Delft University of Technology
Faculty of Electrical Engineering, Mathematics and Computer Science
Computer Engineering Lab
2022-2023

Computer Architecture and Organization (CAO) EE3D11

Assignment 2

Deadline Mar 07, 2023 before 08:00!

Note:
Upload a single file having the name "YourFirstName-LastName-HW-1" to the corresponding assignment in Brightspace.
When needed, the answer should be justified. Show clearly which theory is used to find your final answer.

Exercise 1 [1pts]
Can we use registers $S0, $S1, …., $S7 in a procedure? Justify why or why not?

Exercise 2 [1pts]
Can we use registers $at, $k0 and $k1 as temporary storage in a MIPS assembly code? Justify why or why not?

Exercise 3 [2pts]
Convert the following piece of C/C++ code to MIPS. Use registers $S0, $S1, $S2 and $S3 for i, count and the base addresses of arrays A and B.

```
for(int i = 0; i < 100; i++)
        if(A[i] != B[i]) count++;
```

Exercise 4 [4pts]
Convert the following piece of C/C++ code with nested functions A and B to MIPS assembly code. Also display the content of the stack 1) before jumping to each function (before jal), 2) at the beginning of the function (after jal), and 3) after returning back from each of the functions A and B.

```
int main() {

        int i=3, j=5, n=2, m=8;

        A(i, j, n, m);

        return 0;
}
```

```
int A(int i, int j, int n, int m) {

        return B(i, n) + B(j, m);
}
```

```
int B(int i, int n) {

        return i * n;
}
```

Exercise 5 [1pts]
Convert the following MIPS assembly instructions to machine code. What is exact jump address if PC=0x98765421?

```
Slt $t0, $s1, $t8
j 4660
```

Exercise 6 [1pts]
Convert the following machine codes to MIPS assembly instructions.

```
00010110001010010000000000000101
00000001000101111000000000100111
```

Some fun activities [0pts] *(Do it for yourself. You do not need to submit.)*

- How to know if your system is big endian or little endian? In Linux you can use "lscpu | greb Endian" command, what about windows? Can you use "union" in C/C++ to figure it out yourself?
- Can you ask the compiler to convert your C/C++ code to x86 or MIPS equivalent? If so, then please test it for the code given in exercise 3.
- Can you find an emulator to go through your MIPS code step by step and see the changes of the registers and stack? If so, then test it for the code given in exercise 4.
- In MIPS green card you can see which instructions will do sign extension. But can you verify it yourself in an emulator?

Do you recall floating point representation in binary system, from Digital Systems A and/or B courses?

If no, then have a look at it. It will be helpful when we go to floating point hardware design in section in Chapter 3.