Exercise 6 Solution
I. Ans: Given input \( A[3:0], B[3:0], \) and \( C_{in} \), the result \( S[3:0] \) and the carry out \( C_{out} \) of the 4-bit carry-lookahead adder can be represented as follows.

\[
S[i] = A[i] \oplus B[i] \oplus C[i-1] \quad 0 \leq i \leq 3,
\]

where \( C[-1] \) is \( C_{in} \). In the carry-lookahead adder, the carry out of each bit is calculated by generate, \( G[i] \), and propagate, \( P[i] \), signals. Both signals are defined as

\[
G[i] = A[i]B[i] \quad \text{and} \quad P[i] = A[i] + B[i], \quad 0 \leq i \leq 3.
\]

The carry out of each bit then can be derived from \( G[i] \) and \( P[i] \) as the following equation.

\[
C[i] = G[i] + P[i]C[i-1] \quad 0 \leq i \leq 3,
\]

where \( C[3] \) is \( C_{out} \). After expansion, the expression of each carry out bit is as below:

\[
\begin{align*}
C[0] &= G[0] + P[0]C_{in}, \\
\end{align*}
\]

The schematic diagram is shown as Figure 1.

![Figure 1: 4-bit carry-lookahead adder.](image-url)
II. Ans: (1) Please refer to the figure of slide 27, Lecture 14. The binary tree of output 4 is traced as Figure 2.

Ans: (2) The function of each node is labeled as Figure 2, from $f_0$ to $f_9$. The output functions of each node, except root, include $g_i$ and $p_i$ for leaf nodes and $G_{ij}$ and $P_{ij}$ for internal nodes. We use $<g_i,p_i>$ or $<G_{ij},P_{ij}>$ to represent them. The expressions are listed as below:

$$
\begin{align*}
    f_0 & = <c_in,p_{-1} = 0>, f_1 = <g_0, p_0>, f_2 = <g_1, p_1>, f_3 = <g_2, p_2>, f_4 = <g_3, p_3 > \\
    f_5 & = <G_{0:-1}, P_{0:-1}> \quad \text{where} \\
    G_{0:-1} & = g_0 + p_0 c_{in} \quad \text{and} \quad P_{0:-1} = p_0 p_{-1} \\
    f_6 & = <G_{2:1}, P_{2:1}> \quad \text{where} \\
    G_{2:1} & = g_2 + p_2 g_1 \quad \text{and} \quad P_{2:1} = p_2 p_1 \\
    f_7 & = <G_{2:-1}, P_{2:-1}> \quad \text{where} \\
    G_{2:-1} & = G_{2:1} + P_{2:1} G_{0:-1} = g_2 + p_2 g_1 + p_2 p_1 g_0 + p_2 p_1 p_0 c_{in} \quad \text{and} \quad P_{2:-1} = P_{2:1} P_{0:-1} = p_2 p_1 p_0 p_{-1} = 0 \\
    f_8 & = <G_{3:-1}, P_{3:-1}> \quad \text{where} \\
    G_{3:-1} & = G_{3:3} + P_{3:3} G_{2:-1} = g_3 + p_3 g_2 + p_3 p_2 g_1 + p_3 p_2 p_1 g_0 + p_3 p_2 p_1 p_0 c_{in} \quad \text{and} \quad P_{3:-1} = P_{3:3} P_{2:-1} = p_3 p_2 p_1 p_0 p_{-1} = 0
\end{align*}
$$
III. System Designs: Implement the following algorithm:

Alg(X,Y,C,s)
Input X[7:0], Y[7:0] type bit-vector,
s type boolean;
Output C[7:0] type bit-vector;
Local-object A[7:0], B[7:0] type bit-vector;
S1: if s’ goto S1 else goto S2;
S2: B ← X ∥ A ← Y;
S4: A ← Add(A,1) || goto S6;
S5: B ← Add(B,1);
S6: C ← Add(A,B) || goto S1;
end Alg

Ans: (1) The relationship between statement, operation and control signal is listed in Table. The data subsystem is shown in Figure 3.

