A Calculus

for the

Correctness of Digitwize Arithmetic

Samuel Balco and Roy L. Crole

Informatics, University of Leicester, UK

The Problem

- Columnar/digitwize addition of natural numbers is always correct:

$$\begin{array}{rrr} & \mathbf{5} & \mathbf{8} \\ + & \mathbf{6} & \mathbf{7} \\ \hline \mathbf{1} & \mathbf{1} & \mathbf{0} \\ \hline \mathbf{1} & \mathbf{2} & \mathbf{5} \\ \hline \end{array}$$

- Addition of natural numbers by an Arithmetic Logic Unit (ALU) may or may not be correct.

In texts, one finds conditions for ALU addition correctness. However, details are often informal and incomplete.

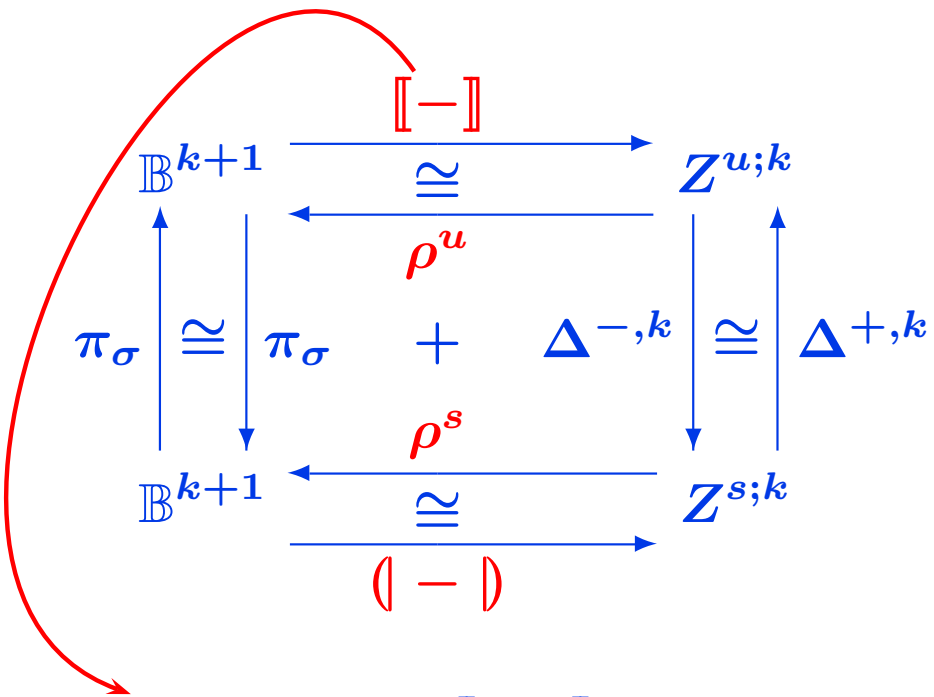
As a teacher of Computer Architecture I always found this frustrating.

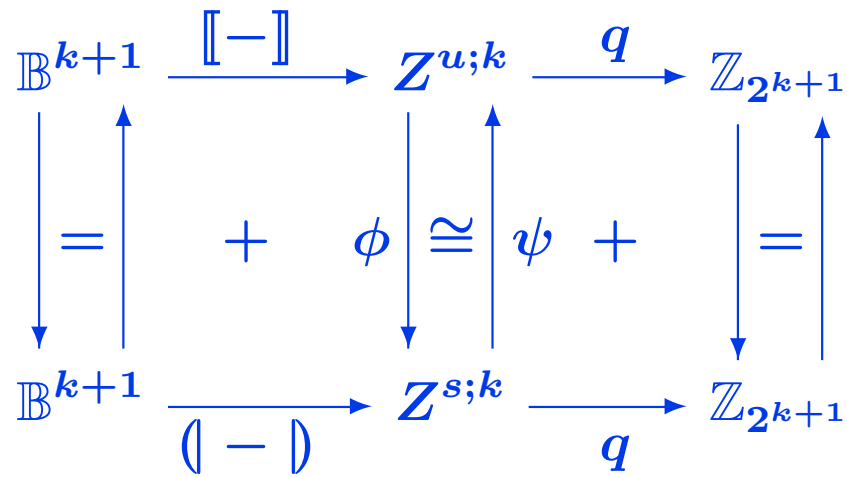
The Response

- We explore digitwize operators for addition and subtraction.
- These split into two groups (defined later on):
  - ALU-operators, and
  - Maths-operators.
- For each group, and each operator within the group, a correctness theorem is proven.

Semantic Equations (Division Algorithm on $\mathbb{Z}$)

- $d_k d_{k-1} \ldots d_1 d_0 \in \mathbb{B}^{k+1}$
- $Z^{u;k} \stackrel{\text{def}}{=} \{\, z \in \mathbb{Z} \mid 0 \leq z \leq 2^{k+1} - 1 \,\}$
- $Z^{s;k} \stackrel{\text{def}}{=} \{\, z \in \mathbb{Z} \mid -2^k \leq z \leq 2^k - 1 \,\}$

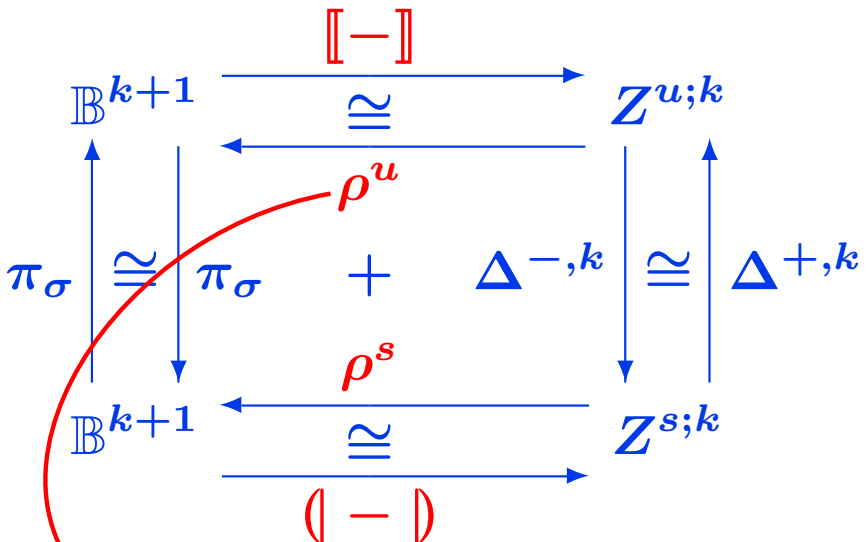


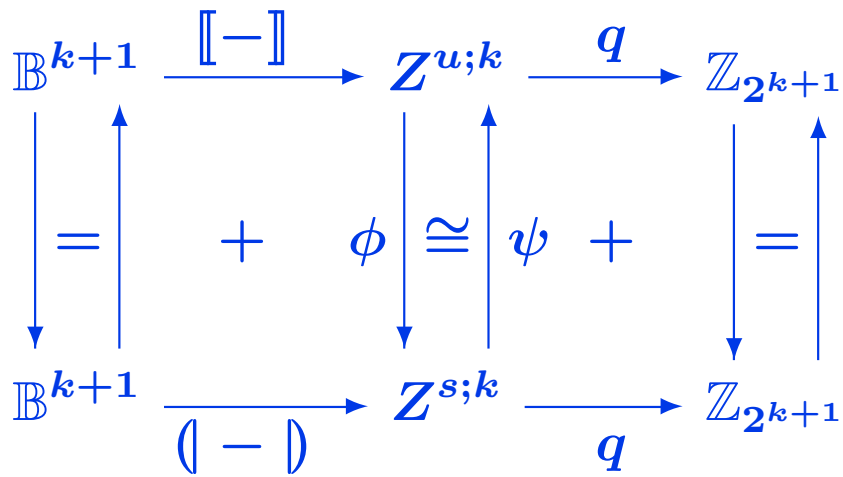


unsigned semantics $[\![101]\!] = 5$

Semantic Equations (Division Algorithm on $\mathbb{Z}$)

- $d_k d_{k-1} \ldots d_1 d_0 \in \mathbb{B}^{k+1}$
- $Z^{u;k} \stackrel{\text{def}}{=} \{\, z \in \mathbb{Z} \mid 0 \leq z \leq 2^{k+1} - 1 \,\}$
- $Z^{s;k} \stackrel{\text{def}}{=} \{\, z \in \mathbb{Z} \mid -2^k \leq z \leq 2^k - 1 \,\}$

