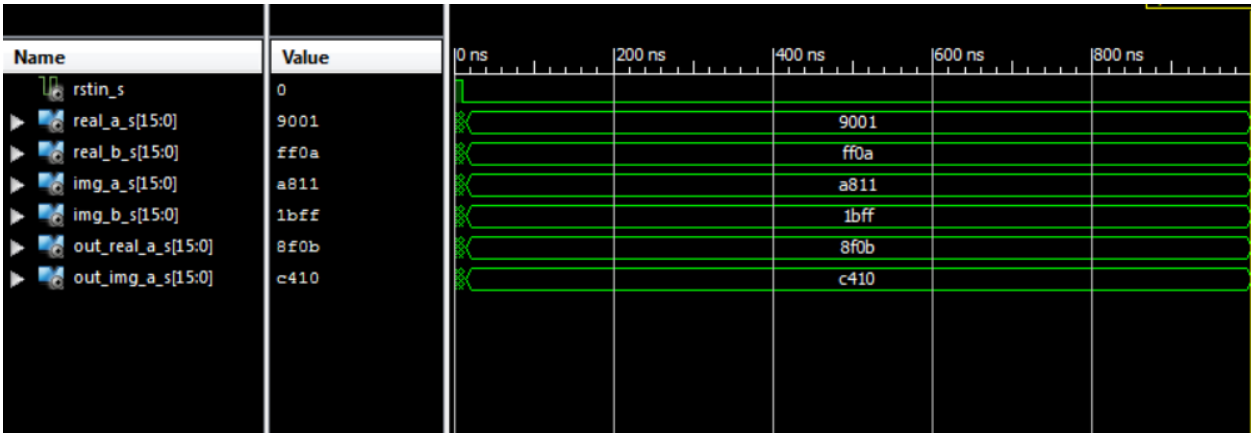
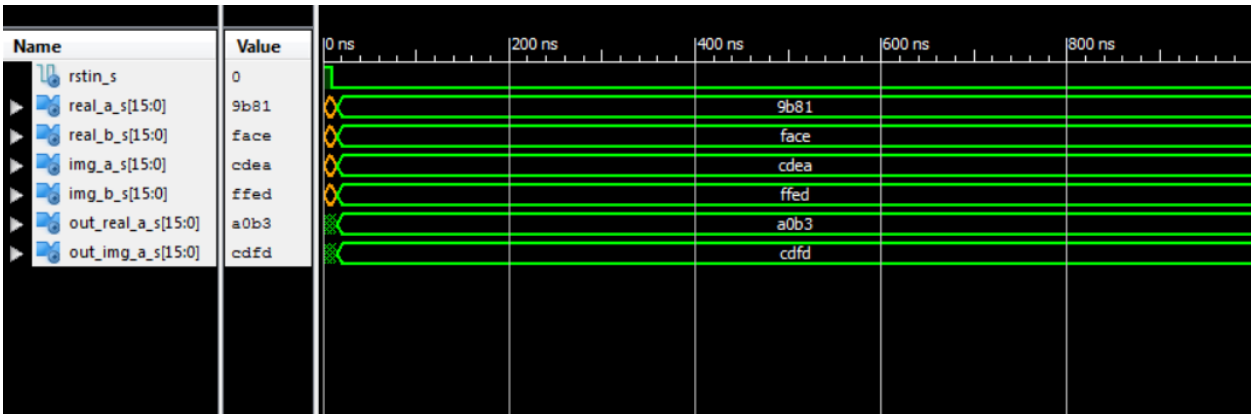
VHDL Simulations

