High Speed CMOS VLSI Design

Project Description

(c) 1998 David Harris

You've been hired to design and implement a high speed floating point multiplier for the AlpahNot processor. Your team is responsible for the circuit design of the "mantissa path" of the multiplier.

1: Teams

You should form teams of 2 engineers for the project. If you want to work in groups of 3, please talk with me.

2: Goal

The primary goal is to build the fastest, i.e. minimum latency, multiplier. You can trade off other parameters in order to minimize latency.

There are two secondary objectives: layout area and active power. The less, the better. You will present your design to the class and will need to convince them you have reasonable numbers. Make sure you have a robust design and have done a credible job of estimating circuit performance.

3: Project Specification

3.1 Circuit Technology

You should simulate your project in the HP26 0.8 micron technology you have used in previous problem sets. Use $V_{DD} = 3.3$ and remember $\lambda = 0.5\mu m$. As you found in Problem Set 2, the Fanout-of-4 (FO4) inverter delay in this process is 427ps in TT process and environment. You may use any circuit families and clocking style you want.

3.2 Function

Your circuit design only must handle the fraction-part (the "mantissa") of IEEE-standard floating point numbers. Somebody else is responsible for exponent manipulation. Also, pretend that you only must handle “double” precision numbers; you do not need to worry about single precision.

If you choose to design your multiplier such that a single carry-propagate add is performed at the very end of the multiplication, you don’t need to design the CPA. Assume someone else will build it and bolt it onto your multiplier. Assume that it consumes an additional clock cycle, which includes the addition, IEEE rounding, final shifts, etc. You need to provide the CPA all the signals necessary to perform the operation correctly. This means that you will need to provide some additional information for rounding, like one extra bit and a sticky bit.

A Verilog description of a simple multiplier is on the class web page. This will give you more information about the features you need to get rounding correct.

While you are not required to create a working Verilog description of your multiplier, it will provide you greater confidence that you have connected the components together in a way that provides the correct answer. It also gives you some extra credit points. You can use the test fixtures provided with the sample code to check your functionality.

3.3 Timing Specification

Your multiplier should take 2 clock cycles to operate, not counting the extra cycle for the CPA and rounding. Your objective is to minimize the clock period in the TT corner. You may assume that the 53 bit A and B inputs are set up well before the first cycle and may drive an arbitrary load. You must include appropriate latching or memory elements, as given by the timing methodology you choose. Your outputs must drive 8/16 inverters.

You may receive any number of clocks of any frequency and duty cycle you want. That could include 4-phase skew-tolerant domino waveforms or pulses for pulsed latches. All clocks have edge rates (0-100%) of 500ps. Your design must handle skews in which each clock edge may arrive up to 1 FO4 earlier than expected.

3.4 Layout Specification

The AlphaNot processor uses a typical datapath arrangement. Bits of the datapath are arrayed linearly from MSB to LSB using a common cell width called the “bit pitch.” For this datapath, the bit pitch has been set at 80 lambda. Remember that each contacted wire track requires at least 8 lambda and that you must allocate space for power and ground routing. You have 3 levels of metal available and have the luxury of choosing which directions are used for which level.

Because of the way multiplication works, you may decide you need a few “extra” bit pitches, making your multiplier mantissa datapath perhaps a little wider than the fraction part of an IEEE double. Use good engineering judgment. The die photos in Appendix A of Hennessey and Patterson might show something about this.

Project Description

Your multiplier hardware sits in the center of the overall FP datapath. The adder is on one side and the FP register file hardware is on the other side. The datapath includes three global busses, called “A bus,” “B bus,” and “R bus.” They are routed to all FP hardware units. Operands to your multiplier arrive on the A and B busses and the multiplier may send its results (a pair of numbers) to the final CPA on the A and B busses. The R bus is not used by your hardware; still you have to let the R bus “pass through” your layout! Pay careful attention to the wiring of your design so that it will fit in the datapath constraint.