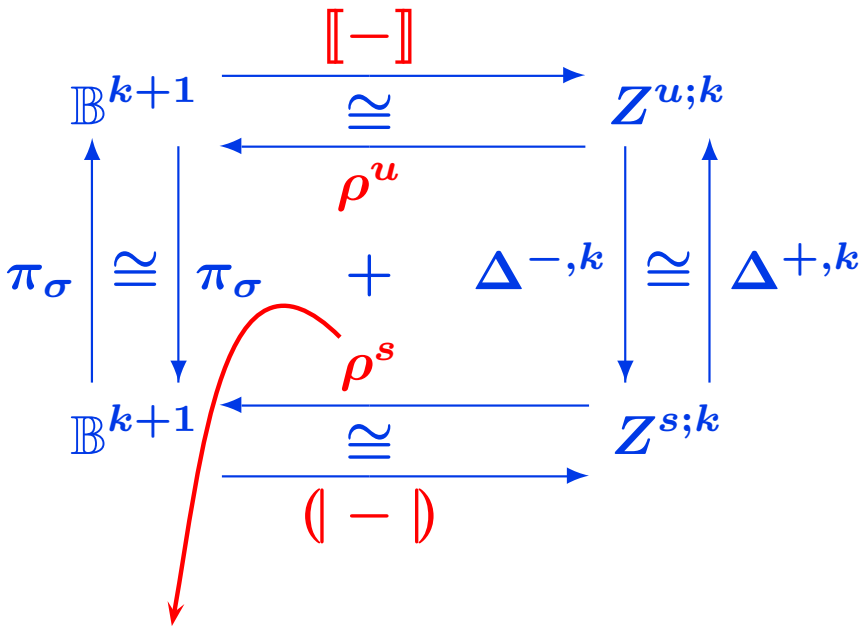


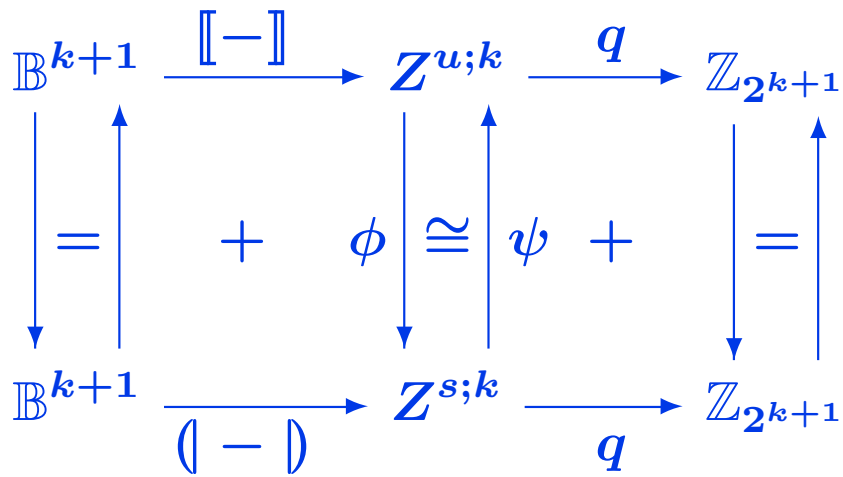


unsigned representation $\rho^u(5) = 101$

Semantic Equations (Division Algorithm on $\mathbb{Z}$)

- $d_k d_{k-1} \ldots d_1 d_0 \in \mathbb{B}^{k+1}$
- $Z^{u;k} \stackrel{\text{def}}{=} \{\, z \in \mathbb{Z} \mid 0 \leq z \leq 2^{k+1} - 1 \,\}$
- $Z^{s;k} \stackrel{\text{def}}{=} \{\, z \in \mathbb{Z} \mid -2^k \leq z \leq 2^k - 1 \,\}$

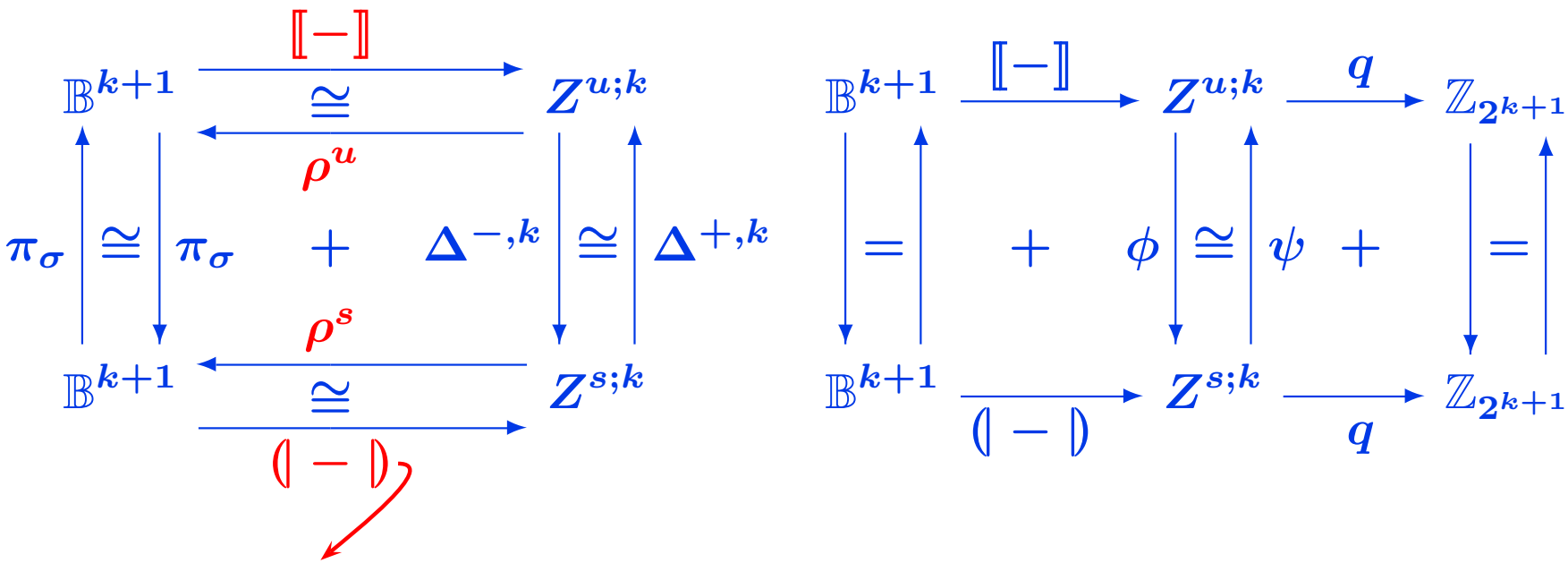




signed representation $\rho^s(-3) = 101$

Semantic Equations (Division Algorithm on $\mathbb{Z}$)

- $d_k d_{k-1} \ldots d_1 d_0 \in \mathbb{B}^{k+1}$
- $Z^{u;k} \stackrel{\text{def}}{=} \{\, z \in \mathbb{Z} \mid 0 \leq z \leq 2^{k+1} - 1 \,\}$
- $Z^{s;k} \stackrel{\text{def}}{=} \{\, z \in \mathbb{Z} \mid -2^k \leq z \leq 2^k - 1 \,\}$



signed semantics (2s-complement) $(\!|101|\!) = -3$

An ALU-Operator with 3-Digit Unsigned Binary

- Consider $1 + 3 =_{\mathbb{N}} 4$ and $3 + 5 =_{\mathbb{N}} 8$.
- 1, 3, 5 appear below as binary operands where $+_{ALU}$ is an ALU-operator.

$$\begin{array}{rrrr} & 0 & 0 & 1 \\ +_{ALU} & 0 & 1 & 1 \\ \hline {}^{c}0 & 1 & 1 & 0 \\ \hline \rightarrow & 1 & 0 & 0 \end{array} \qquad \begin{array}{rrrr} & 0 & 1 & 1 \\ +_{ALU} & 1 & 0 & 1 \\ \hline {}^{c}1 & 1 & 1 & 0 \\ \hline \rightarrow & 0 & 0 & 0 \end{array}$$

