Integrated Circuits Design by FPGA

م.م. أحمد مؤيد عبدالحسين
جامعة الفرات الأوسط التقنية / الكلية التقنية الهندسية / نجف

Lecture 3 :

Part 1 : Using the FPGA board

Objectives of this Lecture

- **To learn how to implement a design on an FPGA board.**

Contents of this Lecture

- **Implementation of a Shift Register design on an FPGA board.**

Implementation of a Shift Register on FPGA board

- Shift Registers are used for data storage or for the movement of data.
- Therefore commonly used inside calculators or computers to store data as binary numbers.
- Convert the data from either a serial to parallel or parallel to serial format.

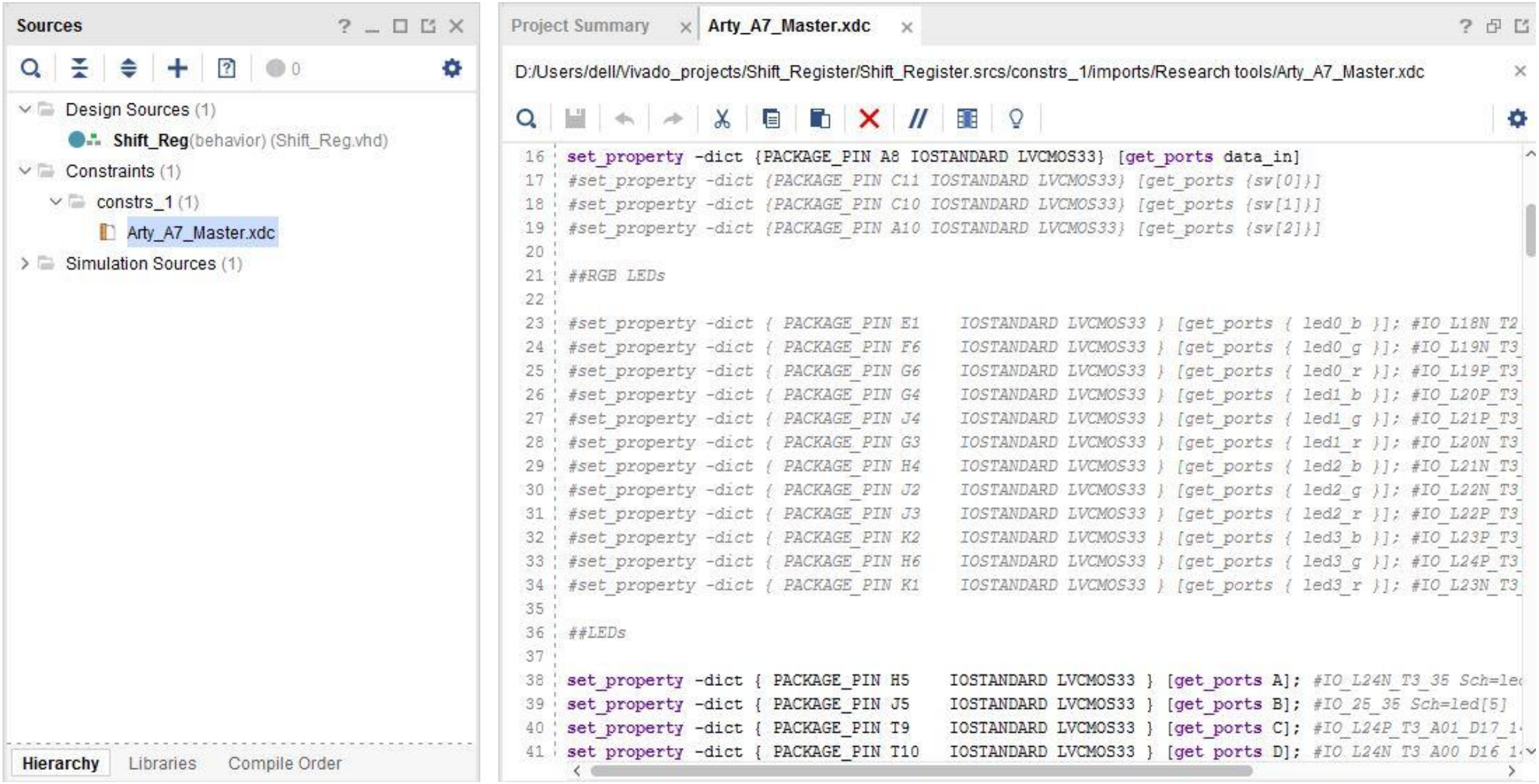
What is a shift register ?

