Addition & Subtraction

Addition and Subtraction

• Motivation: need to add and/or subtract values represented as binary numbers
  – Note need for identification of coding method: unsigned binary, two’s complement, fixed point, floating point, excess code, …
  – Scheme should be extensible; that is, once method identified, should be able to apply method with varying numbers of bits
Adding Two 8-bit Numbers

\[
\begin{array}{cccccccc}
1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
\hline
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0
\end{array}
\]

Adding Two 8-bit Numbers

\[
\begin{array}{cccccccc}
1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\
1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
\hline
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0
\end{array}
\]
### Full Adder Truth Table

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Cin</th>
<th>Cout</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

### K-Map for Three Variables

![K-Map for Three Variables](image)
K-Map Construction

Full Adder – K-Map for F
Full Adder – Kmap for F

\[ F = \overline{A} \cdot \overline{B} \cdot C + \overline{A} \cdot B \cdot \overline{C} + A \cdot B \cdot C + A \cdot \overline{B} \cdot \overline{C} \]

Full Adder – K-Map for Cout

\[ C_{out} = \overline{A} \cdot \overline{B} \cdot C + \overline{A} \cdot B \cdot \overline{C} + A \cdot B \cdot C + A \cdot \overline{B} \cdot \overline{C} \]
Full Adder – K-Map for Cout

Cout = \( B \cdot C + A \cdot B + A \cdot C \)
Basic Identities of Boolean Algebra

\[
\begin{align*}
X + 0 &= X \\
X + 1 &= 1 \\
X + X &= X \\
X + \overline{X} &= 1 \\
\overline{X} &= X
\end{align*}
\]

\[
\begin{align*}
X \cdot 1 &= X \\
X \cdot 0 &= 0 \\
X \cdot X &= X \\
X \cdot \overline{X} &= 0
\end{align*}
\]

Basic Identities of Boolean Algebra

\[
\begin{align*}
X + Y &= Y + X \\
X \cdot Y &= Y \cdot X \\
X + (Y + Z) &= (Y + X) + Z \\
X(YZ) &= (YX)Z \\
X(Y + Z) &= XY + XZ \\
X + YZ &= (X + Y)(X + Z)
\end{align*}
\]

\[
\begin{align*}
\overline{X + Y} &= \overline{X} \cdot \overline{Y} \\
\overline{X \cdot Y} &= \overline{X} + \overline{Y}
\end{align*}
\]
Boolean Algebra

\[ X + XY = X(1 + Y) = X \]
\[ XY + X\overline{Y} = X(Y + \overline{Y}) = X \]
\[ X + \overline{XY} = (X + \overline{X})(X + Y) = X + Y \]
\[ X(X + Y) = X + XY = X \]
\[ (X + Y)(X + \overline{Y}) = X + Y\overline{Y} = X \]
\[ X(\overline{X} + Y) = X\overline{X} + XY = XY \]
\[ XY + \overline{XZ} + YZ = XY + \overline{XZ} \]

---

Full Subtractor

\[ \begin{array}{ccc}
X & Y & Bin \\
\uparrow & \uparrow & \uparrow \\
\downarrow & \downarrow & \downarrow \\
\text{Bout} & \text{F} & \text{Full Subtractor} \\
\end{array} \]
### Full Subtractor Truth Table

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Bin</th>
<th>Bout</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

### Full Subtractor Truth Table

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Bin</th>
<th>Bout</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Full Subtractor – K-Map for $F$

$$F = X \cdot \overline{Y} \cdot B + \overline{X} \cdot Y \cdot \overline{B} + X \cdot Y \cdot B + X \cdot \overline{Y} \cdot \overline{B}$$

Full Subtractor – K-Map for $Bout$
Full Subtractor – K-Map for Bout

Bout = \overline{X} \cdot B + \overline{X} \cdot Y + Y \cdot B

Adder/Subtractor

ADD A B CBin

Adder/Subtractor

CBout F
<table>
<thead>
<tr>
<th>ADD</th>
<th>A</th>
<th>B</th>
<th>CBin</th>
<th>Cbout</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

K-Map For Four Variables

![K-Map](Image)
K-Map Construction (4 bits)

K-Map For F
### K-Map For F

\[ F = \overline{A} \cdot B \cdot C_{Bin} + \overline{A} \cdot B \cdot \overline{C_{Bin}} + A \cdot B \cdot C_{Bin} + A \cdot \overline{B} \cdot \overline{C_{Bin}} \]

### K-Map For CBout

\[ \text{CBin} \]

\[ \text{ADD} \]
K-Map For CBout

CBout = ADD \cdot \overline{A} \cdot CBin + ADD \cdot \overline{A} \cdot B + ADD \cdot A \cdot CBin + ADD \cdot A \cdot B + B \cdot CBin

Timing for Ripple Carry Adder
Return to Carry Equation

\[ Cout = A \cdot B + A \cdot C + B \cdot C \]

\[ Cout = A \cdot B + (A + B) \cdot C \]
Look Ahead System (2 Bits)

CX = G₀ + P₀Cᵢₙ

CY = G₁ + P₁CX
   = G₁ + P₁G₀ + P₁P₀Cᵢₙ

Generate = G₁ + P₁G₀
Propagate = P₁P₀
Look Ahead Method
Look Ahead System

CX = G₀ + P₀C_IN

CY = G₁ + P₁CX
  = G₁ + P₁G₀ + P₁P₀C_IN

CZ = G₂ + P₂CY
  = G₂ + P₂G₁ + P₂P₁G₀ + P₂P₁P₀C_IN
Look Ahead System

\[ C_{\text{NEXT}} = G_3 + P_3CZ \]
\[ = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0 + P_3P_2P_1P_0C_{\text{IN}} \]

\[ G_{\text{OUT}} = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0 \]

\[ P_{\text{OUT}} = P_3P_2P_1P_0 \]

Look-Ahead, 4 bits wide…