Using binary complements

This document is a formalization of the arguments presented in Sections 6.1 and 6.2 of the textbook. We assume the binary system here.

1 Complement operations

The following complements of a binary string

\[ x = x_{m-1}x_{m-2} \ldots x_1x_0 \]  

are considered throughout the text:

\[ x^c = (2^m - x) \mod 2^m \]  
\[ \hat{x}^c = 2^m - 1 - x. \]  

The first one is called *radix* complement and the second one is *diminished* complement. For a binary string \( x \) denote by \( \text{value}(x) \) the numeric value of the corresponding number represented in the radix complement form. Therefore, we have \( x^c = (\hat{x}^c + 1) \mod 2^m \) and

\[
\text{value}(x) = \begin{cases} 
\sum_{i=0}^{m-1} x_i2^i, & \text{if } x_{m-1} = 0 \\
-\left(2^m - \sum_{i=0}^{m-1} x_i2^i\right), & \text{if } x_{m-1} = 1 
\end{cases}
\]

\[-2^{m-1} \leq \text{value}(x) \leq 2^{m-1} - 1 \]

2 Scaling complement numbers

2.1 The left shift operation

The arithmetic left shift of a binary string \( x \) in the form (1) results in the string \( \text{shl}(x) = y = x_{m-2}x_{m-3} \ldots x_00 \). Avoiding trivial cases we assume \( m \geq 3 \).

**Theorem 1** If \(-2^{m-2} \leq \text{value}(x) < 2^{m-2}\) then the arithmetic left shift of \( x \) causes no overflow and \( \text{value}(y) = 2 \cdot \text{value}(x) \).
Proof:

Case a. Assume \(0 \leq \text{value}(x) < 2^{m-2}\). This implies \(x_{m-1} = x_{m-2} = 0\), so \(x = 00x_{m-3} \ldots x_1x_0\). For the string \(y\) one has \(y = 0x_{m-3} \ldots x_00\). We get:

\[
\text{value}(y) = \sum_{i=0}^{m-2} x_i2^{i+1}
\]

\[
= \sum_{i=0}^{m-2} x_i \cdot 2 \cdot 2^i
\]

\[
= 2 \cdot \sum_{i=0}^{m-2} x_i 2^i
\]

\[
= 2 \cdot \sum_{i=0}^{m-1} x_i 2^i \quad \text{(since } x_{m-1} = 0) \]

\[
= 2 \cdot \text{value}(x).
\]

It follows from the form of \(y\) that \(\text{value}(y)\) is non-negative and even, so \(0 \leq \text{value}(y) \leq 2^{m-1} - 2\). Hence, \(\text{value}(y)\) is in the range of positive numbers represented in the radix complement form and no overflow occurs.

Case b. Assume \(-2^{m-2} \leq \text{value}(x) < 0\). This implies \(x_{m-1} = x_{m-2} = 1\), so \(x = 11x_{m-3} \ldots x_1x_0\) and

\[
\text{value}(x) = -\left(2^m - \sum_{i=0}^{m-1} x_i 2^i\right)
\]

\[
= -\left(2^m - 2^{m-1} - 2^{m-2} - \sum_{i=0}^{m-3} x_i 2^i\right)
\]

\[
= -\left(2^{m-2} - \sum_{i=0}^{m-3} x_i 2^i\right). \quad \text{(6)}
\]

On the other hand, \(y = 1x_{m-3} \ldots x_00\). Therefore \(y < 0\), so according to (2),

\[
\text{value}(y) = -\left(2^m - \sum_{i=0}^{m-2} x_i 2^{i+1}\right)
\]

\[
= -\left(2^m - 2 \cdot \sum_{i=0}^{m-2} x_i 2^i\right)
\]

\[
= -\left(2^m - 2^{m-1} - 2 \cdot \sum_{i=0}^{m-3} x_i 2^i\right) \quad \text{(since } x_{m-2} = 1) \]

\[
= -\left(2^{m-1} - 2 \cdot \sum_{i=0}^{m-3} x_i 2^i\right)
\]

\[
= -2 \cdot \left(2^{m-2} - \sum_{i=0}^{m-3} x_i 2^i\right)
\]

\[
= 2 \cdot \text{value}(x) \quad \text{(see (6))}.
\]

For the considered range of \(\text{value}(x)\) one has \(-2^{m-1} \leq \text{value}(y) < 0\). Hence, \(\text{value}(y)\) is in the range of negative numbers and no overflow occurs. \(\square\)
Example 1 Assume \( x = 001101 \) then \( y = \text{shl}(x) = 011010 \). One has \( \text{value}(x) = 13 \) and \( \text{value}(y) = 26 \), so \( \text{value}(y) = 2 \cdot \text{value}(x) \).

