On Correctness of Digitwize Arithmetic

School of Computing and Mathematical Sciences, University of Leicester, University Road, Leicester, LE1 7RH, UK

S. BALCO AND R. L. CROLE.

Mathematical addition of natural numbers performed by a human on paper, where columns of digits are added up and carry and output digits computed, is always correct. Addition of natural numbers, performed by an Arithmetic Logic Unit (ALU) in a computer processor, may or may not be correct.

Generalising, one might consider other operations and their correctness. This paper explores operations each of which is a form of addition or subtraction or negation. The operations are split into two groups: one group has a feature commonly seen in ALUs, and the other has a feature common to the way humans do digit-by-digit columnar computations. For each operation, a correctness theorem is proven, where conditions may be required to guarantee correctness.

In hardware texts, one finds conditions for ALU addition correctness. However, the details are often informal and incomplete. Mathematics texts tend not to discuss correctness. Our new contributions are:

- the precise definition of a range of arithmetic operations, with some not usually considered in the literature, together with an equational calculus;
- the precise definition of a range of comprehensive correctness theorems with full proofs of correctness;
- a thorough analysis of how the theorems were originally derived;
- an exploration of modular approaches to proofs; and
- a representative selection of proofs of theorems by machine, in Isabelle/HOL.

Author's address: S. Balco and R. L. Crole.School of Computing and Mathematical Sciences, University of Leicester, University Road, Leicester, LE1 7RH, UK.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM.

Manuscript submitted to ACM

ACM Reference Format:
S. Balco and R. L. Crole.. 2026. On Correctness of Digitwize Arithmetic. 1, 1 (November 2026), 61 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

1.1 Fundamental Ideas

In this paper, $\mathbb{B} \stackrel{\text{def}}{=} \{ 0, 1 \}$ is the set of Booleans, $\mathbb{N}$ is the set of natural numbers and $\mathbb{Z}$ is the set of integers. We assume there are well defined notions of addition and other operations.

In Java, as in many programming languages, one has `Math.abs(x)` which returns the absolute value of an integer `x`. Except that it doesn't. At least, there is an exception! If `x` is equal to `Integer.MIN_VALUE`, the most negative `int` value, the output is again `Integer.MIN_VALUE` which is incorrect. This exception arises due to the finite memory assigned to `int` values: the *correct* absolute value in $\mathbb{Z}$ of the integer `Integer.MIN_VALUE` is "too large" to be any `int` value, so if the return type of `Math.abs(x)` is `int` the language *cannot possibly* return `Math.abs(Integer.MIN_VALUE)` correctly. The reason that `Integer.MIN_VALUE` is (often) returned is explained later in this paper in Remark 4.7.

There is a clear mathematical definition of $\mathbb{Z}$, and also of the unary operation *abs* on $\mathbb{Z}$. When we use a computer to compute *abs*, the user begins with an integer. This is converted to a binary number, *a (computer) representation*. Computation of the absolute value (indeed other operations such as negation and arithmetic) takes place in the computer processor Arithmetic Logic Unit (ALU). This is an implementation of an *(algorithmic) representation* of *abs*. The ALU output is then converted to an integer and returned to the user.

So we have a representation of data (inputs and outputs), a representation of operations on the data, and as we saw above, in some cases the representations are *incorrect*.

The purpose of this paper is to present in a comprehensive and uniform way the theory that underpins representational correctness for a small set of common operations.

Many text books on Computer Architecture discuss correctness. However, the accounts are often informal, and certainly not comprehensive.[1] For example, the *textbook account* of the digital logic tests for + performed by an ALU corresponds to the *first* of three parts of a theorem in this paper (Theorem 4.9). The other two parts are seemingly never mentioned.

[1]This paper came about after teaching Computer Architecture for a few years. I was never able to find an all-encompassing treatment of this important subject. Neither could I find a research article that unified all the key ideas from both mathematics and engineering.

In the remainder of the introduction we give more details of these ideas, mainly by informal example, so the reader has a good foundation for understanding the formal approach of the paper.

At school one learns how to write down integers using a syntax/notation consisting of sequences of the denary[2] digits $0, 1, 2, \ldots, 9$ prepended by $+/-$. This yields denary representations of the integers. One learns to perform addition algorithmically by adding up columns of digits and writing down remainders and carries. A trivial example is

$$\begin{array}{rrr} & 2 & 8 \\ + & 1 & 9 \\ \hline & 1 & 0 \\ \hline & 4 & 7 \end{array}$$

where the output is 47 and the carry is 10. We will refer to this algorithm as the *Digitwize Algorithm*, *DA*, since it works digit-column by digit-column.

One can do the same for other integer representations, such as sequences of the binary digits $0, 1$, that is, binary numbers (where we consider each one to have $k + 1$ digits in positions k to 0 left to right). Here is an example with $k = 3$ which illustrates our terminology, where bold **1** is called the *overflow* digit (we explain why in due course):

Input/Operand		1	1	1
Input/Operand		0	1	0
Carry	**1**	1	0	0
Output (Remainder digits)	1	0	0	1

Remark 1.1. *In this paper we are going to regard* $\mathbb{N}$ *and* $\mathbb{Z}$ *as semantic domains, in the sense that users understand their data and operations as a natural language, so to speak. Of course, when we write down say* 28*, strictly speaking this is a denary representation of a particular element. None-the-less we will regard the translation of* 11100 *into* 28 *as giving a semantics:* 11100 *denotes* $28 \in \mathbb{Z}$*.*

This paper assumes knowledge of **unsigned** and **signed (twos-complement)** binary number representations: In each case this is a sequence of 0s and 1s, and the semantics maps a sequence into $\mathbb{N}$ and $\mathbb{Z}$ respectively. Examples of signed numbers are 1001 and 0110 with semantics $-7 \in \mathbb{Z}$ and $+6 \in \mathbb{Z}$.

There are some obvious questions:

[2]We refer to denary, not decimal, since we will not work with fractions (bicimals are binary fractions)!!

(1) Is each representation of the integers correct (in some sense)?
(2) Do digitwize algorithms give correct answers to addition, subtraction and negation on the integers? Indeed, what *exactly* does correct mean?
(3) Are there algorithms which are always (ie totally) correct? Can we state precisely when algorithms are conditionally correct? What are the exceptions?

1.2 Exploring the Questions

In regard of Question (1) we do not say anything fundamentally new in this paper. However, for completeness, we do give a formal statement of correctness of binary representations in Theorem 2.3; for now we make the following remark which together with Theorem 2.3 provides an answer to Question (1).

Remark 1.2. *For any integer m (for example -8) we will show that there is a canonical representation $\rho(m)$ as a binary number of least length (for the example,* 1000*), such that there is an inverse function $- \mapsto \{-\}$ returning the original integer $m = \{\rho(m)\}$ (for the example, $\{1000\} = -8$). We will refer to the inverse as a* denotational *or* semantic *function.*

Note that representations might not be unique: -8 is also represented by 111000*. However we will always work with canonical representations by specifying the lengths of the binary numbers we work with. Typically a Theorem will involve k-digit binary numbers.*

We now explore the other questions informally. We look at examples where the calculations mimic the behaviour of an ALU; but also examples mimicking calculations as taught to mathematics students. First we make a definition about overflow, since this often causes confusion.

Definition 1.1. **Overflow** is defined as the *logical negation* of correctness. Thus *overflow* is a synonym for ***in**correctness.*

However, perhaps confusingly, if one has two k-digit *operands*, the carry digit in position $k+1$ will be known as the **overflow digit**.

Remark 1.3. *In this paper, we will divide all of our theorems into those involving "maths-operators" and those involving "ALU-operators". Each operator is defined in terms of the DA. Operands $a = a_k \ldots a_1 a_0$ and $b = b_k \ldots b_1 b_0$ are input. The DA always produces an output (c, r) where c is the carry (sequence of carry digits) and r is "the result of the operator", but we usually refer to the "output" r. Maths-operator outputs take the form $r_{k+1} r_k \ldots r_1 r_0$; and ALU-operator outputs take the form $r_k \ldots r_1 r_0$, thus omitting leftmost digit r_{k+1}. We will*

often denote operators by $\square$*, and we will then write* $a \square b \stackrel{\text{def}}{=} r$*. Examples of such operators are given next where we indicate leftmost digit* r_{k+1} *using a superscript* r *(ditto* c*).*

1.2.1 Example Calculations by a Computer ALU. An example with 3-digit unsigned binary. Consider $1 + 3 =_{\mathbb{N}} 4$ and $3 + 5 =_{\mathbb{N}} 8$. We assume that 1, 3, 5 have correct unsigned representations. These appear as operands in the following digitwise computations, where $+_{ALU}$ is an ALU-operator.

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

The ALU outputs are $001 +_{ALU} 011 = 100$ which is 4, correct, and $011 +_{ALU} 101 = 000$ which is 0, incorrect. Note that the overflow digit ${}^{c}\delta$ equals 0 and 1 respectively. Of course it is very clear for unsigned binary addition that ALU outputs are correct just in case $\delta = 0$, so we may state that overflow occurs just in case $\delta = 1$.

Now an example with signed binary, using the same digitwise computations. Consider $1+3 =_{\mathbb{Z}} 4$ and $3+(-3) =_{\mathbb{Z}} 0$. The ALU outputs are 100 which is -4, incorrect, and 000 which is 0, correct. Note that overflow occurred in computing $1 + 3$ but the overflow digit equals 0. In fact we shall see that overflow for signed binary occurs just in case the overflow digit, and the carry digit to its immediate right, are unequal. Since most computers would operate with "integer inputs" (so to speak), and produce an "integer output", this example matches what one tends to see in Computer Architecture textbooks when the arithmetic performed by the ALU is described. One also meets the so-called sign-bit (sign-digit) conditions: the ALU is correct just in case the first digit of each operand is unequal (this is the case when computing $3 + (-3)$), or these digits are equal and also equal to the first digit of the output. Neither is the case in computing $1 + 3$.

Now in these examples, the binary operands and outputs are either *all* signed or *all* unsigned. However, what might be true if we allow some mixing? For example, when computing $3 + (-3)$, if we regarded the ALU output 000 as *unsigned binary* it *would* be correct.

What about $(+3) - (+5) =_{\mathbb{Z}} -2$ where we regard this as a subtraction of unsigned binary numbers? If the ALU output is also unsigned, it cannot correctly represent -2. The ALU

computation, complement 101 and add 1, is

$$\begin{array}{rccc} & 0 & 1 & 1 \\ -_{ALU} & 1 & 0 & 1 \\ \hline {\scriptstyle c} & & & 1 \\ \hline {\scriptstyle r} & & & \end{array} \qquad \begin{array}{rccc} & 0 & 1 & 1 \\ +_{ALU} & 0 & 1 & 0 \\ \hline {\scriptstyle c}0 & 1 & 1 & 1 \\ \hline {\scriptstyle r}0 & 1 & 1 & 0 \end{array}$$

Then the output 110 *is* correct if it is understood as a signed number!

Finally, we are interested in integer negation, that is, sign swapping. The negation in $\mathbb{Z}$ of -3 is 3. So we seek operators such as $\neg_{ALU}$ which (if correct) would yield $\neg(101) = 011$.

1.2.2 Example Calculations by a Mathematician. Looking back at the unsigned calculations of $1+3=4$ and $3+5=8$, the Maths outputs are 0100 which is 4 and 1000 which is 8. Thus both calculations are now correct. If one picks two unsigned binary numbers of the same length, and adds them, to ensure that the output is correct one always needs to include the leftmost carry digit if it is a 1 (and including 0 is fine in any case).

However, with signed binary, Maths outputs may be wrong. Looking back at the example of $3+(-3) =_{\mathbb{Z}} 0$, we have $011 +_{Math} 101 = 1000$, wrong since 1000 represents -8.

To finish our examples, look back at $(+3)-(+5)$ where we regarded this as a subtraction of unsigned binary numbers. In this case, the ALU output when regarded as signed binary was correct. However we have $011 -_{Math} 101 = 0110$, and so the Maths output is always wrong (it denotes a positive integer, signed or unsigned).

1.3 Making Questions 2 and 3 More Precise

Thinking about the examples above, we have

- $\mathbb{N}$ and $\mathbb{Z}$, with representations using unsigned and signed binary; and
- operations on $\mathbb{N}$ and $\mathbb{Z}$ of addition $+$ and subtraction $-$, with representations such as $+/-_{ALU}$ and $+/-_{Math}$.

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

$$m, n \xrightarrow{\text{compute representations}} \rho(m), \rho(n) \xrightarrow{\text{compute binary operator}} \rho(m) \Box \rho(n) = \rho(r) \xrightarrow{\text{compute semantics}} r$$

Note that by Remark 1.2, since we work with canonical representations, we can equivalently write correctness as

$$\{a\} \text{ op } \{b\} = \{a \,\square\, b\}$$

for all binary numbers a and b of the same length.

This provides a preliminary answer to Question 2 (what is correctness for operations?); and in regard of Question 3, we shall soon formulate simple conditions (such as the sign-digit and overflow-digit conditions, or perhaps single exceptions as was the case with *abs*) that guarantee such correctness equations are true.

Summarising we have

Definition 1.2. **Correctness Equations** take the form

$$\forall a, b \in \mathbb{B}^{k+1}. \quad \underbrace{\{a\} \text{ op } \{b\} = \{a \,\square\, b\}}_{Eq(a,b)} \qquad \underbrace{\text{op } (\{b\}) = \{\square(b)\}}_{Eq(b)} \qquad \dagger$$

We sometimes refer to $\{a\}$ op $\{b\}$ or op $(\{b\})$ as the **correct integer answer**, and to $\{a \,\square\, b\}$ or $\{\square(b)\}$ as the **computer/maths integer** answer (on the assumption the expressions are calculated correctly!).

1.4 Our Contributions

The remainder of this paper is devoted to the study of correctness equations of the form †. We will consider a range of choices for the operators op and $\square$, and for the semantic function $\xi \mapsto \{\xi\}$, and investigate formally (stating and proving theorems) whether the chosen correctness equation is true for all a and b, or whether we can find a condition $\Phi(a, b)$ implying the equation holds.

- We develop a number of properties of our operators $\square$. These are given in Proposition 3.3. The properties amount to an *equational calculus* and are very useful for establishing connections between operators, and are used in many of the formal proofs of Theorems.
- Some of the equation instances are well known in the literature. For example, the combination of signed integers, integer addition on $\mathbb{Z}$, and ALU binary addition $+_{ALU}$, leading to the standard sign-digit conditions for correctness, is well known to computer processor engineers (indeed, first year undergraduates). See Condition Set 1 of Theorem 4.9.
- Condition Set 1 provides the standard *overflow digit test* and the *sign-digit test*. Engineers usually only study these conditions for *signed* binary (hence "sign" tests).

Here, however, the word "sign" test is too suggestive: Similar conditions *also work for unsigned binary, and for Maths operators as well as ALU operators*. These are all defined and explored in this paper, and we believe these have not been worked out before.

- In each theorem, Condition Set 1 states when a correctness equation holds. However, in the failure cases, we give further Condition Sets that inform us how the correct integer answer, $\{a\}$ op $\{b\}$, *can* be computed in terms of $\square$. To our knowledge this has never been fully worked out and documented before. We feel that this gives a very pretty and uniform view.
- Some of the theorems are completely new. For example, the combination of unsigned binary operands, subtraction with natural number operands, ALU binary addition $+_{ALU}$, but signed binary outputs. See Theorem 4.8.
- We also believe that this is the first time there has been a formal definition of (recognition of) Maths-operators. As such, we believe that all the theorems in Section 5 are new. Indeed, in one theorem we even see $\{0\}$ op $\{0\} = 0 \neq \{0 \square 0\}$.
- All of the ALU-operator theorems are verified in Isabelle/HOL.

Remark 1.4 (Mathematical/Human Proofs by Hand). *All of the theorems in this paper have "typical" human proofs. By "typical" we mean that representative examples are given for some but not all induction cases in a proof.*

It is not essential to examine the proofs in detail in order to understand the Theorems. All Theorems are easily tested with pencil and paper, by taking values for binary numbers and performing simple calculations using the definitions in the paper. As such all formal proofs are placed in an Appendix.

However, we have worked out strategies for the most efficient ways of conducting proofs, and these are illustrated in the main text. We also point out some examples of modularity, whereby we can use one theorem to prove another. For these reasons we do feel that the proofs are of interest in there own right. We give more detail than is perhaps typical in a research paper, since experience showed us how easy it is to make minor errors in this work. Indeed, this is one very good reason for giving machine proofs in Isabelle/Hol.

Remark 1.5 (Machine Proofs in Isabelle/HOL). *Proofs which are verified in Isabelle/HOL are labelled* VERIFIED IN HOL. *Please note that all theorems in Section 4.2, Section 4.3 and Section 4.4 are machine checked. Please visit*

https://github.com/goodlyrottenapple/compArith/

and look at the README.md file.

1.5 Organisation of the Paper

Remark 1.6. *The majority of this paper consists of the statement of correctness theorems without proof, along with illustrative remarks and examples. The thought is that, so long as a reader is familiar with the notation and results in Section 2 and Section 3, and also Section 4.1, then the Theorems themselves can be understood fairly easily: one may simply plug and play by computing with values of a and b. Indeed, to some extent one can just dip in at any point and there is no need to read every page.*

The purpose of the proofs is to show that all the claims we make are verified both mathematically and by computer: it is notoriously easy to make "small errors" with this material. However we also feel that the proof methods, some of which are pure inductions, some of which show how certain theorems can be proven by using instances of others, and some which use an "equational binary calculus", are intriguing in their own right.

In Section 2 we give the basic mathematical machinery that underpins our theorems. There is a quick review of unsigned and signed binary. We then present the fundamental theorem that describes the binary/denary conversions used in the paper, and some simple formulas that arise from these conversions.

In Section 3 we formalise the Digitwize Algorithm for computing columnar addition of binary numbers. We then formulate a variety of operations, which formalise the maths/ALU operators given in the introduction, all defined using the Digitwize Algorithm.

Sections 4 and 5 lie at the heart of the paper. In Section 4.1 we outline the *combinations* of unsigned binary, signed binary and the various digitwise operations we will study. In Section 4.2 we present total correctness theorems for ALU-operators, and in Section 4.3 and 4.4 our conditional correctness theorems. In Sections 5.1 and 5.2 we move to Maths-operator theorems, again looking at total and conditional correctness.

In Section 6 we present some elementary group-theoretic theorems. For example, the group operation on k-digit binary numbers that yields an isomorphism with $(\mathbb{Z}_{2^k}, +, 0)$ is *precisely* ALU addition. While such a result is folklore, we are not aware of such a precise treatment in the literature.

2 BASIC DEFINITIONS AND THE FUNDAMENTAL THEOREM

Definition 2.1. Recall that $\mathbb{B} \stackrel{\text{def}}{=} \{0, 1\}$ is the set of Booleans (Boolean **digits**). Let $x \in \mathbb{B}^{k+1}$. We write $x_i \in \mathbb{B}$ for the digit of x at position i where $0 \leq i \leq k$ and sometimes

write $x_k x_{k-1} \ldots x_0$ instead of x. We write x^i for the sequence of digits $x_i x_{i-1} \ldots x_0$ where $0 \leq i \leq k$; so $x = x^k = x_k x^{k-1}$ and so on. If $d \in \mathbb{B}$ then we will write dx for the sequence $dx_k \ldots x_0$, making d the head of the sequence. We write $\overline{d}$ for the complement of d. We shall write $\overline{} : \mathbb{B}^{k+1} \to \mathbb{B}^{k+1}$ for the digitwise complement function, that is, $\overline{x} \stackrel{\text{def}}{=} \overline{x_k x_{k-1}} \ldots \overline{x_0}$.

We will write $[\![-]\!] : \mathbb{B}^{k+1} \to \mathbb{N} \subset \mathbb{Z}$ for the **denotational** function

$$x \mapsto [\![x]\!] \equiv [\![x_k x_{k-1} \ldots x_0]\!] \stackrel{\text{def}}{=} +x_k 2^k + \sum_{i=0}^{k-1} x_i 2^i$$

We will write $(\!|-|\!) : \mathbb{B}^{k+1} \to \mathbb{Z}$ for the **denotational** function

$$x \mapsto (\!|x|\!) \equiv (\!|x_k \ldots x_0|\!) \stackrel{\text{def}}{=} -x_k 2^k + \sum_{i=0}^{k-1} x_i 2^i = -x_k 2^k + [\![x^{k-1}]\!]$$

Given $x \in \mathbb{B}^{k+1}$ we call x_k the **sign-digit**.

We will make use of the functions

$$\sigma : \mathbb{B}^{k+1} \to \mathbb{B} \qquad\qquad \pi_\sigma : \mathbb{B}^{k+1} \to \mathbb{B}^{k+1}$$

(strictly, there is one such function for each k) which *return* the sign digit, that is $\sigma_x \stackrel{\text{def}}{=} x_k$; and *swap* the sign digit, that is $\pi_\sigma x \stackrel{\text{def}}{=} \overline{x_k} x_{k-1} \ldots x_0$.

The sign digit indicates whether the integer $(\!|x|\!)$ is negative or non-negative. To be precise, one traditionally proves

Lemma 2.1. *Given $x \in \mathbb{B}^{k+1}$ then $(\!|x|\!)$ is negative just in case $\sigma_x = 1$.*

However, more generally sign digits indicate ranges of semantic values. The next Lemma generalises Lemma 2.1 and is used regularly in this paper.

