Project Description

The goal of this project is to design an 8-bit adder with minimum delay-power product. No registers (i.e. pipelining) are allowed in the design of this adder.

For systematic approach, you can organize your work in two parts. In phase-1, use the design expertise you acquired in class to find the optimum adder architecture that best optimizes the speed-power goal. Do a quick sketch of several feasible options and figure out the best architecture and circuit style. You may mix circuit styles if that helps. In phase-2, first implement the block-level schematic of the adder and verify the functionality in HSPICE. Then, identify critical path and optimize sizing for minimum delay. In the critical path evaluation, you need to determine not only the gates along the path, but also the input operands that cause worst-case delay between input and output bits. Think of the area as a way of choosing optimal circuit topology. Once you do so, you don’t need to back-off in timing to save area (maybe a chance for some extra credit).

1: Choosing Adder Topology / Circuit Style (1 week)
   a) Determine adder topology that optimizes delay-power metric.
   b) Choose logic style for the implementation. You may mix several logic families.

2: Critical Path Delay Optimization (2 week)
   a) Check functionality of the adder in HSPICE.
   b) Identify input vectors that will exercise critical path. Size the gates for minimum delay.
   c) Verify the critical path delay in HSPICE under worst-case input operands.

3: Final Report (1/2 week)

Prepare a concise (~500-1000words, 1.5 line spacing) summary of your design including your design methodology, the reasons for selecting the adder topology, and how you sized the gates to maximize performance. Summarize key specifications, preferably in a table format to highlight design performance. Include simulation results that clearly demonstrate that the design meets performance requirements (especially critical path timing). Include the schematic of the
basic cells in your design (i.e. full adder cell - with sizing clearly labeled) and the top level architecture. Schematics may be drawn by hand if necessary. Include your netlist and any test files.

**Constraints (READ CAREFULLY!)**

a) **Supply voltage:** maximum of 2.5V

b) **Implementation choices:**
   i. Use only static logic (e.g. CMOS, pass-transistor logic, …). You can use dynamic logic for 10% extra credit (provided that your overall optimization goal is met).

c) **Input operands:**
   i. Both operands are 8-bit numbers. There is an incoming carry at bit position zero.

d) **Loading conditions:**
   i. The input capacitance of all inputs is less than equal to 2 unit sized inverters (per bit). For simulation purposes, the inputs to your adder are driven by a unit sized buffer (chain of two unit sized inverters). The delay is measured as the delay after the input driver (2 inverters) to before the load (16 inverters).
   
   ii. Each sum bit and final carry out at MSB is loaded with $C_L = 16$ unit sized inverters. Implement this load with inverters. **Also add another load of 64 at the output of load gates to suppress Miller kick-back.**
   
   iii. Unit sized inverter is $W_p = 0.96\mu m$, $W_n = 0.48\mu m$, $L_p = L_n = 0.24\mu m$.

**HAVE FUN!**