<table>
<thead>
<tr>
<th>statement</th>
<th>operation</th>
<th>control signals</th>
</tr>
</thead>
<tbody>
<tr>
<td>B ← X</td>
<td>B ← load(X)</td>
<td>C₁, C₃</td>
</tr>
<tr>
<td>A ← Y</td>
<td>A ← load(Y)</td>
<td>C₀, C₂</td>
</tr>
<tr>
<td>A ← Add(A, B)</td>
<td>A ← Add(A, B)</td>
<td>C₀, C₂, C₄</td>
</tr>
<tr>
<td>A ← Add(A, 1)</td>
<td>A ← Inc(A)</td>
<td>C₀, C₂</td>
</tr>
<tr>
<td>B ← Add(B, 1)</td>
<td>B ← Inc(B)</td>
<td>C₁, C₃</td>
</tr>
<tr>
<td>C ← Add(A, B)</td>
<td>C ← Add(A, B)</td>
<td>C₅</td>
</tr>
</tbody>
</table>

Figure 3: Data subsystem of system design 1.
Ans: (2) The state diagram can be derived from the algorithm. The state diagram is shown in Figure 4. Table shows values of the control signals in each state.

![State Diagram](image)

**Figure 4: State diagram of system design 1.**

<table>
<thead>
<tr>
<th></th>
<th>$C_0$</th>
<th>$C_1$</th>
<th>$C_2$</th>
<th>$C_3$</th>
<th>$C_4$</th>
<th>$C_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>S2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>S3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>S4</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>S5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>S6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>1</td>
</tr>
</tbody>
</table>

**Table 2: Control subsystem.**

Ans: (3) The implementation is shown in Figure 5.

![Implementation Diagram](image)

**Figure 5: Implementation of system design 1.**
IV. System Designs: Implement the following algorithm:

```
Alg(X,Y,Z,start,U,done)
Input X[7:0], Y[7:0], Z[7:0], start;
Output U[7:0], done;
Local-object A[7:0], B[7:0], C[7:0];
S1: If start’ goto S1;
S2: done ← 0 || A ← X || B ← Y || C ← Z;
S3: A ← Add(A,B);
S4: If B’[7] goto S3 || B ← Inc(B);
S5: If C’[7] goto S3 || C ← Inc(C);
S6: U ← A || done ← 1 || goto S1;
End Alg
```

Ans: (1) The relationship between statement, operation and control signal is listed in Table. The data subsystem is shown in Figure 6.

<table>
<thead>
<tr>
<th>statement</th>
<th>operation</th>
<th>control signals</th>
</tr>
</thead>
<tbody>
<tr>
<td>A ← X</td>
<td>A ← load(X)</td>
<td>C₀, C₅</td>
</tr>
<tr>
<td>B ← Y</td>
<td>B ← load(Y)</td>
<td>C₁, C₃</td>
</tr>
<tr>
<td>C ← Z</td>
<td>C ← load(Z)</td>
<td>C₂, C₄</td>
</tr>
<tr>
<td>A ← Add(A, B)</td>
<td>A ← Add(A, B)</td>
<td>C₀, C₅</td>
</tr>
<tr>
<td>B ← Inc(B)</td>
<td>B ← Inc(B)</td>
<td>C₁, C₃</td>
</tr>
<tr>
<td>C ← Inc(C)</td>
<td>C ← Inc(C)</td>
<td>C₂, C₄</td>
</tr>
<tr>
<td>U ← A</td>
<td>wire</td>
<td>done</td>
</tr>
</tbody>
</table>

Table 3: System design 2.

Figure 6: Data subsystem of system design 2.
Ans: (2)(i) The state diagram can be derived from the algorithm. The state diagram is shown in Figure 7. Table shows values of the control signals in each state.

Ans: (2)(ii) The implementation is shown in Figure 8.