Lemma 2.2. *Given $x \in \mathbb{B}^{k+1}$ then we have*

$$0 \leq [\![x]\!] \leq 2^{k+1} - 1 \quad \textit{where} \quad \underbrace{0 \leq [\![x]\!] \leq 2^k - 1}_{\textit{if } \sigma_x = 0} \quad \textit{or} \quad \underbrace{2^k \leq [\![x]\!] \leq 2^{k+1} - 1}_{\textit{if } \sigma_x = 1}$$

and

$$-2^k \leq (\!|x|\!) \leq 2^k - 1 \quad \textit{where} \quad \underbrace{-2^k \leq (\!|x|\!) \leq -1}_{\textit{if } \sigma_x = 1} \quad \textit{or} \quad \underbrace{0 \leq (\!|x|\!) \leq 2^k - 1}_{\textit{if } \sigma_x = 0}$$

There is a key theorem that states a number of useful properties of the denotational functions from Definition 2.1. In order to state it, we first set up some notation and some further definitions.

Definition 2.2. We will use the following abbreviations for some simple ideas from Abelian groups:

- $Z^{u;m} \stackrel{\text{def}}{=} \{\ z \in \mathbb{Z} \mid 0 \leq z \leq 2^{m+1} - 1\ \}$
- $Z^{s;m} \stackrel{\text{def}}{=} \{\ z \in \mathbb{Z} \mid -2^m \leq z \leq 2^m - 1\ \}$
- $\mathbb{Z}_m \stackrel{\text{def}}{=} \mathbb{Z}/m\mathbb{Z} = \{\ [z] \mid z \in \mathbb{Z}\ \}$ where for any $z \in \mathbb{Z}$ we have

$$[z] \stackrel{\text{def}}{=} \{\ z \pm \alpha m \mid \alpha \in \mathbb{N}\ \}$$

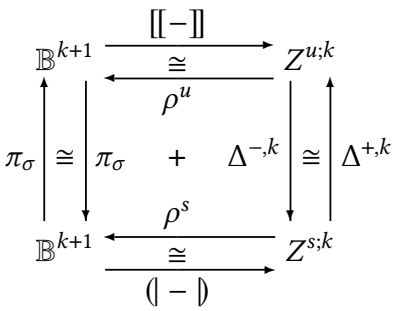
and we write the quotient map as $q : \mathbb{Z} \to \mathbb{Z}_m$ with $q(z) \stackrel{\text{def}}{=} [z]$. Recall that q is a homomorphism of Abelian groups, that is $q(0) = [0]$ and $q(z + z') = q(z) + q(z')$.

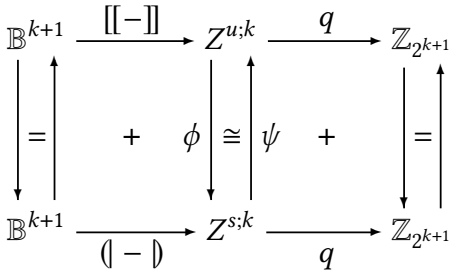
The functions $\rho^u : Z^{u;k} \to \mathbb{B}^{k+1}$ and $\rho^s : Z^{s;k} \to \mathbb{B}^{k+1}$, deliver unsigned binary and signed binary (twos-complement) representations (the formal definitions, and proofs of existence, appear in the Appendix on page 34. The proofs make use of basic number theory and Abelian groups).

We define the following functions for any $x \in \mathbb{B}^{k+1}$ and $z \in \mathbb{Z}$

$$\begin{array}{llll} \Delta^{-,k}(z) & \stackrel{\text{def}}{=} & z - 2^k & \phi(z) \stackrel{\text{def}}{=} \begin{array}{l} 0 \leq z \leq 2^k - 1 \mapsto z \\ 2^k \leq z \leq 2^{k+1} - 1 \mapsto \Delta^{-,k+1}(z) \end{array} \\ \Delta^{+,k}(z) & \stackrel{\text{def}}{=} & z + 2^k & \psi(z) \stackrel{\text{def}}{=} \begin{array}{l} 0 \leq z \leq 2^k - 1 \mapsto z \\ -2^k \leq z \leq -1 \mapsto \Delta^{+,k+1}(z) \end{array} \end{array}$$

whose source and target and relationships to the denotational functions are shown in the (commutative) diagrams below





Given $x \in \mathbb{B}^{k+1}$ we can also define $zx(x) \stackrel{\text{def}}{=} \vec{0}x \in \mathbb{B}^{k+1+k'}$ and $sx(x) \stackrel{\text{def}}{=} \vec{\sigma_x}x \in \mathbb{B}^{k+1+k'}$ by appending k' instances of 0 or the sign-digit σ_x.

The following theorem (see [4] or [16]) underpins this paper. We are not aware of the stated (folklore) properties appearing in the literature in such an organised way.

THEOREM 2.3. ***[Fundamentals of Representations and Semantics]***

The denotational functions $[\![-]\!] : \mathbb{B}^{k+1} \to \mathbb{Z}$ *and* $(\!|-|\!) : \mathbb{B}^{k+1} \to \mathbb{Z}$ *have images* $Z^{u;k}$ *and* $Z^{s;k}$*; in fact they are bijections onto these images and the functions* $\rho^u : Z^{u;k} \to \mathbb{B}^{k+1}$ *and* $\rho^s : Z^{s;k} \to \mathbb{B}^{k+1}$ *delivering unsigned binary and signed binary representations are their respective inverses.*

The commutation of the diagrams in Definition 2.2 amounts to the following equational properties:

- $[\![\pi_\sigma x]\!] = \Delta^{+,k}(\!|x|\!) = (\!|x|\!) + 2^k$
- $(\!|\pi_\sigma x|\!) = \Delta^{-,k}[\![x]\!] = [\![x]\!] - 2^k$
- $\phi([\![x]\!]) = (\!|x|\!)$
- $\psi((\!|x|\!)) = [\![x]\!]$
- $q(z \in Z^{u;k}) = q(\phi(z))$
- $q(z \in Z^{s;k}) = q(\psi(z))$

The following additional properties hold:

- $[\![x]\!] = (\!|x|\!) + \sigma_x 2^{k+1}$
- $[\![1\pi_\sigma x]\!] = [\![x]\!] + 2^k$ *when* $\sigma_x = 1$
- $(\!|1\pi_\sigma x|\!) = (\!|x|\!) - 2^k$ *when* $\sigma_x = 1$
- $$[\![\pi_\sigma x]\!] = \begin{cases} \sigma_x = 0 : & = & [\![x]\!] + 2^k \\ \sigma_x = 1 : & = & [\![x]\!] - 2^k \end{cases}$$
- $$(\!|\pi_\sigma x|\!) = \begin{cases} \sigma_x = 0 : & = & (\!|x|\!) - 2^k \\ \sigma_x = 1 : & = & (\!|x|\!) + 2^k \end{cases}$$
- *The functions* π_σ, *and* $\overline{\ } : \mathbb{B}^{k+1} \to \mathbb{B}^{k+1}$ *are self-inverse bijections and moreover* $\pi_\sigma \overline{x} = \overline{\pi_\sigma x}$.

These operations preserve semantics:
Suppose that $x \in \mathbb{B}^{k+1}$ *and* $y \in \mathbb{B}^{k+1+k'}$*. Then*

- $[\![x]\!] = [\![y]\!] \iff y = zx(x)$
- $(\!|x|\!) = (\!|y|\!) \iff y = sx(x)$

and in particular, any zero extension or sign extension of x *leaves the semantics unchanged.*

3 THE DIGITWIZE ALGORITHM AND DERIVED OPERATORS

3.1 The Digitwize Algorithm

We define the Digitwize Algorithm that formalises operations on binary numbers by computing with columns of digits and outputting remainders and carries.

Definition 3.1. Suppose that $a, b \in \mathbb{B}^{k+1}$ and so $a = a_k a_{k-1} \dots a_0$ and $b = b_k b_{k-1} \dots b_0$. A sequence of binary digits c_i for $0 \leq i \leq k+1$ and r_i for $0 \leq i \leq k$ is calculated by the **Digitwize Algorithm** as follows:

- $c_0 \in \mathbb{B}$ is chosen
- $r_i \stackrel{\text{def}}{=} (a_i + b_i + c_i)\ MOD\ 2$
- $c_{i+1} \stackrel{\text{def}}{=} (a_i + b_i + c_i)\ DIV\ 2$

These definitions yield a map of the form

$$DA_{c_0} : \mathbb{B}^{k+1} \times \mathbb{B}^{k+1} \longrightarrow \mathbb{B}^{k+2} \times \mathbb{B}^{k+1} \qquad (a, b) \mapsto (c, r)$$

The digit c_{k+2} is always referred to as the **sign digit** generated by a and b. (Recall also the sign-digit function σ, so in fact $c_{k+2} = \sigma_c$.)

Remark 3.1. *If the ALU* internal implementation *mimics exactly the DA, it is said to be a* ***ripple-carry adder****. We do not study implementations for efficiency such as carry lookahead, carry saving, carry selection, carry skipping and so on.*

3.2 Formalising the Operations

We now define a number of unary and binary operators $\Box$, on $a \in \mathbb{B}^{k+1}$ and $b \in \mathbb{B}^{k+1}$. They are all derived from the DA; see Definition 3.1.

Definition 3.2. Operators $\Box$ are generated from the following very simple grammar

$$\beta ::= Z \mid S \qquad \alpha ::= A \mid M \qquad \text{op} ::= + \mid - \mid \neg \mid abs \qquad \Box ::= \text{op}_\alpha \mid \text{op}_{\beta\alpha}$$

Their definitions are given in Figure 1. We refer to operators of the form op_A and $\text{op}_{\beta A}$ as **ALU-Operators** and these have target sets $\mathbb{B}^{k+1}$ and $\mathbb{B}^{k+2}$ respectively. We refer to those of the form op_M and $\text{op}_{\beta M}$ as **Maths-Operators** and these have target sets $\mathbb{B}^{k+2}$ and $\mathbb{B}^{k+3}$ respectively.

$$a \in \mathbb{B}^{k+1} \qquad b \in \mathbb{B}^{k+1}$$

$$\mathrm{op}_A\ (b) \quad a\ \mathrm{op}_A\ b \in \mathbb{B}^{k+1} \qquad \mathrm{op}_M\ (b) \quad a\ \mathrm{op}_M\ b \in \mathbb{B}^{k+2}$$

$$\mathrm{op}_{\alpha A}\ (b) \quad a\ \mathrm{op}_{\alpha A}\ b \in \mathbb{B}^{k+2} \qquad \mathrm{op}_{\alpha M}\ (b) \quad a\ \mathrm{op}_{\alpha M}\ b \in \mathbb{B}^{k+3}$$

$$\begin{array}{rcll}
a +_A b & \stackrel{\mathrm{def}}{=} & r & \text{where} \quad (c, r) \stackrel{\mathrm{def}}{=} DA_0(a, b) \\
a +_{ZA} b & \stackrel{\mathrm{def}}{=} & r & \text{where} \quad (c, r) \stackrel{\mathrm{def}}{=} DA_0(zx(a), zx(b)) \\
a +_{SA} b & \stackrel{\mathrm{def}}{=} & r & \text{where} \quad (c, r) \stackrel{\mathrm{def}}{=} DA_0(sx(a), sx(b)) \\
a +_M b & \stackrel{\mathrm{def}}{=} & c_{k+1} r & \text{where} \quad (c, r) \stackrel{\mathrm{def}}{=} DA_0(a, b) \\
a +_{ZM} b & \stackrel{\mathrm{def}}{=} & c_{k+1} r & \text{where} \quad (c, r) \stackrel{\mathrm{def}}{=} DA_0(zx(a), zx(b)) \\
a +_{SM} b & \stackrel{\mathrm{def}}{=} & c_{k+1} r & \text{where} \quad (c, r) \stackrel{\mathrm{def}}{=} DA_0(sx(a), sx(b)) \\
a -_A b & \stackrel{\mathrm{def}}{=} & r & \text{where} \quad (c, r) \stackrel{\mathrm{def}}{=} DA_1(a, \overline{b}) \\
a -_{ZA} b & \stackrel{\mathrm{def}}{=} & r & \text{where} \quad (c, r) \stackrel{\mathrm{def}}{=} DA_1(zx(a), \overline{zx(b)}) \\
a -_{SA} b & \stackrel{\mathrm{def}}{=} & r & \text{where} \quad (c, r) \stackrel{\mathrm{def}}{=} DA_1(sx(a), \overline{sx(b)}) \\
a -_M b & \stackrel{\mathrm{def}}{=} & c_{k+1} r & \text{where} \quad (c, r) \stackrel{\mathrm{def}}{=} DA_1(a, \overline{b}) \\
a -_{ZM} b & \stackrel{\mathrm{def}}{=} & c_{k+1} r & \text{where} \quad (c, r) \stackrel{\mathrm{def}}{=} DA_1(zx(a), \overline{zx(b)}) \\
a -_{SM} b & \stackrel{\mathrm{def}}{=} & c_{k+1} r & \text{where} \quad (c, r) \stackrel{\mathrm{def}}{=} DA_1(sx(a), \overline{sx(b)}) \\
\neg_A(b) & \stackrel{\mathrm{def}}{=} & \overline{b} +_A \mathbb{1} & \\
\neg_{SA}(b) & \stackrel{\mathrm{def}}{=} & \overline{b} +_{SA} \mathbb{1} & \\
\neg_M(b) & \stackrel{\mathrm{def}}{=} & \overline{b} +_M \mathbb{1} & \mathbb{1} \stackrel{\mathrm{def}}{=} \underbrace{0 \ldots\ldots 01}_{k \text{ copies of } 0} \in \mathbb{B}^{k+1}
\end{array}$$

$$\mathrm{abs}_A(b) \stackrel{\mathrm{def}}{=} \text{if } \sigma_b = 0 \text{ then } b \text{ else } \neg_A(b)$$

$$\mathrm{abs}_{SA}(b) \stackrel{\mathrm{def}}{=} \text{if } \sigma_b = 0 \text{ then } zx(b) \text{ else } \neg_{SA}(b)$$

Fig. 1. Defining Operations on Binary Numbers using the DA

Examples 3.1. Let $a = 0111$ and $b = 1001$. Then $(c, r) = DA_1(sx(0111), \overline{sx(1001)}) = DA_1(00111, 00110)$. Then we have, where $c_0 = \mathbf{1}$,

			0	0	1	1	1
		DA	0	0	1	1	0
c	=	0	1	1	1	1	**1**
r	=		0	1	1	1	0

It may help to recall Section 1.2 including Remark 1.3. Let's explore the remark a little more, in the cases where $\square$ is op_A and op_M.

In the Maths-operator case $a\ \mathrm{op}_M\ b$, the output is defined to be $c_{k+1}r \in \mathbb{B}^{k+2}$ (where $(c,r) \stackrel{\text{def}}{=} DA_{\delta\in\mathbb{B}}(a,b)$ is the output from an instance of the DA) thus including the overflow carry digit in position $k+1$: this corresponds to what (usually) happens when **M**athematicians are using a pencil and paper for computing binary sums.

In the ALU-operator case $a\ \mathrm{op}_A\ b$, the output is defined to be $r \in \mathbb{B}^{k+1}$. This mimics the **A**LU of a computer, namely that the overflow digit c_{k+1} is not included in $a\ \mathrm{op}_A\ b$. Any ALU is of a fixed size, e.g. where $k+1 = 64$-bit. The inputs (operands a, b) and output $a\ \mathrm{op}_A\ b$ are of equal size corresponding to the maximum size of the memory words in the computer. Engineers say that the computation is **heterogenous**.

There are also (derived) definitions of the form $\mathrm{op}_{Z\alpha}$ and $\mathrm{op}_{S\alpha}$. In the former case, the operands a, b are always zero-extended before applying the operation op_α; and in the latter case they are sign-extended before applying the operation op_α.

Remark 3.2. *Note there are two* primary *definitions of the form op_A and op_M, in that each $op_{\beta\alpha}$ can be defined in terms of op_α. For example, $a +_{ZA} b = zx(a) +_A zx(b) = 0a +_A 0b$.*

We now state a number of useful properties of the DA operations; *these will be used in the proofs of the correctness properties* that follow in the Appendix. The reader may wish to look back at Definitions 2.1 and 3.2 to revise the notation used. The proposition does not play a direct role in the statements of our correctness theorems. However we feel that it is of interest in its own right, since it provides an *equational calculus* for reasoning about our binary operators. The proof is on page 37.

Proposition 3.3. ***(A Calculus for Binary Operators)*** *Let $a, b \in \mathbb{B}^{k+1}$. Then the equations in Figure 2 hold.*

Examples 3.2. The first four statements look complicated but are trivial. They state that in any columnar computation, if the operands are extended to the left, then the columns to the right of the extension index are all equal. In denary, $89 + 46 = 1|35$ and $310|89 + 221|46 = 532|35$, where e.g $31089^1 = 89$. Note that while we are all very familiar with these ideas, formal statements and proofs are in the Appendix (see Lemma 8.5 and 8.6).

We give just one other example, for $a +_{ZA} b = \pi_\sigma(\pi_\sigma a +_{SA} \pi_\sigma b)$. We have $110 +_{ZA} 011 = 0110 +_A 0011 = 1001$ and also

$$\pi_\sigma(\pi_\sigma 110 +_{SA} \pi_\sigma 011) = \pi_\sigma(010 +_A 111) = \pi_\sigma(0010 +_A 1111) = \pi_\sigma 0001 = 1001.$$

- *If* $0 \leq i \leq k$ *then* $(a +_{ZA} b)^i = (a +_{SA} b)^i = (a +_A b)^i = a^i +_A b^i$
  - *and so* $(a +_{ZA} b)^k = (a +_{SA} b)^k = a +_A b$
- *If* $0 \leq i \leq k$ *then* $(a -_{ZA} b)^i = (a -_{SA} b)^i = (a -_A b)^i = a^i -_A b^i$
  - *and so* $(a -_{ZA} b)^k = (a -_{SA} b)^k = a -_A b$
- *If* $0 \leq i \leq k$ *then* $(a +_{ZM} b)^i = (a +_{SM} b)^i = (a +_A b)^i = a^i +_A b^i$
  - *and so* $(a +_{ZM} b)^k = (a +_{SM} b)^k = a +_A b$
- *If* $0 \leq i \leq k$ *then* $(a -_{ZM} b)^i = (a -_{SM} b)^i = (a -_A b)^i = a^i -_A b^i$
  - *and so* $(a -_{ZM} b)^k = (a -_{SM} b)^k = a -_A b$
- $a +_A b = \pi_\sigma a +_A \pi_\sigma b$
- $a -_A b = \pi_\sigma a -_A \pi_\sigma b$
- $\pi_\sigma a +_A b = a +_A \pi_\sigma b = \pi_\sigma(a +_A b)$
- $\pi_\sigma a -_A b = a -_A \pi_\sigma b = \pi_\sigma(a -_A b)$
- $a +_{ZA} b = \pi_\sigma(\pi_\sigma a +_{SA} \pi_\sigma b)$
- $a -_{ZA} b = \pi_\sigma a -_{SA} \pi_\sigma b$
- $a +_{ZA} b = zx(a +_A b) \iff$ *Condition Set 1 of Theorem 4.6 is valid.*
- $a -_{ZA} b = sx(a -_A b) \iff$ *Condition Set 1 of Theorem 4.8 is valid.*
- $a +_{SA} b = sx(a +_A b) \iff$ *Condition Set 1 of Theorem 4.9 is valid.*
- $a -_{SA} b = sx(a -_A b) \iff$ *Condition Set 1 of Theorem 4.10 is valid.*
- $a +_A b = \pi_\sigma a +_A \pi_\sigma b$
- $a +_M b = \pi_\sigma(\pi_\sigma a +_M \pi_\sigma b) \iff \sigma_a = \sigma_b$
- $a +_M b = \pi_\sigma a +_M \pi_\sigma b \iff \sigma_a \neq \sigma_b$
- $a -_M b = \pi_\sigma(\pi_\sigma a -_M \pi_\sigma b) \iff \sigma_a \neq \sigma_b$
- $a -_M b = \pi_\sigma a -_M \pi_\sigma b \iff \sigma_a = \sigma_b$
- $a +_{ZM} b = a +_{SM} b \iff \sigma_a = \sigma_b = 0$
- $a +_{ZM} b = \pi_\sigma(a +_{SM} b) \iff \sigma_a = \sigma_b = 1$
- $a +_M b = a +_{ZA} b$
- $a +_{ZM} b = zx(a +_M b)$
- $a +_{SM} b = sx(a +_M b) \iff$ *Conditions of Theorem 5.2 is valid* $\iff \sigma_a = \sigma_b$

Fig. 2. Defining a Calculus of Equations for Operations

4 CORRECTNESS THEOREMS FOR ALU-OPERATORS

4.1 Combinations of Semantics and Operators for Correctness Equations

We will shortly present a number of correctness theorems. Having loosely discussed the ideas in the Introduction, let us motivate a typical formal theorem. First a definition.

Definition 4.1. In the remainder of the paper we will consider the validity of correctness equations $Eq(a, b)$ involving $a, b \in \mathbb{B}^{k+1}$. Sometimes the equations will only be true when certain equivalent conditions $\Phi(a, b)$ hold.

