JOURNAL OF FORMALIZED MATHEMATICS
Volume 8, Released 1996, Published 2003
Inst. of Computer Science, Univ. of Białystok

2's Complement Circuit

Katsumi Wasaki
National College of Technology
Nagano

Pauline N. Kawamoto
Shinshu University
Nagano

Summary. This article introduces various Boolean operators which are used in discussing the properties and stability of a 2's complement circuit. We present the definitions and related theorems for the following logical operators which include negative input/output: 'and2a', 'or2a', 'xor2a' and 'nand2a', 'nor2a', etc. We formalize the concept of a 2's complement circuit, define the structures of complementors/incrementors for binary operations, and prove the stability of the circuit.

MML Identifier: TWOSCOMP.

WWW: http://mizar.org/JFM/Vol8/twoscomp.html

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

1. Boolean Operators

Let S be an unsplit non void non empty many sorted signature, let A be a Boolean circuit of S, let s be a state of A, and let v be a vertex of S. Then $s(v)$ is an element of *Boolean*.

The function and_2 from $\mathit{Boolean}^2$ into *Boolean* is defined by:

(Def. 1) For all elements x, y of *Boolean* holds $\mathrm{and}_2(\langle x,y\rangle) = x \wedge y$.

The function and_{2a} from $\mathit{Boolean}^2$ into *Boolean* is defined as follows:

(Def. 2) For all elements x, y of *Boolean* holds $(\mathrm{and}_{2a})(\langle x,y\rangle) = \neg x \wedge y$.

The function and_{2b} from $\mathit{Boolean}^2$ into *Boolean* is defined as follows:

(Def. 3) For all elements x, y of *Boolean* holds $(\mathrm{and}_{2b})(\langle x,y\rangle) = \neg x \wedge \neg y$.

The function nand_2 from $\mathit{Boolean}^2$ into *Boolean* is defined by:

(Def. 4) For all elements x, y of *Boolean* holds $\mathrm{nand}_2(\langle x,y\rangle) = \neg(x \wedge y)$.

The function nand_{2a} from $\mathit{Boolean}^2$ into *Boolean* is defined by:

(Def. 5) For all elements x, y of *Boolean* holds $(\mathrm{nand}_{2a})(\langle x,y\rangle) = \neg(\neg x \wedge y)$.

The function nand_{2b} from $\mathit{Boolean}^2$ into *Boolean* is defined by:

(Def. 6) For all elements x, y of *Boolean* holds $(\mathrm{nand}_{2b})(\langle x,y\rangle) = \neg(\neg x \wedge \neg y)$.

The function or_2 from $\mathit{Boolean}^2$ into *Boolean* is defined as follows:

(Def. 7) For all elements x, y of *Boolean* holds $\mathrm{or}_2(\langle x,y\rangle) = x \vee y$.

The function or_{2a} from $Boolean^2$ into $Boolean$ is defined by:

(Def. 8) For all elements x, y of $Boolean$ holds $(\mathrm{or}_{2a})(\langle x,y\rangle) = \neg x \vee y$.

The function or_{2b} from $Boolean^2$ into $Boolean$ is defined by:

(Def. 9) For all elements x, y of $Boolean$ holds $(\mathrm{or}_{2b})(\langle x,y\rangle) = \neg x \vee \neg y$.

The function nor_2 from $Boolean^2$ into $Boolean$ is defined by:

(Def. 10) For all elements x, y of $Boolean$ holds $\mathrm{nor}_2(\langle x,y\rangle) = \neg(x \vee y)$.

The function nor_{2a} from $Boolean^2$ into $Boolean$ is defined by:

(Def. 11) For all elements x, y of $Boolean$ holds $(\mathrm{nor}_{2a})(\langle x,y\rangle) = \neg(\neg x \vee y)$.

The function nor_{2b} from $Boolean^2$ into $Boolean$ is defined as follows:

(Def. 12) For all elements x, y of $Boolean$ holds $(\mathrm{nor}_{2b})(\langle x,y\rangle) = \neg(\neg x \vee \neg y)$.

The function xor_2 from $Boolean^2$ into $Boolean$ is defined as follows:

(Def. 13) For all elements x, y of $Boolean$ holds $\mathrm{xor}_2(\langle x,y\rangle) = x \oplus y$.

The function xor_{2a} from $Boolean^2$ into $Boolean$ is defined as follows:

(Def. 14) For all elements x, y of $Boolean$ holds $(\mathrm{xor}_{2a})(\langle x,y\rangle) = \neg x \oplus y$.