Implementation of a Shift Register on FPGA board

```vhdl
-- Library's
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.numeric_std.all;

-- Entity Declaration
entity Shift_Reg is
port (
    A               : out std_logic;    -- single bit in register
    B               : out std_logic;    -- single bit in register
    C               : out std_logic;    -- single bit in register
    D               : out std_logic;    -- single bit in register
    data_in         : in std_logic;     -- data input (1 or 0)
    reset           : in std_logic;     -- when this signal goes high clea
                                        -- all bit values on a clock cycle
    clk             : in std_logic);    -- input clock
end Shift_Reg;

-- Architecture Body
architecture behavior of Shift_Reg is

-- Defined Signals used in Architecture
signal A_reg, B_reg : std_logic := '0';
signal C_reg, D_reg : std_logic := '0';
```

```vhdl
-- Begin Architecture
begin

    -- Signal Assignments
    A <= A_reg;
    B <= B_reg;
    C <= C_reg;
    D <= D_reg;

    -- Process that is used to shift values
    --          ** HINT **
    -- (We want this process to be evaluated
    -- on every clock cycle)
    reg_process: process(clk)
    begin
        if(rising_edge(clk)) then
            if(reset = '1') then
                A_reg <= '0';
                B_reg <= '0';
                C_reg <= '0';
                D_reg <= '0';
            else
                --                  ** HINT **
                -- This is where the shifting actually occurs
                -- depending on how you code this, you can have
                -- a shift right or shift left register
                A_reg <= data_in;
                B_reg <= A_reg;
                C_reg <= B_reg;
                D_reg <= C_reg;
            end if;
        end if;
    end process reg_process;
end behavior;
```

Implementation of a Shift Register on FPGA board



Lecture 3

Lecture 3 :

Part 2 : VHDL Test Bench

Objectives of this Lecture

- **To define and introduce a basic Test Bench.**
- **To learn how to construct a Test Bench.**

Contents of this Lecture

- **What is a Test Bench?**
- **Test Bench Outline**
- **Shift Register Example**

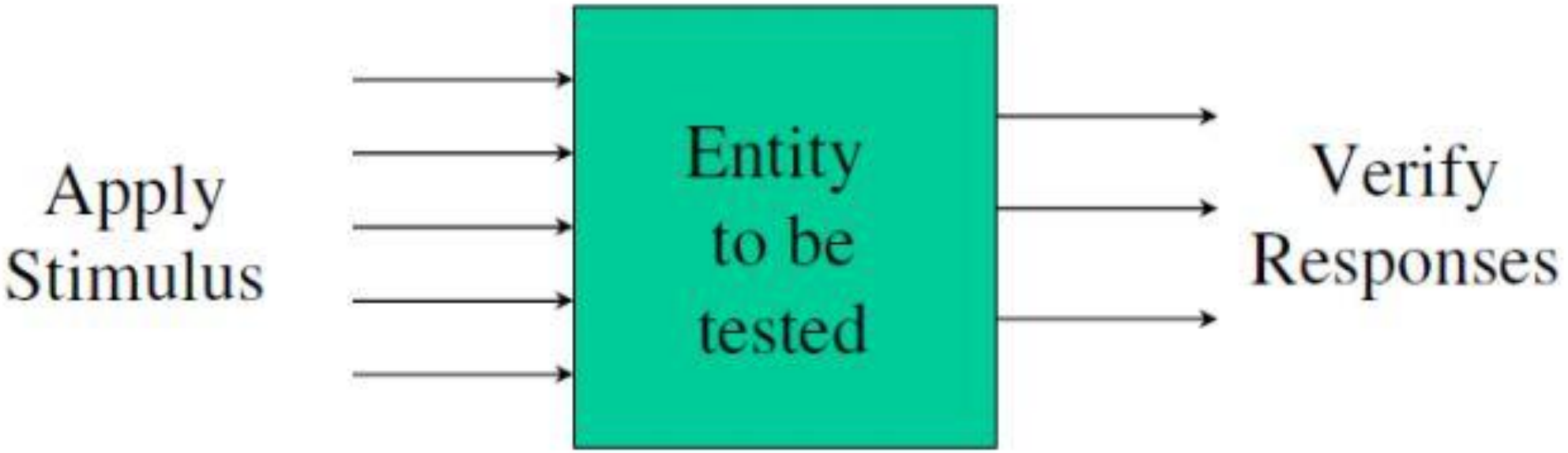
What is a Test Bench?

