FORMALIZED MATHEMATICS
Volume 8, Number 1, 1999
University of Białystok

Full Subtracter Circuit. Part I

Katsumi Wasaki
Shinshu University
Nagano

Noboru Endou
Shinshu University
Nagano

Summary. We formalize the concept of the full subtracter circuit, define the structures of bit subtract/borrow units for binary operations, and prove the stability of the circuit.

MML Identifier: `FSCIRC_1`.

The terminology and notation used in this paper are introduced in the following papers: [11], [14], [13], [10], [17], [3], [4], [1], [16], [9], [12], [8], [6], [7], [5], [15], and [2].

1. Bit Subtract and Borrow Circuit

In this paper x, y, c are sets.

Let x, y, c be sets. The functor BitSubtracterOutput(x, y, c) yields an element of InnerVertices(2GatesCircStr(x, y, c, xor)) and is defined as follows:

(Def. 1) BitSubtracterOutput(x, y, c) = 2GatesCircOutput(x, y, c, xor).

Let x, y, c be sets. The functor BitSubtracterCirc(x, y, c) yields a strict Boolean circuit of 2GatesCircStr(x, y, c, xor) with denotation held in gates and is defined as follows:

(Def. 2) BitSubtracterCirc(x, y, c) = 2GatesCircuit(x, y, c, xor).

Let x, y, c be sets. The functor BorrowIStr(x, y, c) yields an unsplit non void strict non empty many sorted signature with arity held in gates and Boolean denotation held in gates and is defined by:

(Def. 3) BorrowIStr(x, y, c) = 1GateCircStr$(\langle x, y\rangle, \mathrm{and}_{2a})$+· 1GateCircStr$(\langle y, c\rangle, \mathrm{and}_2)$+· 1GateCircStr$(\langle x, c\rangle, \mathrm{and}_{2a})$.

ISSN 1426–2630

Let x, y, c be sets. The functor $\mathrm{BorrowStr}(x,y,c)$ yielding an unsplit non void strict non empty many sorted signature with arity held in gates and Boolean denotation held in gates is defined by:

(Def. 4) $\mathrm{BorrowStr}(x,y,c) = \mathrm{BorrowIStr}(x,y,c) +\cdot \mathrm{1GateCircStr}(\langle\langle\langle x,y\rangle, \mathrm{and}_{2a}\rangle, \langle\langle y,c\rangle, \mathrm{and}_2\rangle, \langle\langle x,c\rangle, \mathrm{and}_{2a}\rangle\rangle, \mathrm{or}_3)$.

Let x, y, c be sets. The functor $\mathrm{BorrowICirc}(x,y,c)$ yielding a strict Boolean circuit of $\mathrm{BorrowIStr}(x,y,c)$ with denotation held in gates is defined by:

(Def. 5) $\mathrm{BorrowICirc}(x,y,c) = \mathrm{1GateCircuit}(x,y,\mathrm{and}_{2a}) +\cdot \mathrm{1GateCircuit}(y,c,\mathrm{and}_2) +\cdot \mathrm{1GateCircuit}(x,c,\mathrm{and}_{2a})$.

The following propositions are true:

(1) $\mathrm{InnerVertices}(\mathrm{BorrowStr}(x,y,c))$ is a binary relation.

(2) For all non pair sets x, y, c holds $\mathrm{InputVertices}(\mathrm{BorrowStr}(x,y,c))$ has no pairs.

(3) For every state s of $\mathrm{BorrowICirc}(x,y,c)$ and for all elements a, b of *Boolean* such that $a = s(x)$ and $b = s(y)$ holds $(\mathrm{Following}(s))(\langle\langle x,y\rangle, \mathrm{and}_{2a}\rangle) = \neg a \wedge b$.

(4) For every state s of $\mathrm{BorrowICirc}(x,y,c)$ and for all elements a, b of *Boolean* such that $a = s(y)$ and $b = s(c)$ holds $(\mathrm{Following}(s))(\langle\langle y,c\rangle, \mathrm{and}_2\rangle) = a \wedge b$.

