

Riešenie príkladu pre MIPSIM  
Princípy počítačového inžinierstva 2022  
Marek Čederle (AIS ID:121193)

Zadanie (č. 31)

Napíšte program, ktorý bude simulovať hru Pac-Man na hracom poli podľa obrázka. Ľavé horné políčko hracieho poľa má súradnice (riadok, stĺpec) = (1,1) a pravé spodné políčko má súradnice (5,5). Na hracom poli sa nachádzajú rozmiestnené bodky. Ak hráč vstúpi na políčko s bodkou, skonzumuje ju a táto bodka zmizne. Za každú skonsumovanú bodku získa hráč 50 bodov. Hráč môže začínať hru na niektorom z voľných políčok a môže vykonávať kroky o 1 políčko smerom na niektorú svetovú stranu.

	1	2	3	4	5
1		1	2	3	4
2	1	•		•	•
3	2				
4	3	•		•	•
5	4				
	5	•		•	•

V pamäti údajov (PÚ) uchovávať riadkovú súradnicu hráča na adrese **a0h** a stĺpcovú na adrese **b0h**. Od adresy **0h** so 4-bajtovými rozstupmi (4h, 8h, ch, 10h, 14h, 18h, 1ch, 20h, atď.) bude pred spustením programu v pamäti údajov uložená postupnosť hodnôt reprezentujúcich pohyby hráča o 1 políčko nasledovne:

- **1h** – pohyb hore,
- **2h** – pohyb vpravo,
- **3h** – pohyb dole,
- **4h** – pohyb vľavo,
- **0h** – koniec.

Po načítaní hodnoty **0h** sa program ukončí. Môžete predpokladať, že v postupnosti sa iné čísla ako **0h-4h** nebudú nachádzať. Zistite, koľko bodov počas vykonávania zadanej postupnosti hráč nazbieral. V prípade, ak nazbieral maximálny počet bodov skonsumovaním všetkých bodiek na hracom poli, program by mal ihneď skončiť a na adresu **c0h** uložte hodnotu „1“. V ostatných prípadoch, teda ak program načíta ukončovací znak **0h**, uložte na adresu **c0h** hodnotu 0. Aktuálnu (a teda aj štartovaciu) riadkovú a stĺpcovú súradnicu hráča uchovávať na adresách **a0h** resp. **b0h**. Aktuálny počet bodov uchovávať na adrese **d0h**.