The function xor_{2b} from $Boolean^2$ into $Boolean$ is defined by:

(Def. 15) For all elements x, y of $Boolean$ holds $(\mathrm{xor}_{2b})(\langle x,y\rangle) = \neg x \oplus \neg y$.

The following propositions are true:

(3)[1] For all elements x, y of $Boolean$ holds $\mathrm{and}_2(\langle x,y\rangle) = x \wedge y$ and $(\mathrm{and}_{2a})(\langle x,y\rangle) = \neg x \wedge y$ and $(\mathrm{and}_{2b})(\langle x,y\rangle) = \neg x \wedge \neg y$.

(4) For all elements x, y of $Boolean$ holds $\mathrm{nand}_2(\langle x,y\rangle) = \neg(x \wedge y)$ and $(\mathrm{nand}_{2a})(\langle x,y\rangle) = \neg(\neg x \wedge y)$ and $(\mathrm{nand}_{2b})(\langle x,y\rangle) = \neg(\neg x \wedge \neg y)$.

(5) For all elements x, y of $Boolean$ holds $\mathrm{or}_2(\langle x,y\rangle) = x \vee y$ and $(\mathrm{or}_{2a})(\langle x,y\rangle) = \neg x \vee y$ and $(\mathrm{or}_{2b})(\langle x,y\rangle) = \neg x \vee \neg y$.

(6) For all elements x, y of $Boolean$ holds $\mathrm{nor}_2(\langle x,y\rangle) = \neg(x \vee y)$ and $(\mathrm{nor}_{2a})(\langle x,y\rangle) = \neg(\neg x \vee y)$ and $(\mathrm{nor}_{2b})(\langle x,y\rangle) = \neg(\neg x \vee \neg y)$.

(7) For all elements x, y of $Boolean$ holds $\mathrm{xor}_2(\langle x,y\rangle) = x \oplus y$ and $(\mathrm{xor}_{2a})(\langle x,y\rangle) = \neg x \oplus y$ and $(\mathrm{xor}_{2b})(\langle x,y\rangle) = \neg x \oplus \neg y$.

(8) For all elements x, y of $Boolean$ holds $\mathrm{and}_2(\langle x,y\rangle) = (\mathrm{nor}_{2b})(\langle x,y\rangle)$ and $(\mathrm{and}_{2a})(\langle x,y\rangle) = (\mathrm{nor}_{2a})(\langle y,x\rangle)$ and $(\mathrm{and}_{2b})(\langle x,y\rangle) = \mathrm{nor}_2(\langle x,y\rangle)$.

(9) For all elements x, y of $Boolean$ holds $\mathrm{or}_2(\langle x,y\rangle) = (\mathrm{nand}_{2b})(\langle x,y\rangle)$ and $(\mathrm{or}_{2a})(\langle x,y\rangle) = (\mathrm{nand}_{2a})(\langle y,x\rangle)$ and $(\mathrm{or}_{2b})(\langle x,y\rangle) = \mathrm{nand}_2(\langle x,y\rangle)$.

(10) For all elements x, y of $Boolean$ holds $(\mathrm{xor}_{2b})(\langle x,y\rangle) = \mathrm{xor}_2(\langle x,y\rangle)$.

(11) $\mathrm{and}_2(\langle 0,0\rangle) = 0$ and $\mathrm{and}_2(\langle 0,1\rangle) = 0$ and $\mathrm{and}_2(\langle 1,0\rangle) = 0$ and $\mathrm{and}_2(\langle 1,1\rangle) = 1$ and $(\mathrm{and}_{2a})(\langle 0,0\rangle) = 0$ and $(\mathrm{and}_{2a})(\langle 0,1\rangle) = 1$ and $(\mathrm{and}_{2a})(\langle 1,0\rangle) = 0$ and $(\mathrm{and}_{2a})(\langle 1,1\rangle) = 0$ and $(\mathrm{and}_{2b})(\langle 0,0\rangle) = 1$ and $(\mathrm{and}_{2b})(\langle 0,1\rangle) = 0$ and $(\mathrm{and}_{2b})(\langle 1,0\rangle) = 0$ and $(\mathrm{and}_{2b})(\langle 1,1\rangle) = 0$.

[1] The propositions (1) and (2) have been removed.

(12) $\mathrm{or}_2(\langle 0,0\rangle) = 0$ and $\mathrm{or}_2(\langle 0,1\rangle) = 1$ and $\mathrm{or}_2(\langle 1,0\rangle) = 1$ and $\mathrm{or}_2(\langle 1,1\rangle) = 1$ and $(\mathrm{or}_{2a})(\langle 0,0\rangle) = 1$ and $(\mathrm{or}_{2a})(\langle 0,1\rangle) = 1$ and $(\mathrm{or}_{2a})(\langle 1,0\rangle) = 0$ and $(\mathrm{or}_{2a})(\langle 1,1\rangle) = 1$ and $(\mathrm{or}_{2b})(\langle 0,0\rangle) = 1$ and $(\mathrm{or}_{2b})(\langle 0,1\rangle) = 1$ and $(\mathrm{or}_{2b})(\langle 1,0\rangle) = 1$ and $(\mathrm{or}_{2b})(\langle 1,1\rangle) = 0$.