- The ALU outputs are 4, correct, and 0, incorrect.
- ALU Outputs are correct just in case
  - $n + m = [\![\rho^u(m) +_{ALU} \rho^u(n)]\!]$ **Correctness**
  - ${}^{c}\delta = 0$ **Overflow Digit**
  - $0 \leq \boxed{n + m} \leq 2^{k+1} - 1 \equiv 7$ **Range**

An ALU-Operator with 3-Digit Unsigned Binary

- Consider $1 + 3 =_{\mathbb{N}} 4$ and $3 + 5 =_{\mathbb{N}} 8$.
- 1, 3, 5 appear below as binary operands where $+_{ALU}$ is an ALU-operator.

$$\begin{array}{rrrrr} & & 0 & 0 & 1 \\ +_{ALU} & & 0 & 1 & 1 \\ \hline & {}^{c}0 & 1 & 1 & 0 \\ \hline & \rightarrow & 1 & 0 & 0 \end{array} \qquad \begin{array}{rrrrr} & & 0 & 1 & 1 \\ +_{ALU} & & 1 & 0 & 1 \\ \hline & {}^{c}1 & 1 & 1 & 0 \\ \hline & \rightarrow & 0 & 0 & 0 \end{array}$$

- The ALU outputs are 4, correct, and 0, incorrect.
- ALU Outputs are correct just in case
  - $n + m = [\![\rho^u(m) +_{ALU} \rho^u(n)]\!]$ **Correctness**
  - ${}^{c}\delta = 0$ **Overflow Digit**
  - $0 \leq \boxed{n + m} \leq 2^{k+1} - 1 \equiv 7$ **Range**

An ALU-Operator with $\mathbf{3}$-Digit Signed Binary

- Let's use the same binary operands: $\mathbf{001}$, $\mathbf{011}$, $\mathbf{101}$.
- The integer operands are now $\mathbf{1}$, $\mathbf{3}$, $\mathbf{-3}$.
- Consider $\mathbf{1 + 3 =_{\mathbb{Z}} 4}$. The ALU output is
  - $\mathbf{100}$ which is $\mathbf{-4}$, incorrect.
- Consider $\mathbf{3 + (-3) =_{\mathbb{Z}} 0}$. The ALU output is
  - $\mathbf{000}$ which is $\mathbf{0}$, correct.
- Outputs are correct just in case
  - the overflow digit, and the carry digit to its right, are unequal
  - sign-bit (sign-digit) conditions: the ALU is correct just in case the first digit of each operand is unequal, **OR**, the first digit of each operand and the first digit of the output are equal.
- see Computer Architecture textbooks.

$$\pi_\sigma\ \mathbf{101} = \mathbf{001} \qquad sx(\mathbf{101}) = \mathbf{1101} \qquad zx(\mathbf{101}) = \mathbf{0101}$$

An ALU-Operator with $\mathbf{3}$-Digit Signed Binary

- Let's use the same binary operands: $\mathbf{001}$, $\mathbf{011}$, $\mathbf{101}$.
- The integer operands are now $\mathbf{1}$, $\mathbf{3}$, $\mathbf{-3}$.
- Consider $\mathbf{1 + 3 =_{\mathbb{Z}} 4}$. The ALU output is
  - $\mathbf{100}$ which is $\mathbf{-4}$, incorrect.
- Consider $\mathbf{3 + (-3) =_{\mathbb{Z}} 0}$. The ALU output is
  - $\mathbf{000}$ which is $\mathbf{0}$, correct.
- Outputs are correct just in case
  - the overflow digit, and the carry digit to its right, are unequal
  - sign-bit (sign-digit) conditions: the ALU is correct just in case the first digit of each operand is unequal, **OR**, the first digit of each operand and the first digit of the output are equal.
- see Computer Architecture textbooks.

$$\pi_\sigma\ \mathbf{101} = \mathbf{001} \qquad \boldsymbol{sx}(\mathbf{101}) = \mathbf{1101} \qquad \boldsymbol{zx}(\mathbf{101}) = \mathbf{0101}$$

An ALU-Operator with $\mathbf{3}$-Digit Signed Binary

- Let's use the same binary operands: $\mathbf{001}$, $\mathbf{011}$, $\mathbf{101}$.
- The integer operands are now $\mathbf{1}$, $\mathbf{3}$, $\mathbf{-3}$.
- Consider $\mathbf{1 + 3 =_{\mathbb{Z}} 4}$. The ALU output is
  - $\mathbf{100}$ which is $\mathbf{-4}$, incorrect.
- Consider $\mathbf{3 + (-3) =_{\mathbb{Z}} 0}$. The ALU output is
  - $\mathbf{000}$ which is $\mathbf{0}$, correct.
- Outputs are correct just in case
  - the overflow digit, and the carry digit to its right, are unequal
  - sign-bit (sign-digit) conditions: the ALU is correct just in case the first digit of each operand is unequal, **OR**, the first digit of each operand and the first digit of the output are equal.
- see Computer Architecture textbooks.

$$\pi_\sigma\ \mathbf{101} = \mathbf{001} \qquad sx(\mathbf{101}) = \mathbf{1101} \qquad zx(\mathbf{101}) = \mathbf{0101}$$

An ALU-Operator with $\mathbf{3}$-Digit mixed Signed/Unsigned Binary

- What about $(+\mathbf{3}) - (+\mathbf{5}) =_{\mathbb{Z}} -\mathbf{2}$ where we regard this as a subtraction of unsigned binary numbers?
- If the ALU output is unsigned: overflow!
- The ALU subtraction (add $\overline{\mathbf{101}}$ and add $\mathbf{1}$) is

$$\begin{array}{rcccc} & & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ -_{\boldsymbol{ALU}} & & \mathbf{1} & \mathbf{0} & \mathbf{1} \\ \hline & \boldsymbol{c} & & & \mathbf{1} \\ \hline \end{array} \quad \mapsto \quad \begin{array}{rcccc} & & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ +_{\boldsymbol{ALU}} & & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \hline & \boldsymbol{c}\mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ \hline & & \mathbf{1} & \mathbf{1} & \mathbf{0} \end{array}$$

Then the output $\mathbf{110}$ *is* correct if it is understood as a signed number!

An ALU-Operator with $\mathbf{3}$-Digit mixed Signed/Unsigned Binary

- What about $(+\mathbf{3}) - (+\mathbf{5}) =_{\mathbb{Z}} -\mathbf{2}$ where we regard this as a subtraction of unsigned binary numbers?
- If the ALU output is unsigned: overflow!
- The ALU subtraction (add $\overline{\mathbf{101}}$ and add $\mathbf{1}$) is

$$\begin{array}{rcccc} & & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ -_{ALU} & & \mathbf{1} & \mathbf{0} & \mathbf{1} \\ \hline & c & & & \mathbf{1} \\ \hline \end{array} \quad \mapsto \quad \begin{array}{rcccc} & & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ +_{ALU} & & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \hline & {}^{c}\mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ \hline & & \mathbf{1} & \mathbf{1} & \mathbf{0} \end{array}$$

Then the output $\mathbf{110}$ *is* correct if it is understood as a signed number!

An ALU-Operator with $\mathbf{3}$-Digit mixed Signed/Unsigned Binary

- What about $(+\mathbf{3}) - (+\mathbf{5}) =_{\mathbb{Z}} -\mathbf{2}$ where we regard this as a subtraction of unsigned binary numbers?
- If the ALU output is unsigned: overflow!
- The ALU subtraction (add $\overline{\mathbf{101}}$ and add $\mathbf{1}$) is

$$\begin{array}{rcccc} & & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ -_{\boldsymbol{ALU}} & & \mathbf{1} & \mathbf{0} & \mathbf{1} \\ \hline & \boldsymbol{c} & & & \mathbf{1} \\ \hline \end{array} \quad \mapsto \quad \begin{array}{rcccc} & & \mathbf{0} & \mathbf{1} & \mathbf{1} \\ +_{\boldsymbol{ALU}} & & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \hline & \boldsymbol{c}\mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ \hline & & \mathbf{1} & \mathbf{1} & \mathbf{0} \end{array}$$

Then the output $\mathbf{110}$ *is* correct if it is understood as a signed number!

