1. Scan of completed “Lab2 Submission Form” as the report front page. Don’t forget to fill out the “total logic elements” and the Worst Case Speed (ns) tpd in the form.

2. Implementation procedure, design decisions, encountered problems or bugs with solution to them, debugging

techniques and RTL view of your circuit (2 pages max).

Required to implement a simple calculator with 4 pre-defined operations: AND, OR, XOR, ADD. The type of operation is specified by a 2-bit input signal using switches (17:16), 00 for AND, 01 for OR, 10 for XOR, and 11 for ADD. Operand 1 and Operand 2 are both 8-bit input signals, defined by switches (7:0) and switches (15:8), respectively. The hexadecimal values of the operands are each displayed on a pair of seven-segment blocks, hex5, hex4 and hex7, hex6, respectively. The corresponding 2-bit value of the operator is represented with two red LEDs. The result of the operation is represented on the three seven segment displays (hex2, hex1, hex0). Since the addition of two 8-bit numbers can cause overflow, hex2 can display 1 when needed, or be blank otherwise.

3. Fully commented VHDL code.

4. Functional simulation waveform with coverage for critical cases. Mark your simulation waveforms explaining several different scenarios for each operation. You must prove that what you have designed is working. For instance, testing with the inputs set to 0 does not allow one to distinguish one operation from another or if any operation in particular is fully working. Often one tests critical cases - the limits where things may break (i.e. overflow). Do not print waveforms of all possible cases - give samples for critical cases of each operation and explain how you checked for proper operation. The goal is to have a simulation to prove that the circuit works; without doing a full exhaustive test of all possible inputs. It should prove that all operations work correctly for enough numbers to give confidence that the circuit is fully functional.