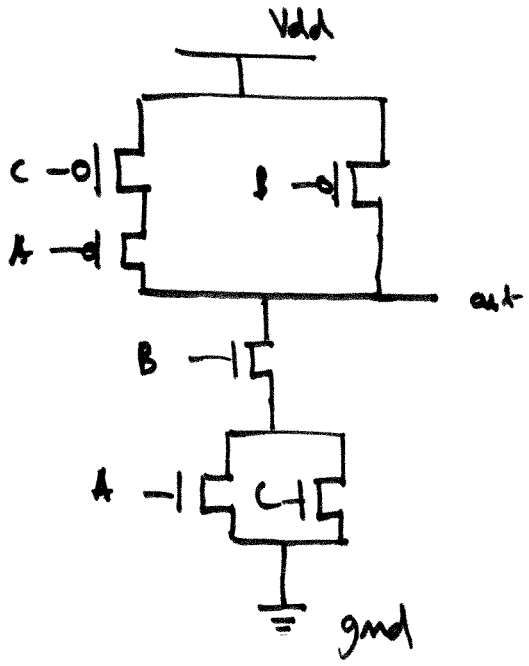
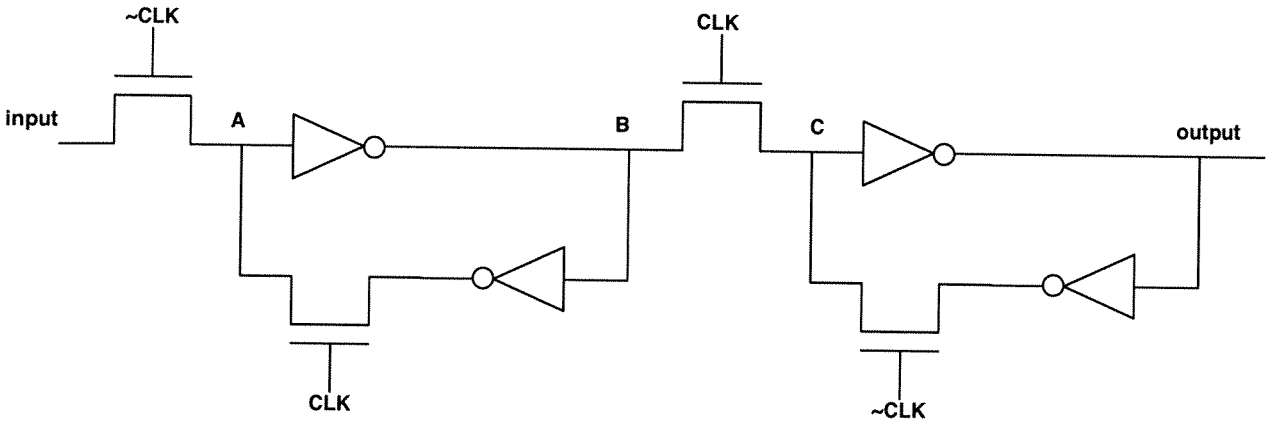
Question 1) CMOS transistor level circuits

Part 1 (20 points): Draw the CMOS transistor level circuit that implements the following function: ~((A + C)B). Use as few transistors as possible. Make sure to label ground, Vdd and the input and output signals. This problem will be graded on the quality of the circuit as well as its correctness.