Adder



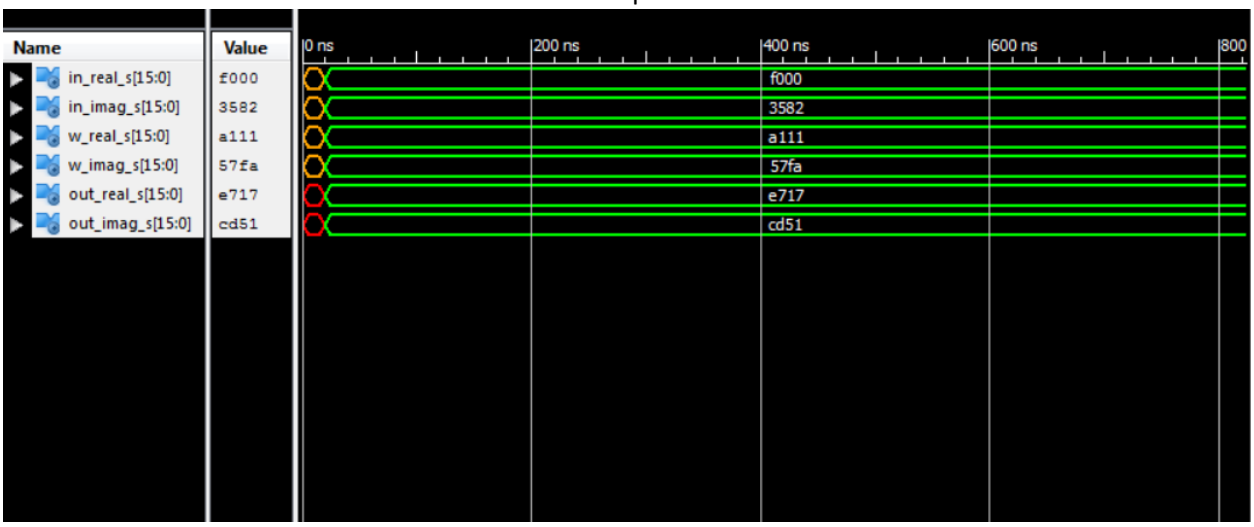
The adder adds the two inputs, creating a separate output for the real and imaginary inputs. The adder **does not** operate on a clock cycle.

Subtractor



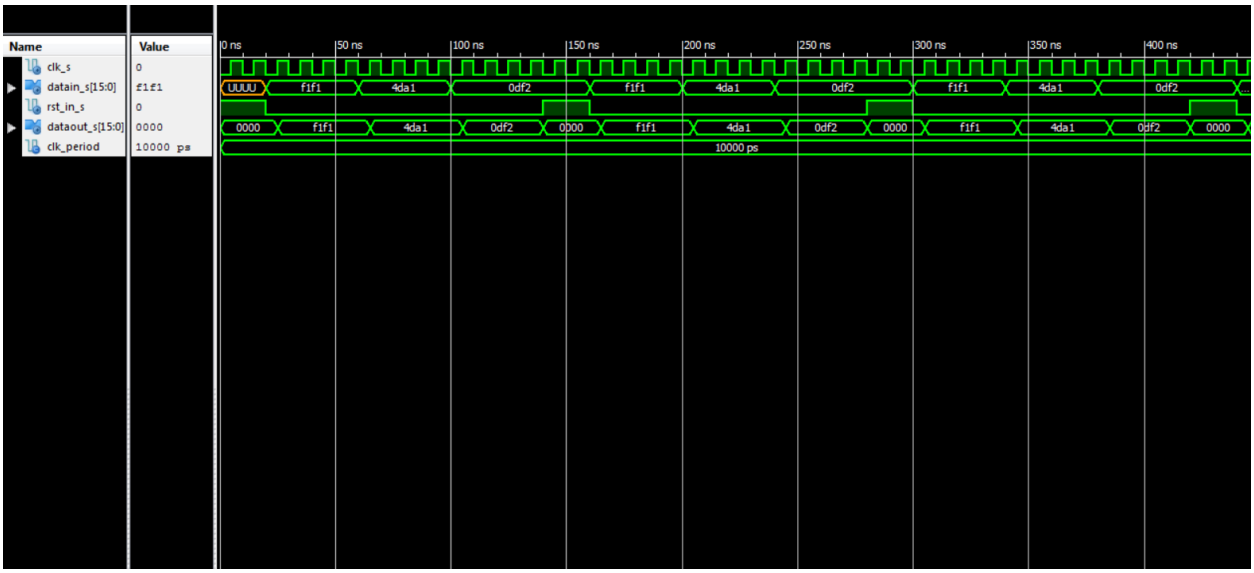
The subtractor finds the difference between the two inputs (a – b), creating a separate output for the real and imaginary inputs. The subtractor **does not** operate on a clock cycle.