A Maths-Operator with 3-Digit Binary: Examples Revisited

$$\begin{array}{rcccc} & & 0 & 0 & 1 \\ +_M & & 0 & 1 & 1 \\ \hline & {}^{c}0 & 1 & 1 & 0 \\ \hline & 0 & 1 & 0 & 0 \end{array} \qquad \begin{array}{rcccc} & & 0 & 1 & 1 \\ +_M & & 1 & 0 & 1 \\ \hline & {}^{c}1 & 1 & 1 & 0 \\ \hline & 1 & 0 & 0 & 0 \end{array}$$

- Maths-Operator outputs include the overflow digit.
- Unsigned binary: $1 + 3 = 4$ and $3 + 5 = 8$ are now correct.
- However, with signed binary, in the second example of $3 + (-3) = 0$, the output 1000 is wrong being -8.
- But note the correct integer answer is in range

$$-2^{k+1} \equiv -8 \leq 0 \leq 2^{k+1} - 1 \equiv 7$$

- For $(+3) - (+5)$, regarded as subtraction of unsigned binary numbers, the signed binary ALU output 110 is correct. However signed $011 -_M 101 = 0110$ is incorrect.

A Maths-Operator with 3-Digit Binary: Examples Revisited

$$\begin{array}{rcccc} & & 0 & 0 & 1 \\ +_M & & 0 & 1 & 1 \\ \hline & {}^{c}0 & 1 & 1 & 0 \\ \hline & 0 & 1 & 0 & 0 \end{array} \qquad \begin{array}{rcccc} & & 0 & 1 & 1 \\ +_M & & 1 & 0 & 1 \\ \hline & {}^{c}1 & 1 & 1 & 0 \\ \hline & 1 & 0 & 0 & 0 \end{array}$$

- Maths-Operator outputs include the overflow digit.
- Unsigned binary: $1 + 3 = 4$ and $3 + 5 = 8$ are now correct.
- However, with signed binary, in the second example of $3 + (-3) = 0$, the output 1000 is wrong being -8.
- But note the correct integer answer is in range

$$-2^{k+1} \equiv -8 \leq 0 \leq 2^{k+1} - 1 \equiv 7$$

- For $(+3) - (+5)$, regarded as subtraction of unsigned binary numbers, the signed binary ALU output 110 is correct. However signed $011 -_M 101 = 0110$ is incorrect.

A Maths-Operator with 3-Digit Binary: Examples Revisited

$$\begin{array}{rrrrr} & & 0 & 0 & 1 \\ +_M & & 0 & 1 & 1 \\ \hline & {}^{c}0 & 1 & 1 & 0 \\ \hline & 0 & 1 & 0 & 0 \end{array} \qquad \begin{array}{rrrrr} & & 0 & 1 & 1 \\ +_M & & 1 & 0 & 1 \\ \hline & {}^{c}1 & 1 & 1 & 0 \\ \hline & 1 & 0 & 0 & 0 \end{array}$$

- Maths-Operator outputs include the overflow digit.
- Unsigned binary: $1 + 3 = 4$ and $3 + 5 = 8$ are now correct.
- However, with signed binary, in the second example of $3 + (-3) = 0$, the output 1000 is wrong being -8.
- But note the correct integer answer is in range

$$-2^{k+1} \equiv -8 \leq 0 \leq 2^{k+1} - 1 \equiv 7$$

- For $(+3) - (+5)$, regarded as subtraction of unsigned binary numbers, the signed binary ALU output 110 is correct. However signed $011 -_M 101 = 0110$ is incorrect.

Formalizing k-Digit Correctness

- Suppose that
  - m and n are integers, with *canonical* (k-digit) binary representations $\rho(m)$ and $\rho(n)$.
  - Suppose that op is a 2-argument operator on $\mathbb{Z}$ and $\square$ is a 2-argument digitwize operator on binary numbers.
  - Suppose that $m \text{ op } n =_{\mathbb{Z}} r$.
- Then the operator $\square$ is *correct* if

$$\rho(r) = \rho(m) \square \rho(n)$$

or equivalently, for $\rho^{-1} : \xi \in \mathbb{B}^k \mapsto \{\xi\} \in \mathbb{Z}$,

$$\{a\} \text{ op } \{b\} = \{a \square b\} \in \mathbb{Z}$$

for all k-digit binary numbers a and b.

Formalizing k-Digit Correctness

- Suppose that
  - m and n are integers, with *canonical* (k-digit) binary representations $\rho(m)$ and $\rho(n)$.
  - Suppose that op is a 2-argument operator on $\mathbb{Z}$ and $\square$ is a 2-argument digitwize operator on binary numbers.
  - Suppose that m op $n =_{\mathbb{Z}} r$.
- Then the operator $\square$ is *correct* if

$$\rho(r) = \rho(m) \square \rho(n)$$

or equivalently, for $\rho^{-1} : \xi \in \mathbb{B}^k \mapsto \{\xi\} \in \mathbb{Z}$,

$$\{a\} \text{ op } \{b\} = \{a \square b\} \in \mathbb{Z}$$

for all k-digit binary numbers a and b.

- Let $\rho(\xi) := \rho^u(\xi) \mid \rho^s(\xi)$ and $\{\xi\} := [\![\xi]\!] \mid (\!|\xi|\!)$.

The Additive Digitwize/Columnar Algorithm:

$$(1110, 000) \stackrel{\text{def}}{=} DA_{c_0 \stackrel{\text{def}}{=} 0}(011, 101)$$

Selected examples:

$$\begin{array}{rcl@{\qquad}rcl} a, b, a \text{ op}_A b & \in & \mathbb{B}^{k+1} & a \text{ op}_M b & \in & \mathbb{B}^{k+2} \\ a \text{ op}_{\alpha A} b & \in & \mathbb{B}^{k+2} & a \text{ op}_{\alpha M} b & \in & \mathbb{B}^{k+3} \end{array}$$

$$\begin{array}{rcll} a +_A b & \stackrel{\text{def}}{=} & r & \text{where} \quad (c, r) \stackrel{\text{def}}{=} DA_0(a, b) \\ a +_M b & \stackrel{\text{def}}{=} & c_{k+1} r & \text{where} \quad (c, r) \stackrel{\text{def}}{=} DA_0(a, b) \\ a +_{SA} b & \stackrel{\text{def}}{=} & r & \text{where} \quad (c, r) \stackrel{\text{def}}{=} DA_0(sx(a), sx(b)) \\ a -_{SA} b & \stackrel{\text{def}}{=} & r & \text{where} \quad (c, r) \stackrel{\text{def}}{=} DA_1(sx(a), \overline{sx(b)}) \end{array}$$

$$\neg_M(b) \stackrel{\text{def}}{=} \overline{b} +_M \mathbb{1} \quad \mathbb{1} \stackrel{\text{def}}{=} \underbrace{0 \ldots\ldots 0}_{k \text{ copies of } 0} 1 \in \mathbb{B}^{k+1}$$

The Additive Digitwize/Columnar Algorithm:

$$(1110, 000) \stackrel{\text{def}}{=} DA_{c_0 \stackrel{\text{def}}{=} 0}(011, 101)$$

Selected examples:

$$\begin{array}{rcl@{\qquad}rcl} a, b, a \text{ op}_A b & \in & \mathbb{B}^{k+1} & a \text{ op}_M b & \in & \mathbb{B}^{k+2} \\ a \text{ op}_{\alpha A} b & \in & \mathbb{B}^{k+2} & a \text{ op}_{\alpha M} b & \in & \mathbb{B}^{k+3} \end{array}$$

$$\begin{array}{rcll} a +_A b & \stackrel{\text{def}}{=} & r & \text{where} \quad (c, r) \stackrel{\text{def}}{=} DA_0(a, b) \\ a +_M b & \stackrel{\text{def}}{=} & c_{k+1} r & \text{where} \quad (c, r) \stackrel{\text{def}}{=} DA_0(a, b) \\ a +_{SA} b & \stackrel{\text{def}}{=} & r & \text{where} \quad (c, r) \stackrel{\text{def}}{=} DA_0(sx(a), sx(b)) \\ a -_{SA} b & \stackrel{\text{def}}{=} & r & \text{where} \quad (c, r) \stackrel{\text{def}}{=} DA_1(sx(a), \overline{sx(b)}) \end{array}$$

$$\neg_M(b) \stackrel{\text{def}}{=} \overline{b} +_M \mathbb{1} \quad \mathbb{1} \stackrel{\text{def}}{=} \underbrace{0 \ldots\ldots 0}_{k \text{ copies of } 0} 1 \in \mathbb{B}^{k+1}$$

