Dustin Kendall

kendalld@pnw.edu

Abstract

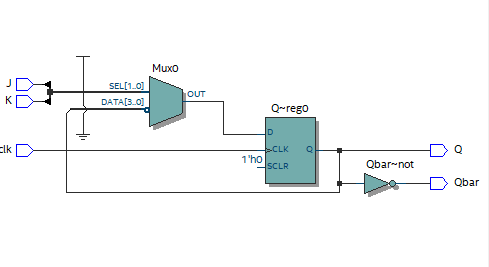
Using process statements to create counters, shift registers, and flip flops.

Lab 3 Process Statements

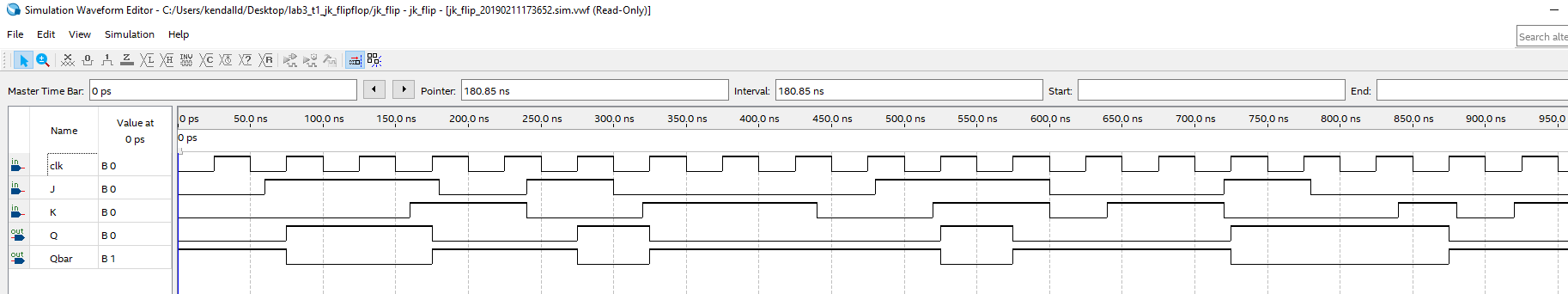
ECE 371 Computer Organization and Design

# Task 1 JK Flip Flop

## RTL Viewer



## Functional Simulation



## VHDL Code

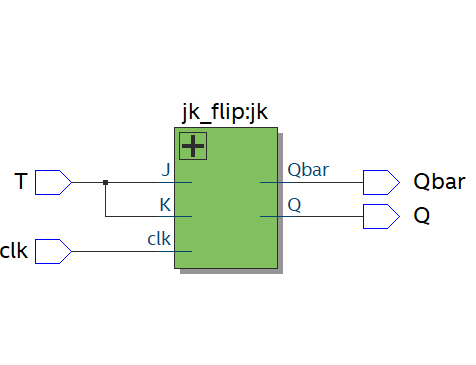


# Task 1 T Flip Flop

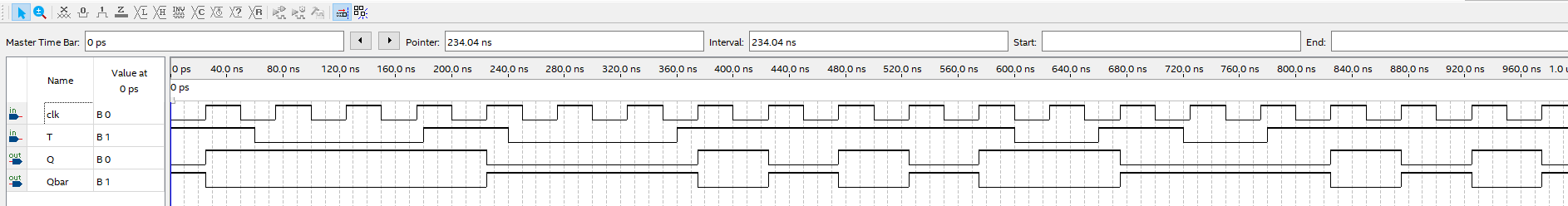
## Method

A T flip flop is just a JK flip flop with inputs J and K tied to a single input T. It toggles if T = 1, otherwise it holds its value.