(13) $\mathrm{xor}_2(\langle 0,0\rangle) = 0$ and $\mathrm{xor}_2(\langle 0,1\rangle) = 1$ and $\mathrm{xor}_2(\langle 1,0\rangle) = 1$ and $\mathrm{xor}_2(\langle 1,1\rangle) = 0$ and $(\mathrm{xor}_{2a})(\langle 0,0\rangle) = 1$ and $(\mathrm{xor}_{2a})(\langle 0,1\rangle) = 0$ and $(\mathrm{xor}_{2a})(\langle 1,0\rangle) = 0$ and $(\mathrm{xor}_{2a})(\langle 1,1\rangle) = 1$.

The function and_3 from $Boolean^3$ into *Boolean* is defined as follows:

(Def. 16) For all elements x, y, z of *Boolean* holds $\mathrm{and}_3(\langle x,y,z\rangle) = x \wedge y \wedge z$.

The function and_{3a} from $Boolean^3$ into *Boolean* is defined by:

(Def. 17) For all elements x, y, z of *Boolean* holds $(\mathrm{and}_{3a})(\langle x,y,z\rangle) = \neg x \wedge y \wedge z$.

The function and_{3b} from $Boolean^3$ into *Boolean* is defined by:

(Def. 18) For all elements x, y, z of *Boolean* holds $(\mathrm{and}_{3b})(\langle x,y,z\rangle) = \neg x \wedge \neg y \wedge z$.

The function and_{3c} from $Boolean^3$ into *Boolean* is defined as follows:

(Def. 19) For all elements x, y, z of *Boolean* holds $(\mathrm{and}_{3c})(\langle x,y,z\rangle) = \neg x \wedge \neg y \wedge \neg z$.

The function nand_3 from $Boolean^3$ into *Boolean* is defined as follows:

(Def. 20) For all elements x, y, z of *Boolean* holds $\mathrm{nand}_3(\langle x,y,z\rangle) = \neg(x \wedge y \wedge z)$.

The function nand_{3a} from $Boolean^3$ into *Boolean* is defined as follows:

(Def. 21) For all elements x, y, z of *Boolean* holds $(\mathrm{nand}_{3a})(\langle x,y,z\rangle) = \neg(\neg x \wedge y \wedge z)$.

The function nand_{3b} from $Boolean^3$ into *Boolean* is defined by:

(Def. 22) For all elements x, y, z of *Boolean* holds $(\mathrm{nand}_{3b})(\langle x,y,z\rangle) = \neg(\neg x \wedge \neg y \wedge z)$.

The function nand_{3c} from $Boolean^3$ into *Boolean* is defined by:

(Def. 23) For all elements x, y, z of *Boolean* holds $(\mathrm{nand}_{3c})(\langle x,y,z\rangle) = \neg(\neg x \wedge \neg y \wedge \neg z)$.

The function or_3 from $Boolean^3$ into *Boolean* is defined by:

(Def. 24) For all elements x, y, z of *Boolean* holds $\mathrm{or}_3(\langle x,y,z\rangle) = x \vee y \vee z$.

The function or_{3a} from $Boolean^3$ into *Boolean* is defined as follows:

(Def. 25) For all elements x, y, z of *Boolean* holds $(\mathrm{or}_{3a})(\langle x,y,z\rangle) = \neg x \vee y \vee z$.

The function or_{3b} from $Boolean^3$ into *Boolean* is defined by:

(Def. 26) For all elements x, y, z of *Boolean* holds $(\mathrm{or}_{3b})(\langle x,y,z\rangle) = \neg x \vee \neg y \vee z$.

The function or_{3c} from $Boolean^3$ into *Boolean* is defined by:

(Def. 27) For all elements x, y, z of *Boolean* holds $(\mathrm{or}_{3c})(\langle x,y,z\rangle) = \neg x \vee \neg y \vee \neg z$.

The function nor_3 from $Boolean^3$ into *Boolean* is defined as follows:

(Def. 28) For all elements x, y, z of *Boolean* holds $\mathrm{nor}_3(\langle x,y,z\rangle) = \neg(x \vee y \vee z)$.

