
0x000011d0	223
0x000011d4	224
0x000011d8	225
0x000011dc	226

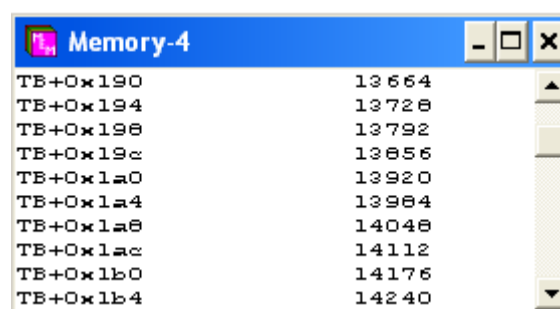
Pierwsze 10 elementów tablicy TB:



A screenshot of a debugger window titled "Memory-3". It displays a list of memory addresses and their corresponding values. The addresses are in hexadecimal, and the values are in decimal. The list shows the first 10 elements of array TB.

TB	7264
TB+0x4	7328
TB+0x8	7392
TB+0xc	7456
TB+0x10	7520
TB+0x14	7584
TB+0x18	7648
TB+0x1c	7712
TB+0x20	7776
TB+0x24	7840

Ostatnie 10 elementów tablicy TB:



A screenshot of a debugger window titled "Memory-4". It displays a list of memory addresses and their corresponding values. The addresses are in hexadecimal, and the values are in decimal. The list shows the last 10 elements of array TB.

TB+0x190	13664
TB+0x194	13728
TB+0x198	13792
TB+0x19c	13856
TB+0x1a0	13920
TB+0x1a4	13984
TB+0x1a8	14048
TB+0x1ac	14112
TB+0x1b0	14176
TB+0x1b4	14240