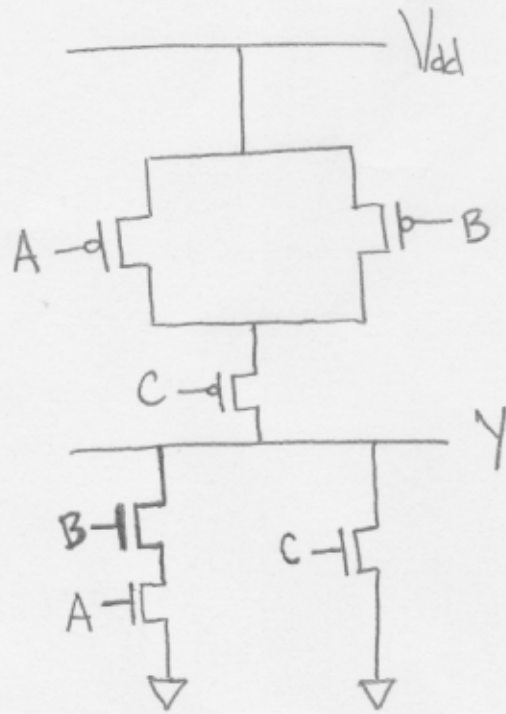
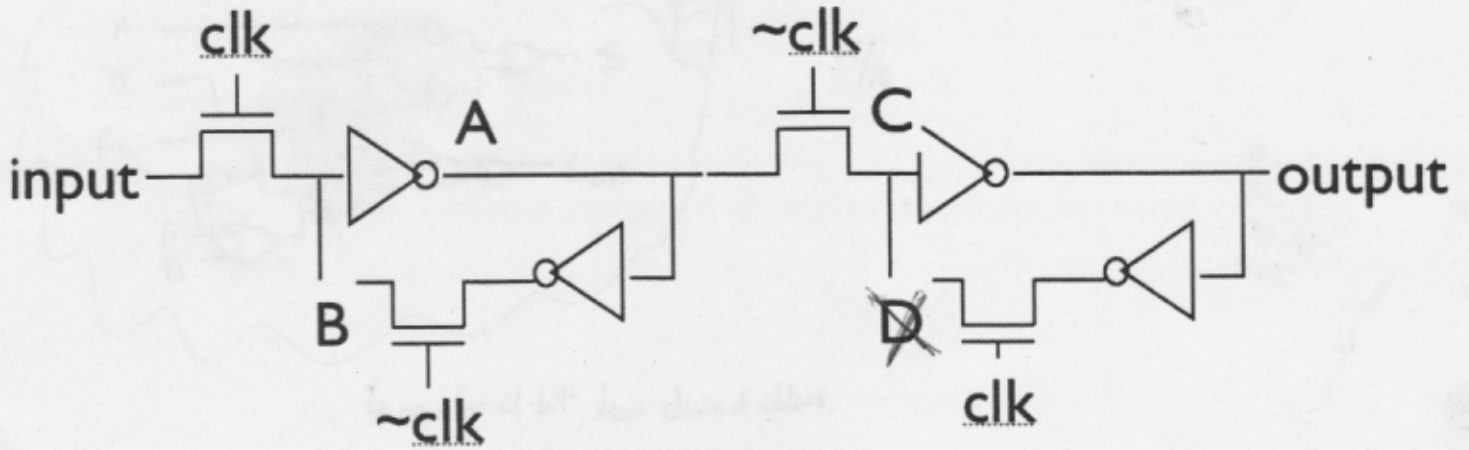


Question 1) Draw the CMOS transistor level circuit that implements the following function:  $\sim(AB + C)$  Use as few transistors as possible. This problem will be graded on the quality of the circuit as well as its correctness.



Question 2) Given the following flip-flop:

Part 1: Draw the gate-level circuit for a 4 bit carry-chain adder:



Fill in the following table. If something cannot be known, place an X in that region.

