CS2310 Lab 7: Introduction to Sequential Logic

Semester: Jul-Nov 2024

Due: October 22, 2024 (in class)

1 SR Latch

In this question you will implement a SR Latch in two different ways.

a) Implement the SR Latch with behavorial modelling in Verilog, that is, with an `always` block. Use the module signature `module SR_a(output Q, output Q_bar, input S, input R)` for uniformity. **[10]**

b) Implement the SR Latch with gate-level modelling in Verilog. Note that you are not allowed to use the `always` block for this part. Use the module signature `module SR_b(output Q, output Q_bar, input S, input R)` for uniformity. **[10]**

c) Implement a gated SR Latch with behavorial modelling. The truth table for a gated SR Latch is shown below. Use the module signature `module SR_c(output Q, output Q_bar, input S, input R, input E)` for uniformity. **[15]**

Enable (E)	Set (S)	Reset (R)	Q_{next}	$\overline{Q}_{\text{next}}$
0	X	X	Q	$\overline{Q}$
1	0	0	Q	$\overline{Q}$
1	0	1	0	1
1	1	0	1	0
1	1	1	Metastable	Metastable

Note that reset (R) is an active-high signal, i.e., values are reset when R = 1.
Also, note that the metastable condition will not be tested in the testbench.

2 D Latch

In this question you will implement a D Latch in two different ways.

1. Implement the D Latch using behavorial modelling in Verilog. The module should have the signature `module D_a(input D, input en, input rstn, output Q)`. **[10]**
2. Implement the D Latch using gate-level modelling in Verilog. The module should have the signature`module D_a(input D, input en, input rstn, output Q)`. **[15]**

Note that rstn is an active-low signal, i.e., values are reset when rstn = 0.