Lauren Chambers, Kenzie Milhous
Logic Design with Verilog
April 2021

Multiplier

In this folder you will find 2 implementations of an accumulation-based multiplier. The files multiplier_1.v and multipler_4.v both contain modules with 2 16-bit inputs arg1 and arg1, a 32-bit product output, and some other i/o signals. Accumulation is performed by adding arg1 to a register arg2 number of times. The multiplier_1 module accumulates 1 time per clock cycle and multiplier_4 up to 4 times. The latter module requires additional combinatorial logic feeding into the accumulator register.

Timing and Resource Analysis: multiplier_1	
Total runtime computing 67*43	435 ns
Specified clock period	100 ns
Worst Negative Slack **WNS**	98.286 ns
Minimum clock period **Tmin**	1.714 ns
Maximum clock frequency **Fmax**	583 MHz
# look up tables **LUT**	17
# of registers	32

Timing and Resource Analysis: multiplier_4	
Total runtime computing 67*43	115 ns
Specified clock period	100 ns
Worst Negative Slack **WNS**	97.087 ns
Minimum clock period **Tmin**	2.913 ns
Maximum clock frequency **Fmax**	343 MHz
# look up tables **LUT**	109
# of registers	66

Lauren Chambers, Kenzie Milhous
Logic Design with Verilog
April 2021

There is a tradeoff for each design: Multiplier_1 uses less hardware so it can be clocked faster (1.7ns versus 2.9 ns). Multiplier_4 uses extra hardware and must be clocked slower but uses roughly ¼ of the clock cycles that multiplier_1 does.