CSEE W4823x **Homework 3** Handout 18
Prof. Steven Nowick October 12, 2016

This homework is due **at the beginning of class** on Tuesday, October 18. (I will also accept it with no penalty under my office door, until Wed. Oct. 19 at 5pm.) No group work is allowed.

Note: A correct answer without adequate explanation or derivation will have points deducted. To get full credit, (a) write legibly/type, and (b) show all work (label relevant items, show deriviations, include explanations).

1. (25 points) **Conditional Sum Adders.** In this problem, you will explore the basic operation of an 8-bit conditional sum adder. (These adders are covered in Handouts #15 and #16, and lecture #10 slides [part 2]).

   Assume you are given two unsigned 8-bit inputs, $X = 0101\ 1100$ (i.e. $x_7 \ldots x_0$) and $Y = 0010\ 0010$ (i.e. $y_7 \ldots y_0$), which are applied to the inputs of the conditional sum adder in Handout #16. Assume also that $c_{in} = 0$.

   Show the detailed results of this addition. In particular, (a) for each of the 15 FA's, list both its sum and carryout value, and (b) list the output of each of the 11 MUXes. The result will be a snapshot of the adder after the addition.

   Note: For ease of presentation, simply make a copy of p. 2 of Handout #16, and clearly mark each signal value in the figure.

2. (30 points) **Carry-Lookahead Adders.** In this problem, you are to analyze the operation of a 16-bit CLA adder. (These adders are covered in Handouts #13 and #15, and lecture #10 slides [part 1]).

   Refer to slide #3 of lecture #10 slides [part 1]), which shows a micro-architectural diagram of a 16-bit hierarchical CLA. Assume a 16-bit addition is complete, and *you are given the following subset of signal values:*

   - carry-in: $c_0 = 0$
   - bitwise generate: $g_{14} = 0, g_6 = 1, g_3 = 0, g_2 = 0$
   - bitwise propagate: $p_{13} = 1, p_{12} = 1, p_7 = 1, p_4 = 1, p_2 = 0, p_0 = 1$
   - group generate: $G_{0/15} = 1$
   - group propagate: $P_{8/11} = 1$

   Using this information, identify as many other signal values as you can, which can be derived from the above partial simulation result. You should *only list signal values that can be precisely determined, based on the above given values.* You will not be able to derive all signal values, just identify as many as possible, which are totally determined by the above given subset of values.

   In particular, list values for any (i) primary inputs (A_i, B_i), (ii) bitwise generate (g_i) and propagate (p_i) signals, (iii) group generate ($P_{i/j}$) and propagate ($P_{i/j}$) signals, (iv) carries (c_i), and (v) sum bits (s_i).

   To ensure partial credit, give short explanation and justification for your assignments.

3. (45 points) **Adder Performance Analysis.** In this problem, you will roughly compare the expected worst-case delay of 4 adders, both 16-bit and 64-bit. You will then summarize your analytical results in a table.

   What To Do. Consider the following 4 adders: (i) *ripple-carry,* (ii) *carry-select,* (iii) *carry-lookahead,* and (iv) *conditional sum.* For simplicity, *assume any basic component has the same unit delay* (i.e. delay is 1), including: full adder, half adder, 4-bit CLA block, and 2-to-1 MUX. (While this is a crude simplification, it is not entirely unreasonable, since at the gate-level design, these blocks are not dissimilar in their performance.)

For example, this means that any delay through an FA block (e.g. inputs to sum [if carry-in is correct], inputs to carry-out, carry-in to carry-out) has a delay of "1 unit".

Assumptions: all inputs, a_i and b_i, as well as carry-in (c_0) arrive at time t=0. The "worst-case delay" of the adder will be determined by the longest path needed to compute all final sum bit values.

(a) (20 points) *16-Bit Addition Comparison.* Derive the worst-case delay of each of the above 4 adders, assuming 16-bit addition. *For partial credit, give short but clear explanations justifying your analysis.* Place the result in a small table.

Note that while your reading gives some complete 16-bit adder designs (e.g. CLA, Lecture slides #10 [part 2, p. 3]), it gives 8-bit designs for other examples (carry-select, conditional sum). Hence you will need to generalize the designs to handle 16 bits. (You do *not* need to draw the 16-bit adders, just provide an accurate analysis.)

(b) (20 points) *64-Bit Addition Comparison.* Derive the worst-case delay of each of the above 4 adders, now assuming 64-bit addition. *For partial credit, give short but clear explanations justifying your analysis.* Follow the same guidelines as in part (a), above. Place the result in a small table.

(c) (5 points) *Discussion.* Clearly discuss the important trends within each of the above two tables, as well as between the two tables (i.e. moving from 16- to 64-bit addition). Provide any useful insights and observations. [3-6 sentences]