CLK									
input									
A	X								
B	X								
C	X	X							
D	X	X							
output	X	X							

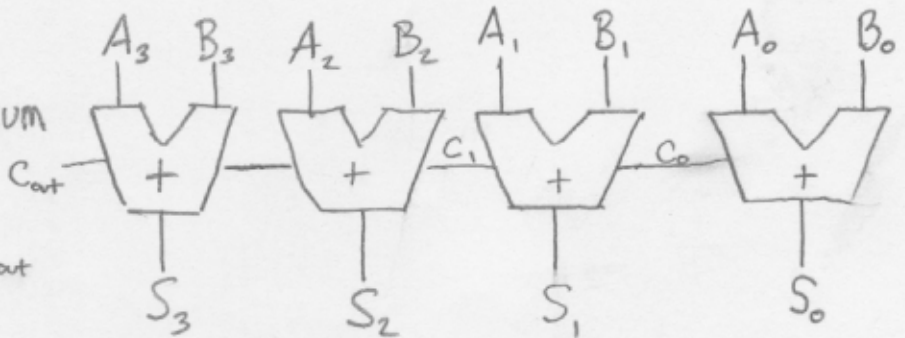
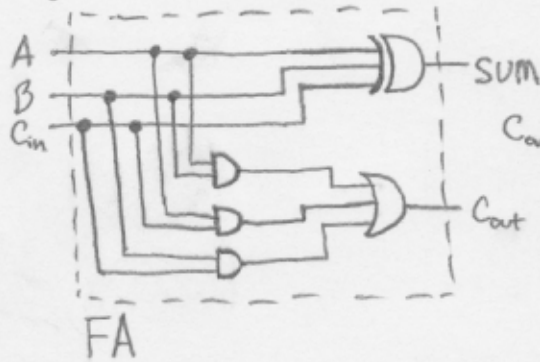
Question 3) (2 pages)

Part 1: Draw the gate-level circuit for a 4 bit carry-chain adder:

$$\text{SUM} = A \oplus B \oplus C_{in}$$


$$\text{Carry} = AB + C_{in}A + BC_{in}$$

gives the 1-bit full-adder



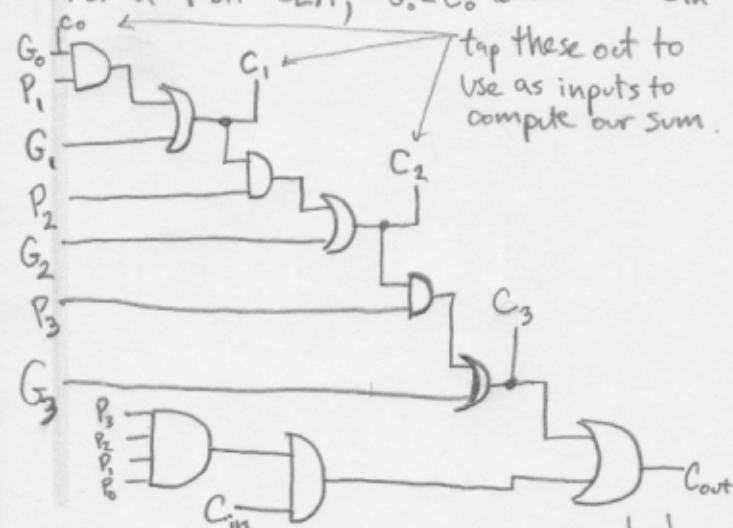
Part 2: Draw the gate-level circuit for a 4 bit carry-lookahead adder: (see Harris & Harris Fig 5.6)

The sum signal for each 1-bit CLA block will use the 3-input XOR

where  $S_i = A_i + B_i + C_{i-1}$ . 

Then draw the carry-logic, we'll simplify our notation with, generate,  $G_i = A_i B_i$  and propagate,  $P_i = A_i + B_i$  signals.

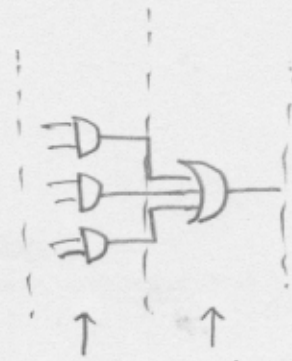
For a 4-bit CLA,  $G_0 = C_0$  with no  $C_{in}$  on the first bit.



which represents the eqn,  $C_{4:i} = G_{4:i} + C_{3:0:i} P_{3:0:i}$

Part 3: Given the following delay model:

- 1 input gates = 1 FO1
- 2 input gates = 3 FO1
- 3 input gates = 5 FO1
- 4 input gates = 6 FO1
- 5 input gates = 7 FO1
- 6 input gates = 8 FO1



take longest delay in each section,

What is the critical path delay of the carry-chain adder (without any significant optimizations).

The critical path in the carry-chain FA is in the carry-logic, see above.

The longest possible path goes through a single 2-input gate (any one of the AND gates) then the 3-input OR gate. So the delay for a single FA is

2-input gate  $\Rightarrow 3FO1$   
 3-input gate  $\Rightarrow 5FO1 = 8FO1$ . We had a 4-bit adder, so then  $4(8FO1) \Rightarrow \boxed{32FO1}$  for our whole chain.