## RTL Viewer



## Functional Simulation



## VHDL Code

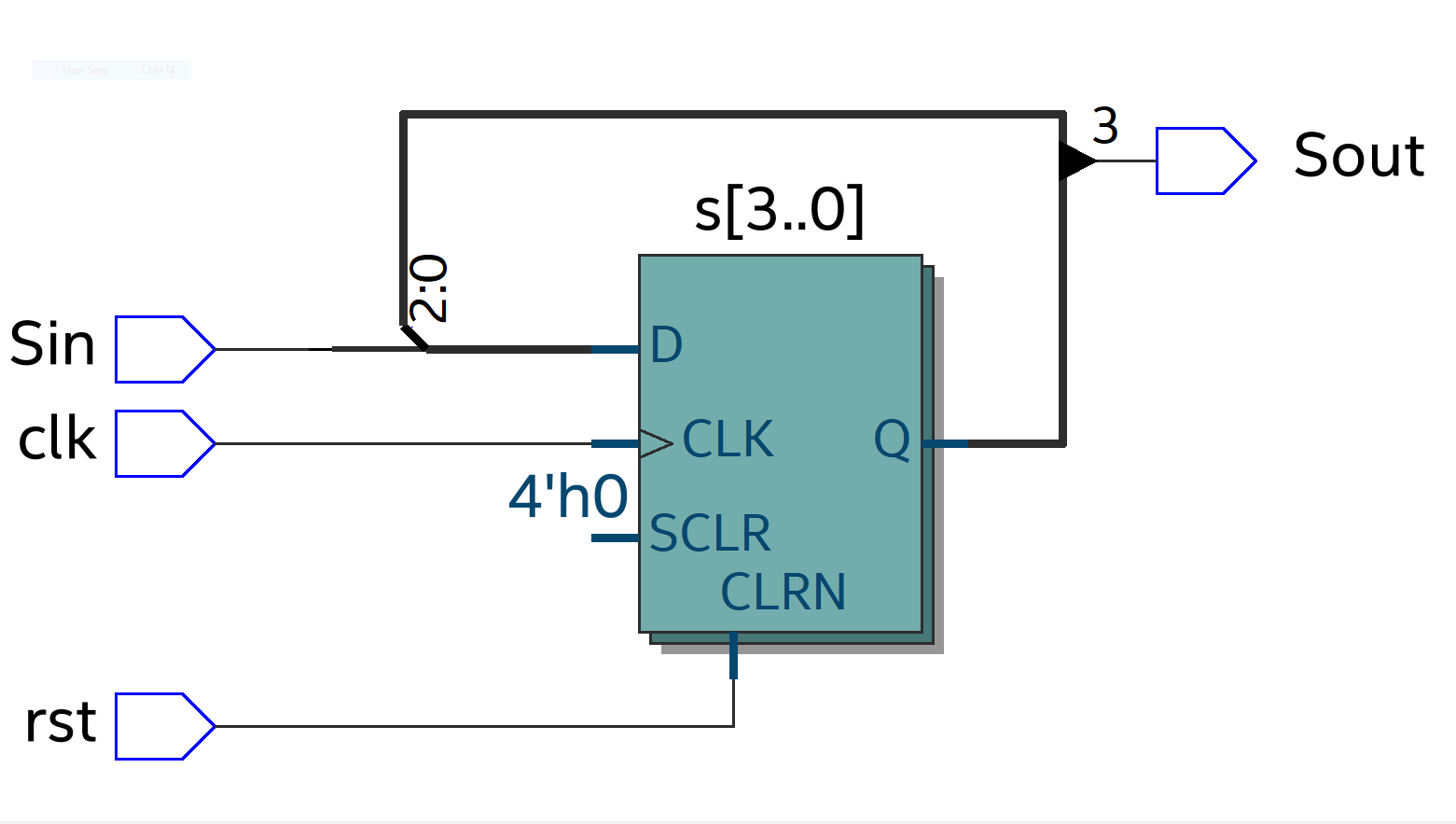


# Task 2 Shift Register

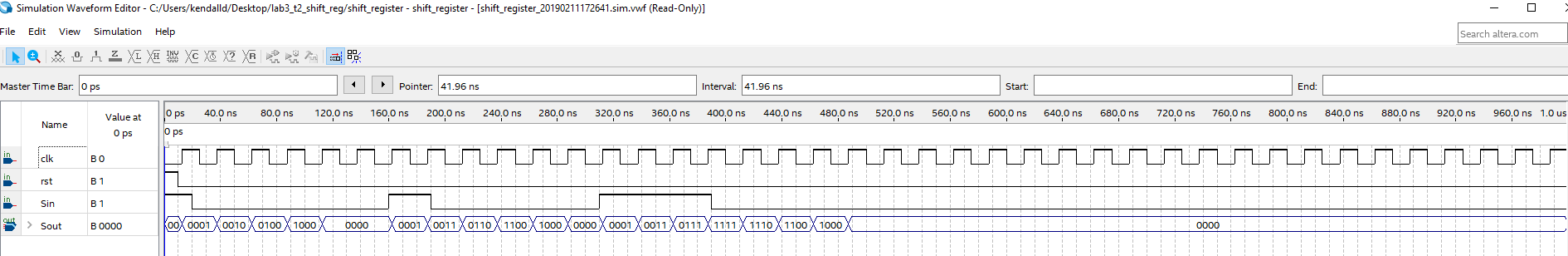
## Method

Used a process statement to model an N-bit shift register. Shifting the bits to the right each clock cycle

