

Architektura i organizacja komputerów

Sprawozdanie z laboratorium nr 6

Temat zajęć: Realizacja operacji
arytmetycznych w komputerze
DLX

Borkowski Kamil WCY22IY1S1

Data wykonania: 2023.12.11

A.

Lab6 (11-12) Y1 prawdziwe

Pierwszym parametrem w tym zadaniu będzie k = reszta z dzielenia całkowitego ostatniej cyfry numeru albumu autorki/ autora sprawozdania przez 5. Oczywiście wartość tej reszty, czyli wynik operacji (ostatnia_cyfra mod 5) przyjmować będzie wartości ze zbioru $[0,1,2,3,4]$.

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

Wszystkie zamieszczane obrazki mają być czytelne = wyraźne i duże, pokazujące wybrane fragmenty ekranu z interfejsem WinDLX a nie cały ekran Windows.

Napisać program w assemblerze 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 $100+nr$, 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 ($k=2$, $nr = 5$, liczba elementów wektora = $10 + k = 12$): 105, 117, 129, ..., 225, 237.

2. W pętli policzy sumę elementów wektora dla jego wartości początkowych do rejestru Rnr (dla osoby o numerze 5 do R5, dla osoby o numerze 10 do R10 itd.) a następnie zapisze zawartość Rnr do zmiennej suma1.

Dla powyższych danych suma1 = 2052.

3. Zadeklaruje stałą stałą, równą iloczynowi $(k+1)$ i nr , np. dla powyższego przykładu

stała = $(2+1) * 5 = 3 * 5 = 15$.

4. W pętli zwiększy zawartość każdego elementu wektora o stałą stałą i zapisze w miejscu dotychczasowego elementu. Na przykład dla powyższych danych nowe zawartości wektora byłyby równe odpowiednio 120, 132, 144, ..., 240, 252.

5. W pętli policzy sumę elementów wektora dla jego wartości po modyfikacji do rejestru Rnr a następnie zapisze zawartość Rnr do zmiennej suma2. Dla powyższych danych suma2 = 2232.

6. W rejestrze Rnr obliczy (różnicę = suma2 - suma1) i wynik zapisze do zmiennej roznica. Dla powyższych danych roznica = 180.

7. W rejestrze Rnr obliczy iloczyn = rozmiar x stała i wynik zapisze do zmiennej iloczyn. Dla powyższych danych iloczyn = 180.

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 w sprawozdaniu (nie dla wyników programu). To ma być wzorzec, z którym porównujecie wyniki programu.

Macie szansę uniknąć "głupich błędów".

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 zrzuty ekranu z WinDLX z uzyskanymi wynikami, 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. $k=0,1$ rozkaz typu load albo branch (do wyboru);

b. $k=2$ rozkaz typu arithmetic immediate;

c. $k=3,4$ 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 .

d. 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).

11.12.2023, 15:36 Lab6 Y1

https://www.ita.wat.edu.pl/~a.miktus/AOK/Lab6/Zadanie_Lab6.html 2/2

W przypadku stwierdzenia niesamodzielnej pracy (działania dla nie swojej ostatniej cyfry numeru albumu albo nieswojego numeru w dzienniku)= 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.

W dniu przeprowadzenia zajęć nie trzeba przysyłać do mnie żadnych plików

k = 4; nr = 7;

wektor przed

wektor = 107, 121, 135, 149, 163, 177, 191, 205, 219, 233, 247, 261, 275, 289;

wektor po

wektor = 107+35, 121+35, 135+35, 149+35, 163+35, 177+35, 191+35, 205+35, 219+35, 233+35, 247+35, 261+35, 275+35, 289+35;

wektor = 142, 156, 170, 184, 198, 212, 226, 240, 254, 268, 282, 296, 310, 324;

suma1 = 107+121+135+149+163+177+191+205+219+233+247+261+275+289= 2772;

suma2 = 142+156+170+184+198+212+226+240+254+268+282+296+310+324=3262;

roznica = suma2 – suma1 = 3262 – 2772 = 490;

iloczyn = rozmiar * stala = 14 * 35 = 490;

B.