- If we can prove $\forall a, b.\quad Eq(a, b)$ then $Eq(a, b)$ is called a **total** equation.
- If we can prove $\forall a, b.\quad \Phi(a, b) \Longleftrightarrow Eq(a, b)$ then $Eq(a, b)$ is called a **conditional** equation.

Definition 4.2. The **correctness** equations we study in this paper take the form

$$\begin{array}{llll} [\![a]\!] + [\![b]\!] =_{\mathbb{N}} [\![a \square b]\!] & (\!|a|\!) + (\!|b|\!) =_{\mathbb{Z}} (\!|a \square b|\!) & -[\![b]\!] =_{\mathbb{Z}} (\!|\square(b)|\!) \\ [\![a]\!] - [\![b]\!] =_{\mathbb{Z}} (\!|a \square b|\!) & (\!|a|\!) - (\!|b|\!) =_{\mathbb{Z}} (\!|a \square b|\!) & -(\!|b|\!) =_{\mathbb{Z}} (\!|\square(b)|\!) \end{array}$$

Note that there are other possibilities. For example one might (weirdly?) consider $[\![a]\!] + (\!|b|\!) =_{\mathbb{Z}} [\![a \square b]\!]$. The equations in the Definition seem to encompass a "sensible" blend of operators and semantics. Further refinements can be made by considering the instances of $\square$. For example, for the equations over $\mathbb{N}$, we will consider $\square \in \{ +_A, +_M, +_{ZA}, +_{ZM} \}$ but not $\square \in \{ +_{SA}, +_{SM} \}$.

The reader may care to re-read Section 1.5 before continuing.

4.2 Total Correctness Theorems for $\mathbf{op}_{\beta A}$

Theorem 4.3. *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) $(\!|a|\!) - (\!|b|\!) = (\!|a -_{SA} b|\!)$

VERIFIED IN HOL

Proof. One *must* prove Equations 1 and 2 by induction on k. One can then show that $(1 \Longleftrightarrow 3)$ and $(2 \Longleftrightarrow 4)$ by direct calculation. However, interestingly, we show that by using the calculus of Proposition 3.3, combined with semantic equations from Theorem 2.3, (3) and (4) can be deduced from (1) and (2) very easily. □

Examples 4.1. Take $a = 1011$ and $b = 0101$ and consider Equation 2. We should have in denary $(+11) - (+5) = +6$. Now $1011 -_{ZA} 0101 = r$ where

$$(c, r) = DA_1(zx(1011), \overline{zx(0101)}) = DA_1(01011, \overline{00101}) = DA_1(01011, 11010)$$

and then with $c_0 \stackrel{\text{def}}{=} \mathbf{1}$

			0	1	0	1	1
		DA	1	1	0	1	0
c	=	1	1	0	1	1	**1**
r	=		0	0	1	1	0

and of course $(\!|00110|\!) = +6$.

4.3 Conditional Correctness Theorems for $\mathbf{op}_{\beta A}$

Each of our ALU-operator conditional correctness theorems consist of five equivalent conditions. To state the conditions we need further definitions of new notation.

Definition 4.4. Suppose that $a, b \in \mathbb{B}^{k+1}$ and (c, r) is defined by the Digitwize Algorithm.

- We call any triple of binary digits a **Sign-Occurrence** (for example $\langle 1, 0, 0\rangle$). These are used to specify the possible values of $\langle \sigma_a, \sigma_b, \sigma_r\rangle$ which lead to correctness.
- We refer to σ_c as the **Overflow** digit.
- We refer to equations of the form

$$\text{op } \sigma_a \text{ op } \sigma_b \text{ op } \sigma_r = e \qquad \text{where } e \in \{ -2, \ldots, +2 \} \subset \mathbb{Z}$$

as **Sign-Equations**.

Definition 4.5. Suppose that $a, b \in \mathbb{B}^{k+1}$. The five conditions we consider in each of our correctness theorems are described below. In reference to Definition 4.1, the first condition $\Phi(a, b)$ is a *conditional equation* and the other four conditions of the general form $\Phi(a, b, c, r)$ are equivalent to equation validity.

(1) **Correctness:** A **correctness** equation, introduced and motivated in Section 1.3 and defined in Definition 1.2 and Definition 4.2.
(2) **Range:** The correct natural number sum lies within the range of integers that have representations with $k + 1$ digits.
(3) **Sign Occurrences:** This specifies values of $\langle \sigma_a, \sigma_b, \sigma_r\rangle$ as a *disjunction* of sign occurrences.
(4) **Overflow Digit:** This states that the overflow digit σ_c must equal a specified value; the value is usually given in terms of r and or c.
(5) **Sign Equations:** This is a single sign equation op σ_a op σ_b op $\sigma_r = e$ with the value of e specified.

The reader may wonder how these conditions were discovered in the first place. Some hints are encapsulated in the following remarks, in the context of Theorem 4.6:

Remark 4.1. *Suppose that $a, b \in \mathbb{B}^{k+1}$, with c and r defined in Figure 1.*

(1) ***Correctness:*** *See Section 1.3 and Definition 1.2.*

(2) ***Range:*** *The **Range** conditions come from thinking about the bounds for the left hand side and the right hand side of a **Correctness** equation. For example, in any theorem, the least bound (greatest bound) occuring across the **Range** conditions is always computed by minimizing (maximizing) the the correct integer answer. The intermediate bounds arise from minimizing (maximizing) the computer integer answer).*

(3) ***Sign Occurrences:*** *These are simply a complete solution set for the **Sign Equation** (see below). It follows from elementary algebra that the complete solution set determines the **Sign Equation** and vice versa.*

(4) ***Overflow Digit:*** *By using the Total Correctness equations of Theorem 4.3, from a correct integer answer (such as $[[a]] + [[b]]$) we can work out its binary representation $dx \in \mathbb{B}^{k+2}$. Using Proposition 3.3 one can then show that $x = r$, so in particular we know σ_r. From d and σ_r we can compute σ_c.*

(5) ***Sign Equations:*** *Starting (as above) from*

$$[[a]] + [[b]] = [[dr]] \qquad \equiv \qquad [[\sigma_a a^{k-1}]] + [[\sigma_b b^{k-1}]] = [[d\sigma_r r^{k-1}]]$$

*we can expand the right hand equation using the definition of the semantics, and then the body of the **Sign Equation** appears naturally. If one then analyses the upper and lower bounds of the sum $[[a^{k-1}]] + [[b^{k-1}]]$, one can obtain integer values for e.*

Readers can find a more detailed analysis, in the context of Theorem 4.10, Appendix page 53.

Before we proceed, a couple of comments. The bold-text item labels in Theorem 4.6 will be referred to as **condition types**. We also refer to the **first set** of conditions, **second set**, and so on if there are further sets.

Remark 4.2. *We can adopt a standard proof strategy. In Theorem 4.6 (and those that follow), for each condition set bar the final one we shall prove*

- *$(3 \iff 4 \iff 5)$*
- *$(1 \implies 2 \implies 4 \implies 1)$ or $(1 \implies 2 \implies 3 \implies 1)$*

and then deduce the final condition set through some simple propositional logic.

THEOREM 4.6. *Let $a, b \in \mathbb{B}^{k+1}$. Then the following conditions are* pairwise equivalent

Condition Set 1

(1) ***Correctness:*** $[[a]] + [[b]] = [[a +_A b]]$
(2) ***Range:*** $0 \leq [[a]] + [[b]] \leq 2^{k+1} - 1$
(3) ***Sign Occurrences:*** $\langle \sigma_a, \sigma_b, \sigma_r \rangle$ *can be any one of the following four sign-occurrences* $\langle 0, 0, 0/1 \rangle$, $\langle 0, 1, 1 \rangle$, $\langle 1, 0, 1 \rangle$, *but no others.*
(4) ***Overflow Digit:*** $\sigma_c = 0$
(5) ***Sign Equation:*** $\sigma_a + \sigma_b - \sigma_r = 0, -1$

and the following conditions are pairwise equivalent

Condition Set 2

(1) ***Correctness:*** $[[a]] + [[b]] = [[1(a +_A b)]]$
(2) ***Range:*** $2^{k+1} \leq [[a]] + [[b]] \leq 2 * (2^{k+1} - 1)$
(3) ***Sign Occurrences:*** $\langle \sigma_a, \sigma_b, \sigma_r \rangle$ *can be any one of the following four sign-occurrences* $\langle 0, 1, 0 \rangle$, $\langle 1, 0, 0 \rangle$, $\langle 1, 1, 0/1 \rangle$, *but no others.*
(4) ***Overflow Digit:*** $\sigma_c = 1$
(5) ***Sign Equation:*** $\sigma_a + \sigma_b - \sigma_r = 2, 1$

and moreover for each ***condition type*** *(such as* ***Correctness****), one instance must hold (along with its condition set equivalents).* VERIFIED IN HOL

PROOF. The Appendix proof follows Remark 4.2. To deduce **Condition Set 2** from **Condition Set 1**, the basic thought is that each condition in the second set, bar **Correctness**, is the logical negation of its counterpart in the first set. And **Correctness** itself follows by consulting Theorem 4.3, and then using Proposition 3.3 along with **Overflow Digit** to perform simple rewrites. □

Example 4.7. Let $a = 11$ and $b = 10$ with $k = 1$. So $[[11]] = 3$ and $[[10]] = 2$ and $11 +_{ZA} 10 = 101$ with carry 100. Correctness requires $3 + 2 \equiv 5 = [[101]]$. The range requires $2^2 \leq 5 \leq 2 * (2^2 - 1)$. The sign occurrence is $\langle 1, 1, 0 \rangle$. The overflow digit σ_c is indeed 1. And finally the sign equation is $1 + 1 - 0 = 2$. So all the conditions are true.

REMARK 4.3. *WARNING ABOUT TERMINOLOGY: The (common) use of the phrase "overflow occurs", in this paper, must now be applied in the context of a Correctness Equation (or equivalent condition) in a* **SPECIFIED** *Condition Set. Asserting "overflow occurs" means all conditions of the* **SPECIFIED** *set fail to hold true.*

THEOREM 4.8. *Let $a, b \in \mathbb{B}^{k+1}$. Then the following conditions are pairwise* pairwise equivalent

Condition Set 1

(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:*** $\sigma_c \neq \sigma_r$.
(5) ***Sign Equation:*** $-\sigma_a + \sigma_b - \sigma_r = 0, -1$

and the following conditions are pairwise equivalent

Condition Set 2

(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:*** $\sigma_c = \sigma_r = 0$.
(5) ***Sign Equation:*** $-\sigma_a + \sigma_b - \sigma_r = 1$

and the following conditions are pairwise equivalent

Condition Set 3

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

and moreover for each ***condition type*** *one instance must hold (along with equivalents of the instance).*

VERIFIED IN HOL

PROOF. The Appendix proof follows Remark 4.2. To deduce **Condition Set 3** from **Condition Set 1 and 2**: note that for any type of condition, bar **Correctness**, the conjunction of all three matching conditions, one from each set, must be true, and moreover the conditions are mutually exclusive. And **Correctness** itself again follows by consulting Theorem 4.3, and then using Proposition 3.3 along with **Overflow Digit** to perform rewrites. □

Remark 4.4. *Notice that in the second and third condition sets, the correctness equation has a conjunct stipulating the value of the sign-digit of the result. Conveniently, the first condition set which contains the "fundamental" statement of correctness, does not. However, when overflow occurs, the alternative equations that do hold may only be determined exclusively by knowing about the value of σ_r! To see why this is so, suppose that $[[a]] - [[b]] = (\!|1(a -_A b)|\!)$ and that $\sigma_r \equiv \sigma_{a-_A b} = 1$. Then we would have $(\!|1(a -_A b)|\!) = (\!|a -_A b|\!)$ and so we land back in Condition Set 1. So additional information is required to distinguish the correctness equations in the second and third condition sets, and explicitly flagging σ_r suffices.*

Theorem 4.9. *Let $a, b \in \mathbb{B}^{k+1}$. Then the following conditions are* pairwise equivalent

Condition Set 1

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

and the following conditions are pairwise equivalent

Condition Set 2

(1) ***Correctness:*** $(\!|a|\!) + (\!|b|\!) = (\!|1(a +_A b)|\!) \wedge \sigma_r = 0$
(2) ***Range:*** $-2^{k+1} \leq (\!|a|\!) + (\!|b|\!) \leq -2^k - 1$
(3) ***Sign Occurrences:*** *The sign-occurrence $\langle 1,1,0\rangle$.*
(4) ***Overflow Digit:*** $\sigma_c = 1$ and $c_k = 0$.
(5) ***Sign Equation:*** $-\sigma_a - \sigma_b + \sigma_r = -2$

and the following conditions are pairwise equivalent

Condition Set 3

(1) ***Correctness:*** $(\!|a|\!) + (\!|b|\!) = (\!|0(a +_A b)|\!) \wedge \sigma_r = 1$
(2) ***Range:*** $2^k \leq (\!|a|\!) + (\!|b|\!) \leq 2(2^k - 1)$
(3) ***Sign Occurrences:*** *The sign-occurrence$\langle 0,0,1\rangle$.*
(4) ***Overflow Digit:*** $\sigma_c = 0$ and $c_k = 1$.
(5) ***Sign Equation:*** $-\sigma_a - \sigma_b + \sigma_r = 1$

and moreover for each ***condition type*** *one instance must hold (along with equivalents of the instance).*

VERIFIED IN HOL

Remark 4.5. *Two proofs are given. One approach is the same as for the previous two theorems. However, another interesting method is to make use of Theorem 4.6 that involved unsigned binary, along with the semantic equations of Theorem 2.3 which give many useful relations between (denotations of) signed and unsigned binary along with the calculus of Proposition 3.3, and hence "translate" Theorem 4.6 into Theorem 4.8.*

Theorem 4.10. *Let $a, b \in \mathbb{B}^{k+1}$. Then the following conditions are pairwise* equivalent

Condition Set 1

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

and the following conditions are pairwise equivalent

Condition Set 2

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

and the following conditions are equivalent

Condition Set 3

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

and moreover for each ***condition type*** *one instance must hold (along with equivalents of the instance).*

VERIFIED IN HOL

Remark 4.6. *Please see Remark 8.3 on Appendix page 53 for the promised (in Remark 4.1) detailed analysis of how the Condition Set conditions can be derived from first principles.*

4.4 Conditional Correctness Theorems for $\neg_{SA}$ and $\neg_A$

Next we look at using some of the arithmetic operations to compute integer negations. In the work that appears below, instances of the binary number $1\vec{0}$ correspond to `Integer.MIN_VALUE` in the Introduction. The second part of the Theorem below shows formally that using ALU-operator negation will fail exceptionally on this input. See Remark 4.7.

THEOREM 4.11. *Let $a, b \in \mathbb{B}^{k+1}$. Recall from Figure 1 that $\neg_{SA}(b) \stackrel{\text{def}}{=} \overline{b} +_{SA} \mathbb{1}$ and $\neg_A(b) \stackrel{\text{def}}{=} \overline{b} +_A \mathbb{1}$. Then we have*

- $-(\!|b|\!) = (\!|\neg_{SA}(b)|\!)$
- $(\!|a -_{SA} b|\!) = (\!|\sigma_a a +_{SA} (\neg_{SA}(b))|\!)$
- $\sigma_{a-_{SA}b}(a -_{SA} b) = \sigma_a a +_{SA} (\neg_{SA}(b)) \in \mathbb{B}^{k+3}$ *and so* $a -_{SA} b = (\sigma_a a +_{SA} (\neg_{SA}(b)))^{k+2} \in \mathbb{B}^{k+2}$
- $abs(\!|b|\!) = (\!|\mathrm{abs}_{SA}(b)|\!)$

and

- $-(\!|b|\!) = (\!|\neg_A(b)|\!)$ *for* $b \neq 1\vec{0}$
- $(\!|a -_A b|\!) = (\!|a +_A (\neg_A(b))|\!)$
- $a -_A b = a +_A (\neg_A(b))$
- $abs(\!|b|\!) = (\!|\mathrm{abs}_{SA}(b)|\!)$ *for* $b \neq 1\vec{0}$

VERIFIED IN HOL

PROOF. We remark only that the proof of the theorem relies heavily on Theorem 4.10. □

REMARK 4.7. *From Theorem 4.11 we have $-(\!|b|\!) = (\!|\neg_A(b)|\!)$ in all cases bar $b = 1\vec{0}$. In a $(k+1)$-bit ALU, $1\vec{0}$ is precisely the signed binary representation of* `Integer.MIN_VALUE` : `int`*; and of course $\sigma_{1\vec{0}} = 1$. Hence we have*

$$\begin{aligned}
\texttt{Math.abs(Integer.MIN_VALUE)} &= [\![\mathrm{abs}_A(\rho^s(\texttt{Integer.MIN_VALUE}))]\!] \\
&= [\![\neg_A(\rho^s(\texttt{Integer.MIN_VALUE}))]\!] \\
&= [\![\overline{\rho^s(\texttt{Integer.MIN_VALUE})} +_A \mathbb{1}]\!] \\
&= [\![\overline{1\vec{0}} +_A \mathbb{1}]\!] \\
&= [\![1\vec{0}]\!] \\
&= \texttt{Integer.MIN_VALUE}
\end{aligned}$$

Remark 4.8. *In general* $(\!|a -_A b|\!) = (\!|a|\!) - (\!|b|\!)$ *just in case the conditions of Theorem 4.10 hold. However it is possible for this equality to hold, with the binary subtraction computed in an ALU by complementing* b *and adding 1, even though* $-(\!|b|\!) \neq (\!|\neg_A(b)|\!)$.

Indeed when $b = \vec{10}$ *we have* $\neg_A(b) = b$ *so that* $(\!|\neg_A(b)|\!) = (\!|b|\!)$. *However, taking* $a \stackrel{\text{def}}{=} 1100$ *with* $k = 3$, *and noting that* $a -_A b = 1100 -_A 1000 = 1100 +_A (\overline{1000} +_A 1) = 1100 +_A 1000 = 0100$,

$$(\!|a|\!) - (\!|b|\!) = -4 - (-8) = 4 = (\!|a -_A b|\!)$$

Theorem 4.12. *Let* $a, b, c \in \mathbb{B}^{k+1}$. $+_A$ *and* $+_{\alpha A}$ *are commutative.* $+_A$ *is associative. In particular, and further*

- $a +_A (b +_A c) = (a +_A b) +_A c$
- $0a +_{ZA} (b +_{ZA} c) = (a +_{ZA} b) +_{ZA} 0c$

VERIFIED IN HOL

5 CORRECTNESS THEOREMS FOR MATHS-OPERATORS

Now we move on to Maths-operator theorems.

5.1 Total Correctness Theorems for $\mathbf{op}_{\beta M}$ and $\mathbf{op}_M$

Theorem 5.1. *Let* $a, b \in \mathbb{B}^{k+1}$. *The following equations hold for every such* a *and* b *(total correctness)*

- $[\![a]\!] + [\![b]\!] = [\![a +_M b]\!]$
- $[\![a]\!] + [\![b]\!] = [\![a +_{ZM} b]\!]$

The following in-equations hold for any such a *and* b *(total incorrectness)*

- $[\![a]\!] - [\![b]\!] \neq (\!|a -_M b|\!)$
- $[\![a]\!] - [\![b]\!] \neq (\!|a -_{ZM} b|\!)$
- $[\![a]\!] - [\![b]\!] \neq (\!|a -_{SM} b|\!)$

5.2 Conditional Correctness Theorems for $\mathbf{op}_{\beta M}$ and $\mathbf{op}_M$

Theorem 5.2. *Let* $a, b \in \mathbb{B}^{k+1}$. *The following conditions are* pairwise equivalent

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

as are

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

and moreover one condition must hold (along with equivalents of the condition) from each set.

The following conditions are pairwise equivalent

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

as are

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

and moreover one condition must hold (along with equivalents of the condition) from each set.

5.3 Conditional Correctness Theorems for $\neg_M$

Theorem 5.3. *Let* $1, a, b \in \mathbb{B}^{k+1}$*. Then we have*

- $-(\!|b|\!) = (\!|\neg_M(b)|\!)$ *for* $\sigma_b = 1$.
- $-(\!|b|\!) = (\!|\pi_\sigma(\neg_M(b))|\!)$ *for* $\sigma_b = 0$.
- $(\!|a -_M b|\!) = (\!|(0a +_M (\neg_M(b)))^{k+2}|\!)$.
- $0(a -_M b) = 0a +_M (\neg_M(b))$ *and so* $a -_M b = (0a +_M (\neg_M(b)))^{k+2}$
- $abs((\!|b|\!)) = (\!|\mathrm{abs}_M(b)|\!)$

Proof. We remark that the proof of the theorem relies heavily on Theorem 5.2. Note also that the final item is a total correctness equation. This is the case since the first equation holds just in case $\sigma_b = 1$, and so the negation that is computed is correct. □

Theorem 5.4. *Let* $a, b, c \in \mathbb{B}^{k+1}$*.* $+_M$ *and* $+_{\alpha M}$ *are commutative. Further*

- $0a +_M (b +_M c) = (a +_M b) +_M 0c$
- $00a +_{ZM} (b +_{ZM} c) = (a +_{ZM} b) +_{ZM} 00c$

Proof. Commutativity is straightforward. The equations are easily proven by appealing to the total correctness equations for $+_M$ and $+_{ZM}$ from Theorem 5.1, and Theorem 2.3. Note that the first equation also follows from Theorem 4.12 since trivially $+_M = +_{ZA}$. □

6 RELATING $\mathbf{op}_A$ AND $\mathbf{op}_M$ AND MODULAR ARITHMETIC

We present some further theorems about the properties of the operators op_α.