- A test bench model is used to exercise and verify that your VHDL design is working correctly as expected.
- A test bench **cannot** be synthesized and placed onto an FPGA board.
- You can code the test benches using Notepad++ , or Vivado IDE.
- Vivado IDE is used to execute the simulation.

What is a Test Bench?

- Test Benches consist of:
  - Entity
    - Contains no ports or generics
  - Architecture
    - Declares, instantiates, and wires together the driver model and the model under test
    - The driver model provides stimulus and verifies the response of the model under test

What is a Test Bench?

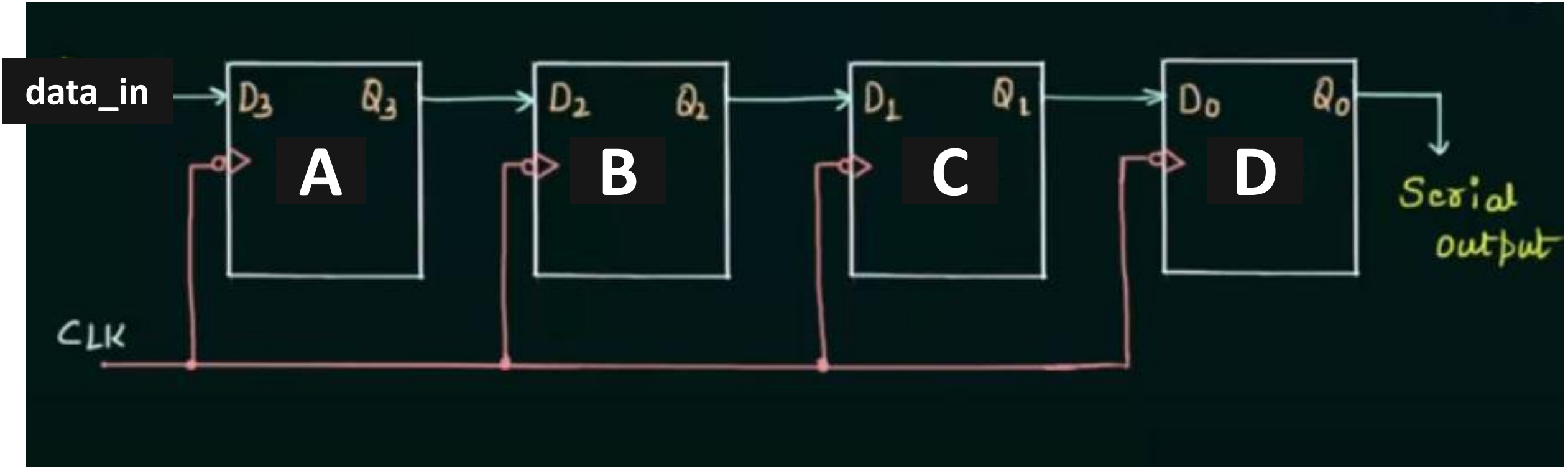


Test Bench Outline

```vhdl
--Libraries
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.numeric_std.all;

--these two libraries are only
--supprtted in Test Bench (these
--two libraries will not be
-- synthesized)
use std.textio.all ;
use ieee.std_logic_textio.all ;

-- Entity
entity Test_Bench_name is
end;
```

```vhdl
-- Architecture
architecture testing of Test_Bench_name is
-- intantiate component of the VHDL entity you
-- are testing
component LFSR3
port (
    clk : in STD_LOGIC;
    outp: out STD_LOGIC_VECTOR ( 2 downto 0 ));
end component;

-- Simulation signals
signal clk_sim              : std_logic := '0';
signal outp_sim             : std_logic_vector(2 downto 0);

begin
    -- Sometimes this is called UUT (Unit Under Test)
    dev_to_test:  LFSR3
        port map(clk_sim, outp_sim);

    -- Simulate the input clock to your design
    clk_proc : process
        begin
        wait for 50 ns;
        clk_sim <= not clk_sim;
    end process clk_proc;

end testing;
```

