

Design of Digital Circuits: Lab Report		
Lab 7: Writing Assembly Code		
Date		Grade
Names		
		Lab session / lab room

You have to submit this report and the required codes via Moodle (<https://moodle-app2.let.ethz.ch/mod/assign/view.php?id=239263>).

Use a zip file or tarball that contains the report and the codes. Only one of the members of each group should submit. All member of the group will get the same grade.

The name of the submitted file should be *Lab7_LastName1_LastName2.zip* (or *.tar*), where *LastName1* and *LastName2* are the family names of the members of the group.

Part 1

Show an assistant that your code produces the correct sum within the MARS simulator.

Submit your code via Moodle. Include explanations/comments in your code. Include all the necessary files to run your code on the simulator. They should be in a folder called *Part_1*.

Part 2

Show an assistant that your code correctly implements the SAD algorithm within the MARS simulator.

Submit your code via Moodle. Include explanations/comments in your code. Include all the necessary files to run your code on the simulator. They should be in a folder called *Part_2*.

Part 3

Assume that instead of using black and white images, for which pixels are described with a single value, we use color images, i.e., every point is described by the combination of three colors (R,G,B).



The `abs_diff` function can be redefined as:

```
int abs_diff_color(int R1, int G1, int B1, int R2, int G2, int B2) {  
    int abs_diff = abs(R1-R2) + abs(G1-G2) + abs(B1-B2);  
    return abs_diff;  
}
```

This new function requires 6 arguments, but MIPS only provides 4 registers (\$a0-\$a3) to pass the arguments. How would you modify the function `abs_diff()` so that it adjusts to the calling convention in MIPS?

You should implement the function and submit it via Moodle. Include explanations on how the function works. In your submission, Part 3 should be in a folder called *Part_3*.