Title: **Creating n-valued functions from switches and inverters**

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

Date: **May 10, 2007**

Realizing n-valued functions

Binary logic has 16 2-place or 2-input/single output functions. From those effectively only the important commutative functions are used (AND, NAND, OR, NOR, XOR and EQUAL). It is possible to create all binary switching functions from a limited set of functions, which is called the adequate set of connectives. For instance the NAND with an inverter form such a set. It is possible to create such sets also for n-valued logic. However forming n-valued functions may become quite complicated. This white paper describes how one can create an n-valued logic or switching function from n-valued switches and inverters. Both the n-valued inverter and the n-valued switch are explained in separate white papers.

A binary function

A simple example can be provided by realizing a well known binary function, for instance the XOR function, with switches and inverters.

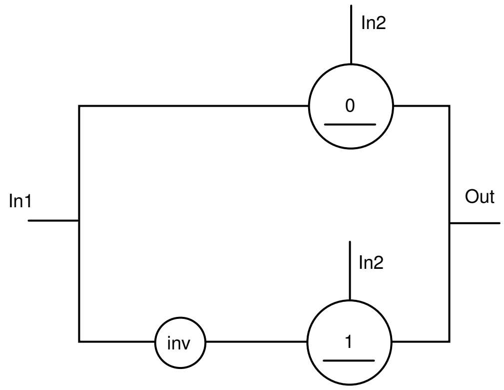
The first step is to analyze the XOR function's truth table, which is shown below.

XOR		In2	
	Out	0	1
In1	0	0	1
	1	1	0

One way to look at the truth table is to see it as built from two columns: one column under 0 for In2 (the blue column) and one column under 1 for In2 (the red column).

One may actually say that for In2 = 0, In1 'sees' the blue column; and for In2 = 1 In1 'sees' the red column. In other words: for In2 =0, In1 'sees' an identity inverter; and for In2 = 1, In1 'sees' a binary reversible inverter. Thus the problem of realizing a truth table is reduced to activating or enabling a path for In1 through an inverter based on a state of In2.

The following figure shows such a realization.



The upper path is activated when In2 =0, because then the switch with 0 in the circle is then conducting and the switch with the 1 in the circle is non-conducting. Accordingly In1 will go through the upper branch which has no inverter and is an identity. It realizes the first column of the truth table.

When In2 = 1 the switch in the lower branch is conducting and the switch in the upper branch is non-conducting. In1 has to go through inverter 'inv' and In1 is inverted when reaching Out. The circuit as shown thus realizes the XOR function.

An important element herein is that absence of signal represents state 0. One can also avoid using the absence of signal as a state. However in that case a non-conducting state should provide absence of signal at its output.

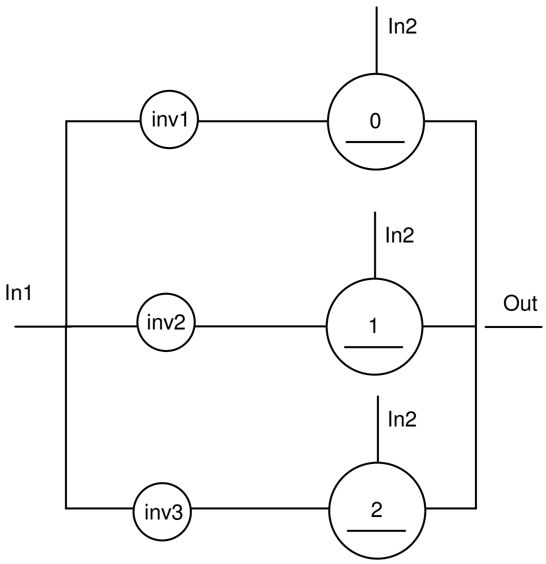
Realizing a ternary function

The same principle applies to realizing an n-valued function such as for instance the function defined by the following truth table.

		In2		
	Out	0	1	2
In1	0	0	2	1
	1	2	1	0
	2	1	0	2

The ternary function as shown may be considered to be built from 3 columns, each column representing a ternary reversible inverter. The blue column which is activated when In2 = 0 is an inverter inv1 = [0 2 1]. The red column which is activated when In2 =1 is an inverter inv2 = [2 1 0]. The yellow column which is activated when In2 = 2 is an inverter inv3 = [1 0 2].

A circuit that realizes the truth table is shown in the following figure.

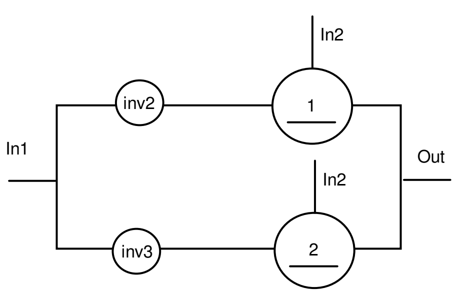


Depending on the state of In2 the top branch, the middle branch or the bottom branch is enabled by a switch becoming conducting.

In this case the truth table requires reversible inverters. This is not always the case. For instance the ternary function may be the carry function for a modulo-3 adder. Its truth table is provided next.

		In2		
	Out	0	1	2
In1	0	0	0	0
	1	0	0	1
	2	0	1	1

The inverter inv1 = [0 0 0] which is an open connection. Inverter inv2 = [0 0 1] and inverter inv3 = [0 1 1]. The realization of that function is shown in the following figure.



Because absence of signal is the state 0 and there is no conducting path when In2 = 0 the inverter inv1 = [0 0 0] is automatically realized.

Many variations are possible.

N-valued logic functions

Also 4-valued or any n-valued 2-input/single output logic function can be realized by using n-valued switches and inverters.

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.