Multiplier



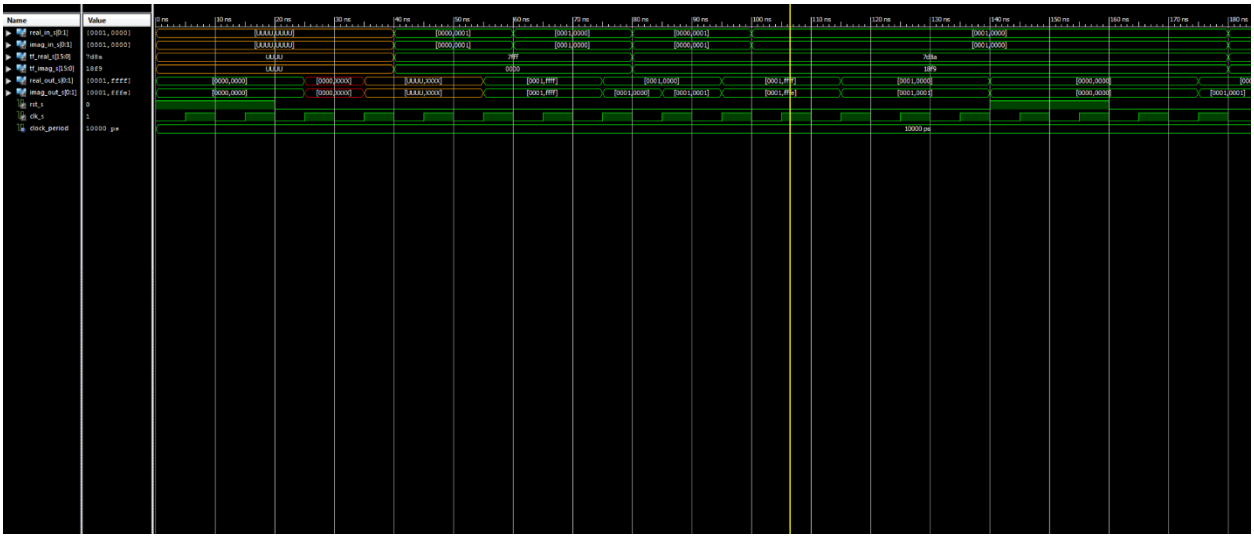
The multiplier finds the product between the two inputs (a – b), creating a separate output for the real and imaginary inputs. The subtractor **does not** operate on a clock cycle.

Register



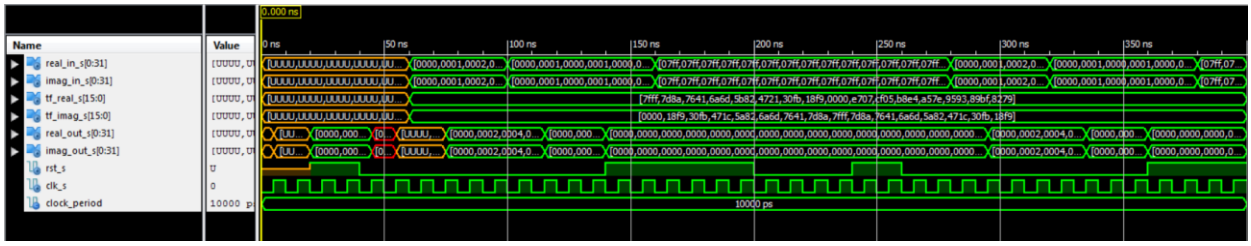
The register passes the provided input after 1 clock cycle has passed. The register is used to create a clock cycle that the entire FFT can run on.