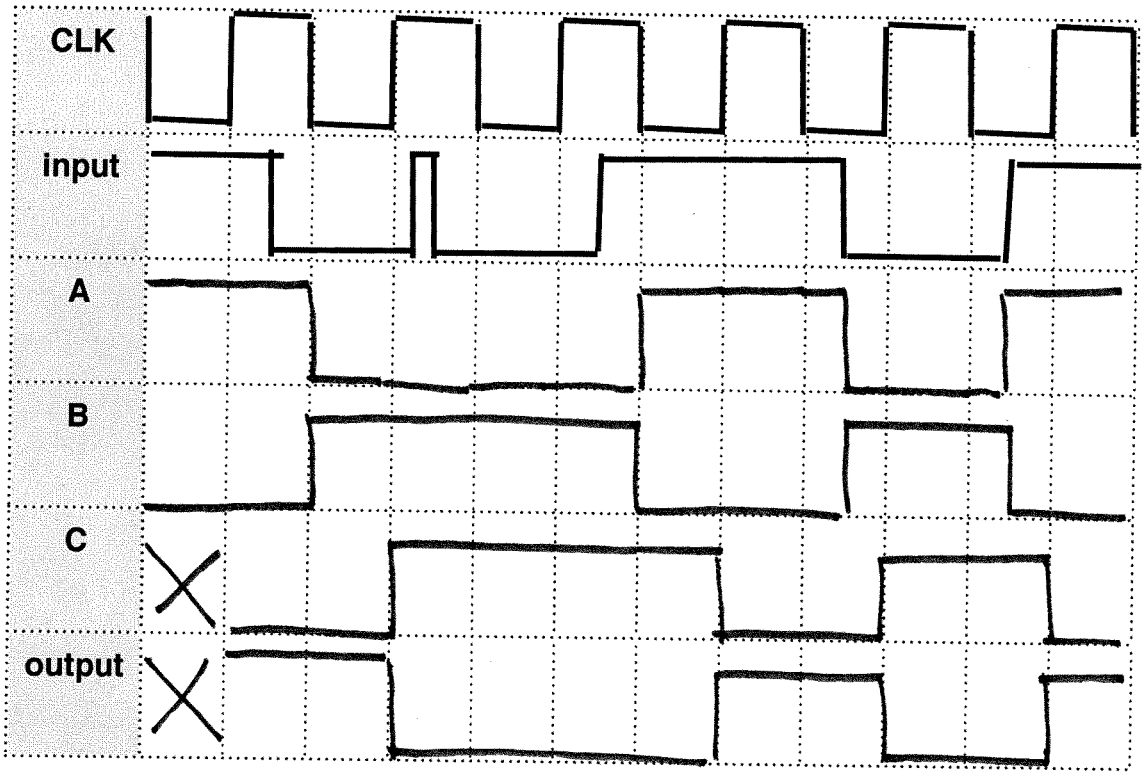


Question 2) Flip-flops

Part 1 (15 points): Given the following flip-flop (PMOS transistors are not shown for simplicity):



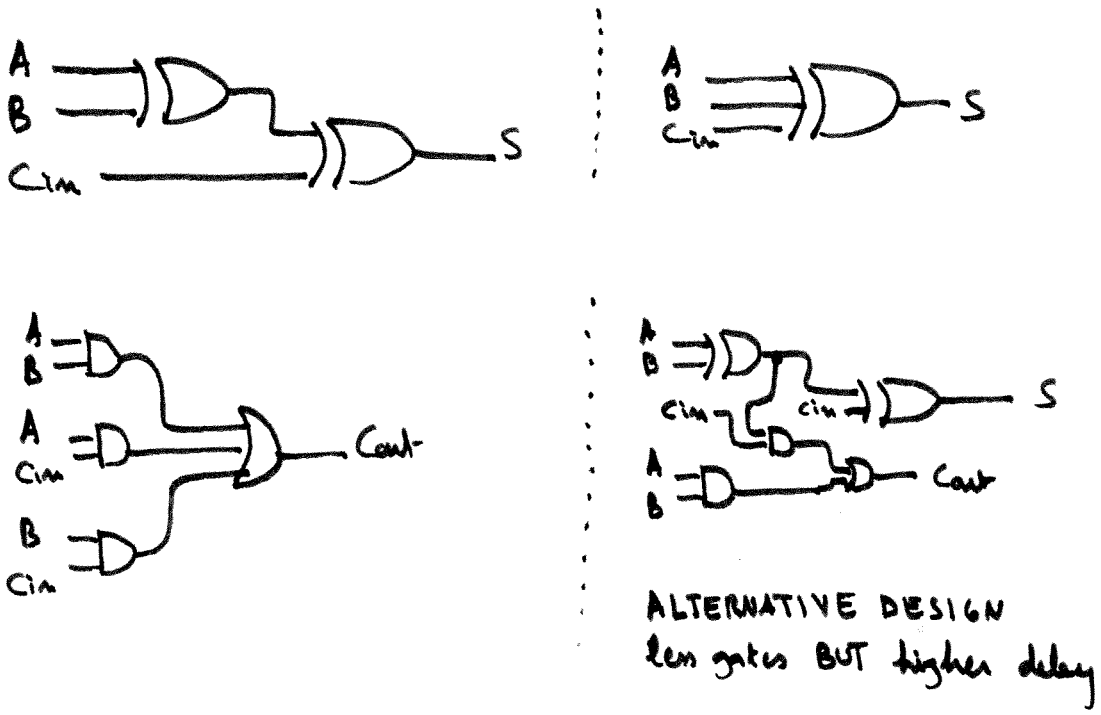
Fill in the following table. If something cannot be known, place an X in that region.



