Multi-Valued Logic Gates using Verilog HDL

Zhanibek Kalkayev

June 25, 2020

1 Introduction

Right after the getting acknowledged with Multi-Valued Logic theory and brief background, the next step was to design basic and foundation gates such as NOT, AND, NAND, OR, NOR, XOR and XNOR gates. As a base for logic the quaternary logic with four logic levels (0, 1, 2, 3) was chosen. Observing theoretical background of quaternary algebra, the truth table of basic gates were derived using default functional quaternary operators such as MIN, MAX and Sum Of:

1.1 NOT Gate

As well as in binary logic, NOT Gate has one input and produces one output in quaternary algebra. Truth table is given below:

Table 1: NOT Gate Truth Table

Operand	A	0	1	2	3
Output	Y	3	2	1	0

1.2 AND and NAND Gates

AND Gate is described as a basic digital logic gate which implements logical conjuction. In binary logic, the high signal appears only in case of all inputs are high. As in quaternary algebra there are 4 levels of truth, the output is determined by comparing both binary values of inputs. For example, for inputs 1 and 2 we obtain binary values of 01 and 10, consequently using classic binary AND gate for 0 and 1, 1 and 0, we get the final value of 00, thus the output is 0. NAND Gate is the opposite of AND gate. Both AND and NAND Gates have two inputs and produce one output in quaternary algebra. Truth tables are given below:

Table 2: AND Gate Truth Table

Operand	A	0	0	0	0	1	1	1	2	2	3
Operand	B	0	1	2	3	1	2	3	2	3	3
Output	Y	0	0	0	0	1	0	1	2	2	3

