Complements

- Subtraction of numbers requires a different algorithm from that for addition
- Adding the complement of a number is equivalent to subtraction
- We will discuss two complements:
  - Diminished Radix Complement
  - Radix Complement
- Subtraction will be done by adding the complement of the subtrahend
Diminished Radix Complement

- Given a number $N$ in Base $r$ having $n$ digits, the $(r - 1)'s$-complement (called the Diminished Radix Complement) is defined as: $(r^n - 1) - N$

- Example:
  - For $r = 10$, $N = 1234_{10}$, $n = 4$ (4 digits), we have:
    
    $(r^n - 1) = 10,000 - 1 = 9999_{10}$

  - The 9's complement of 1234\(_{10}\) is then:
    
    $9999_{10} - 1234_{10} = 8765_{10}$

Binary 1's Complement

- For $r = 2$, $N = 01110011_2$, $n = 8$ (8 digits):
  
  $(r^n - 1) = 256 - 1 = 255_{10}$ or $11111111_2$

- The 1's complement of 01110011\(_2\) is then:
  
  $11111111$

  $- 01110011$

  $10001100$

- Since the $2^n - 1$ factor consists of all 1's and since $1 - 0 = 1$ and $1 - 1 = 0$, the one's complement is obtained by complementing each individual bit (bitwise NOT).
Radix Complement

- Given a number $N$ in Base $r$ having $n$ digits, the $r$'s complement (called the radix complement) is defined as:
  - $r^n - N$ for $N \neq 0$ and
  - 0 for $N = 0$
- The radix complement is obtained by adding 1 to the diminished radix complement
- Example:
  - For $r = 10$, $N = 1234_{10}$, $n = 4$ (4 digits), we have: $r^n = 10,000_{10}$
  - The 10's complement of 1234 is then $10,000 - 1234 = 8766_{10}$ or $8765 + 1$ (9's complement plus 1)

Binary 2's Complement

- For $r = 2$, $N = 01110011_2$, $n = 8$ (8 digits), we have: $(r^n) = 25610$ or $1000000002$
- The 2's complement of 01110011 is then: $100000000 - 01110011 = 10001101$
- Note the result is the 1's complement plus 1
Alternate 2’s Complement

- Given: an n-bit binary number, beginning at the right and proceeding left:
  - Copy all least significant 0’s
  - Copy the first 1
  - Complement all bits thereafter.

- 2’s Complement Example:
  
  10010100
  - Copy underlined bits:
    
    100
  - and complement bits to the left:
    
    01101100

Subtraction with Radix Complements

- For n-digit, unsigned numbers M and N, find M – N in base r:
  - Add the r's complement of the subtrahend N to the minuend M:
    
    M + (r^n – N) = M – N + r^n
  - If M ≥ N, the sum produces end carry r^n which is discarded; from above, M – N remains.
  - If M < N, the sum does not produce an end carry and, from above, is equal to r^n – ( N – M ), the r's complement of ( N – M ).
  - To obtain the result – (N – M), take the r's complement of the sum and place a – in front.
Unsigned 10’s Complement
Subtraction Example 1

- Find $543_{10} - 123_{10}$

\[
\begin{array}{c}
543 \\ - 123 \text{ 10's comp} \\
\end{array}
\]

\[
\begin{array}{c}
1543 \\
+ 877 \\
420
\end{array}
\]

- The carry of 1 indicates that no correction of the result is required.

Unsigned 10’s Complement
Subtraction Example 2

- Find $123_{10} - 543_{10}$

\[
\begin{array}{c}
123 \\ - 543 \text{ 10's comp} \\
\end{array}
\]

\[
\begin{array}{c}
0 123 \\
+ 457 \\
580 \text{ 10's comp} \\
\end{array}
\]

- The carry of 0 indicates that a correction of the result is required.
- Result = $- (520)$
Unsigned 2’s Complement Subtraction

Example 1

- Find $01010100_2 - 01000011_2$

\[
\begin{align*}
01010100 & \quad 101010100 \\
- 01000011 & \quad 2's\ comp \\
\hline
00010001 & \quad + 10111101 \\
\hline
11101111 & \quad 2's\ comp \\
00010001 & \quad 00010001
\end{align*}
\]

- The carry of 1 indicates that no correction of the result is required.

Example 2

- Find $01000011_2 - 01010100_2$

\[
\begin{align*}
01000011 & \quad 00100011 \\
- 01010100 & \quad 2's\ comp \\
\hline
11101111 & \quad + 10101100 \\
\hline
00010001 & \quad 2's\ comp \\
00010001 & \quad 00010001
\end{align*}
\]

- The carry of 0 indicates that a correction of the result is required.

