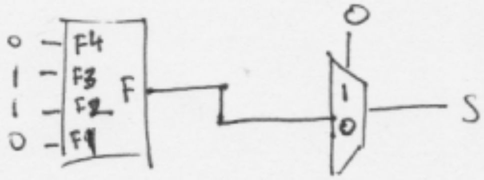
1) For the half-adder we have the truth table

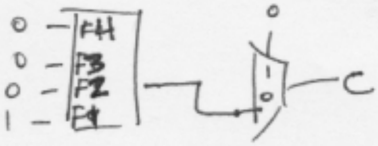
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

For programming/setting the values of the FLUT, we implement the outputs of Sum & carry in 2 separate CLBs,

CLB1 - Sum

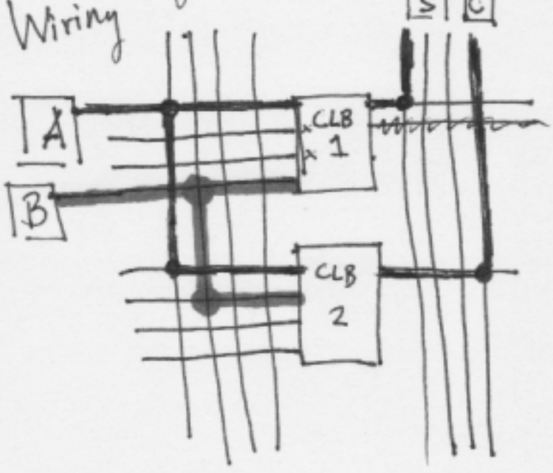


CLB-2 - Carry

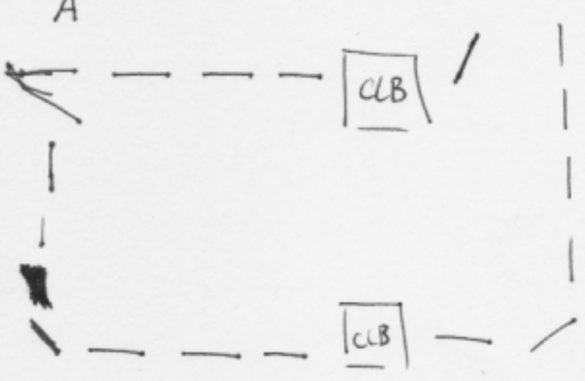


we just select the combinational logic we implemented in the F-LUT, we don't care about using the flip-flop in the CLB.

Finally, to wire this up into our CLB grid,

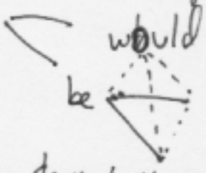


Pass gate connection



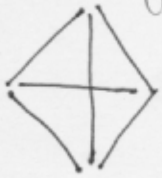
where

— represents the cross-through part of the bridge



down to the right and straight through.

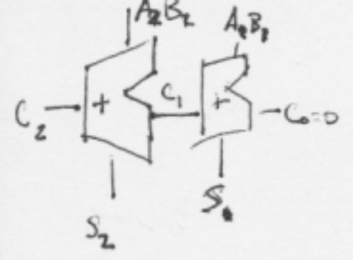
for the pass gate bridge

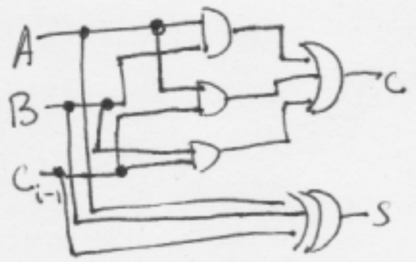


For simplicity our CLB will just be 4inputs on the left and one output on the right, the problem was misleading on this.