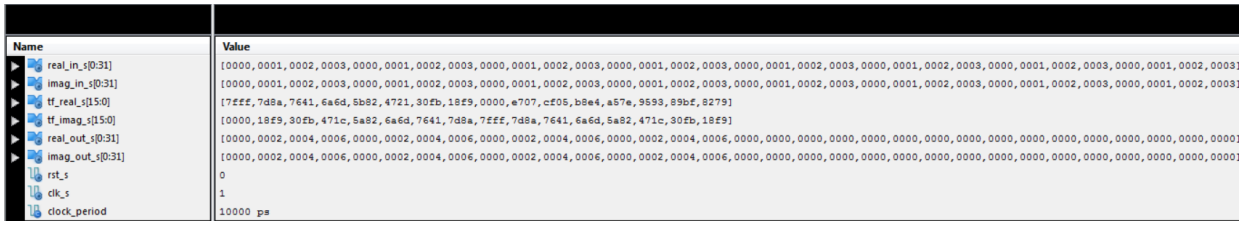
Single_DFT



The SINGLE DFT performs a single Butterfly Cross operation. The Single DFT takes two clock cycles to perform, and can operate on a reset. The outputs are delayed by two clock cycles. The unknowns in the simulation for the input are because the input has not been declared yet.

Stage 1



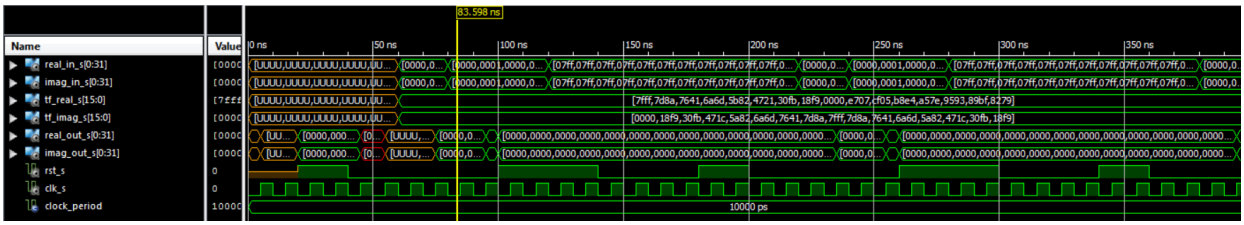


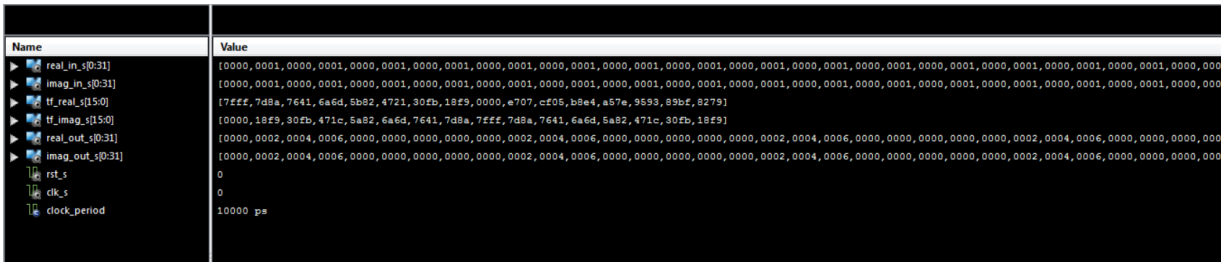
The Stage 1 Entity performs 16 Butterfly Crosses (using single dft) in parallel. Since it is acting in parallel. It only takes a full 2 clock cycles to perform (the same as single dft). The unknowns in the input in the simulation are due to the fact that the inputs have not been declared yet. Where the outputs are 0, this is because the cross is of the same number, causing the output of the “subtracting leg” to go to 0 (this was a tool we used to verify proper operation of the entity).