- Result = $- (00010001)$
Subtraction with Diminished Radix Complement

- For n-digit, unsigned numbers M and N, find \( M - N \) in base \( r \):
  - Add the \((r - 1)\)'s complement of the subtrahend \( N \) to the minuend \( M \):
    \[ M + (r^n - 1 - N) = M - N + r^n - 1 \]
  - If \( M \geq N \), the result is excess by \( r^n - 1 \). The end carry \( r^n \) when discarded removes \( r^n \), leaving a result short by 1. To fix this shortage, whenever and end carry occurs we add 1 in the LSB position. This is called end-around carry.
  - If \( M < N \), the sum does not produce an end carry and, from above, is equal to \( r^n - 1 - (N - M) \), the \( r - 1 \)'s complement of \( (N - M) \).
  - To obtain the result \( -(N - M) \), take the \( r - 1 \)'s complement of the sum and place a 1 in front.

Unsigned 1’s Complement Subtraction

Example 1

- Find \( 01010100_2 - 01000011_2 \)

\[
\begin{array}{c}
01010100 \\
- 01000011 \\
\text{1’s comp} \\
\hline
10111100 \\
+ \hline
00010000 \\
+1 \\
\hline
00010001
\end{array}
\]

- The end-around carry occurs.
Unsigned 1’s Complement Subtraction

Example 2

- Find $01000011_2 - 01010100_2$

\[
\begin{align*}
01000011 & \quad 01000011 \\
- 01010100 & \quad \text{1’s comp} + 10101011 \\
\hline
11101110 & \quad \text{1’s comp} \\
& \quad 00010001
\end{align*}
\]

- The carry of 0 indicates that a correction of the result is required.
- Result = $- (00010001)$

Signed Integers

- Positive numbers and zero can be represented by unsigned $n$-digit, radix $r$ numbers. We need a representation for negative numbers.
- To represent a sign (+ or −) we need exactly one more bit of information (1 binary digit gives $2^1 = 2$ elements which is exactly what is needed).
- Since computers use binary numbers, by convention, (and, for convenience), the most significant bit is interpreted as a sign bit:

\[ s \ a_{n-2} \ldots a_2a_1a_0 \]

where:
- $s = 0$ for Positive numbers
- $s = 1$ for Negative numbers
- and $a_i = 0$ or 1 represent in some form the magnitude.
Signed Integer Representations

- **Signed-Magnitude** – here the n – 1 digits are interpreted as a positive magnitude.
- **Signed-Complement** – here the digits are interpreted as the rest of the complement of the number. There are two possibilities here:
  - **Signed One's Complement** –
    - Uses 1's Complement Arithmetic
  - **Signed Two's Complement** –
    - Use 2's Complement Arithmetic

Signed Integer Representation Example

- \( r = 2, n = 3 \)

<table>
<thead>
<tr>
<th>Number</th>
<th>Sign-Mag.</th>
<th>1's Comp.</th>
<th>2's Comp.</th>
</tr>
</thead>
<tbody>
<tr>
<td>+3</td>
<td>011</td>
<td>011</td>
<td>011</td>
</tr>
<tr>
<td>+2</td>
<td>010</td>
<td>010</td>
<td>010</td>
</tr>
<tr>
<td>+1</td>
<td>001</td>
<td>001</td>
<td>001</td>
</tr>
<tr>
<td>+0</td>
<td>000</td>
<td>000</td>
<td>000</td>
</tr>
<tr>
<td>–0</td>
<td>100</td>
<td>111</td>
<td>—</td>
</tr>
<tr>
<td>–1</td>
<td>101</td>
<td>110</td>
<td>111</td>
</tr>
<tr>
<td>–2</td>
<td>110</td>
<td>101</td>
<td>110</td>
</tr>
<tr>
<td>–3</td>
<td>111</td>
<td>100</td>
<td>101</td>
</tr>
<tr>
<td>–4</td>
<td>—</td>
<td>—</td>
<td>100</td>
</tr>
</tbody>
</table>
Signed-Magnitude Arithmetic

- **Addition:**
  - If signs are the same:
    1. Add the magnitudes.
    2. Check for overflow (a carry into the sign bit).
    3. The sign of the result is the same.
  - If the signs differ:
    1. Subtract the subtrahend from the minuend
    2. If a borrow occurs, take the two’s complement of result
       and make the sign the complement of the sign of the minuend.
    3. Overflow will never occur.

- **Subtraction:**
  - Complement the sign bit of the number you are subtracting and
    follow the rules for addition.

Sign-Magnitude Examples
Signed-Complement Arithmetic

- **Addition:**
  1. Add the numbers including the sign bits, discarding a carry out of the sign bits (2's Complement), or using an end-around carry (1's Complement).
  2. If the sign bits were the same for both numbers and the sign of the result is different, an overflow has occurred.
  3. The sign of the result is computed in step 1.

- **Subtraction:**
  Form the complement of the number you are subtracting and follow the rules for addition.

Signed 2’s Complement Examples
2’s Complement Adder/Subtractor

- Subtraction can be accomplished by addition of the Two's Complement.
  1. Complement each bit (One's Comp.)
  2. Add one to the result.
- The following circuit computes A - B:
- When the Carry-In is 1, the 2’s comp of B is formed using XORS to form the 1’s comp and adding the 1 on C(0).