Theorem 6.1. *$(\mathbb{B}^{k+1}, +_A, \vec{0})$ is a group and it is isomorphic to $(\mathbb{Z}_{2^{k+1}}, +, [0])$ where we write $[z]$ for modular equivalence classes but overload the $\mathbb{Z}$ addition operator $+$ on $\mathbb{Z}_{2^{k+1}}$. 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 $q^{-1}([z]) \overset{\text{def}}{=} z \ MOD \ 2^{k+1}$.

Proof. It is very easy to see that q^{-1} is well defined (we choose to calculate the positive modular value if $z < 0$), and that it is an inverse of q. The other (set) bijection was proven in Theorem 2.3. It follows that $q \circ [[-]] : \mathbb{B}^{k+1} \to \mathbb{Z}_{2^{k+1}}$ is a bijection too. So it remains only to prove that this composition is a homomorphism of groups. Thus we need to prove that

$$q([[a +_A b]]) = q([[a]]) +_{\mathbb{Z}_{2^{k+1}}} q([[b]])$$

Note that $q = q_{\mathbb{Z}} \circ i$ where $q_{\mathbb{Z}}$ is the quotient map with source $\mathbb{Z}$, and of course $q_{\mathbb{Z}}$ is a homomorphism of groups ($Z^{u;k}$ is certainly not a group!). So in fact we need to prove that

$$q_{\mathbb{Z}}([[a +_A b]]) = q_{\mathbb{Z}}([[a]] + [[b]])$$

But from the **Correctness** conditions of Theorem 4.6 we see immediately that $[[a]] + [[b]]$ equals either $[[a +_A b]]$ or $[[1(a +_A b)]] = 2^{k+1} + [[a +_A b]]$ and so we are done. □

Corollary 6.1. *It follows immediately from Theorem 6.1 and Theorem 4.6 that*

$$\mathbb{B}^{k+1} \xrightarrow{(|-|)} Z^{s;k} \xrightarrow{q} \mathbb{Z}_{2^{k+1}}$$

is also an isomorphism of groups. In particular notice that $q \circ [[-]] = q_{\mathbb{Z}} \circ [[-]]$ and $q \circ (|-|) = q_{\mathbb{Z}} \circ (|-|)$ where we mildly overload the notation for the semantic maps, regarding both targets as $\mathbb{Z}$.

Theorem 6.2. *Noting that* $op_A : \mathbb{B}^{k+1} \times \mathbb{B}^{k+1} \longrightarrow \mathbb{B}^{k+1}$ *and* $op_M : \mathbb{B}^{k+1} \times \mathbb{B}^{k+1} \longrightarrow \mathbb{B}^{k+2}$ *and letting* $F ::= q_{\mathbb{Z}} \circ [\![-]\!] \mid q_{\mathbb{Z}} \circ (\!|-|\!)$*, consider the diagram (where* $P(\alpha)$ *is* $k+1$ *or* $k+2$*)*

$$\begin{array}{ccc} \mathbb{B}^{k+1} \times \mathbb{B}^{k+1} & \xrightarrow{op_\alpha} & \mathbb{B}^{P(\alpha)} \\ \downarrow{\scriptstyle F \times F} & & \downarrow{\scriptstyle F} \\ \mathbb{Z}_{2^{k+1}} \times \mathbb{Z}_{2^{k+1}} & \xrightarrow[op]{} & \mathbb{Z}_{2^{k+1}} \end{array}$$

Of course $-_\alpha$ *and* $+_M$ *are not operations that yield group structures on* $\mathbb{B}^{k+1}$ *(for trivial reasons). However, in many cases, suitable* F *yield homomorphisms for the operations. In fact we have*

- $q_{\mathbb{Z}}[\![a +_A b]\!] = q_{\mathbb{Z}}[\![a]\!] + q_{\mathbb{Z}}[\![b]\!]$
- $q_{\mathbb{Z}}(\!|a -_A b|\!) = q_{\mathbb{Z}}[\![a]\!] - q_{\mathbb{Z}}[\![b]\!]$
- $q_{\mathbb{Z}}(\!|a +_A b|\!) = q_{\mathbb{Z}}(\!|a|\!) + q_{\mathbb{Z}}(\!|b|\!)$
- $q_{\mathbb{Z}}(\!|a -_A b|\!) = q_{\mathbb{Z}}(\!|a|\!) - q_{\mathbb{Z}}(\!|b|\!)$
- $q_{\mathbb{Z}}[\![a +_M b]\!] = q_{\mathbb{Z}}[\![a]\!] + q_{\mathbb{Z}}[\![b]\!]$
- $q_{\mathbb{Z}}(\!|a +_M b|\!) = q_{\mathbb{Z}}(\!|a|\!) + q_{\mathbb{Z}}(\!|b|\!)$
- $q_{\mathbb{Z}}(\!|a -_M b|\!) = q_{\mathbb{Z}}(\!|a|\!) - q_{\mathbb{Z}}(\!|b|\!)$

Proof. This follows immediately from the various correctness proofs, and since $q_{\mathbb{Z}}$ is a homomorphism of groups (and hence preserves both + and − in the Abelian group $\mathbb{Z}_{2^{k+1}}$). See the previous proof, of Theorem 6.1, plus one further illustrative example: for the final equation, we simply need to show that $q_{\mathbb{Z}}(\!|a -_M b|\!) = q_{\mathbb{Z}}(\!|a|\!) - q_{\mathbb{Z}}(\!|b|\!)$. However from Theorem 5.1 we know that $(\!|a|\!) - (\!|b|\!)$ is either $(\!|a -_M b|\!)$, or $(\!|\pi_\sigma(a -_M b)|\!)$ and so differs from $(\!|a -_M b|\!)$ by $\pm 2^{k+1}$ since the image set of $-_M$ is $\mathbb{B}^{k+2}$. □

Remark 6.1. *Recall Theorem 4.11 and Remark 4.7.* $\mathbb{Z}_2$ *acts on* $\mathbb{Z}_{2^{k+1}}$ *by computing negations. There are* 2^{k+1} *elements of* $\mathbb{Z}_{2^{k+1}}$ *(all of them!) fixed by the action of* $0 \in \mathbb{Z}^2$*.* $0 \in \mathbb{Z}_{2^{k+1}}$ *is fixed by* 1 *and so the number of points fixed by* 1 *is at least one. Burnside's Lemma shows that these sum to* $|\mathbb{Z}_2| * O$ *where* O *is the number of orbits. So the sum is even. Hence there must be* at least one other element of $\mathbb{Z}_{2^{k+1}}$ *fixed by* $1 \in \mathbb{Z}_2$*, that is, another element equal to it's own negation. And in* $\mathbb{B}^{k+1}$ *this is* $\ldots\vec{10}$*, i.e.* `Integer.MIN_VALUE`.

7 RELATED WORK

The authoritative book by Hennessy and Patterson [11] is a great introduction to computer processors. The comprehensive book by Ercegovac and Lang [8] contains a great deal of arithmetic theory, perhaps directed more towards applications, and of course a much wider range of operations and representations than presented in our paper. Other viewpoints can be found in the classic text [7] by Donald E. Knuth. The books and articles [2, 14, 17, 20, 21] study various aspects of computer Arithmetic in detail, though they are oriented towards engineering and application in general.

There are many many computer based formal verification systems, and certainly too many to be able to mention them all here. We explicitly mention two interactive Theorem Provers, and these are Isabelle [3] and HOl [1, 9]. The first reason is that they have had a major impact in the field of verification over the past few decades, and the second reason is that we have used Isabelle in our work and in particular Higher Order Logic within Isabelle. The book [19] by Nipkow and Klein is an excellent introduction to Isabelle.

An article whose results are closely related to the work presented here was written by Dawson [6]. The results are closely related at a foundational level, though not directly. Certainly our main Theorems do not appear in [6]. Dawson describes a collection of Isabelle theories which facilitate reasoning about machine words: in our work a binary number is effectively a machine word. For each possible word length, Dawson describes a type of words of that length, and then generic theorems which can be applied to any such type. From this, relationships are developed between such words and signed and unsigned integers, along with other types such as lists of booleans and certain kinds of functions. This work is centred around the Isabelle Theorem Prover and its development.

As far back as thirty years ago, perhaps more, the attractiveness of the generality of theorem provers for verification of system designs was understood. Combining this generality with good automation was much more difficult. Gradual progress has been made over the years. In this section we overview just a few of the many contributions, mainly in the verification of ALU hardware, and software/algorithms for arithmetic calculations. There are many others, and again these are too numerous to cite all of them.

Bit slicing was a technique that was common with early processors in the mid 1970s, especially the AMD (Advanced Micro Devices) 2900 series. One combines an ALU of a fixed size, such as 4-bits, to build a $4k$-bit processor. The overall ALU is said to be made up of k 4-bit slices, and arithmetic computations are undertaken in parallel. Hunt and Brock [22] verify a bit-slice ALU using a mechanized theorem prover. The ALU and

related specifications are written in the Boyer-Moore logic and the verification uses the Boyer-Moore theorem prover.

Cyrluk, Rajan, Shankar and Srivas [5] describe an approach for enhancing the effectiveness of theorem provers for hardware verification through the use of efficient automatic procedures for rewriting, arithmetic and equality reasoning, and an off-the-shelf BDD-based propo-sitional simplifier.

These automatic procedures are applied to various examples including an N-bit adder, the Saxe pipelined processor, and the benchmark Tamarack microprocessor design. These examples illustrate the basic design philosophy underlying PVS where powerful and efficient low-level inferences are employed within high-level user-defined proof strategies. This approach is contrasted with approaches based on tactics or batch-oriented theorem proving.

Not so many years later "microprocessors" (as they became known) begin to be used in many safety-critical environments, such as cars or planes. This is of course the case today, where safety critical systems are pervasive. Kroening [15] wrote a thesis at the turn of the millennium on this topic. Errors often remained undiscovered even when considerable testing was done. A well known example of this was the Pentium IV bug. This cost Intel a great deal of money; afterwards Intel were to invest much more heavily in using tools such as theorem provers to verify their processor designs.

John Harrison was employed by Intel soon after the Pentium IV incident. He has done considerable work in using theorem provers to verify hardware. This and similar work eventually led to [13] which is a unique and comprehensive monograph that covers techniques in formal and practical logic and automation. For a general non-technical overview of automation in mathematics see for example [10, 12].

There has been extensive work in verification of floating point arithmetic [2, 20, 21]. While pervasive in modern computing, due to restricted ranges and of course precision loss, floating point arithmetic can be complex and indeed lead to failure. For this latter reason, computer-assisted verification of correctness proofs is vital. Just one example of computer-assisted verification is the work by Daniel Kroening, Tom Melham et al have applied various tools to a commercial floating-point arithmetic unit from ARM and an open-source dual-path floating-point adder [18]. This work provides a comprehensive account of formally specified and verified floating-point algorithms with the Coq proof assistant. The authors provide a formalization of floating-point arithmetic and methods to automate proofs. They give the specification and verification of many algorithms, from

error-free transformations to a numerical scheme for a partial differential equation. The examples cover not only mathematical algorithms but also C programs as well as issues related to compilation.

Finally we would like to thank Simon Gay, Alexander Kurz, Andrew Pitts, and Emilio Tuosto for discussions about this work, and to John Harrsion for help with pointers to related work.

REFERENCES

[1] The HOL Theorem Prover. 2019. https://hol-theorem-prover.org/.

[2] Sylvie Boldo and Guillaume Melquiond. *Computer Arithmetic and Formal Proofs*. Elsevier, 2017.

[3] Clemens Ballarin and Stefan Berghofer and Jasmin Blanchette and Timothy Bourke and Lukas Bulwahn and Amine Chaieb and Lucas Dixon and Florian Haftmann and Brian Huffman and Lars Hupel and Gerwin Klein and Alexander Krauss and Ondřej Kunčar and Andreas Lochbihler and Tobias Nipkow and Lars Noschinski and David von Oheimb and Larry Paulson and Sebastian Skalberg and Christian Sternagel and Dmitriy Traytel. *The Isabelle/Isar Reference Manual*. 2019. https://isabelle.in.tum.de/doc/isar-ref.pdf.

[4] P.M. Cohn. *Algebra*, volume 2. John Wiley and Sons Ltd., 1979.

[5] Cyrluk D., Rajan S., Shankar N., and Srivas M.K. Effective Theorem Proving for Hardware Verification. *Lecture Notes in Computer Science*, 1995. ISBN: 978-3-540-59047-7.

[6] Jeremy E. Dawson. Isabelle Theories for Machine Words. 250:55–70, 2009. In Seventh International Workshop on Automated Verification of Critical Systems (AVOCS'07). Full version appears at.

[7] Donald E. Knuth. *The Art of Computer Programming: Fundamental Algorithms*, volume 1. Addison Wesley Longman Publishing Co., 3rd edition, 1997.

[8] Milos D. Ercegovac and Tomás Lang. *Digital Arithmetic*. Elsevier (Morgan Kaufmann), 2003.

[9] M. J. C. Gordon and T. F. Melham. *Introduction to HOL: A theorem proving environment for higher order logic*. Cambridge University Press, 1993.

[10] John Harrison. A HOL Theory of Euclidean Space. In *In Theorem Proving in Higher Order Logics*, volume 3603 of *Lecture Notes in Computer Science*, pages 114–129, 2005.

[11] John L. Hennessy and David A. Patterson. *Computer Architecture: A Quantitative Approach*. Elsevier (Morgan Kaufmann), 2017.

[12] Jeremy Avigad and John Harrison. Formally Verified Mathematics. *Communications of the ACM*, 57(4):66–75, 2014.

[13] John Harrison. *Handbook of Practical Logic and Automated Reasoning*. Cambridge University Press, 2009.

[14] Peter Kornerup and David W. Matula. *Finite Precision Number Systems and Arithmetic*. Cambridge University Press, 2010. Appeared online in 2013.

[15] Daniel Kroening. *Formal Verification of Pipelined Microprocessors*, pages 71–80. Gesellschaft fuer Informatik, 2002.

[16] S. Mac Lane. *Algebra*. American Mathematical Society, 3 edition, 1999.

[17] Silvia M. Mueller and Wolfgang J. Paul. *Computer Architecture Complexity and Correctness*. Springer, 2000.

[18] Rajdeep Mukherjee, Saurabh Joshi, Andreas Griesmayer, Daniel Kroening, and Tom Melham. Equivalence checking of a floating-point unit against a high-level C model. In *FM 2016: Formal Methods - 21st International Symposium, Limassol, Cyprus, November 9-11, 2016, Proceedings*, pages 551–558, 2016.

[19] Tobias Nipkow and Gerwin Klein. *Concrete Semantics*. 2019. http://concrete-semantics.org/.

[20] B. Parhami. *Computer Arithmetic: Algorithms and Hardware Designs*. Oxford University Press, 2010. 2nd Edition.

[21] David M. Russinoff. *Formal Verification of Floating-Point Hardware Design. A Mathematical Approach*. Springer, 2019.

[22] Hunt W.A. and Brock B.C. The verification of a bit-slice ALU. *Lecture Notes in Computer Science*, 408(2):103–112, 1990. Hardware Specification, Verification and Synthesis: Mathematical Aspects.

8 APPENDIX

8.1 Fundamental Theorem

Lemma 8.1. *Given $x \in \mathbb{B}^{k+1}$ then $(\!|x|\!)$ is negative just in case $\sigma_x = 1$.*

Proof. If $\sigma_x = 1$ then $(\!|x|\!) \leq -2^k + \Sigma_{i=0}^{k-1} 2^i = -2^k + (2^k - 1) = -1 < 0$. If $\sigma_x = 0$ it is trivial that $(\!|x|\!) \geq 0$. □

Lemma 8.2. *Given $x \in \mathbb{B}^{k+1}$ then we have*

$$0 \leq [\![x]\!] \leq 2^{k+1} - 1 \quad \textit{where} \quad \underbrace{0 \leq [\![x]\!] \leq 2^k - 1}_{\textit{if } \sigma_x=0} \quad \textit{or} \quad \underbrace{2^k \leq [\![x]\!] \leq 2^{k+1} - 1}_{\textit{if } \sigma_x=1}$$

and

$$-2^k \leq (\!|x|\!) \leq 2^k - 1 \quad \textit{where} \quad \underbrace{-2^k \leq (\!|x|\!) \leq -1}_{\textit{if } \sigma_x=1} \quad \textit{or} \quad \underbrace{0 \leq (\!|x|\!) \leq 2^k - 1}_{\textit{if } \sigma_x=0}$$

Proof. Follows from the definitions, and basic arithmetic such as that in the previous proof. □

Theorem 2.3. ***[Fundamentals of Representations and Semantics]***
The denotational functions $[\![-]\!] : \mathbb{B}^{k+1} \to \mathbb{Z}$ and $(\!|-|\!) : \mathbb{B}^{k+1} \to \mathbb{Z}$ have images $Z^{u;k}$ and $Z^{s;k}$; in fact they are bijections onto these images and the functions $\rho^u : Z^{u;k} \to \mathbb{B}^{k+1}$ and $\rho^s : Z^{s;k} \to \mathbb{B}^{k+1}$ delivering unsigned binary and signed binary representations are their respective inverses.

The commutation of the diagrams in Definition 2.2 amounts to the following equational properties:

- $[\![\pi_\sigma x]\!] = \Delta^{+,k}(\!|x|\!) = (\!|x|\!) + 2^k$
- $(\!|\pi_\sigma x|\!) = \Delta^{-,k}[\![x]\!] = [\![x]\!] - 2^k$
- $\phi([\![x]\!]) = (\!|x|\!)$
- $\psi((\!|x|\!)) = [\![x]\!]$
- $q(z \in Z^{u;k}) = q(\phi(z))$
- $q(z \in Z^{s;k}) = q(\psi(z))$

The following additional properties hold:

- $[\![x]\!] = (\!|x|\!) + \sigma_x 2^{k+1}$
- $[\![1\pi_\sigma x]\!] = [\![x]\!] + 2^k$ *when* $\sigma_x = 1$
- $(\!|1\pi_\sigma x|\!) = (\!|x|\!) - 2^k$ *when* $\sigma_x = 1$

- $$[[\pi_\sigma x]] = \begin{cases} \sigma_x = 0 : & = & [[x]] + 2^k \\ \sigma_x = 1 : & = & [[x]] - 2^k \end{cases}$$
- $$(\!|\pi_\sigma x|\!) = \begin{cases} \sigma_x = 0 : & = & (\!|x|\!) - 2^k \\ \sigma_x = 1 : & = & (\!|x|\!) + 2^k \end{cases}$$
- *The functions* π_σ, *and* $= : \mathbb{B}^{k+1} \to \mathbb{B}^{k+1}$ *are self-inverse bijections and moreover* $\pi_\sigma \overline{x} = \overline{\pi_\sigma x}$.

These operations preserve semantics:
Suppose that $x \in \mathbb{B}^{k+1}$ *and* $y \in \mathbb{B}^{k+1+k'}$. *Then*

- $[[x]] = [[y]] \iff y = zx(x)$
- $(\!|x|\!) = (\!|y|\!) \iff y = sx(x)$

and in particular, any zero extension or sign extension of x *leaves the semantics unchanged.*

Proof. From Lemma 2.2 we know that the denotational functions have the specified (finite) images.

Explicit Definitions of ρ^u *and* ρ^s*:*

The existence of $\rho^u : Z^{u;k+1} \to \mathbb{B}^{k+1}$ follows from the division algorithm on $\mathbb{N}$. Take any $z \in Z^{u;k+1}$. Then we have

$$\begin{aligned} z &= 2q_0 + a_0 \\ q_0 &= 2q_1 + a_1 \\ &\vdots \\ q_i &= 2q_{i+1} + a_{i+1} \\ &\vdots \\ q_{n-1} &= 2q_n + a_n \end{aligned}$$

where the quotients $q_i \geq 0$ and remainders $a_i \in \mathbb{B}$ are uniquely determined. Since the q_i are strictly decreasing $q_i = 0$ for some least i, namely n above. If $n = 0$ then $\rho^u(z) \stackrel{\text{def}}{=} \vec{0}z \equiv \vec{0}a_0 \in \mathbb{B}^{k+1}$ by appending a sequence of k copies of 0. Otherwise $n \geq 1$. Note that since $q_i > 0$ for $i < n$, we have $a_n = 1$ since if not, $q_{n-1} = 0$ which is a contradiction. One can see that for $i < n$ we have $q_i = \Sigma_{\alpha=i+1}^{n} a_\alpha 2^{\alpha-(i+1)}$ and hence that $z = [[a_n a_{n-1} \ldots a_0]]$.

If $n > k$ then since $a_n = 1$ we have $z = [\![a_n a_{n-1} \dots a_0]\!] \geq 2^n \geq 2^{k+1} > z$, a contradiction; the final inequality holds since $z \in Z^{u;k} = \{\ z \in \mathbb{Z}\ \mid\ 0 \leq z \leq 2^{k+1} - 1\ \}$. So $n \leq k$.