The function nor_{3a} from $Boolean^3$ into *Boolean* is defined by:

(Def. 29) For all elements x, y, z of *Boolean* holds $(\mathrm{nor}_{3a})(\langle x,y,z\rangle) = \neg(\neg x \vee y \vee z)$.

The function nor_{3b} from $Boolean^3$ into *Boolean* is defined as follows:

(Def. 30) For all elements x, y, z of *Boolean* holds $(\mathrm{nor}_{3b})(\langle x,y,z\rangle) = \neg(\neg x \vee \neg y \vee z)$.

The function nor_{3c} from $Boolean^3$ into *Boolean* is defined as follows:

(Def. 31) For all elements x, y, z of *Boolean* holds $(\mathrm{nor}_{3c})(\langle x,y,z\rangle) = \neg(\neg x \vee \neg y \vee \neg z)$.

The function xor_3 from $Boolean^3$ into *Boolean* is defined by:

(Def. 32) For all elements x, y, z of *Boolean* holds $\mathrm{xor}_3(\langle x,y,z\rangle) = x \oplus y \oplus z$.

The following propositions are true:

(14) For all elements x, y, z of *Boolean* holds $\mathrm{and}_3(\langle x,y,z\rangle) = x \wedge y \wedge z$ and $(\mathrm{and}_{3a})(\langle x,y,z\rangle) = \neg x \wedge y \wedge z$ and $(\mathrm{and}_{3b})(\langle x,y,z\rangle) = \neg x \wedge \neg y \wedge z$ and $(\mathrm{and}_{3c})(\langle x,y,z\rangle) = \neg x \wedge \neg y \wedge \neg z$.

(15) Let x, y, z be elements of *Boolean*. Then $\mathrm{nand}_3(\langle x,y,z\rangle) = \neg(x \wedge y \wedge z)$ and $(\mathrm{nand}_{3a})(\langle x,y,z\rangle) = \neg(\neg x \wedge y \wedge z)$ and $(\mathrm{nand}_{3b})(\langle x,y,z\rangle) = \neg(\neg x \wedge \neg y \wedge z)$ and $(\mathrm{nand}_{3c})(\langle x,y,z\rangle) = \neg(\neg x \wedge \neg y \wedge \neg z)$.

(16) For all elements x, y, z of *Boolean* holds $\mathrm{or}_3(\langle x,y,z\rangle) = x \vee y \vee z$ and $(\mathrm{or}_{3a})(\langle x,y,z\rangle) = \neg x \vee y \vee z$ and $(\mathrm{or}_{3b})(\langle x,y,z\rangle) = \neg x \vee \neg y \vee z$ and $(\mathrm{or}_{3c})(\langle x,y,z\rangle) = \neg x \vee \neg y \vee \neg z$.

(17) Let x, y, z be elements of *Boolean*. Then $\mathrm{nor}_3(\langle x,y,z\rangle) = \neg(x \vee y \vee z)$ and $(\mathrm{nor}_{3a})(\langle x,y,z\rangle) = \neg(\neg x \vee y \vee z)$ and $(\mathrm{nor}_{3b})(\langle x,y,z\rangle) = \neg(\neg x \vee \neg y \vee z)$ and $(\mathrm{nor}_{3c})(\langle x,y,z\rangle) = \neg(\neg x \vee \neg y \vee \neg z)$.

(19)[2] For all elements x, y, z of *Boolean* holds $\mathrm{and}_3(\langle x,y,z\rangle) = (\mathrm{nor}_{3c})(\langle x,y,z\rangle)$ and $(\mathrm{and}_{3a})(\langle x,y,z\rangle) = (\mathrm{nor}_{3b})(\langle z,y,x\rangle)$ and $(\mathrm{and}_{3b})(\langle x,y,z\rangle) = (\mathrm{nor}_{3a})(\langle z,y,x\rangle)$ and $(\mathrm{and}_{3c})(\langle x,y,z\rangle) = \mathrm{nor}_3(\langle x,y,z\rangle)$.

(20) For all elements x, y, z of *Boolean* holds $\mathrm{or}_3(\langle x,y,z\rangle) = (\mathrm{nand}_{3c})(\langle x,y,z\rangle)$ and $(\mathrm{or}_{3a})(\langle x,y,z\rangle) = (\mathrm{nand}_{3b})(\langle z,y,x\rangle)$ and $(\mathrm{or}_{3b})(\langle x,y,z\rangle) = (\mathrm{nand}_{3a})(\langle z,y,x\rangle)$ and $(\mathrm{or}_{3c})(\langle x,y,z\rangle) = \mathrm{nand}_3(\langle x,y,z\rangle)$.

