In this chapter you will learn about:
• Representation of numbers in computers;
• Circuits used to perform arithmetic operations;
• Performance issues in large circuits.
7-Bit ASCII Code

<table>
<thead>
<tr>
<th>Bit positions 000</th>
<th>Bit positions 004</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0000</td>
</tr>
<tr>
<td>0001</td>
<td>0001</td>
</tr>
<tr>
<td>0010</td>
<td>0010</td>
</tr>
<tr>
<td>0011</td>
<td>0011</td>
</tr>
<tr>
<td>0100</td>
<td>0100</td>
</tr>
<tr>
<td>0101</td>
<td>0101</td>
</tr>
<tr>
<td>0110</td>
<td>0110</td>
</tr>
<tr>
<td>0111</td>
<td>0111</td>
</tr>
<tr>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>1001</td>
<td>1001</td>
</tr>
<tr>
<td>1010</td>
<td>1010</td>
</tr>
<tr>
<td>1011</td>
<td>1011</td>
</tr>
<tr>
<td>1100</td>
<td>1100</td>
</tr>
<tr>
<td>1101</td>
<td>1101</td>
</tr>
<tr>
<td>1110</td>
<td>1110</td>
</tr>
<tr>
<td>1111</td>
<td>1111</td>
</tr>
</tbody>
</table>

NUL  Space  SI  Shift in
SOH  Start of header  DEL  Data link escape
STX  Start of text  DLE  Device control
ETX  End of text  NAK  Negative acknowledgment
EOT  End of transmission  SYN  Synchronous idle
ENQ  Enquiry  ETB  End of transmitted block
ACK  Acknowledge  CAN  Cancel (error in data)
BEL  Bell  BS  End of record
BS  Back space  SUB  Special sequence
HT  Horizontal tab  ESC  Escape
LF  Line feed  FF  File separator
VT  Vertical tab  GS  Group separator
FF  Form feed  RS  Record separator
CR  Carriage return  US  Tash separator
SO  Shift out  DEL  Delete/Backspace

Bit positions of code format: 010111111111

Unsigned vs. Signed Numbers

(a) Unsigned number

\[ b_{n-1} \ldots b_1 b_0 \]

MSB

Magnitude

(b) Signed number

\[ b_{n-1} b_{n-2} \ldots b_1 b_0 \]

Sign

0 denotes +

1 denotes –

MSB

Magnitude
Interpretation of Four-Bit Signed Integers

<table>
<thead>
<tr>
<th>abcd</th>
<th>Sign and magnitude</th>
<th>1’s complement</th>
<th>2’s complement</th>
</tr>
</thead>
<tbody>
<tr>
<td>0111</td>
<td>+7</td>
<td>+7</td>
<td>+7</td>
</tr>
<tr>
<td>0110</td>
<td>+6</td>
<td>+6</td>
<td>+6</td>
</tr>
<tr>
<td>0101</td>
<td>+5</td>
<td>+5</td>
<td>+5</td>
</tr>
<tr>
<td>0100</td>
<td>+4</td>
<td>+4</td>
<td>+4</td>
</tr>
<tr>
<td>0011</td>
<td>+3</td>
<td>+3</td>
<td>+3</td>
</tr>
<tr>
<td>0010</td>
<td>+2</td>
<td>+2</td>
<td>+2</td>
</tr>
<tr>
<td>0001</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
</tr>
<tr>
<td>0000</td>
<td>+0</td>
<td>+0</td>
<td>0</td>
</tr>
<tr>
<td>1000</td>
<td>-0</td>
<td>-7</td>
<td>-8</td>
</tr>
<tr>
<td>1001</td>
<td>-1</td>
<td>-6</td>
<td>-7</td>
</tr>
<tr>
<td>1010</td>
<td>-2</td>
<td>-5</td>
<td>-6</td>
</tr>
<tr>
<td>1011</td>
<td>-3</td>
<td>-4</td>
<td>-5</td>
</tr>
<tr>
<td>1100</td>
<td>-4</td>
<td>-3</td>
<td>-4</td>
</tr>
<tr>
<td>1101</td>
<td>-5</td>
<td>-2</td>
<td>-3</td>
</tr>
<tr>
<td>1110</td>
<td>-6</td>
<td>-1</td>
<td>-2</td>
</tr>
<tr>
<td>1111</td>
<td>-7</td>
<td>-0</td>
<td>-1</td>
</tr>
</tbody>
</table>

Sign and Magnitude

- Magnitude of positive and negative numbers represented in the same way;
- Sign used to distinguish them;
- Simple to understand;
- Complicates hardware design.
Sign and Magnitude: Add/Subtract

- If both operands have the same sign, then addition is simple
  - Add magnitudes and copy the sign.
- If they have opposite signs, then must subtract smaller from the larger.
- This is complicated, so sign and magnitude is not used in computers anymore.