- We define $\rho^u\ :\ Z^{u;k} \longrightarrow \mathbb{B}^{k+1}$ by setting $\rho^u(z) \stackrel{\text{def}}{=} \vec{0}a_n \dots a_0 \in \mathbb{B}^{k+1}$, appending a (possibly empty) sequence of 0s.
- We define $\rho^s\ :\ Z^{s;k} \longrightarrow \mathbb{B}^{k+1}$ by setting $\rho^s(z) \stackrel{\text{def}}{=} \pi_\sigma(\rho^u(\Delta^{+,k}(z)))$.

Bijections:

Note that π_σ and $=$ are trivially self-inverse hence also a bijection. $\Delta^{-,k}$ and $\Delta^{+,k}$ trivially form a bijection.

To show that $[\![-]\!]$ is an injection one can argue by induction on k: Let $x, y \in \mathbb{B}^{k+2}$ such that $x_{k+1}2^{k+1} + [\![x^k]\!] = y_{k+1}2^{k+1} + [\![y^k]\!]$. In the case $x_{k+1} = y_{k+1}$ we are immediately done by induction yielding $x^k = y^k$. In the case $x_{k+1} \neq y_{k+1}$, wlog take $x_{k+1} = 1$. Then using Lemma 2.2 we have the contradiction $2^{k+1} > [\![y^k]\!] = 2^{k+1} + [\![x^k]\!] \geq 2^{k+1}$. Given that $[\![-]\!]$ is an injection it follows that $(\!|-|\!)$ is an injection too by a very similar argument.

Since $[\![-]\!]$ and $(\!|-|\!)$ are both injections between finite sets, and the size of $\mathbb{B}^{k+1}$ is 2^{k+1} which is the size of both $Z^{u;k}$ and $Z^{s;k}$ too, both functions are bijections as stated. In fact ρ^u and ρ^s are the respective inverses. That $[\![\rho^u(z)]\!] = [\![a_n \dots a_0]\!] = z$ is immediate from the definition of ρ^u. And $(\!|\rho^s(z)|\!) = z$ follows from the definition of ρ^s, and the fact that $\Delta^{-,k} \circ [\![-]\!] = (\!|-|\!) \circ \pi_\sigma$ (which itself follows from a simple computation).

It is immediate from the definitions that ϕ and ψ also form a bijection. It is worth noting explicitly that

$$Z^{u;k} = \{\ [\![x]\!]\ \mid\ x \in \mathbb{B}^{k+1}\ \} \qquad \text{and} \qquad Z^{s;k} = \{\ (\!|x|\!)\ \mid\ x \in \mathbb{B}^{k+1}\ \}$$

Properties:

It now follows that any two point-to-point (composition) paths in the left hand diagram commute.

For the right hand diagram, to see that $\psi((\!|x|\!)) = [\![x]\!]$, consider the cases when x is $0x^{k-1}$ and $1x^{k-1}$. In the first case ψ acts as the identity, but $(\!|0x^{k-1}|\!) = [\![0x^{k-1}]\!]$; and in the second case the left hand side is $-2^k + [\![x^{k-1}]\!] + 2^{k+1}$ which is $[\![1x^{k-1}]\!]$. The property for ϕ is immediate since it is inverse to ψ.

That $q \circ \phi = q$ and $q \circ \psi = q$ is also immediate since z, $\phi(z)$ and $\psi(z)$ always pairwise differ by $\pm\alpha * 2^{k+1}$ with $\alpha ::= 0 \mid 1$.

The additional properties are proven with simple calculations; note that the final two equations involving denotation of $\pi_\sigma x$ follow easily from the left hand diagram. For the extensions we give more detail:

$\Longleftarrow$ is trivial from the definitions of the semantics and basic algebra. $\Longrightarrow$ requires a short case analysis after expanding the definitions: for example if $(\!|x|\!) = (\!|y|\!)$, then from Lemma 2.2 it follows that $x_k = \sigma_x = \sigma_y = y_{k+k'}$. That $y_i = \sigma_x$ for every $k \leq i < k' + k$ then also follows from Lemma 2.2: consider cases $\sigma_x = 0$ and $\sigma_x = 1$, and in each case one gets a contradiction $(\!|x|\!) \neq (\!|y|\!)$ if there is any $y_{i_0} \neq \sigma_x$. One then has $[\![x^{k-1}]\!] = [\![y^{k-1}]\!]$ immediately in case $\sigma_x = 0$, and also in case $\sigma_x = 1$ since $-2^k = -2^{k+k'} + \ldots + 2^k$. $x^{k-1} = y^{k-1}$ follows from the bijection, and thus $y = sx(x)$. □

8.2 Basic Lemmas

To prove many of the theorems in this paper, we need simple properties of the Digitwize Algorithm, and of modular arithmetic. These we collect together in this short section.

Lemma 8.3. *Suppose that $a, b \in \mathbb{B}^{k+1}$. Then if $(c, r) \stackrel{\text{def}}{=} DA_{c_0}(a, b)$, it is always the case that*

$$a_i + b_i + c_i = 2c_{i+1} + r_i$$

Proof. Immediate from the definitions of *DIV* and *MOD*. □

We often compute $(x + y + z) \diamond 2$ for $\diamond$ either *DIV* or *MOD*, and x, y, z binary digits. While such computations are trivial, the next lemma is often helpful.

Lemma 8.4. *Let $d \in \mathbb{B}$. Then*

$$\begin{array}{llllll} d\ DIV\ 2 = 0 & (1+d)\ DIV\ 2 = d & (2+d)\ DIV\ 2 = 1 \\ d\ MOD\ 2 = d & (1+d)\ MOD\ 2 = \overline{d} & (2+d)\ MOD\ 2 = d \end{array}$$

Given a *DA* computation on a and b, one often then wants to compute the *DA* on a^i and b^i for some i, that is, on portions of the inputs of equal length.

Lemma 8.5. *Let $a, b \in \mathbb{B}^{k+1}$ and suppose that $(c, r) \stackrel{\text{def}}{=} DA_{c_0}(a, b)$. Then for any $0 \leq i \leq k$ we have $DA_{c_0}(a^i, b^i) = (c^{i+1}, r^i)$.*

Proof. 8.5 Let $(\hat{c}, \hat{r}) \stackrel{\text{def}}{=} DA_{c_0}(a^i, b^i)$. We need to show that for any j where $0 \leq j \leq i$, we have $\hat{c}_j = c_j^{i+1}$ and $\hat{r}_j = r_j^i$.

In the case that $j = 0$ note that $\hat{c}_0 \stackrel{\text{def}}{=} c_0 \stackrel{\text{def}}{=} c_0^{i+1}$. We also have (working inductively)

$$\hat{c}_{j+1} = (a_j^i + b_j^i + \hat{c}_j)\ DIV\ 2 = (a_j^i + b_j^i + c_j^{i+1})\ DIV\ 2 = c_{j+1}^{i+1}$$

and

$$\hat{r}_j = (a_j^i + b_j^i + \hat{c}_j)\ MOD\ 2 = (a_j^i + b_j^i + c_j^{i+1})\ MOD\ 2 = r_j^i$$

□

If we apply the DA to two numbers a and b, then extend the inputs in the most significant positions to Aa and Bb and apply the DA, we only need to compute the new carries and remainders (called K and R below).

Lemma 8.6. *Let $a, b \in \mathbb{B}^{k+1}$ and let $(c, r) \stackrel{\text{def}}{=} DA_{c_0}(a, b) \in \mathbb{B}^{k+2} \times \mathbb{B}^{k+1}$. Let $A, B \in \mathbb{B}^p$ for $p \geq 1$. Then $DA_{c_0}(Aa, Bb) = (Kc, Rr) \in \mathbb{B}^{p+k+2} \times \mathbb{B}^{p+k+1}$ where $K, R \in \mathbb{B}^p$ and are defined for $0 \leq j \leq p-1$ by*

$$R_j \stackrel{\text{def}}{=} \begin{cases} (A_0 + B_0 + c_{k+1}) \ MOD \ 2 & (j = 0) \\ (A_j + B_j + K_{j-1}) \ MOD \ 2 & (1 \leq j \leq p-1) \end{cases}$$

$$K_j \stackrel{\text{def}}{=} \begin{cases} (A_0 + B_0 + c_{k+1}) \ DIV \ 2 & (j = 0) \\ (A_j + B_j + K_{j-1}) \ DIV \ 2 & (1 \leq j \leq p-1) \end{cases}$$

Proof. Let $(\kappa, \rho) \stackrel{\text{def}}{=} DA_{c_0}(Aa, Bb)$. Then by lemma 8.5 we have

$$(\kappa^k, \rho^k) = DA_{c_0}(a, b) = (c, r)$$

Let us define $K \stackrel{\text{def}}{=} \kappa \ dropLeft \ c \in \mathbb{B}^{(p+k+2)-(k+2)}$ and also define $R \stackrel{\text{def}}{=} \rho \ dropLeft \ r \in \mathbb{B}^{(p+k+1)-(k+1)}$. Then for $0 \leq j \leq p-1$ we have $K_j = \kappa_{j+(k+2)}$ and $R_j = \rho_{j+(k+1)}$. □

Note that we will make frequent appeals to the above lemmas. Of course, with practice, the calculations they allow one to perform soon become second nature and indeed very intuitive: most if not all readers will be very experienced in doing pencil and paper arithmetic by adding up columns of digits since childhood. When we make such appeals, sometimes we will reference a lemma directly, and sometimes we refer to multiple appeals to these lemmas by stating "by Digitwize Algorithm Calculations" or "DAC" for short.

8.3 Proofs of Propositions and Theorems

Proposition 3.3. ***(A Calculus for Binary Operators)*** *Let $a, b \in \mathbb{B}^{k+1}$. Then the equations in Figure 2 hold.*

Proof. We prove only the final equation of differences; the other equations are all proven in a similar way, by unravelling the DA definitions, and using the DAC Lemmas from Section 3.1.

Define $(c, r) \stackrel{\text{def}}{=} DA_1(a, \overline{b})$. Then by Lemma 8.4 $a -_{ZA} b = \overline{c_{k+1}} r$. To compute $\pi_\sigma a -_{SA} \pi_\sigma b$ first note that $\pi_\sigma a = \overline{a_k} a^{k-1}$ and $\pi_\sigma b = \overline{b_k} b^{k-1}$. Since by Lemma 8.5 we have $(c^k, r^{k-1}) \stackrel{\text{def}}{=}$

- *If* $0 \leq i \leq k$ *then* $(a +_{ZA} b)^i = (a +_{SA} b)^i = (a +_A b)^i = a^i +_A b^i$
  - *and so* $(a +_{ZA} b)^k = (a +_{SA} b)^k = a +_A b$
- *If* $0 \leq i \leq k$ *then* $(a -_{ZA} b)^i = (a -_{SA} b)^i = (a -_A b)^i = a^i -_A b^i$
  - *and so* $(a -_{ZA} b)^k = (a -_{SA} b)^k = a -_A b$
- *If* $0 \leq i \leq k$ *then* $(a +_{ZM} b)^i = (a +_{SM} b)^i = (a +_A b)^i = a^i +_A b^i$
  - *and so* $(a +_{ZM} b)^k = (a +_{SM} b)^k = a +_A b$
- *If* $0 \leq i \leq k$ *then* $(a -_{ZM} b)^i = (a -_{SM} b)^i = (a -_A b)^i = a^i -_A b^i$
  - *and so* $(a -_{ZM} b)^k = (a -_{SM} b)^k = a -_A b$
- $a +_A b = \pi_\sigma a +_A \pi_\sigma b$
- $a -_A b = \pi_\sigma a -_A \pi_\sigma b$
- $\pi_\sigma a +_A b = a +_A \pi_\sigma b = \pi_\sigma(a +_A b)$
- $\pi_\sigma a -_A b = a -_A \pi_\sigma b = \pi_\sigma(a -_A b)$
- $a +_{ZA} b = \pi_\sigma(\pi_\sigma a +_{SA} \pi_\sigma b)$
- $a -_{ZA} b = \pi_\sigma a -_{SA} \pi_\sigma b$
- $a +_{ZA} b = zx(a +_A b) \iff$ *Condition Set 1 of Theorem 4.6 is valid.*
- $a -_{ZA} b = sx(a -_A b) \iff$ *Condition Set 1 of Theorem 4.8 is valid.*
- $a +_{SA} b = sx(a +_A b) \iff$ *Condition Set 1 of Theorem 4.9 is valid.*
- $a -_{SA} b = sx(a -_A b) \iff$ *Condition Set 1 of Theorem 4.10 is valid.*
- $a +_A b = \pi_\sigma a +_A \pi_\sigma b$
- $a +_M b = \pi_\sigma(\pi_\sigma a +_M \pi_\sigma b) \iff \sigma_a = \sigma_b$
- $a +_M b = \pi_\sigma a +_M \pi_\sigma b \iff \sigma_a \neq \sigma_b$
- $a -_M b = \pi_\sigma(\pi_\sigma a -_M \pi_\sigma b) \iff \sigma_a \neq \sigma_b$
- $a -_M b = \pi_\sigma a -_M \pi_\sigma b \iff \sigma_a = \sigma_b$
- $a +_{ZM} b = a +_{SM} b \iff \sigma_a = \sigma_b = 0$
- $a +_{ZM} b = \pi_\sigma(a +_{SM} b) \iff \sigma_a = \sigma_b = 1$
- $a +_M b = a +_{ZA} b$
- $a +_{ZM} b = zx(a +_M b)$
- $a +_{SM} b = sx(a +_M b) \iff$ *Conditions of Theorem 5.2 is valid* $\iff \sigma_a = \sigma_b$

Fig. 3. Defining a Calculus of Equations for Operations

$DA_1(a^{k-1}, \overline{b}^{k-1}) = DA_1(a^{k-1}, \overline{b^{k-1}})$, by Lemma 8.6 $\pi_\sigma a -_{SA} \pi_\sigma b$ will follow from computing

$$\pi_2 \circ DA_1(\overline{a_k a_k} a^{k-1}, b_k b_k \overline{b^{k-1}}) = \pi_2(Kc^k, Rr^{k-1}) = Rr^{k-1}$$

for $K, R \in \mathbb{B}^2$ where $R_0 = (\overline{a_k} + b_k + c_k)\ MOD\ 2$ and $R_1 = (\overline{a_k} + b_k + K_0)\ MOD\ 2$.

We need to show that $Rr^{k-1} \equiv R_1 R_0 r^{k-1} = \overline{c_{k+1}} r$, so all we need is $R_1 R_0 = \overline{c_{k+1}} r_k$. That $R_0 = r_k$ is immediate. That $R_1 = \overline{c_{k+1}}$ follows using $K_0 = (\overline{a_k} + b_k + c_k)\ DIV\ 2$ and tedious

but simple calculations: we require

$$(\overline{a_k} + b_k + (\overline{a_k} + b_k + c_k)\ DIV\ 2)\ MOD\ 2 = \overline{(a_k + \overline{b_k} + c_k)\ DIV\ 2}$$

which follows from Lemma 8.4 and a case analysis of the four values of (a_k, b_k). □

THEOREM 4.3. *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) $(\!|a|\!) - (\!|b|\!) = (\!|a -_{SA} b|\!)$

VERIFIED IN HOL

PROOF. Proof strategy: We could prove each equation using induction on k. However this is unnecessarily complicated. We can prove 1 and 2 using induction and then easily derive 3 and 4 using Theorem 2.3 (recalling that π_σ is self-inverse) and Proposition 3.3 as follows:

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

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

REMARK 8.1. *For completeness, we also give the full induction proofs of 1, 2, and proofs of $1 \Longleftrightarrow 3$, and $2 \Longleftrightarrow 4$:*

Equation 1 is valid: We use induction to show

$$\boxed{(\forall k \in \mathbb{N})(\forall a, b \in \mathbb{B}^{k+1})([\![a]\!] + [\![b]\!] = [\![a +_{ZA} b]\!])}$$

$(k = 0)$: the case is trivial by appealing to Lemma 8.3.

($k \Longrightarrow k{+}1$): Let $a, b \in \mathbb{B}^{k+2}$. We calculate $a{+}_{ZA}b$ using the Digitwize Algorithm. Suppose that $(c, r) \stackrel{\text{def}}{=} DA_0(a, b) \in \mathbb{B}^{k+3}{\times}\mathbb{B}^{k+2}$. Then by Lemma 8.6 we have $DA_0(0a, 0b) = (Kc, Rr)$ where $K, R \in \mathbb{B}$. In fact $R = (0 + 0 + c_{k+2})\ MOD\ 2 = c_{k+2}$ and $K = (0 + 0 + c_{k+1})\ DIV\ 2 = 0$ using Lemma 8.4. It follows that $DA_0(0a, 0b) = (0c^{k+2}, cr)$ and so $a +_{ZA} b = c_{k+2}r$. Hence we need to prove that

$$[\![a]\!] + [\![b]\!] = [\![c_{k+2}r]\!] = c_{k+2}2^{k+2} + r_{k+1}2^{k+1} + [\![r^k]\!]$$

Now we also have

$$[\![a]\!] + [\![b]\!] = a_{k+1}2^{k+1} + b_{k+1}2^{k+1} + [\![a^k]\!] + [\![b^k]\!]$$

But by induction $[\![a^k]\!] + [\![b^k]\!] = [\![a^k +_{ZA} b^k]\!]$ so we compute the digitwise sum. By Lemma 8.5 we have $DA_0(a^k, b^k) = (c^{k+1}, r^k)$ and so by Lemma 8.6 we have $DA_0(0a^k, 0b^k) = (0c^{k+1}, c_{k+1}r^k)$. Thus $a^k +_{ZA} b^k = c_{k+1}r^k$ and so

$$[\![a]\!] + [\![b]\!] = a_{k+1}2^{k+1} + b_{k+1}2^{k+1} + c_{k+1}2^{k+1} + [\![r^k]\!]$$

So it is sufficient to show $c_{k+2}2 + r_{k+1} = a_{k+1} + b_{k+1} + c_{k+1}$ which is immediate from Lemma 8.4.

Equation 2 is valid: We use induction.

($k = 0$): It is trivial to check all of the four cases. For example if $a_0 = 0$ and $b_0 = 1$ then $00 -_{ZA} 01 = 11$ and $[\![00]\!] - [\![01]\!] = 0 - 1 = -1 = -2 + 1 = (\!|11|\!)$.

($k \Longrightarrow k{+}1$): Let $a, b \in \mathbb{B}^{k+2}$. We calculate $a{-}_{ZA}b$ using the Division Algorithm. Suppose that $(c, r) \stackrel{\text{def}}{=} DA_1(a, \overline{b}) \in \mathbb{B}^{k+3} \times \mathbb{B}^{k+2}$. Hence by Lemma 8.6 we have $DA_0(0a, 1\overline{b}) = (Kc, Rr)$ where $K, R \in \mathbb{B}$. In fact $R = (0 + 1 + c_{k+2})\ MOD\ 2 = \overline{c_{k+2}}$ and $K = (0 + 1 + c_{k+2})\ DIV\ 2 = c_{k+2}$ using Lemma 8.4. It follows that $DA_0(0a, 1\overline{b}) = (c_{k+2}c, \overline{c_{k+2}}r)$ and so $a -_{ZA} b = \overline{c_{k+2}}r$. Hence we need to prove that

$$[\![a]\!] - [\![b]\!] = (\!|\overline{c_{k+2}}r|\!) = -\overline{c_{k+2}}2^{k+2} + r_{k+1}2^{k+1} + (\!|r^k|\!)$$

But also

$$[\![a]\!] - [\![b]\!] = a_{k+1}2^{k+1} - b_{k+1}2^{k+1} + [\![a^k]\!] - [\![b^k]\!]$$

Now by induction $[\![a^k]\!] - [\![b^k]\!] = (\!|a^k -_{ZA} b^k|\!)$ so we compute the digitwise sum. By Lemma 8.5 we have $DA_1(a^k, \overline{b}^k) = (c^{k+1}, r^k)$ and so by Lemma 8.6 $DA_1(0a^k, 1\overline{b}^k) = (c_{k+1}c^{k+1}, \overline{c_{k+1}}r^k)$. Thus $a^k -_{ZA} b^k = \overline{c_{k+1}}r^k$ and so

$$[\![a]\!] - [\![b]\!] = a_{k+1}2^{k+1} - b_{k+1}2^{k+1} - \overline{c_{k+1}}2^{k+1} + [\![r^k]\!]$$

and it is STP

$$a_{k+1}2^{k+1} - b_{k+1}2^{k+1} - \overline{c_{k+1}}2^{k+1} = -\overline{c_{k+2}}2^{k+2} + r_{k+1}2^{k+1}$$

or

$$a_{k+1} - b_{k+1} - \overline{c_{k+1}} + \overline{c_{k+2}}2 - r_{k+1} = 0$$

We can complete the proof with a case analysis on values for (a_{k+1}, b_{k+1}). Here are a couple of cases, using

$$c_{k+2} = (a_{k+1} + \overline{b_{k+1}} + c_{k+1})\ DIV\ 2 \qquad r_{k+1} = (a_{k+1} + \overline{b_{k+1}} + c_{k+1})\ MOD\ 2$$

(0,0): Since $c_{k+2} = c_{k+1}$ and $r_{k+1} = \overline{c_{k+1}}$ we have $0 - 0 - \overline{c_{k+1}} + \overline{c_{k+1}}2 - \overline{c_{k+1}} = 0$

(0,1): Since $c_{k+2} = 0$ and $r_{k+1} = c_{k+1}$ we get

$$0 - 1 - \overline{c_{k+1}} + 1 * 2 - c_{k+1} = 1 - (\overline{c_{k+1}} + c_{k+1}) = 0$$

Equation 2 is valid: We use Equation 1 and induction.