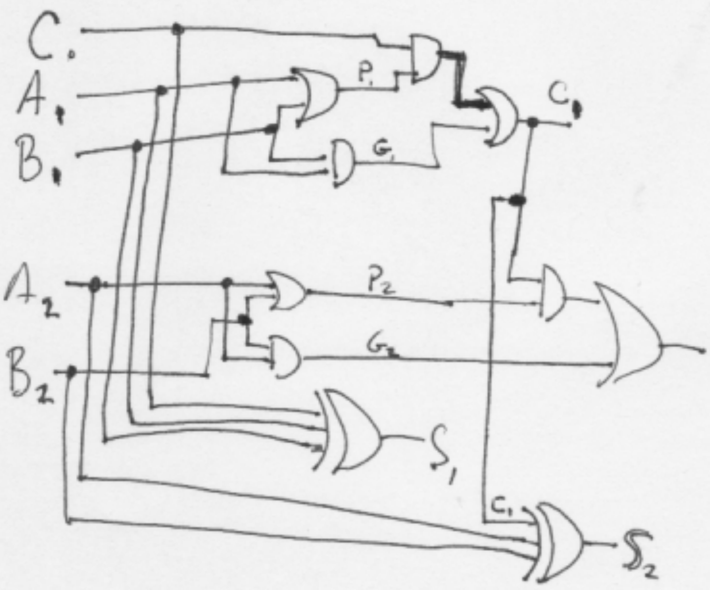
2) Carry look ahead uses more gates, but utilizes them in a more parallel fashion to reduce the overall critical path of the adder. It uses a separate block to compute the carry bit not dependent on the full ~~completion~~ computation like in the ripple adder.

Ripple 2-bit adder





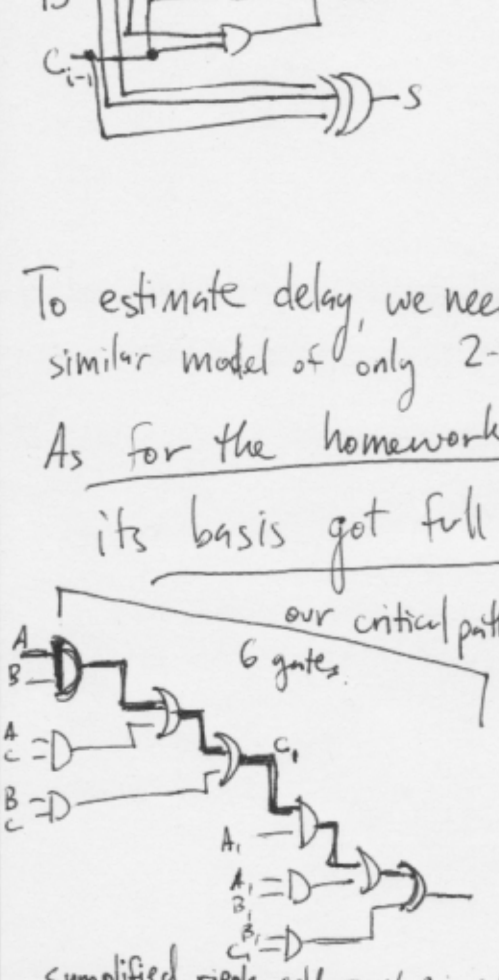
Carry Look Ahead



To estimate delay, we need to model using our critical path, to evaluate this as a similar model of only 2-input gates (the critical path is in the carry signal).

As for the homework grading any valid model using critical path as its basis got full credit, this is one way to make an estimate.

our critical path has 6 gates.

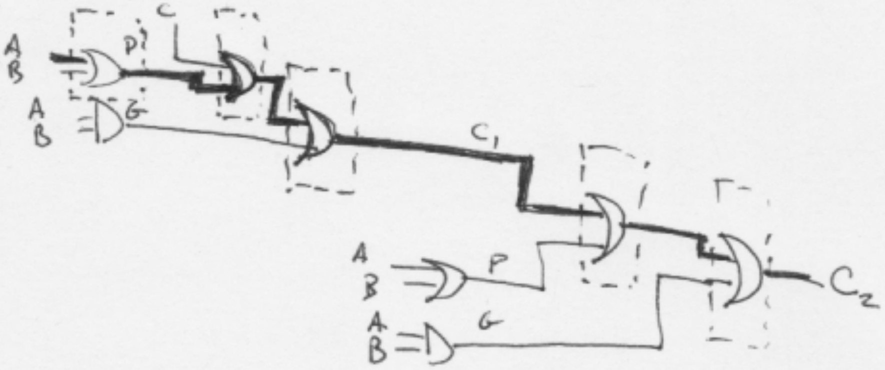


symplified ripple adder w/ 2-input gates.

3gates per FA. We are given the ripple carry FA has a delay of 10 ms,

∴ a 2-input gate has approximately 3.33 ms delay for our model.

Now building a CLA out of 2-input gates,



We see we have a critical path of 5 gates, which using our estimate from the ripple adder delay gives us an estimated delay in the CLA of 16.66 ms.