

Chapter 11 PROBLEMS

1. [E, None, 11.6] For this problem you are given a cell library consisting of full adders and two-input Boolean logic gates (i.e. AND, OR, INVERT, etc.).
 - a. Design an N -bit two's complement subtracter using a minimal number of Boolean logic gates. The result of this process should be a diagram in the spirit of Figure 11.5 . Specify the value of any required additional signals (e.g., C_{in}).
 - b. Express the delay of your design as a function of N , t_{carry} , t_{sum} , and the Boolean gate delays (t_{and} , t_{or} , t_{inv} , etc.).
2. [M, None, 11.6] A magnitude comparator for unsigned numbers can be constructed using full adders and Boolean logic gates as building blocks. For this problem you are given a cell library consisting of full adders and arbitrary fan-in logic gates (i.e., AND, OR, INVERTER, etc.).
 - a. Design an N -bit magnitude comparator with outputs $A \geq B$ and $A = B$ using a minimal number of Boolean logic gates. The result of this process should be a diagram in the spirit of Figure 11.5. Specify the value of any required control signals (e.g., C_{in}).
 - b. Express the delay of your design in computing the two outputs as a function of N , t_{carry} , t_{sum} , and the Boolean gate delays (t_{and} , t_{or} , t_{inv} , etc.).
3. [E, None, 11.6] Show how the arithmetic module in Figure 0.1 can be used as a comparator. Derive an expression for its propagation delay as a function of the number of bits.

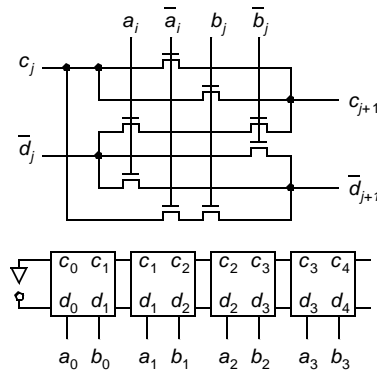


Figure 0.1 Arithmetic module.

4. [E, None, 11.6] The circuit of Figure 11.2 implements a 1-bit datapath function in dynamic (precharge/evaluate) logic.
 - a. Write down the Boolean expressions for outputs F and G . On which clock phases are outputs F and G valid?
 - b. To what datapath function could this unit be most directly applied (e.g., addition, subtraction, comparison, shifting)?
5. [M, None, 11.3] Consider the dynamic logic circuit of Figure 0.2 .
 - a. What is the purpose of transistor M_1 ? Is there another way to achieve the same effect, but with reducing capacitive loading on the clock Φ ?

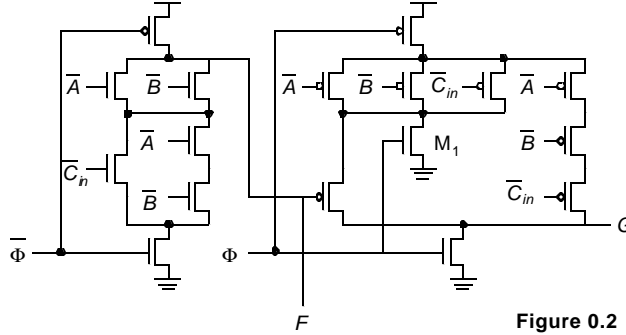


Figure 0.2 Datapath module bit-slice.

- b. How can the evaluation phase of F be sped up by rearranging transistors? No transistors should be added, deleted, or resized.
- c. Can the evaluation of G be sped up in the same manner? Why or why not?
6. [M, SPICE, 11.3] The adder circuit of Figure 0.3 makes extensive use of the transmission gate XOR. $V_{DD} = 2.5$ V.
 - a. Explain how this gate operates. Derive the logic expression for the various circuit nodes. Why is this a good adder circuit?
 - b. Derive a first-order approximation of the capacitance on the C_o -node in equivalent gate-capacitances. Assume that gate and diffusion capacitances are approximately identical. Compare your result with the circuit of Figure 11-6.
 - c. Assume that all transistors with the exception of those on the carry path are minimum-size. Use 4/0.25 NMOS and 8/0.25 PMOS devices on the carry-path. Using SPICE simulation, derive a value for all important delays (input-to-carry, carry-to-carry, carry-to-sum).

