ESD - Elettronica dei Sistemi Digitali

Exercises on Sequential Circuits

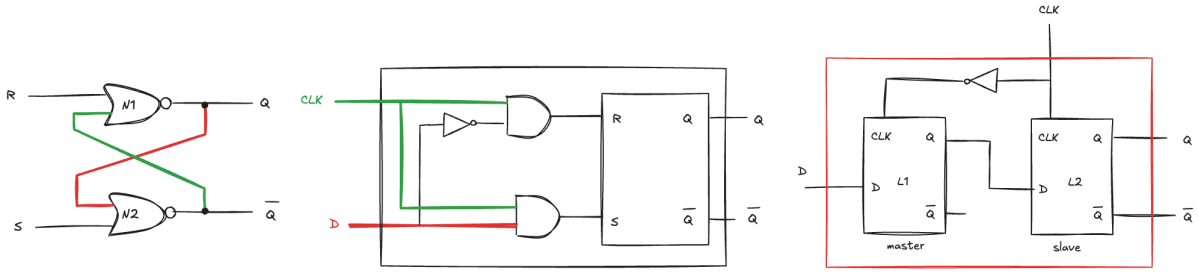
Riccardo Berta

2025.11.24

1 Latch and Flip-Flop Exercises

1.1 Exercise 1

Implement a D flip-flop in DEEDS and simulate its behavior using a timing diagram that shows it is transparent only at the rising edge of the clock and that, at all other times, it retains its previously stored value. Then use this circuit to create a reusable block that can be incorporated into other designs:



1.2 Exercise 2

Consider the following behavioral descriptions of different types of flip-flops. For each type, design a possible implementation, build it in DEEDS, verify its operation using a timing diagra and write the VHDL code that describes its behavior.

1.2.1 (a) Toggle Flip-Flop

The toggle (T) flip-flop inverts its output on every active clock edge when T is high, producing a clean alternating sequence that effectively divides the clock frequency by two.

1.2.2 (b) JK Flip-Flop

A JK flip-flop receives a clock and two inputs, J and K. It updates its output on each clock edge by setting Q when J=1 and K=0, resetting it when J=0 and K=1, holding its value when both inputs are 0, and toggling when J and K are both 1, making it a versatile device that can act as a set/reset flip-flop, a memory cell, or a toggle depending on its inputs.

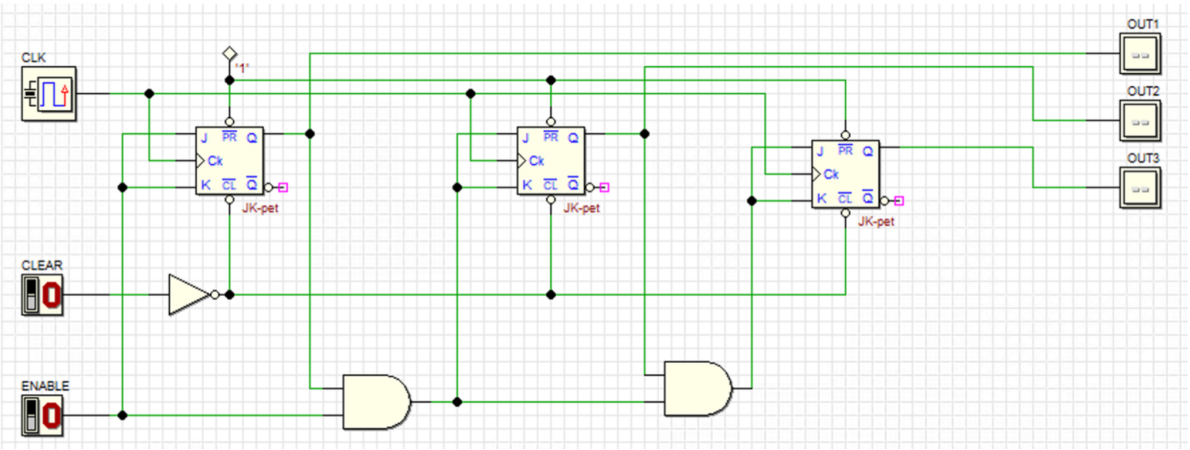
1.2.3 (c) Asynchronous Preset and Clear Flip-Flop