.data

;nr 7 k 4

rozmiar: .word 14

wektor: .word 107, 121, 135, 149, 163, 177, 191, 205, 219, 233, 247, 261, 275, 289

suma1: .word 0

stala: .word 35

suma2: .word 0

roznica: .word 0

iloczyn: .word 0

.text

addi r20, r0, wektor ; r20 dla elementow tablicy

addi r30, r0, #14 ; r30 dla petli

addi r7, r0, #0 ; r7 dla sumy

start1:

lw r2, 0(r20)

add r7, r7, r2 ; zwieksz sume

```
subi r30, r30, #1 ; i--  
addi r20, r20, #4 ; kolejny element tab  
bnez r30, start1
```

```
sw suma1, r7  
;AD. 4  
lw r1, stala  
addi r20, r0, wektor  
addi r30, r0, #14 ; r30 dla petli  
addi r7, r0, #0 ; r7 dla sumy
```

```
start2:  
lw r2, 0(r20)  
add r10, r1, r2 ; r10 dla nowego wektora  
sw 0(r20), r10  
add r7, r7, r10  
subi r30, r30, #1  
addi r20, r20, #4  
bnez r30, start2  
sw suma2, r7
```

```
;AD 6  
lw r1, suma1  
lw r2, suma2  
sub r7, r2, r1  
sw roznica, r7
```

```
;AD 7  
lw r1, rozmiar  
lw r2, stala  
mult r7, r1, r2
```

sw iloczyn, r7

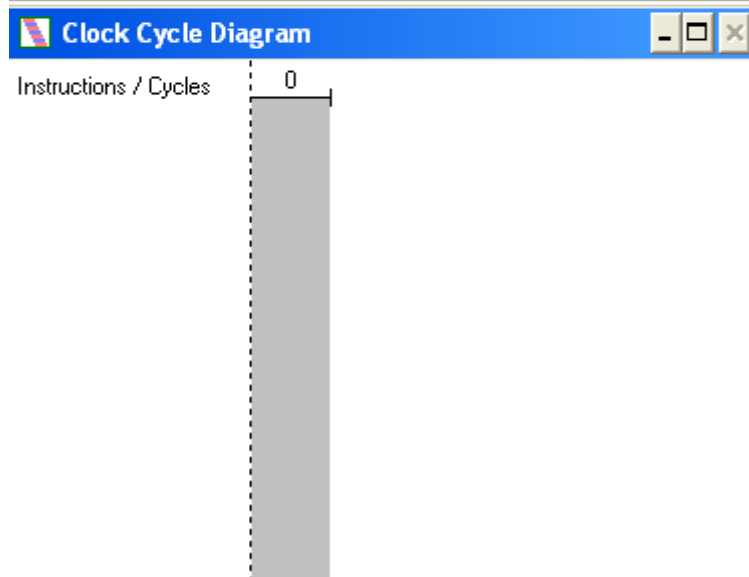
trap 0

C.

Przed:

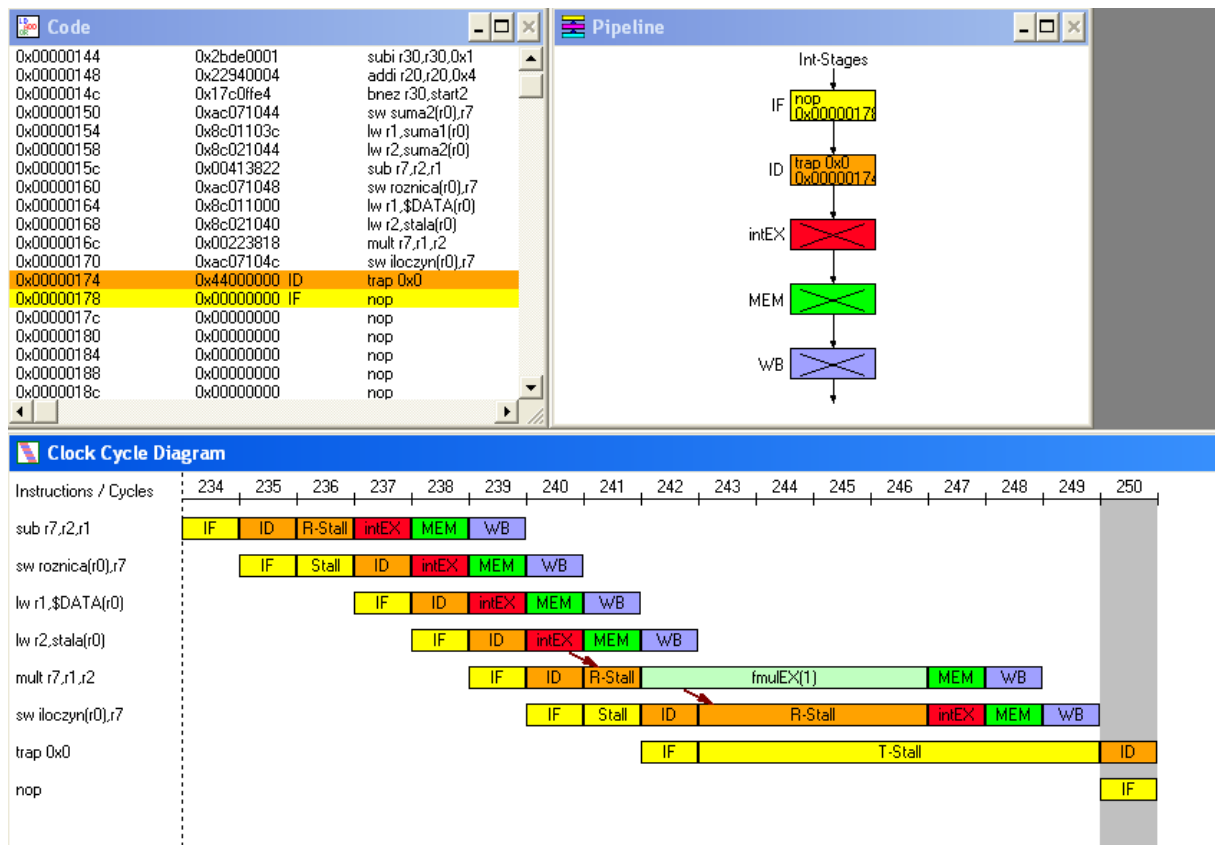
Memory-1		Register			
0x00000ffe	0	PC=	256	R4=	0 R24=
\$DATA	14	IMAR=	0	R5=	0 R25=
wektor	107	IR=	0	R6=	0 R26=
wektor+0x4	121	A=	0	R7=	0 R27=
wektor+0x8	135	AHI=	0	R8=	0 R28=
wektor+0xc	149	B=	0	R9=	0 R29=
0x00001014	163	BHI=	0	R10=	0 R30=
0x00001018	177	BTA=	0	R11=	0 R31=
0x0000101c	191	ALU=	0	R12=	0
0x00001020	205	ALUHI=	0	R13=	0
0x00001024	219	FP3R=	0	R14=	0
0x00001028	233	DMAR=	0	R15=	0
0x0000102c	247	SDR=	0	R16=	0
0x00001030	261	SDRHI=	0	R17=	0
0x00001034	275	LDR=	0	R18=	0
0x00001038	289	LDRHI=	0	R19=	0
suma1	0	RO=	0	R20=	0
stala	35	R1=	0	R21=	0
suma2	0	R2=	0	R22=	0
roznica	0	R3=	0	R23=	0
iloczyn	0				
0x00001050	0				