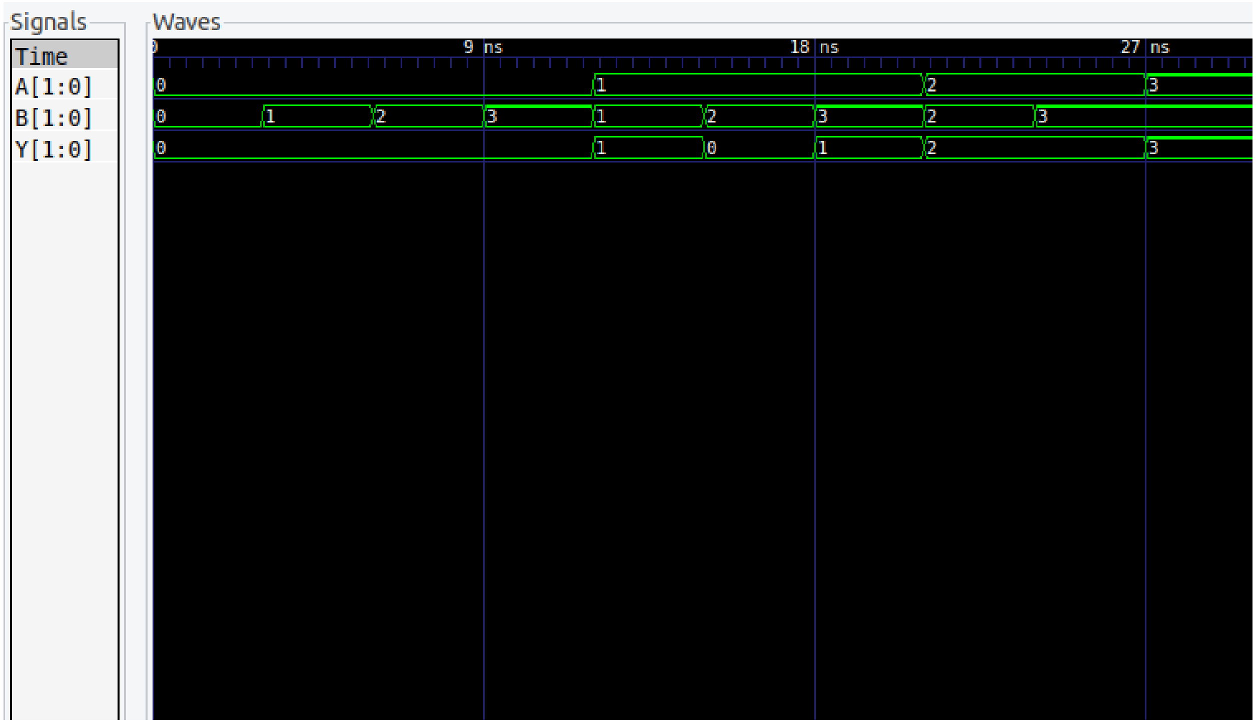
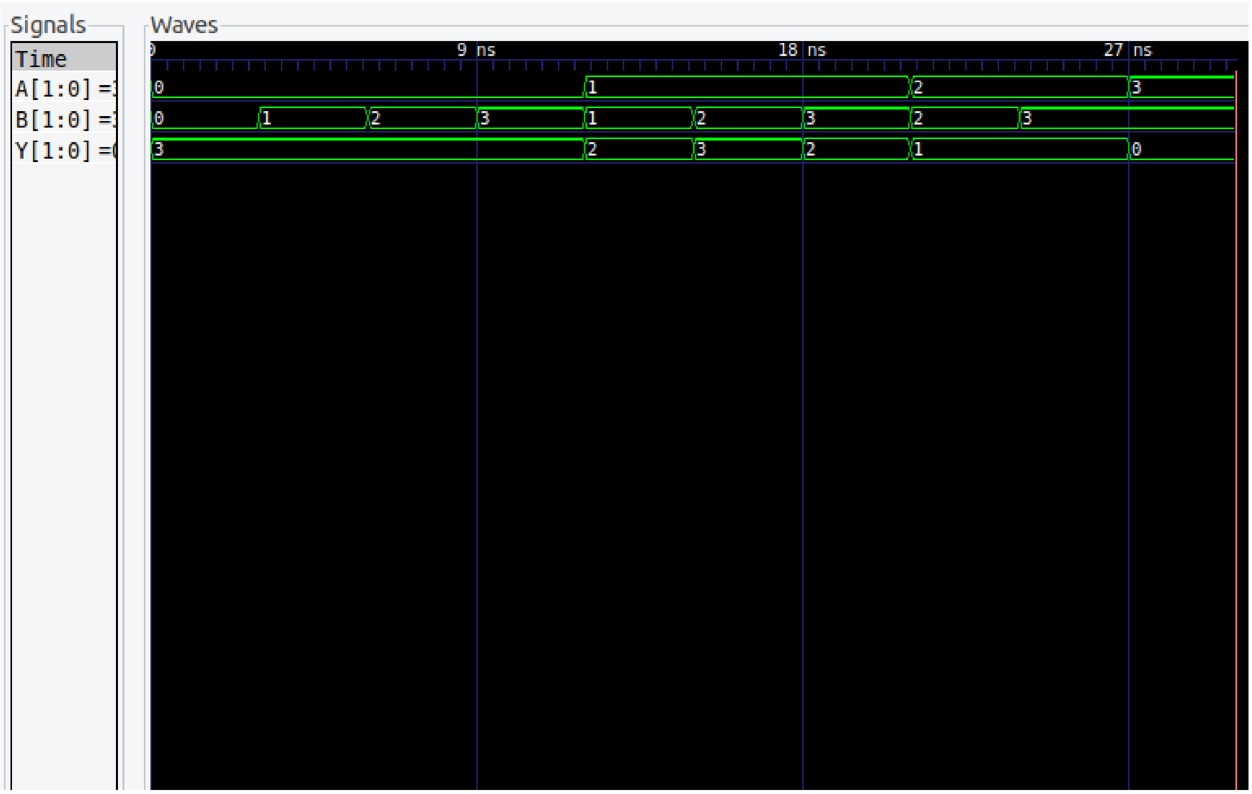


Table 3: NAND Gate Truth Table

Operand	A	0	0	0	0	1	1	1	2	2	3
Operand	B	0	1	2	3	1	2	3	2	3	3
Output	Y	3	3	3	3	2	3	2	1	1	0



1.3 OR and NOR Gates

OR Gate is described as a basic digital logic gate which implements logical disjunction. In binary logic, the high signal appears in any case while one of the inputs is high. As it was mentioned before, the output in quaternary algebra is determined by comparing both binary values of inputs. For inputs 1 and 2 we obtain binary values of 01 and 10, consequently using classic binary OR gate for 0 and 1, 1 and 0, we get the final value of 11, thus the output is 3. NOR Gate is the opposite of OR gate. Both OR and NOR Gates have two inputs and produce one output in quaternary algebra. Truth tables are given below:

Table 4: OR Gate Truth Table

Operand	A	0	0	0	0	1	1	1	2	2	3
Operand	B	0	1	2	3	1	2	3	2	3	3
Output	Y	0	1	2	3	1	3	3	2	3	3

