**Computer Hardware Experiments**

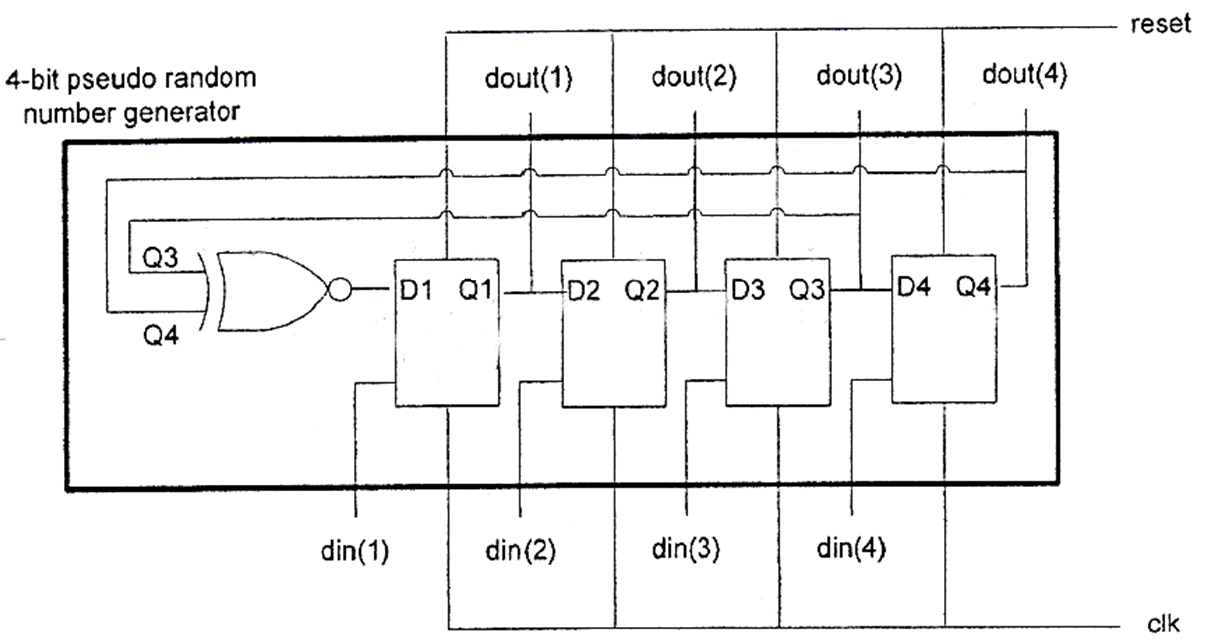
Lab\_08: Pseudo Random Number Generator

**Name : Student\_ID:**

Purpose:

To understand the mechanism of pseudo random number generator.

You are going to implement a pseudo random number generator which generates a random number every clock cycle. The following figure shows a 4-bit random number generator. At each clock cycle, a 4-bit random number is generator on *dout*. “*reset*” is an asynchronous reset, when *reset* = 1, *dout* = *din*.



To implement the above 4-bit random number generator in VHDL, we can use the following entity:

entity myrandom is

port (

clk : in std\_logic;

