JOURNAL OF FORMALIZED MATHEMATICS
Volume 11, Released 1999, Published 2003
Inst. of Computer Science, Univ. 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.

WWW: http://mizar.org/JFM/Vol11/fscirc_1.html

The articles [9], [8], [11], [13], [3], [14], [1], [10], [5], [6], [4], [12], [2], and [7] provide the notation and terminology for this paper.

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 by:

(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) yielding an unsplit non void strict non empty many sorted signature with arity held in gates and Boolean denotation held in gates is defined as follows:

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

Let x, y, c be sets. The functor BorrowStr(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 as follows:

(Def. 4) BorrowStr(x,y,c) = BorrowIStr(x,y,c) $+\cdot$ 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 BorrowICirc(x,y,c) yielding a strict Boolean circuit of BorrowIStr(x,y,c) with denotation held in gates is defined as follows:

(Def. 5) BorrowICirc(x,y,c) = 1GateCircuit(x,y,and_{2a}) $+\cdot$ 1GateCircuit(y,c,and_2) $+\cdot$ 1GateCircuit(x,c,and_{2a}).

The following propositions are true:

(1) InnerVertices(BorrowStr(x,y,c)) is a binary relation.

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

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

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

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

Let x, y, c be sets. The functor BorrowOutput(x,y,c) yielding an element of InnerVertices(BorrowStr(x,y,c)) is defined as follows:

(Def. 6) 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 BorrowCirc(x,y,c) yielding a strict Boolean circuit of BorrowStr(x,y,c) with denotation held in gates is defined as follows:

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

The following propositions are true:

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

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

(8) For all non pair sets x, y, c holds $x \in$ InputVertices(BorrowStr(x,y,c)) and $y \in$ InputVertices(BorrowStr(x,y,c)) and $c \in$ InputVertices(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)) $= \{\langle\langle x,y\rangle, \mathrm{and}_{2a}\rangle, \langle\langle y,c\rangle, \mathrm{and}_2\rangle, \langle\langle x,c\rangle, \mathrm{and}_{2a}\rangle\} \cup$ {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 (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 (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 (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 (Following$(s)$)(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 (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 (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 (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 (Following$(s,2)$)(BorrowOutput(x,y,c)) $= \neg a_1 \wedge a_2 \vee a_2 \wedge a_3 \vee \neg a_1 \wedge a_3$.

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

We now state three propositions:

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

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

(21) Let S be a non empty many sorted signature. Suppose $S = \mathrm{BitSubtracterWithBorrowStr}(x,y,c)$. 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 $\mathrm{BitSubtracterWithBorrowCirc}(x,y,c)$ yielding a strict Boolean circuit of $\mathrm{BitSubtracterWithBorrowStr}(x,y,c)$ with denotation held in gates is defined by:

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

The following propositions are true:

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

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

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

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

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

References

[1] Grzegorz Bancerek. The fundamental properties of natural numbers. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/nat_1.html`.

[2] Grzegorz Bancerek and Yatsuka Nakamura. Full adder circuit. Part I. *Journal of Formalized Mathematics*, 7, 1995. `http://mizar.org/JFM/Vol7/facirc_1.html`.

[3] Czesław Byliński. Functions and their basic properties. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/funct_1.html`.

[4] Yatsuka Nakamura and Grzegorz Bancerek. Combining of circuits. *Journal of Formalized Mathematics*, 7, 1995. `http://mizar.org/JFM/Vol7/circcomb.html`.

[5] Yatsuka Nakamura, Piotr Rudnicki, Andrzej Trybulec, and Pauline N. Kawamoto. Preliminaries to circuits, II. *Journal of Formalized Mathematics*, 6, 1994. `http://mizar.org/JFM/Vol6/msafree2.html`.

[6] Yatsuka Nakamura, Piotr Rudnicki, Andrzej Trybulec, and Pauline N. Kawamoto. Introduction to circuits, II. *Journal of Formalized Mathematics*, 7, 1995. `http://mizar.org/JFM/Vol7/circuit2.html`.

[7] Takaya Nishiyama and Yasuho Mizuhara. Binary arithmetics. *Journal of Formalized Mathematics*, 5, 1993. `http://mizar.org/JFM/Vol5/binarith.html`.

[8] Andrzej Trybulec. Enumerated sets. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/enumset1.html`.

[9] Andrzej Trybulec. Tarski Grothendieck set theory. *Journal of Formalized Mathematics*, Axiomatics, 1989. http://mizar.org/JFM/Axiomatics/tarski.html.

[10] Andrzej Trybulec. Many sorted algebras. *Journal of Formalized Mathematics*, 6, 1994. http://mizar.org/JFM/Vol6/msualg_1.html.

[11] Zinaida Trybulec. Properties of subsets. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/subset_1.html.

[12] Katsumi Wasaki and Pauline N. Kawamoto. 2's complement circuit. *Journal of Formalized Mathematics*, 8, 1996. http://mizar.org/JFM/Vol8/twoscomp.html.

[13] Edmund Woronowicz. Relations and their basic properties. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/relat_1.html.

[14] Edmund Woronowicz. Many-argument relations. *Journal of Formalized Mathematics*, 2, 1990. http://mizar.org/JFM/Vol2/margrel1.html.

Received March 13, 1999

Published January 2, 2004