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

$$S_{111} = p_{111} \oplus c_{111}$$

$$c_{111} = g_{110:0} + p_{110:0} \cdot c_0$$

$$g_{110:0} = g_{110:47} + p_{110:47} \cdot g_{46:0}$$

$$g_{110:47} = g_{110:79} + p_{110:79} \cdot g_{78:47}$$

$$g_{110:79} = g_{110:95} + p_{110:95} \cdot g_{94:79}$$

$$g_{110:95} = g_{110:103} + p_{110:103} \cdot g_{102:95}$$

$$g_{110:103} = g_{110:107} + p_{110:107} \cdot g_{106:103}$$

$$g_{110:107} = g_{110:109} + p_{110:109} \cdot g_{108:107}$$

$$g_{110:109} = g_{110} + p_{110} \cdot g_{109}$$

Delay: 2d (for p_{110}) + 2d (for $g_{110:109}$) + 2d*6 (up to $g_{110:0}$) + d (c_{111}) + 2d

Answer: 19d

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:

$g_i = A_i \cdot B_i$: one AND → 32 ANDs

$p_i = A_i \oplus B_i$: one XOR → 32 XORs

$C_i = g_{i-1:0} + p_{i-1:0} \cdot C_0$: one OR and one AND → 32 ORs and 32 ANDs

$s_i = p_i \oplus C_i$: one XOR gate → 32 XOR gates

$g_{i+1:i}$ and $p_{i+1:i}$: 2 ANDs and 1 OR (for i=0 to 30)

$g_{i+3:i}$ and $p_{i+3:i}$: 2 ANDs and 1 OR (for i=0 to 28)

$g_{i+7:i}$ and $p_{i+7:i}$: 2 ANDs and 1 OR (for i=0 to 24)

$g_{i+15:i}$ and $p_{i+15:i}$: 2 ANDs and 1 OR (for i=0 to 16)

$g_{i+31:i}$ and $p_{i+31:i}$: 2 ANDs and 1 OR (for i=0)

2 ANDs and 1 OR: 2:0, 4:0, 5:0, 6:0, …, 31:0 except those merging 2^k bits = 26, so 26*2 ANDs and 26*1 ORs

AND: 32 + 32 + 2*(31 + 29 + 25 + 17 + 1) + 2*26 = 322

OR: 32 + 1*(31 + 29 + 25 + 17 + 1) + 1*26 = 161

XOR: 64

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, $A_0, \ldots, A_{31}, B_0, \ldots, B_{31}, C_{in}$ are all input nets.)

Primary inputs: 32 (A) + 32 (B) + 1 (Cin) = 65

$g_i = A_i \cdot B_i$: 32

$p_i = A_i \oplus B_i$: 32

$C_i = g_{i-1:0} + p_{i-1:0} \cdot C_0$: 2*32 (the output of the AND gate is a net)

$s_i = p_i \oplus C_i$: 32

$g_{i+1:i}$ and $p_{i+1:i}$: 2 (from $g_{i+1:i}$) +1 (from $p_{i+1:i}$) = 3 (for i=0 to 30)

$g_{i+3:i}$ and $p_{i+3:i}$: 2+1 = 3 (for i=0 to 28)

$g_{i+7:i}$ and $p_{i+7:i}$: 2+1 = 3 (for i=0 to 24)

$g_{i+15:i}$ and $p_{i+15:i}$: 2+1 = 3 (for i=0 to 16)

$g_{i+31:i}$ and $p_{i+31:i}$: 2+1 = 3 (for i=0)

2:0, 4:0, 5:0, 6:0, …, 31:0 except those merging 2^k bits = (30 – 4) * 3 nets = 78

Thus, 65 + 32 + 32 + 2*32 + 32 + 3*(31+29+25+17+1) + 78 = 612.