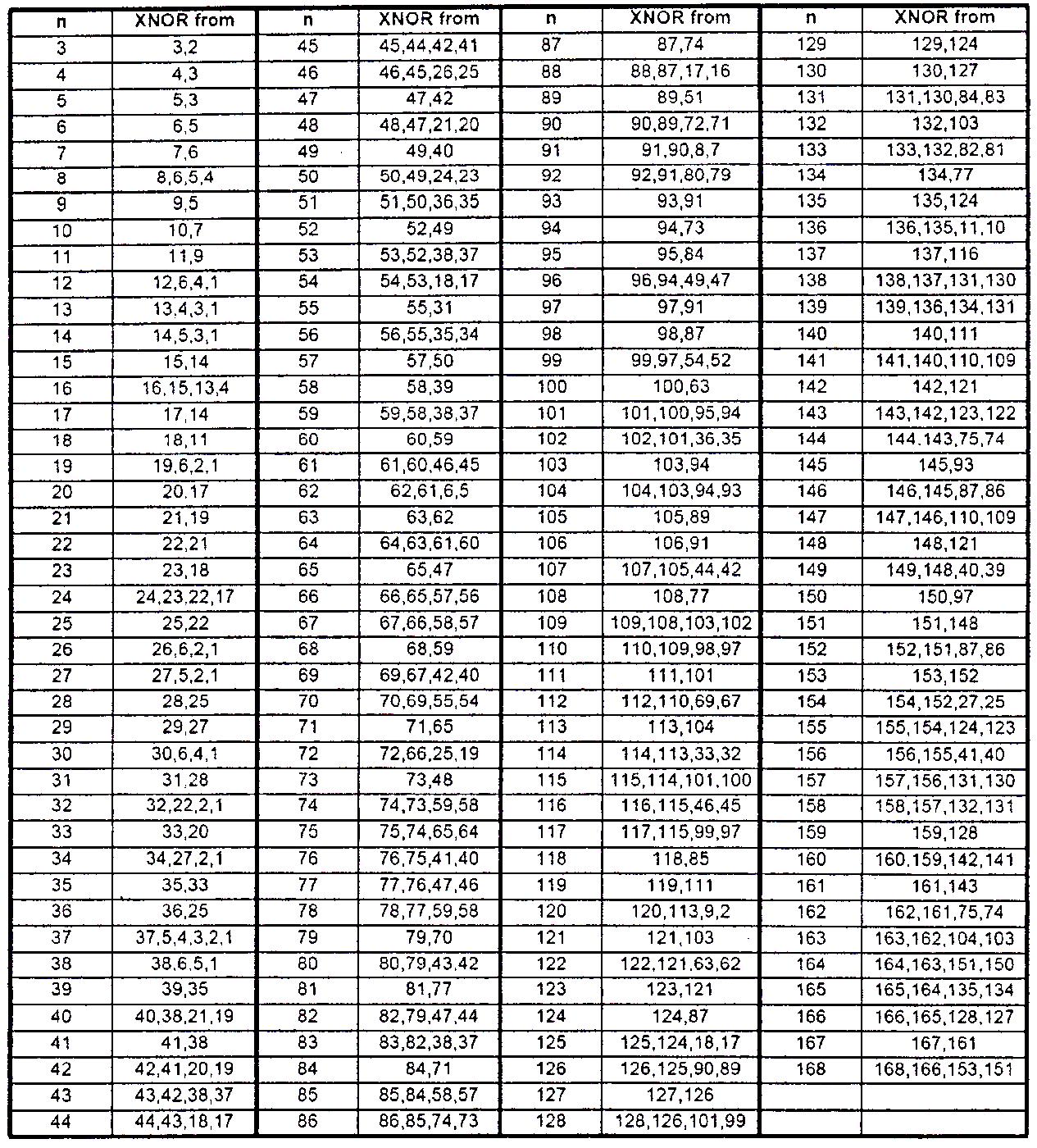
reset : in std\_logic;

din : in std\_logic\_vector(4 downto 1);

dout : out std\_logic\_vector(4 downto 1) );

end myrandom;

From the above figure, you can see that a 4-bit random number generator requires 4 FFs(flipflops) and 1 XNOR gate with two inputs. Actually, this circuit can be MODIFIED to implement an *n*-bit random number generator. For example, to construct a 8-bit random number generator, we need 8 FFs and 1 XNOR, the width of *din* and *dout* are 8 bits, i.e. std\_logic\_vector(8 down to 1). Instead of connecting Q3 and Q4 to the XNOR gate in the 4-bit generator, now we need to connect Q4, Q5, Q6 and Q8 to the XNOR, which means the XNOR in the 8-bit generator has 4 inputs. As a result, to construct an *n*-bit generator, we need *n* FFs and 1 XNOR, the number of inputs of the XNOR gate depends on how many Qs will be connected. The following table shows which Qs must be connected to the XNOR for different generators (different *n*).



Each student is required to implement an individual generator that depends on your student number. The generator you need to implement is:

n = student number

For example, if your student number is 43 (ask the teacher for your student number), then you need to implement a 43-bit pseudo random number generator, and the entity for your design will be as follows:

entity myrandom is

port (

clk : in std\_logic;

reset: in std\_logic;

din : in std\_logic\_vector(43 downto 1);

dout : out std\_logic\_vector(43 downto 1) );

end myrandom;

Change the width of *din* and *dout* based on which generator you are going to implement. Download the test bench file lab06\_test.vhd from the FTP server, modify the *din* and *dout* size, simulate your design and demonstrate you result.