## Riešenie

### Pamäť programu

Adr.	Label	Inštrukcia	Komentár
0h		LW \$25,00a0(\$0)	načítame začiatočnú riadkovú súradnicu z PÚ z adresy a0h do registra R25
4h		LW \$26,00b0(\$0)	načítame začiatočnú stĺpcovú súradnicu z PÚ z adresy b0h do registra R26
8h		LI \$27,0000	Načítame „výhernú nulu“ do registra R27
ch	zac	LW \$22,0000(\$20)	do registra R22 načítame prvok postupnosti z PÚ z adresy, na ktorú ukazuje ukazovateľ v registri R20
10h		ADDI \$20,\$20,0004	zvážšime ukazovateľ v registri R20 o 4, aby ukazoval na ďalší prvok postupnosti v poradí
14h		NOP	Jeden NOP aby bola medzera medzi závislými premennými aspoň 2 nezávislé inštrukcie
18h		BEQ \$22,\$13,jedna	ak je načítaný prvok postupnosti v reg. R22 rovný 1 (konštantu 1 máme uloženú v reg. R13) skoč na podprogram pre vykonanie pohybu hore ktorý sa nachádza na labeli „jedna“
1ch		BEQ \$22,\$14,dva	ak je načítaný prvok postupnosti v reg. R22 rovný 2 (konštantu 2 máme uloženú v reg. R14) skoč na podprogram pre vykonanie pohybu vpravo ktorý sa nachádza na labeli „dva“
20h		BEQ \$22,\$15,tri	ak je načítaný prvok postupnosti v reg. R22 rovný 3 (konštantu 3 máme uloženú v reg. R15) skoč na podprogram pre vykonanie pohybu dole ktorý sa nachádza na labeli „tri“
24h		BEQ \$22,\$16,styri	ak je načítaný prvok postupnosti v reg. R22 rovný 4 (konštantu 4 máme uloženú v reg. R16) skoč na podprogram pre vykonanie pohybu vľavo ktorý sa nachádza na labeli „styri“
28h		BEQ \$0,\$0,kon	Ak nebolo ani jedno z predchadzajúcich musí byť 0 tým pádom ideme na koniec programu
2ch		NOP	Iba esteticky oddeľovací NOP
30h	jedna	SUBI \$25,\$25,0001	Odčíta riadkovú súradnicu tým pádom nás posunie nahor
34h		BEQ \$0,\$0,oriad	Skok na overovanie riadkovej súradnice guľčiek
38h		NOP	Iba esteticky oddeľovací NOP
3ch	dva	ADDI \$26,\$26,0001	Pričíta stĺpcovú súradnicu tým pádom nás posunie vpravo
40h		BEQ \$0,\$0,oriad	Skok na overovanie riadkovej súradnice guľčiek
44h		NOP	Iba esteticky oddeľovací NOP
48h	tri	ADDI \$25,\$25,0001	Pričíta riadkovú súradnicu tým pádom nás posunie dole
4ch		BEQ \$0,\$0,oriad	Skok na overovanie riadkovej súradnice guľčiek
50h		NOP	Iba esteticky oddeľovací NOP
54h	styri	SUBI \$26,\$26,0001	Odčíta stĺpcovú súradnicu tým pádom nás posunie vľavo
58h		BEQ \$0,\$0,oriad	Skok na overovanie riadkovej súradnice guľčiek
5ch		NOP	Iba esteticky oddeľovací NOP
60h	oriad	BEQ \$25,\$13,ostlp	Skok na overenie stĺpca pre prvý riadok
64h		BEQ \$25,\$15,ostl	Skok na overenie stĺpca pre tretí riadok
68h		BEQ \$25,\$17,ost	Skok na overenie stĺpca pre piaty riadok
6ch		BEQ \$0,\$0,gula	Skok na prácu s dátami ak sme nenašli guľčku
70h		NOP	Iba esteticky oddeľovací NOP
74h	ostlp	BEQ \$26,\$13,one	Skok na overenie či sme danú guľčku zjedli
78h		BEQ \$26,\$15,two	Skok na overenie či sme danú guľčku zjedli
7ch		BEQ \$26,\$17,three	Skok na overenie či sme danú guľčku zjedli
80h		BEQ \$0,\$0,gula	Skok na prácu s dátami ak sme nenašli guľčku

