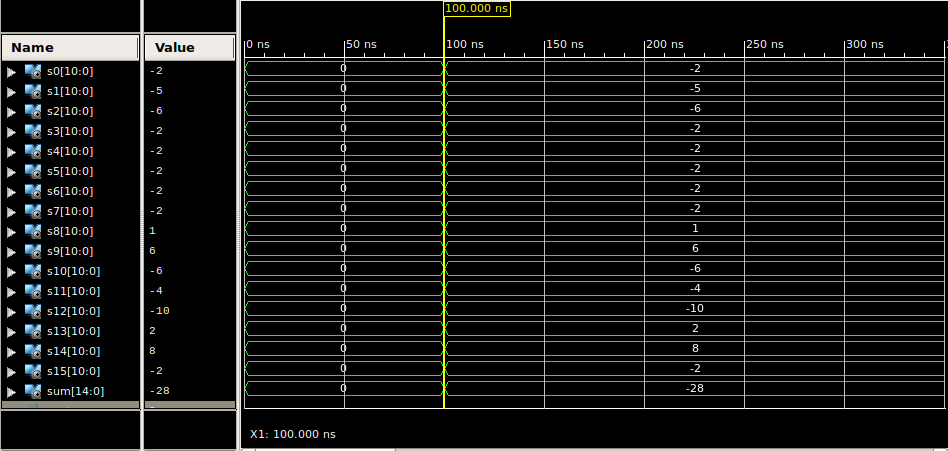
**Lab Assignment 2 Report**

**Part I: Simulation results for each part of the system**

1. Wave form for each stage

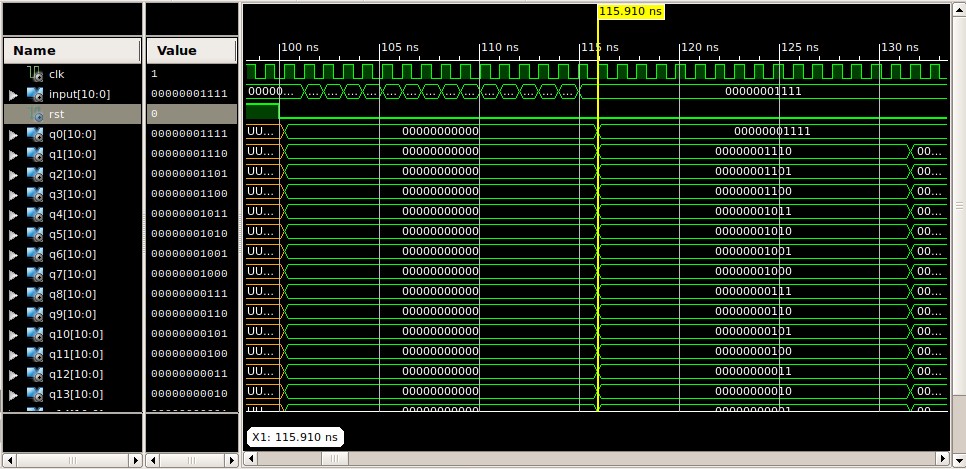
1) 16 Input Carry Save Adder

Behavioral Simulation Wave Form Design 1:

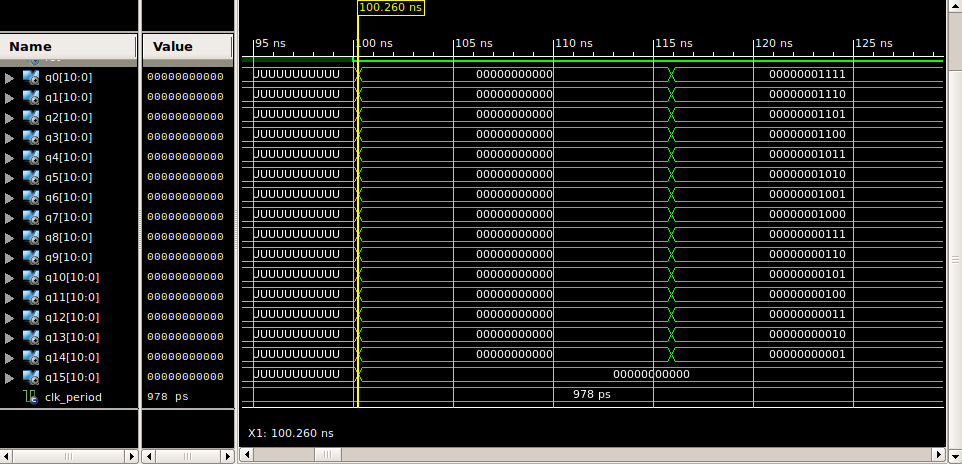


1. Register Array

Behavioral Simulation Wave Form Design 1:

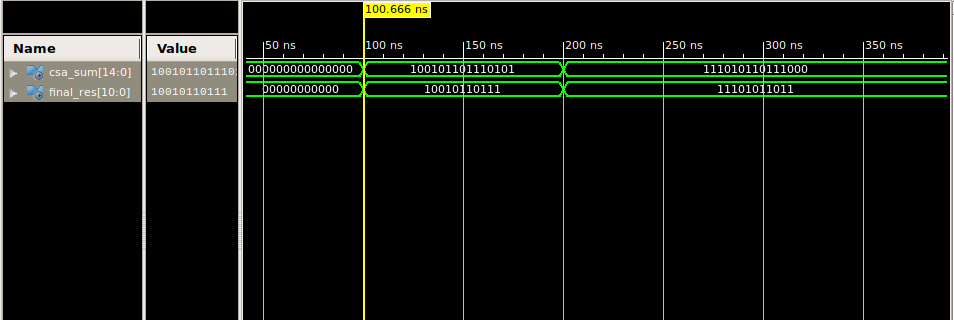


Behavioral Simulation Wave Form Design 2:



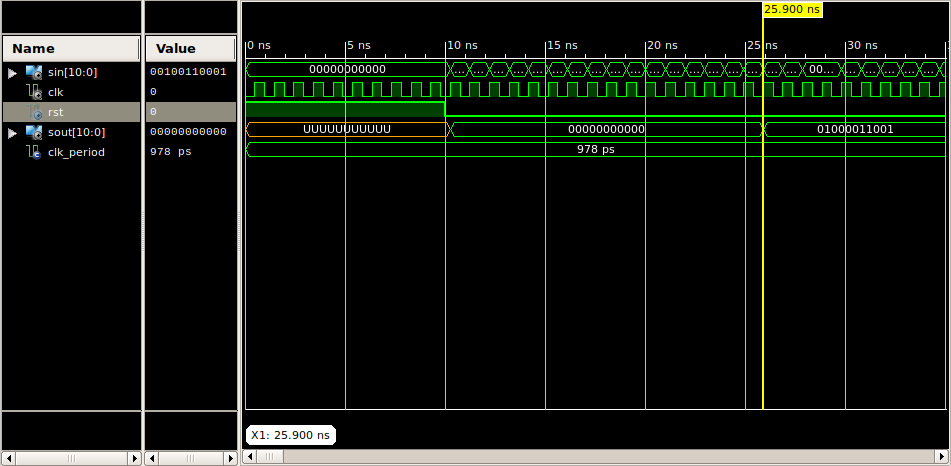
1. Shifting

Behavioral Simulation Wave Form:



1. Moving Average Filter

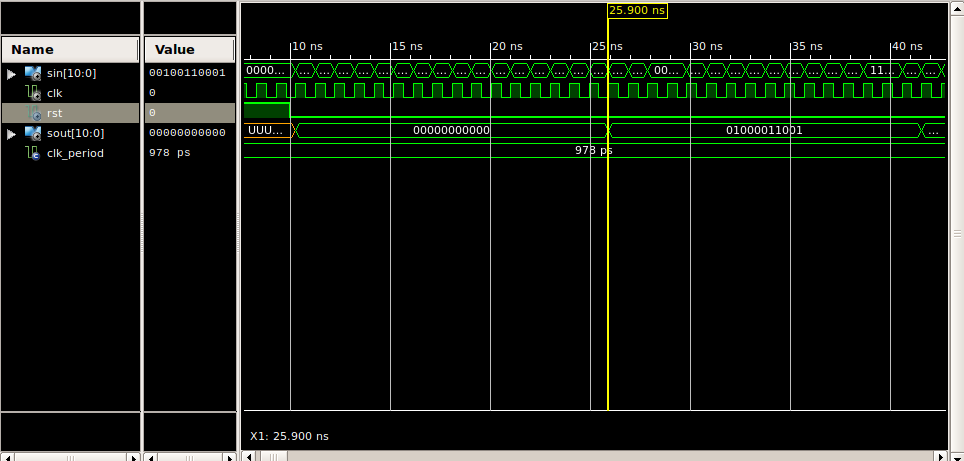
Behavioral Simulation Wave Form Design 1:



PAR Simulation Wave Form Design 1:



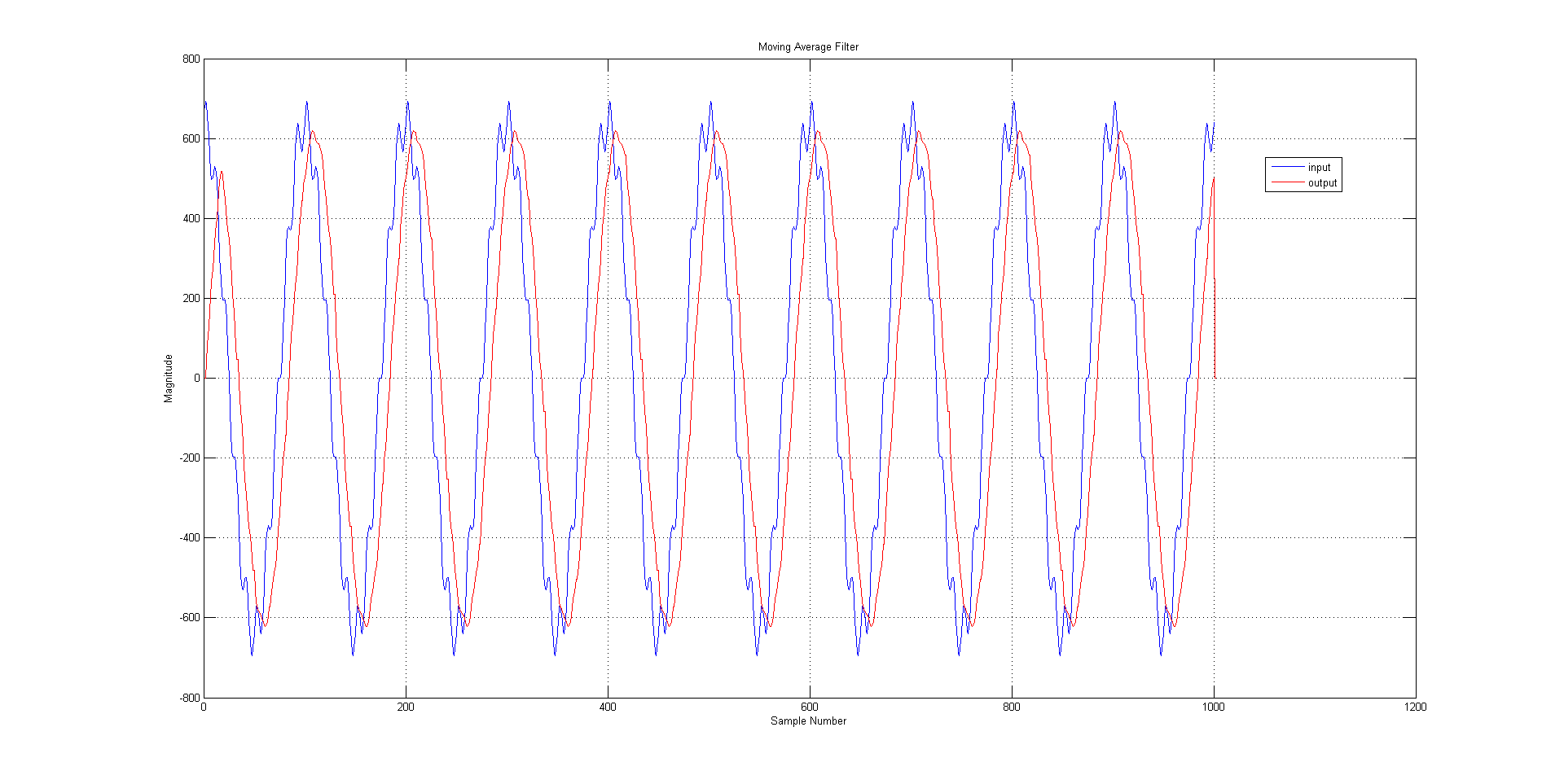
Behavioral Simulation Wave Form Design 2:

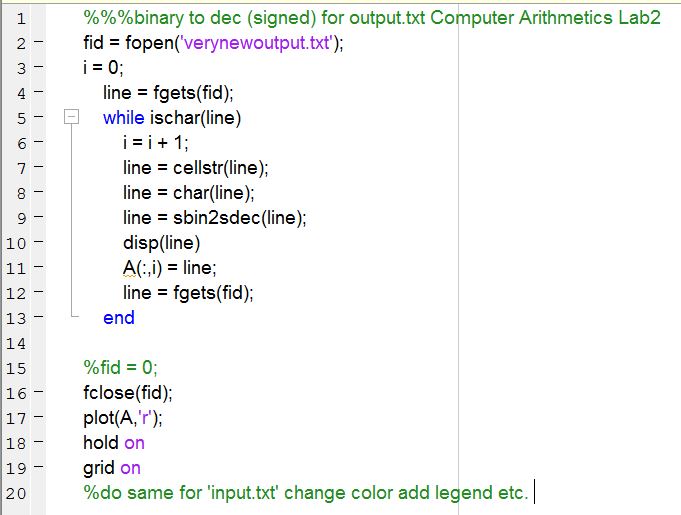


PAR Simulation Wave Form Design 2:

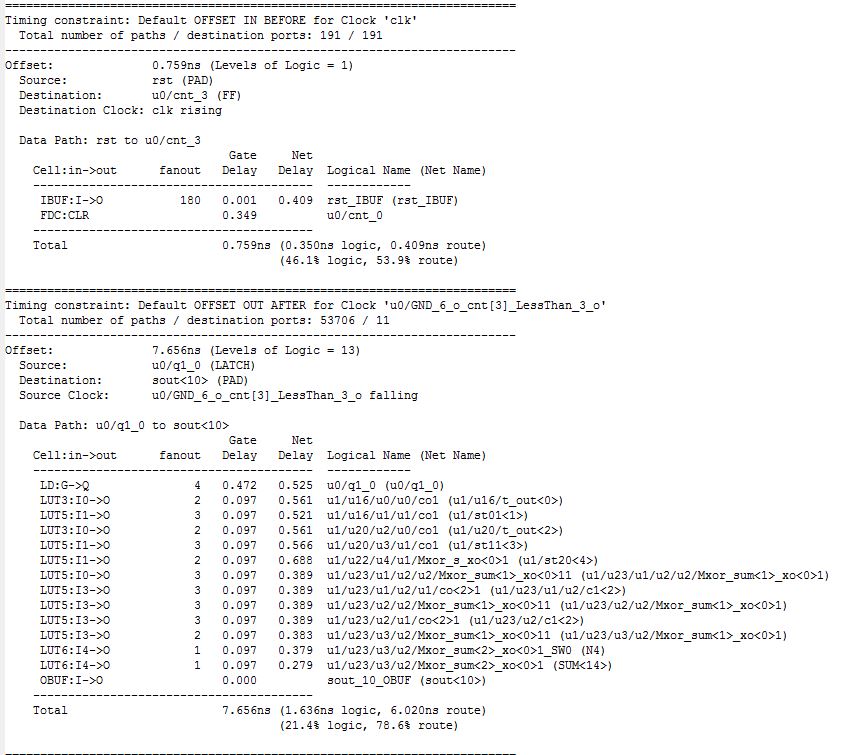


B. Input and Output plot

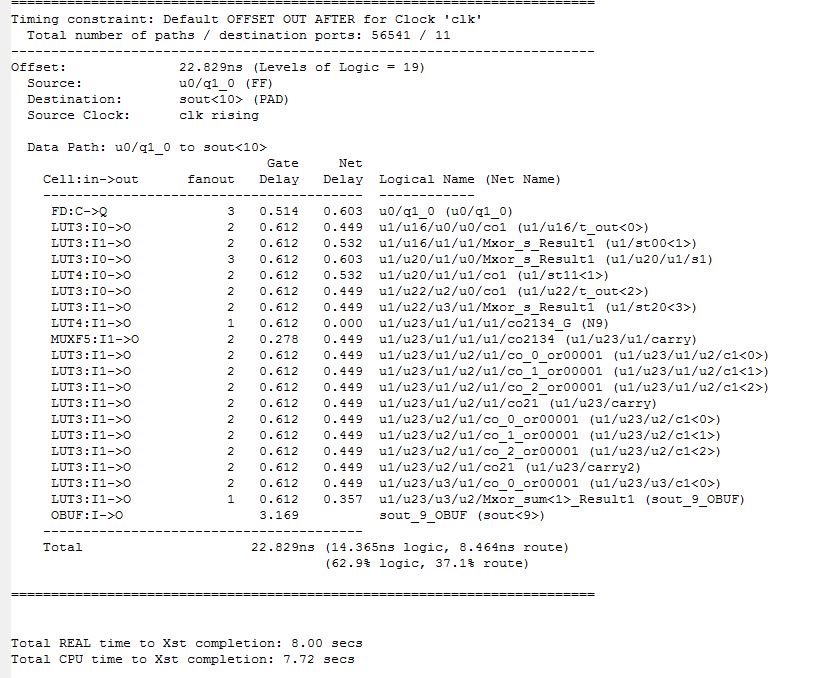




Timing Report for Design 1



Timing Report for Design 2



Operating frequency of Design 2 is 130MHz.

Modifications : In Design 1, he have implemented the whole structure as 3 modules; Register, Summer and Divider. For this we’ve used a SIPO input register, a CSA structure, a fast summer as a CLAA and a Shifter.

We were able to get PAR simulation for the whole structure of moving average filter using the provided testbench file . But, the testbench files we wrote were unable to provide us PAR simulations, and the most common error regarding this situation was there about a connection problem between modules of the MAF; u0 and u2 which stands for input register and shifter. Thus we’ve cancelled the shifter as a different in the second design and implementing the shifting or dividing phase simply by mapping outputs. But it didn’t solve the issue and we still weren’t able to get PAR simulations for modules other than MAF. It turned out that we had to include Clock in all testbenchs files of modules if we want PAR simulations.

However, we have found out that, taking out the third module Shifter had made the whole structure faster, since less logic and look up tables were used that way.

A better design can be implemented by adding registers in between all stages, then one clock cycle can be as short as the longest logical operation in any one of the stages.