WES237C: Project 4 FFT and OFDM Receiver

Yi Song
yis057@ucsd.edu
PID# A53266440

Sebastian Nevarez
senevarez@ucsd.edu
PID# A59021322

---

1. Fast Fourier Transform (FFT)

Our 1024-point FFT design starts with a bit reverse function to change the order of the input IQ samples so that the output of the FFT will be in an increasing numerical order. We then set up 10 stages of the FFT algorithm to run sequentially to generate the output. We separated the first and last stage since we know the bound and so we can optimize further in this aspect.

The first optimization we did was to separate the input and output matrix of each function. By doing this we avoid data being read and written to the same matrix during the same cycle. Additionally, this decreases the II from 2 to 1 which we used `#pragma pipeline II=1` to do.

The next optimization done was to make VITIS understand our loop bounds. The majority of the code runs in nested for loops with the outer loop iterating butterfly times and the inner loop iterating 1024 times. As the outer loop varies with FFT stages, it was a good idea to flip the fixed bound of the inner loop and the varying bound of the outer loop, so VITIS has a better idea of how long the loop is going to run. At this stage of development, we noticed that the latency for the bit-reverse function was over 2 times that of the FFT stages.

Finally, using separate inputs and outputs in the bit-reverse function, we are able to remove memory swapping and reduce the II of the stage to 1. This improved the latency of all functions to about 1000 or less, achieving a throughput of 1000/(7374*10ns) = 13.56MHz. See Figures 1 & 2 below for screenshots from our synthesis.

Timing Estimate

Target	Estimated	Uncertainty
10.00 ns	7.300 ns	2.70 ns

Performance & Resource Estimates

Modules Loops

Modules & Loops	Issue Type	Violation Type	Distance	Slack	Latency(cycles)	Latency(ns)	Iteration Latency	Interval	Trip Count	Pipelined	BRAM	DSP	FF	LUT	URAM
fft				-	7374	7.374E4	-	1049	-	dataflow	98	213	28503	42817	0
entry_proc				-	0	0.0	-	0	-	no	0	0	2	29	0
bit_reverse				-	1035	1.035E4	-	1035	-	no	0	0	236	432	0
fft_stage_first				-	1041	1.041E4	-	1041	-	no	0	9	1464	1852	0
fft_stages_1				-	530	5.300E3	-	530	-	no	2	24	2625	3921	0
fft_stages_2				-	530	5.300E3	-	530	-	no	2	24	2626	3924	0
fft_stages_3				-	530	5.300E3	-	530	-	no	2	24	2627	3927	0
fft_stages_4				-	530	5.300E3	-	530	-	no	2	24	2628	3929	0
fft_stages_5				-	530	5.300E3	-	530	-	no	2	24	2629	3931	0
fft_stages_6				-	530	5.300E3	-	530	-	no	2	24	2630	3933	0
fft_stages_7				-	530	5.300E3	-	530	-	no	2	24	2631	3935	0
fft_stages				-	530	5.300E3	-	530	-	no	2	24	2632	3935	0
fft_stage_last				-	1048	1.048E4	-	1048	-	no	2	12	2335	2741	0