(21) $\mathrm{and}_3(\langle 0,0,0\rangle) = 0$ and $\mathrm{and}_3(\langle 0,0,1\rangle) = 0$ and $\mathrm{and}_3(\langle 0,1,0\rangle) = 0$ and $\mathrm{and}_3(\langle 0,1,1\rangle) = 0$ and $\mathrm{and}_3(\langle 1,0,0\rangle) = 0$ and $\mathrm{and}_3(\langle 1,0,1\rangle) = 0$ and $\mathrm{and}_3(\langle 1,1,0\rangle) = 0$ and $\mathrm{and}_3(\langle 1,1,1\rangle) = 1$.

(22) $(\mathrm{and}_{3a})(\langle 0,0,0\rangle) = 0$ and $(\mathrm{and}_{3a})(\langle 0,0,1\rangle) = 0$ and $(\mathrm{and}_{3a})(\langle 0,1,0\rangle) = 0$ and $(\mathrm{and}_{3a})(\langle 0,1,1\rangle) = 1$ and $(\mathrm{and}_{3a})(\langle 1,0,0\rangle) = 0$ and $(\mathrm{and}_{3a})(\langle 1,0,1\rangle) = 0$ and $(\mathrm{and}_{3a})(\langle 1,1,0\rangle) = 0$ and $(\mathrm{and}_{3a})(\langle 1,1,1\rangle) = 0$.

(23) $(\mathrm{and}_{3b})(\langle 0,0,0\rangle) = 0$ and $(\mathrm{and}_{3b})(\langle 0,0,1\rangle) = 1$ and $(\mathrm{and}_{3b})(\langle 0,1,0\rangle) = 0$ and $(\mathrm{and}_{3b})(\langle 0,1,1\rangle) = 0$ and $(\mathrm{and}_{3b})(\langle 1,0,0\rangle) = 0$ and $(\mathrm{and}_{3b})(\langle 1,0,1\rangle) = 0$ and $(\mathrm{and}_{3b})(\langle 1,1,0\rangle) = 0$ and $(\mathrm{and}_{3b})(\langle 1,1,1\rangle) = 0$.

(24) $(\mathrm{and}_{3c})(\langle 0,0,0\rangle) = 1$ and $(\mathrm{and}_{3c})(\langle 0,0,1\rangle) = 0$ and $(\mathrm{and}_{3c})(\langle 0,1,0\rangle) = 0$ and $(\mathrm{and}_{3c})(\langle 0,1,1\rangle) = 0$ and $(\mathrm{and}_{3c})(\langle 1,0,0\rangle) = 0$ and $(\mathrm{and}_{3c})(\langle 1,0,1\rangle) = 0$ and $(\mathrm{and}_{3c})(\langle 1,1,0\rangle) = 0$ and $(\mathrm{and}_{3c})(\langle 1,1,1\rangle) = 0$.

(25) $\mathrm{or}_3(\langle 0,0,0\rangle) = 0$ and $\mathrm{or}_3(\langle 0,0,1\rangle) = 1$ and $\mathrm{or}_3(\langle 0,1,0\rangle) = 1$ and $\mathrm{or}_3(\langle 0,1,1\rangle) = 1$ and $\mathrm{or}_3(\langle 1,0,0\rangle) = 1$ and $\mathrm{or}_3(\langle 1,0,1\rangle) = 1$ and $\mathrm{or}_3(\langle 1,1,0\rangle) = 1$ and $\mathrm{or}_3(\langle 1,1,1\rangle) = 1$.

(26) $(\mathrm{or}_{3a})(\langle 0,0,0\rangle) = 1$ and $(\mathrm{or}_{3a})(\langle 0,0,1\rangle) = 1$ and $(\mathrm{or}_{3a})(\langle 0,1,0\rangle) = 1$ and $(\mathrm{or}_{3a})(\langle 0,1,1\rangle) = 1$ and $(\mathrm{or}_{3a})(\langle 1,0,0\rangle) = 0$ and $(\mathrm{or}_{3a})(\langle 1,0,1\rangle) = 1$ and $(\mathrm{or}_{3a})(\langle 1,1,0\rangle) = 1$ and $(\mathrm{or}_{3a})(\langle 1,1,1\rangle) = 1$.

(27) $(\mathrm{or}_{3b})(\langle 0,0,0\rangle) = 1$ and $(\mathrm{or}_{3b})(\langle 0,0,1\rangle) = 1$ and $(\mathrm{or}_{3b})(\langle 0,1,0\rangle) = 1$ and $(\mathrm{or}_{3b})(\langle 0,1,1\rangle) = 1$ and $(\mathrm{or}_{3b})(\langle 1,0,0\rangle) = 1$ and $(\mathrm{or}_{3b})(\langle 1,0,1\rangle) = 1$ and $(\mathrm{or}_{3b})(\langle 1,1,0\rangle) = 0$ and $(\mathrm{or}_{3b})(\langle 1,1,1\rangle) = 1$.

