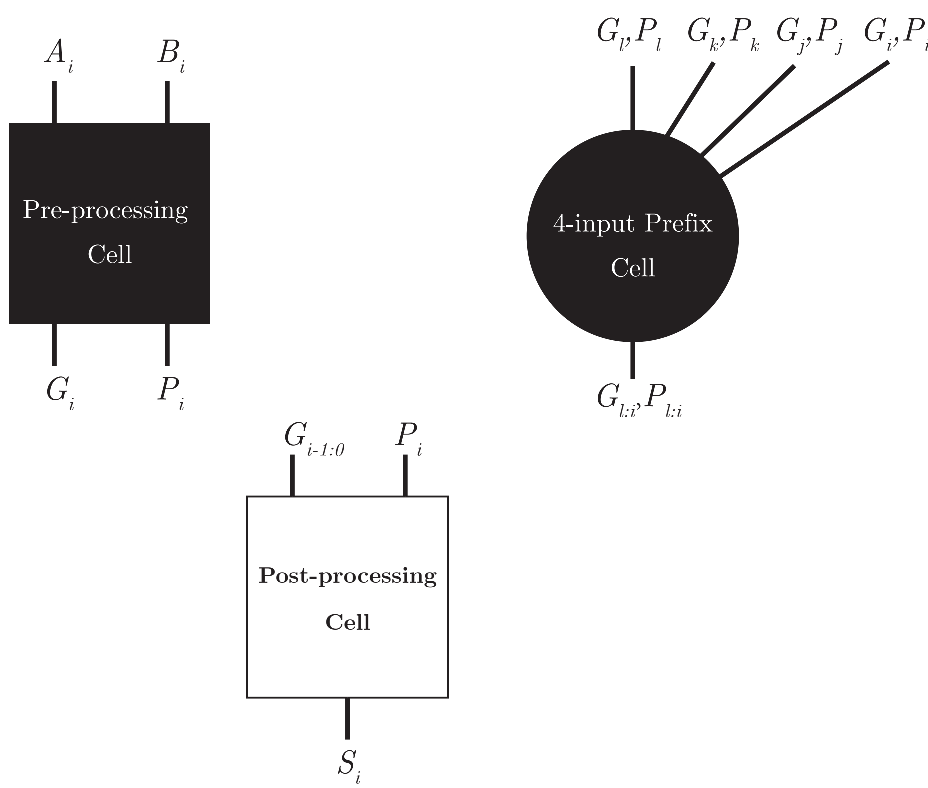
EECS 151/251A Homework 8

Due Friday, November 18^{th}, 2022 11:59PM

Problem 1: Carry Lookaheadache

Recall from the lecture that Kogge–Stone adder is a parallel prefix form carry look-ahead adder (CLA). In this problem, we'd like to design and analyze a 16-bit radix-4 Kogge-Stone adder.

(a) Write down the Boolean functions of the gate implementation of following building blocks for the radix-4 Kogge-Stone adder. You may use AND, OR and XOR only. Also, how should you handle cases where certain 4-input prefix cells have fewer than 4 pairs of input?



(b) Which of the following are true for radix-4 Kogge-Stone adder?

____ Compared to Radix-2 adders, Radix-4 adders reduce the depth of the tree by a factor of 4 (excluding the input and output stages).

____ An N-bit Radix-4 adders has $\mathcal{O}(\sim log_4(N))$ in time.

____ It is possible to implement a 32-bit Kogge-Stone adder using only the building blocks in part (a).

(c) Suppose given the following propagation delays:

$$t_{AND} = 5ps, \quad t_{OR} = 4ps, \quad t_{XOR} = 7ps$$

Derive the critical path of the 16-bit Kogge-Stone adder based on your implementation in part (a), ignoring the delays in routing. (Hint: If you are not sure about the topology, take a look at the slides in Lecture 19.)

(d) Based on your result above, what's the maximum clock frequency for this 16-bit adder?

(e) **(251A Only)** In reality, those prefix cells will not be built using the basic 2-input AND, OR, XOR gates. Instead, they will be built as a big CMOS gate (and inverters). Draw the schematic of the 4-input prefix cell for $\bar{G}_{l:i}$ and $\bar{P}_{l:i}$ respectively with the minimum number of transistors.

Problem 2: Booth's Bizarre Invention

Refer to the following table of the behavior of Booth recoding.

B_{K+1}	B_K	B_{K-1}	Action
0	0	0	add 0
0	0	1	add A
0	1	0	add A
0	1	1	add 2A
1	0	0	sub 2A
1	0	1	sub A
1	1	0	sub A
1	1	1	add 0

Write down the sequence of operation and the final result given the following unsigned two input numbers:

$$\begin{array}{r} 01100111\ (A) \\ \times\ 10110010\ (B) \end{array}$$

Answer should be in the format of:

Suppose A=1010	B=1001
(B[1:-1]=010):	add A,
(B[3: 1]=100):	sub 2A,
	...
result	= A - (2A<<2) + ...
	= 0000(1010) - 00(1010)00 + ...
	= ...