EE466

VLSI System Design

Midterm Exam

Oct. 19, 2023. (4:20pm – 5:35pm)

Instructor: Dae Hyun Kim (daehyun@eecs.wsu.edu)

Name:

WSU ID:

Problem	Points	
1	10	
2	20	
3	20	
4	10	
5	20	
6	20	
7	20	
8	40	
Total	160	

* Allowed: Textbooks, cheat sheets, class notes, notebooks, calculators, watches

* Not allowed: Electronic devices (smart phones, tablet PCs, laptops, etc.) except calculators and watches

Problem #1 (Kogge-Stone Adder, 10 points)

For the 128-bit Kogge-Stone adder, show one of the critical paths to calculate S_{111}. What is the delay of the critical path? Use the following delay values for logic gates.

- 2-input AND, OR: d
- XOR: 2d

Problem #2 (Kogge-Stone Adder, 20 points)

Count the # following gates required to implement the 32-bit Kogge-Stone adder (including the generation of C_{32}).

- 2-input AND gates:
- 2-input OR gates:
- 2-input XOR gates:

Problem #3 (Kogge-Stone Adder, 20 points)

Count the # nets required to implement the 32-bit Kogge-Stone adder (including the generation of C_{32}). Include the primary input/output nets too (e.g., if S=A+B+Cin, $A_0, \dots, A_{31}, B_0, \dots, B_{31}, C_{in}$ are all input nets.)

Problem #4 (Carry-Lookahead Adder, 10 points)

For the 256-bit carry-lookahead adder, show one of the critical paths to calculate S_{240}. What is the delay of the critical path? Use the following delay values for logic gates.

- 2-, 3-, 4-input AND, OR: d
- XOR: 2d

Problem #5 (Carry-Lookahead Adder, 20 points)

Count the # following gates required to implement the 32-bit Carry-Lookahead adder (including the generation of C_{32}). For the first-level units (computing the sum bits + generating $g_i, p_i, g_{i+3:i}, p_{i+3:i}$), assume that c_i in a unit is computed by $c_i = g_{i-1:k} + p_{i-1:k} \cdot c_k$ where c_k is the carry signal fed into the unit. (For example, $c_7 = g_{6:4} + p_{6:4} \cdot c_4$ and $g_{6:4}$ is computed by $g_6 + p_6 g_5 + p_6 p_5 g_4$ and $p_{6:4}$ is computed by $p_6 p_5 p_4$.)

- 2,3,4-input AND gates (i.e., # 2-input ANDs + # 3-input ANDs + # 4-input ANDs):
- 2,3,4-input OR gates:
- 2-input XOR gates:

Problem #6 (Carry-Lookahead Adder, 20 points)

Count the # nets in the 32-bit carry-Lookahead adder (including the generation of C_{32}). For the first-level units (computing the sum bits + generating $g_i, p_i, g_{i+3:i}, p_{i+3:i}$), assume that c_i in a unit is computed by $c_i = g_{i-1:k} + p_{i-1:k} \cdot c_k$ where c_k is the carry signal fed into the unit. (For example, $c_7 = g_{6:4} + p_{6:4} \cdot c_4$ and $g_{6:4}$ is computed by $g_6 + p_6 g_5 + p_6 p_5 g_4$ and $p_{6:4}$ is computed by $p_6 p_5 p_4$.) Include the primary input/output nets too (e.g., if S=A+B+Cin, $A_0, \ldots, A_{31}, B_0, \ldots, B_{31}, C_{in}$ are all input nets.)

