Project 4 Fast Fourier Transform

Name: Arunkumar Ravichandran and Debosmit Majumder
PID : A53244124, A53242244
Email: arravich@eng.ucsd.edu, debosmit.majumdar30@gmail.com

Question 1.

Implemented FFT Module and QPSK Decoder and integrated them into an OFDM receiver. **Refer to designs "fft1024_baseline " , "fft1024_best" and "OFDM_receiver".**

The software version of the FFT was implemented. The software version had nested loops and the loops had variable bounds so, the latency and the interval were shown by "?" on synthesis.

So, fft first stage and fft last stage was implemented. And the intermediary stages were implemented using a generic function. So, the outermost loop was removed and the inner 2 loops remained. **HLS_tripcount** was used to mention manually the number of iterations each loop takes. So, that removed the "?" on synthesis.
It was observed that the outer loop took a maximum of "numBF" that is 512 iterations for stage 10 and a minimum of one iteration for stage 1.
Meanwhile, the inner loop took a maximum of 512 iterations for stage 1 and minimum of 1 iteration for stage 10. Hence, the product of inner loop and outer loop iterations was constant and equal to **512 for a 1024_pt FFT**.

The FFT was divided into multiple stages to enable task-level pipelining (dataflow) and to make it hardware friendly. The stages were bit-reversal, fft_first_stage,fft_(2-9)_stages and fft_last_stage. Also, to enable **dataflow** pipelining to be used, the "reverse_bits" function interface was modified and input and output interfaces were separated. The FFT functions were also modified to enable a single-producer single-consumer data-flow. Temporary variables like "Y_R" and "Y_I" were added to enable the above kind of data-flow. This enabled to execute all the stages in parallel. The final throughput hence depended on the throughput of the slowest stage.
Meanwhile, the directive "**HLS pipeline**" was added, which unrolled the loops. It was observed to achieve an initiation interval of 1 among the fft_stages. Each of the fft_stages took only 524 cycles to complete. The throughput was determined by the slowest stage which was "bit-reversal". It achieved an initiation interval of 3 due to interdependency between iterations and a trip count of 1024.
The "**inline**" directive was also used in bit_reversal(for "**reverse_bits**") but it didn't show any effect because the HLS tool was already performing in-lining of the function.

The **final throughput** was found to be → 1/(3075 * 8.70 * 10^-9) → **37.379 kHz.**