The Additive Digitwize/Columnar Algorithm:

$$(1110, 000) \stackrel{\text{def}}{=} DA_{c_0 \stackrel{\text{def}}{=} 0}(011, 101)$$

Selected examples:

$$\begin{array}{rcl@{\qquad}rcl} a, b, a \text{ op}_A b & \in & \mathbb{B}^{k+1} & a \text{ op}_M b & \in & \mathbb{B}^{k+2} \\ a \text{ op}_{\alpha A} b & \in & \mathbb{B}^{k+2} & a \text{ op}_{\alpha M} b & \in & \mathbb{B}^{k+3} \end{array}$$

$$\begin{array}{rcll} a +_A b & \stackrel{\text{def}}{=} & r & \text{where} \quad (c, r) \stackrel{\text{def}}{=} DA_0(a, b) \\ a +_M b & \stackrel{\text{def}}{=} & c_{k+1} r & \text{where} \quad (c, r) \stackrel{\text{def}}{=} DA_0(a, b) \\ a +_{SA} b & \stackrel{\text{def}}{=} & r & \text{where} \quad (c, r) \stackrel{\text{def}}{=} DA_0(sx(a), sx(b)) \\ a -_{SA} b & \stackrel{\text{def}}{=} & r & \text{where} \quad (c, r) \stackrel{\text{def}}{=} DA_1(sx(a), \overline{sx(b)}) \end{array}$$

$$\neg_M(b) \stackrel{\text{def}}{=} \overline{b} +_M \mathbb{1} \quad \mathbb{1} \stackrel{\text{def}}{=} \underbrace{0 \ldots\ldots 0}_{k \text{ copies of } 0} 1 \in \mathbb{B}^{k+1}$$

The Additive Digitwize/Columnar Algorithm:

$$(1110, 000) \stackrel{\text{def}}{=} DA_{c_0 \stackrel{\text{def}}{=} 0}(011, 101)$$

Selected examples:

$$\begin{array}{rcl@{\qquad}rcl} a, b, a \text{ op}_A b & \in & \mathbb{B}^{k+1} & a \text{ op}_M b & \in & \mathbb{B}^{k+2} \\ a \text{ op}_{\alpha A} b & \in & \mathbb{B}^{k+2} & a \text{ op}_{\alpha M} b & \in & \mathbb{B}^{k+3} \end{array}$$

$$\begin{array}{rcll} a +_A b & \stackrel{\text{def}}{=} & r & \text{where} \quad (c, r) \stackrel{\text{def}}{=} DA_0(a, b) \\ a +_M b & \stackrel{\text{def}}{=} & c_{k+1} r & \text{where} \quad (c, r) \stackrel{\text{def}}{=} DA_0(a, b) \\ a +_{SA} b & \stackrel{\text{def}}{=} & r & \text{where} \quad (c, r) \stackrel{\text{def}}{=} DA_0(sx(a), sx(b)) \\ a -_{SA} b & \stackrel{\text{def}}{=} & r & \text{where} \quad (c, r) \stackrel{\text{def}}{=} DA_1(sx(a), \overline{sx(b)}) \end{array}$$

$$\neg_M(b) \stackrel{\text{def}}{=} \overline{b} +_M \mathbb{1} \quad \mathbb{1} \stackrel{\text{def}}{=} \underbrace{0 \ldots\ldots 01}_{k \text{ copies of } 0} \in \mathbb{B}^{k+1}$$

Let $a, b \in \mathbb{B}^{k+1}$. Then

- $a +_A b = \pi_\sigma a +_A \pi_\sigma b$
- $a -_A b = \pi_\sigma a -_A \pi_\sigma b$
- $\pi_\sigma a +_A b = a +_A \pi_\sigma b = \pi_\sigma(a +_A b)$
- $\pi_\sigma a -_A b = a -_A \pi_\sigma b = \pi_\sigma(a -_A b)$
- $\pi_\sigma a +_{ZA} \pi_\sigma b = \pi_\sigma(a +_{SA} b)$
- $a -_{ZA} b = \pi_\sigma a -_{SA} \pi_\sigma b$
- $a +_M b = \pi_\sigma(\pi_\sigma a +_M \pi_\sigma b) \iff \sigma_a = \sigma_b$
- $a +_M b = \pi_\sigma a +_M \pi_\sigma b \iff \sigma_a \neq \sigma_b$

$$\pi_\sigma 101 +_{ZA} \pi_\sigma 011 = 001 +_{ZA} 111 = 0001 +_A 0111 = 1000$$
$$\pi_\sigma(101 +_{SA} 011) = \pi_\sigma(1101 +_A 0011) = \pi_\sigma 0000 = 1000$$

Let $a, b, \in \mathbb{B}^{k+1}$. Then the following equations hold for every such a and b (total correctness)

1. $[\![a]\!] + [\![b]\!] = [\![a +_{ZA} b]\!]$
2. $[\![a]\!] - [\![b]\!] = (\!|a -_{ZA} b|\!)$
3. $(\!|a|\!) + (\!|b|\!) = (\!|a +_{SA} b|\!)$
4. $(\!|a|\!) - (\!|b|\!) = (\!|a -_{SA} b|\!)$

- One *must* prove Equations 1 and 2 by induction on k.
- One can show $(1 \iff 3)$ and $(2 \iff 4)$ by using the calculus of equations.

$$\begin{aligned}
(\!|a|\!) + (\!|b|\!) &= [\![\pi_\sigma a]\!] - 2^k + [\![\pi_\sigma b]\!] - 2^k \\
&= [\![\pi_\sigma a +_{ZA} \pi_\sigma b]\!] - 2^k - 2^k \\
&= [\![\pi_\sigma (a +_{SA} b)]\!] - 2^k - 2^k \\
&= (\!|a +_{SA} b|\!) + 2^{k+1} - 2^k - 2^k
\end{aligned}$$

$\boldsymbol{a, b, r} \stackrel{\mathbf{def}}{=} \boldsymbol{a -_A b} \in \mathbb{B}^{k+1}$ (correct answer is $[\![a]\!] - [\![b]\!] \in \mathbb{Z}$).
Then the following conditions are pairwise *pairwise equivalent*

Condition 1

1. **Correctness:** $[\![a]\!] - [\![b]\!] = (\!|r|\!)$
2. **Ranges$_{\mathbf{middle+to-}}$:** $-2^k \leq [\![a]\!] - [\![b]\!] \leq 2^k - 1$
3. **Sign Occurrences** $\langle \sigma_a, \sigma_b, \sigma_r \rangle$**:** Any of the following six occurrences $\langle 1, 0, 0 \rangle$, $\langle 0, 1, 1 \rangle$, $\langle 1, 1, 0/1 \rangle$, $\langle 0, 0, 0/1 \rangle$, but not the other two.
4. **Overflow:** $\sigma_c \neq \sigma_r$.
5. **Sign Equation:** $-\sigma_a + \sigma_b - \sigma_r = 0, -1$

range of $(k+1)$*-digit signed binary*

$a, b, r \stackrel{\text{def}}{=} a -_A b \in \mathbb{B}^{k+1}$ (correct answer is $[\![a]\!] - [\![b]\!] \in \mathbb{Z}$).
Then the following conditions are pairwise *pairwise equivalent*

Condition 2

1. **Correctness:** $[\![a]\!] - [\![b]\!] = (\!|1r|\!) \wedge \sigma_r = 0$
2. **Ranges$_{\text{lower}-}$:** $-2^{k+1} + 1 \leq [\![a]\!] - [\![b]\!] \leq -2^k - 1$
3. **Sign Occurrences:** $\langle 0, 1, 0 \rangle$,
4. **Overflow:** $\sigma_c = \sigma_r = 0$.
5. **Sign Equation:** $-\sigma_a + \sigma_b - \sigma_r = 1$

min of $[\![a]\!] - [\![b]\!]$ *to min*-1 *of* $(k+1)$*-digit signed binary*

$a, b, r \stackrel{\text{def}}{=} a -_A b \in \mathbb{B}^{k+1}$ (correct answer is $[\![a]\!] - [\![b]\!] \in \mathbb{Z}$).
Then the following conditions are pairwise *pairwise equivalent*

Condition 3

1. **Correctness:** $[\![a]\!] - [\![b]\!] = (\!|0r|\!) \wedge \sigma_r = 1$
2. **Ranges$_{\text{upper}+}$:** $2^k \leq [\![a]\!] - [\![b]\!] \leq 2^{k+1} - 1$
3. **Sign Occurrences:** $\langle 1, 0, 1 \rangle$,
4. **Overflow:** $\sigma_c = \sigma_r = 1$.
5. **Sign Equation:** $-\sigma_a + \sigma_b - \sigma_r = -2$

max$+1$ *of* $(k+1)$*-digit signed binary to max of* $[\![a]\!] - [\![b]\!]$

and moreover for all a and b

- the **Conditions 1, 2, 3** are mutually exclusive; and
- at least one **Condition i** must be true.

