EE466

VLSI System Design

Midterm Exam

Oct. 7, 2020. (4:20pm – 5:35pm)

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

Name:

WSU ID:

Problem	Points	
1	10	
2	10	
3	10	
4	10	
5	20	
6	10	
7	10	
Total	80	

* 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 1024-bit Kogge-Stone adder, show one of the critical paths to calculate S_{139}. What is the delay of the critical path? Use the following delay values for logic gates.

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

$$S_{139} = p_{139} \oplus C_{139}$$

$$C_{139} = g_{138:0} + p_{138:0} \cdot C_0$$

$$g_{138:0} = g_{138:11} + p_{138:11} \cdot g_{10:0}$$

$$g_{138:11} = g_{138:75} + p_{138:75} \cdot g_{74:11}$$

$$g_{138:75} = g_{138:107} + p_{138:107} \cdot g_{106:75}$$

$$g_{138:107} = g_{138:123} + p_{138:123} \cdot g_{122:107}$$

$$g_{138:123} = g_{138:131} + p_{138:131} \cdot g_{130:123}$$

$$g_{138:131} = g_{138:135} + p_{138:135} \cdot g_{134:131}$$

$$g_{138:135} = g_{138:137} + p_{138:137} \cdot g_{136:135}$$

$$g_{138:137} = g_{138} + p_{138} \cdot g_{137}$$

$$p_{138} = A_{138} \oplus B_{138}$$

Delay: 3d + (2d * 8) + 2d + 3d = 24d

(actually, it is 23d because computing C139 takes one cycle)

Problem #2 (Carry-Lookahead Adder, 10 points).

For the 1024-bit Carry-lookahead adder, show one of the critical paths to calculate S_{139}. What is the delay of the critical path? Use the following delay values for logic gates.

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

$$S_{139} = p_{139} \oplus C_{139}$$

$$C_{139} = g_{138} + p_{138} \cdot g_{137} + p_{138} \cdot p_{137} \cdot g_{136} + p_{138} \cdot p_{137} \cdot p_{136} \cdot C_{136}$$

$$C_{136} = g_{135:132} + p_{135:132} \cdot g_{131:128} + p_{135:132} \cdot p_{131:128} \cdot C_{128}$$

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

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

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

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

$$p_3 = A_3 \oplus B_3$$

Delay: 3d + (2d * 6) + 3d = 18d

Problem #3 (Conditional Sum Adder, 10 points)

Complete the following table.

	i:	7	6	5	4	3	2	1	0
	A_i:	1	1	1	1	0	1	0	0
	B_i:	1	0	1	1	1	1	0	1
Step 1	S_i^0:	0	1	0	0	1	0	0	1
	CO_i^0:	1	0	1	1	0	1	0	0
	S_i^1:	1	0	1	1	0	1	1	0
	CO_i^1:	1	1	1	1	1	1	0	1
Step 2	S_i^0:	0	1	1	0	0	0	0	1
	CO_i^0:	1		1		1		0	
	S_i^1:	1	0	1	1	0	1	1	0
	CO_i^1:	1		1		1		0	
Step 3	S_i^0:	1	0	1	0	0	0	0	1
	CO_i^0:	1				1			
	S_i^1:	1	0	1	1	0	0	1	0
	CO_i^1:	1				1			
Step 4	S_i^0:	1	0	1	1	0	0	0	1
	CO_i^0:	1							
	S_i^1:	1	0	1	1	0	0	1	0
	CO_i^1:	1							

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

For the 256-bit carry-lookahead adder, the input is $A_{255:0}$ and $B_{255:0}$ and the output is $S_{255:0}$ and C_{256}. We can calculate the delay of each sum bit S_k using the techniques we studied. How many among the 256 sum bits have the longest delay? You can estimate the number (i.e., the number doesn't need to be very accurate. Just a good estimation will be enough to get 10 points). Use the following assumptions for the delay calculation.

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

First of all, we need 64 1st-level carry-lookahead units (CLUs), 16 2nd-level CLUs, four 3rd-level CLUs, and one 4th-level CLUs.

The carry signals generated in the 1st-level CLUs: $C_4, C_8, C_{12}, C_{20}, \ldots$

The carry signals generated in the 2nd-level CLUs: $C_{16}, C_{32}, C_{48}, C_{80}, \ldots$

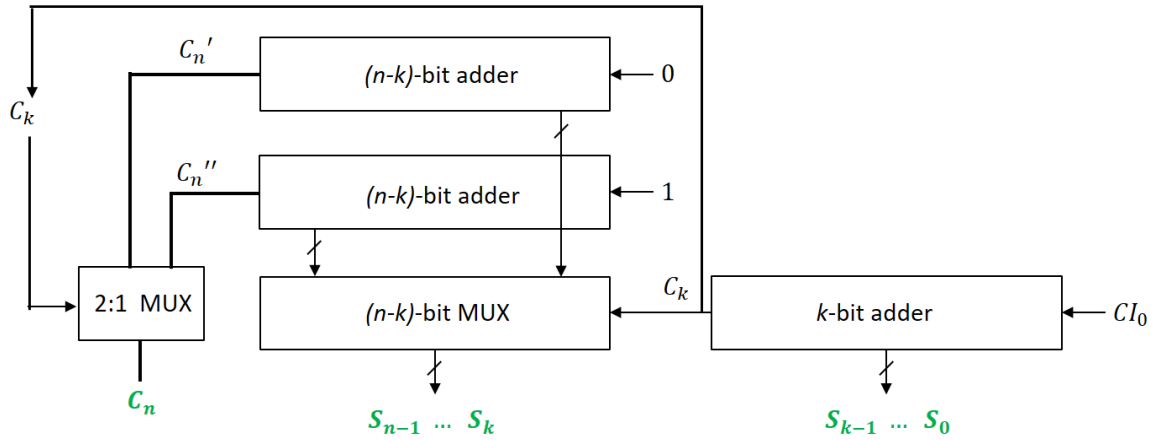
The carry signals generated in the 3rd-level CLUs: $C_{64}, C_{128}, C_{192}, C_{320}, \ldots$

