Západočeská univerzita v Plzni  
Fakulta aplikovaných věd  
Katedra informatiky a výpočetní techniky

**Semestrální práce z předmětu UPA**

Mediánový filtr

**David Pivovar** 1. 12. 2015  
A13B0403P

**1. Zadání úlohy**

Postupně se budou zadávat čísla int16. Na výstupu se bude postubně zobrazovat hodnota, určená jako medián z 5 posledních zadaných hodnot.

**2. Popis algoritmu řešení**

Do paměti se ukládá 5 čísel tak, jak jsou za sebou zadávány. Po každém zadání se čísla načtou do registrů a seřadí. Na obrazovku se vypíše číslo z “prostředního” registru. Zadáním čísla -9999 se program ukončí.

**3. Výpis programu**

# ------ datovy segment --------------------------

.data

text\_b: .asciiz "Program pro vypocet medianu\npro ukonceni zadejte cislo -9999\n"

text\_z: .asciiz "\nZadej cislo: "

text\_m: .asciiz "\nMedian: "

text\_k: .asciiz "\nKonec."

cis0: .half -9999 #cislo pro ukonceni programu: -9999

cis1: .half 0x0

cis2: .half 0x0

cis3: .half 0x0

cis4: .half 0x0

cis5: .half 0x0

# ------ kodovy segment --------------------------

.text

.globl main

main: li $v0,4 # syscall print\_string - cislo sluzby do $v0

la $a0,text\_b # adresa textu do $a0

syscall # volani systemovych sluzeb

nop

move $s7, $ra #ulozeni navratove adresy z programu do $s7

lh $s0, cis0 #ulozeni ukoncovaciho cisla do $s0

nop

#smycka

la $t0, loop

pseudoinstrukce:

lui $1, 64 [loop]

ori $8, $1, 152 [loop]

jr $t0

nop

loop: #nacteni noveho cisla

la $t0, load #adresa load do $t0

jalr $t1, $t0 #skok na load - navratova adresa v $t1

nop

#Vypis medianu

li $v0,4 # syscall print\_string - cislo sluzby do $v0

la $a0,text\_m # adresa textu do $a0

syscall # volani systemovych sluzeb

nop

li $v0, 1 #syscall print\_int - cislo sluzby do $v0

move $a0, $s3 #v $s3 median do $a0 pro syscall

pseudoinstrukce:

addu $4, $0, $19

syscall

nop

#smycka #skok na zacatek - smycka dokud se zadava cislo

la $t0, loop

jr $t0

nop

load: li $v0,4 # syscall print\_string - cislo sluzby do $v0

la $a0,text\_z # adresa textu do $a0

syscall # volani systemovych sluzeb

nop

li $v0, 5 #syscall read\_int

syscall

nop

#test na konec

beq $v0, $s0, end #pokud se zadane cislo rovna -9999, skoci se na konec

V následující části kódu se projevuje zpožděné čtení dt z paměti – možnost datového hazardu

#presun cisel do registru

#zahazuje se cis5

move $s1, $v0

lh $s2, cis1

pseudoinstrukce:

lui $1, 4097

lh $18, 98($1)

lh $s3, cis2

lh $s4, cis3

lh $s5, cis4

#ulozeni novych hodnot do pameti

sh $s1, cis1

pseudoinstrukce:

lui $1, 4097

ls $17, 98($1)

sh $s2, cis2

sh $s3, cis3

sh $s4, cis4

sh $s5, cis5

#bubble sort - serazeni cisel v registrech

la $t0, sort

jalr $t2 $t0 #skok na sort - navratova adresa v $t2

nop

jalr $t2 $t0

nop

jalr $t2 $t0

nop

jalr $t2 $t0

nop

jr $t1 #skok zpet do loop (po load)

sort: sub $t3, $s1, $s2

bgezal $t3, swap12 #skok na swap, pokud je $t3 vetsi nez 0 (prvni cislo je vestsi

nop

sub $t3, $s2, $s3

bgezal $t3, swap23 #skok na swap, pokud je $t3 vetsi nez 0 (prvni cislo je vestsi nop

sub $t3, $s3, $s4

bgezal $t3, swap34 #skok na swap, pokud je $t3 vetsi nez 0 (prvni cislo je vestsi) nop

sub $t3, $s4, $s5

bgezal $t3, swap45 #skok na swap, pokud je $t3 vetsi nez 0 (prvni cislo je vestsi) nop

jr $t2 #skok zpet do load

swap12: move $t3, $s1

move $s1, $s2

move $s2, $t3

jr $ra

swap23: move $t3, $s2

move $s2, $s3

move $s3, $t3

jr $ra

swap34: move $t3, $s3

move $s3, $s4

move $s4, $t3

jr $ra

swap45: move $t3, $s4

move $s4, $s5

move $s5, $t3

jr $ra

end: li $v0,4 # syscall print\_string - cislo sluzby do $v0

la $a0,text\_k # adresa textu do $a0

syscall # volani systemovych sluzeb

nop

jr $s7

nop

**4. Spuštění programu**

Program se spouští pomocí simulátoru QtSpim se zapnutou simulací zpoždění instrukce load a skoků.

Program běží v cyklu, po každém zadání čísla se spočte medián.

Zadáním čísla -9999 se program ukončí.