# Digital Design Laboratory

Dr. Oniga István
University of Debrecen, Faculty of Informatics

This work was supported by the construction EFOP-3.4.3-16-2016-00021.
The project was supported by the European Union, co-financed by the European Social Fund.

# 7. Laboratory assignments

- Simple sequential logic circuits
  - D latches
  - D flip-flops
  - T flip-flops

# Lab7_1a: D latch

- Create a new HDL project (Lab7_1)
- Add a new Verilog source file Lab7_1.v.
- Describe a D latch behavior using Verilog language
- Simulate the circuit using stimulus signals described lower.

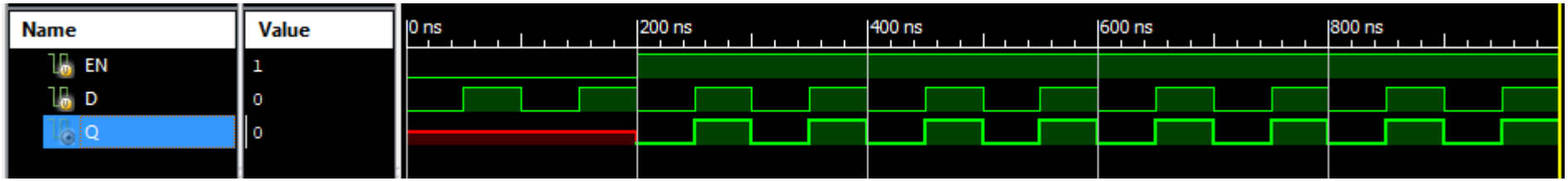
D-latch Verilog code

```
module v_Dlatch_G (input EN, D, output reg Q);
    always @(EN or D)
        begin
                if (EN)
                        Q = D;
end endmodule
```

Stimulus signal

```
    // Add stimulus here
    #100;
    EN = 1;

end
always #50
    D <= ~ D;
```



# Lab7_2a-d: Flip-Flops

- Create a new project (Lab7_2)
- Add a new VERILOG source (Lab7_2.v).
- Use the Verilog description below (Verilog codes could be downloaded from laboratory website).
- Simulate the circuit using a Verilog test fixture .

```
//
// Flip-Flop with Positive-Edge Clock
//

module v_registers_1 (input C, D, output reg Q);

    always @(posedge C)
    begin
        Q <= D;
    end

endmodule
```

# Lab7_2a-d: Flip-Flops

- Create a new project (Lab7_2)
- Add a new VERILOG source (Lab7_2.v).
- Use the Verilog description below (Verilog codes could be downloaded from laboratory website).
- Simulate the circuit using a Verilog test fixture .

```
//
// Flip-Flop with Negative-Edge Clock and Asynchronous Clear
//

module v_registers_2 (input C, D, CLR, output reg Q);

    always @(negedge C or posedge CLR)
    begin
        if (CLR)
            Q <= 1'b0;
        else
            Q <= D;
    end

endmodule
```

# Lab7_2a-d: Flip-Flops

- Create a new project (Lab7_2)
- Add a new VERILOG source (Lab7_2.v).
- Use the Verilog description below (Verilog codes could be downloaded from laboratory website).
- Simulate the circuit using a Verilog test fixture .

```
//
// Flip-Flop with Positive-Edge Clock and Clock Enable
//

module v_registers_4 (input C, D, CE, output reg Q);

    always @(posedge C)
    begin
        if (CE)
            Q <= D;
    end

endmodule
```

# Lab7_2a-d: Flip-Flops

- Create a new project (Lab7_2)
- Add a new VERILOG source (Lab7_2.v).
- Use the Verilog description below (Verilog codes could be downloaded from laboratory website).
- Simulate the circuit using a Verilog test fixture .

```
//
// Flip-Flop with Positive-Edge Clock and Synchronous Set
//

module v_registers_3 (input C, D, S, output reg Q);

    always @(posedge C)
    begin
        if (S)
            Q <= 1'b1;
        else
            Q <= D;
        end

endmodule
```

# Lab7_3a: T-FF with asynchronous reset

- Create a new project (Lab7_3)
- Add a new VERILOG source (Lab7_3a.v).
- Use the Verilog description below for a T-FF with asynchronous reset (Verilog codes could be downloaded from laboratory website).
- Simulate the circuit using a Verilog test fixture .

```
//-------------------------------------------------------
// T flip-flop async reset
//-------------------------------------------------------
module tff_async_reset ( input data, clk, reset, output reg q);

always @ ( posedge clk or negedge reset)
if (~reset) begin
  q <= 1'b0;
end else if (data) begin
  q <= !q;
end

endmodule
```

# Lab7_3b: T-FF with synchronous reset

- Create a new project (Lab7_3)
- Add a new VERILOG source (Lab7_3a.v).
- Use the Verilog description below for a T-FF with synchronous reset (Verilog codes could be downloaded from laboratory website).
- Simulate the circuit using a Verilog test fixture .

```
//-------------------------------------------------------
// T flip-flop sync reset
//-------------------------------------------------------
module tff_sync_reset ( input data, clk, reset, output reg q);

always @ ( posedge clk)
if (~reset) begin
  q <= 1'b0;
end else if (data) begin
  q <= !q;
end

endmodule
```