COMPSCI 2GA3

Assignment/Homework 2, Oct. 5th 2021
Assignment due date: Oct. 17th, 23:59:59.
Note: Please work on this assignment individually. Students copying each other's answer will get a zero and will perform poor on midterm and final.

Written Exercises

Complete the following questions from *Computer Organization and Design: The Hardware Software Interface: Computer Organization and Design The Hardware/Software Interface: RISC-V Edition.*

Chapter 2: Instructions: Language of the Computer (18 Marks)

1. Exercise 2.14 (4 Marks) For the of instruction described by the following RISC-V fields:
   `opcode=0x33, funct3=0x0, funct7=0x20, rs2=5, rs1=7, rd=6`
   provide

   a. (1 mark) The instruction type

   b. (1 mark) Assembly language instruction

   c. (2 marks) Binary representation

   Solution:

   a. R -type

   b. `sub x6, x7, x5`

   c. `(0x40538333: 0100 0000 0101 0011 1000 0011 0011 0011)`

2. Exercise 2.29 (6 Marks) Implement the following C code in RISC-V assembly.

```
int fib(int n){
    if (n==0)
        return 0;
    else
        if (n == 1)
            return 1;
        else
            return fib(n-1) + fib(n-2);
}
```

Solution:

```
fib:
beq x10, x0, done  // If n==0, return 0
addi x5, x0, 1
beq x10, x5, done  // If n==1, return 1
addi x2, x2, -8   // Allocate stack space
sw x1, 0(x2)  // Save the return address
sw x10, 4(x2)  // Save the current n
addi x10, x10, -1  // x10 = n-1
jal x1, fib   // fib(n-1)
lw x5, 4(x2)  // Load old n from the stack
sw x10, 4(x2)  // Push fib(n-1) onto the stack
addi x10, x5, -2  // x10 = n-2
jal x1, fib  // Call fib(n-2)
lw x5, 4(x2)  // x5 = fib(n-1)
add x10, x10, x5  // x10 = fib(n-1)+fib(n-2)
// Clean up:
lw x1, 0(x2)  // Load saved return address
addi x2, x2, 8  // Pop two words from the stack
done:
jalr x0, x1
```

3. (4 Marks) Translate procedure `f` into RISC-V assembly language. The C code for `f` is:

```
long int f (long int g, long int h,
  long int i, long int j){
  return ((g+h)-i + ((g-h)+j));
}
```

Assume arguments `g, h, i, j` are in registers `x10, x11, x12, x13`. Return value should be in `x10`. Please put comments in your code.

Solution:

```
addi sp, sp, -16 # adjust stack to make room for 4 items
sw x5, 12(sp)    # save x5
sw x6, 8(sp)     # save  x6
sw x7, 4(sp)     # save x7
sw x28, 0(sp)    # save  x28
add x5, x10, x11  # x5 = g + h
sub x6, x5, x12   # x6 = (g + h) - i
sub x7, x10, x11  # x7 = g - h
add x28, x7, x13  # x28 = (g - h) + j
add x6, x6, x28   # x6 = (g + h) - i + ((g - h) + j)
addi x10, x6, 0   # copies value in x6 to x10 to return value for f
lw x28, 0(sp)    # restore  x28
lw x7, 4(sp)     # restore x7
lw x6, 8(sp)     # restore x6
lw x5, 12(sp)    # restore x5
addi sp, sp, 16  # adjust stack
jalr x0, 0(x1)   # back to calling routine
```

4. Exercise 2.40 (4 Marks) Assume that for a given program 70% of the executed instructions are arithmetic, 10% are load/store, and 20% are branch.

   a. (2 marks) Given this instruction mix and the assumption that an arithmetic instruction requires two cycles, a load/store instruction takes six cycles, and a branch instruction takes three cycles, find the average CPI.

   b. (1 mark) For a 25% improvement in performance, how many cycles, on average, may an arithmetic instruction take if load/store and branch instructions are not improved at all?

   c. (1 mark) For a 50% improvement in performance, how many cycles, on average, may an arithmetic instruction take if load/store and branch instructions are not improved at all?

a. Take the weighted average: $0.7*2 + 0.1*6 + 0.2*3 = 2.6$

b. For a 25% improvement, we must reduce the CPI to $2.6*.75 = 1.95$. Th us, we want $0.7*x + 0.1*6 + 0.2*3 <= 1.95$. Solving for x shows that the arithmetic instructions must have a CPI of at most 1.07.

   If you assume that 25% improvement reduce CPI to 80% (1/1.25 = 0.8) of its original value that is correct too, so we will accept the solution
   $2.6*.8 = 2.08$
   $0.7*x + 0.1*6 + 0.2*3 <= 2.08$
   Solving for x shows that the
   arithmetic instructions must have a CPI of at most 1.26.

c. For a 50% improvement, we must reduce the CPU to $2.6*.5 = 1.3$. Th us, we want $0.7*x + 0.1*6 + 0.2*3 <= 1.3$. Solving for x shows that the arithmetic instructions must have a CPI of at most 0.14

   If you assume that 50% improvement reduce CPI to 66% of its original vale $2.6*.66 = 1.716$
   $0.7*x + 0.1*6 + 0.2*3 <= 1.716$
   Solving for x shows that the arithmetic instructions must have a CPI of at most 0.73