84h		NOP	Iba esteticky oddeľovací NOP
88h	ostl	BEQ \$26,\$13,four	Skok na overenie či sme danú guličku zjedli
8ch		BEQ \$26,\$15,five	Skok na overenie či sme danú guličku zjedli
90h		BEQ \$26,\$17,six	Skok na overenie či sme danú guličku zjedli
94h		BEQ \$0,\$0,gula	Skok na prácu s dátami ak sme nenašli guličku
98h		NOP	Iba esteticky oddeľovací NOP
9ch	ost	BEQ \$26,\$13,seven	Skok na overenie či sme danú guličku zjedli
A0h		BEQ \$26,\$15,eight	Skok na overenie či sme danú guličku zjedli
A4h		BEQ \$26,\$17,nine	Skok na overenie či sme danú guličku zjedli
A8h		BEQ \$0,\$0,gula	Skok na prácu s dátami ak sme nenašli guličku
ach		NOP	Iba esteticky oddeľovací NOP
B0h	one	BEQ \$3,\$13,un	Skok na prácu s dátami danej guličky aj ju ideme zjesť
B4h		BEQ \$0,\$0,gula	Skok na prácu s dátami ak sme nezjedli guličku
B8h		NOP	Iba esteticky oddeľovací NOP
Bch	two	BEQ \$4,\$13,deux	Skok na prácu s dátami danej guličky aj ju ideme zjesť
C0h		BEQ \$0,\$0,gula	Skok na prácu s dátami ak sme nezjedli guličku
C4h		NOP	Iba esteticky oddeľovací NOP
C8h	Three	BEQ \$5,\$13,trois	Skok na prácu s dátami danej guličky aj ju ideme zjesť
Cch		BEQ \$0,\$0,gula	Skok na prácu s dátami ak sme nezjedli guličku
D0h		NOP	Iba esteticky oddeľovací NOP
D4h	Four	BEQ \$6,\$13,quatr	Skok na prácu s dátami danej guličky aj ju ideme zjesť
D8h		BEQ \$0,\$0,gula	Skok na prácu s dátami ak sme nezjedli guličku
Dch		NOP	Iba esteticky oddeľovací NOP
E0h	Five	BEQ \$7,\$13,cinq	Skok na prácu s dátami danej guličky aj ju ideme zjesť
E4h		BEQ \$0,\$0,gula	Skok na prácu s dátami ak sme nezjedli guličku
E8h		NOP	Iba esteticky oddeľovací NOP
Ech	Six	BEQ \$8,\$13,siks	Skok na prácu s dátami danej guličky aj ju ideme zjesť
F0h		BEQ \$0,\$0,gula	Skok na prácu s dátami ak sme nezjedli guličku
F4h		NOP	Iba esteticky oddeľovací NOP
F8h	Seven	BEQ \$9,\$13,sept	Skok na prácu s dátami danej guličky aj ju ideme zjesť
Fch		BEQ \$0,\$0,gula	Skok na prácu s dátami ak sme nezjedli guličku
100h		NOP	Iba esteticky oddeľovací NOP
104h	Eight	BEQ \$10,\$13,huit	Skok na prácu s dátami danej guličky aj ju ideme zjesť
108h		BEQ \$0,\$0,gula	Skok na prácu s dátami ak sme nezjedli guličku
10ch		NOP	Iba esteticky oddeľovací NOP
110h	nine	BEQ \$11,\$13,neuf	Skok na prácu s dátami danej guličky aj ju ideme zjesť
114h		BEQ \$0,\$0,gula	Skok na prácu s dátami ak sme nezjedli guličku
118h		NOP	Iba esteticky oddeľovací NOP
11ch	Un	SUBI \$3,\$3,0001	Zjedenie danej guličky
120h		BEQ \$0,\$0,befor	Skok na prácu s dátami pre upravenie poctu zjedených guličiek
124h		NOP	Iba esteticky oddeľovací NOP
128h	Deux	SUBI \$4,\$4,0001	Zjedenie danej guličky
12ch		BEQ \$0,\$0,befor	Skok na prácu s dátami pre upravenie poctu zjedených guličiek
130h		NOP	Iba esteticky oddeľovací NOP
134h	Trois	SUBI \$5,\$5,0001	Zjedenie danej guličky
138h		BEQ \$0,\$0,befor	Skok na prácu s dátami pre upravenie poctu zjedených guličiek
13ch		NOP	Iba esteticky oddeľovací NOP
140h	Quatr	SUBI \$6,\$6,0001	Zjedenie danej guličky
144h		BEQ \$0,\$0,befor	Skok na prácu s dátami pre upravenie poctu zjedených guličiek
148h		NOP	Iba esteticky oddeľovací NOP
14ch	Cinq	SUBI \$7,\$7,0001	Zjedenie danej guličky