1’s Complement

- n-bit negative number found by subtracting its positive form from $2^n - 1$
  - $K_t = (2^n - 1) - P$
- Found by just complementing each bit.
- Used in earlier computers (pre-1970) but made hardware difficult with two representations for 0.
2’s Complement

- n-bit negative number found by subtracting its positive form from $2^n$
  - $K_2 = 2^n - P$
  - $K_2 = K_1 + 1$
- Complement each bit and add 1.
- Probably the only form for signed numbers used in computers today.

### Graphical Interpretation of Four-bit 2’s Complement Numbers

![Graphical Interpretation of Four-bit 2’s Complement Numbers](image-url)
2’s Complement Addition Examples

\[
\begin{align*}
(+5) & \quad 0101 \\
+ (+2) & \quad +0010 \\
\hline
(+7) & \quad 0111 \\
\end{align*}
\]

\[
\begin{align*}
(-5) & \quad 1011 \\
+ (+2) & \quad +0010 \\
\hline
(-3) & \quad 1101 \\
\end{align*}
\]

\[
\begin{align*}
(+5) & \quad 0101 \\
+ (-2) & \quad +1110 \\
\hline
(+3) & \quad 10011 \\
\end{align*}
\]

\[
\begin{align*}
(-5) & \quad 1011 \\
+ (-2) & \quad +1110 \\
\hline
(-7) & \quad 11001 \\
\end{align*}
\]

ignore

ignore

2’s Complement Subtraction Examples

\[
\begin{align*}
(+5) & \quad 0101 \\
- (+2) & \quad -0010 \\
\hline
(+3) & \quad 0011 \\
\end{align*}
\]

\[
\begin{align*}
0101 & \quad 0101 \\
+1110 & \quad +1110 \\
\hline
10011 & \quad 10011 \\
\end{align*}
\]

ignore

\[
\begin{align*}
(-5) & \quad 1011 \\
- (+2) & \quad -0010 \\
\hline
(-7) & \quad 11001 \\
\end{align*}
\]

\[
\begin{align*}
1011 & \quad 1011 \\
+1110 & \quad +1110 \\
\hline
11001 & \quad 11001 \\
\end{align*}
\]

ignore

\[
\begin{align*}
(+5) & \quad 0101 \\
- (-2) & \quad -1110 \\
\hline
(+7) & \quad 0111 \\
\end{align*}
\]

\[
\begin{align*}
0101 & \quad 0101 \\
+0010 & \quad +0010 \\
\hline
0111 & \quad 0111 \\
\end{align*}
\]

\[
\begin{align*}
(-5) & \quad 1011 \\
- (-2) & \quad -1110 \\
\hline
(-3) & \quad 1101 \\
\end{align*}
\]

\[
\begin{align*}
1011 & \quad 1011 \\
+0010 & \quad +0010 \\
\hline
1101 & \quad 1101 \\
\end{align*}
\]
Adder Circuits – Half Adder

(a) The four possible cases

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>c</th>
<th>s</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>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

(b) Truth table

(c) Circuit

(d) Graphical symbol

Adder Circuits – Full Adder

(a) Truth table

<table>
<thead>
<tr>
<th>C_i</th>
<th>x_i</th>
<th>y_i</th>
<th>c_{i+1}</th>
<th>s_i</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>1</td>
<td>0</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>

(b) Karnaugh maps

(c) Circuit
Alternative Full Adder

(a) Block diagram

(b) Detailed diagram

N-bit Ripple Carry Adder

• Major issue – propagation delay.
Faster Adders – Carry Lookahead

Adder / Subtractor Circuit
Binary Coded Decimal (BCD) Digits

<table>
<thead>
<tr>
<th>Decimal digit</th>
<th>BCD code</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0000</td>
</tr>
<tr>
<td>1</td>
<td>0001</td>
</tr>
<tr>
<td>2</td>
<td>0010</td>
</tr>
<tr>
<td>3</td>
<td>0011</td>
</tr>
<tr>
<td>4</td>
<td>0100</td>
</tr>
<tr>
<td>5</td>
<td>0101</td>
</tr>
<tr>
<td>6</td>
<td>0110</td>
</tr>
<tr>
<td>7</td>
<td>0111</td>
</tr>
<tr>
<td>8</td>
<td>1000</td>
</tr>
<tr>
<td>9</td>
<td>1001</td>
</tr>
</tbody>
</table>

One Digit BCD Adder
Other Arithmetic Circuits

- Multipliers;
- Dividers;
- Floating-point Processors.

Summary

In this chapter you learned about:
- Representation of numbers in computers;
- Circuits used to perform arithmetic operations;
- Performance issues in large circuits.