The carry signals generated in the 4th-level CLUs: $C_{256}, C_{512}, \ldots$

Thus, the longest delay comes from the forward paths $\rightarrow C_{64}, C_{128}, C_{192} \rightarrow C_{80}, C_{96}, C_{112}, C_{144}, C_{160}, C_{176}, C_{208}, C_{224}, C_{240}$

Each of the above nine carry signals will be used in the 1st-level CLUs to generate the sum bits. Thus, 9*4 = 36 sum bits have the longest delay.

Problem #5 (Hybrid Adder, 20 points)



The figure above shows an n-bit carry-select adder. We split it into two groups, the first $(n-k)$ bits and the second k bits as shown. The k-bit adder is a ripple-carry adder and the two $(n-k)$-bit adders are carry-lookahead adders. We want to find the optimal value of k to minimize the total delay. Answer the following questions using the following assumptions.

- Delay of a 2-, 3-, and 4-input AND (or OR) gate: d
- Delay of a full adder (FA): $2d$
- Delay of an XOR gate: $2d$
- Delay of a 1-bit MUX: $2d$

(1) Since the delay of an FA is $2d$, the delay of the carry-out signal C_k is _$2kd$_____.

(2) The delay of a g_i signal is _d_____.

(3) The delay of a p_i signal is _$2d$_____.

(4) The delay of a $g_{i:i-3}$ signal is _$4d$_____.

(5) The delay of a $g_{i:i-15}$ signal is _$6d$_____.

(6) The delay of a $g_{i:i-(4^m-1)}$ signal is _$(2m+2)d$_____.

(7) The delay of a $C_{4^{m-1}}$ (or $C_{2\cdot4^{m-1}}$ or $C_{3\cdot4^{m-1}}$) signal is _$(2m+2)d$_____.

(8) The delay of a $C_{4^{m-2}}$ (or $C_{2\cdot4^{m-2}}$ or $C_{3\cdot4^{m-2}}$) signal is _$(2m+4)d$_____.

(9) The delay of a C_{4t} (t is a generally large integer) signal is _$(2m+2(m-1))d = (4m-2)d$__.

(10) The delay of a C_{4t+1} (or C_{4t+2} or C_{4t+3}) signal is _$(4m)d$____.

(11) The delay of a S_{4t+1} (or S_{4t+2} or S_{4t+3}) signal is _$(4m+2)d$___.

(12) If $(n-k)$ can be represented approximately by 4^m, m is _$\log_4(n-k)$_ (represent it using n and k).

(13) Thus, the delay of an $(n-k)$-bit CLA is approximately $\{4\log_4(n-k)+2\}d$.

Hint. note: The longest delay of the whole carry-select adder is minimized when the longest delay of the $(n-k)$-bit adder is equal to the delay of C_k. This requires solving the following equation (expressing k with respect to n).

$$2\log_4(n-k) = k-1$$

We can't solve it analytically, but we can numerically.

For $n = 64$, optimal k is approximately 6.

For $n = 256$, optimal k is approximately 8.

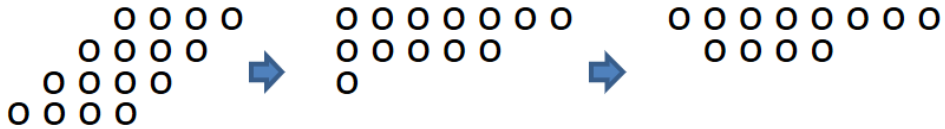
For $n = 1{,}024$, optimal k is approximately 10.

Problem #6 (Wallace Tree, 10 points)

(1) If you multiply two 3-bit unsigned binary numbers, how many carry-save-adder stages do you need to reduce the number of partial product rows down to two?

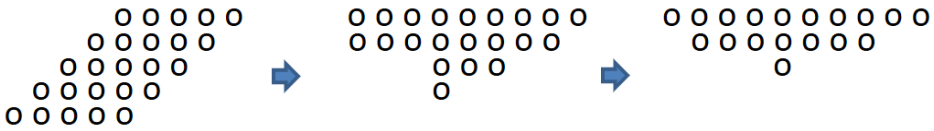
Answer: 1

(2) Repeat it for two 4-bit unsigned binary numbers.



Answer: 2

(3) Repeat it for two 5-bit unsigned binary numbers.



Answer: 3 (I didn't show the third stage.)

Problem #7 (Modified Booth Encoding, 10 points)

Use the modified Booth encoding technique to calculate the following multiplication (see page 9 in the multiplier lecture notes). Assume that all the numbers are unsigned.

A	0 1 1 0 1 1 0 1
$* X$	0 0 1 1 1 0 0 1
A	0 1 1 0 1 1 0 1
$* Y$	0 1 0 0 $\bar{1}$ 0 0 1
	+A
	-2A
	+0
	+A
0 0 0 0 0 0 0 0 0 1 1 0 1 1 0 1	
1 1 1 1 1 1 0 0 1 0 0 1 1	
0 0 0 0 0 0 0 0 0 0 0 0 0	
0 0 0 1 1 0 1 1 0 1	
0 0 0 1 1 0 0 0 0 1 0 0 0 1 0 1	