This proof is very slightly shorter than the previous one. It is typical in that it shows how correctness theorems are in some sense hierarchical.

Let $a, b \in \mathbb{B}^{k+1}$. We may show that $[[a]] = (\!|a -_{ZA} b|\!) + [[b]]$. Reasoning as above we compute $a -_{ZA} b$: we have $DA_1(0a, 1\overline{b}) = (c_{k+1}c^{k+1}, \overline{c_{k+1}}r^k)$ where $(c, r) \stackrel{\text{def}}{=} DA_1(a, \overline{b})$. Hence it is STP

$$[[a]] = (\!|\overline{c_{k+1}}r^k|\!) + [[b]] = -\overline{c_{k+1}}2^{k+1} + [[r^k]] + [[b]] = -\overline{c_{k+1}}2^{k+1} + [[r^k +_{ZA} b]]$$

with the final step following from Equation 1. Let us calculate $r^k +_{ZA} b$.

By DAC if $(K, R) \stackrel{\text{def}}{=} DA_0(r^k, b)$ we have $DA_0(0r^k, 0b) = (0K^{k+1}, K_{k+1}R^k)$. Hence we need to show that

$$[[a]] = -\overline{c_{k+1}}2^{k+1} + K_{k+1}2^{k+1} + [[R^k]] \qquad (*)$$

First we prove that $K_i = \overline{c_i}$ for each i by induction. Note that $K_0 = 0 = \overline{c_0}$. Now let $0 \le i \le k$. Since $K_i = \overline{c_i}$ inductively, we have

$$K_{i+1} = (r_i + b_i + \overline{c_i})\ DIV\ 2 \quad c_{i+1} = (a_i + \overline{b_i} + c_i)\ DIV\ 2 \quad r_i = (a_i + \overline{b_i} + c_i)\ MOD\ 2$$

and so the theorem follows from a case analysis on (a_i, b_i). Here are two cases:

(0,0): $K_{i+1} = (\overline{c_i} + \overline{c_i})\ DIV\ 2 = \overline{c_i} = \overline{c_{i+1}}$ and

(1,0): $K_{i+1} = (c_i + \overline{c_i})\ DIV\ 2 = 0 = \overline{c_{i+1}}$.

Hence using $(*)$ all we need to show now is that $[[a]] = [[R^k]]$. Note that

$$R_{i+1} = ((a_i + \overline{b_i} + c_i)\ MOD\ 2 + b_i + \overline{c_i})\ MOD\ 2$$

and from Lemma 8.4, since $(\overline{b_i} + c_i, b_i + \overline{c_i})$ is always $(1,1)$, $(2,0)$ or $(0,2)$, it follows that $R_{i+1} = a_{i+1}$. The case of $R_0 = a_0$ is easy to check.

Equation 3 is valid: In fact we only need show that $(1 \Longleftrightarrow 3)$

$(1 \Longrightarrow 3)$:

Let $(c,r) \stackrel{\text{def}}{=} DA_o(a,b) \in \mathbb{B}^{k+2} \times \mathbb{B}^{k+1}$. Then we have $a^{k-1} +_{ZA} b^{k-1} = c_k r^{k-1}$ and also $a +_{SA} b = Rr$ where $R = (\sigma_a + \sigma_b + c_{k+1})\ MOD\ 2$ Note that

$$\begin{array}{rcl} (\!|a|\!) + (\!|b|\!) & \equiv & (\!|\sigma_a a^{k-1}|\!) + (\!|\sigma_b b^{k-1}|\!) \\ & = & -\sigma_a 2^k - \sigma_b 2^k + [\![a^{k-1}]\!] + [\![b^{k-1}]\!] \\ & = & -\sigma_a 2^k - \sigma_b 2^k + [\![c_k r^{k-1}]\!] \\ & = & -\sigma_a 2^k - \sigma_b 2^k + c_k 2^k + [\![r^{k-1}]\!] \end{array}$$

and also $(\!|a +_{SA} b|\!) = -R2^{k+1} + \sigma_r 2^k + [\![r^{k-1}]\!]$. Hence it is STP $-\sigma_a - \sigma_b + c_k = -R2 + \sigma_r$. This follows from simple calculations for the cases of (σ_a, σ_b). For example

$(0,1)$: The left hand side is $-1 + c_k$. The right hand side is

$$-2 * (1 + (1 + c_k)\ DIV\ 2)\ MOD\ 2 + (1 + c_k)\ MOD\ 2 = -(1 + c_k)\ MOD\ 2 = -\overline{c_k}$$

which equals the left hand side.

$(3 \Longrightarrow 1)$:

Let $(c,r) \stackrel{\text{def}}{=} DA_o(a,b) \in \mathbb{B}^{k+2} \times \mathbb{B}^{k+1}$. Then $a +_{ZA} b = c_{k+1} r$ and $0a +_{SA} 0b = 0c_{k+1} r$ follows by DAC. We have

$$[\![a]\!] + [\![b]\!] = (\!|0a|\!) + (\!|0b|\!) =_{(3)} (\!|0a +_{SA} 0b|\!) = (\!|0c_{k+1} r|\!) = [\![a +_{ZA} b]\!]$$

Equation 2 is valid: In fact we only need show that $(2 \Longleftrightarrow 4)$

$(2 \Longrightarrow 4)$:

We have by DAC $a -_{SA} b = Rr$ where $R = (\sigma_a + \overline{\sigma_b} + c_{k+1})\ MOD\ 2$ and $a^{k-1} -_{ZA} b^{k-1} = \overline{c_k} r^{k-1}$. Then

$$\begin{array}{rcl} (\!|a|\!) - (\!|b|\!) & = & -\sigma_a 2^k + \sigma_b 2^k + [\![a^{k-1}]\!] - [\![b^{k-1}]\!] \\ & =_{(2)} & -\sigma_a 2^k + \sigma_b 2^k + (\!|\overline{c_k} r^{k-1}|\!) \\ & = & -\sigma_a 2^k + \sigma_b 2^k - \overline{c_k} 2^k + [\![r^{k-1}]\!] \end{array}$$

and $(\!|a -_{SA} b|\!) = -R2^{k+1} + \sigma_r 2^k + [\![r^{k-1}]\!]$. Hence it is STP $-\sigma_a + \sigma_b - \overline{c_k} + 2R - \sigma_r = 0$. This follows from simple calculations for the cases of (σ_a, σ_b). For example

$(0,1)$:

$$1 - \overline{c_k} + 2 * (0 + (0 + c_k)\ DIV\ 2)\ MOD\ 2 - (0 + c_k)\ MOD\ 2 = 1 - \overline{c_{k+1}} - c_k = 0.$$

$(1,1)$:

$$-\overline{c_k}+2*(1+(1+c_k)\ DIV\ 2)\ MOD\ 2-(1+c_k)\ MOD\ 2 = -\overline{c_k}+(1+c_k)\ MOD\ 2 = -\overline{c_k}+\overline{c_k} = 0$$

$(4 \Longrightarrow 2)$:

Let $(c,r) \stackrel{\text{def}}{=} DA_o(a,b) \in \mathbb{B}^{k+2} \times \mathbb{B}^{k+1}$. Then $0a -_{SA} 0b = \overline{c_{k+1}c_{k+1}}r$ and $a -_{ZA} b = \overline{c_{k+1}}r$ by DAC. So we have

$$[[a]] - [[b]] = (|0a|) - (|0b|) =_{(4)} (|\overline{c_{k+1}c_{k+1}}r|) =_* (|\overline{c_{k+1}}r|) = (|a -_{ZA} b|)$$

with $=_*$ following by trivial calculation.

□

Theorem 4.6. *Let $a, b \in \mathbb{B}^{k+1}$. Then the following conditions are* pairwise equivalent

Condition Set 1

(1) ***Correctness:*** $[[a]] + [[b]] = [[a +_A b]]$
(2) ***Range:*** $0 \le [[a]] + [[b]] \le 2^{k+1} - 1$
(3) ***Sign Occurrences:*** $\langle \sigma_a, \sigma_b, \sigma_r \rangle$ *can be any one of the following four sign-occurrences* $\langle 0,0,0/1 \rangle$, $\langle 0,1,1 \rangle$, $\langle 1,0,1 \rangle$, *but no others.*
(4) ***Overflow Digit:*** $\sigma_c = 0$
(5) ***Sign Equation:*** $\sigma_a + \sigma_b - \sigma_r = 0, -1$

and the following conditions are pairwise equivalent

Condition Set 2

(1) ***Correctness:*** $[[a]] + [[b]] = [[1(a +_A b)]]$
(2) ***Range:*** $2^{k+1} \le [[a]] + [[b]] \le 2 * (2^{k+1} - 1)$
(3) ***Sign Occurrences:*** $\langle \sigma_a, \sigma_b, \sigma_r \rangle$ *can be any one of the following four sign-occurrences* $\langle 0,1,0 \rangle$, $\langle 1,0,0 \rangle$, $\langle 1,1,0/1 \rangle$, *but no others.*
(4) ***Overflow Digit:*** $\sigma_c = 1$
(5) ***Sign Equation:*** $\sigma_a + \sigma_b - \sigma_r = 2, 1$

and moreover for each ***condition type*** *(such as* ***Correctness****), one instance must hold (along with its condition set equivalents).* VERIFIED IN HOL

Proof. We prove the first set of conditions are equivalent explicitly, by induction.

First we show that $(3 \Longleftrightarrow 4 \Longleftrightarrow 5)$.

$(3 \Longleftrightarrow 5)$: is a simple calculation.

Both $(4 \Longleftrightarrow 3)$ or $(4 \Longleftrightarrow 5)$ are of similar complexity; each requires a short calculation. We look at $(3 \Longrightarrow 4)$.

($3 \implies 4$): We have, using DAC lemmas, $\sigma_c = (0 + c_k)\ DIV\ 2 = 0$ or $\sigma_c = (1 + c_k)\ DIV\ 2$. But in this second case, $1 = \sigma_r = (1 + c_k)\ MOD\ 2 = \overline{c_k}$ so $c_k = 0$ and then $\sigma_c = 0$.

($4 \implies 3$): just examine the possibilities for the digits σ_a, σ_b, σ_r and c_k when $\sigma_c = 0$ by using Lemma 8.3 which tells us that

$$\sigma_a + \sigma_b + c_k = 2\sigma_c + \sigma_r$$

Next we show that ($1 \implies 2 \implies 4 \implies 1$).

First, define $(c, r) \stackrel{\text{def}}{=} DA_0(a, b)$ and so by DAC (especially Lemma 3.3) we have $a +_{ZA} b = \sigma_c r^k$ and $a +_A b = r^k$. So from theorem 4.3 $[[a]] + [[b]] = [[\sigma_c r^k]]$.

($1 \implies 2$): is trivial.

($2 \implies 4$): Suppose that (2) holds. We must have

$$0 \leq \sigma_c 2^{k+1} + [[r^k]] \leq 2^k - 1$$

The case $\sigma_c = 1$ can't happen (Lemma 2.2) and so $\sigma_c = 0$ which is (4).

($4 \implies 1$): If (4) holds then $[[a]] + [[b]] = [[r^k]] = [[a +_A b]]$ and so (1) holds.

The pairwise equivalence of the second set of conditions then follows easily: Note that the correct integer sum $[[a]] + [[b]]$ lies between 0 and $2(2^{k+1} - 1)$ using Lemma 2.2; that there are trivially eight mutually exclusive **Sign Occurrence** conditions; the overflow digit is either 1 or 0; and that the **Sign Equation** can only have values $-1, 0, 1, 2$.

Thus any single condition from 2 to 5 of the second set is the negation of the respective condition from the first set. Since conditions 2 to 5 of the first set are pairwise equivalent, their negations must be pairwise equivalent.

The fact that 1 of the second set is equivalent to 4 of the second set follows from Theorem 4.3, Equation 1, together with Proposition 3.3 and DAC:

$$[[a]] + [[b]] = [[a +_{ZA} b]] = [[\sigma_c (a +_A b)]]$$

Finally, to show that for each condition type one instance must hold, it is now sufficient to show that either the first or second range condition holds. But this is immediate from Lemma 2.2.

□

Theorem 4.8. *Let $a, b \in \mathbb{B}^{k+1}$. Then the following conditions are pairwise* pairwise equivalent

Condition Set 1

(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:*** $\sigma_c \neq \sigma_r$.
(5) ***Sign Equation:*** $-\sigma_a + \sigma_b - \sigma_r = 0, -1$

and the following conditions are pairwise equivalent

Condition Set 2

(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:*** $\sigma_c = \sigma_r = 0$.
(5) ***Sign Equation:*** $-\sigma_a + \sigma_b - \sigma_r = 1$

and the following conditions are pairwise equivalent

Condition Set 3

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

and moreover for each ***condition type*** *one instance must hold (along with equivalents of the instance).*

VERIFIED IN HOL

Proof. We again use the proof strategy of Remark 4.2. The first set of conditions is proved pairwize. The proof for the second set is omitted, but is very similar in structure (and further, there is a proof in the Appendix for the second condition set of Theorem 4.9). We can deduce the validity of the third set from the first two.

Recall Remark 4.2.

The first set of conditions is proven pairwise equivalent:

First we show that $(3 \iff 4 \iff 5)$.

$(3 \iff 5)$: by simple calculation.

To prove $(3 \iff 4)$ it is easier to prove $(\neg 3 \iff \neg 4)$, i.e. that $\langle \sigma_a, \sigma_b, \sigma_c\rangle$ is either $\langle 0,1,0\rangle$ or $\langle 1,0,1\rangle$ just in case $\sigma_c = \sigma_r$.

Recall $\sigma_c = (\sigma_a + \overline{\sigma_b} + c_k)$ *DIV* 2 and $\sigma_r = (\sigma_a + \overline{\sigma_b} + c_k)$ *MOD* 2.

($4 \implies 3$): Assume that we have $\langle 0,1,0\rangle$ or $\langle 1,0,1\rangle$. Then $\sigma_c = c_k \ DIV \ 2 = 0 = \sigma_r$ or $\sigma_c = (2 + c_k) \ DIV \ 2 = 1 = \sigma_r$.

($3 \implies 4$): Assume that $\sigma_c = \sigma_r$ and note that if $\sigma_c = \sigma_r = 0$ then $\sigma_a + \overline{\sigma_b} + c_k = 0$ and hence $\sigma_a = 0$ and $\sigma_b = 1$, giving **Sign Occurrence** condition $\langle 0,1,0\rangle$. If $\sigma_c = \sigma_r = 1$ then $\sigma_a + \overline{\sigma_b} + c_k = 3$ and hence $\sigma_a = 1$ and $\sigma_b = 0$, giving **Sign Occurrence** condition $\langle 1,0,1\rangle$.

Next we show that ($1 \implies 2 \implies 4 \implies 1$).

First, let $(c,r) \stackrel{\text{def}}{=} DA_1(a,\overline{b})$ and so by DAC we have $a -_{ZA} b = \overline{\sigma_c} r^k$ and $a -_A b = (a -_{ZA} b)^k = (\overline{\sigma_c} r^k)^k = r^k$. From theorem 4.3 $[\![a]\!] - [\![b]\!] = (\!|\overline{\sigma_c} r^k|\!)$.

($1 \implies 2$): is trivial.

($2 \implies 4$): Let's suppose that 2 holds. Hence we must have

$$-2^k \leq (\xi \stackrel{\text{def}}{=} -\overline{\sigma_c} 2^{k+1} + \sigma_r 2^k + [\![r^{k-1}]\!]) \leq 2^k - 1$$

In the case that $\sigma_c = \sigma_r = 0$ we have $\xi \leq -2^{k+1} + (2^k - 1) = -2^k - 1 < -2^k$ and if $\sigma_c = \sigma_r = 1$ we have $2^k - 1 < 2^k \leq \xi$ neither of which can happen. Hence we must have $\sigma_c \neq \sigma_r$ and so 4 holds.

($4 \implies 1$): When 4 holds note that $\overline{\sigma_c} = \sigma_r$ is equivalent. Then 1 holds since

$$[\![a]\!] - [\![b]\!] = -\sigma_r 2^{k+1} + \sigma_r 2^k + [\![r^{k-1}]\!] = -\sigma_r 2^k + [\![r^{k-1}]\!] = (\!|r^k|\!) = (\!|a -_A b|\!)$$

Thus the first set of conditions is proven pairwise equivalent.

Now let us prove the pairwise equivalence of conditions from the third set. Choose any **condition type** apart from **Correctness**. Then, arguing in a similar manner to that in the proof of Theorem 4.6, one can see that the three conditions of that **condition type** are *mutually exclusive*. Since conditions 2 to 5 of the first and second set are pairwise equivalent, the corresponding sets of negations must be pairwise equivalent. From elementary propositional reasoning, conditions 2 to 5 of the third set are therefore pairwise equivalent.

The fact that 1 of the third condition set is equivalent to 4 of the third set follows from

$$[\![a]\!] - [\![b]\!] = (\!|a -_{ZA} b|\!) = (\!|\overline{\sigma_c}(a -_A b)|\!) \qquad \dagger$$

where the first equality is Equation 2 of Theorem 4.3, and the second equality uses the trivial fact that $x \equiv x_{k+1} x^k$ for $x \in \mathbb{B}^{k+2}$ along with Proposition 3.3 and by DAC

$$(a -_{ZA} b)_{k+1} \stackrel{\text{def}}{=} (0 + 1 + \sigma_c) \ MOD \ 2 = (1 + \sigma_c) \ MOD \ 2 = \overline{\sigma_c}$$

From $\dagger$ ($4 \implies 1$) is trivial (note the conjunct $\sigma_r = 1$).

$(1 \implies 4)$ uses the bijection of Theorem 2.3 to deduce from † that $\overline{\sigma_c}(a -_A b) = 0(a -_A b)$ and so $\overline{\sigma_c} = 0$ and hence $\sigma_c = 1$ (again, note the conjunct $\sigma_r = 1$ from 1).

Finally we establish that for each **condition type** one instance must hold. For any a and b, from Lemma 2.2 we can *deduce* that $-2^{k+1} + 1 \leq [[a]] - [[b]] \leq 2^{k+1} - 1$. Therefore one of the (three mutually exclusive) instances of **Range** must hold. □

THEOREM 4.9. *Let $a, b \in \mathbb{B}^{k+1}$. Then the following conditions are* pairwise equivalent

Condition Set 1

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

and the following conditions are pairwise equivalent

Condition Set 2

(1) ***Correctness:*** $(\!|a|\!) + (\!|b|\!) = (\!|1(a +_A b)|\!) \wedge \sigma_r = 0$
(2) ***Range:*** $-2^{k+1} \leq (\!|a|\!) + (\!|b|\!) \leq -2^k - 1$
(3) ***Sign Occurrences:*** *The sign-occurrence* $\langle 1, 1, 0\rangle$.
(4) ***Overflow Digit:*** $\sigma_c = 1$ and $c_k = 0$.
(5) ***Sign Equation:*** $-\sigma_a - \sigma_b + \sigma_r = -2$

and the following conditions are pairwise equivalent

Condition Set 3

(1) ***Correctness:*** $(\!|a|\!) + (\!|b|\!) = (\!|0(a +_A b)|\!) \wedge \sigma_r = 1$
(2) ***Range:*** $2^k \leq (\!|a|\!) + (\!|b|\!) \leq 2(2^k - 1)$
(3) ***Sign Occurrences:*** *The sign-occurrence*$\langle 0, 0, 1\rangle$.
(4) ***Overflow Digit:*** $\sigma_c = 0$ and $c_k = 1$.
(5) ***Sign Equation:*** $-\sigma_a - \sigma_b + \sigma_r = 1$

and moreover for each ***condition type*** *one instance must hold (along with equivalents of the instance).*

VERIFIED IN HOL

PROOF. Note that an approach to proving Theorem 4.9 is to exploit Theorem 4.6. This is illustrated later in the Appendix on page 50; see Remark 8.2.

Recall Remark 4.2.

The first set of conditions is proven pairwise equivalent.

First we show that $(3 \iff 4 \iff 5)$.

$(3 \iff 5)$: is easy.

To prove $(3 \iff 4)$ it is easier to prove $(\neg 3 \iff \neg 4)$ i.e. that $\sigma_c \neq c_k$ just in case either $\langle 0,0,1\rangle$ or $\langle 1,1,0\rangle$.

Recall $\sigma_c = (\sigma_a + \sigma_b + c_k)\ DIV\ 2$ and $\sigma_r = (\sigma_a + \sigma_b + c_k)\ MOD\ 2$. For

$(4 \implies 3)$: Assume that we have $\langle 0,0,1\rangle$ or $\langle 1,1,0\rangle$. Then either $\sigma_c = c_k\ DIV\ 2 = 0$ and also $1 = c_k\ MOD\ 2$ so $c_k = 1$; or $\sigma_c = (2 + c_k)\ DIV\ 2 = 1$ and also $0 = (2 + c_k)\ MOD\ 2$ so $c_k = 0$. For

$(3 \implies 4)$: Assume that $\sigma_c \neq c_k$ and note that if $\sigma_c = 0$ and $c_k = 1$ then $(\sigma_a + \sigma_b + 1)\ DIV\ 2 = 0$ and hence $\sigma_a + \sigma_b = 0$ implying $\sigma_r = 1\ MOD\ 2 = 1$, giving **Sign Occurrence** condition $\langle 0,0,1\rangle$. If $\sigma_c = 1$ and $c_k = 0$ then $(\sigma_a + \sigma_b)\ DIV\ 2 = 1$ and hence $\sigma_a + \sigma_b = 2$ implying $\sigma_r = 2\ MOD\ 2 = 0$, giving **Sign Occurrence** condition $\langle 1,1,0\rangle$.

