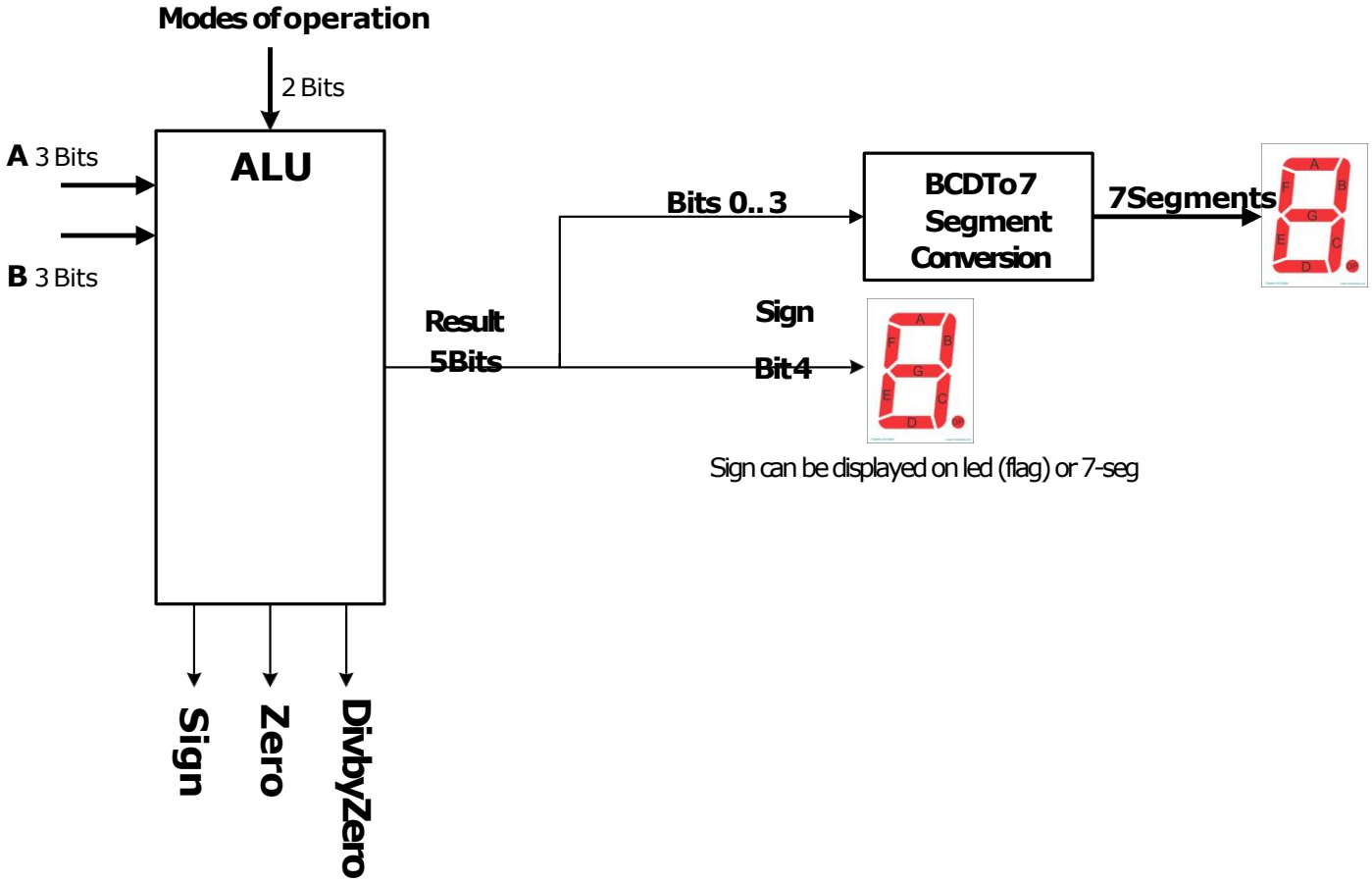
Project Description Document

Introduction

In this project, students are required to design and implement an arithmetic unit that is capable of adding, subtracting and multiplying two signed numbers, and displays the result of the operation performed along with some additional flags regarding the operation and the result.



