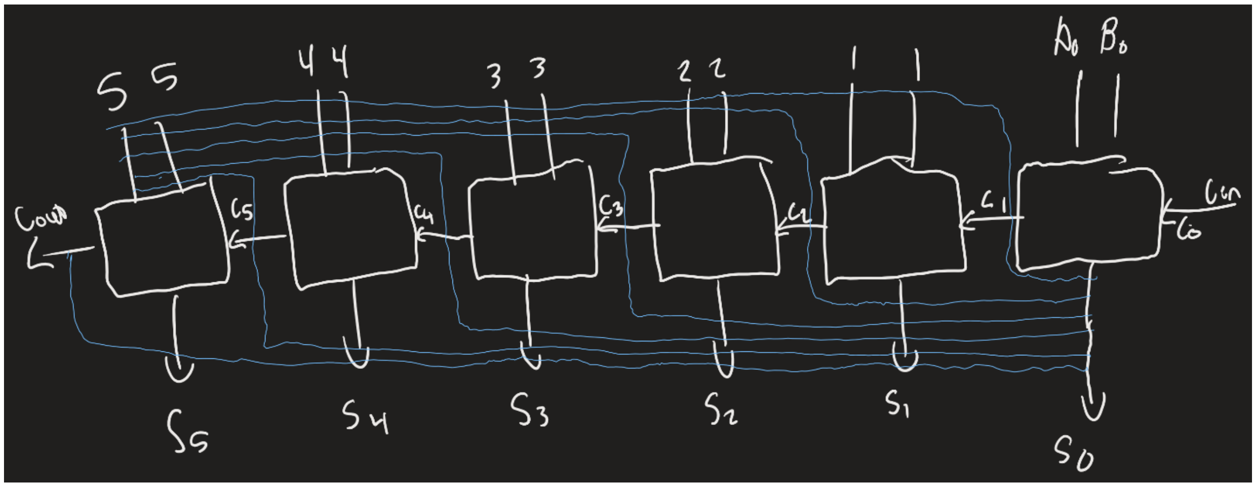
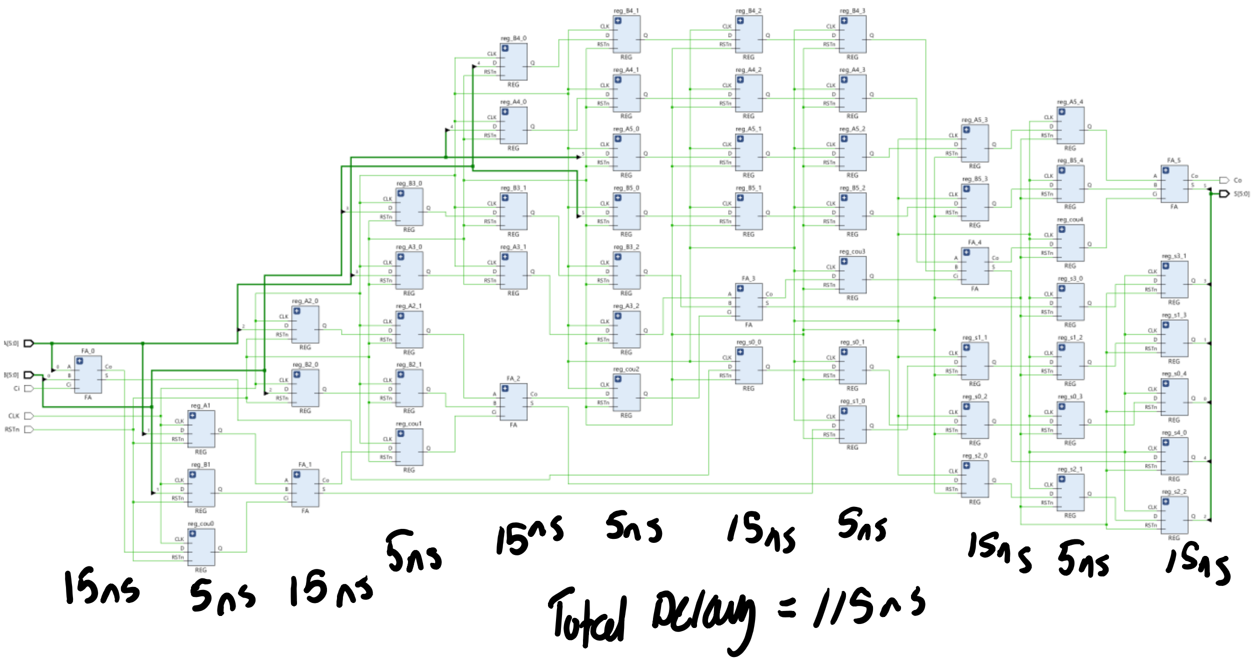
6-bit synchronous 6 pipeline parallel adder design:



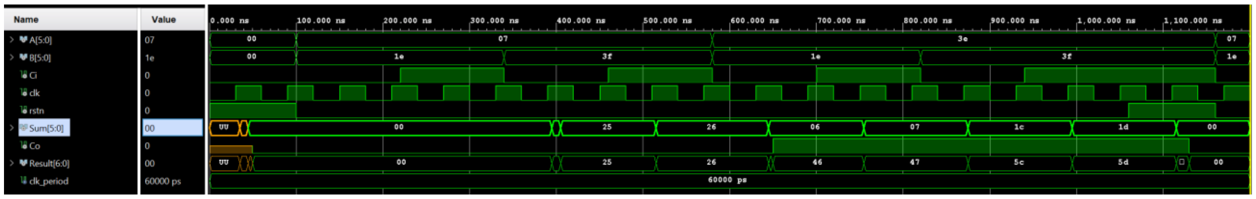
Schematic for 6-bit synchronous 6-pipeline parallel adder:



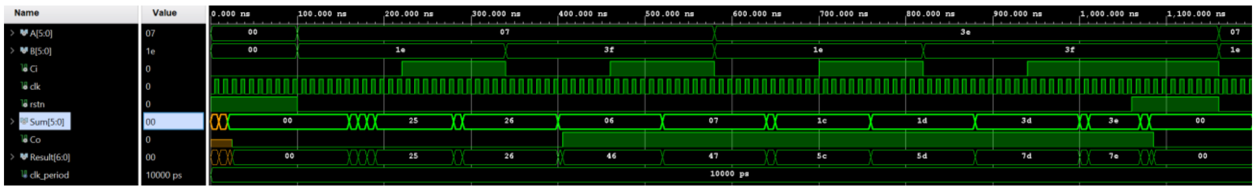
The total delay between when an input is given, and the output is set will be at most 115ns according to the schematic of my design above. With this in mind I set the wait delay to 120ns after each input is read and the clock period to 60ns which is the delay time for 12 registers on one clock cycle (the most registers each clock could encounter).

Theoretically the delay of one register is 5ns so I also tested a 10ns clock waveform to see if it would net any positive results. The waveform is posted below as well.

6-bit synchronous 6-pipeline parallel adder waveform 60ns clock:

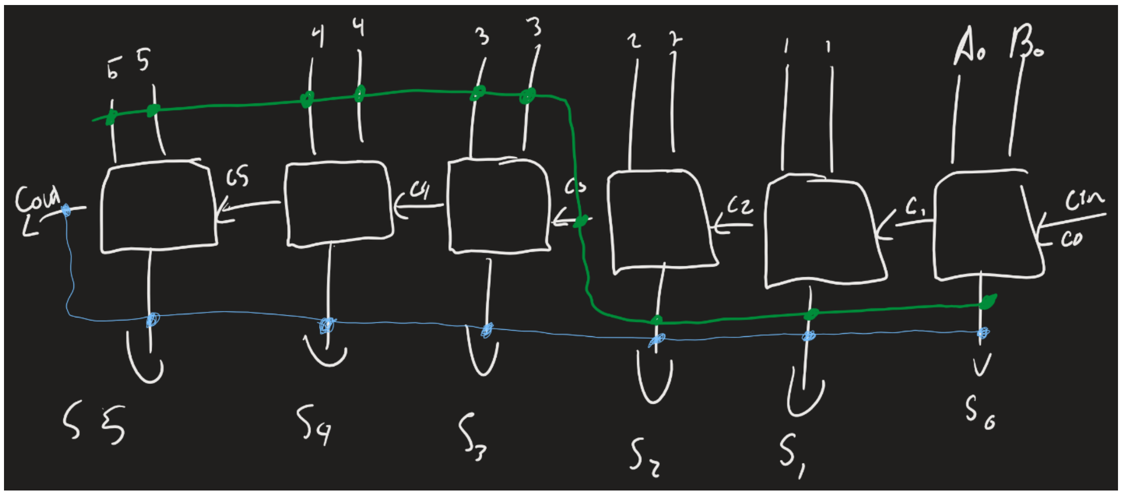


6-pipeline 10ns clock:

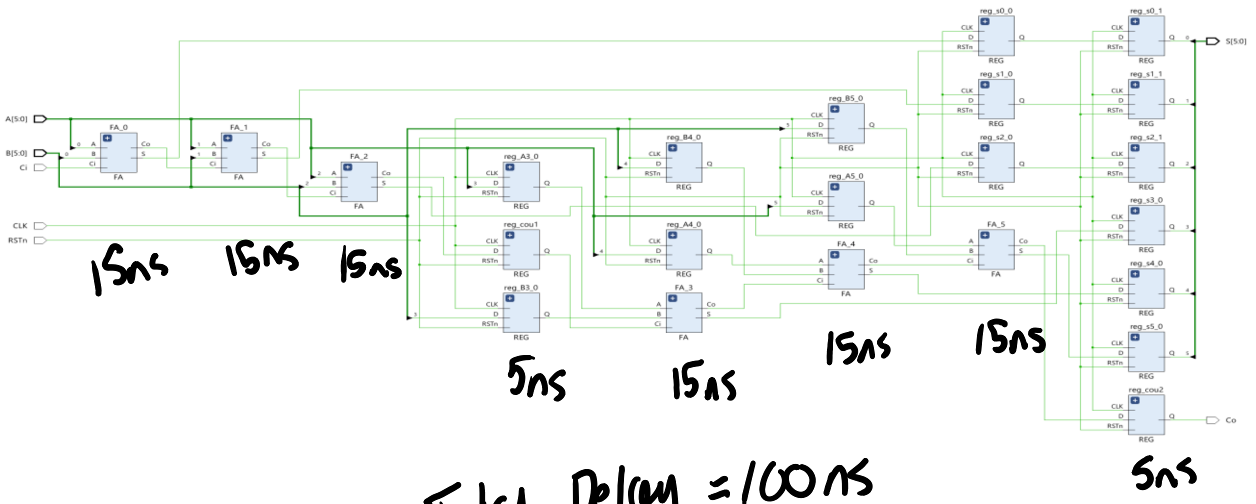


Between the two tested values the 60ns clock does not run fast enough to attain all the results before the reset is called again. The 10ns clock is too quick though and has erroneous values we don't want in our output. Of the two the 10ns clock speed is potentially better because the correct values are attained the majority of the time.

2pipeline design:



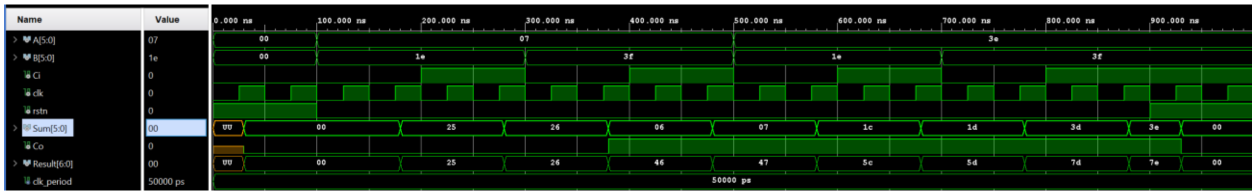
Schematic for 6-bit synchronous 2-pipeline parallel adder:



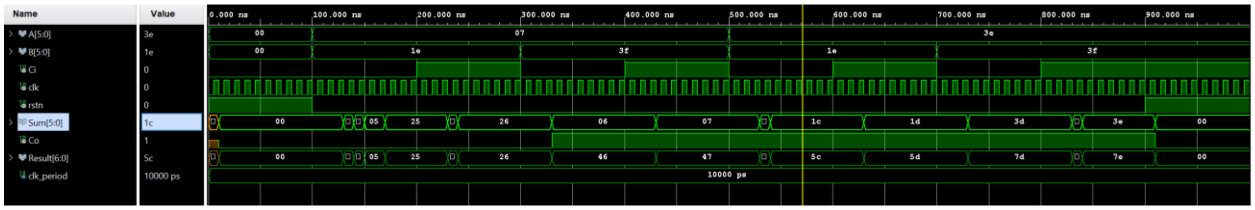
Total Delay = 100ns

The total delay between new inputs and outputs will be 100ns at the most. Therefore, the delay will be 100ns after input is received. Each register takes 5ns to run. The most registers needed to run on one clock cycle is 10. This means the minimum clock speed can be 50ns. With the delays set the output waveform is as follows.
I also tested a 10ns clock speed to see if it had any useful results.

6-bit synchronous 2-pipeline parallel adder waveform 50ns clock:



10ns clock:



In this design the 50ns clock speed with 100ns delay is producing the appropriate outputs with almost no noise where the 10ns clock speed produces multiple erroneous outputs. In this case the 50ns clock speed appears appropriate.