Figure 1: Performance and Resource Utilization

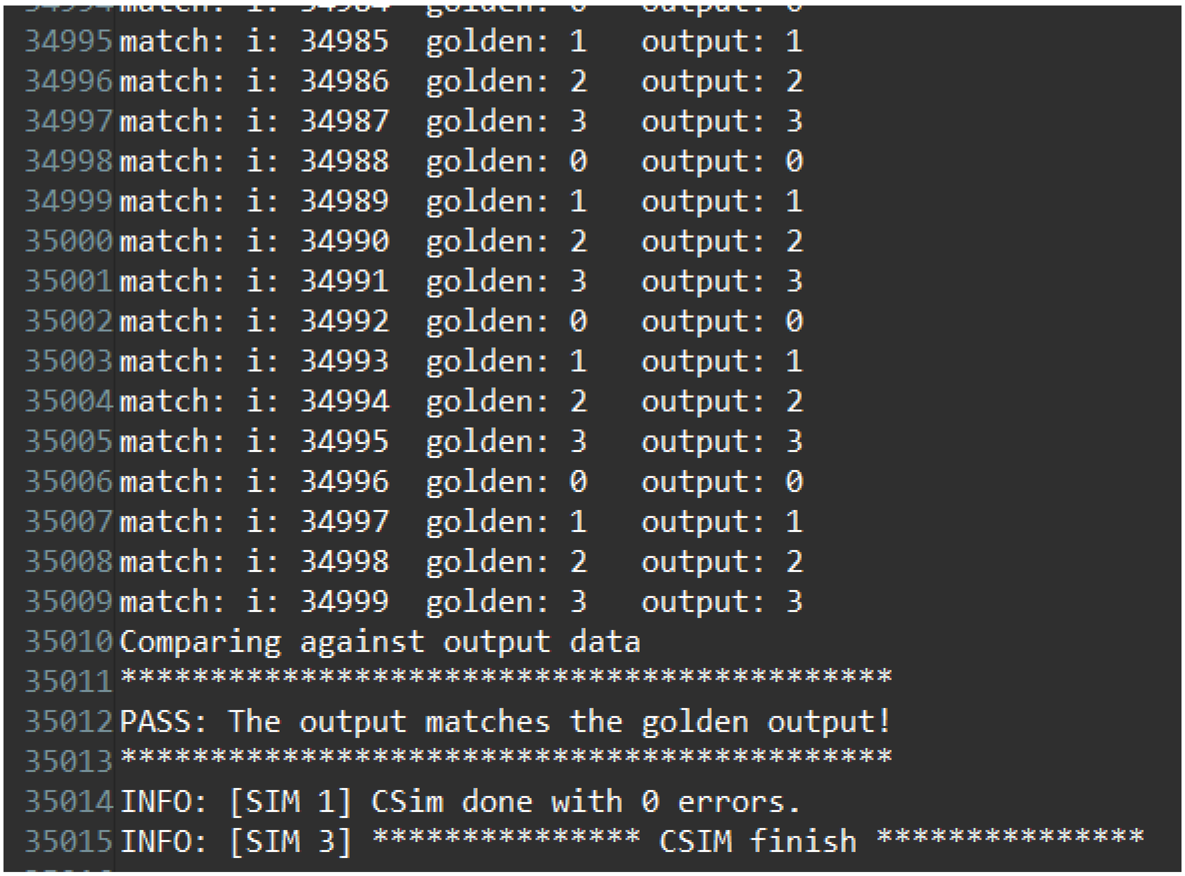
Summary

Name	BRAM_18K	DSP	FF	LUT	URAM
DSP	-	-	-	-	-
Expression	-	-	0	132	-
FIFO	-	-	198	134	-
Instance	18	213	28283	42353	-
Memory	80	-	0	0	0
Multiplexer	-	-	-	198	-
Register	-	-	22	-	-
Total	98	213	28503	42817	0
Available	280	220	106400	53200	0
Utilization (%)	35	96	26	80	0

