Binary Arithmetic

COE 202
Digital Logic Design
Dr. Muhamed Mudawar
King Fahd University of Petroleum and Minerals
Adding Bits

- $1 + 1 = 2$, but 2 should be represented as $(10)_2$ in binary
- Adding two bits: the sum is $S$ and the carry is $C$

<table>
<thead>
<tr>
<th>X</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ Y</td>
<td>+ 0</td>
<td>+ 1</td>
<td>+ 0</td>
<td>+ 1</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>S</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

- Adding three bits: the sum is $S$ and the carry is $C$

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>+ 0</td>
<td>+ 1</td>
<td>+ 0</td>
<td>+ 1</td>
<td>+ 0</td>
<td>+ 1</td>
<td>+ 0</td>
<td>+ 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Binary Addition

- Start with the least significant bit (rightmost bit)
- Add each pair of bits
- Include the carry in the addition, if present

```
carry 1 1 1 1

0 0 1 1 0 1 1 0  (54)
+ 0 0 0 1 1 1 0 1  (29)
-------
0 1 0 1 0 0 1 1  (83)
```

bit position: 7 6 5 4 3 2 1 0
Subtracting Bits

- Subtracting 2 bits \((X - Y)\): we get the difference \((D)\) and the **borrow-out** \((B)\) shown as 0 or -1

\[
\begin{array}{c c c c c c}
& X & 0 & 0 & 1 & 1 \\
- Y & -0 & -1 & -0 & -1 \\
\hline
B & 0 & 0 & -1 & 0 & 0 \\
D & 0 & 1 & 0 & 0 & 0 \\
\end{array}
\]

- Subtracting two bits \((X - Y)\) with a **borrow-in** \(= -1\): we get the difference \((D)\) and the **borrow-out** \((B)\)

\[
\begin{array}{c c c c c c}
\text{borrow-in} & -1 & -1 & -1 & -1 & -1 \\
& X & 0 & 0 & 1 & 1 \\
- Y & -0 & -1 & -0 & -1 \\
\hline
B & -1 & 1 & -1 & 0 & -1 \\
D & -1 & 0 & 0 & 0 & -1 \\
\end{array}
\]
Binary Subtraction

- Start with the least significant bit (rightmost bit)
- Subtract each pair of bits
- Include the borrow in the subtraction, if present

\[
\begin{array}{ccccccc}
borrow & -1 & -1 & -1 \\
0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\
\hline
- & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\
\hline
0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\
\end{array}
\]

bit position: 7 6 5 4 3 2 1 0

= (25)

= (29)

= (54)
Binary Multiplication

- Binary Multiplication table is simple:
  
  \[0 \times 0 = 0, \quad 0 \times 1 = 0, \quad 1 \times 0 = 0, \quad 1 \times 1 = 1\]

<table>
<thead>
<tr>
<th>Multiplicand</th>
<th>(1100_2 = 12)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiplier</td>
<td>(\times 1101_2 = 13)</td>
</tr>
</tbody>
</table>

\[\begin{array}{c}
1100 \\
0000 \\
1100 \\
1100 \\
\end{array}\]

Product \(10011100_2 = 156\)

- \(n\)-bit multiplicand \(\times\) \(n\)-bit multiplier = \(2n\)-bit product
- Accomplished via shifting and addition

Binary multiplication is easy

- \(0 \times\) multiplicand = 0
- \(1 \times\) multiplicand = multiplicand
Hexadecimal Addition

- Start with the least significant hexadecimal digits
- Let Sum = summation of two hex digits
- If Sum is greater than or equal to 16
  - Sum = Sum – 16 and Carry = 1
- Example:

```
carry  1 1 1
  9 C 3 7 2 8 6 5
+ 1 3 9 5 E 8 4 B
------------------
A F C D 1 0 B 0
```

5 + B = 5 + 11 = 16
Since Sum ≥ 16
Sum = 16 – 16 = 0
Carry = 1
Hexadecimal Subtraction

- Start with the least significant hexadecimal digits
- Let Difference = subtraction of two hex digits
- If Difference is negative
  - Difference = 16 + Difference and Borrow = -1
- Example:

\[
\begin{array}{cccccc}
\text{borrow} & -1 & -1 & -1 & -1 \\
9 & C & 3 & 7 & 2 & 8 & 6 & 5 \\
\hline
1 & 3 & 9 & 5 & E & 8 & 4 & B \\
\hline
8 & 8 & A & 1 & 4 & 0 & 1 & A \\
\end{array}
\]

Since 5 < B, Difference < 0
Difference = 16+5–11 = 10
Borrow = -1