Description

The arithmetic unit takes two 3-bits signed inputs, A and B, and an additional input called Mode of Operation, which informs the arithmetic unit which function to perform on A and B:

1. Addition $C = A + B$
   During the addition, A, B are 3-bits signed numbers. And C is a 4-bits signed number.

2. Subtraction $C = A - B$
   During the addition, A, B are 3-bits signed numbers. And C is a 4-bits signed number.

3. Multiplication $C = A \times B$

   During the multiplication, A and B are 3-bits signed numbers and C is a 5-bits signed number. The multiplication of 2-bits by 2-bits yields a result of 4-bits, therefore C is composed of 4-bits for the value and 1-bit for the sign.
   The multiplication of $3 \times 3 = 9$, which in binary is $(11)_2 \times (11)_2 = (1001)_2$

   Remainder: Result = A % B
   During the division A, B and Result are 3-bits signed numbers.
   A % 0 is forbidden and must output 0 and Div by Zero Flag must be asserted
   A % B, has the same sign as A.

   For example: +2%+3 =+2, +2%-3 =+2, -2%+3=-2, -2%-3=-2

Flags

- Sign Flag:
- The sign flag indicates if the result is negative. The flag is set to 1 if the result is negative and 0 otherwise.

- Zero Flag:
- The zero flag indicates if the result is zero. The flag is set to 1 if the result is zero and 0 otherwise.

- Div by Zero Flag:
- The divide by zero flag indicates if we divide by zero. The flag is set to 1 if B operand equal zero in division operation and 0 otherwise.

Notes

- The outputs: Result number should be displayed on a seven segment and the flags should be displayed on leds, sign can be displayed on either.
  - The input can be entered through switches.
- You can use Full adder, half adder, multiplexor or any other ready IC instead of making its hardware.
- IC number 7447 can be used to convert from BCD to seven segments, you can use it in converting result to be displayed on seven segments.

Team Formation

Each team should consist of **4 members of the same tutorial**, that leaves 3 members one from each tutorial they will end up in the same team with a reduced load of not having to implement the remainder circuit.

Deliverables

- For each team member: Name, ID and work done in the project.
- Integrated Circuit.
- Inputs and Outputs.
- All verilog files.
  - Grading is automated so strictly adhere to the description.
  - Alu.v is your top module that integrates all the circuits together
    Make sure to have only 4 ports (A, B, S, R) and 3 Additional bits (SF, ZF, DZF) for the flags with these exact names.
  - A Verilog file for each module.
  - A testbench file for each module to test your modules separately over all test cases from in the order [-3, -2, -1, 0, 1, 2, 3] and save the output using the functions `$fopen, $fclose` of Verilog into a text file mult.txt, add.txt, sub.txt and rem.txt (the ground truth shall be posted)
    e.g. from A = -3, B = -3 to A = 3, B = 3
  - Files should be zipped like this.

```
<tutorial_id>._<team_id>.zip
|
|
|_______ alu.v
|_______ add_sub.v
|_______ mul.v
|_______ rem.v
|_______ add_sub_tb.v
|_______ mul_tb.v
|_______ rem_tb.v
```

- Demo video (especially for hardware when it works just in case).

Deadline

1. The simulation will be delivered on 05/05
2. HW will be delivered on 12/05
3. The discussion will be in the tutorial time
4. Each group consist of 4 members from the same lab tutorial.
   a) Two of them is responsible for Addition and subtraction operation
   b) one implement multiplication, BCD to seven segments (use 7447 or 7448)
   c) one implement remainder
   d) All the team members cooperate in integration of different components
   e) You need to make the design hierarchy by implement the different circuits in separate files to facilitate the debugging operation
   f) If the integration does not work, then the team will lose the integration grade only and **each member will be graded upon his part in the project.**

The deadline to join your team is 15/04/2024