(28) $(\mathrm{or}_{3c})(\langle 0,0,0\rangle) = 1$ and $(\mathrm{or}_{3c})(\langle 0,0,1\rangle) = 1$ and $(\mathrm{or}_{3c})(\langle 0,1,0\rangle) = 1$ and $(\mathrm{or}_{3c})(\langle 0,1,1\rangle) = 1$ and $(\mathrm{or}_{3c})(\langle 1,0,0\rangle) = 1$ and $(\mathrm{or}_{3c})(\langle 1,0,1\rangle) = 1$ and $(\mathrm{or}_{3c})(\langle 1,1,0\rangle) = 1$ and $(\mathrm{or}_{3c})(\langle 1,1,1\rangle) = 0$.

(29) $\mathrm{xor}_3(\langle 0,0,0\rangle) = 0$ and $\mathrm{xor}_3(\langle 0,0,1\rangle) = 1$ and $\mathrm{xor}_3(\langle 0,1,0\rangle) = 1$ and $\mathrm{xor}_3(\langle 0,1,1\rangle) = 0$ and $\mathrm{xor}_3(\langle 1,0,0\rangle) = 1$ and $\mathrm{xor}_3(\langle 1,0,1\rangle) = 0$ and $\mathrm{xor}_3(\langle 1,1,0\rangle) = 0$ and $\mathrm{xor}_3(\langle 1,1,1\rangle) = 1$.

[2] The proposition (18) has been removed.

2. 2'S COMPLEMENT CIRCUIT PROPERTIES

Let x, b be sets. The functor $\mathrm{CompStr}(x,b)$ 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. 33) $\mathrm{CompStr}(x,b) = \mathrm{1GateCircStr}(\langle x,b\rangle, \mathrm{xor}_{2a})$.

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

(Def. 34) $\mathrm{CompCirc}(x,b) = \mathrm{1GateCircuit}(x,b,\mathrm{xor}_{2a})$.

Let x, b be sets. The functor $\mathrm{CompOutput}(x,b)$ yielding an element of $\mathrm{InnerVertices}(\mathrm{CompStr}(x,b))$ is defined by:

(Def. 35) $\mathrm{CompOutput}(x,b) = \langle\!\langle \langle x,b\rangle, \mathrm{xor}_{2a}\rangle\!\rangle$.

Let x, b be sets. The functor $\mathrm{IncrementStr}(x,b)$ 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. 36) $\mathrm{IncrementStr}(x,b) = \mathrm{1GateCircStr}(\langle x,b\rangle, \mathrm{and}_{2a})$.

Let x, b be sets. The functor $\mathrm{IncrementCirc}(x,b)$ yields a strict Boolean circuit of $\mathrm{IncrementStr}(x,b)$ with denotation held in gates and is defined as follows:

(Def. 37) $\mathrm{IncrementCirc}(x,b) = \mathrm{1GateCircuit}(x,b,\mathrm{and}_{2a})$.

Let x, b be sets. The functor $\mathrm{IncrementOutput}(x,b)$ yielding an element of $\mathrm{InnerVertices}(\mathrm{IncrementStr}(x,b))$ is defined by:

(Def. 38) $\mathrm{IncrementOutput}(x,b) = \langle\!\langle \langle x,b\rangle, \mathrm{and}_{2a}\rangle\!\rangle$.

Let x, b be sets. The functor $\mathrm{BitCompStr}(x,b)$ 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. 39) $\mathrm{BitCompStr}(x,b) = \mathrm{CompStr}(x,b) {+\cdot}\, \mathrm{IncrementStr}(x,b)$.

Let x, b be sets. The functor $\mathrm{BitCompCirc}(x,b)$ yields a strict Boolean circuit of $\mathrm{BitCompStr}(x,b)$ with denotation held in gates and is defined as follows:

(Def. 40) $\mathrm{BitCompCirc}(x,b) = \mathrm{CompCirc}(x,b) {+\cdot}\, \mathrm{IncrementCirc}(x,b)$.

One can prove the following propositions:

(30) For all non pair sets x, b holds $\mathrm{InnerVertices}(\mathrm{CompStr}(x,b))$ is a binary relation.

(31) For all non pair sets x, b holds $x \in$ the carrier of $\mathrm{CompStr}(x,b)$ and $b \in$ the carrier of $\mathrm{CompStr}(x,b)$ and $\langle\!\langle \langle x,b\rangle, \mathrm{xor}_{2a}\rangle\!\rangle \in$ the carrier of $\mathrm{CompStr}(x,b)$.

