# Activity 02 – Frequency Divider & Edge Detections

Richard Groves

09/15/2023

Class: ECE4/530 - Digital Hardware Design

# Introduction

## Objective

The objective of this activity was to design, synthesize, simulate, and implement sequential components utilizing Verilog. The focus was on frequency division and edge detection mechanisms. This involved: creating a frequency divider and developing a synchronization and edge detection system.

# Part 1 - Frequency Divider

## Introduction

The first segment of this activity was to design and implement a frequency divider. The board's 125MHz clock was divided to get 1Hz and 5Hz outputs.

## Design Approach

A frequency divider module was created. Two instantiations of the frequency divider module were used to get the two desired output clock frequencies.

A diagram of a number of dividers

Description automatically generated

## Included Files

|  |  |
| --- | --- |
| File Name | File Description |
| Freq\_Div\_Board.bit | Bitstream file for Freq\_Div\_Board. |
| Freq\_Div\_Schematics\_High\_Level.jpg | An image of the high-level schematic. |
| Freq\_Div\_Schematics\_Low\_Level.jpg | An image of the low-level schematic. |
| Freq\_Div.mp4 | A video of the implemented Freq\_Div\_Board on the ZYBO-Z7 board. |
| Freq\_Div\_Imp.png | An image of the implemented design. |
| Freq\_Div\_Sim.png | An image of the simulated waveforms. |
| Freq\_Div\_Util.txt | The utilization report for the design. |
| Freq\_Div.v | This module divides the frequency of the input clock signal (clkin) by a specified divisor. The module also supports a negative reset (n\_rst), which resets the clock divider and the output clock signal to their initial states. |
| Freq\_Div\_Board.v | This module generates two different frequency clock signals, 1Hz and 5Hz, from the input clock signal. |
| tb\_Freq\_Div\_Board.v | This testbench tests the functionality of the Freq\_Div\_Board module. It generates a clock signal with a specified period and toggles a reset signal to test the reset functionality of the Freq\_Div\_Board module. |

# Part 2 - Synchronization and Edge Detection

## Introduction

The second segment of this activity was to develop a synchronization and edge detection system.

## Design Approach

Three D flip-flops were to achieve synchronization and edge detection. These flip-flops were configured with an inverted reset and were directly connected to the system clock. Additionally, a compare module was implemented to identify signal transitions.A diagram of a circuit

Description automatically generated

## Included Files

|  |  |
| --- | --- |
| File Name | File Description |
| Edge\_Det\_and\_Synch.bit | Bitstream file for Edge\_Det\_and\_Synch. |
| Edge\_Det\_and\_Synch\_Schematics\_High\_Level.jpg | An image of the high-level schematic. |
| Edge\_Det\_and\_Synch\_Schematics\_Low\_Level.jpg | An image of the low-level schematic. |
| Edge\_Det\_and\_Synch.mp4 | A video of the implemented Edge\_Det\_and\_Synch on the ZYBO-Z7 board. |
| Edge\_Det\_and\_Synch\_Imp.png | An image of the implemented design. |
| Edge\_Det\_and\_Synch\_Sim.png | An image of the simulated waveforms. |
| Edge\_Det\_and\_Synch\_Util.txt | The utilization report for the design. |
| D\_FF.v | This module implements a D-type flip-flop (D FF) with a negative reset input. The flip-flop captures the value on the D input at the rising edge of the clock and transfers it to the Q output, unless the reset is active, in which case it resets the Q output to 0. |
| Edge\_Det.v | This module detects the rising and falling edges between two input signals, in\_1 and in\_2. A rising edge is detected when in\_1 is high and in\_2 is low, setting the 'rise' output high. A falling edge is detected when in\_1 is low and in\_2 is high, setting the 'fall' output high. In all other cases, both 'rise' and 'fall' outputs are set low. |
| Freq\_Div.v | This module divides the frequency of the input clock signal (clkin) by a specified divisor. The module also supports a negative reset (n\_rst), which resets the clock divider and the output clock signal to their initial states. |
| Edge\_Det\_and\_Synch.v | This module integrates the functionalities of frequency division, flip-flops, and edge detection to synchronize and detect edges in a signal. It takes a signal input and synchronizes it with a 1Hz clock, then detects rising and falling edges in the synchronized signal. |
| tb\_Edge\_Det\_and\_Synch.v | This testbench module tests the Edge\_Det\_and\_Synch module by applying a series of clock and signal transitions and observing the rise and fall outputs. |