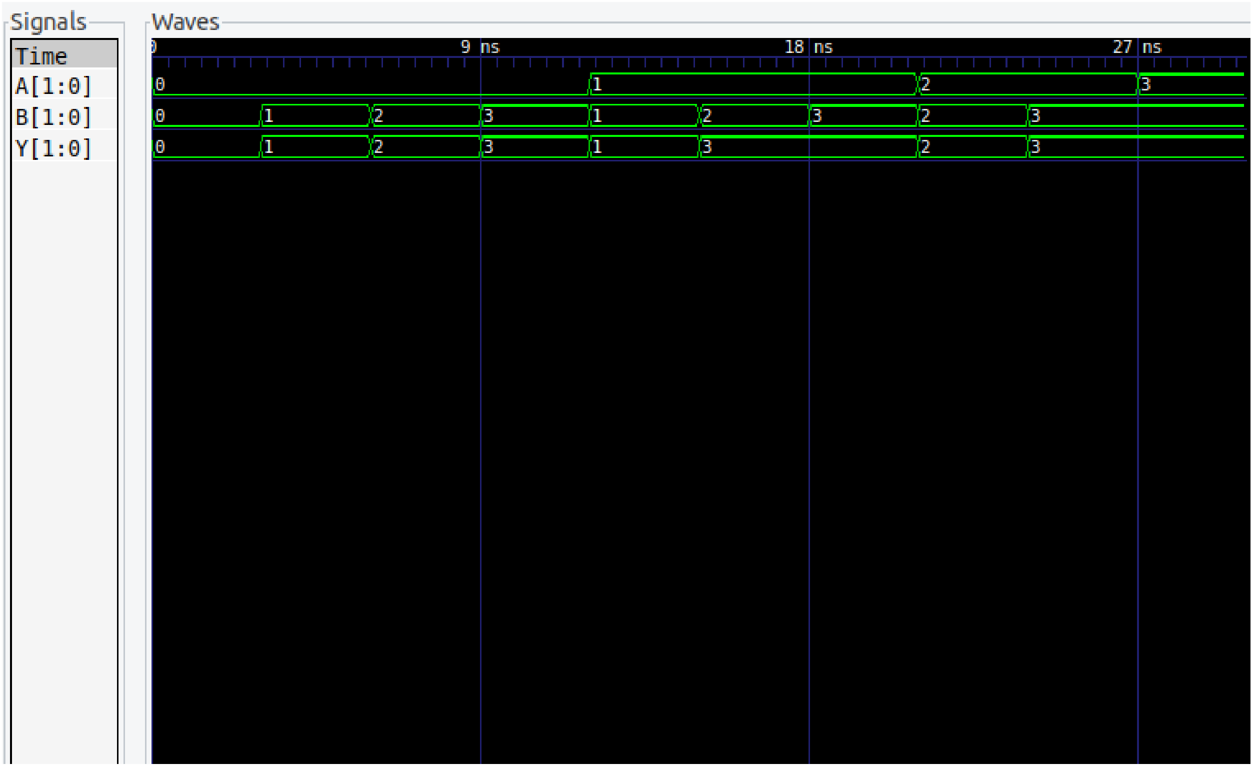
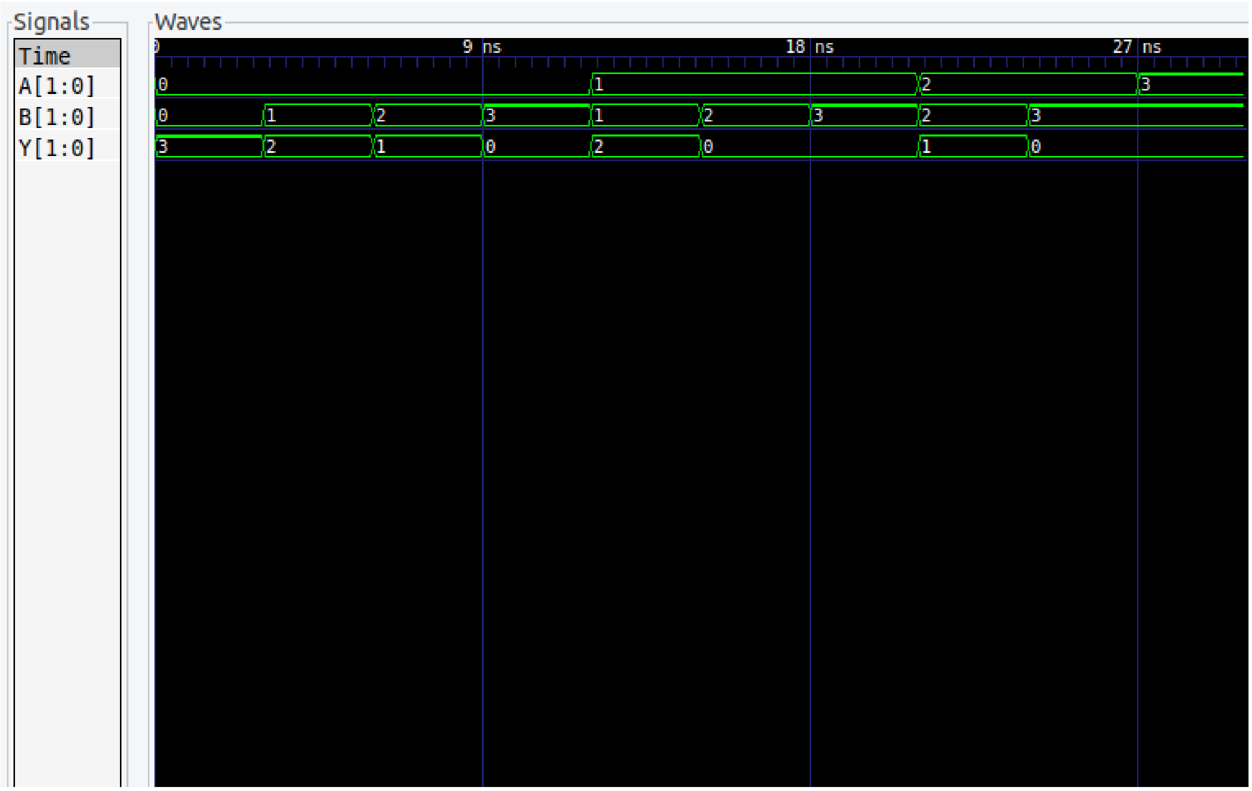