(5) For every state s of $\mathrm{BorrowICirc}(x,y,c)$ and for all elements a, b of *Boolean* such that $a = s(x)$ and $b = s(c)$ holds $(\mathrm{Following}(s))(\langle\langle x,c\rangle, \mathrm{and}_{2a}\rangle) = \neg a \wedge b$.

Let x, y, c be sets. The functor $\mathrm{BorrowOutput}(x,y,c)$ yields an element of $\mathrm{InnerVertices}(\mathrm{BorrowStr}(x,y,c))$ and is defined by:

(Def. 6) $\mathrm{BorrowOutput}(x,y,c) = \langle\langle\langle\langle x,y\rangle, \mathrm{and}_{2a}\rangle, \langle\langle y,c\rangle, \mathrm{and}_2\rangle, \langle\langle x,c\rangle, \mathrm{and}_{2a}\rangle\rangle, \mathrm{or}_3\rangle$.

Let x, y, c be sets. The functor $\mathrm{BorrowCirc}(x,y,c)$ yielding a strict Boolean circuit of $\mathrm{BorrowStr}(x,y,c)$ with denotation held in gates is defined by:

(Def. 7) $\mathrm{BorrowCirc}(x,y,c) = \mathrm{BorrowICirc}(x,y,c) +\cdot \mathrm{1GateCircuit}(\langle\langle x,y\rangle, \mathrm{and}_{2a}\rangle, \langle\langle y,c\rangle, \mathrm{and}_2\rangle, \langle\langle x,c\rangle, \mathrm{and}_{2a}\rangle, \mathrm{or}_3)$.

Next we state a number of propositions:

(6) $x \in$ the carrier of $\mathrm{BorrowStr}(x,y,c)$ and $y \in$ the carrier of $\mathrm{BorrowStr}(x,y,c)$ and $c \in$ the carrier of $\mathrm{BorrowStr}(x,y,c)$.

(7) $\langle\langle x,y\rangle, \mathrm{and}_{2a}\rangle \in \mathrm{InnerVertices}(\mathrm{BorrowStr}(x,y,c))$ and $\langle\langle y,c\rangle, \mathrm{and}_2\rangle \in \mathrm{InnerVertices}(\mathrm{BorrowStr}(x,y,c))$ and $\langle\langle x,c\rangle, \mathrm{and}_{2a}\rangle \in \mathrm{InnerVertices}(\mathrm{BorrowStr}(x,y,c))$.

(8) For all non pair sets x, y, c holds $x \in \mathrm{InputVertices}(\mathrm{BorrowStr}(x,y,c))$ and $y \in \mathrm{InputVertices}(\mathrm{BorrowStr}(x,y,c))$ and $c \in \mathrm{InputVertices}(\mathrm{BorrowStr}(x,y,c))$.

(9) For all non pair sets x, y, c holds InputVertices(BorrowStr$(x, y, c)) = \{x, y, c\}$ and InnerVertices(BorrowStr$(x, y, c)) = \{\langle\langle x, y\rangle, \mathrm{and}_{2a}\rangle, \langle\langle y, c\rangle, \mathrm{and}_2\rangle, \langle\langle x, c\rangle, \mathrm{and}_{2a}\rangle\} \cup \{\mathrm{BorrowOutput}(x, y, c)\}$.

(10) Let x, y, c be non pair sets, s be a state of BorrowCirc(x, y, c), and a_1, a_2 be elements of *Boolean*. If $a_1 = s(x)$ and $a_2 = s(y)$, then $(\mathrm{Following}(s))(\langle\langle x, y\rangle, \mathrm{and}_{2a}\rangle) = \neg a_1 \wedge a_2$.

(11) Let x, y, c be non pair sets, s be a state of BorrowCirc(x, y, c), and a_2, a_3 be elements of *Boolean*. If $a_2 = s(y)$ and $a_3 = s(c)$, then $(\mathrm{Following}(s))(\langle\langle y, c\rangle, \mathrm{and}_2\rangle) = a_2 \wedge a_3$.

