Complements

- Subtraction of numbers requires a different algorithm than addition.
- Adding a complement of a number is equivalent to subtraction.
- We will discuss two complements:
  - Diminished Radix Complement
  - Radix Complement
- Subtraction will be accomplished by adding a complement.
**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:

$$(rn - 1) - N$$

**Example:**

For $r = 10$, $N = 1234_{10}$, $n = 4$ (4 digits), we have:

$$(rn - 1) = 10,000 - 1 = 9999_{10}$$

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), we have:

$$(rn - 1) = 256 - 1 = 255_{10} \text{ or } 11111111_2$$

The 1's complement of $01110011_2$ is then:

$$11111111_2 - 01110011_2 = 10001100_2$$

**NOTE:** Since the $2^n - 1$ factor consists of all 1's and since 1 - 0 = 1 and 1 - 1 = 0, forming the one's complement consists of complementing each individual bit.
Radix Complement

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

\[
\begin{align*}
    r^n - N & \quad \text{for } N \neq 0 \quad \text{and} \\
    0 & \quad \text{for } N = 0
\end{align*}
\]

Note that 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\(_{10}\) is then

\[
    10,000_{10} - 1234_{10} = 8766_{10} \text{ or } 8765 + 1 \quad (9\text{'s complement plus 1})
\]

Binary 2's Complement

For \( r = 2, N = 01110011_2, n = 8 \) (8 digits), we have:

\[
    (r^n) = 256_{10} \text{ or } 100000000_2
\]

The 2's complement of 01110011\(_{2}\) is then:

\[
\begin{align*}
    100000000_2 \\
    - 01110011_2 \\
    \hline
    10001101_2
\end{align*}
\]

Note that this is the 1's complement plus 1.
Binary 2's Complement Examples

\[
\begin{align*}
100000000 & - 11011100 \\
\hline
00100100 & \\
\end{align*}
\]
(The 2's complement of 0 is zero!)

\[
\begin{align*}
100000000 & - 00000000 \\
\hline
00000000 & \\
\end{align*}
\]

\[
\begin{align*}
100000000 & - 11111111 \\
\hline
00000001 & \Leftarrow (\text{could this be } -1)\
\end{align*}
\]
\[
\begin{align*}
100000000 & - 11111111 \\
\hline
00000001 & \Leftarrow (\text{could this be } +1)\
\end{align*}
\]

Efficient 2's Complement

Given: an n-bit binary number:
\[
an_{-1}a_{n-2} \ldots a_{i+1}10\ldots0
\]
Where for some digit position i, \( a_i \) is 1 and all digits to the right are 0, form the twos complement value this way:

- Leave \( a_i \) equal to 1 (unchanged),
- Leave rightmost digits 0 (unchanged),
- Complement all other digits to the left of \( a_i \).
  (0 replaces 1, 1 replaces 0)
Two's Complement Example

01101011100011100000
First 1 from right

Complement leftmost digits

10010100011100

Leave these

This ⇒ 0110100111100
becomes 1001011000100

This ⇒ 1000000000000
becomes 1000000000000

Subtraction with Radix Complements

Subtract two n-digit, unsigned numbers M − N, in base r as follows:

1. Add the minuend M to the r's complement of the subtrahend N to perform:
   \[ M + (r^n - N) = M - N + r^n \]

2. If M ≥ N, the sum will produce an end carry, r^n which is discarded; what is left is the result, M − N.

3. If M < N, the sum does not produce an end carry and is equal to r^n - ( N - M ), which is the r's complement of ( N − M ). To obtain the answer in a familiar form, take the r's complement of the sum and place a negative sign in front.
Example: Find $543_{10} - 123_{10}$

1). Form 10's complement of 123:

\[
\begin{array}{c}
1000 \\
- 123 \\
\hline
877
\end{array}
\]

2). Add the two:

\[
\begin{array}{c}
543 \\
(+) 877 \\
\hline
1420
\end{array}
\]

3). Since $M > N$, we discard the carry.