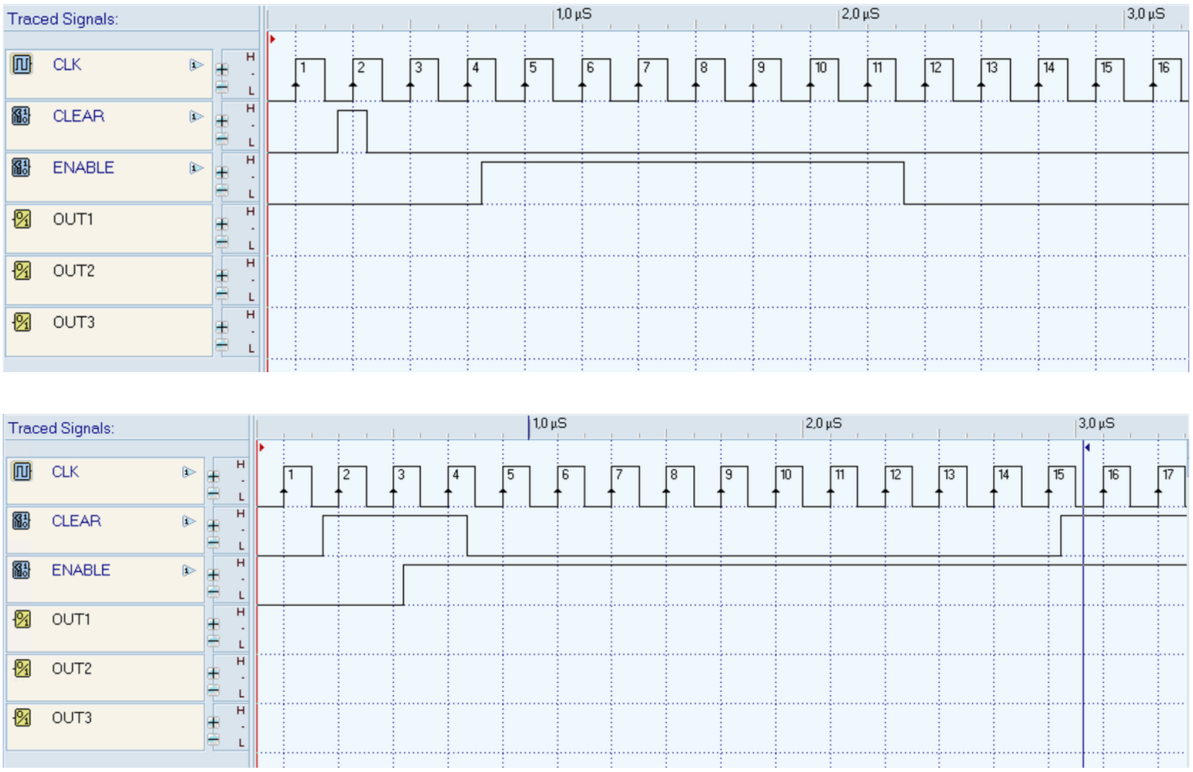
We already considered a reset input for flip-flops, which allows us to force the output Q to 0 regardless of the clock or data inputs. However, we considered only a **synchronous reset**, which acts only on the active clock edge. In many applications, it is useful to have asynchronous inputs that can override the normal operation of the flip-flop at any time, without waiting for a clock event. Preset and Clear are **asynchronous inputs** that let us immediately force a flip-flop into a known state (Preset driving the output high and Clear driving it low), so the circuit can **start up correctly**, **recover from faults**, or be placed instantly in a controlled condition without waiting for a clock edge. Modify the D flip-flop block in order to add a Set inputs, and simulate its operation with a timing diagram that shows how this input override the normal D and clock behavior. Then try the DEEDS library flip-flop with preset and clear inputs and compare its behavior with your design.