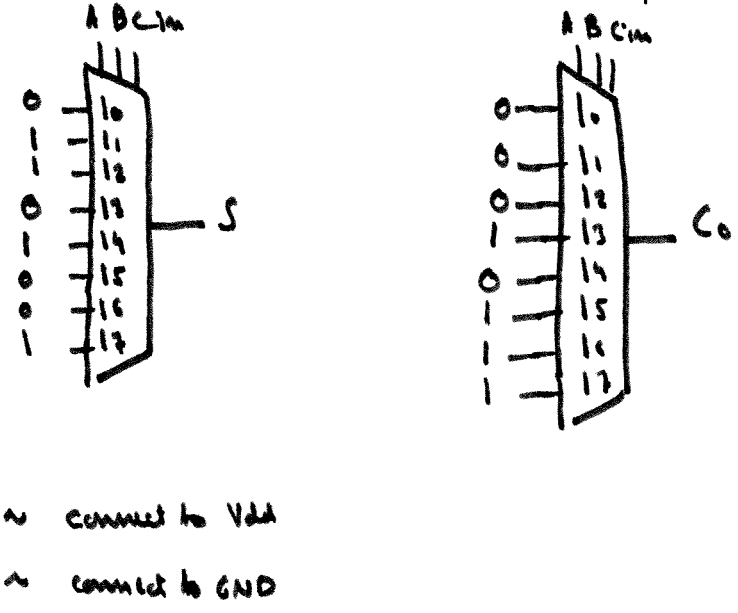
Part 2 (5 points): Is this a positive edge triggered flip flop or negative edge triggered flip flop?

Question 3) Full Adders

Part 1 (10 points): Draw the gate-level circuit for a full adder. Clearly label the inputs, and outputs.



Part 2 (10 points): Implement a full adder using two 8:1 multiplexers. The inputs of the multiplexers can only be connected to Vdd, GND or the inputs of your full adder.

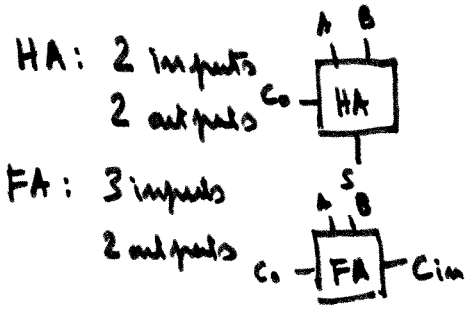


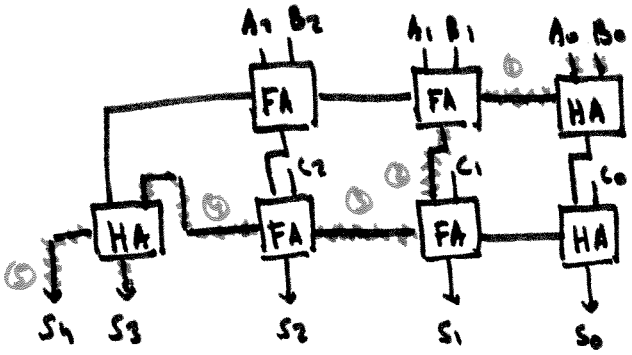
Question 4) Arithmetic Circuits (2 pages!)

Part 1 (10 points): Design a circuit using only full adders (FA) and half adders (HA) that computes the sum of three unsigned 3-bit integers A[2:0], B[2:0], C[2:0]. ***You will be graded based on correctness and efficiency.***

Your full adders and half adders can be represented as block diagrams. For each adder, indicate which output is the sum, and which output is the carry. Make sure to clearly label your circuit to leave no room for ambiguity.

Soln 1





critical path

A0 or B0 → S0 or S4

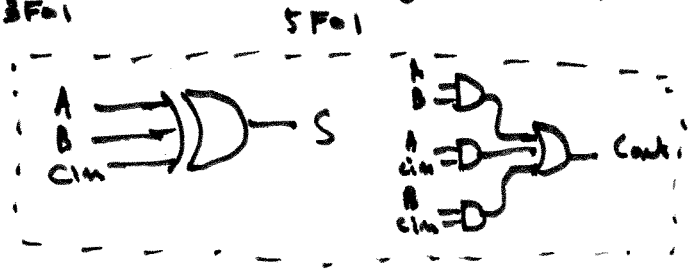
HA: delay from input to C_o: 1 × 2-input gate ∴ delay HA-C = 3FO1

delay from input to S: 1 × 2-input gate ∴ delay HA-S = 3FO1

FA: delay from input to S: 1 × 3-input gate ∴ delay FA-S = 5FO1

delay from input to C_o: 1 × 2-input gate + 1 × 3-input gate ∴ delay FA-C = 8FO1

we assume the following adder:



∴ critical path delay is ① delay HA-C + ② delay FA-S + ③ delay FA-C + ④ delay FA-C + ⑤ delay HA-C