Example 2 Assume \( x = 110101 \) then \( y = \text{shl}(x) = 101010 \). One has \( \text{value}(x) = -11 \) and \( \text{value}(y) = -22 \), so \( \text{value}(y) = 2 \cdot \text{value}(x) \).

The left shift is considered to generate an overflow if the signs of the original and shifted values are different. If \( \text{value}(x) > 0 \) and \( \text{value}(\text{shl}(x)) < 0 \) then \( x_{m-1} = 0 \) and \( x_{m-1} = 1 \). This implies \( 2^{m-2} \leq \text{value}(x) \leq 2^{m-1} - 1 \). On the other hand, if \( \text{value}(x) < 0 \) and \( \text{value}(\text{shl}(x)) > 0 \) then \( x_{m-1} = 1 \) and \( x_{m-1} = 0 \). This implies \( -2^{m-1} \leq \text{value}(x) < 2^{m-2} \). Therefore, the range of \( x \) in Theorem 1 is a necessary and sufficient condition for the absence of an overflow.

Note that Theorem 1 is not generally true for the diminished complement. For example if \( x = 110101 \) and \( y = \text{shl}(x) = 101010 \) one has \( \text{value}(x) = -10 \) and \( \text{value}(y) = -21 \), so \( \text{value}(y) \neq 2 \cdot \text{value}(x) \).

2.2 The right shift operation

The arithmetic right shift of a binary string \( x \) in the form (1) results in the string \( \text{shr}(x) = y = x_{m-1}x_{m-2} \ldots x_1x_0 \). Avoiding trivial cases we assume \( m \geq 3 \).

Theorem 2 If \( -2^{m-1} \leq \text{value}(x) \leq 2^{m-1} - 1 \) then the arithmetic right shift of \( x \) causes no overflow and \( \text{value}(y) = \lfloor \text{value}(x)/2 \rfloor \).

Proof:
Case a. Assume \( 0 \leq x \leq 2^{m-1} - 1 \). Then \( x_{m-1} = 0 \), so \( x = 0 x_{m-2} \ldots x_1 x_0 \) and \( y = 00 x_{m-2} \ldots x_1 \). One has

\[
\text{value}(y) = \sum_{i=1}^{m-1} x_i 2^{i-1} = \frac{1}{2} \cdot \sum_{i=1}^{m-1} x_i 2^i = \frac{1}{2} \cdot \left( \sum_{i=0}^{m-1} x_i 2^i - x_0 \right) = \frac{1}{2} \cdot \left( \text{value}(x) - x_0 \right)
\] (7)

If \( x_0 = 0 \) then \( x \) is even, so \( x = 2k \) for some integer \( k \). Then (7) implies \( \text{value}(y) = \frac{1}{2} \cdot 2k = k = \lfloor \text{value}(x)/2 \rfloor \).

If \( x_0 = 1 \) then \( x \) is odd, so \( x = 2k + 1 \) for some integer \( k \). Then (7) implies \( \text{value}(y) = \frac{1}{2} \cdot (2k + 1 - 1) = k = \lfloor \text{value}(x)/2 \rfloor \).
Case b. Assume $-2^{m-1} \leq x < 0$. Then $x_{m-1} = 1$, so $x = 1x_{m-2}\ldots x_1x_0$ and $y = 11x_{m-2}\ldots x_1$. One has

$$
\text{value}(y) = -\left(2^m - \left(\sum_{i=1}^{m-1} x_i2^{i-1} + 2^{m-1}\right)\right) \\
= -\left(2^{m-1} - \frac{1}{2} \cdot \sum_{i=1}^{m-1} x_i2^i\right) \\
= -\frac{1}{2} \cdot \left(2^m - \sum_{i=0}^{m-1} x_i2^i + x_0\right) \\
= \frac{1}{2} \cdot (\text{value}(x) - x_0). \tag{8}
$$

If $x_0 = 0$ then $x$ is even, so $x = -2k$ for some integer $k$. Then (8) implies $\text{value}(y) = \frac{1}{2} \cdot -2k = -k = \lfloor \text{value}(x)/2 \rfloor$.

If $x_0 = 1$ then $x$ is odd, so $x = -(2k + 1)$ for some integer $k$. Then (8) implies $\text{value}(y) = \frac{1}{2} \cdot (-2k - 1 - 1) = -(k + 1) = \lfloor \text{value}(x)/2 \rfloor$.

