EE287 Project S'18
256 point complex FFT

This semester, you are designing a 256 point complex FFT. The inputs are 20 bits of real, and 20 bits imaginary data. The binary point is 18 bits from the right. The inputs are ranged between -1 and +1. The interface has a start signal, with the first input data point. This is followed by 255 samples of complex data.

An FFT and IFFT require a scaling factor. This is traditionally placed on the IFFT. This project places the scaling (/256) on the FFT output. This implies that the range of the output will also be between -1 and +1.

The connections are:

Name	Bits & dir	Comment
clk	1 in	A positive edge clock
reset	1 in	An active high reset. (You should reset all flip flops)
realin	20 in	A 2.18 2's complement real input value
imagin	29 in	A 2.18 2's complement imaginary input value
startin	1 in	Indicates the start of a flow of 256 inputs (Includes an input)
realout	20 out	A 2.18 2's complement real output value
imagout	20 out	A 2.18 2's complement imaginary output
startout	1 out	Indicates the start of 256 continous cycles of FFT results (Includes an output)

You will need to research how the FFT works, and create code to perform the FFT. The test bench allows an error of +/- 3 lsb. The data is computed in floating point, and converted to scaled integer. More bits are required in internal calculations than the number input and output.

You are allowed 256 clocks to complete one FFT. The FFT requires 8 layers of 128 butterfly operations. This is 1024 butterfly operations. To complete in time, you must have 4 butterfly operations happening concurrently. Each butterfly operation requires a complex multiply. The complex multiply requires 4 actual multiplies. Your design is limited to 4 butterfly operations of 4 multipliers each. The total design. (outer level, and all included hierarchies) must not have more than 16 multiplies.

The design must synthesize and run at the gate level at 300 MHz. (3.3ns cycle time)

This design is big. Start early, and leave enough time for long synthesis runs.

The design code will be checked for copying. Any copying will results in a score of 0. I make no determination as to who copied who. Don't share your code.