1. Conditional sum adder
   a) Find an error in a table that demonstrates conditional sum addition in slide 24, of lecture 19 notes.

   b) Demonstrate the conditional summation by a similar table for inputs $x = 10110110$ and $y = 01001101$. 
2. Carry-skip adder
a) Sketch the design of a 32-b carry skip adder mentioned in this paper that has a maximum of 6 pass transistors in series.
b) Is it possible to design a 32-bit adder with only 5 pass transistors in series? If yes, please sketch that design and compare it to the design in a).

(b) in order to achieve 5 pass transistors worst case, need to build shorters for the critical paths shown, which shortcut arrangement is optimal is probably heavily dependent on device specific, here is a possibility.

the skip cell spanning s2 and s5 provides the fix for bits A and C, but why not span s3 and s2 instead? because we don't want to create another critical path that replaces sp. after all, this p/k signal will take longer to generate than sp, s2, s5, sp.

although extra capacitance added to the carry path might end up increasing the delay, of course there is a remote possibility that Intel didn't do this, because they are over the power budget already.
3. Ling adder
Read the article “A sub-nanosecond 0.5μm 64-bit adder design,” by S. Naffziger presented at 1996 IEEE International Solid-State Circuits Conference, Digest of Technical Papers, pp. 362-363.

a) Reconstruct the key logic equations (the equations that fully describe the operation of this adder, from the inputs to the outputs) in the design of this adder.

b) Draw the parallel prefix tree that this adder implements.
4. Sparse Ling adder

Draw the sum-precompute gates for a sparse 64-bit Ling adder with a sparseness of 2. The final sum for both odd and even bits should be performed by selecting one of the multiple possible precomputed sums using the available carry. Use domino logic.

For bit with $H_{i+1}$ available:

- $S_{i}^{a} = \overline{a_{i} \oplus b_{i}}$
- $S_{i}^{b} = a_{i} \oplus b_{i} \oplus (\overline{a_{i} + b_{i-1}}) = a_{i} \oplus b_{i} \oplus \overline{g_{i-1}}$

Assume $a_{i}, b_{i}$ and their complements $\overline{a_{i}}, \overline{b_{i}}$ are available.

Add a foot transistor to XOR's because the complement signals may not be monotonically rising.
Sums may be selected with MUX2

for bits where only H42 is available

\[ S_i = a_i \oplus b_i \oplus a_i \oplus b_{i-1} = a_i \oplus b_i \oplus k_{i-1} \]

\[ S_i = a_i \oplus b_i \oplus (a_i \oplus b_{i-1} \oplus \overline{(a_i \oplus b_{i-1})}) = a_i \oplus b_i \oplus \overline{j_i-1} \]

Note that this setup requires two clock phases (not including the final one): clk sets up all intermediate j' and k signals, clk2 must be a hard-edge due to illegal domino connections from j and k gates. In general, a hard-edge is expected for domino XOR and MUX like gates. A fast-arising prevents short-circuits from inputs which may not be monotonically rising, Precharge gates may be necessary on the XORs, but are not shown here.