Answer: 612 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

$$S_{240} = p_{240} \oplus C_{240}$$

$$C_{240} = g_{239:224} + p_{239:224} \cdot g_{223:208} + p_{239:224} \cdot p_{223:208} \cdot g_{207:192} + p_{239:224} \cdot p_{223:208} \cdot p_{207:192} \cdot C_{192}$$

$$C_{192} = g_{191:128} + p_{191:128} \cdot g_{127:64} + p_{191:128} \cdot p_{127:64} \cdot g_{63:0} + p_{191:128} \cdot p_{127:64} \cdot p_{63:0} \cdot C_0$$

$$g_{63:0} = g_{63:48} + p_{63:48} \cdot g_{47:32} + p_{63:48} \cdot p_{47:32} \cdot g_{31:16} + p_{63:48} \cdot p_{47:32} \cdot p_{31:16} \cdot g_{15:0}$$

$$g_{15:0} = g_{15:12} + p_{15:12} \cdot g_{11:8} + p_{15:12} \cdot p_{11:8} \cdot g_{7:4} + p_{15:12} \cdot p_{11:8} \cdot p_{7:4} \cdot g_{3:0}$$

$$g_{3:0} = g_3 + p_3 \cdot g_2 + p_3 \cdot p_2 \cdot g_1 + p_3 \cdot p_2 \cdot p_1 \cdot g_0$$

$$p_3 = A_3 \oplus B_3$$

Delay: 2d (for p_3) + 2d (for $g_{3:0}$) + 2d (for $g_{15:0}$) + 2d (for $g_{63:0}$) + 2d (for C_{192}) + 2d (for C_{240}) + 2d (for S_{240})

Answer: 14d

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:

$g_i = A_i \cdot B_i$: one AND → 32 ANDs

$p_i = A_i \oplus B_i$: one XOR → 32 XORs

$s_i = p_i \oplus C_i$: one XOR → 32 XORs

In each level-1 carry-lookahead unit (there are eight L-1 units, i=0,4,8,12,16,20,24,28)

- $g_{i+1:i} = g_{i+1} + p_{i+1} \cdot g_i$
- $p_{i+1:i} = p_{i+1} \cdot p_i$
- $g_{i+2:i} = g_{i+2} + p_{i+2} \cdot g_{i+1} + p_{i+2} \cdot p_{i+1} \cdot g_i$
- $p_{i+2:i} = p_{i+2} \cdot p_{i+1} \cdot p_i$
- $g_{i+3:i} = g_{i+3} + p_{i+3} \cdot g_{i+2} + p_{i+3} \cdot p_{i+2} \cdot g_{i+1} + p_{i+3} \cdot p_{i+2} \cdot p_{i+1} \cdot g_i$
- $p_{i+3:i} = p_{i+3} \cdot p_{i+2} \cdot p_{i+1} \cdot p_i$
- $C_{i+1} = g_i + p_i \cdot C_i$
- $C_{i+2} = g_{i+1:i} + p_{i+1:i} \cdot C_i$
- $C_{i+3} = g_{i+2:i} + p_{i+2:i} \cdot C_i$

In each level-2 carry-lookahead unit (there are two L-2 units)

- $C_4 = g_{3:0} + p_{3:0} \cdot C_0$
- $C_8 = g_{7:4} + p_{7:4} \cdot g_{3:0} + p_{7:4} \cdot p_{3:0} \cdot C_0$
- $C_{12} = g_{11:8} + p_{11:8} \cdot g_{7:4} + p_{11:8} \cdot p_{7:4} \cdot g_{3:0} + p_{11:8} \cdot p_{7:4} \cdot p_{3:0} \cdot C_0$
- $g_{15:0} = g_{15:12} + p_{15:12} \cdot g_{11:8} + p_{15:12} \cdot p_{11:8} \cdot g_{7:4} + p_{15:12} \cdot p_{11:8} \cdot p_{7:4} \cdot g_{3:0}$
- $p_{15:0} = p_{15:12} \cdot p_{11:8} \cdot p_{7:4} \cdot p_{3:0}$

In a level-3 unit

- $C_{16} = g_{15:0} + p_{15:0} \cdot C_0$
- $C_{32} = g_{31:16} + p_{31:16} \cdot g_{15:0} + p_{31:16} \cdot p_{15:0} \cdot C_0$

L1: 8*(12 ANDs + 6 ORs) = 96 ANDs + 48 ORs

L2: 2*(10 ANDs + 4 ORs) = 20 ANDs + 8 ORs

L3: 3 ANDs + 2 ORs

Answer:

AND: 151

OR: 58

XOR: 64

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.)