(12) Let x, y, c be non pair sets, s be a state of BorrowCirc(x, y, c), and a_1, a_3 be elements of *Boolean*. If $a_1 = s(x)$ and $a_3 = s(c)$, then $(\mathrm{Following}(s))(\langle\langle x, c\rangle, \mathrm{and}_{2a}\rangle) = \neg a_1 \wedge a_3$.

(13) Let x, y, c be non pair sets, s be a state of BorrowCirc(x, y, c), and a_1, a_2, a_3 be elements of *Boolean*. If $a_1 = s(\langle\langle x, y\rangle, \mathrm{and}_{2a}\rangle)$ and $a_2 = s(\langle\langle y, c\rangle, \mathrm{and}_2\rangle)$ and $a_3 = s(\langle\langle x, c\rangle, \mathrm{and}_{2a}\rangle)$, then $(\mathrm{Following}(s))(\mathrm{BorrowOutput}(x, y, c)) = a_1 \vee a_2 \vee a_3$.

(14) Let x, y, c be non pair sets, s be a state of BorrowCirc(x, y, c), and a_1, a_2 be elements of *Boolean*. If $a_1 = s(x)$ and $a_2 = s(y)$, then $(\mathrm{Following}(s, 2))(\langle\langle x, y\rangle, \mathrm{and}_{2a}\rangle) = \neg a_1 \wedge a_2$.

(15) Let x, y, c be non pair sets, s be a state of BorrowCirc(x, y, c), and a_2, a_3 be elements of *Boolean*. If $a_2 = s(y)$ and $a_3 = s(c)$, then $(\mathrm{Following}(s, 2))(\langle\langle y, c\rangle, \mathrm{and}_2\rangle) = a_2 \wedge a_3$.

(16) Let x, y, c be non pair sets, s be a state of BorrowCirc(x, y, c), and a_1, a_3 be elements of *Boolean*. If $a_1 = s(x)$ and $a_3 = s(c)$, then $(\mathrm{Following}(s, 2))(\langle\langle x, c\rangle, \mathrm{and}_{2a}\rangle) = \neg a_1 \wedge a_3$.

(17) Let x, y, c be non pair sets, s be a state of BorrowCirc(x, y, c), and a_1, a_2, a_3 be elements of *Boolean*. If $a_1 = s(x)$ and $a_2 = s(y)$ and $a_3 = s(c)$, then $(\mathrm{Following}(s, 2))(\mathrm{BorrowOutput}(x, y, c)) = \neg a_1 \wedge a_2 \vee a_2 \wedge a_3 \vee \neg a_1 \wedge a_3$.

(18) For all non pair sets x, y, c and for every state s of BorrowCirc(x, y, c) holds Following$(s, 2)$ is stable.

2. Bit Subtracter with Borrow Circuit

Let x, y, c be sets. The functor BitSubtracterWithBorrowStr(x, y, c) yields an unsplit non void strict non empty many sorted signature with arity held in gates and Boolean denotation held in gates and is defined by:

(Def. 8) BitSubtracterWithBorrowStr(x, y, c) = 2GatesCircStr(x, y, c, xor) $+\cdot$ BorrowStr(x, y, c).

The following propositions are true:

(19) For all non pair sets x, y, c holds InputVertices(BitSubtracterWithBorrowStr(x, y, c)) $= \{x, y, c\}$.

(20) For all non pair sets x, y, c holds InnerVertices(BitSubtracterWithBorrowStr(x, y, c)) $= \{\langle\langle x, y\rangle, \mathrm{xor}\rangle$, 2GatesCircOutput$(x, y, c, \mathrm{xor})\} \cup \{\langle\langle x, y\rangle, \mathrm{and}_{2a}\rangle, \langle\langle y, c\rangle, \mathrm{and}_2\rangle, \langle\langle x, c\rangle, \mathrm{and}_{2a}\rangle\} \cup \{$BorrowOutput$(x, y, c)\}$.