Since the signs of $\text{value}(x)$ and $\text{value}(\text{shl}(x))$ are the same, no overflow occurs. \qed

Example 3 Assume $x = 001101$ then $y = \text{shr}(x) = 000110$. One has $\text{value}(x) = 13$ and $\text{value}(y) = 6$, so $\text{value}(y) = \lfloor \text{value}(x)/2 \rfloor$.

Example 4 Assume $x = 110101$ then $y = \text{shr}(x) = 111010$. One has $\text{value}(x) = -11$ and $\text{value}(y) = -6$, so $\text{value}(y) = \lfloor \text{value}(x)/2 \rfloor$.

Note that Theorem 2 is not generally true for the diminished complement. For example if $x = 110101$ and $y = \text{shr}(x) = 111010$ one has $\text{value}(x) = -11$ and $\text{value}(y) = -5$. Since $\lfloor -11/2 \rfloor = -6$, $\text{value}(y) \neq \lfloor \text{value}(x)/2 \rfloor$.

3 Addition and subtraction operations

Since $a - b = a + (-b)$, it is sufficient to analyse the addition only. Let $x$ and $y$ be numbers to sum up and let them be represented by binary strings $\text{rep}(x) = x_{m-1}x_{m-2}\ldots x_0$ and $\text{rep}(y) = y_{m-1}y_{m-2}\ldots y_0$ in the radix complement system. Denote by $\text{rep}(x) \oplus \text{rep}(y)$ the unsigned sum of these strings.

Theorem 3 Let $x$ and $y$ be numbers such that $-2^{m-1} \leq x \leq 2^{m-1} - 1$ and $-2^{m-1} \leq y \leq 2^{m-1} - 1$. Then $\text{rep}(x) \oplus \text{rep}(y) = \text{rep}(x + y)$ if no overflow occurs.

Proof: We consider the following three cases.
Case a. Assume $x$ and $y$ have different signs. Without loss of generality assume $x \geq 0$ and $y < 0$. In this case
\[
\text{rep}(x) = 0x_{m-2} \cdots x_1x_0 = x \\
\text{rep}(y) = 1y_{m-2} \cdots y_1y_0 = 2^m - |y|
\]
Denote $s = \text{rep}(x) \oplus \text{rep}(y) = s_{m-1}s_{m-2} \ldots s_0$. One has
\[
s = (x + (2^m - |y|)) \mod 2^m = (2^m + (x - |y|)) \mod 2^m
\]
If $x - |y| \geq 0$ then $s_{m-1} = 0$, so
\[
\text{value}(s) = (2^m + (x - |y|)) \mod 2^m = (x - |y|) \mod 2^m = x - |y|
\]
since $0 \leq x - |y| < 2^m - 1$, so no overflow occurs.
If $x - |y| < 0$ then $s_{m-1} = 1$, so
\[
\text{value}(s) = -(2^m - |x - |y||).
\]
Hence, $s$ represents negative number $-|x - |y|| = x - |y|$. Again, there is no overflow.

Case b. Assume $x \geq 0$ and $y \geq 0$. Then
\[
\text{rep}(x) = 0x_{m-2} \cdots x_1x_0 = x \\
\text{rep}(y) = 0y_{m-2} \cdots y_1y_0 = y
\]
Then $s$ represents the number $x + y$ provided $s_{m-1} = 0$, i.e. $x + y \leq 2^{m-1} - 1$.

Case c. Assume $x < 0$ and $y < 0$. Then
\[
\text{rep}(x) = 1x_{m-2} \cdots x_1x_0 = 2^m - |x| \\
\text{rep}(y) = 1y_{m-2} \cdots y_1y_0 = 2^m - |y|
\]
Taking into account that $|x| + |y| = |x + y|$ whenever $x$ and $y$ have the same signs, we get
\[
\text{value}(s) = -((2^m - |x|) + (2^m - |y|)) \mod 2^m = -((2^m + (2^m - |x| - |y|)) \mod 2^m) = -((2^m - |x + y|) \mod 2^m)
\]
Hence, $s$ represents the number $x + y$ provided $s_{m-1} = 1$, i.e. $|x + y| \leq 2^{m-1}$.

Corollary 1 Let $c_{\text{in}}$ and $c_{\text{out}}$ be the carry-in and carry-out of the sign bit. Then no overflow occurs if and only if $c_{\text{in}} \oplus c_{\text{out}} = 0$.  

\[\square\]