150h		BEQ \$0,\$0,befor	Skok na prácu s dátami pre upravenie poctu zjedených guličiek
154h		NOP	Iba esteticky oddeľovací NOP
158h	Siks	SUBI \$8,\$8,0001	Zjedenie danej guličky
15ch		BEQ \$0,\$0,befor	Skok na prácu s dátami pre upravenie poctu zjedených guličiek
160h		NOP	Iba esteticky oddeľovací NOP
164h	sept	SUBI \$9,\$9,0001	Zjedenie danej guličky
168h		BEQ \$0,\$0,befor	Skok na prácu s dátami pre upravenie poctu zjedených guličiek
16ch		NOP	Iba esteticky oddeľovací NOP
170h	Huit	SUBI \$10,\$10,0001	Zjedenie danej guličky
174h		BEQ \$0,\$0,befor	Skok na prácu s dátami pre upravenie poctu zjedených guličiek
178h		NOP	Iba esteticky oddeľovací NOP
17ch	neuf	SUBI \$11,\$11,0001	Zjedenie danej guličky
180h		BEQ \$0,\$0,befor	Skok na prácu s dátami pre upravenie poctu zjedených guličiek
184h		NOP	Iba esteticky oddeľovací NOP
188h	befor	SUBI \$1,\$1,0001	Odčítanie poctu nezjedených guličiek
18ch		ADDI \$28,\$28,0032	Pridanie 50 bodov čo predstavuje číslo 32 hexadecimálne
190h		NOP	Dva NOPy na oddelenie závislých inštrukcií
194h		NOP	Dva NOPy na oddelenie závislých inštrukcií
198h		SW \$28,00d0(\$0)	Zapísanie bodov
19ch	gula	BEQ \$1,\$0,won	Kontrola či sme vyhrali, ak hej skočí na návěstie won
1a0h		SW \$25, 00a0(\$0)	Ak sme nevyhrali tak zapíše riadkovú súradnicu
1a4h		SW \$26, 00b0(\$0)	Ďalej zapíše stĺpcovú súradnicu
1a8h		BEQ \$0,\$0,zac	Skočí na začiatok programu
1ach	won	LI \$27,0001	Ak sme vyhrali tak načíta výhernú jednotku do registra časti
1b0h		NOP	Dva NOPy na oddelenie závislých inštrukcií
1b4h		NOP	Dva NOPy na oddelenie závislých inštrukcií
1b8h		SW \$27,00c0(\$0)	Ak sme vyhrali tak zapíše výhernú jednotku do dátovej časti
1bch	kon	NOP	Konečný NOP
1c0h		...	Samé nopy na konci

Simulujeme nasledujúce 4 postupnosti:

riadky	2	5	3	1
stĺpce	2	4	4	2
0h	2	1	4	2
4h	2	1	4	2
8h	3	4	4	2
ch	3	2	1	3
10h	2	2	1	3
14h	4	3	2	4
18h	3	3	2	4
1ch	1	1	2	3
20h	4	1	2	3
24h	4	3	3	1
28h	3	3	3	1
2ch	1	0	3	0
30h	1		3	
34h	1		4	
38h	1		4	
3ch	0		4	
40h			4	

44h			1	
48h			1	
4ch			0	

Obsah registrov a pamäti údajov pred spustením programu

Registre sú vždy na začiatku rovnaké pre všetky postupnosti len dáta sú rôzne.

Register window showing 32 registers (R0 to R31) with their current values. The values are mostly 0, with some non-zero values in R1, R14, and R15. Buttons 'Clear', 'Load', and 'Save' are visible at the bottom.

1. Postupnosť

data memory window showing a table of memory addresses and their corresponding data values. The table has 5 columns of data. The address column ranges from 00000000 to 00000150. The data columns show various values, mostly 0, with some non-zero values in the first few rows. Buttons 'load', 'save', 'clear', and 'close' are visible at the bottom.