Shift Register Example

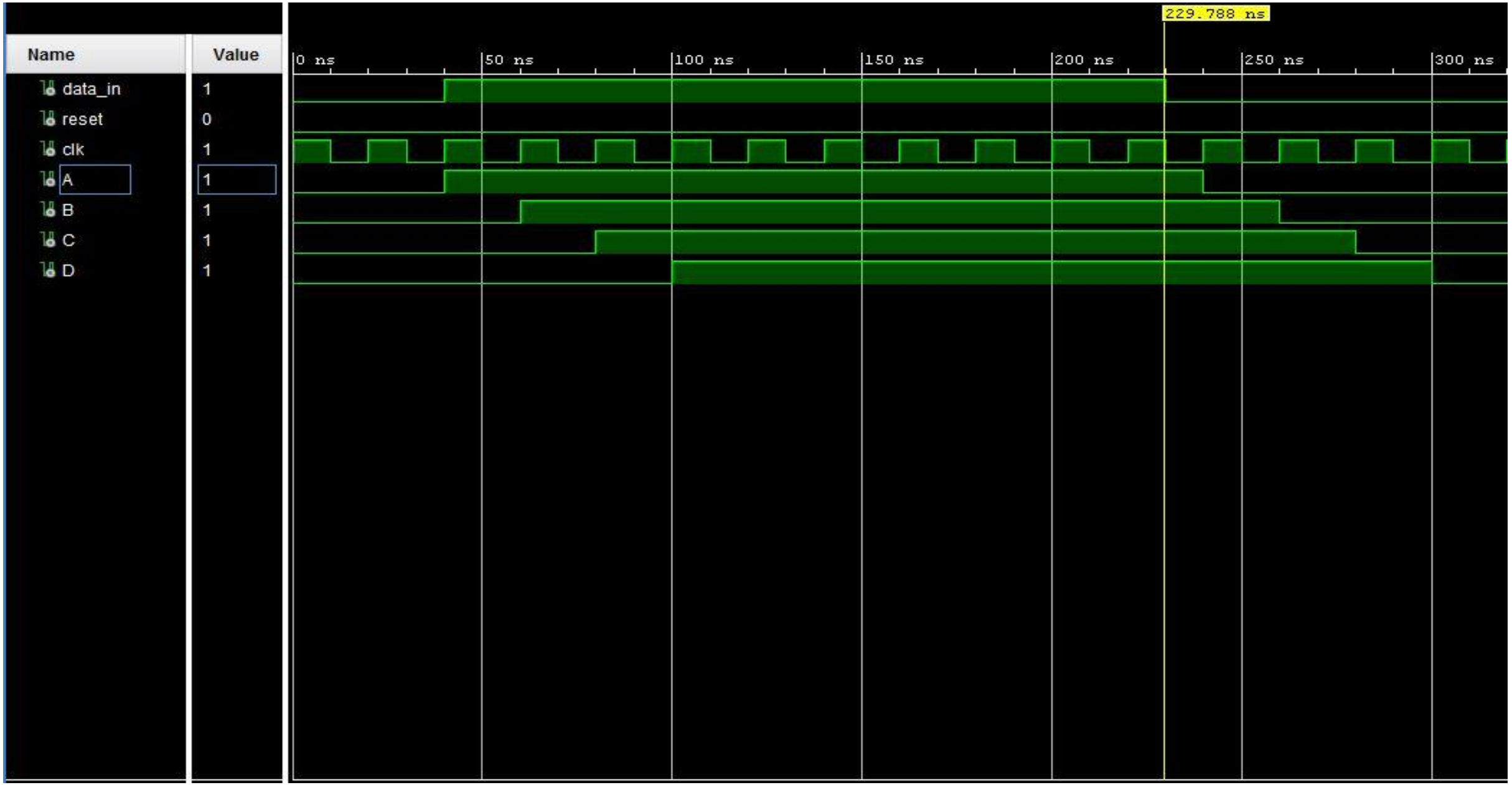


Shift Register Example

```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.numeric_std.all;

entity test_Shift_Reg is
end;

architecture test of test_Shift_Reg is

component Shift_Reg
port (A:        out std_logic;
      B:        out std_logic;
      C:        out std_logic;
      D:        out std_logic;
      data_in:  in  std_logic;
      reset:        in  std_logic;
      clk:      in  std_logic);
end component;

signal data_in : std_logic := '0';
signal reset : std_logic := '0';
signal clk : std_logic := '1';
signal A, B, C, D: std_logic;