Next we show that $(1 \implies 2 \implies 4 \implies 1)$.

$(1 \implies 2)$: is trivial.

As ever let $(c, r) \stackrel{\text{def}}{=} DA_0(a, b)$ and so $a +_{SA} b = Rr^k$ where $R = (\sigma_a + \sigma_b + \sigma_c)\ MOD\ 2$. Further $a -_A b = r^k$, the latter by appealing to Corollary 3.3. From theorem 4.3 $(\!|a|\!) + (\!|b|\!) = (\!|Rr^k|\!)$.

$(2 \implies 4)$: Let's suppose that 2 holds. Hence we must have

$$-2^k \leq (\xi \stackrel{\text{def}}{=} -R2^{k+1} + \sigma_r 2^k + [\![r^{k-1}]\!]) \leq 2^k - 1$$

Arguing much as in the proof of Theorem 4.8 we conclude we must have $R = \sigma_r$. Hence

$$(\sigma_a + \sigma_b + \sigma_c)\ MOD\ 2 = (\sigma_a + \sigma_b + c_k)\ MOD\ 2 \qquad (*)$$

If $\sigma_c \neq c_k$ this cannot be, and so $\sigma_c = c_k$. Hence 4 holds.

$(4 \implies 1)$: When 4 holds we have $R = \sigma_r$ and so

$$(\!|a|\!) + (\!|b|\!) = -\sigma_r 2^{k+1} + \sigma_r 2^k + [\![r^{k-1}]\!] = -\sigma_r 2^k + [\![r^{k-1}]\!] = (\!|r^k|\!) = (\!|a +_A b|\!)$$

Thus the first set of conditions is proven pairwise equivalent.

The second set of conditions is proven pairwise equivalent.

First we show that $(3 \iff 4 \iff 5)$.

$(3 \iff 5)$: is easy.

Recall $\sigma_c = (\sigma_a + \sigma_b + c_k)\ DIV\ 2$ and $\sigma_r = (\sigma_a + \sigma_b + c_k)\ MOD\ 2$. For

$(4 \implies 3)$: We have $1 = (\sigma_a + \sigma_b)\ DIV\ 2$ and so $\sigma_a = \sigma_b = 1$. Hence $\sigma_r = 2\ MOD\ 2 = 0$.

$(3 \implies 4)$: We have $\sigma_c = (2 + c_k)\ DIV\ 2$ and $\sigma_r = 0 = (2 + c_k)\ MOD\ 2$. 4 is immediate.

Next we show that $(1 \implies 2 \implies 3 \implies 1)$.

$(1 \implies 2)$: From 1 we have $S \stackrel{\text{def}}{=} (\!|a|\!) + (\!|b|\!) = -2^{k+1} + [\![a +_A b]\!]$. Then note that from Lemma 2.2 we have $-2 * 2^k \leq S$. Since $\sigma_r = 0$ we have $-2^{k+1} + [\![a +_A b]\!] \leq -2^{k+1} + (2^k - 1)$. Thus 2 is true.

$(2 \implies 3)$: Let's suppose that 2 holds. Reasoning much as we did for the first condition set we must have

$$-2^{k+1} \leq (\xi \stackrel{\text{def}}{=} -R2^{k+1} + \sigma_r 2^k + [\![r^{k-1}]\!]) \leq -2^k - 1$$

Since $\xi < 0$ we have $R = 1$. Therefore $\sigma_r 2^k + [\![r^{k-1}]\!] \leq 2^k - 1$ and so $\sigma_r = 0$. Thus we have $1 = R = (\sigma_a + \sigma_b + \sigma_c)\ MOD\ 2 \quad (*)$ and $0 = \sigma_r = (\sigma_a + \sigma_b + c_k)\ MOD\ 2$. So $(\sigma_a + \sigma_b + c_k) = 0$ or 2. If the former, from $(*)$ we have $\sigma_c = 1$. But $\sigma_c \stackrel{\text{def}}{=} (\sigma_a + \sigma_b + c_k)\ DIV\ 2 = 0$. Hence $(\sigma_a + \sigma_b + c_k) = 2$. Therefore we still have $\sigma_c = 1$ implying $\sigma_a + \sigma_b = 0$ or 2. In the former case we have $(0 + 0 + c_k) = 2$, and so finally we have $\sigma_a = \sigma_b = 1$. Hence 3 is true.

$(3 \implies 1)$: Suppose 3. We have to prove $\sigma_r = 0$ and $R \stackrel{\text{def}}{=} (\sigma_a + \sigma_b + \sigma_c)\ MOD\ 2 = 1$. The former is immediate. For the latter we only need to show that $\sigma_c = 1$. But $\sigma_c = (\sigma_a + \sigma_b + c_k)\ DIV\ 2(2 + c_k)\ DIV\ 2 = 1$. So 1 is true.

Thus the second set of conditions is proven pairwise equivalent.

We can then deduce that condition set 3 is pairwise equivalent as before using simple logic, but in the deduction below we show $(1 \iff 3)$ and not $(1 \iff 4)$ as in our proofs of Theorems 4.6 and 4.8.

Choose any such **condition type** apart from **Correctness**. Then, arguing in a similar manner to that in the proof of Theorem 4.6, one can see that the three conditions of that **condition type** are *mutually exclusive*. It follows easily that conditions 2 to 5 of the third set are pairwise equivalent.

The fact that 1 of the third condition set is equivalent to 3 of the third set follows from

$$(\!|a|\!) + (\!|b|\!) = (\!|a +_{SA} b|\!) = (\!|(\sigma_a + \sigma_b + \sigma_c)\ MOD\ 2(a +_A b)|\!) \qquad \dagger$$

where the first equality is Equation 3 of Theorem 4.3, and the second equality uses the trivial fact that $x \equiv x_{k+1}x^k$ for $x \in \mathbb{B}^{k+2}$ along with Proposition 3.3 and

$$(a +_{SA} b)_{k+1} \stackrel{\text{def}}{=} (\sigma_a + \sigma_b + \sigma_c)\ MOD\ 2$$

$(3 \implies 1)$: Assuming 3 we have $\sigma_a = 0$ and $\sigma_b = 0$ and $\sigma_r = 1$. The right conjunct of 1 is immediate. Therefore we need to show that $(\sigma_a + \sigma_b + \sigma_c)\ MOD\ 2 = \sigma_c\ MOD\ 2 = 0$. But

since $\sigma_r = 1$, we have $c_k \ MOD\ 2 = 1$ and so $c_k = 1$, and then $\sigma_c = 1 \ DIV\ 2 = 0$. Thus we are done.

$(1 \implies 3)$: From † and Theorem 2.3 we deduce that $(\sigma_a + \sigma_b + \sigma_c) \ MOD\ 2 = 0 \quad (*)$. Now assume 1, so $\sigma_r = 1$ is immediate. So it remains to prove that $\sigma_a = \sigma_b = 0$. From Lemma 8.3 we have $2\sigma_c + \sigma_r = 2\sigma_c + 1 = \sigma_a + \sigma_b + c_k$. Now from $(*)$ we have $\sigma_a + \sigma_b + \sigma_c = 0$ or 2. In the latter case we then have $2\sigma_c + 1 = 2 - \sigma_c + c_k$ which cannot be true. Hence $\sigma_a + \sigma_b + \sigma_c = 0$ and we are done.

Finally we establish that for each **condition type** one instance must hold. For any a and b, from Lemma 2.2 we can *deduce* that $-2^{k+1} \leq (|a|) + (|b|) \leq 2(2^k - 1)$. Therefore one of the (three) instances of **Range** must hold.

Remark 8.2. *It is also possible to prove pairwise equivalence of each condition set from Theorem 4.9 by making use of Theorem 4.6.*

Some of these details we now present here; we first explain which details we choose to provide:

- For the first condition set, we look only at $(1 \implies 2 \implies 3 \implies 1)$. There is not much to be gained by proving $(3 \iff 4 \iff 5)$ using Theorem 4.6; for example Theorem 4.6 gives conditions for $\sigma_c \stackrel{\text{def}}{=} \sigma_c$ to be 0 or 1, but Theorem 4.9 has Overflow Digit statements involving both $\sigma_c \stackrel{\text{def}}{=} \sigma_c$ and c_k. So the approach is not so helpful.
- For the second condition set, we also show $(1 \implies 2 \implies 3 \implies 1)$
- For the third condition set, we can easily derive the equivalences logically using the equivalences of the first and second sets, as in the main text.

Let's look at $(1 \implies 2 \implies 3 \implies 1)$ for the first condition set.

$(1 \implies 2)$ is trivial.

$(2 \implies 3)$: We have

$$-2^k \leq (|a|) + (|b|) \leq 2^k - 1$$

and so by Theorem 2.3

we have

$$-2^k \leq ([[\pi_\sigma a]] - 2^k) + ([[\pi_\sigma b]] - 2^k) \leq 2^k - 1$$

or equivalently

$$2^k \leq [[\pi_\sigma a]] + [[\pi_\sigma b]] \leq 2 * 2^k + (2^k - 1)$$

and in particular

$$\underbrace{0 \leq 2^k \leq [[\pi_\sigma a]] + [[\pi_\sigma b]] \leq 2 * 2^k - 1}_{\alpha} < \underbrace{2 * 2^k \leq [[\pi_\sigma a]] + [[\pi_\sigma b]] \leq 3 * 2^k - 1 \leq 2(2^{k+1} - 1)}_{\beta}$$

(Case α): Using α we deduce the first **Correctness** condition of Theorem 4.6 holds. Therefore $[[\pi_\sigma a]] + [[\pi_\sigma b]] = [[\pi_\sigma a +_A \pi_\sigma b]]$. From Proposition 3.3 we know that $\pi_\sigma a +_A \pi_\sigma b = a +_A b$. Since also, using α, $2^k \leq [[a +_A b]]$ we we can conclude from Lemma 2.2 that $\sigma_{a+_A b} = 1$.

Likewise from α we deduce the first **Sign Occurrence** condition of Theorem 4.6 holds. Hence we know that $\langle \sigma_{\pi_\sigma a}, \sigma_{\pi_\sigma b}, \sigma_{a+_A b)} \rangle$ may be any of the triples $\langle 0, 0, 0/1 \rangle$, $\langle 0, 1, 1 \rangle$, $\langle 1, 0, 1 \rangle$. However, since we have $\sigma_{a+_A b} = 1$, $\langle \sigma_a, \sigma_b, \sigma_{a+_A b} \rangle$ must be one of $\langle 1, 1, 1 \rangle$, $\langle 1, 0, 1 \rangle$, $\langle 0, 1, 1 \rangle$.

(Case β): From β we deduce second **Correctness** condition of Theorem 4.6 holds. Together with Proposition 3.3 we deduce (reasoning similarly to Case α)

$$2 * 2^k \leq [[1(a +_A b)]] = 2 * 2^k + [[a +_A b]] \leq 3 * 2^k - 1 \leq 2(2^{k+1} - 1)$$

and so

$$0 \leq [[a +_A b]] \leq 2^k - 1$$

that is $\sigma_{a+_A b} = 0$.

Likewise from β second **Sign Occurrence** condition of Theorem 4.6 holds. Hence $\langle \sigma_a, \sigma_b, \sigma_{a+_A b} \rangle$ must be one of $\langle 0, 0, 0 \rangle$, $\langle 1, 0, 0 \rangle$, $\langle 0, 1, 0 \rangle$.

(3 $\Longrightarrow$ 1): By Theorem 2.3 we need to prove that

$$[[\pi_\sigma a]] + [[\pi_\sigma b]] - 2 * 2^k = [[\pi_\sigma (a +_A b)]] - 2^k$$

that is

$$[[\pi_\sigma a]] + [[\pi_\sigma b]] - 2^k = [[\pi_\sigma (a +_A b)]].$$

The assumption, 3, states that six possible sign occurrences may hold. First, consider the cases of 3 when $\langle \sigma_a, \sigma_b, \sigma_{a+_A b} \rangle$ is one of $\langle 0, 0, 0 \rangle$, $\langle 1, 0, 0 \rangle$, $\langle 0, 1, 0 \rangle$ and hence $\langle \sigma_{\pi_\sigma a}, \sigma_{\pi_\sigma b}, \sigma_{\pi_\sigma a +_A \pi_\sigma b} = \sigma_{a+_A b} \rangle$ is one of $\langle 1, 1, 0 \rangle$, $\langle 0, 1, 0 \rangle$, $\langle 1, 0, 0 \rangle$. In each case, appeal to the equivalence of the second **Sign Occurrence** and **Correctness** conditions of Theorem 4.6 to deduce that $[[\pi_\sigma a]] + [[\pi_\sigma b]] = [[1(a +_A b)]]$. Therefore $[[\pi_\sigma a]] + [[\pi_\sigma b]] - 2^k = [[a +_A b]] + 2^k$, and since $\sigma_{a+_A b} = 0$ in each case we have $[[a +_A b]] + 2^k = [[\pi_\sigma (a +_A b)]]$ by Theorem 2.3.

In the other cases of 3, we have $\langle \sigma_{\pi_\sigma a}, \sigma_{\pi_\sigma b}, \sigma_{\pi_\sigma a +_A \pi_\sigma b} = \sigma_{a+_A b} \rangle$ is one of $\langle 0, 0, 1 \rangle$, $\langle 0, 1, 1 \rangle$, $\langle 1, 0, 1 \rangle$. Thus in each case, from Theorem 4.6, $[[\pi_\sigma a]] + [[\pi_\sigma b]] = [[a +_A b]]$. Thus $[[\pi_\sigma a]] +$

$[\![\pi_\sigma b]\!] - 2^k = [\![a +_A b]\!] - 2^k = [\![\pi_\sigma(a +_A b)]\!]$, with the last step following from Theorem 2.3, together with $\sigma_{a+_A b} = 1$ in each case. Hence first condition 1 holds.

Let's prove the equivalence of the second set of conditions:

$(1 \implies 2)$: easy.

$(2 \implies 3)$: We have

$$-2 * 2^k \leq (\!|a|\!) + (\!|b|\!) \leq -2^k - 1$$

and so by Theorem 2.3 we have

$$-2 * 2^k \leq ([\![\pi_\sigma a]\!] - 2^k) + ([\![\pi_\sigma b]\!] - 2^k) \leq -2^k - 1$$

or equivalently

$$0 \leq [\![\pi_\sigma a]\!] + [\![\pi_\sigma b]\!] \leq 2^k - 1 \qquad (*)$$

Since $2^k - 1 \leq 2 * 2^k - 1$, the first **Range** condition from Theorem 4.6 holds.

Therefore by the equivalent first **Correctness** condition from Theorem 4.6 (and Proposition 3.3) we get $[\![\pi_\sigma a]\!] + [\![\pi_\sigma b]\!] = [\![a +_A b]\!]$, and so by $(*)$ and Lemma 2.2 we have $\sigma_{a+_A b} = 0$.

Further, by the equivalent first **Sign Occurrence** condition from Theorem 4.6 only $\langle \sigma_{\pi_\sigma a}, \sigma_{\pi_\sigma b}, \sigma_{\pi_\sigma a +_A \pi_\sigma b} = \sigma_{a+_A b} \rangle = \langle 0, 0, 0 \rangle$ holds. Therefore $\langle \sigma_a, \sigma_b, \sigma_{a+_A b} \rangle = \langle 1, 1, 0 \rangle$ which is the **Sign Occurrence** condition 3 to be proven.

$(3 \implies 1)$: We need to show that

$$[\![\pi_\sigma a]\!] + [\![\pi_\sigma b]\!] - 2 * 2^k = -2^{k+1} + ([\![\pi_\sigma(a +_A b)]\!] - 2^k)$$

But 3 states that $\langle \sigma_a, \sigma_b, \sigma_{a+_A b} \rangle = \langle 1, 1, 0 \rangle$ and so $\langle \sigma_{\pi_\sigma a}, \sigma_{\pi_\sigma b}, \sigma_{\pi_\sigma a +_A \pi_\sigma b} = \sigma_{a+_A b} \rangle = \langle 0, 0, 0 \rangle$. Hence using Theorem 4.6 and Proposition 3.3 we can deduce that $[\![\pi_\sigma a]\!] + [\![\pi_\sigma b]\!] = [\![\pi_\sigma a + \pi_\sigma b]\!] = [\![a +_A b]\!]$. Since $\sigma_{a+_A b} = 0$ we have $[\![a +_A b]\!] = [\![\pi_\sigma(a +_A b)]\!] - 2^k$ from Theorem 2.3. Thus we are done. □

Theorem 4.10. *Let $a, b \in \mathbb{B}^{k+1}$. Then the following conditions are pairwise* equivalent

Condition Set 1

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

and the following conditions are pairwise equivalent

Condition Set 2

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

and the following conditions are equivalent

Condition Set 3

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

and moreover for each ***condition type*** *one instance must hold (along with equivalents of the instance).*

VERIFIED IN HOL

Before the proof, we give some additional explanations of the Condition Sets, and how they may be derived from first principles. In order to be as general as possible, given any $a, b \in \mathbb{B}^{k+1}$, we write $C_{a,b} \in \mathbb{Z}$ for the correct integer answer and $O_{a,b} \in \mathbb{Z}$ for the computer output (so in Theorem 4.10 we have $C_{a,b} \stackrel{\text{def}}{=} (\!|a|\!) - (\!|b|\!)$ and $O_{a,b} \stackrel{\text{def}}{=} (\!|a -_A b|\!)$).

Remark 8.3. *The* ***Correctness*** *equations for $a, b \in \mathbb{B}^{k+1}$ such as $(\!|a|\!) - (\!|b|\!) = (\!|a -_A b|\!)$ are simply formal statements of the intuitive notion of correctness.*

Each ***Range*** *condition states that a correctness equation holds when the Correct integer answer $C_{a,b}$ lies within a given range. One may deduce these ranges as follows. Given $a, b \in \mathbb{B}^{k+1}$, each of a and b has the same maximum and minimum denotation in $\mathbb{Z}$: maximum is $2^k - 1$ and lower is -2^k. Given the definition of $C_{a,b}$ here, the maximum must be $(2^k - 1) - (-2^k) = 2^{k+1} - 1$. And the minimum is $-2^k - (2^k - 1) = -2^{k+1} + 1$. Thus overall we have*

$$-2^{k+1} + 1 \leq C_{a,b} \leq 2^{k+1} - 1$$

and it's easy to see that $C_{a,b}$ attains every value. We can apply a similar process to any $C_{a,b}$ in the other theorems. Now consider the correctness equation of the first ***Range*** *Condition, which in general states that $C_{a,b} = O_{a,b}$. Since $O_{a,b}$ is the denotation of a $k+1$ digit number,*

it follows that we must have

$$-2^{k+1}+1 \leq C_{a,b} \leq -2^k \leq C_{a,b} \leq 2^k-1 \leq C_{a,b} \leq 2^{k+1}-1 \qquad (*)$$

with the middle range of $(*)$ *being forced when* $C_{a,b} = O_{a,b}$*. The left range corresponds to a second* ***Range*** *Condition, and the right one to a third* ***Range*** *Condition. These mutually exclusive ranges of course cover all possible values of* $C_{a,b} \in \mathbb{Z}$*.*

Next the ***Overflow*** *conditions. From Theorem 4.3, if we know* $C_{a,b}$ *we can work out a binary representation* $dx \in \mathbb{B}^{k+2}$ *and show* $x = r$*; from* d *and* r *we can compute* σ_c*. Here we use Theorem 4.3 to obtain* $C_{a,b} = (\!|a -_{SA} b|\!)$ *and hence obtain* $dx \equiv a -_{SA} b$*. From Proposition 3.3 we have*

$$a -_{SA} b = (a -_{SA} b)_{k+1}(a -_{SA} b)^k = (a -_{SA} b)_{k+1}(a -_A b) = (a -_{SA} b)_{k+1} r$$

and therefore putting $\xi \stackrel{\text{def}}{=} \sigma_a + \overline{\sigma_b} \in \{0,1,2\}$ *we have*

$$(a -_{SA} b)_{k+1} = (\xi + \sigma_c) \; MOD \; 2 \qquad \sigma_r = (\xi + c_k) \; MOD \; 2$$

$$\sigma_c = (\xi + c_k) \; DIV \; 2$$

The values of the first two equations have just been computed; given the constraints on ξ*, it is then always possible to compute* σ_c *outright or in terms of* c_k*.*

The ***Sign Equation*** *conditions follow by further analysing*

$$C_{a,b} \stackrel{\text{def}}{=} (\!|a|\!) - (\!|b|\!) = (\!|dr|\!)$$

where $d \stackrel{\text{def}}{=} (a -_{SA} b)_{k+1}$*. Note that*

$$-\sigma_a 2^k + \sigma_b 2^k + (\!|a^{k-1}|\!) - (\!|b^{k-1}|\!) = -d2^{k+1} + \sigma_r 2^k + (\!|r^{k-1}|\!)$$

leading to

$$\xi \stackrel{\text{def}}{=} (\!|a^{k-1}|\!) - (\!|b^{k-1}|\!) = \underbrace{(\sigma_a - \sigma_b + \sigma_r - d2)}_{\alpha} 2^k + (\!|r^{k-1}|\!)$$

However, since ξ *is the difference* $(\!|a^{k-1}|\!) - (\!|b^{k-1}|\!)$ *we have* $-2^k + 1 \leq \xi \leq 2^k - 1$*. Using Theorem 2.2 we can easily see that we get contradictions if* $\alpha > 0$ *and* $\alpha < -1$*; further* $\alpha \in \{-1, 0\}$ *is consistent. One can observe that*

- ***Sign Equations*** *are instances of (analogues of)* $\alpha \in \{-1, 0\}$*; given any* a *and* b *we can compute* d *as above.*

- *For example, in the case of Theorem 4.10 and the third condition set, we know that* $d = 0$*, and in this particular case one can also determine that* $\sigma_r = 1$*. Hence* α *must equal* 0 *and we have* $\sigma_a - \sigma_b - \sigma_r = \alpha - 2\sigma_r = -2$*.*

Proof. Now we give some details of the proof of Theorem 4.10. The second set of conditions is proven pairwise equivalent.