Figure 2: Resource Utilization Summary

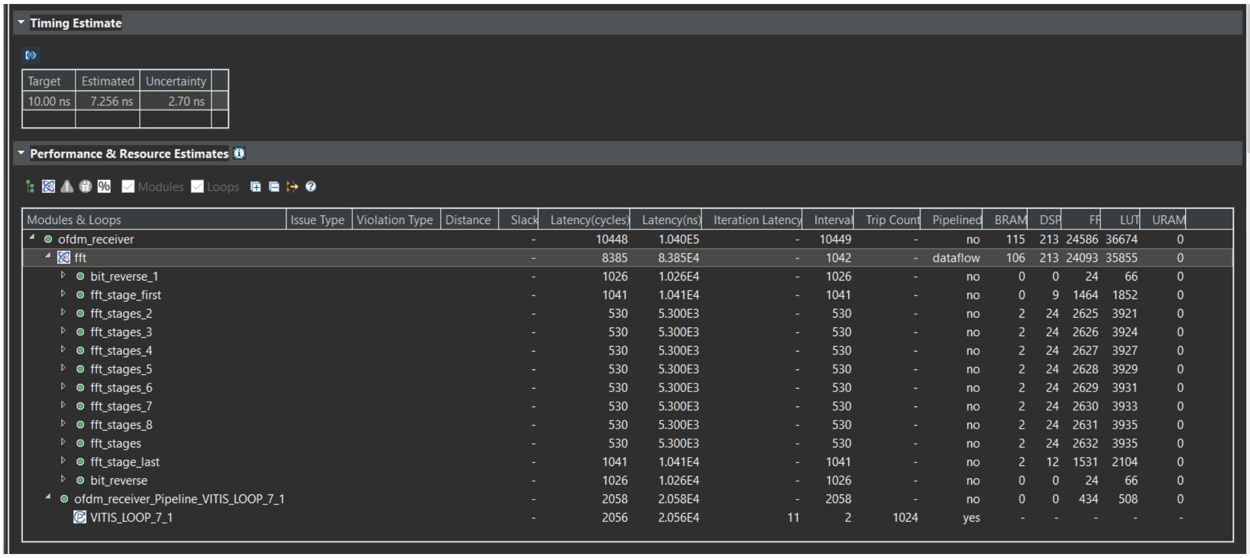
2. OFDM Receiver

Similar to the FFT project, we started with the FFT bit-reverse function and FFT stages function copied to the OFDM project. The next thing we needed to do to complete the QPSK decode function was to decode the IQ samples into messages (integer values from 0 to 3). We created four if statements to check which quadrant the samples fall into and assigned different integer values to the samples that fell into each quadrant. The throughput calculation for our OFDM receiver is 1000/(7374*10ns) = 9.57MHz. See Figures 3 & 4 below for screenshots from our synthesis.



```
match: i: 34985  golden: 1   output: 1
match: i: 34986  golden: 2   output: 2
match: i: 34987  golden: 3   output: 3
match: i: 34988  golden: 0   output: 0
match: i: 34989  golden: 1   output: 1
match: i: 34990  golden: 2   output: 2
match: i: 34991  golden: 3   output: 3
match: i: 34992  golden: 0   output: 0
match: i: 34993  golden: 1   output: 1
match: i: 34994  golden: 2   output: 2
match: i: 34995  golden: 3   output: 3
match: i: 34996  golden: 0   output: 0
match: i: 34997  golden: 1   output: 1
match: i: 34998  golden: 2   output: 2
match: i: 34999  golden: 3   output: 3
Comparing against output data
*******************************************
PASS: The output matches the golden output!
*******************************************
INFO: [SIM 1] CSim done with 0 errors.
INFO: [SIM 3] *************** CSIM finish ***************
```

Figure 3: CSIM Output Verification



Timing Estimate

Target	Estimated	Uncertainty
10.00 ns	7.256 ns	2.70 ns

Performance & Resource Estimates

Modules Loops

Modules & Loops	Issue Type	Violation Type	Distance	Slack	Latency(cycles)	Latency(ns)	Iteration Latency	Interval	Trip Count	Pipelined	BRAM	DSP	FF	LUT	URAM
ofdm_receiver				-	10448	1.040E5	-	10449	-	no	115	213	24586	36674	0
fft				-	8385	8.385E4	-	1042	-	dataflow	106	213	24093	35855	0
bit_reverse_1				-	1026	1.026E4	-	1026	-	no	0	0	24	66	0
fft_stage_first				-	1041	1.041E4	-	1041	-	no	0	9	1464	1852	0
fft_stages_2				-	530	5.300E3	-	530	-	no	2	24	2625	3921	0
fft_stages_3				-	530	5.300E3	-	530	-	no	2	24	2626	3924	0
fft_stages_4				-	530	5.300E3	-	530	-	no	2	24	2627	3927	0
fft_stages_5				-	530	5.300E3	-	530	-	no	2	24	2628	3929	0
fft_stages_6				-	530	5.300E3	-	530	-	no	2	24	2629	3931	0
fft_stages_7				-	530	5.300E3	-	530	-	no	2	24	2630	3933	0
fft_stages_8				-	530	5.300E3	-	530	-	no	2	24	2631	3935	0
fft_stages				-	530	5.300E3	-	530	-	no	2	24	2632	3935	0
fft_stage_last				-	1041	1.041E4	-	1041	-	no	2	12	1531	2104	0
bit_reverse				-	1026	1.026E4	-	1026	-	no	0	0	24	66	0
ofdm_receiver_Pipeline_VITIS_LOOP_7_1				-	2058	2.058E4	-	2058	-	no	0	0	434	508	0
VITIS_LOOP_7_1				-	2056	2.056E4	11	2	1024	yes	-	-	-	-	-

Figure 4: Performance & Resource Utilization