```

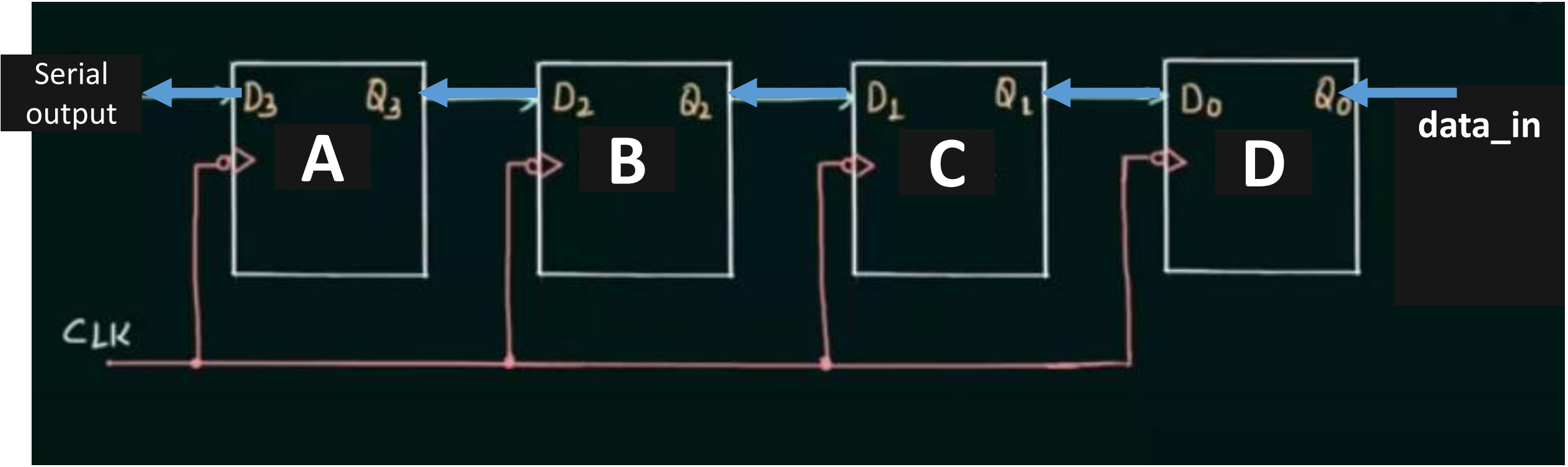
```vhdl
begin

    dev_to_test:  shift_reg
        port map(A, B, C, D, data_in, reset, clk);


    clk_stimulus:  process
    begin
        wait for 10 ns;
        clk <= not clk;
    end process clk_stimulus;

    data_stimulus:  process
    begin
        wait for 40 ns;
        data_in <= not data_in;
        wait for 150 ns;
    end process data_stimulus;

end test;
```

Shift Register Example



Assignment

- For the same VHDL code of shift register in this lecture. How you can make the shifting process in the **left** direction?



End of lecture 3

Any Questions ?