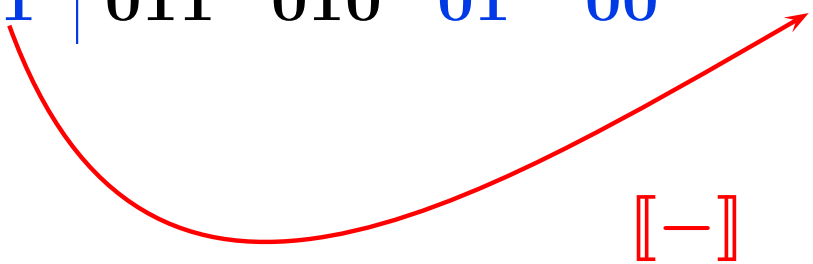
Operator Tables and Conditions

Let $a, b \in \mathbb{B}^2$. We can visualize the conditions within an operator table.

Condition 3, $+$. Condition 1, $+$ to -. Condition 2, -.

$-_A$	00	01	10	11
00	00	11	10	101
01	01	00	11	10
10	010	01	00	11
11	011	010	01	00

$-_{\mathbb{Z}}$	0	1	2	3
0	0	-1	-2	-3
1	1	0	-1	-2
2	2	1	0	-1
3	3	2	1	0



$101 = 101$ $\qquad$ $011 = 011$

Operator Tables and Conditions

Let $a, b \in \mathbb{B}^2$. We can visualize the conditions within an operator table.

Condition 3, $+$. Condition 1, $+$ to -. Condition 2, -.

$-_A$	00	01	10	11
00	00	11	10	101
01	01	00	11	10
10	010	01	00	11
11	011	010	01	00

$-_{\mathbb{Z}}$	0	1	2	3
0	0	-1	-2	-3
1	1	0	-1	-2
2	2	1	0	-1
3	3	2	1	0

$$[\![-]\!]$$

$$101 = 101 \qquad 011 = 011$$

Operator Tables and Conditions

Let $a, b \in \mathbb{B}^2$. We can visualize the conditions within an operator table.

Condition 3, $+$. Condition 1, $+$ to -. Condition 2, -.

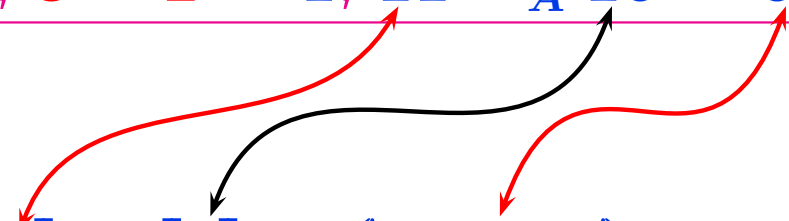
$-_A$	00	01	10	11
00	00	11	10	101
01	01	00	11	10
10	010	01	00	11
11	011	010	01	00

$-_{\mathbb{Z}}$	0	1	2	3
0	0	-1	-2	-3
1	1	0	-1	-2
2	2	1	0	-1
3	3	2	1	0

$$(\!| - |\!)$$

$$101 = 101 \qquad\qquad 011 = 011$$

Examples: $k = 1$, $\mathbf{3 - 2 = 1}$, $11 -_A 10 = 01$ with carry 111



Condition 1. True.

1. **Correctness:** $[\![a]\!] - [\![b]\!] = (\!|a -_A b|\!)$
2. **Ranges:** $-2^k \leq [\![a]\!] - [\![b]\!] \leq 2^k - 1$
3. **Sign Occurrences:** Any of the following six occurrences $\langle 1, 0, 0\rangle$, $\langle 0, 1, 1\rangle$, $\langle 1, 1, 0/1\rangle$ $\langle 0, 0, 0/1\rangle$, but not the other two.
4. **Overflow:** $\sigma_r \neq \sigma_c$.
5. **Sign Equation:** $-\sigma_a + \sigma_b - \sigma_r = 0, -1$

1. **Correctness:** $[\![11]\!] - [\![10]\!] = (\!|01|\!)$
2. **Ranges** $-2 \leq 1 \leq 1$
3. **Sign Occurrences:** $\langle 1, 1, 0\rangle$.
4. **Overflow:** $0 \neq 1$.
5. **Sign Equation:** $-1 + 1 - 0 = 0$

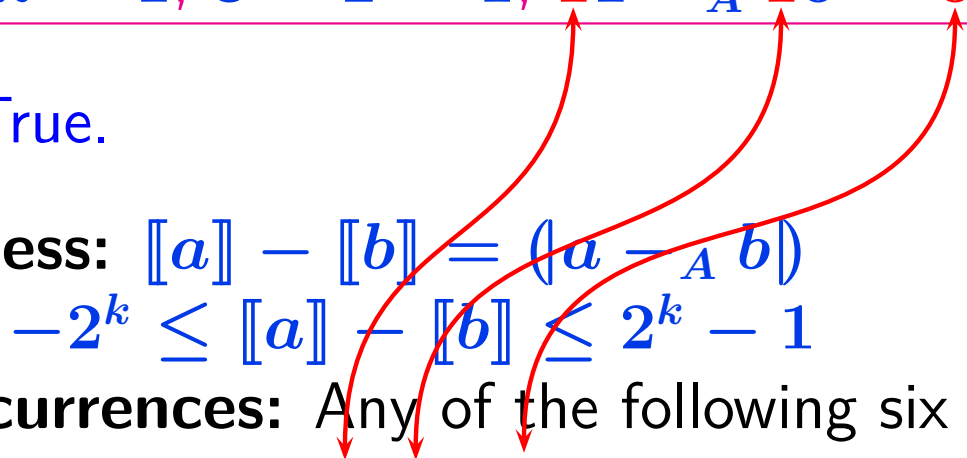
Examples: $k = 1$, $3 - 2 = 1$, $11 -_A 10 = 01$ with carry 111

Condition 1. True.

1. **Correctness:** $[\![a]\!] - [\![b]\!] = (\!|a -_A b|\!)$
2. **Ranges:** $-2^k \leq [\![a]\!] - [\![b]\!] \leq 2^k - 1$
3. **Sign Occurrences:** Any of the following six occurrences $\langle 1, 0, 0\rangle$, $\langle 0, 1, 1\rangle$, $\langle 1, 1, 0/1\rangle$ $\langle 0, 0, 0/1\rangle$, but not the other two.
4. **Overflow:** $\sigma_r \neq \sigma_c$.
5. **Sign Equation:** $-\sigma_a + \sigma_b - \sigma_r = 0, -1$

1. **Correctness:** $[\![11]\!] - [\![10]\!] = (\!|01|\!)$
2. **Ranges** $-2 \leq 1 \leq 1$
3. **Sign Occurrences:** $\langle 1, 1, 0\rangle$.
4. **Overflow:** $0 \neq 1$.
5. **Sign Equation:** $-1 + 1 - 0 = 0$

Examples: $k = 1$, $3 - 2 = 1$, $11 -_A 10 = 01$ with carry 111



Condition 1. True.