What is the minimum critical path delay of the carry-lookahead adder? (Note carefully that word minimum).

The shortest path (minimum), comes from the  $C_{in}$  input to the  $C_{out}$ , which passes through a 2-input AND then a 2-input OR, as the effect due to that input is only delayed by  $2(3FO1) = \boxed{6FO1}$  given our delay model.



Question 4) Short answer (2 pages!)

4.1) Explain the difference between the = and <= operators in Verilog.

=, blocking operator, current statement must complete before moving onto next (for inter-dependent statements).

=>, non-blocking operator, assignments are concurrently computed, assignments should be independent, gives speed & efficiency gains.

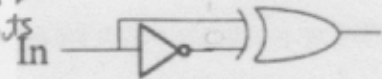
4.2) always@(posedge clk) and always@(clk) are both useful constructs. What is the functionality of each

posedge specifies only rising edge of clock cycle, so only one event per clock cycle, whereas always@(clk) will trigger twice a clock cycle, rising & falling edge.

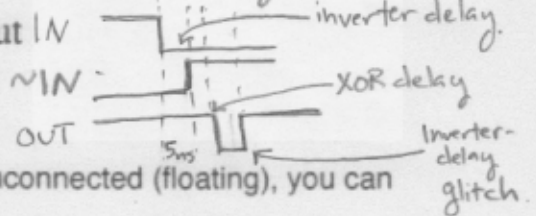
4.3) Can the output of this circuit ever be 0? YES (and possibly NO)

No if we do not consider delay, in pure logic terms we only have 2 possible inputs

IN	~IN	OUT
0	1	1
1	0	1



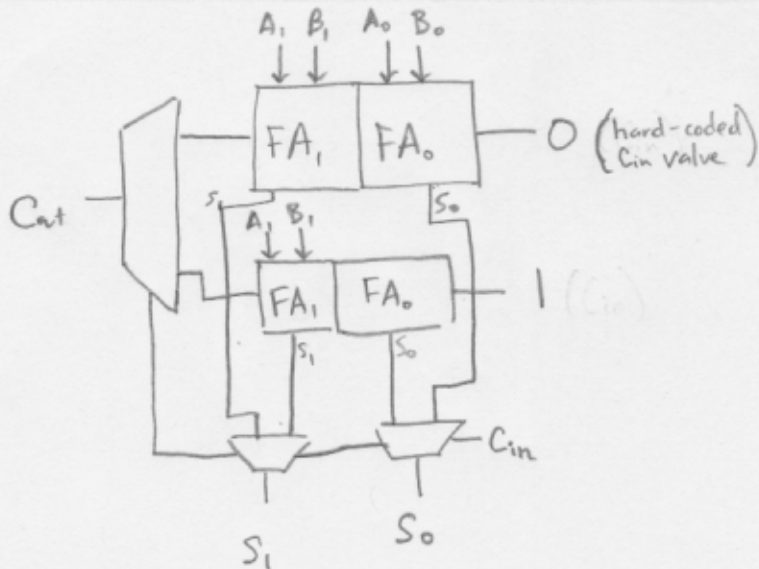
However, we need to think of glitching due to the delay in the inverter,



4.4) If there's an Enable signal on your chip and it is left unconnected (floating), you can observe that the circuit could work occasionally. Why?

Floating pins are susceptible to noise, so a lot of noise on the EN wire can cause the enable to toggle off/on due to noise fluctuations.

4.5) Draw the circuit for a 2 bit carry select adder.



FA = One-bit carry-chain (or ripple carry full adder).

calculates each sum & carry assuming the value for C<sub>in</sub>, then uses the actual value of each C<sub>in<sub>i</sub></sub> to select the correct value.

4.6) Why would you choose to use an ASIC instead of an FPGA to build a product?

- Faster than FPGA
- typically more efficient
- more flexible design.

4.7) Why would you choose to use an FPGA instead of an ASIC to build a product?

- Reconfigurable
- Cheaper, FPGAs have minimal upfront cost.

4.8) Why would you choose to build an ASIC instead of full-custom to build a product?

Significantly faster time to market in less time spent in design iterations, which translates to a higher profit.

4.9) Why is there such a thing as a refresh cycle with DRAM?

The capacitor used to store each value leaks  $e^-$  over time, needs a refreshed charge to maintain its logic state.

4.10) Why are enable lines on chips often active-low?

Lower amounts of noise in the line around 0V versus  $V_{dd}$ . Active low  $\Rightarrow$  more stable signal.