We omit the easy details for $(3 \iff 4 \iff 5)$.

Now we show that $(1 \implies 2 \implies 3 \implies 1)$.

$(1 \implies 2)$: From 1 we have $S \stackrel{\text{def}}{=} (|a|) - (|b|) = -2^{k+1} + [[a -_A b]]$. Then note that from Lemma 2.2 we have $0 - (2^{k+1} - 1) \leq S$. Since $\sigma_r = 0$ we have $-2^{k+1} + [[a -_A b]] \leq -2^{k+1} + (2^k - 1)$. Thus 2 is true.

$(2 \implies 3)$: Let's suppose that 2 holds. Reasoning much as we did for the first condition set we must have

$$-2^{k+1} + 1 \leq (\xi \stackrel{\text{def}}{=} -R2^{k+1} + \sigma_r 2^k + [[r^{k-1}]]) \leq -2^k - 1$$

Since $\xi < 0$ we have $R = 1$. Therefore $\sigma_r 2^k + [[r^{k-1}]] \leq 2^k - 1$ and so $\sigma_r = 0$. Thus we have $1 = R = (\sigma_a + \overline{\sigma_b} + \sigma_c)\ MOD\ 2 \quad (*)$ and $0 = \sigma_r = (\sigma_a + \overline{\sigma_b} + c_k)\ MOD\ 2$. So $(\sigma_a + \overline{\sigma_b} + c_k) =$ 0 or 2. If the former, from $(*)$ we have $\sigma_c = 1$. But $\sigma_c \stackrel{\text{def}}{=} (\sigma_a + \overline{\sigma_b} + c_k)\ DIV\ 2 = 0$. Hence $(\sigma_a + \overline{\sigma_b} + c_k) = 2$. Therefore we still have $\sigma_c = 1$ and so $(*)$ implies $\sigma_a + \overline{\sigma_b} = 0$ or 2. In the former case we have $(0 + 0 + c_k) = 2$, and so finally we have $\sigma_a = \sigma_b = 1$. Hence 3 is true.

$(3 \implies 1)$: Suppose 3. We have to prove $\sigma_r = 0$ and $R \stackrel{\text{def}}{=} (\sigma_a + \overline{\sigma_b} + \sigma_c)\ MOD\ 2 = 1$. The former is immediate. For the latter we only need to show that $\sigma_c = 1$. But $\sigma_c = (\sigma_a + \overline{\sigma_b} + c_k)\ DIV\ 2 = (2 + c_k)\ DIV\ 2 = 1$. So 1 is true.

Thus the second set of conditions is proven pairwise equivalent. □

Theorem 4.11. *Let* $a, b \in \mathbb{B}^{k+1}$*. Recall from Figure 1 that* $\neg_{SA}(b) \stackrel{\text{def}}{=} \overline{b} +_{SA} \mathbb{1}$ *and* $\neg_A(b) \stackrel{\text{def}}{=} \overline{b} +_A \mathbb{1}$*. Then we have*

- $-(|b|) = (|\neg_{SA}(b)|)$
- $(|a -_{SA} b|) = (|\sigma_a a +_{SA} (\neg_{SA}(b))|)$
- $\sigma_{a -_{SA} b}(a -_{SA} b) = \sigma_a a +_{SA} (\neg_{SA}(b)) \in \mathbb{B}^{k+3}$ *and so* $a -_{SA} b = (\sigma_a a +_{SA} (\neg_{SA}(b)))^{k+2} \in \mathbb{B}^{k+2}$
- $abs(|b|) = (|\mathrm{abs}_{SA}(b)|)$

and

- $-(|b|) = (|\neg_A(b)|)$ *for* $b \neq \vec{10}$
- $(|a -_A b|) = (|a +_A (\neg_A(b))|)$

- $a -_A b = a +_A (\neg_A(b))$
- $abs(\lfloor b \rfloor) = (\lfloor \mathrm{abs}_{SA}(b) \rfloor)$ *for* $b \neq 1\vec{0}$

VERIFIED IN HOL

Proof. Consider the first three equations:

From Theorem 4.3 we have $(\lfloor \overline{b} \rfloor) + (\lfloor b \rfloor) = (\lfloor \overline{b} +_{SA} b \rfloor) = (\lfloor \vec{1} \rfloor) = -1$, and so $-(\lfloor b \rfloor) = (\lfloor \overline{b} \rfloor) + 1$; and we also have $(\lfloor \neg_{SA}(b) \rfloor) \stackrel{\text{def}}{=} (\lfloor \overline{b} +_{SA} 1 \rfloor) = (\lfloor \overline{b} \rfloor) + (\lfloor 1 \rfloor)$. So the first equation holds.

The second equation follows by using the first one along with simple calculations using the correctness equations 3 and 4 from Theorem 4.3, and Theorem 2.3 (sign extensions).

For the final equation note that by Theorem 2.3 (sign extensions) and the second equation, we have

$$(\lfloor \sigma_{a-_{SA}b}(a -_{SA} b) \rfloor) = (\lfloor a -_{SA} b \rfloor) = (\lfloor \sigma_a a +_{SA} (\neg_{SA}(b)) \rfloor)$$

so we are done by Theorem 2.3.

Now consider the second three equations:

The first equation will hold just in case $(\lfloor \overline{b} +_A \mathbb{1} \rfloor) = (\lfloor \overline{b} \rfloor) + (\lfloor \mathbb{1} \rfloor)$ is correct, using similar reasoning to the case of $\neg_{SA}$. But from Theorem 4.9 and the **Range** condition, this happens just in case

$$-2^k - 1 \leq (\lfloor \overline{b} \rfloor) \leq 2^k - 2$$

Then from Lemma 2.2 the left-hand inequality always holds, but the right-hand fails just in case $(\lfloor \overline{b} \rfloor) = 2^k - 1$, that is $b = 1\vec{0}$.

The second equation takes a little more work to establish. If $b = 1\vec{0}$ then we wish to establish that $(\lfloor a -_A 1\vec{0} \rfloor) = (\lfloor a +_A 1\vec{0} \rfloor)$. But we can easily check that $a -_A 1\vec{0} = a +_A 1\vec{0}$ directly from the DA and use Theorem 2.3.

If $b \neq 1\vec{0}$ then $-(\lfloor b \rfloor) = (\lfloor \neg_A(b) \rfloor)$. Hence $S = S'$ where

$$S \stackrel{\text{def}}{=} (\lfloor a \rfloor) - (\lfloor b \rfloor) = (\lfloor a \rfloor) + (\lfloor \neg_A(b) \rfloor) \stackrel{\text{def}}{=} S'$$

By using Theorem 4.10 we can see that we have S

$$-2^{k+1} + 1 \leq S \leq -2^k - 1 < -2^k \leq S \leq 2^k - 1 < 2^k \leq S \leq 2^{k+1} - 1$$

and in each case we have S equal to, respectively, the values below

$$(\lfloor 1(a -_A b) \rfloor) \leq (\lfloor a -_A b \rfloor) \leq (\lfloor 0(a -_A b) \rfloor)$$

Note that since $b \neq 1\vec{0}$ we have $-2^k+1 \leq (\lfloor b \rfloor)$ and so in fact $S \leq (2^k-1)+(2^k-1) = 2(2^k-1)$.

Similarly for S' using Theorem 4.9 (note the rightmost value!)

$$-2^{k+1} \leq S' \leq -2^k - 1 < -2^k \leq S' \leq 2^k - 1 < 2^k \leq S' \leq 2(2^k - 1)$$

and in each case we have S' equal to, respectively, the values below

$$(\!|1(a +_A (\neg_A(b)))|\!) \leq (\!|a +_A (\neg_A(b))|\!) \leq (\!|0(a +_A (\neg_A(b)))|\!)$$

Note that since $b \neq 1\vec{0}$ we have $\neg_A(b) \neq 1\vec{0}$ and so $-2^k + 1 \leq (\!|\neg_A(b)|\!)$ and so in fact $-2^{k+1} + 1 \leq -2^k + (-2^k + 1) \leq S'$.

Hence for each range of values, since S and S' are equal, from Theorem 2.3 we can deduce (care!) that the third equation holds overall, since it holds inside each of the three ranges. Thus the second equation is immediate.

□

Theorem 4.12. *Let $a, b, c \in \mathbb{B}^{k+1}$. $+_A$ and $+_{\alpha A}$ are commutative. $+_A$ is associative. In particular, and further*

- $a +_A (b +_A c) = (a +_A b) +_A c$
- $0a +_{ZA} (b +_{ZA} c) = (a +_{ZA} b) +_{ZA} 0c$

VERIFIED IN HOL

Proof. That $+_A$ and $+_{\alpha A}$ are commutative follows from the commutativity of + on $\mathbb{Z}$ and the definition of the *DA*.

Next we prove the second itemised equation, by showing each side has the same denotation and then using injectivity (Proposition 2.3). We calculate easily the denotations using Theorem 4.3 and Theorem 2.3 (sign extensions); equality follows from associativity of + on the integers.

To see that $+_A$ is associative, consider first the left hand side of $0a +_{ZA} (b +_{ZA} c) = (a +_{ZA} b) +_{ZA} 0c \in \mathbb{B}^{k+2}$, along with the steps below which follow from Proposition 3.3.

$$\begin{aligned} ((0a +_{ZA} (b +_{ZA} c))^{k+1})^k &= (0a +_A (b +_{ZA} c))^k && (1)\\ &= (0a)^k +_A (b +_{ZA} c)^k && (2)\\ &= a +_A (b +_A c) && (3) \end{aligned}$$

Now perform an identical equational argument starting with the right hand side. □

Theorem 5.1. *Let $a, b \in \mathbb{B}^{k+1}$. The following equations hold for every such a and b (total correctness)*

- $[[a]] + [[b]] = [[a +_M b]]$

- $[[a]] + [[b]] = [[a +_{ZM} b]]$

The following in-equations hold for any such a and b (total incorrectness)

- $[[a]] - [[b]] \neq (\!|a -_{M} b|\!)$
- $[[a]] - [[b]] \neq (\!|a -_{ZM} b|\!)$
- $[[a]] - [[b]] \neq (\!|a -_{SM} b|\!)$

Proof. The total correctness equation for $+_M$ can be proven by a straightforward induction; the proof is omitted. The equation for $+_{ZM}$ follows since it is easy to see that $a +_{ZM} b = 0(a +_M b)$ with DAC.

We show why the first of the in-equations holds. Let $(c, r) \stackrel{\text{def}}{=} DA_1(a, \overline{b})$, so that $\sigma_{a-_M b} = (0 + 0 + c_{k+1})\ MOD\ 2 = c_{k+1}$.

From Lemma 8.6 we have $DA_1(0a, 1\overline{b}) = (Kc, Rr)$ where $R = (0+1+c_{k+1})\ MOD\ 2 = \overline{c_{k+1}}$, so that $\sigma_{a-_{ZM} b} = \overline{c_{k+1}}$.

Hence the sign digits of $a -_M b$ and $a -_{ZM} b$ are complimentary, so they have unequal denotations. We saw in Theorem 4.3 that $[[a]] - [[b]] = (\!|a -_{ZA} b|\!)$ always, so the theorem follows.

□

Theorem 5.2. *Let $a, b \in \mathbb{B}^{k+1}$. The following conditions are* pairwise equivalent

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

as are

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

and moreover one condition must hold (along with equivalents of the condition) from each set.

The following conditions are pairwise equivalent

- ***Correctness:*** $(\!|a|\!) - (\!|b|\!) = (\!|a -_M b|\!)$
- ***Correctness:*** $(\!|a|\!) - (\!|b|\!) = (\!|a -_{SM} b|\!)$

- ***Sign Occurrences:*** *$\langle \sigma_a, \sigma_b, \sigma_r \rangle$ can be any one of the following four sign-occurrences $\langle 0, 1, 0/1 \rangle$: more simply, $\sigma_a \neq \sigma_b$.*

as are

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

and moreover one condition must hold (along with equivalents of the condition) from each set.

Proof. We give only exemplar proofs for two parts of the Theorem.

(A): Suppose that $(\!|a|\!) - (\!|b|\!) = (\!|a -_M b|\!)$. Let $(c, r) \stackrel{\text{def}}{=} DA_1(a, \overline{b})$ so that equivalently we have

$$-\sigma_a 2^k + \sigma_b 2^k + [\![a^{k-1}]\!] - [\![b^{k-1}]\!] = -c_{k+1} 2^{k+1} + \sigma_r 2^k + [\![r^{k-1}]\!] \qquad (*)$$

We know from Theorem 4.3 that $[\![a^{k-1}]\!] - [\![b^{k-1}]\!] = (\!|a^{k-1} -_{ZA} b^{k-1}|\!)$. To compute $a^{k-1} -_{ZA} b^{k-1}$ note that from Lemmas 8.5 and Lemma 8.6 we have $DA_1(0a^{k-1}, 1b^{k-1}) = (Kc^k, Rr^{k-1})$ where $R = (1 + c_k)\ MOD\ 2 = \overline{c_k}$ and so $a^{k-1} -_{ZA} b^{k-1} = \overline{c_k} r^{k-1}$. Thus

$$[\![a^{k-1}]\!] - [\![b^{k-1}]\!] = -\overline{c_k} 2^k + [\![r^{k-1}]\!]$$

and so $(*)$ holds just in case

$$-\sigma_a 2^k + \sigma_b 2^k + (-\overline{c_k} 2^k) = -c_{k+1} 2^{k+1} + \sigma_r 2^k$$

just in case

$$-\sigma_a + \sigma_b - \overline{c_k} = -c_{k+1} 2 + \sigma_r$$

and on expanding the definitions we have

$$-\sigma_a + \sigma_b - \overline{c_k} = -2 * (\sigma_a + \overline{\sigma_b} + c_k)\ DIV\ 2 + (\sigma_a + \overline{\sigma_b} + c_k)\ MOD\ 2 \qquad (\dagger)$$

If we show that if $\sigma_a = \sigma_b$ then $(\dagger)$ fails, and if $\sigma_a \neq \sigma_b$ then $(\dagger)$ holds, this part of the proof is complete. We look at two cases. Let $\sigma_a = \sigma_b$. Then we have

$$-\overline{c_k} = -2 * (1 + c_k)\ DIV\ 2 + (1 + c_k)\ MOD\ 2$$

that is

$$-\overline{c_k} = -2 * c_k + \overline{c_k}$$

which clearly is (always) false.

Now let $\sigma_a = 0$ and $\sigma_b = 1$. Then

$$1 - \overline{c_k} = -2 * c_k \; DIV \; 2 + c_k \; MOD \; 2 \iff 1 - \overline{c_k} = c_k \iff \mathsf{true}$$

(B): Suppose that $(\!|a|\!) + (\!|b|\!) = (\!|\pi_\sigma(a +_M b)|\!)$. This equation expands to

$$-\sigma_a 2^k - \sigma_b 2^k + c_k 2^k + [[r^{k-1}]] = -\overline{c_{k+1}} 2^{k+1} + \sigma_r 2^k + [[r^{k-1}]]$$

that is

$$-\sigma_a - \sigma_b + c_k = -2 * \overline{(\sigma_a + \sigma_b + c_k) \; DIV \; 2} + (\sigma_a + \sigma_b + c_k) \; MOD \; 2$$

and the rest of the reasoning is routine. □

Theorem 5.3. *Let* $1, a, b \in \mathbb{B}^{k+1}$. *Then we have*

- $-(\!|b|\!) = (\!|\neg_M(b)|\!)$ *for* $\sigma_b = 1$.
- $-(\!|b|\!) = (\!|\pi_\sigma(\neg_M(b))|\!)$ *for* $\sigma_b = 0$.
- $(\!|a -_M b|\!) = (\!|(0a +_M (\neg_M(b)))^{k+2}|\!)$.
- $0(a -_M b) = 0a +_M (\neg_M(b))$ *and so* $a -_M b = (0a +_M (\neg_M(b)))^{k+2}$
- $abs((\!|b|\!)) = (\!|\mathrm{abs}_M(b)|\!)$

Proof. For the second equation: Since $\sigma_b = 0$ we have $\sigma_{\overline{b}} = 1$, and $\sigma_{\mathbb{1}} = 0$. Thus we use Theorem 5.2 to see that $(\!|\pi_\sigma(\overline{b} +_M \mathbb{1})|\!) = (\!|\overline{b}|\!) + (\!|\mathbb{1}|\!)$. But of course by Theorem 4.3 we have $(\!|b|\!) + (\!|\overline{b}|\!) = (\!|\overline{b} +_{SA} b|\!) = (\!|\vec{1}|\!) = -1$ and so the theorem follows.

For the remaining equations, we only need to do work to prove $0(a -_M b) = 0a +_M (\neg_M b)$; the other equations follow immediately. We will consider cases according to sign digits. We make frequent appeals to Theorem 5.2.

$(\sigma_a = 0 \neq \sigma_b = 1)$: It follows that $\sigma_{\neg_M b} = 0$. Note also that, using the sign digit values, both $(\!|a -_M b|\!) = (\!|a|\!) - (\!|b|\!)$ and $0 \leq (\!|a|\!) - (\!|b|\!)$. Hence $\sigma_{a -_M b} = 0$. Then we have

$$\begin{aligned}
(\!|0(a -_M b)|\!) &= [[a -_M b]] \\
&= (\!|a -_M b|\!) \\
&= (\!|a|\!) - (\!|b|\!) \\
&= (\!|0a|\!) + (\!|\neg_M b|\!) \\
&= (\!|0a +_M (\neg_M b)|\!)
\end{aligned}$$

$(\sigma_a = 0 = \sigma_b = 0)$: We consider two cases, with $b = 0\vec{0}$ and $b \neq 0\vec{0}$.

In the first case, $\neg_M b = 1\vec{0}$. Further, proving the equality directly from the definitions is easy.

Moving to the second case, first note that $b = 0\vec{x}$ where $x \in \mathbb{B}^k$ has at least one digit equal to 1. One may see from DAC that $(\overline{b} +_M \mathbb{1})_{k+2} = 0$, that is $\sigma_{\neg_M b} = 0$. Now we have

$$(\!|a|\!) - (\!|b|\!) = (\!|\pi_\sigma(a -_M b)|\!) = [\![a -_M b]\!] - 2^{k+1}$$

by Theorem 5.2 and Theorem 2.3. With similar reasoning we have

$$\begin{aligned}
(\!|0(a -_M b)|\!) &= [\![a -_M b]\!] \\
&= (\!|a|\!) - (\!|b|\!) + 2^{k+1} \\
&= (\!|0a|\!) - (\!|b|\!) + 2^{k+1} \\
&= (\!|0a|\!) + (\!|\pi_\sigma(\neg_M(b))|\!) + 2^{k+1} \\
&= (\!|0a|\!) + [\![\neg_M(b)]\!] - 2^{k+1} + 2^{k+1} \\
&= (\!|0a|\!) + (\!|\neg_M(b)|\!) \\
&= (\!|0a +_M (\neg_M(b))|\!)
\end{aligned}$$

$(\sigma_a = 1 = \sigma_b = 1)$: We consider only the case $b \neq 0\vec{0}$ where we can show as before $\sigma_{\neg_M b} = 0$. Then have

$$(\!|a|\!) - (\!|b|\!) = (\!|\pi_\sigma(a -_M b)|\!) = [\![a -_M b]\!] - 2^{k+1}$$

and also

$$\begin{aligned}
(\!|0(a -_M b)|\!) &= [\![a -_M b]\!] \\
&= (\!|a|\!) - (\!|b|\!) + 2^{k+1} \\
&= (-2^k + [\![a^{k-1}]\!]) - (\!|b|\!) + 2^{k+1} \\
&= ([\![a^{k-1}]\!] + 2^k) - (\!|b|\!) \\
&= [\![a]\!] - (\!|b|\!) \\
&= (\!|0a|\!) - (\!|b|\!) \\
&= (\!|0a|\!) + (\!|\neg_M(b)|\!) \\
&= (\!|0a +_M (\neg_M b)|\!)
\end{aligned}$$

□