(21) Let S be a non empty many sorted signature. Suppose $S =$ BitSubtracterWithBorrowStr(x, y, c). Then $x \in$ the carrier of S and $y \in$ the carrier of S and $c \in$ the carrier of S.

Let x, y, c be sets. The functor BitSubtracterWithBorrowCirc(x, y, c) yields a strict Boolean circuit of BitSubtracterWithBorrowStr(x, y, c) with denotation held in gates and is defined as follows:

(Def. 9) BitSubtracterWithBorrowCirc$(x, y, c) =$ BitSubtracterCirc(x, y, c) $+\cdot$ BorrowCirc(x, y, c).

We now state several propositions:

(22) InnerVertices(BitSubtracterWithBorrowStr(x, y, c)) is a binary relation.

(23) For all non pair sets x, y, c holds InputVertices(BitSubtracterWithBorrowStr(x, y, c)) has no pairs.

(24) BitSubtracterOutput$(x, y, c) \in$ InnerVertices(BitSubtracterWithBorrowStr(x, y, c)) and BorrowOutput $(x, y, c) \in$ InnerVertices(BitSubtracterWithBorrowStr(x, y, c)).

(25) Let x, y, c be non pair sets, s be a state of BitSubtracterWithBorrowCirc (x, y, c), and a_1, a_2, a_3 be elements of *Boolean*. Suppose $a_1 = s(x)$ and $a_2 = s(y)$ and $a_3 = s(c)$. Then (Following$(s, 2)$)(BitSubtracterOutput(x, y, c)) $= a_1 \oplus a_2 \oplus a_3$ and (Following$(s, 2)$)(BorrowOutput(x, y, c)) $= \neg a_1 \wedge a_2 \vee a_2 \wedge a_3 \vee \neg a_1 \wedge a_3$.

(26) For all non pair sets x, y, c and for every state s of BitSubtracterWithBorrowCirc(x, y, c) holds Following$(s, 2)$ is stable.

References

[1] Grzegorz Bancerek and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Formalized Mathematics*, 1(**1**):107–114, 1990.

[2] Grzegorz Bancerek and Yatsuka Nakamura. Full adder circuit. Part I. *Formalized Mathematics*, 5(**3**):367–380, 1996.

[3] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.

[4] Czesław Byliński. Functions from a set to a set. *Formalized Mathematics*, 1(**1**):153–164, 1990.

[5] Yatsuka Nakamura and Grzegorz Bancerek. Combining of circuits. *Formalized Mathematics*, 5(**2**):283–295, 1996.

[6] Yatsuka Nakamura, Piotr Rudnicki, Andrzej Trybulec, and Pauline N. Kawamoto. Introduction to circuits, I. *Formalized Mathematics*, 5(**2**):227–232, 1996.

[7] Yatsuka Nakamura, Piotr Rudnicki, Andrzej Trybulec, and Pauline N. Kawamoto. Introduction to circuits, II. *Formalized Mathematics*, 5(**2**):273–278, 1996.

[8] Yatsuka Nakamura, Piotr Rudnicki, Andrzej Trybulec, and Pauline N. Kawamoto. Preliminaries to circuits, II. *Formalized Mathematics*, 5(**2**):215–220, 1996.
[9] Takaya Nishiyama and Yasuho Mizuhara. Binary arithmetics. *Formalized Mathematics*, 4(**1**):83–86, 1993.
[10] Andrzej Trybulec. Enumerated sets. *Formalized Mathematics*, 1(**1**):25–34, 1990.
[11] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[12] Andrzej Trybulec. Many sorted algebras. *Formalized Mathematics*, 5(**1**):37–42, 1996.
[13] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.
[14] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[15] Katsumi Wasaki and Pauline N. Kawamoto. 2's complement circuit. *Formalized Mathematics*, 6(**2**):189–197, 1997.
[16] Edmund Woronowicz. Many–argument relations. *Formalized Mathematics*, 1(**4**):733–737, 1990.
[17] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received March 13, 1999