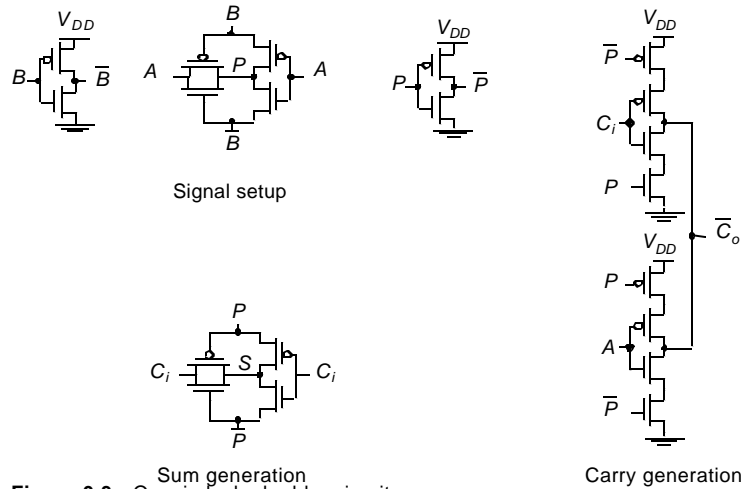
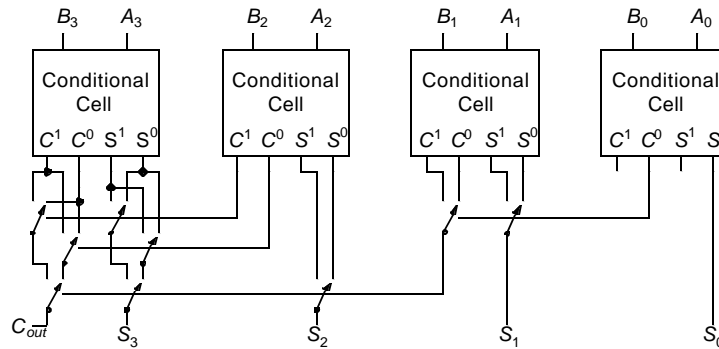
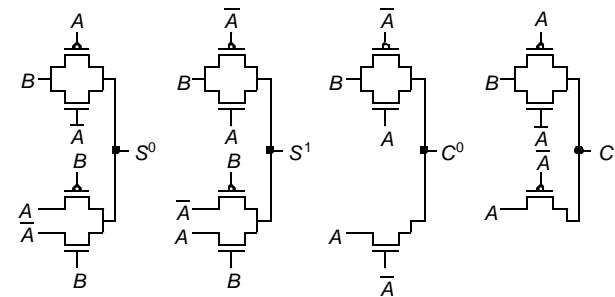


Figure 0.3 Quasi-clocked adder circuit.

7. [M, None, 11.3] The dynamic implementation of the 4-bit carry-lookahead circuitry from Fig. 11-21 can significantly reduce the required transistor count.
 - a. Design a domino-logic implementation of Eq. 11.17. Compare the transistor counts of the two implementations.
 - b. What is the worst-case propagation delay path through this new circuit?
 - c. Are there any charge-sharing problems associated with your design? If so, modify your design to alleviate these effects.
8. [C, None, 11.3] Figure 0.4 shows a popular adder structure called the conditional-sum adder. Figure 0.4.a shows a four-bit instance of the adder, while 0.4.b gives the schematics of the basic adder cell. Notice that only pass-transistors are used in this implementation.
 - a. Derive Boolean descriptions for the four outputs of the one-bit conditional adder cell.
 - b. Based on the results of describe how the schematic of 0.4.a results in an addition.
 - c. Derive an expression for the propagation delay of the adder as a function of the number of bits N . You may assume that a switch has a constant resistance R_{on} when active and that each switch is identical in size.



(a) Four-bit conditional-sum adder



(b) Conditional adder cell
Figure 0.4 Conditional-sum adder.