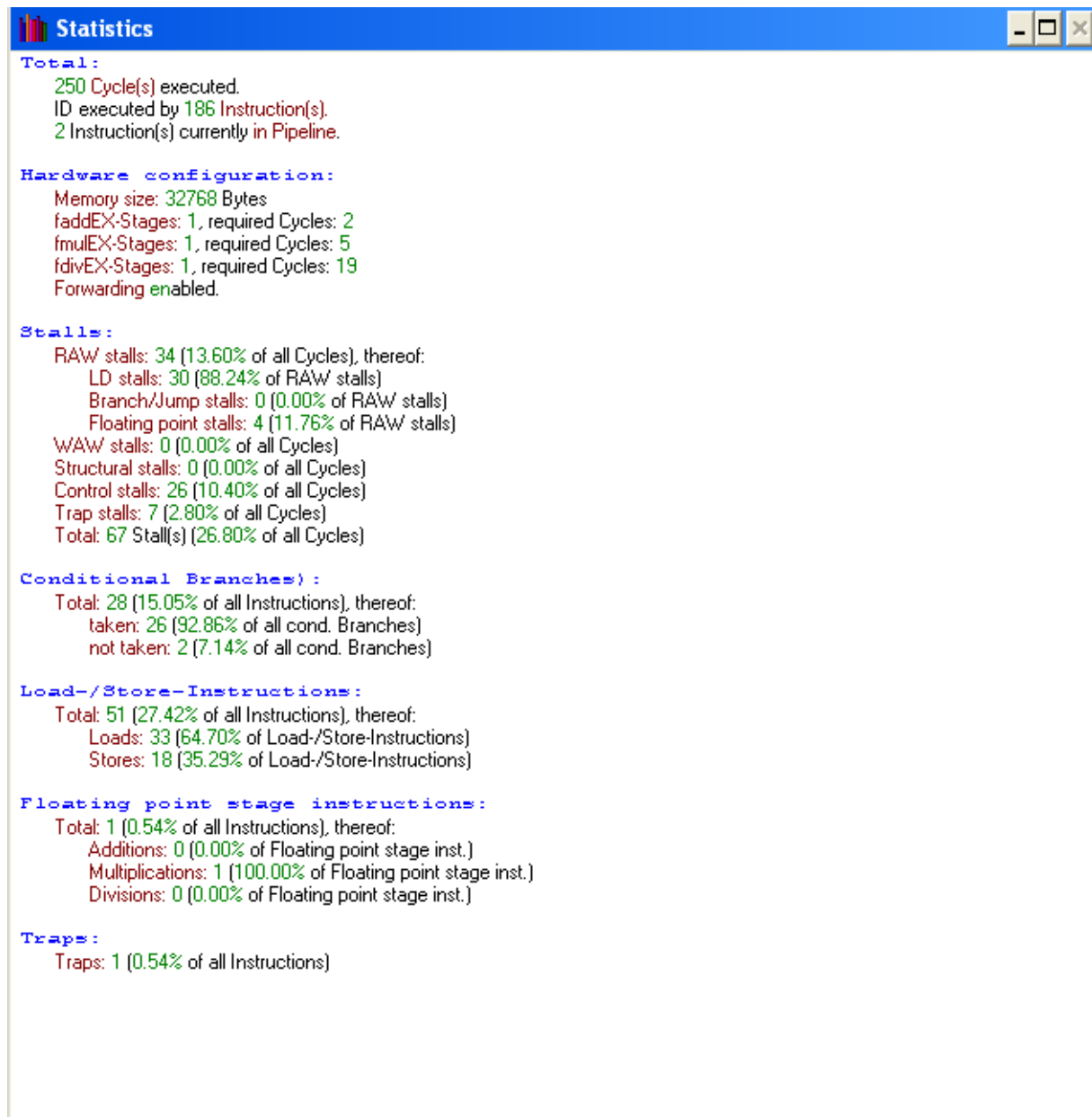
Code		
\$TEXT	0x20141004	addi r20,r0,0x1004
\$TEXT+0x4	0x201e000e	addi r30,r0,0xe
\$TEXT+0x8	0x20070000	addi r7,r0,0x0
start1	0x8e820000	lw r2,0x0(r20)
start1+0x4	0x00e23820	add r7,r7,r2
start1+0x8	0x2bde0001	subi r30,r30,0x1
start1+0xc	0x22940004	addi r20,r20,0x4
0x0000011c	0x17c0ffec	bnez r30,start1
0x00000120	0xac07103c	sw suma1(r0),r7
0x00000124	0x8c011040	lw r1,stala(r0)
0x00000128	0x20141004	addi r20,r0,0x1004
0x0000012c	0x201e000e	addi r30,r0,0xe
0x00000130	0x20070000	addi r7,r0,0x0
start2	0x8e820000	lw r2,0x0(r20)
start2+0x4	0x00225020	add r10,r1,r2
start2+0x8	0xae8a0000	sw 0x0(r20),r10
start2+0xc	0x00ea3820	add r7,r7,r10
0x00000144	0x2bde0001	subi r30,r30,0x1
0x00000148	0x22940004	addi r20,r20,0x4