Ans: 420

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

1). Form 10's complement of 543:

\[
\begin{array}{c}
1000 \\
- 543 \\
\hline
457
\end{array}
\]

2). Add the two:

\[
\begin{array}{c}
123 \\
(+) 457 \\
\hline
580 \text{ (no carry)}
\end{array}
\]

3). Since $M < N$, form complement

\[
\begin{array}{c}
580 \\
1000 \\
(-) 580 \\
\hline
420
\end{array}
\]

Answer is $(-)420$. 
Binary Example

Compute: 1010100 - 1000011
1). Form 2's complement of 1000011: 1010100
   0111101
2). Add the two:
   1010100
   0111101
   (+) ------------
   1 0010001  (a carry)
3). Since M \geq N, discard the carry.   Ans. = 0010001

Another Binary Example

Compute: 1000011 - 1010100
1). Form 2's complement of 1010100: 1000011
   0101100
2). Add the two:
   1010100
   0101100
   (+) ------------
   1101111  (no carry)
3). Since M < N, complement the result.
   Ans. = (−) 0010001
Subtract: Add 1’s Complement

We can use addition of the 1's complement to subtract two numbers with a minor modification.

Since (r-1)'s complement is one less than the r's complement, the result produces a sum which is one less than the correct sum when an end carry occurs.

We can simply add in the end carry when it occurs to correct the answer.

If the end carry does not occur, the result is negative and we can use the 1's complement to represent the negative result.

1’s Complement Subtraction

Use 1's complement to compute 1010100 - 1000011

1). Form 1's complement of 1000011:
   1000011
   0111100

2). Add the two:
   1010100
   0111100
   (+) 10010000 (a carry)

3). Add carry, end around
   0000001
   (+) 0010001
   Ans. = 0010001
1’s Complement Subtraction

Use 1’s complement for computing 1000011 - 1010100

1). Form 1’s complement of 1010100:
   \[ \begin{array}{c}
   1010100 \\
   \hline
   0101011
   \end{array} \]

2). Add the two:
   \[ \begin{array}{c}
   1000011 \\
   0101011 \\
   \hline
   \text{(no carry)}
   \end{array} \]

3). Form 1’s complement:
   \[ 0010001 \]

4). The answer has a negative sign.

\[ \text{Ans.} = (\neg) 0010001 \]

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 most computers use binary numbers, by convention, (and for convenience), the most significant bit is interpreted as a sign bit as shown below:

\[ \begin{array}{c}
\text{sa}_{n-2} \cdots \text{a}_2 \text{a}_1 \text{a}_0 \\
\end{array} \]

Where:

- \( s = 0 \) for Positive numbers
- \( s = 1 \) for Negative numbers

and

- \( a_j \) are 0 or 1
Interpreting the Other Digits

Given \( n \) binary digits, the digit with weight \( 2^{(n-1)} \) is the sign and the digits with weights \( 2^{(n-2)} \) down to \( 2^{(0)} \) can be used to represent \( 2^{(n-1)} \) distinct elements.

There are several ways to interpret the other digits. Here are three popular choices:

1. Signed-Magnitude -- here the \( n-1 \) digits are interpreted as a positive magnitude.

2. Signed-Complement -- here the digits are interpreted as the rest of the complement of the number. There are two possibilities here:
   - 2a. Signed One's Complement --
     (use the 1's Complement to compute)
   - 2b. Signed Two's Complement --
     (use the 2's Complement to compute)

Example: Given \( r=2, n=3 \)

We have the following interpretations for signed integer representation:

<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>
Addition with Signed Numbers

Caution: If you use all \( r^n \) possible combinations of \( n \) radix \( r \) digits, some operations on elements of the set will produce elements which will not be represented in the set.

Example: Add unsigned, 3-bit integers 101\(_2\) to 100\(_2\) to get 1001\(_2\) \((5 + 4 = 9)\). This result cannot be represented in the set of 3-bit unsigned integers. An overflow is said to have occurred.

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 magnitude of the smaller from the magnitude of the larger.
2. Use the sign of the larger magnitude for the sign of the result.
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

Same signs

000 + 001 = 001  (signs are the same)
010 + 010 = x00  (Overflow into sign bit)
101 + 101 = 110 (signs are the same)
110 + 110 = x00 (Overflow into sign bit)

Different signs

001 + 110 = 101 (010 −−−− 001, take −−−− sign)
111 + 010 = 101 (011 −−−− 001, take −−−− sign)
101 + 010 = 001 (010 −−−− 001, take + sign)
100 + 000 = ?00 (is it + or - zero?)

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.
Binary Adder/Subtractor

Subtraction can be accomplished by addition of the Two's Complement. Thus we:

1. Complement each bit (One's Comp.)
2. Add one to the result.

The following circuit computes $A - B$:

It computes $A - B$ when the Carry-In is "$1" by forming the One's Complement of $B$ using EXOR gates and adding one.