## RTL Viewer



## Functional Simulation



## VHDL Code

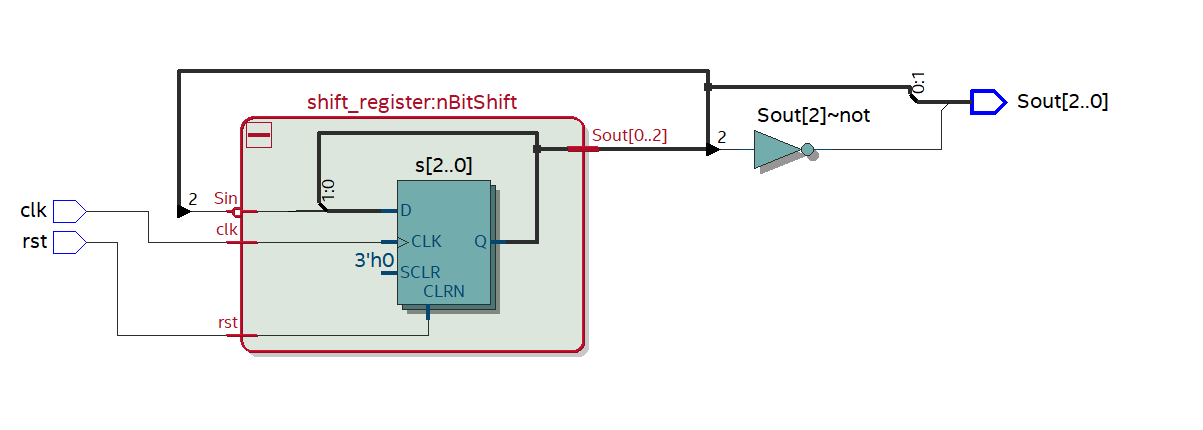


# Task 3 N-bit Johnson Counter

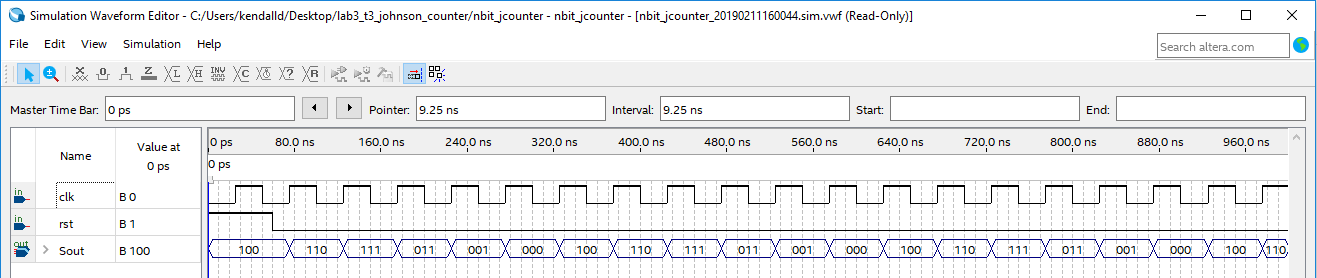
## Method

Using the shift register in the previous task, I used a structural model to port the output of the N-bit shift register to the input.

## RTL Viewer



## Functional Simulation



## VHDL Code

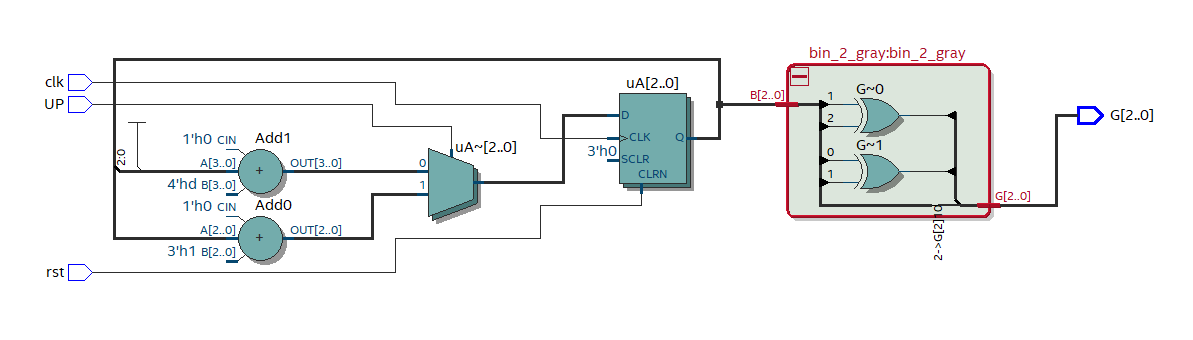


