HOMEWORK 01 (6 pts)

Due: 26.10.2018, Friday @17:00

deliver printed copy to my office, A433

Problem 1 [10 pts] Compilers can have a significant impact on application performance. Having a program, you will test two compilers: Compiler A(CA) has 1.0E9 instructions and 1.1s execution time. Compiler B (CB) includes 1.5E9 instructions with 1.5s execution time.

a. Assume the compiled programs run on the same machine. Having a processor with clock cycle time of 1 ns, find the average CPI for each program.

b. Assume the compiled programs run on two different processors having the same execution time. How much faster is the processor clock running the code of CA versus the processor clock running the code of CB?

c. A third compiler (CC) is developed that uses only 7.0E8 instructions and has an average CPI of 1.1. What is the speedup of using this new compiler versus using compiler A or B on the original processor?

Problem 2 [20 pts] Convert the following code to MIPS assembly. Assume *$s1* stores *i*, *$s2* store *j, $s3* stores *&arr[0]*. Comment each line of MIPS code.

```
if(i<20 || i>30)
    i+=j;
else
    j+=i;
i = i-j;
```

```
for(int i = 9; i>0; i--)
    arr[i-1] = arr[i];
```

Problem 3 [20 pts] For each instruction in the table below, you are asked to evaluate each of the code fragments. For each of the loop below, write the equivalent high-level code routine.

```
       addi $s2, $zero, 0
       addi $s1, $zero, 10
       addi $t0, $zero, 0
LOOP1: addi $t1, $zero, 10
LOOP2: addi $s2, $s2, 1
       subi $t1, $t1, 1
       bne  $t1, $zero, LOOP2
       addi $t0, $t0, 1
       bne  $t0, $s1, LOOP1
```

```
       addi $s2, $zero, 0
       addi $t0, $zero, 0
       addi $s4, $zero, 5
LOOP1: addi $t1, $zero, 0
       addi $s1, $zero, 1
LOOP2: slt  $t3, $t1, $t0
       beq  $t3, $zero, L1
       sll  $s1, $s1, 1
       addi $t1, $t1, 1
       j LOOP2
L1:    add  $s2, $s2, $s1
       addi $t0, $t0, 1
       blt  $t0, $s4, LOOP1
DONE:
```

Problem 4 [20 pts] You are given the following C program. What does it compute? Translate this function into MIPS assembly.

```
#include <stdio.h>
int main(){
    int maxnum;
    printf("Enter a number: ");
    scanf("%d",&maxnum);

    int num = 2;
    while (num<=maxnum){
        int isprime = 1;
        for(int j=2; j<=num/2; j++){
            if(num%j==0)
               isprime=0;
        }
        if (isprime)
            printf("%d ", num);
        num++;
    }
    return 0;
}
```

MIPS Assembly

```
##############################################
# Data Segment
##############################################
.data
msg:
     .asciiz      "Enter a number: "
.text
##############################################
# Main Routine
##############################################
main:
   la  $a0, msg
   jal printf_str #call function to print message
   jal scanf      #call function to read an integer
   add $s0, $v0, $zero
   # your code will come here



exit:
   addi $v0, $zero, 10  # system code for exit
   syscall              # exit main routine

##############################################
scanf:                  # $v0 contains the read int
    addi $v0, $zero, 5  # system code for read_int
    syscall             # read it
    jr $ra              # return
##############################################
printf_str:             # $a0 has string to be printed
    addi $v0, $zero, 4  # system code for print_str
    syscall             # print it
    jr $ra              # return
##############################################
printf_int:             # $a0 has integer to be printed
    addi $v0, $zero, 1  # system code to print check
    syscall             # print it
    jr $ra              # return
```

Problem 5 [30 pts] For the following code, write the equivalent recursive code using MIPS.

```
#include <stdio.h>

int recursiveSum(int arg){

    if (arg==0)
        return 0;
    else
        return recursiveSum(arg-1)+arg;
}

int main(){
    int num;
    printf("Enter a number: ");
    scanf("%d",&num);

    printf("%d ", recursiveSum(num));
    return 0;
}
```

- **You may use the MARS MIPS simulator, which is helpful for this homework.** **http://courses.missouristate.edu/kenvollmar/mars/**
- **You are required to submit the *printout copy* of your homework (including codes) by due date and time to room A433.**