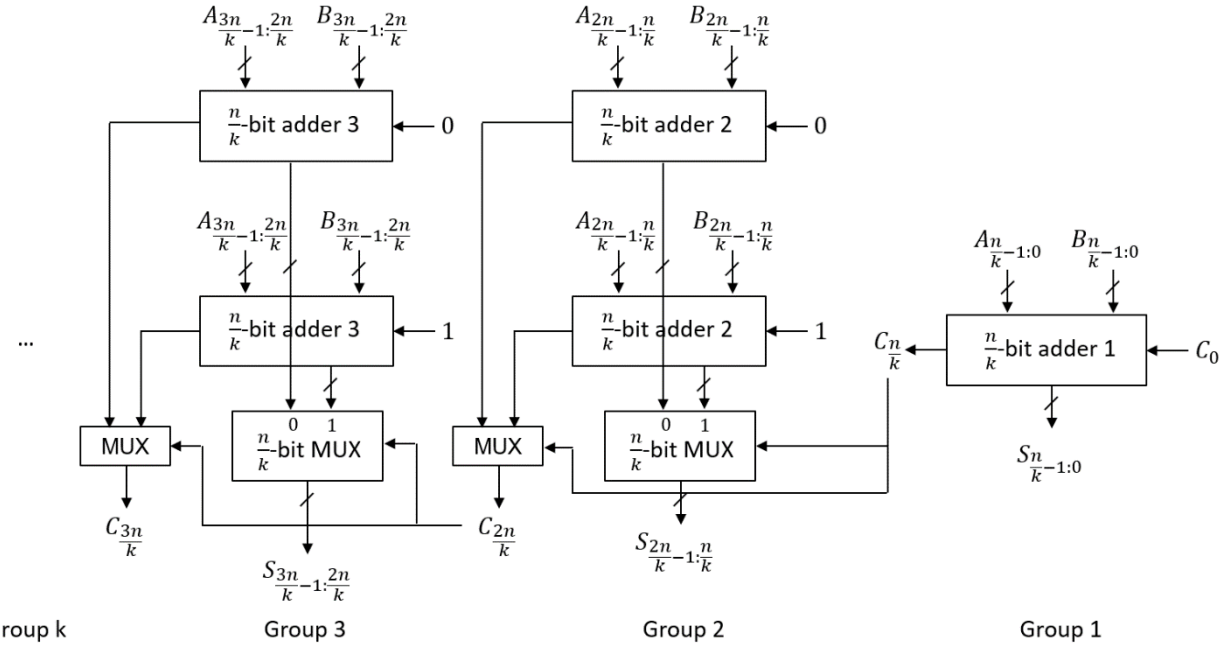
Problem #7 (Carry-Lookahead Adder, 20 points)

We want to design a 256-bit carry-lookahead adder with max. fan-in of 3 (i.e., we use 1-, 2-, and 3-input gates, but not 4-input gates). Thus, we should group three bits in level-1 modules and then three modules in level-k (k=2, 3, …) modules. Show one of the critical paths to calculate S_{245}. What is the delay of the critical path? Use the following delay values for logic gates.

- 2-, 3-input AND, OR: d
- XOR: 2d

Problem #8 (Hybrid Adder, 45 points)

An n-bit carry select adder (n is a given constant) can be designed using $\frac{n}{k}$-bit adders in multiple stages as follows (k will be determined):



Notice that there are total k groups and each group processes $\frac{n}{k}$ bits (so total n bits).

(1) Suppose we use an $\frac{n}{k}$-bit conditional sum adder for each $\frac{n}{k}$-bit adder. Express the delay of the n-bit carry-select adder as a function of the following parameters (10 points):

- m: Delay of a MUX
- d: Delay of a 1-bit full-adder (used in the first step of the conditional sum adder)

(2) Then, differentiate the above delay value with respect to k (notice that n, m, d are all constants and k is the only variable) and set it to zero. This value will give you the optimal value of k minimizing the total delay. (10 points)

(Hint: $\frac{d}{dk}(\log_2 f(k)) = \frac{f'(k)}{f(k)\cdot \ln 2}$ and $\log_2 \frac{a}{b} = \log_2 a - \log_2 b$)

(3) Answer the following questions (Correct: +5, Wrong: -5, No answer: 0)

- If n increases, the optimal value of k increases too. (True / False)
- If m increases, the optimal value of k increases too. (True / False)
- If d increases, the optimal value of k increases too. (True / False)

(4) Suppose we split the n bits into s groups (s is a constant), and the groups are Group 1 (the rightmost one in the figure), Group 2, …, Group s (the leftmost one). Let the # bits processed in Group p be n_p. (Thus, it will satisfy $n_1 + n_2 + \cdots + n_s = n$, the total # bits). In the figure above, $n_1 = n_2 = \cdots = n_s = \frac{n}{s}$. In this problem, however, they could be different. Answer the following questions (Correct: +5, Wrong: -5, No answer: 0).

- Suppose we use the ripple-carry adder design for the adders in each group. In this case, if we optimally design the carry-select adder, $n_a \geq n_b$ should be satisfied when $a \geq b$ (In other words, for example, the # bits processed in Group 10 should be greater than or equal to the # bits processed in Group 7). (True / False)