## 2. Postupnosť

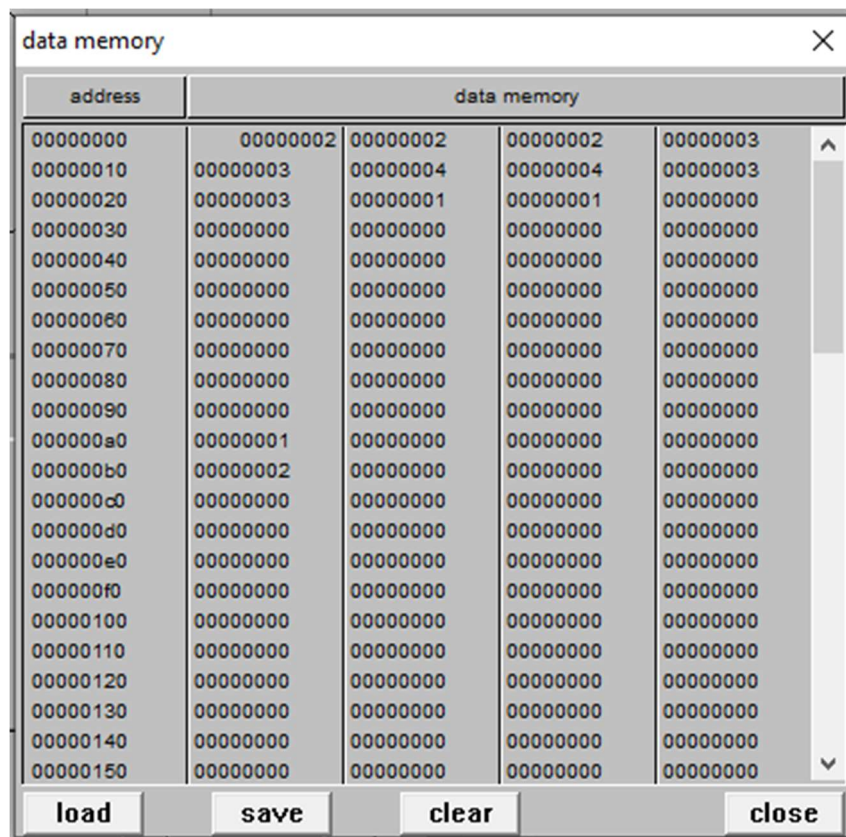
data memory					×
address	data memory				
00000000	00000001	00000001	00000004	00000002	^
00000010	00000002	00000003	00000003	00000001	
00000020	00000001	00000003	00000003	00000000	
00000030	00000000	00000000	00000000	00000000	
00000040	00000000	00000000	00000000	00000000	
00000050	00000000	00000000	00000000	00000000	
00000060	00000000	00000000	00000000	00000000	
00000070	00000000	00000000	00000000	00000000	
00000080	00000000	00000000	00000000	00000000	
00000090	00000000	00000000	00000000	00000000	
000000a0	00000005	00000000	00000000	00000000	
000000b0	00000004	00000000	00000000	00000000	
000000c0	00000000	00000000	00000000	00000000	
000000d0	00000000	00000000	00000000	00000000	
000000e0	00000000	00000000	00000000	00000000	
000000f0	00000000	00000000	00000000	00000000	
00000100	00000000	00000000	00000000	00000000	
00000110	00000000	00000000	00000000	00000000	
00000120	00000000	00000000	00000000	00000000	
00000130	00000000	00000000	00000000	00000000	
00000140	00000000	00000000	00000000	00000000	
00000150	00000000	00000000	00000000	00000000	v
load	save	clear			close

## 3. Postupnosť

data memory					×
address	data memory				
00000000	00000004	00000004	00000004	00000001	^
00000010	00000001	00000002	00000002	00000002	
00000020	00000002	00000003	00000003	00000003	
00000030	00000003	00000004	00000004	00000004	
00000040	00000004	00000001	00000001	00000000	
00000050	00000000	00000000	00000000	00000000	
00000060	00000000	00000000	00000000	00000000	
00000070	00000000	00000000	00000000	00000000	
00000080	00000000	00000000	00000000	00000000	
00000090	00000000	00000000	00000000	00000000	
000000a0	00000003	00000000	00000000	00000000	
000000b0	00000004	00000000	00000000	00000000	
000000c0	00000000	00000000	00000000	00000000	
000000d0	00000000	00000000	00000000	00000000	
000000e0	00000000	00000000	00000000	00000000	
000000f0	00000000	00000000	00000000	00000000	
00000100	00000000	00000000	00000000	00000000	
00000110	00000000	00000000	00000000	00000000	
00000120	00000000	00000000	00000000	00000000	
00000130	00000000	00000000	00000000	00000000	
00000140	00000000	00000000	00000000	00000000	
00000150	00000000	00000000	00000000	00000000	v
load	save	clear			close