= 3 + 5 + 8 + 8 + 3 = 27FO1

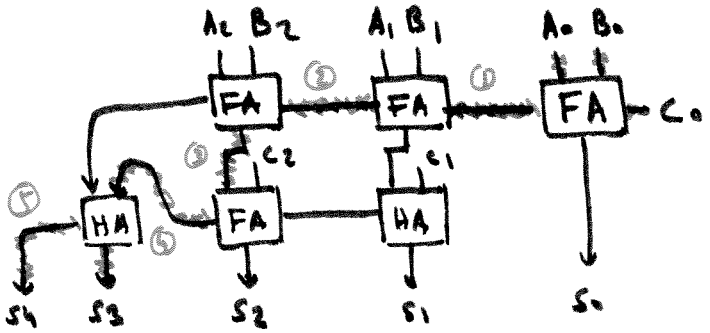
Part 2 (10 points):

Based on the following delays:
- 1 input gate = 1 FO1
- 2 input gate = 3 FO1
- 3 input gate = 5 FO1

(4 points) What is the critical path of your circuit (indicate on your diagram)?

(6 points) What is the critical path **delay** of your circuit? Show your work and state your assumptions.

Soln 2



critical path delay

A0 or B0 or C0 → S0 or S4

∴ critical path delay is :

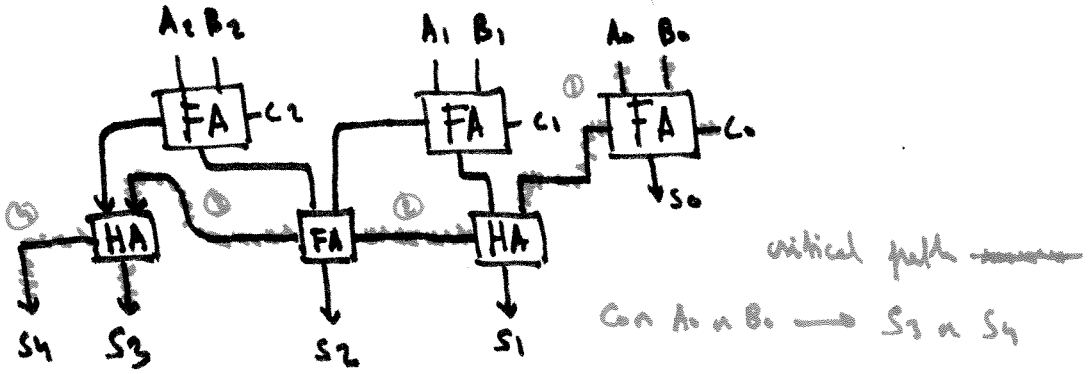
delay FA-C + delay FA-C + delay FA-S + delay FA-C + delay HA-C

= 8 + 8 + 5 + 8 + 3

= 32 FO1 ∴ slower than Soln 1

but 1 less HA used !

Soln 3



critical path delay = delay FA_C + delay HA_C + delay FA_C + delay HA_C

= 8 + 3 + 8 + 3

= 22 Fol

∴ Faster than Soln 1 & soln 2 !

Same resources as soln 1

Question 5) Short answer (2 pages!)

4.1) (3 points) Order SRAM, DRAM and flip-flop by increasing number of transistors used per bit cell?

less DRAM, SRAM, flip-flop more

4.2) (3 points) Provide *one* reason for using FPGAs vs. ASIC to implement a product.

- lower up-front cost
- faster time to market
- simpler design cycle
- field reprogrammability

4.3) (3 points) Why are glitches undesirable?

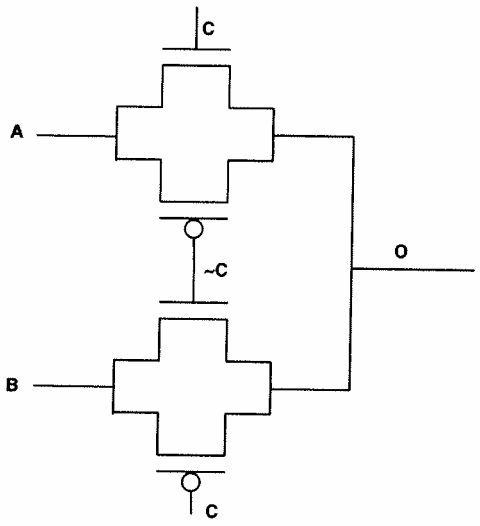
glitches consume power

4.4) (3 points) Discuss the area and delay trade-offs between using a ripple-carry adder vs. a carry-select adder:

Ripple carry adder: smaller area but longer delay

Carry select adder: larger area but shorter delay

4.5) (4 points) What does this circuit implement?



A 2:1 MUX

4.6) (4 points) This is the LFSR that you put together in Lab 3:

Derives the values of A, B, C, D at every clock tick for 4 consecutive clock ticks.

Time	A	B	C	D
t=0	1	0	0	1
t=1	1	1	0	0
t=2	0	1	1	0
t=3	1	0	1	1
t=4	0	1	0	1