BCD Incrementor Circuit

Signature and Grading Sheet

Group #:_________   Name(s):____ .

Signature

Section 4.2(f): _____________________.

Grading

- Section 4.1(a): VHDL code (30 points):______.
  Attach code printout
- Section 4.1(b) : Simulation (20 points):______.
  Attach simulation timing diagram printout.
- Section 4.1 (c) (5 points):______.
  Attach one-page report printout with # of LEs circled.
- Section 4.1 (d) (15 points):______.
  Attach one-page schematic diagram.
- Section 4.2(b) : VHDL code (10 points):______.
  Attach code printout.
- Section 4.2(f) demo signature (20 points):______.

Total points: __________.

Experiment
BCD Incrementor Circuit

1 Purpose

To design a intermediate-sized combinational circuit

2 Reading

Chapter 4 of *Embedded SoPC Design with Nios II Processor and VHDL Examples*

3 Project specification

The binary-coded-decimal (BCD) format uses 4 bits to represent 10 decimal digits. For example, 259 is represented as "0010 0101 1001" in BCD format. A BCD incrementor adds 1 to a number in BCD format. For example, after incrementing, "0010 0101 1001" (i.e., 259) becomes "0010 0110 0000" (i.e., 260). We want to design the circuit and display the results on three 7-segment LED displays. No VHDL process is allowed.

The input and output of the incrementor are

- input:
  - `b2, b1, b0`: three 4-bit inputs representing 3 BCD digits and `b2` is the most significant digit.
- output
  - `y2, y1, y0`: three 4-bit outputs representing 3 incremented BCD digits and `y2` is the most significant digit.

4 Design Procedures

4.1 BCD incrementor synthesis

(a) Design a three-digit BCD incrementor. The entity declaration of this design is

```
entity bcd_inc is
    port(
         b2, b1, b0: in std_logic_vector(3 downto 0);
         y2, y1, y0: out std_logic_vector(3 downto 0)
    );
END bcd_inc;
```

Derive the architecture body. No VHDL process is allowed. The VHDL code must be properly documented.

(b) Compile the design and perform simulation to verify its operation. In the timing diagram, the input and output signals should be clearly arranged and represented in proper format so that the simulation result can be easily understood. Use at least 15 input test patterns, including 000, 123, 109, 199, and 999.

(c) Look at the compiling report and determine the number of LEs used in circuit.

(d) Derive the conceptual diagram of your VHDL code (similar to Figure 4.3 in book).

4.2 Implementation and testing

(a) Combine the BCD incrementor and three hex digit to seven-segment LED decoders so that the incrementor's output can be shown on three seven-segment LED displays.

(b) Derive the VHDL code for the combined circuit. You can either cut-and-paste the previous codes or create a top-level VHDL file with component instantiation.

(c) Use 8 switches for the `b1` and `b0` and 4 pushbuttons for `b2`.

(d) Perform pin assignment and synthesize the code.

(e) Download the file to the FPGA device and verify the operation of physical circuit. Note that the pushbutton's output is '1' when not pressed.

(f) Demonstrate the circuit to instructor and get signature.