Spring 2019

California State University, Northridge

Department of Electrical & Computer Engineering

Lab Experiment 4

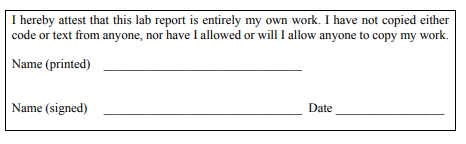
Behavioral Modeling of a Counter

February 28, 2019

ECE 526L

(Late)

Written By: Juan Silva

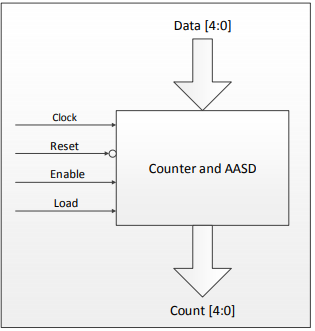


**Introduction:**

In this experiment, a five bit counter will be constructed using behavioral modeling. Behavioral modeling is the highest level of hardware design allowing for complex circuits to be built. This is done using top-down design and will be used in this experiment. This counter will have a top-level module that will instantiate lower level modules. The counter module and the Asynchronous Assert, Synchronous Deassert. The counter will load a five bit data input and increment the value by one at every rising edge of the clock. When the reset is called, an synchronous reset will be applied. That’s where the AASD circuit is used. The AASD module will take into account reset capabilities by applying an asynchronous assert, synchronous deassert pulse to two D flip flops. The output of the second flip flop will be a stable reset for the counter to use a reset to be applied without entering a recovery violation.

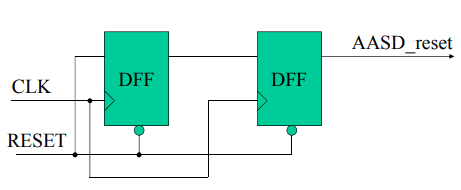
**Procedure:**

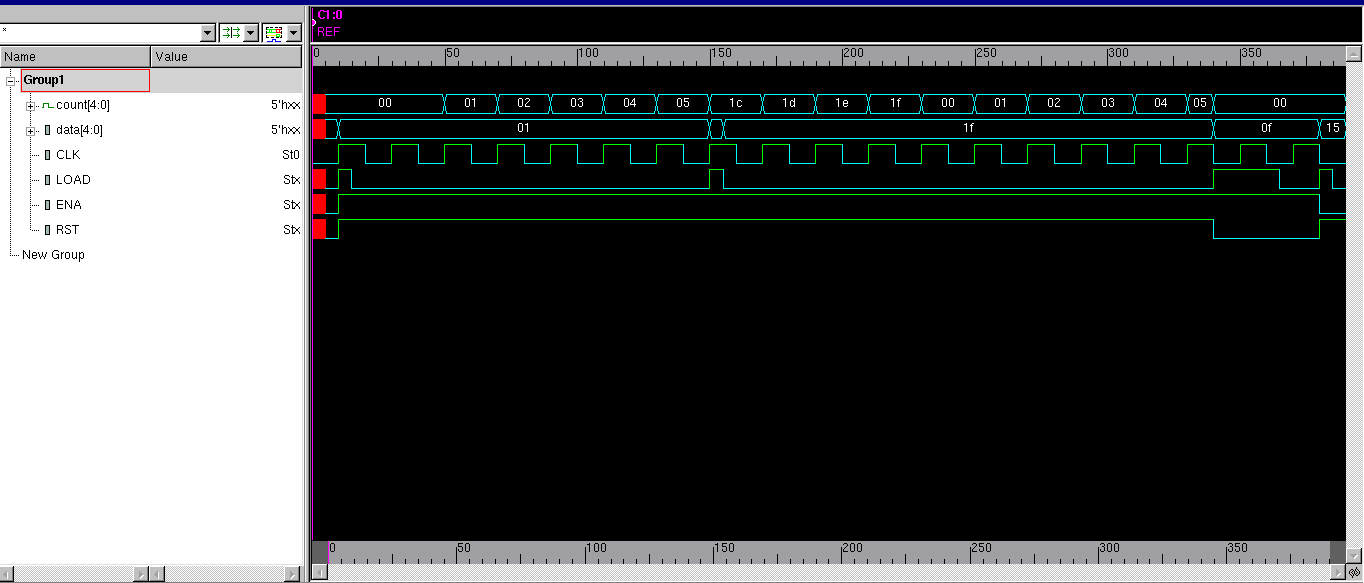
1. Identify the each level of the circuit. The top-level design will count and apply a stable asynchronous reset. To do this, two lower modules will be defined to model the top-level module.



***Figure 1.1*** *- 5 Bit Counter WIth AASD Capabilities*

1. A counter is defined using conditional statements with various parameters. This counter, when enabled, will either load data into the counter. When loaded, the counter will then increment its loaded value until more data is loaded or when enabled is low. If the counter were to receive an input low reset, the counter will reset to zero regardless of the state of the counter.
2. The Asynchronous Assert, Synchronous Deassert will consist of two D Flipflops. A reset is applied in the input of the first flip flop and the output of the second flip flip will be the new reset used for the top-level counter.

  
***Figure 1.2*** *- AASD Circuit*

**Results:**

***Figure 1.3*** *- Waveform Output of 5 Bit Counter*

**Discussion:**

Behavioral modeling introduces a top-down design methodology allowing for complex circuits to be designed. The top level module will consist of two lower-level modules to be instantiated. These lower level modules will be the counter and the AASD feeding the asynchronous reset to the counter. One advantage for using behavioral modeling for this counter is the flexibility to add additional modules such as the AASD circuit for reset. If the counter requires additional features, the top level design will all for it. This circuit is necessary to stabilize the reset pulse by allowing enough time for the asynchronous reset to be applied to the counter. This is to prevent desync from occurring and allow the circuit to function properly

Behavioral modeling eliminates the need to build from a gate level. This is far more efficient than using primitives allowing for circuit behavior to be designed using conditional statements rather than building from gates. The counter is modeled to behave conditionally such that the highest priority of the inputs will be the reset and enable. A nested if-statement allows for additionally condition to occur if enabled is applied.

For testing, behavioral code allows for modules to be stored individually. If the simulator is unable to compile, the simulator will notify the user the culprit. Additionally, these modules can be reused for later experiments if needed. This reusability is the first step into building a library of modules for future use. When testing the functionality of the counter, a clock generator is used with a fixed time delay. This is done using procedural blocks to allow for the clock to rise and fall at a given period. This bypasses the need to manual set a clock and allows for easier testing.

**Conclusion:**

Behavioral modeling is a preferable design methodology when building large or complex circuits. It allowed for circuits to be designed by its behavioral and not have to worry about the inner circuitry involved to define its function. For this counter, conditional statements were sufficient to define the counters behavior based on the input it receives. Plus, this allows for top level design to be implemented. SImply define the behavior of this block based on its inputs and design lower blocks that can achieve the behavior defined. For the counter, it only requires a module that counts and a module that stabilizes an asynchronous reset.