(32) For all non pair sets x, b holds the carrier of $\mathrm{CompStr}(x,b) = \{x,b\} \cup \{\langle\!\langle \langle x,b\rangle, \mathrm{xor}_{2a}\rangle\!\rangle\}$.

(33) For all non pair sets x, b holds $\mathrm{InnerVertices}(\mathrm{CompStr}(x,b)) = \{\langle\!\langle \langle x,b\rangle, \mathrm{xor}_{2a}\rangle\!\rangle\}$.

(34) For all non pair sets x, b holds $\langle\!\langle \langle x,b\rangle, \mathrm{xor}_{2a}\rangle\!\rangle \in \mathrm{InnerVertices}(\mathrm{CompStr}(x,b))$.

(35) For all non pair sets x, b holds $\mathrm{InputVertices}(\mathrm{CompStr}(x,b)) = \{x,b\}$.

(36) For all non pair sets x, b holds $x \in \mathrm{InputVertices}(\mathrm{CompStr}(x,b))$ and $b \in \mathrm{InputVertices}(\mathrm{CompStr}(x,b))$.

(37) For all non pair sets x, b holds $\mathrm{InputVertices}(\mathrm{CompStr}(x,b))$ has no pairs.

(38) For all non pair sets x, b holds $\mathrm{InnerVertices}(\mathrm{IncrementStr}(x,b))$ is a binary relation.

(39) For all non pair sets x, b holds $x \in$ the carrier of IncrementStr(x,b) and $b \in$ the carrier of IncrementStr(x,b) and $\langle\langle x,b\rangle, \mathrm{and}_{2a}\rangle \in$ the carrier of IncrementStr(x,b).

(40) For all non pair sets x, b holds the carrier of IncrementStr$(x,b) = \{x,b\} \cup \{\langle\langle x,b\rangle, \mathrm{and}_{2a}\rangle\}$.

(41) For all non pair sets x, b holds InnerVertices(IncrementStr$(x,b)) = \{\langle\langle x,b\rangle, \mathrm{and}_{2a}\rangle\}$.

(42) For all non pair sets x, b holds $\langle\langle x,b\rangle, \mathrm{and}_{2a}\rangle \in$ InnerVertices(IncrementStr(x,b)).

(43) For all non pair sets x, b holds InputVertices(IncrementStr$(x,b)) = \{x,b\}$.

(44) For all non pair sets x, b holds $x \in$ InputVertices(IncrementStr(x,b)) and $b \in$ InputVertices(IncrementStr(x,b)).

(45) For all non pair sets x, b holds InputVertices(IncrementStr(x,b)) has no pairs.

(46) For all non pair sets x, b holds InnerVertices(BitCompStr(x,b)) is a binary relation.

(47) Let x, b be non pair sets. Then

(i) $x \in$ the carrier of BitCompStr(x,b),

(ii) $b \in$ the carrier of BitCompStr(x,b),

(iii) $\langle\langle x,b\rangle, \mathrm{xor}_{2a}\rangle \in$ the carrier of BitCompStr(x,b), and

(iv) $\langle\langle x,b\rangle, \mathrm{and}_{2a}\rangle \in$ the carrier of BitCompStr(x,b).

(48) For all non pair sets x, b holds the carrier of BitCompStr$(x,b) = \{x,b\} \cup \{\langle\langle x,b\rangle, \mathrm{xor}_{2a}\rangle, \langle\langle x,b\rangle, \mathrm{and}_{2a}\rangle\}$.

(49) For all non pair sets x, b holds InnerVertices(BitCompStr$(x,b)) = \{\langle\langle x,b\rangle, \mathrm{xor}_{2a}\rangle, \langle\langle x,b\rangle, \mathrm{and}_{2a}\rangle\}$.

(50) For all non pair sets x, b holds $\langle\langle x,b\rangle, \mathrm{xor}_{2a}\rangle \in$ InnerVertices(BitCompStr(x,b)) and $\langle\langle x, b\rangle, \mathrm{and}_{2a}\rangle \in$ InnerVertices(BitCompStr(x,b)).

(51) For all non pair sets x, b holds InputVertices(BitCompStr$(x,b)) = \{x,b\}$.

(52) For all non pair sets x, b holds $x \in$ InputVertices(BitCompStr(x,b)) and $b \in$ InputVertices(BitCompStr(x,b)).

(53) For all non pair sets x, b holds InputVertices(BitCompStr(x,b)) has no pairs.