1.3 Exercise 3

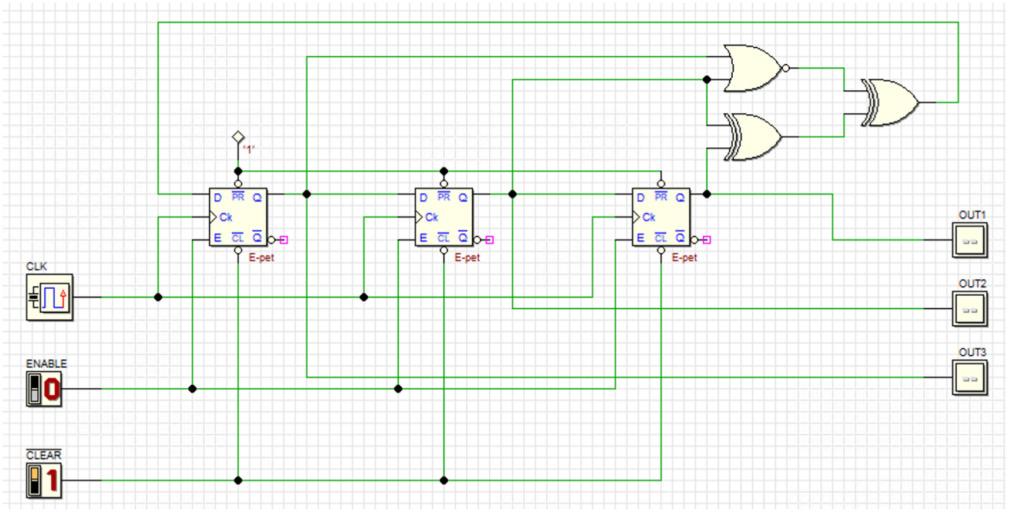
Analyze the following synchronous sequential circuits by completing the timing diagrams. First draw the waveforms by hand and then use DEEDS only to verify your solutions.

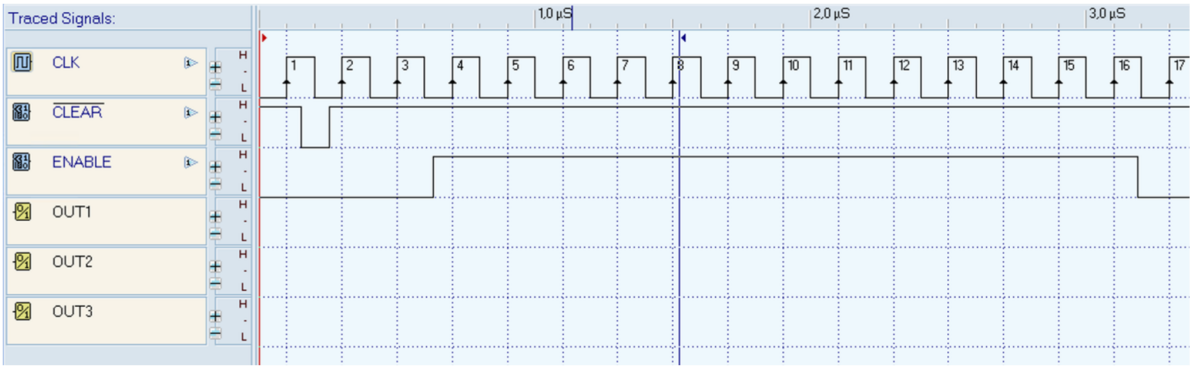
1.3.1 (a)

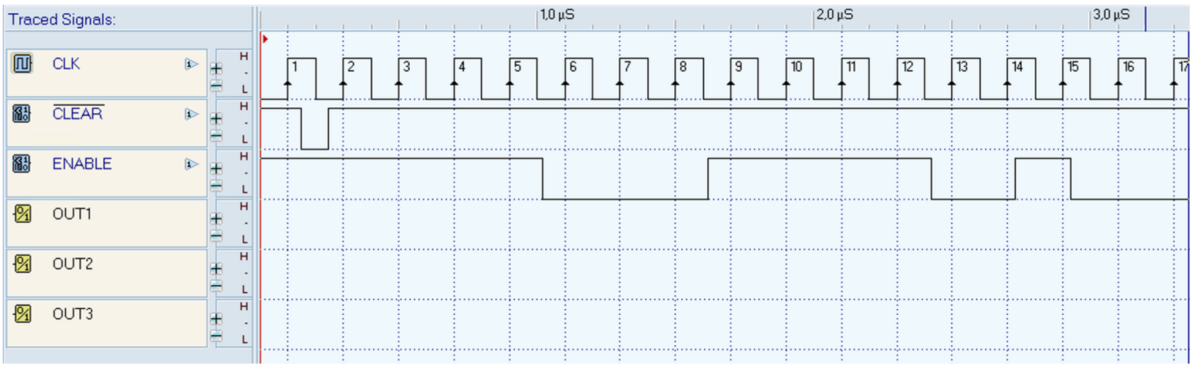



Traced Signals:
1,0 µS
2,0 µS
3,0 µS
CLK
CLEAR
ENABLE
OUT1
OUT2
OUT3
Traced Signals:
1,0 µS
2,0 µS
3,0 µS
CLK
CLEAR
ENABLE
OUT1
OUT2
OUT3