Stage 2

The Stage 2 Entity performs 16 Butterfly Crosses (using single dft) in parallel; however, these are not the same 16 crosses as the Stage 1 entity. Since it is acting in parallel, it only takes a full 2 clock cycles to perform (the same as single dft). The unknowns in the input in the simulation are due to the fact that the inputs have not been declared yet. Where the outputs are 0, this is because the cross is of the same number, causing the output of the “subtracting leg” to go to 0 (this was a tool we used to verify proper operation of the entity).

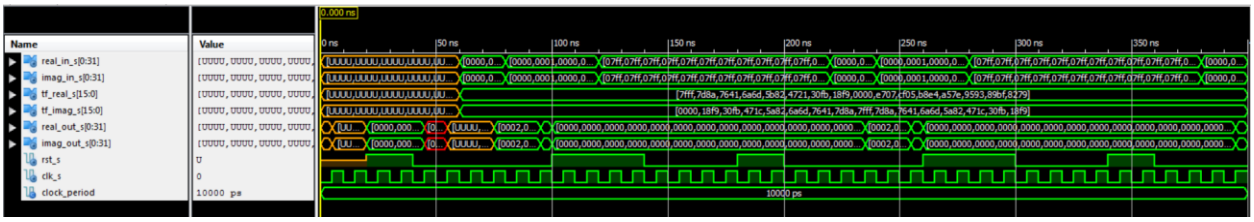
Stage 3

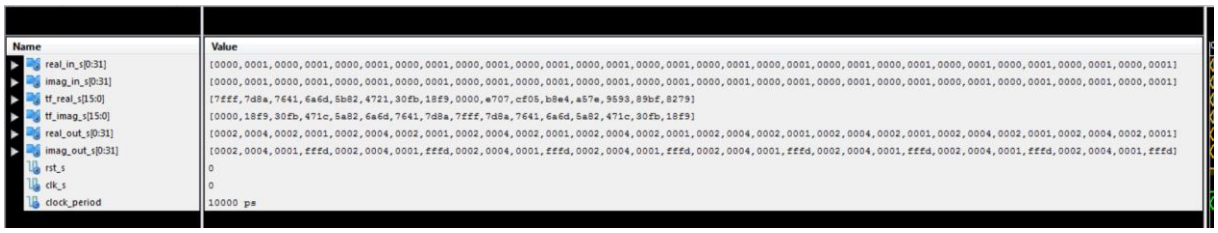




The Stage 3 Entity performs 16 Butterfly Crosses (using single dft) in parallel; however, these are not the same 16 crosses as the Stage 1 or Stage 2 entity. Since it is acting in parallel, it only takes a full 2 clock cycles to perform (the same as single dft). The unknowns in the input in the simulation are due to the fact that the inputs have not been declared yet. Where the outputs are 0, this is because the cross is of the same number, causing the output of the “subtracting leg” to go to 0 (this was a tool we used to verify proper operation of the entity).