#### 4. Postupnosť



Register	Údaj	Komentár
R1	9h	Počet nezjedených guľičiek
R3	1h	1 znamená že guľička na súradniciach 1,1 nie je zjedená
R4	1h	1 znamená že guľička na súradniciach 1,3 nie je zjedená
R5	1h	1 znamená že guľička na súradniciach 1,5 nie je zjedená
R6	1h	1 znamená že guľička na súradniciach 3,1 nie je zjedená
R7	1h	1 znamená že guľička na súradniciach 3,3 nie je zjedená
R8	1h	1 znamená že guľička na súradniciach 3,5 nie je zjedená
R9	1h	1 znamená že guľička na súradniciach 5,1 nie je zjedená
R10	1h	1 znamená že guľička na súradniciach 5,3 nie je zjedená
R11	1h	1 znamená že guľička na súradniciach 5,5 nie je zjedená
R13	1h	konštanta 1 na porovnávanie
R14	2h	konštanta 2 na porovnávanie
R15	3h	konštanta 3 na porovnávanie
R16	4h	konštanta 4 na porovnávanie
R17	5h	konštanta 5 na porovnávanie
R20	0h	ukazovateľ do postupnosti prvkov, na začiatku ukazuje na 1. prvok
R22	0h	sem sa bude načítavať prvok postupnosti z pamäte údajov
R25	0h	sem sa načíta začiatočná riadková súradnica z PÚ z adresy a0h následne sa bude počas behu programu aktualizovať
R26	0h	sem sa načíta stĺpcová súradnica z PÚ z adresy b0h následne sa bude počas behu programu aktualizovať
R27	0h	1/0 pre výhru/prehru
R28	0h	Počet bodov

Adresa	Údaj	Komentár
0h – 4ch	2h, 2h, 3h, 3h, 2h, ..., 0h	postupnosť krokov
a0h	2h	začiatková riadková súradnica
b0h	2h	začiatková stĺpcová súradnica
c0h	0h	1/0 pre výhru/prehru
d0h	0h	Počet bodov

Obsah registrov a pamäti údajov po spustení programu

### 1. Postupnosť

Register

R0 00000000

R1 00000009

R2 00000000

R3 00000001

R4 00000001

R5 00000001

R6 00000001

R7 00000001

R8 00000001

R9 00000001

R10 00000001

R11 00000001

R12 00000000

R13 00000001

R14 00000002

R15 00000003

R16 00000004

R17 00000005

R18 00000000

R19 00000000

R20 00000040

R21 00000000

R22 00000000

R23 00000000

R24 00000000

R25 00000001

R26 00000002

R27 00000000

R28 00000000

R29 00000000

R30 00000000

R31 00000000

Clear

Load

Save

data memory

address

data memory

00000000

00000010

00000020

00000030

00000040

00000050

00000060

00000070

00000080

00000090

000000a0

000000b0

000000c0

000000d0

000000e0

000000f0

00000100

00000110

00000120

00000130

00000140

00000150

00000002

00000002

00000004

00000004

00000001

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000002

00000004

00000004

00000001

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000003

00000003

00000003

00000001

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000003

00000001

00000001

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

00000000

load

save

clear

close



## 2. Postupnost

Register

×

R0	00000000	R16	00000004
R1	00000006	R17	00000005
R2	00000000	R18	00000000
R3	00000001	R19	00000000
R4	00000001	R20	00000030
R5	00000001	R21	00000000
R6	00000001	R22	00000000
R7	00000000	R23	00000000
R8	00000000	R24	00000000
R9	00000001	R25	00000005
R10	00000001	R26	00000005
R11	00000000	R27	00000000
R12	00000000	R28	00000096
R13	00000001	R29	00000000
R14	00000002	R30	00000000
R15	00000003	R31	00000000

Clear

Load

Save

data memory

×

address	data memory			
00000000	00000001	00000001	00000004	00000002
00000010	00000002	00000003	00000003	00000001
00000020	00000001	00000003	00000003	00000000
00000030	00000000	00000000	00000000	00000000
00000040	00000000	00000000	00000000	00000000
00000050	00000000	00000000	00000000	00000000
00000060	00000000	00000000	00000000	00000000
00000070	00000000	00000000	00000000	00000000
00000080	00000000	00000000	00000000	00000000
00000090	00000000	00000000	00000000	00000000
000000a0	00000005	00000000	00000000	00000000
000000b0	00000005	00000000	00000000	00000000
000000c0	00000000	00000000	00000000	00000000
000000d0	00000096	00000000	00000000	00000000
000000e0	00000000	00000000	00000000	00000000
000000f0	00000000	00000000	00000000	00000000
00000100	00000000	00000000	00000000	00000000
00000110	00000000	00000000	00000000	00000000
00000120	00000000	00000000	00000000	00000000
00000130	00000000	00000000	00000000	00000000
00000140	00000000	00000000	00000000	00000000
00000150	00000000	00000000	00000000	00000000