9. [M, None, 11.3] Consider replacing all of the NMOS evaluate transistors in a dynamic Manchester carry chain with a single common pull-down as shown in Figure 0.5.a. Assume that each NMOS transistor has $(W/L)_N = 0.5/0.25$ and each PMOS has $(W/L)_P = 0.75/0.25$. Further assume that parasitic capacitances can be modeled by a 10 fF capacitor on each of the

internal nodes: $A, B, C, D, E,$ and F . Assume all transistors can be modeled as linear resistors with an on-resistance, $R_{on} = 5 \text{ k}\Omega$.

- a. Does this variation perform the same function as the original Manchester carry chain? Explain why or why not.
- b. Assuming that all inputs are allowed only a single zero-to-one transition during evaluation, will this design involve charge-sharing difficulties? Justify your answer.
- c. Complete the waveforms in Figure 0.5b for $P_0 = P_1 = P_2 = P_3 = 2.5 \text{ V}$ and $G_0 = G_1 = G_2 = G_3 = 0 \text{ V}$. Compute and indicate t_{pHL} values for nodes $A, E,$ and F . Compute and indicate when the 90% precharge levels are obtained.

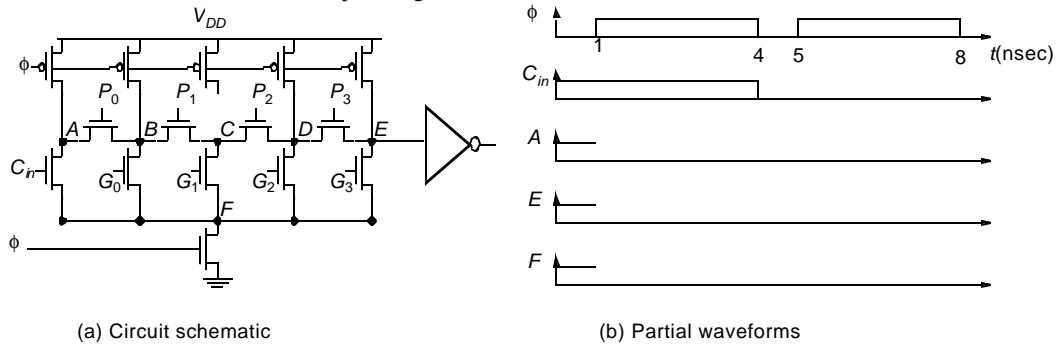


Figure 0.5 Alternative dynamic Manchester carry-chain adder.

10. [M, None, 11.3] Consider the two implementations of Manchester carry gates in Figure 11-8.
 - a. Compare the delay per segment of the two implementations
 - b. Compare the layout complexities of the two gates using stick diagrams.
 - c. In the precharged Manchester carry chain using the gate from b. find the probability that the carry signal is propagated from the 15th to the 16th bit of a 32-bit adder, assuming random inputs.
11. [C, None, 11.3] Consider the Radix-4 and Radix-2 Kogge-Stone adders from Figures 11-22 and 11-27 extended to 64-bits. All gates are implemented in domino and all gates in a stage have the same size. The adders have an overall fanout (electrical effort) of 6.
 - a. Using logical effort, identify the critical path.
 - b. Size the gates for minimum delay (hint: don't forget to factor in branching). Which adder is faster?
 - c. Let's now consider sparse versions of each of the above trees. In a tree with a sparseness of 2, only every other carry is computed and it is used to select 2 sums. Similarly, a tree with a sparseness of 4 computes every fourth carry - and that carry signal is used to select 4 sums. Repeat a. and b. for Radix-2 and Radix-4 trees with sparseness of 2 and 4 and compare their speed. Which adder is fastest?
 - d. Compare the switching power of all adders analyzed in this problem.
12. [C, None, 11.3] In this problem we will analyze a carry-lookahead adder proposed by H. Ling more than 20 years ago, but still among the fastest adders available. In a conventional adder, in order to add two numbers

$$A = a_{n-1}2^{n-1} + a_{n-2}2^{n-2} + \dots + a_02^0$$

$$B = b_{n-1}2^{n-1} + b_{n-2}2^{n-2} + \dots + b_02^0$$

we first compute the local carry generate and propagate terms:

$$g_i = a_i b_i$$

$$p_i = a_i + b_i$$

then, with a ripple or a tree circuit we form the global carry-out terms resulting from the recurrence relation:

$$G_i = g_i + p_i G_{i-1}$$

Finally, we form the sum of A and B using local expressions:

$$S_i = p_i \oplus G_{i-1}$$

In the conventional adder, the terms G_i have, as described, a physical significance. However, an arbitrary function could be propagated, as long as sum terms could be derived. Ling's approach is to replace G_i with:

$$H_i = G_i + G_{i-1}$$

i.e. H_i is true if "something happens at bit i " - there is a carry out or a carry in. H_i is so-called "Ling's pseudo-carry".

