Lab 1: DriveLock Example

Digital Design Lab

Overview

- Lab 1 methodology
- DriveLock example
  - Lab procedure walkthrough

Methodology

Before creating the VHDL code, we should do the following:

1. Understand the problem

2. Derive the truth table
3. Derive a minimal Boolean expression
4. Optimize Boolean expression for the given gate library
5. Estimate input/output delay

6. Write behavioral model with estimated time (see VHDL videos)
7. Simulate and submit the behavioral model

8. Write the structural model (see VHDL videos)
9. Simulate the structural model and compare simulation results
10. Submit the structural model

DriveLock: Problem Description

Design a small digital circuit, called DriveLock that locks the car driving (by setting an output w to 1) when either of the following conditions are met:

- a car's key is in the car's ignition slot (indicated by an input k being 1), and a passenger is seated (indicated by an input p being 1), and the passenger's seat belt is not buckled (indicated by an input s being 0)
- a car's key is in the car's ignition slot (indicated by an input k being 1), and a passenger is seated (indicated by an input p being 1), and the car door is not closed (indicated by an input d being 0).

Design this circuit using only 2-input NOR gates (1.4ns delay each).

1. Understand the Problem

How does DriveLock work?

- How can the DriveLock behavior be represented?

What is a NOR gate?

- Can NOR gates be used to represent this circuit?

What steps do we need to perform before coding?

2-4. Truth Table and Expressions

DriveLock truth table:

Inputs				Output
k	**p**	**s**	**d**	**w**
0	x	x	x	0
x	0	x	x	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

kp \ sd	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	0	1
10	0	0	0	0

Drivelock minimal sum-of-products Boolean expression:

w = kps' + kpd' -- from K-Map

Convert to 2-input NOR gates (use Boolean logic rules):

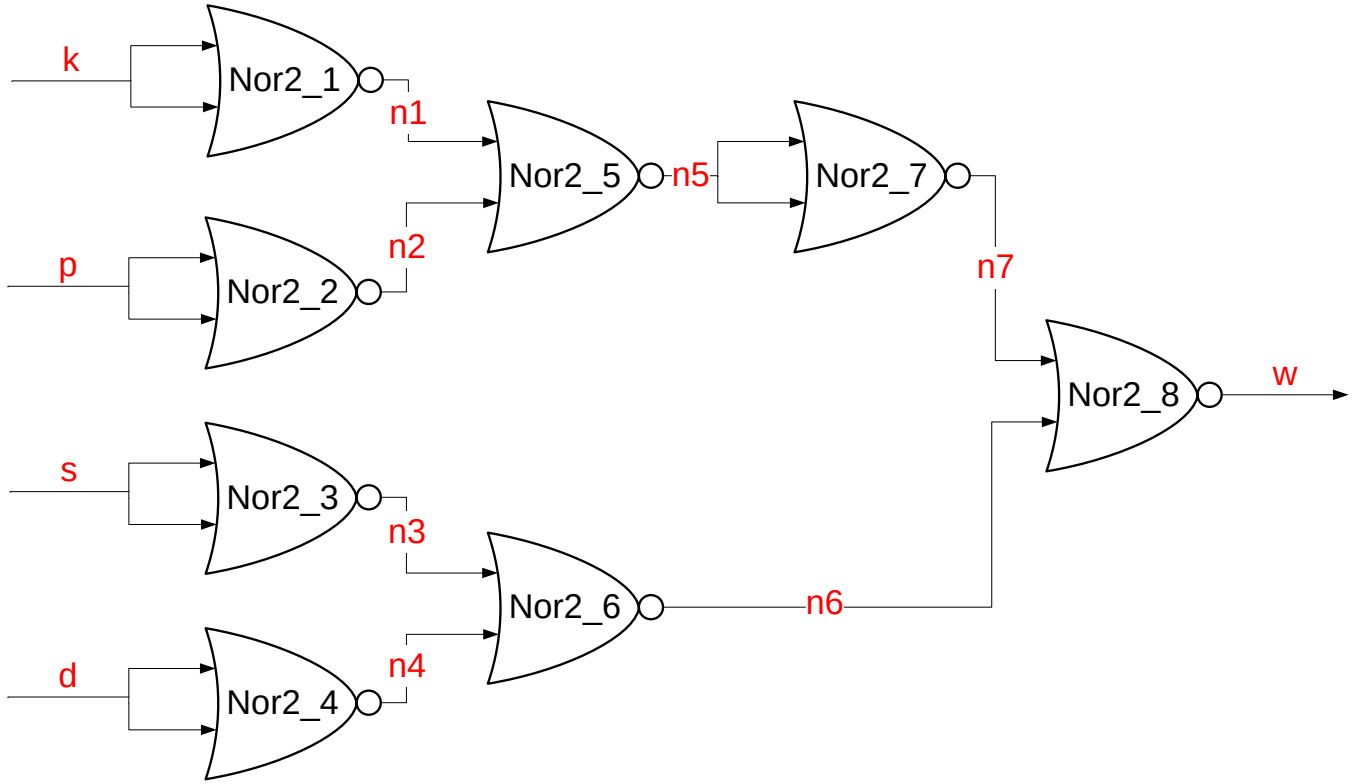
w = (kp)(s' + d') -- Distributive Law
= (k' + p')'(s' + d') -- De Morgan's Theorem
= (((k' + p')')' + (s' + d')')' -- De Morgan's Theorem

(((k' + p')')' + (s' + d')')'
0
0
1
1
1
0

5. Calculate Delay

Draw design and find delay:

w = (((k' + p')')' + (s' + d')')'



Delay: 5.6 ns

6-10. VHDL Coding

Ready for Xilinx!

Summary

- Lab 1 methodology
- DriveLock example
  - Lab procedure walkthrough