load

save

clear

close

### 3. Postupnost

Register

R0	00000000	R16	00000004
R1	00000000	R17	00000005
R2	00000000	R18	00000000
R3	00000000	R19	00000000
R4	00000000	R20	00000044
R5	00000000	R21	00000000
R6	00000000	R22	00000004
R7	00000000	R23	00000000
R8	00000000	R24	00000000
R9	00000000	R25	00000005
R10	00000000	R26	00000001
R11	00000000	R27	00000001
R12	00000000	R28	00001c2
R13	00000001	R29	00000000
R14	00000002	R30	00000000
R15	00000003	R31	00000000

Clear

Load

Save

data memory

address	data memory			
00000000	00000004	00000004	00000004	00000001
00000010	00000001	00000002	00000002	00000002
00000020	00000002	00000003	00000003	00000003
00000030	00000003	00000004	00000004	00000004
00000040	00000004	00000001	00000001	00000000
00000050	00000000	00000000	00000000	00000000
00000060	00000000	00000000	00000000	00000000
00000070	00000000	00000000	00000000	00000000
00000080	00000000	00000000	00000000	00000000
00000090	00000000	00000000	00000000	00000000
000000a0	00000005	00000000	00000000	00000000
000000b0	00000002	00000000	00000000	00000000
000000c0	00000001	00000000	00000000	00000000
000000d0	00001c2	00000000	00000000	00000000
000000e0	00000000	00000000	00000000	00000000
000000f0	00000000	00000000	00000000	00000000
00000100	00000000	00000000	00000000	00000000
00000110	00000000	00000000	00000000	00000000
00000120	00000000	00000000	00000000	00000000
00000130	00000000	00000000	00000000	00000000
00000140	00000000	00000000	00000000	00000000
00000150	00000000	00000000	00000000	00000000

load

save

clear

close

#### 4. Postupnosť

Register	Value
R0	00000000
R1	00000004
R2	00000000
R3	00000001
R4	00000000
R5	00000000
R6	00000001
R7	00000000
R8	00000000
R9	00000001
R10	00000000
R11	00000001
R12	00000000
R13	00000001
R14	00000002
R15	00000003
R16	00000004
R17	00000005
R18	00000000
R19	00000000
R20	00000030
R21	00000000
R22	00000000
R23	00000000
R24	00000000
R25	00000003
R26	00000003
R27	00000000
R28	000000fa
R29	00000000
R30	00000000
R31	00000000

Clear Load Save

address	data memory			
00000000	00000002	00000002	00000002	00000003
00000010	00000003	00000004	00000004	00000003
00000020	00000003	00000001	00000001	00000000
00000030	00000000	00000000	00000000	00000000
00000040	00000000	00000000	00000000	00000000
00000050	00000000	00000000	00000000	00000000
00000060	00000000	00000000	00000000	00000000
00000070	00000000	00000000	00000000	00000000
00000080	00000000	00000000	00000000	00000000
00000090	00000000	00000000	00000000	00000000
000000a0	00000003	00000000	00000000	00000000
000000b0	00000003	00000000	00000000	00000000
000000c0	00000000	00000000	00000000	00000000
000000d0	000000fa	00000000	00000000	00000000
000000e0	00000000	00000000	00000000	00000000
000000f0	00000000	00000000	00000000	00000000
00000100	00000000	00000000	00000000	00000000
00000110	00000000	00000000	00000000	00000000
00000120	00000000	00000000	00000000	00000000
00000130	00000000	00000000	00000000	00000000
00000140	00000000	00000000	00000000	00000000
00000150	00000000	00000000	00000000	00000000

load save clear close

#### Zhodnotenie

Myslím si že som relatívne dobre použil prúdové spracovanie inštrukcií keďže veľa nepoužívam NOPy ako oddeľovače od závislých dát ale iba ako estetické pomôcky. Snažil som sa vtesnať inštrukcie aby dávali za sebou zmysel a to je dôležité. Program MIPSIM mi však prišiel menej prehľadný ako program RegSim avšak určite inštrukcie spracúval rýchlejšie.