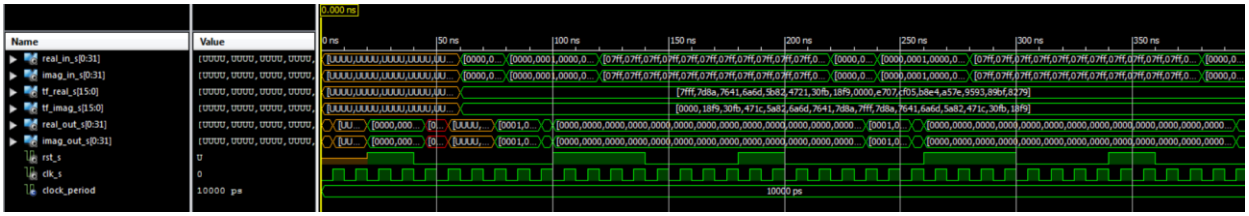
Stage 4

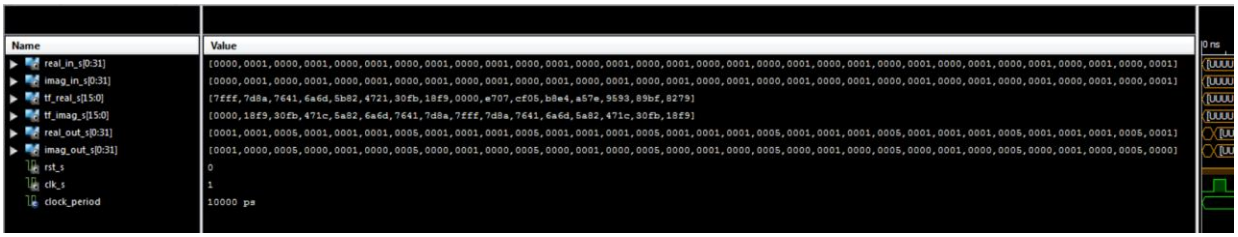




The Stage 4 Entity performs 16 Butterfly Crosses (using single dft) in parallel; however, these are not the same 16 crosses as the Stage 1 or Stage 2 entity. Since it is acting in parallel, it only takes a full 2 clock cycles to perform (the same as single dft). The unknowns in the input in the simulation are due to the fact that the inputs have not been declared yet. Where the outputs are 0, this is because the cross is of the same number, causing the output of the "subtracting leg" to go to 0 (this was a tool we used to verify proper operation of the entity).

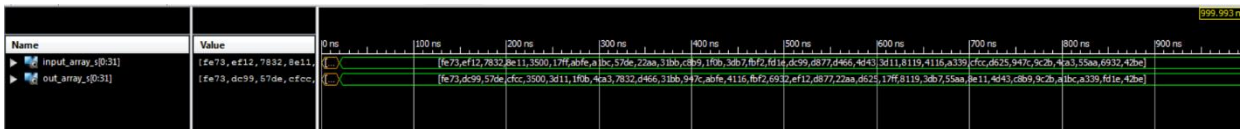
Stage 5





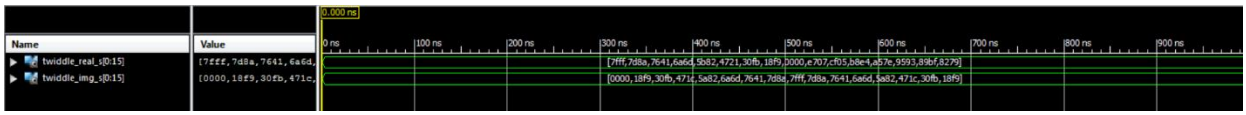
The Stage 5 Entity performs 16 Butterfly Crosses (using single dft) in parallel; however, these are not the same 16 crosses as the Stage 1 or Stage 2 entity. Since it is acting in parallel, it only takes a full 2 clock cycles to perform (the same as single dft). The unknowns in the input in the simulation are due to the fact that the inputs have not been declared yet. Where the outputs are 0, this is because the cross is of the same number, causing the output of the "subtracting leg" to go to 0 (this was a tool we used to verify proper operation of the entity).

Swapper



The swapper entity swaps the outputs to output the Full FFT response in a linear fashion as expected. The swapper **does not** operate on a clock cycle.

Twiddle



The twiddle factor takes in no inputs and was simulated for demonstration purposes only. The Twiddle can not be tested and synthesized as it only outputs twiddle factors. The decision was made to create a single entity, or lookup table, to output all twiddle factors from a single “lookup table.”

Full FFT – With Working Output, but unverified accuracy

The Full FFT operates on 10 clock cycles, and is the cascade collection of each Stage and the Swapper. The Full_FFt is the top level view of the 32 Bit FFT project as a whole. Where the outputs are 0, this is because the cross is of the same number, causing the output of the “subtracting leg” to go to 0 (this was a tool we used to verify proper operation of the entity).