|  |  |
| --- | --- |
| **Student**: | Michael Escue |
| **Assignment**: | Assignment 5 |
| **Class**: | ECE 351 |
| **Professor**: | Dr. Garrison Greenwood |
| **Term**: | Spring 2019 |

**Questions:**

1. **Briefly explain what “latch inference” means.**

Latch inference occurs when the synthesizer isn’t instructed on how to update a previous value, so a latch is created to store the last value since it never changes (no statement to address the value).

1. **What are two ways latch inference occurs. How can you prevent if from occurring?**

The two ways latch inference occurs, is if there is no default statement inside of a case statement, or if there is not a pre-defined value for an if-then-else statement (a default “else” case) which will always update a value based on the conditional statement to be evaluated. To prevent latch inference, include the default statements for both the if-then-else and case statements so that all values are updated, and no values are latched.

1. **What is a synthesis directive?**

The synthesis directive instructs the compiler to take specific action on statements. For example, with the case statement, a synthesis directive can be included in a situation where one-hot encoding is utilized, but no default case is defined. The “synthesis full\_case” directive, in this case, would allow the statement to be written without the default case explicitly and provide improved readability, but will prevent latch inference because the synthesizer knows that the statement is missing defined states.

1. **It is recommended that module outputs be registered. Why?**

It becomes easier to analyze the timing between modules. It also improves any critical paths between modules by relaxing the hold time a module needs to keep a signal active while it propagates to the next module input. This effectively pipelines signals and thereby can provide speedup of Input, output, and high-fanout signals.