EECE 320 – Digital Systems Design

Project - Combinational

Three-digit BCD Adder

1. Introduction

In this project, you will design and test a three-digit binary-coded-decimal (BCD) adder capable of adding positive and negative BCD numbers. The project has a combinational part and a sequential part. You will implement it in two phases; first the combinational part and next the sequential part.

2. Project Behavioral Specification

- The BCD adder you design must be capable of adding two three-digit BCD operands to produce a three-digit sum. Each of the two operands can be either positive or negative, so the adder must, in effect, be capable of both addition and subtraction.
- You will represent the operands and sum using 13-bit binary vectors where 12 of the bits are the three 4-bit BCD digits and the 13^{th} bit is a sign bit (0 if the number is positive and 1 if it is negative). This is essentially a sign-plus-magnitude representation where the magnitude is represented with the BCD code.
- The adder should also have a one-bit overflow output signal that is 1 if the addition results in an overflow (a magnitude that is too big to be represented with 3 BCD digits) and 0 otherwise. *BCD codes and arithmetic are discussed in the textbook*.

3. Project Organization

- *One-Digit Adder*

  One way to implement the digit adder is by first doing a binary addition/subtraction and then correcting the sum, if it is outside the range of valid BCD digits (i.e., greater than 9). The correction can be implemented by adding a correction value to the intermediate result produced by the first addition or subtraction.

  - You can implement this as a ripple carry 4-bit binary adder/subtractor to do the first addition/subtraction and another binary adder to perform the correction.
  - You must determine when to do the corrections and what correction values to add. You will need logic to generate the correction value.

Next, you combine three copies of the digit adder using a structural VHDL module that uses the other modules as components.

- *Three-Digit Adder*

  The adder is to be organized as three identical one-digit adders connected in a ripple-carry configuration. Each of the digit-adders will be capable of adding two 4-bit BCD digits and a 1-bit carry/borrow input to produce a 4-bit BCD digit sum and 1-bit carry/borrow output. The adder should implement the following general algorithm:

  1. The adder checks the signs and relative magnitude of the two operands.
  2. If signs are the same, the adder adds the operands and makes the sign of the result equal the signs of the operands.
  3. If the signs differ, then the adder compares the magnitudes of the operands, subtracts the smaller from the larger, and sets the sign of the result equal to the sign of the larger. Note that if the result is 0, the sign of the result must be positive (0 sign bit).

4. Design and Implementation

In this phase, you will develop an algorithm for BCD addition and subtraction and write a ***behavioral*** VHDL module to implement this algorithm. Test the algorithm for correctness by simulating the VHDL module. You should use high-level VHDL operations such as ***processes***, ***conditional operations (IF, CASE, etc.***), and ***comments*** to make it easy for the reader to understand your algorithm.

Note: You are not supposed to use any available integer operations such as addition (+), comparison (>, <) etc.

5. Testing

It is necessary to select a small subset of test signals that are likely to catch most of the possible errors. Part of this project is to decide how to test it (i.e., which test inputs to use).

While there is very little theory to help you select test inputs, there are some guidelines you should consider. For example:

1. Test the digit-adder by itself to make sure it is working before using it in the full adder. *(0+0; 0+1; 2+3; 4+5; 5+6; 9+9)*
2. Find test signals for the full adder that generate carry signals between stages and check all the extreme cases and any special cases.

   (*000+000; 000+001; 123+345; 234+567; 789+777; 999+999*)

   (*000+(-000)); 000+(-001); 345+(-123); 567+(-234); 777+(-789); 999+(-999))*

6. Deliverables

Upload to Moodle your (.vhd) files only of all the components that you designed in addition to the testbench file only for the overall adder. This testbench file will have the test cases that you selected.

A high-level block diagram for the project is given in Figure 1.

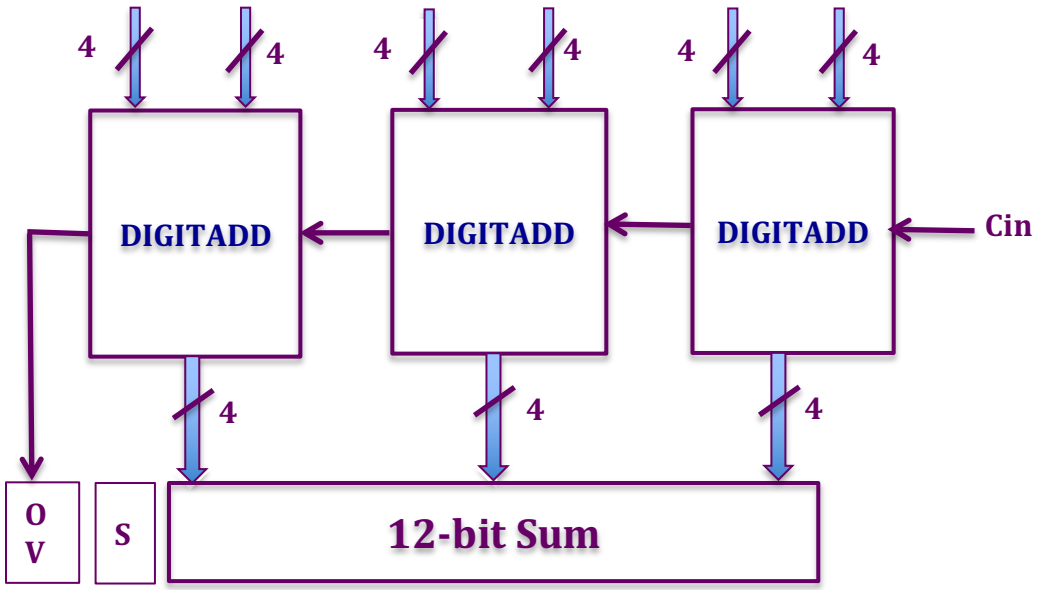

Figure 1: Three-digit BCD Adder High Level Block Diagram