High Speed CMOS VLSI Design

Problem Set 3

1.0 Domino Charge Sharing

Consider a domino gate with a clocked pulldown transistor which computes NOT(A & (B | (C & (D | (E & (F | G)))))). We call this an AOAOAOI gate and frequently encounter it in fast adders.

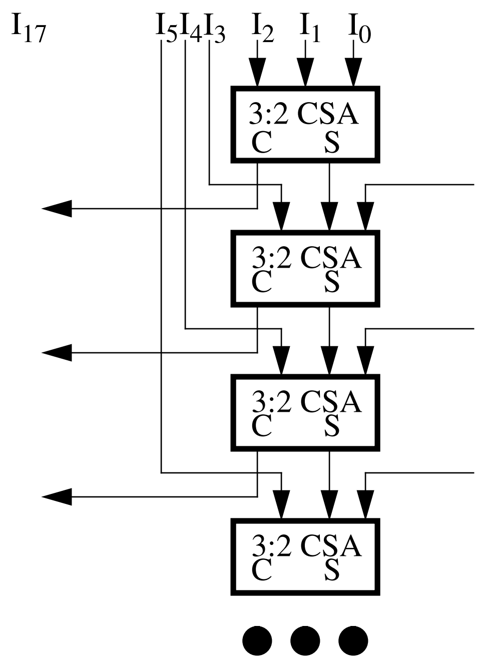
a) Sketch a domino gate which implements this function. Let the largest NMOS transistor be 48 λ and size transistors so the worst case drive strength is equal through any series path. Remember to convert $\lambda = 0.5\ \mu m$ when entering your SPICE deck!

b) Model the gate in SPICE. Assume that every transistor has 6 λ of source and 6 λ of drain diffusion and that adjacent transistors do not share diffusion (this is conservative, but good layout of a complex gate like this is tricky). Let the output drive an inverter with 48f λ of input capacitance. What is the worst case pattern of inputs for charge sharing? If all of the internal nodes were low, what is the percentage drop in output voltage due to charge sharing for f=1, 2, 4, and 8? Use the .ic statement to set the initial voltage on the nodes. What is the worst case delay for f=2 (remember worst case delay occurs when internal nodes were high)?

c) Redesign the gate with secondary precharge devices on intermediate nodes where necessary such that the worst case charge sharing reduces the output voltage by 15%. Size the secondary devices to precharge each node to 95% of VDD in less than 4 FO4 delays. Now, what is the worst case delay? How much did the secondary precharge increase delay?

2.0 CSA Design

A 3:2 Carry Save Adder (also known as a full adder) adds three single bit inputs (A, B, and C_{in}) to produce a two-bit output {C_{out}, Sum}; hence the 3:2 designation. The multiplier in the AlphaNot Processor uses a linear array of 3:2 CSAs to add up partial products. We'll study multipliers later, but for now, assume the AlphaNot uses Radix 8 Booth encoding and must therefore sum 18 inputs {I_0, ..., I_{17}} using 16 cascaded CSAs. One bitslice of the array is shown below. A bunch of binary values are presented at the top of the array for summation. The carry out of each adder is sent to the next more significant bit, while the carry in from the previous bit is received. Note that we didn't label which arrow goes to which input (A, B, or Cin). You can optimize your array such that the slowest output is connected to the fastest input.



Your job is to design 3:2 CSAs suitable for the multiplier array. You should provide outputs suitable for cascading into another identical CSA (i.e. you can use true and complementary inputs if you want, but in such case must produce true and complementary outputs). Exception: if you want, you may design two CSAs; one takes true inputs and produces complementary outputs; the second takes complementary inputs and produces true outputs; in such a case, measure your average delay as half the total delay through the two CSAs. The I inputs are driven by 16/8 inverters (PMOS = 16 λ, NMOS = 8 λ) and the final outputs must also drive 16/8 inverters.

Optimize the worst case delay from I transitioning to the multiplier output changing. For each of the parts below, design a 3:2 CSA using the requested circuit family. Simulate your delay in HSPICE and report the worst case delay and power through the multiplier array bitslice (all 16 CSAs). You may prefer to simulate only one or two CSAs and estimate the delay of the total path from that information. You may use a schematic editor in your design, but be sure that you include realistic diffusion capacitance in the SPICE netlist.

We have attached some published CSA designs to the problem set for your reference. You may also wish to refer to Weste or to other adders with which you are familiar.

a) Design the adder using static logic (static CMOS, transmission gates, pseudo-NMOS, current-mode logic, or any other unclocked logic family).

b) Design the adder using domino gates. Remember to include precharge power in your total power budget.

There will be prizes for the fastest convincing static and domino adders.