Primary inputs: 32 (A) + 32 (B) + 1 (Cin) = 65

In each level-1 unit (there are eight L-1 units), assuming its carry-in is C_i

- $g_i = A_i \cdot B_i$: 1 net (total 4 nets in a L-1 unit)
- $p_i = A_i \oplus B_i$: 1 net (total 4 nets in a L-1 unit)
- $g_{i+1:i} = g_{i+1} + p_{i+1} \cdot g_i$
- $p_{i+1:i} = p_{i+1} \cdot p_i$
- $g_{i+2:i} = g_{i+2} + p_{i+2} \cdot g_{i+1} + p_{i+2} \cdot p_{i+1} \cdot g_i$
- $p_{i+2:i} = p_{i+2} \cdot p_{i+1} \cdot p_i$
- $g_{i+3:i} = g_{i+3} + p_{i+3} \cdot g_{i+2} + p_{i+3} \cdot p_{i+2} \cdot g_{i+1} + p_{i+3} \cdot p_{i+2} \cdot p_{i+1} \cdot g_i$
- $p_{i+3:i} = p_{i+3} \cdot p_{i+2} \cdot p_{i+1} \cdot p_i$
- $C_{i+1} = g_i + p_i \cdot C_i$
- $C_{i+2} = g_{i+1:i} + p_{i+1:i} \cdot C_i$
- $C_{i+3} = g_{i+2:i} + p_{i+2:i} \cdot C_i$
- $s_i = p_i \oplus C_i$

Thus, 4*(1 + 1) + 2 + 1 + 3 + 1 + 4 + 1 + 2 + 2 + 2 + 4 = 30 nets in a L-1 unit. Total eight L-1 units, so 8*30 = 240 nets.

Then, we pass $g_{i+3:i}$ and $p_{i+3:i}$ to the two L-2 units.

In the first level-2 unit (there are two L-2 units),

- $C_4 = g_{3:0} + p_{3:0} \cdot C_0$
- $C_8 = g_{7:4} + p_{7:4} \cdot g_{3:0} + p_{7:4} \cdot p_{3:0} \cdot C_0$
- $C_{12} = g_{11:8} + p_{11:8} \cdot g_{7:4} + p_{11:8} \cdot p_{7:4} \cdot g_{3:0} + p_{11:8} \cdot p_{7:4} \cdot p_{3:0} \cdot C_0$
- $g_{15:0} = g_{15:12} + p_{15:12} \cdot g_{11:8} + p_{15:12} \cdot p_{11:8} \cdot g_{7:4} + p_{15:12} \cdot p_{11:8} \cdot p_{7:4} \cdot g_{3:0}$
- $p_{15:0} = p_{15:12} \cdot p_{11:8} \cdot p_{7:4} \cdot p_{3:0}$

Thus, 2 + 3 + 4 + 4 + 1 = 14 nets in a L-2 unit. Total 2*14 = 28 nets.

In the level-3 unit (there is only one L-3 unit),

- $C_{16} = g_{15:0} + p_{15:0} \cdot C_0$
- $C_{32} = g_{31:16} + p_{31:16} \cdot g_{15:0} + p_{31:16} \cdot p_{15:0} \cdot C_0$

Thus, 2 + 3 = 5 nets in a L-3 unit.

Answer: 240 + 28 + 5 = 273 nets.

(Notice that the 32-bit KSA requires 612 nets. The 32-bit CLA requires less than a half of that.)

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

$$S_{245} = p_{245} \oplus C_{245}$$

$$C_{245} = g_{244} + p_{244} \cdot g_{243} + p_{244} \cdot p_{243} \cdot C_{243}$$

$$C_{243} = g_{242:0} + p_{242:0} \cdot C_0$$

$$g_{242:0} = g_{242:162} + p_{242:162} \cdot g_{161:81} + p_{242:162} \cdot p_{161:81} \cdot g_{80:0}$$

$$g_{80:0} = g_{80:54} + p_{80:54} \cdot g_{53:27} + p_{80:54} \cdot p_{53:27} \cdot g_{26:0}$$

