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

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Sum</th>
<th>Carry</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

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

CLB-1 - Sum

CLB-2 - Carry

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

Finally, to wire up into our CLB grid,

Pass gate connection

For the pass gate bridge

For simplicity our CLB will just be 4 inputs 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 computation computation like in the ripple adder.

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.

\[
\text{our critical path has } 6 \text{ gates.} \quad \frac{3\text{ gates per FA.}}{}
\]

We are given the ripple carry FA has a delay of 10 ms, so:

\[
\quad \text{: a 2-input gate has approximately } 3.33 \text{ ms delay.}
\]

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 $16.66 \text{ ms}$. 