1. **Correctness:** $[\![a]\!] - [\![b]\!] = (\!|a -_A b|\!)$
2. **Ranges:** $-2^k \leq [\![a]\!] - [\![b]\!] \leq 2^k - 1$
3. **Sign Occurrences:** Any of the following six occurrences $\langle 1, 0, 0\rangle$, $\langle 0, 1, 1\rangle$, $\langle 1, 1, 0/1\rangle$ $\langle 0, 0, 0/1\rangle$, but not the other two.
4. **Overflow:** $\sigma_r \neq \sigma_c$.
5. **Sign Equation:** $-\sigma_a + \sigma_b - \sigma_r = 0, -1$

1. **Correctness:** $[\![11]\!] - [\![10]\!] = (\!|01|\!)$
2. **Ranges** $-2 \leq 1 \leq 1$
3. **Sign Occurrences:** $\langle 1, 1, 0\rangle$.
4. **Overflow:** $0 \neq 1$.
5. **Sign Equation:** $-1 + 1 - 0 = 0$

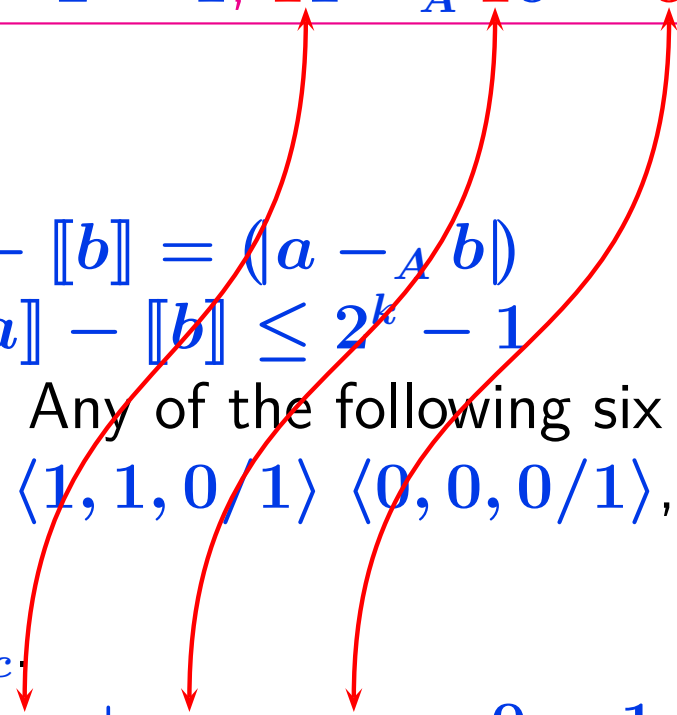
Examples: $k = 1$, $3 - 2 = 1$, $11 -_A 10 = 01$ with carry 111

Condition 1. True.

1. **Correctness:** $[\![a]\!] - [\![b]\!] = (\!|a -_A b|\!)$
2. **Ranges:** $-2^k \leq [\![a]\!] - [\![b]\!] \leq 2^k - 1$
3. **Sign Occurrences:** Any of the following six occurrences $\langle 1, 0, 0\rangle$, $\langle 0, 1, 1\rangle$, $\langle 1, 1, 0/1\rangle$ $\langle 0, 0, 0/1\rangle$, but not the other two.
4. **Overflow:** $\sigma_r \neq \sigma_c$.
5. **Sign Equation:** $-\sigma_a + \sigma_b - \sigma_r = 0, -1$

1. **Correctness:** $[\![11]\!] - [\![10]\!] = (\!|01|\!)$
2. **Ranges** $-2 \leq 1 \leq 1$
3. **Sign Occurrences:** $\langle 1, 1, 0\rangle$.
4. **Overflow:** $0 \neq 1$.
5. **Sign Equation:** $-1 + 1 - 0 = 0$

Examples: $k = 1$, $3 - 2 = 1$, $11 -_A 10 = 01$ with carry 111

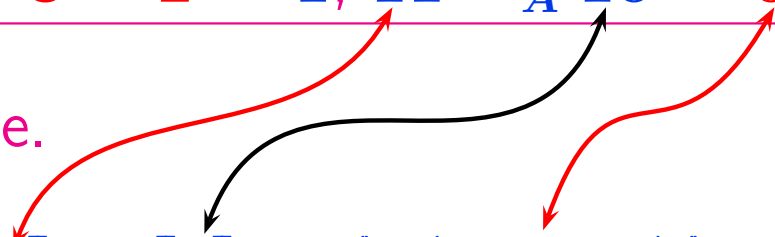


Condition 1. True.

1. **Correctness:** $[\![a]\!] - [\![b]\!] = (\!|a -_A b|\!)$
2. **Ranges:** $-2^k \leq [\![a]\!] - [\![b]\!] \leq 2^k - 1$
3. **Sign Occurrences:** Any of the following six occurrences $\langle 1, 0, 0\rangle$, $\langle 0, 1, 1\rangle$, $\langle 1, 1, 0/1\rangle$ $\langle 0, 0, 0/1\rangle$, but not the other two.
4. **Overflow:** $\sigma_r \neq \sigma_c$.
5. **Sign Equation:** $-\sigma_a + \sigma_b - \sigma_r = 0, -1$

1. **Correctness:** $[\![11]\!] - [\![10]\!] = (\!|01|\!)$
2. **Ranges** $-2 \leq 1 \leq 1$
3. **Sign Occurrences:** $\langle 1, 1, 0\rangle$.
4. **Overflow:** $0 \neq 1$.
5. **Sign Equation:** $-1 + 1 - 0 = 0$

Examples: $k = 1$, $\mathbf{3 - 2 = 1}$, $\mathbf{11 -_A 10 = 01}$ with carry $\mathbf{111}$



Condition Set 2. False.

1. **Correctness:** $[\![a]\!] - [\![b]\!] = (\!|1(a -_A b)|\!) \wedge \sigma_r = 0$
2. **Ranges:** $-2^{k+1} + 1 \leq [\![a]\!] - [\![b]\!] \leq -2^k - 1$
3. **Sign Occurrences:** $\langle 0, 1, 0 \rangle$
4. **Overflow:** $\sigma_r = \sigma_c = 0$.
5. **Sign Equation:** $-\sigma_a + \sigma_b - \sigma_r = 1$

1. **Correctness:** $[\![\mathbf{11}]\!] - [\![\mathbf{10}]\!] = (\!|1(\mathbf{01})|\!)$
2. **Ranges:** $-3 \leq 1 \leq -3$
3. **Sign Occurrences:** $\langle 1, 1, 0 \rangle$.
4. **Overflow:** $0 = 1 = 0$.
5. **Sign Equation:** $-1 + 1 - 0 = 0$

Examples: $k = 1$, $3 - 2 = 1$, $11 -_A 10 = 01$ with carry 111

Condition Set 2. False.

1. **Correctness:** $[\![a]\!] - [\![b]\!] = (\!|1(a -_A b)|\!) \wedge \sigma_r = 0$
2. **Ranges:** $-2^{k+1} + 1 \leq [\![a]\!] - [\![b]\!] \leq -2^k - 1$
3. **Sign Occurrences:** $\langle 0, 1, 0 \rangle$
4. **Overflow:** $\sigma_r = \sigma_c = 0$.
5. **Sign Equation:** $-\sigma_a + \sigma_b - \sigma_r = 1$

1. **Correctness:** $[\![11]\!] - [\![10]\!] = (\!|1(01)|\!)$
2. **Ranges:** $-3 \leq 1 \leq -3$
3. **Sign Occurrences:** $\langle 1, 1, 0 \rangle$.
4. **Overflow:** $0 = 1 = 0$.
5. **Sign Equation:** $-1 + 1 - 0 = 0$

Examples: $k = 1$, $3 - 2 = 1$, $11 -_A 10 = 01$ with carry 111

Condition Set 2. False.

1. **Correctness:** $[\![a]\!] - [\![b]\!] = (\!|1(a -_A b)|\!) \wedge \sigma_r = 0$
2. **Ranges:** $-2^{k+1} + 1 \leq [\![a]\!] - [\![b]\!] \leq -2^k - 1$
3. **Sign Occurrences:** $\langle 0, 1, 0 \rangle$
4. **Overflow:** $\sigma_r = \sigma_c = 0$.
5. **Sign Equation:** $-\sigma_a + \sigma_b - \sigma_r = 1$

1. **Correctness:** $[\![11]\!] - [\![10]\!] = (\!|1(01)|\!)$
2. **Ranges:** $-3 \leq 1 \leq -3$
3. **Sign Occurrences:** $\langle 1, 1, 0 \rangle$.
4. **Overflow:** $0 = 1 = 0$.
5. **Sign Equation:** $-1 + 1 - 0 = 0$

Examples: $k = 1$, $3 - 2 = 1$, $11 -_A 10 = 01$ with carry 111

Condition Set 2. False.