Table 5: NOR Gate Truth Table

Operand	A	0	0	0	0	1	1	1	2	2	3
Operand	B	0	1	2	3	1	2	3	2	3	3
Output	Y	3	2	1	0	2	0	0	1	0	0



1.4 XOR and XNOR Gates

XOR (EOR, EXOR, Exclusive OR) is characterized as a digital logic gate which gives a high signal output when the number of true inputs is odd. To find the output of XOR Gate in quaternary logic, function Sum Of is used. For inputs 1 and 2 (01 and 10), we add first digits and second digits to each other: 0 + 1, 1 + 0, thus obtaining 11, which is decimal 3. XNOR Gate is the opposite of XOR. Both XOR and XNOR Gates have two inputs and produce one output in quaternary algebra. Truth tables are given below:

Table 6: XOR Gate Truth Table

Operand	A	0	0	0	0	1	1	1	2	2	3
Operand	B	0	1	2	3	1	2	3	2	3	3
Output	Y	0	1	2	3	0	3	2	0	1	0

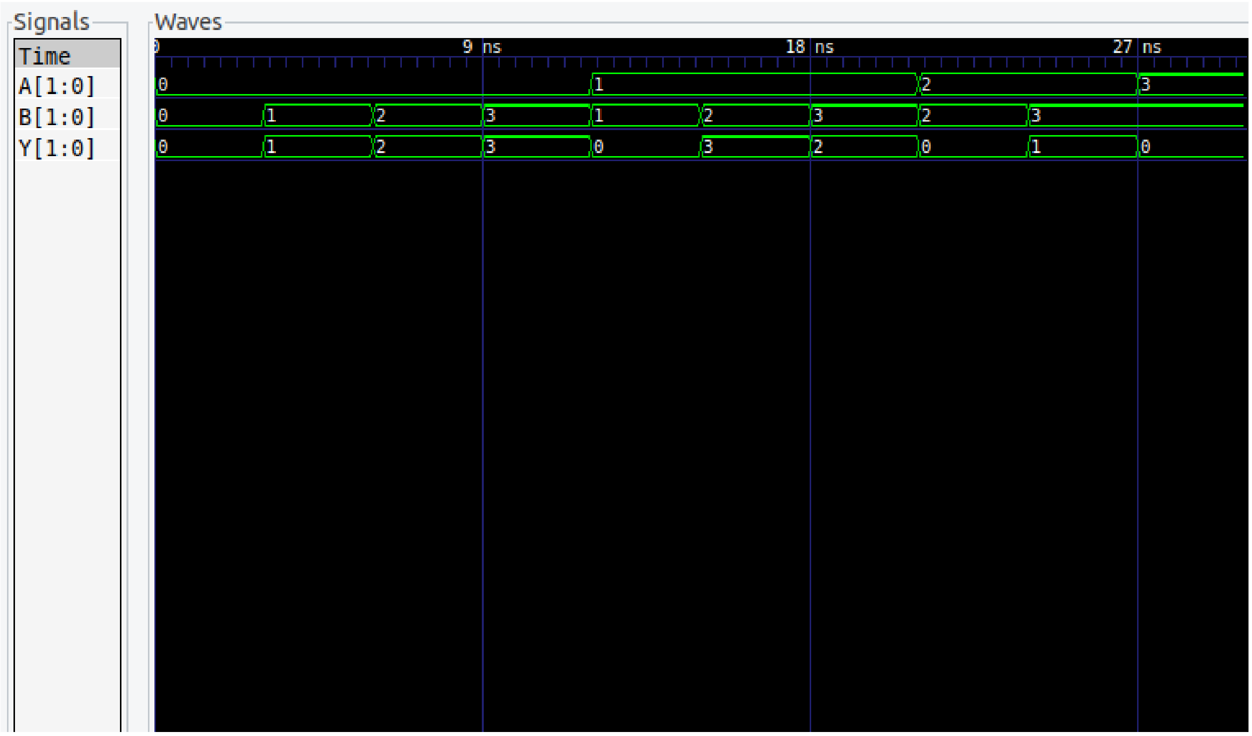
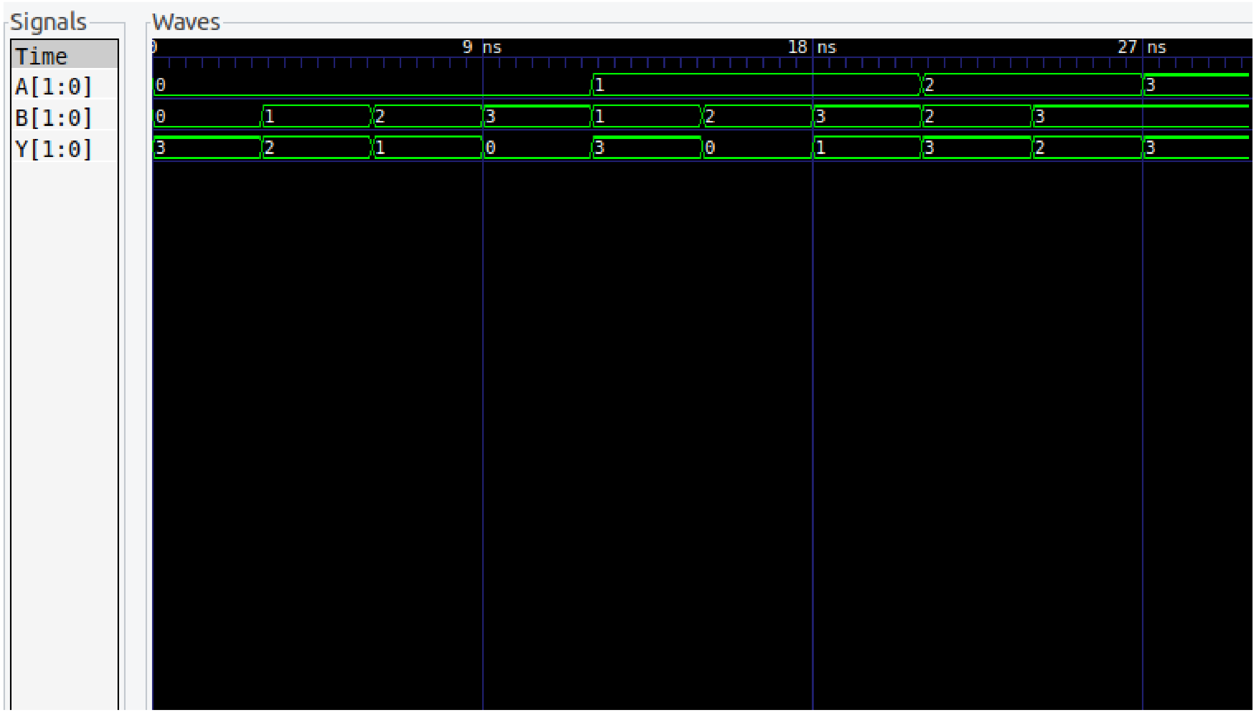


Table 7: XNOR Gate Truth Table

Operand	A	0	0	0	0	1	1	1	2	2	3
Operand	B	0	1	2	3	1	2	3	2	3	3
Output	Y	3	2	1	0	3	0	1	3	2	3


Signals
Waves
Time
A[1:0]
B[1:0]
Y[1:0]
9 ns
18 ns
27 ns