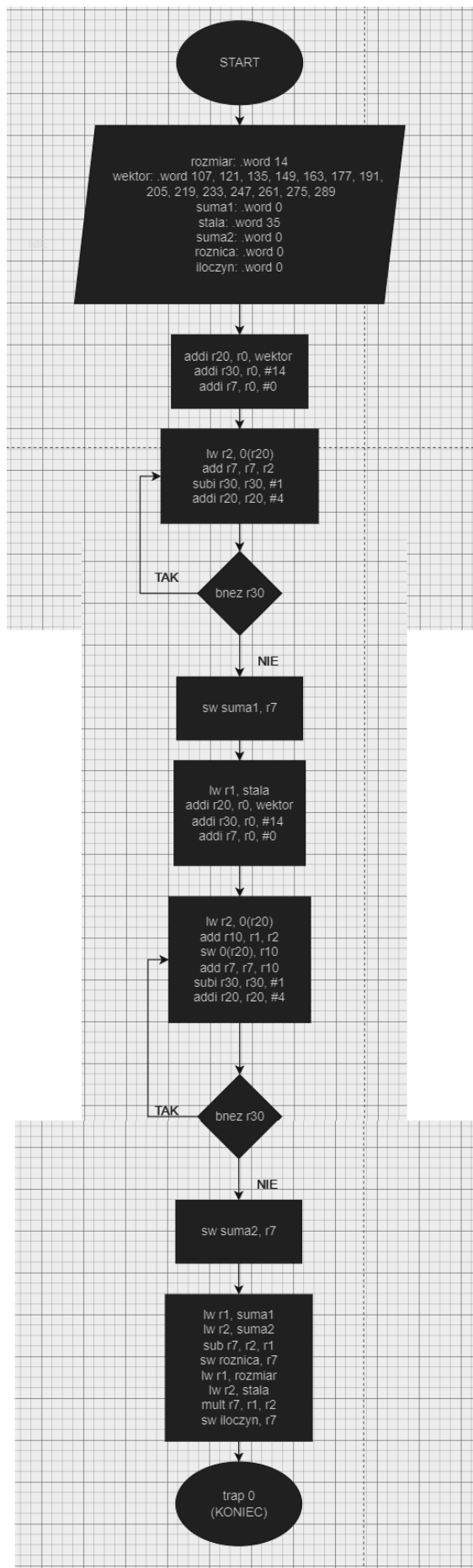
Po:

Memory-1		Register			
0x00000ffc	0	PC=	360	R4=	0
\$DATA	14	IMAR=	376	R5=	0
vektor	142	IR=	0	R6=	0
vektor+0x4	156	A=	0	R7=	490
vektor+0x8	170	AHI=	0	R8=	0
vektor+0xc	184	B=	0	R9=	0
0x00001014	198	BHI=	0	R10=	324
0x00001018	212	BTA=	0	R11=	0
0x0000101c	226	ALU=	0	R12=	0
0x00001020	240	ALUHI=	0	R13=	0
0x00001024	254	FP3R=	0	R14=	0
0x00001028	268	DMAR=	4172	R15=	0
0x0000102c	282	SDR=	490	R16=	0
0x00001030	296	SDRHI=	0	R17=	0
0x00001034	310	LDR=	0	R18=	0
0x00001038	324	LDRHI=	0	R19=	0
suma1	2772	RO=	0	R20=	4156
stala	35	R1=	14	R21=	0
suma2	3262	R2=	35	R22=	0
roznica	490	R3=	0	R23=	0
iloczyn	490				
0x00001050	0				





D.



Program na początku deklaruje zmienne rozmiar, wektor, suma1, stała, suma2, roznica, iloczyn i przypisuje im odpowiednie wartości. W rejestrze r20 przechowywany jest pierwszy element wektora, a w r30 wartość równa liczbie elementów wektora, w tym przypadku 14. W r7 będziemy przechowywać wyniki obliczeń.

Do r2 ładujemy pierwszy element wektora i dodajemy go do r7. Zmniejszamy r30 o jeden. Do r20 dodajemy 4, przez co r20 wskazuje teraz na kolejny element wektora. Po tym sprawdzamy czy wartość r30 nie jest równa 0. Jeżeli r30 nie jest równe zero to ładujemy do r2 kolejny element wektora i dodajemy go do r7, znowu odejmujemy jeden od wartości r30, dodajemy 4 do r20, żeby wskazywać kolejny element wektora i znów sprawdzamy czy wartość r30 nie jest równa 0. Postępujemy tak, aż wartość rejestru r30 będzie równa 0, wtedy wartość rejestru r7 zapisujemy w zmiennej suma1.

Następnie ładujemy do r1 wartość zmiennej stała, która w tym przypadku wynosi 35. Rejestry r20 i r30 ustawiamy jak na początku programu, tj. wartość r30 równa się 14, a r20 wskazuje na pierwszy element zmiennej wektor. Zerujemy wartość rejestru r7.

Do r2 ładujemy pierwszy element wektora, a sumę wartości r2 i r1 zapisujemy w rejestrze r10. Następnie wartość rejestru r10 zapisujemy jako pierwszy element tablicy wektor. W ten sposób udało się zwiększyć wartość elementu wektora o wartość zmiennej stała. Wartość rejestru r10 dodajemy do rejestru r7, dekrementujemy r30 i dodajemy 4 do r20, żeby ten rejestr wskazywał na kolejny element tablicy wektor.

Sprawdzamy czy wartość rejestru r30 nie równa się zero. Póki wartość r30 jest nierówna 0 to powtarzamy operacje z powyższego akapitu na kolejnych elementach tablicy wektor. Gdy wartość rejestru r30 będzie równa 0 zapiszemy wartość rejestru r7 do zmiennej suma2.

Do rejestru r1 ładujemy wartość zmiennej suma1, a do r2 ładujemy wartość zmiennej suma2. W rejestrze r7 zapisujemy różnicę r2 i r1, a następnie wartość rejestru r7 zapisujemy w zmiennej roznica.

Po tym nadpisujemy wartość rejestru r1 wartością zmiennej rozmiar, a wartość rejestru r2 nadpisujemy wartością zmiennej stała. W rejestrze r7 zapisujemy wynik mnożenia r1 i r2, następnie wartość r7 zapisujemy do zmiennej iloczyn.

Na koniec program wykonuje polecenie trap 0, co kończy program.