$$g_{26:0} = g_{26:18} + p_{26:18} \cdot g_{17:9} + p_{26:18} \cdot p_{17:9} \cdot g_{8:0}$$

$$g_{8:0} = g_{8:6} + p_{8:6} \cdot g_{5:3} + p_{8:6} \cdot p_{5:3} \cdot g_{2:0}$$

$$g_{2:0} = g_2 + p_2 \cdot g_1 + p_2 \cdot p_1 \cdot g_0$$

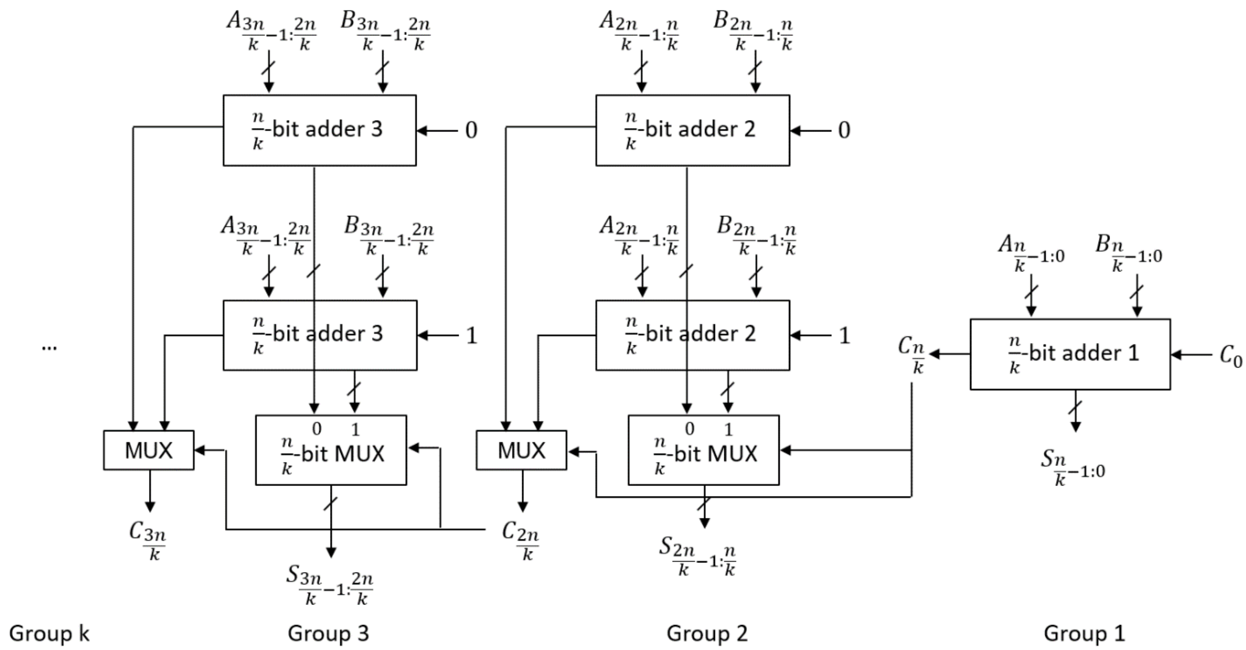
$$p_2 = A_2 \oplus B_2$$

Delay: 2d (for p_2) + 2d (for $g_{2:0}$) + 2d (for $g_{8:0}$) + 2d (for $g_{26:0}$) + 2d (for $g_{80:0}$) + 2d (for $g_{242:0}$) + 2d (for C_{243}) + 2d (for C_{245}) + 2d (for S_{245})

Answer: 18d

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)

Delay of an $\frac{n}{k}$-bit conditional sum adder: $d + \left(\log_2 \frac{n}{k} - 1\right) \cdot m$

It goes through k-1 carry propagation stages. Thus the delay is

$$\tau = d + \left(\log_2 \frac{n}{k} - 1\right) \cdot m + (k-1) \cdot m$$

(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)

$$\frac{d\tau}{dk} = m - m \cdot \frac{1}{k \ln 2} = 0$$

Thus, $k = \frac{1}{\ln 2}$

(Since k = 1.44…, so this just means 1 stage is the best for the design.)

(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)