(54) For all non pair sets x, b and for every state s of CompCirc(x,b) holds (Following(s))(CompOutput$(x,b)) = (\mathrm{xor}_{2a})(\langle s(x), s(b)\rangle)$ and (Following(s))$(x) = s(x)$ and (Following(s))$(b) = s(b)$.

(55) Let x, b be non pair sets, s be a state of CompCirc(x,b), and a_1, a_2 be elements of *Boolean*. If $a_1 = s(x)$ and $a_2 = s(b)$, then (Following(s))(CompOutput$(x,b)) = \neg a_1 \oplus a_2$ and (Following(s))$(x) = a_1$ and (Following(s))$(b) = a_2$.

(56) For all non pair sets x, b and for every state s of BitCompCirc(x,b) holds (Following(s))(CompOutput$(x,b)) = (\mathrm{xor}_{2a})(\langle s(x), s(b)\rangle)$ and (Following(s))$(x) = s(x)$ and (Following(s))$(b) = s(b)$.

(57) Let x, b be non pair sets, s be a state of BitCompCirc(x,b), and a_1, a_2 be elements of *Boolean*. If $a_1 = s(x)$ and $a_2 = s(b)$, then (Following(s))(CompOutput$(x,b)) = \neg a_1 \oplus a_2$ and (Following(s))$(x) = a_1$ and (Following(s))$(b) = a_2$.

(58) For all non pair sets x, b and for every state s of IncrementCirc(x,b) holds (Following(s))(IncrementOutput$(x,b)) = (\mathrm{and}_{2a})(\langle s(x), s(b)\rangle)$ and (Following(s))$(x) = s(x)$ and (Following(s))$(b) = s(b)$.

(59) Let x, b be non pair sets, s be a state of IncrementCirc(x,b), and a_1, a_2 be elements of *Boolean*. If $a_1 = s(x)$ and $a_2 = s(b)$, then $(\text{Following}(s))(\text{IncrementOutput}(x,b)) = \neg a_1 \wedge a_2$ and $(\text{Following}(s))(x) = a_1$ and $(\text{Following}(s))(b) = a_2$.

(60) For all non pair sets x, b and for every state s of BitCompCirc(x,b) holds $(\text{Following}(s))(\text{IncrementOutput}(x,b)) = (\text{and}_{2a})(\langle s(x), s(b)\rangle)$ and $(\text{Following}(s))(x) = s(x)$ and $(\text{Following}(s))(b) = s(b)$.

(61) Let x, b be non pair sets, s be a state of BitCompCirc(x,b), and a_1, a_2 be elements of *Boolean*. If $a_1 = s(x)$ and $a_2 = s(b)$, then $(\text{Following}(s))(\text{IncrementOutput}(x,b)) = \neg a_1 \wedge a_2$ and $(\text{Following}(s))(x) = a_1$ and $(\text{Following}(s))(b) = a_2$.

(62) Let x, b be non pair sets and s be a state of BitCompCirc(x,b). Then $(\text{Following}(s))(\text{CompOutput}(x,b)) = (\text{xor}_{2a})(\langle s(x), s(b)\rangle)$ and $(\text{Following}(s))(\text{IncrementOutput}(x,b)) = (\text{and}_{2a})(\langle s(x), s(b)\rangle)$ and $(\text{Following}(s))(x) = s(x)$ and $(\text{Following}(s))(b) = s(b)$.

(63) Let x, b be non pair sets, s be a state of BitCompCirc(x,b), and a_1, a_2 be elements of *Boolean*. Suppose $a_1 = s(x)$ and $a_2 = s(b)$. Then $(\text{Following}(s))(\text{CompOutput}(x,b)) = \neg a_1 \oplus a_2$ and $(\text{Following}(s))(\text{IncrementOutput}(x,b)) = \neg a_1 \wedge a_2$ and $(\text{Following}(s))(x) = a_1$ and $(\text{Following}(s))(b) = a_2$.

(64) For all non pair sets x, b and for every state s of BitCompCirc(x,b) holds Following(s) is stable.

REFERENCES

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

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

[3] Czesław Byliński. Functions from a set to a set. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/funct_2.html`.

[4] Czesław Byliński. Finite sequences and tuples of elements of a non-empty sets. *Journal of Formalized Mathematics*, 2, 1990. `http://mizar.org/JFM/Vol2/finseq_2.html`.

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

[6] 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`.

[7] 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`.

[8] Takaya Nishiyama and Yasuho Mizuhara. Binary arithmetics. *Journal of Formalized Mathematics*, 5, 1993. `http://mizar.org/JFM/Vol5/binarith.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] Edmund Woronowicz. Relations and their basic properties. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/relat_1.html`.

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

Received October 25, 1996

Published January 2, 2004