1. **Correctness:** $[\![a]\!] - [\![b]\!] = (\!|1(a -_A b)|\!) \wedge \sigma_r = 0$
2. **Ranges:** $-2^{k+1} + 1 \leq [\![a]\!] - [\![b]\!] \leq -2^k - 1$
3. **Sign Occurrences:** $\langle 0, 1, 0 \rangle$
4. **Overflow:** $\sigma_r = \sigma_c = 0$.
5. **Sign Equation:** $-\sigma_a + \sigma_b - \sigma_r = 1$

1. **Correctness:** $[\![11]\!] - [\![10]\!] = (\!|1(01)|\!)$
2. **Ranges:** $-3 \leq 1 \leq -3$
3. **Sign Occurrences:** $\langle 1, 1, 0 \rangle$.
4. **Overflow:** $0 = 1 = 0$.
5. **Sign Equation:** $-1 + 1 - 0 = 0$

Examples: $k = 1$, $3 - 2 = 1$, $11 -_A 10 = 01$ with carry 111

Condition Set 2. False.

1. **Correctness:** $[\![a]\!] - [\![b]\!] = (\!|1(a -_A b)|\!) \wedge \sigma_r = 0$
2. **Ranges:** $-2^{k+1} + 1 \leq [\![a]\!] - [\![b]\!] \leq -2^k - 1$
3. **Sign Occurrences:** $\langle 0, 1, 0 \rangle$
4. **Overflow:** $\sigma_r = \sigma_c = 0$.
5. **Sign Equation:** $-\sigma_a + \sigma_b - \sigma_r = 1$

1. **Correctness:** $[\![11]\!] - [\![10]\!] = (\!|1(01)|\!)$
2. **Ranges:** $-3 \leq 1 \leq -3$
3. **Sign Occurrences:** $\langle 1, 1, 0 \rangle$.
4. **Overflow:** $0 = 1 = 0$.
5. **Sign Equation:** $-1 + 1 - 0 = 0$

Condition 1.

The following are *pairwise equivalent*

- **Correctness:** $(\!|a|\!) + (\!|b|\!) = (\!|a +_M b|\!)$
- **Sign Occurrences:** $\langle \sigma_a, \sigma_b, \sigma_r \rangle$ can be any one of the following four sign-occurrences $\langle 0, 0, 0/1 \rangle$, $\langle 1, 1, 0/1 \rangle$.

Condition 2.

as are

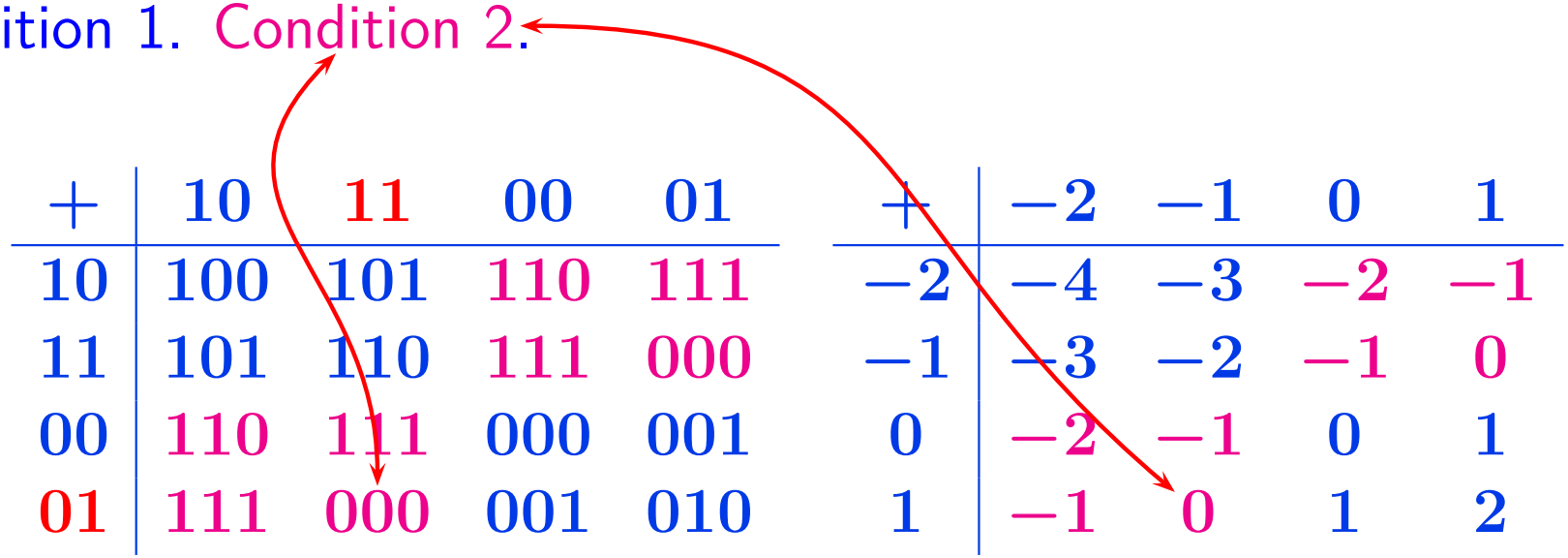
- **Correctness:** $(\!|a|\!) + (\!|b|\!) = (\!|\pi_\sigma(a +_M b)|\!)$
- **Sign Occurrences:** $\langle \sigma_a, \sigma_b, \sigma_r \rangle$ can be any one of the following four sign-occurrences $\langle 0, 1, 0/1 \rangle$.

and moreover for all a and b

- The **Conditions 1, 2** are mutually exclusive; and
- at least one **Condition i** must be true.

Let $a, b \in \mathbb{B}^2$. We can visualize the conditions within an operator table.

Condition 1. Condition 2.



$+$	10	11	00	01
10	100	101	110	111
11	101	110	111	000
00	110	111	000	001
01	111	000	001	010

$+$	-2	-1	0	1
-2	-4	-3	-2	-1
-1	-3	-2	-1	0
0	-2	-1	0	1
1	-1	0	1	2

$$\begin{cases} (\!|01|\!) + (\!|11|\!) = 1 + -1 = 0 \\ (\!|\pi_\sigma(01 +_M 11)|\!) = (\!|\pi_\sigma(100)|\!) = (\!|000|\!) = 0 \end{cases}$$

$(\mathbb{B}^{k+1}, +_A, \vec{0})$ is a group and it is isomorphic to $(\mathbb{Z}_{2^{k+1}}, +, [0])$.

A choice of isomorphism is given by the function *compositions*

$$\mathbb{B}^{k+1} \underset{\rho^u}{\overset{[\![-]\!]}{\underset{\cong}{\rightleftarrows}}} Z^{u;k} \underset{q^{-1}}{\overset{q}{\underset{\cong}{\rightleftarrows}}} \mathbb{Z}_{2^{k+1}}$$

where $q^{-1}([z]) \stackrel{\text{def}}{=} z \ MOD \ 2^{k+1}$.

Miscellaneous Observations

- Java negation of `Integer.MIN_VALUE` is `Integer.MIN_VALUE`. Ooops!
- `Integer.MIN_VALUE` is Fixed by negation.
- Let $\mathbf{1} \in \mathbb{Z}_{\mathbf{2}}$ act on $\mathbb{Z}_{\mathbf{2}^{k+1}}$ by computing negations.
- $\mathbf{0} \in \mathbb{Z}_{\mathbf{2}^{k+1}}$ is fixed by $\mathbf{1}$ and so the number of points fixed by $\mathbf{1}$ is at least one, and ...
- equals $|\mathbb{Z}_{\mathbf{2}}| * |\mathbb{Z}_{\mathbf{2}^{k+1}}|$, even, by Burnside's Lemma.
- Hence at least one other element of $\mathbb{Z}_{\mathbf{2}^{k+1}}$ is fixed by $\mathbf{1} \in \mathbb{Z}_{\mathbf{2}}$, and so is equal to it's own negation.

Summary and Conclusions

- Clarified text book accounts of ALU signed binary correctness.
- Worthwhile: (good) students appreciate a uniform and consolidated account.
- In a paper we also outline
  - Full intuitions of mutual correctness conditions, explaining how they arise naturally (from each other).
  - Many different approaches to proofs.
  - The history behind arithmetic correctness and hardware verifications.
- I found it quite fun to do this work; I am not aware that anyone has ever written down these results in such a compact form, and the calculus and Maths-Operators are not presented in the literature.