EGCP281 Assignment #5

A. Objectives:

- Using VHDL to design real-life application sub-systems
- Understand the basics of bit error detection mechanism used in digital applications
- Data paths

B. Submission:

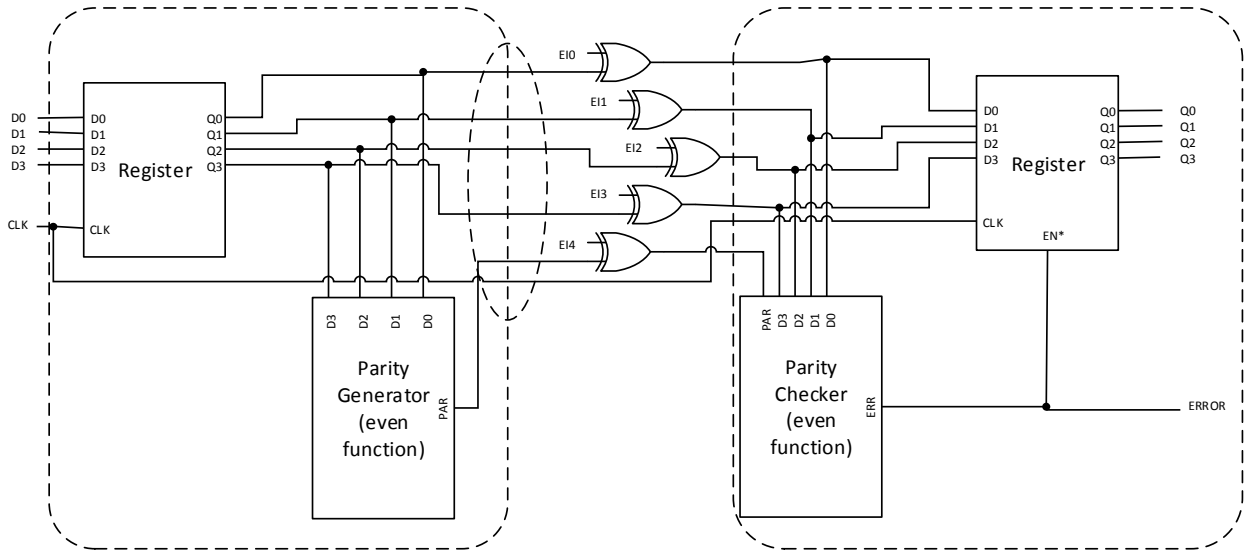
Submit one zip file containing only two files: VHDL source code (assignment5.vhd) and the report (assignment5.pdf). Submit in Titanium before the deadline.

The report assignment5.pdf file shall include the readable VHDL source code as well as all the relevant simulation screenshots and answers to the requested questions in the following paragraphs.

C. Introduction:

When data is sent from point A to point B there is a probability that error occurs. For example a 0 binary bit may be turned into a binary 1 or vice-versa due to nearby noises and spikes. It is therefore important to implement a simple means to detect if error had happened or not. In this assignment we investigate a basic error detection circuit using parity between a sending and a receiving module.

The block diagram for our experiment is shown below:



To simplify we use only 4 bits data path. This setup ensures only good 4-bit data are forwarded to the receiver side upon the rising-edge of the clock. If an error occurred the bad data is not forwarded. In other words the receiving side receive-register only contains error-free data.

A parity generator is used on the transmitting side to generate a parity bit to be sent along with the 4-bit data. We use even parity scheme which means the 4-bit data together with the parity bit are always supposed to have even number of 1's.

A parity checker is used on the receiving side to check if the 5-bits received are even parity (having an even number of 1's). If there is no error the data is forwarded to the output Q vector.

We inserted 5 XOR gates to allow us to inject errors into the transmission line for testing the operation of the circuits.

As you may expect the above circuit can only detect 1-bit errors. If there were an even number of bits in error the circuit will not be able to flag the error.

D. Tasks:

a. Complete the truth table below. This table specifies all the test vectors and expected results that we need to use in the test bench step for this assignment. Note that the output vector Q shown in the truth table is valid upon the following rising-edge of the clock signal AND if there was no error. If there was error the Q vector is not updated.

D3	D2	D1	D0	PAR	EI4	EI3	EI2	EI1	EI0		ERROR	Q3	Q2	Q1	Q0
1	0	0	0		0	0	0	0	0						
0	1	0	0		0	0	0	0	0						
0	0	1	0		0	0	0	0	0						
0	0	0	1		0	0	0	0	0						
1	0	0	0		1	0	0	0	0						
0	1	0	0		0	1	0	0	0						
0	0	1	0		0	0	1	0	0						
0	0	0	1		0	0	0	1	0						
1	1	1	1		0	0	0	0	1						

b. Design in VHDL the clock circuit (complete with entity and declaration) as a component. The output clock pulse has a duty cycle time of 50ns and a period of 100ns. This clock signal will be connected to the CLK signal of the system under test.
c. Design in VHDL the sender register (complete with entity and declaration) as a component
d. Design in VHDL the receiver register (complete with entity and declaration) as a component. Note that the receiver register has an EN* input which means it only loads the input data when EN* signal is low (binary 0).
e. Design in VHDL the error injection block (complete with entity and declaration) as a component
f. Design in VHDL the parity generator (complete with entity and declaration) as a component
g. Design in VHDL the parity checker (complete with entity and declaration) as a component. Note that the output signal ERR is low (binary 0) if there were no error.

h. Using structural design style integrate and connect all above components to form the system under test having:
    i. inputs: vector D(3 downto 0), CLK, vector EI(4 downto 0)
    ii. outputs: vector Q(3 downto 0), ERROR

    Note: the CLK signal of the system is connected to the output of the clock generator in the test bench.

i. Design a test bench to test the system. The test bench should behave as described below:
    i. Report a message “test begins…” at the beginning of the test
    ii. Report a message “test ends…” at the end of the test
    iii. For each test vector assert the expected output results conditions and report a message “test failed on vector <specific vector value>” if the simulated outputs do not match with the expected results
    iv. If all test vectors passed, report message “All test vectors passed.”
    v. Test vectors shall cover all binary combinations (D vector and EI vector) listed in the Truth Table in step (a).
    vi. Use For loop in test bench to cover large number of test vectors
j. Capture screenshots for:
    i. Required messages above
    ii. Wave simulation for each test vector.

E. Conclusions:
    a. Highlights what have you learned from this experiment?
    b. Describe a challenge you ran into and how you resolve it in order to complete this assignment.