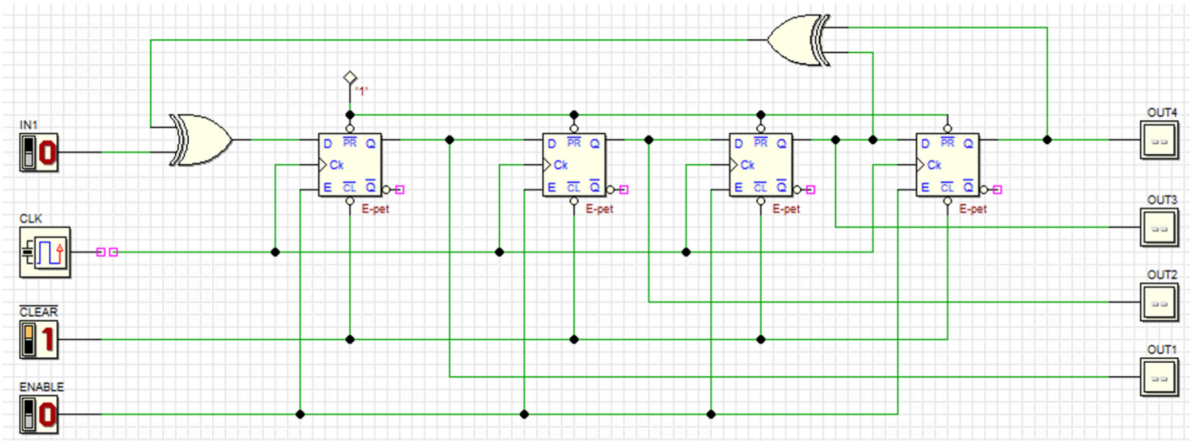
1.3.2 (b)

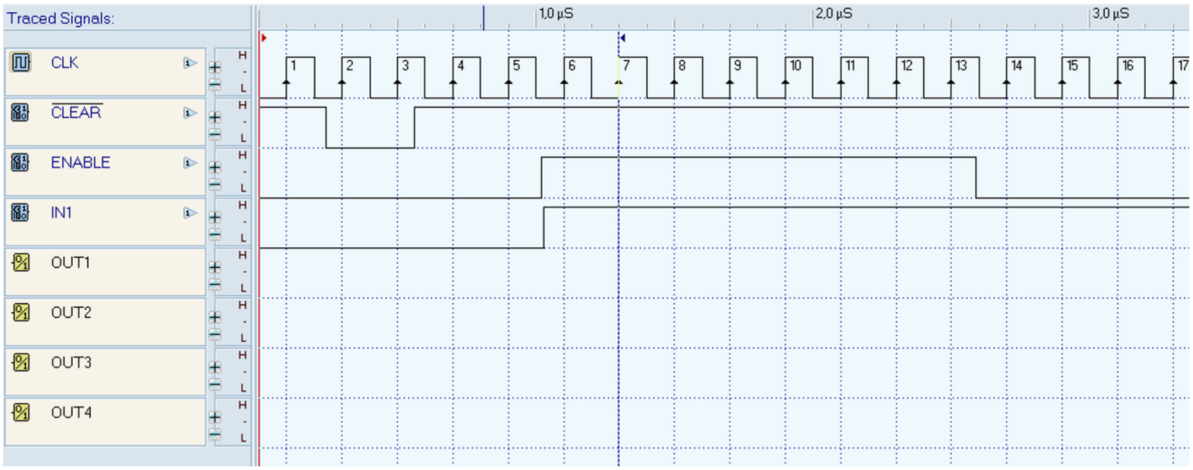

'1'
E-pet
E-pet
E-pet
CLK
ENABLE
CLEAR
OUT1
OUT2
OUT3


Traced Signals:
1,0 µS
2,0 µS
3,0 µS
CLK
CLEAR
ENABLE
OUT1
OUT2
OUT3


Traced Signals:
1,0 µS
2,0 µS
3,0 µS
CLK
CLEAR
ENABLE
OUT1
OUT2
OUT3

1.3.3 (c)


IN1
CLK
CLEAR
ENABLE
OUT4
OUT3
OUT2
OUT1
E-pet


Traced Signals:
CLK
CLEAR
ENABLE
IN1
OUT1
OUT2
OUT3
OUT4
1,0 µS
2,0 µS
3,0 µS
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17