a. Show that:

$$H_i = g_i + t_{i-1} H_{i-1}$$

where $p_i = a_i + b_i$ (it was Ling's idea to change the notation).

b. Find a formula for computing the sum out of the operands and Ling's pseudo-carry.

c. Unroll the recursions for G_i and H_i for $i = 3$. You should get the expressions for G_3 and H_3 as a function of the bits of input operands. Simplify the expressions as much as possible.

d. Implement the two functions using n-type dynamic gates. Draw the two gates and size the transistors. Which one helps us build a faster adder? Explain your answer.

13. [M, None, 11.4] An array multiplier consists of rows of adders, each producing partial sums that are subsequently fed to the next adder row. In this problem, we consider the effects of pipelining such a multiplier by inserting registers between the adder rows.

a. Redraw Figure 11-31 by inserting word-level pipeline registers as required to achieve maximal benefit to throughput for the 4×4 multiplier. Hint: you must use additional registers to keep the input bits synchronized to the appropriate partial sums.

b. Repeat for a carry-save, as opposed to ripple-carry, architecture.

c. For each of the two multiplier architectures, compare the critical path, throughput, and latency of the pipelined and nonpipelined versions.

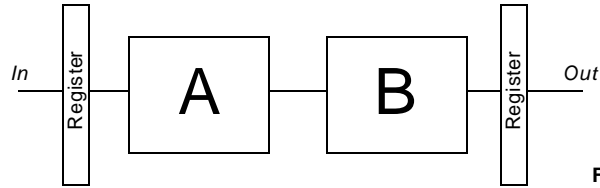
d. Which architecture is better suited to pipelining, and how does the choice of a vector-merging adder affect this decision?

14. [M, None, 11.4] Estimate the delay of a 16×16 Wallace tree multiplier with the final adder implemented using a Radix-4 tree. One FA has a delay of t_p , a HA $2/3 t_p$ and a CLA stage $1/2 t_p$.

15. [E, None, 11.5] The layout of shifters is dominated by the number of wires running through a cell. For both the barrel shifter and the logarithmic shifter, estimate the width of a shifter cell as a function of the maximum shift-width M and the metal pitch p .

16. [E, None, 11.7] Consider the circuit from Figure 0.7. Modules A and B have a delay of 10 ns and 32 ns at 2.5V, and switch 15 pF and 56 pF respectively. The register has a delay of 2 ns and switches 0.1 pF. Adding a pipeline register allows for reduction of the supply voltage while maintaining throughput. How much power can be saved this way? Delay with respect to V_{DD} can be approximated from Figure 11-57.

17. [E, None, 11.7] Repeat Problem 16, using parallelism instead of pipelining. Assume that a 2-to-1 multiplexer has a delay of 4 ns at 2.5 V and switches 0.3 pF. Try parallelism levels of 2 and by 4. Which one is preferred?

**Figure 0.6** Pipelined datapath.

DESIGN PROBLEM

Using the 0.25 μm CMOS technology, design a static 32-bit adder, with the following constraints:

1. input capacitance on each bit is limited to not more than 50fF.
2. each bit is loaded with 100fF.

Use a carry lookahead tree of your choice for implementation. The goal is to achieve the shortest propagation delay.

Determine the logic design of the adder and W and L of all transistors. Initially size the design using the method of logical effort. Estimate the capacitance of carry signal wires based on the floorplan. Verify and optimize the design using SPICE. Compute also the energy consumed per transition. If you have a layout editor available, perform the physical design, extract the real circuit parameters, and compare the simulated results with the ones obtained earlier. For implementation use the 144λ bit-slice pitch, that corresponds to 36 metal-1 tracks. Use metal 1 for cell-level power distribution and intra-cell routing, metal-2 for short interconnect and metal-3 and metal-4 for long carries.