Digital Integrated Circuits – A Design Perspective 2/e

Jan M. Rabaey, Anantha Chandrakasan, Borivoje Nikolić

Chapter 11

Design Project: Divider Design and Optimization

The goal of this project is to design an 8-bit binary divider with **minimum overall delay it takes to perform a divide operation (see also supplemented notes)**. This delay is comprised of the number of clock cycles needed to perform a computation multiplied by the clock period, Delay = $N_{cycles} \cdot T_{Clk}$. Your task is to find optimal compromise between N_{cycles} and T_{Clk} in such a way as to minimize the overall delay of the divide operation.

As in the first project, work in a group of 2 students. If you have specific problems finding a partner, please let Prof. Rabaey or the TAs know. No need to sign up.

For systematic approach, you can organize your work in two phases. In phase-1, use the design expertise you acquired in the class to find your optimum divider architecture. Then implement the block-level schematics of the divider in SUE and verify the functionality in IRSIM or HSPICE (preferably IRSIM). In phase-2, identify the critical path of the divider, optimize and size the critical path for minimum delay and verify in SPICE. 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 register files. Below is more detailed explanation of the steps you need to take to ensure the success of your project.

Phase 1: Schematic and functional verification (1.5 weeks)

a) **Design the divider architecture with minimum overall delay Delay = $N_{cycles} \cdot T_{Clk}$.**

b) **Implement the block level divider schematic in SUE.**

c) **Verify the design functionality in IRSIM with the provided input pattern: Dividend = 0000 0111, Divisor =0000 0010. (focus on four LSBs)**

Your final presentation must include:

- *Divider architecture with clear explanation of your design methodology*
- *Picture of the divider schematic in SUE*
- *IRSIM (or HSPICE) simulation waveforms of quotient and remainder outputs (if you have real problems with IRSIM, use SPICE showing the 4 LSBs)*

Phase 2: Critical Path Delay Optimization (1 week)

a) **Size the critical path for minimum delay.**

b) **Implement the critical path of your design in SPICE.**

c) **Verify the critical path delay under worst-case input operands (which you need to identify).**

Your final presentation must include:

- *Block diagram of the critical path with gate-level schematics of underlying circuits*
- *Critical path sizing methodology and results of sizing for minimum delay*
- *SPICE simulation waveforms of quotient and remainder output, with critical path delay annotated clearly.*

Phase 3: Sleep (8 hours)

The purpose of this task is to ensure that you will be able to clearly present your work! The goal is to get through few REM cycles and come to your presentation relaxed. You do not have to minimize the number of REM cycles multiplied by the time you spend in each.

Phase 4: Poster Presentation (10')

Prepare a 9-slide poster (template to be provided soon) representing your effort. Sign up for a 10' slot. Present your results to the faculty or TAs. Be crisp: show what your main decisions have been, explain why they are the best thing in the world, and prove that they really worked out (or did not).

Constraints (READ CAREFULLY!)

a) **Supply voltage: 2V**

b) **Implementation choices:**

   a. Use only static logic (CMOS, pass-transistor logic, …). NO dynamic gates.

   b. Use the register shown in f). You can adjust it to become a shift register as well.

   c. **You are to use AT MOST 4 adder modules (that is, N-bit adders) in your design.**

c) **Input operands:**

   a. Dividend and divisor are 8 bits wide (based on this, you need to determine the width of the remainder and quotient)

   b. Both dividend and divisor are positive (leading zero)

d) **Clock waveform**

a. Input rise and fall times of the clock waveform are equal $t_{rise} = t_{fall} = 100ps$

b. Clock swing is from 0 to 2V

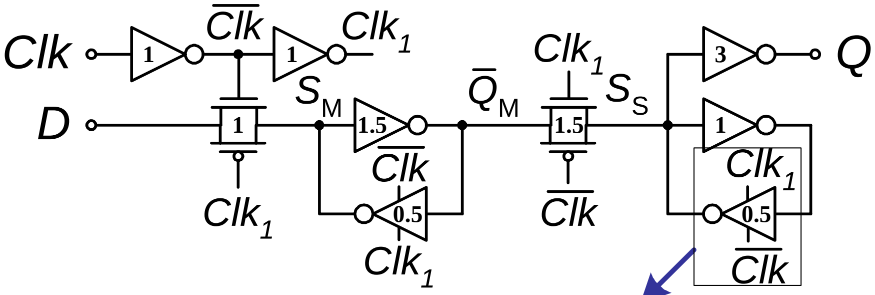
e) Loading conditions

a. The input capacitance of your divisor and divider is C_{in} = 1 unit sized inverter (per bit)

b. Unit sized inverter is Wp = 2μm, Wn = 1μm, Lp = Ln = 0.25μm

c. Each bit of the quotient and reminder is loaded with C_L = 25 unit sized inverters (as you already know, you need extra load to the load to suppress Miller effect)

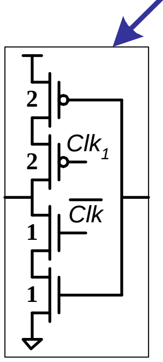
f) Register cell with annotated sizes in terms of unit inverters is given below.



Gate sizes are normalized to unit inverter and indicate drive strength relative to the unit inv. (W_p/W_n=2/1).

Shown on the right is transistor-level implementation of the gated inverter.

Unit transmission gate is W_p=W_n=1μm.



HAVE FUN!