# Task 4 Gray Code Counter

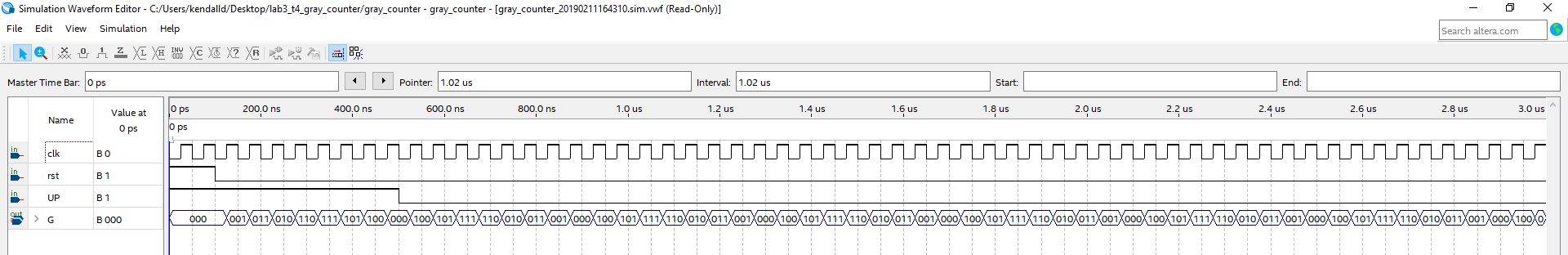
## Method

Using some of the combinational logic from before I created a generic parameterized model of a gray code counter. It uses the unsigned package to increment and decrement the bits within a process statement, then lastly uses a structural model to output the gray code equivalent.

## RTL Viewer



## Functional Simulation



## VHDL Code

