

**Title:** Reducing n-valued functions with inverters at input

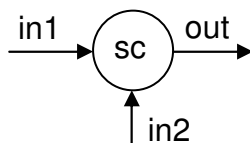
**Author:** Peter Lablans, Ternarylogic LLC.

**Date:** May 11, 2007

### Functions and truth tables

N-valued logic offers significant opportunities for reduction of complexities. First some ground rules will be established. An n-valued logic function will be considered a 2-place or 2-inputs/single output function. An n-valued function will be determined by an n-valued truth table. A function has several properties such as: commutativity, associativity, reversibility, etc. A first concern is if a function is commutative.

A device or a method implementing an n-valued function ‘sc’ is modeled by the following diagram:



The diagram has a first input providing an n-valued symbol ‘in1’, a second input providing an n-valued symbol ‘in2’, and an output providing an n-valued symbol ‘out’. The relationship between the inputs and output is provided by:  
**in1 sc in2 → out.** An arrow is used instead of ‘=’ as an assignment operator.

A function ‘sc’ may be used as an operator such as the ‘+’, keeping in mind that ‘sc’ is defined by an  $n \times n$  truth table. To keep things relatively simple n is herein assumed to be 3 and all functions and symbols are ternary. However, the approach applies to any  $n > 2$ .

Assume that ‘sc’ is the modulo-3 adder of which the truth table is provided in the following table.

+ mod3		in2		
	out	0	1	2
in1	0	0	1	2
	1	1	2	0
	2	2	0	1

This function is commutative: it does not matter if one executes (in1, in2) or (in2, in1). Or in an example:  $1 \text{ sc } 2 = 2 \text{ sc } 1$ , as both have 0 as result.

Many n-valued functions are non-commutative and one should take care in the order of execution. To make sure that one can maintain an order of execution each input of the diagram is assigned to either the columns or the rows of a truth table. In this case the input represented by the horizontal arrow of the diagram is assigned to the rows (in1) of the truth table and the input represented by the vertical arrow is assigned to the columns (in2) of the truth table.

Further more in an expression:  $\text{in1 sc in2} \rightarrow \text{out}$ , the input 'in1' to the left of 'sc' represents the rows of the truth table and the input 'in2' to the right of 'sc' represents the columns of the truth tables.

### Non commutative functions

In a non-commutative n-valued function 'sc' in general:  
 $\text{in1 sc in2} \neq \text{in2 sc in1}$ .

As an example one may use the ternary function 'sc1' as provided in the following truth table.

sc1		in2		
	out	0	1	2
in1	0	0	2	1
	1	1	0	2
	2	2	1	0

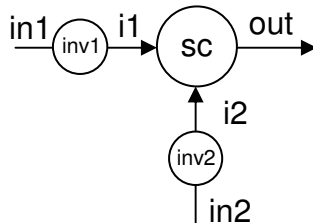
It is easy to see that in most cases:  $(\text{in1 sc1 in2} \neq \text{in2 sc in1})$ . For instance:  $0 \text{ sc1 } 1 \rightarrow 2$  and  $1 \text{ sc1 } 0 \rightarrow 1$ . In fact 'sc1' is the ternary function  $(\text{in1} - \text{in2}) \bmod 3$ . This is different from the function  $\text{sc2} = (\text{in2} - \text{in1}) \bmod 3$  of which the following table provided the truth table.

sc2		in2		
	out	0	1	2
in1	0	0	1	2
	1	2	0	1
	2	1	2	0

This demonstrates first of all that the 'inverse' of modulo-3 addition (unlike the binary or 2-valued case) is not again modulo-3 addition. Secondly modulo-3 addition has to inverse functions.

## Functions with inverters

Switching functions often have inverters (such as modulo-n multipliers) at its inputs. A configuration wherein each input to an n-valued function has an inverter is shown in the following diagram.



Assume that 'sc' is the modulo-3 addition,  $inv1 = [0\ 2\ 1]$  and  $inv2 = [0\ 2\ 1]$ . Inverter 'inv1' represents multiplication modulo-3 of 'in1' by a factor 2. The same applies to inverter 'inv2'. The transformation of 'in1', being the input to inverter 'inv1', to 'i1', being the output of inverter 'inv1' can be represented by  $[0\ 1\ 2] \rightarrow [0\ 2\ 1]$ .

The relationships are:  $i1 \rightarrow inv1(in1)$ ;  $i2 \rightarrow inv2(in2)$ ; and  $i1\ sc\ i2 \rightarrow out$ .

Unfortunately this means that in order to determine 'out' one first has to calculate 'i1' and 'i2' and then apply 'i1' and 'i2' to the function 'sc'. It would be advantageous if one could directly execute a function 'sr' so that:  $in1\ sr\ in2 \rightarrow out$ , thereby reducing the number of steps.

## The reduced function

The function 'sr' is called the reduced function of 'sc' over inverters 'inv1' and 'inv2'. The reduction scheme works as follows:

1. When an input represents the rows of a truth table, then an inverter in that input transforms the order of the rows, but leaves the columns intact;
2. When an input represents the columns of a truth table, then an inverter in that input transforms the order of the columns, but leaves the rows intact;
3. An inverter  $[x\ y\ z]$  has elements  $\{x,y,z\}$  each with a position and a state. Element x has position 0 and state x, element y has position 1 and state y, and element z has position 2 and state z.
4. An inverter  $[x\ y\ z]$  in an input representing rows will perform the following transformation of rows:
  - the row of the new truth table in position 0 will be the row in position x in the old truth table;
  - the row of the new truth table in position 1 will be the row in position y in the old truth table; and
  - the row of the new truth table in position 2 will be the row in position z in the old truth table.

5. An inverter [x y z] in an input representing columns will perform the following transformation of rows:

the column of the new truth table in position 0 will be the column in position x in the old truth table;

the column of the new truth table in position 1 will be the column in position y in the old truth table; and

the column of the new truth table in position 2 will be the column in position z in the old truth table.

The order of the transformation does not matter.

### Examples

As an example the reduced truth table of the modulo-3 adder with at each input an inverter [0 2 1] will be provided in the following truth table.

+r mod3		in2		
	out	0	1	2
in1	0	0	2	1
	1	2	1	0
	2	1	0	2

Another example is a reduced function sc1r (the modulo-3 subtraction) also with an inverter [0 2 1] in each input. The reduced truth table of 'sc1r' is provided in the following table.

sc1r		in2		
	out	0	1	2
in1	0	0	1	2
	1	2	0	1
	2	1	2	0

**About Ternarylogic LLC:** Ternarylogic LLC is a New Jersey based R&D company. Its mission is to create novel MVL technology. The company owns a portfolio of inventions related to scramblers/descramblers, sequence generators and sequence detectors, sequence correlators, gates and inverters based circuitry, non-binary multipliers, latches and other non-volatile memory elements, optical disk applications and MC-DSSS technology.

### Intellectual Property Rights

Methods and apparatus disclosed herein are part of Ternarylogic's IP rights. Please review our portfolio.

© 2007, Ternarylogic LLC. All rights reserved.