

Bivalent correction of IEEE Std 1800-2017 (Verilog) and Std 1164-1993 (VHDL)

© Copyright 2018 by Colin James III All rights reserved.

Abstract: We evaluate the multivalued logic SystemVerilog in IEEE Std 1800-2017. The classical logic proof tables for the connectives And, Or, Xor, and negations are based on the bivalency of 1, 0, X, Z as $0=\sim 1$ and $Z=\sim X$. This refutes and corrects the standard. We also retrofit and correct IEEE Std 1164-1993 (SynopsysVHDL) for the same.

We assume the method and apparatus of Meth8/VL4 with Tautology as the designated *proof* value, **F** as contradiction, N as truthity (non-contingency), and C as falsity (contingency). The 16-valued truth table is row-major and horizontal. The 16-valued truth table is row-major and horizontal, or repeating fragments of 128-tables, sometimes with table counts, for more variables. (See ersatz-systems.com.)

LET: \sim Not, \neg ; + Or, \vee, \cup ; - Not Or; & And, \wedge, \cap ; \ Not And;
 > Imply, greater than, \rightarrow, \supset ; < Not Imply, less than, \Leftarrow
 = Equivalent, \equiv, ε , @ Not Equivalent, \neq ;
 % possibility, for one or some, \exists, \diamond ; # necessity, for every or all, \forall, \square ;
 $\sim(y < x)$ ($x \leq y$), ($x \subseteq y$); (p=p) Tautology.

See: ieeexplore.ieee.org/document/1560791;

Table 28-3—Truth tables for multiple input logic gates

and	0	1	x	z	or	0	1	x	z	xor	0	1	x	z	(28.3.1)
0	0	0	0	0	0	0	1	x	x	0	0	1	x	x	
1	0	1	x	x	1	1	1	1	1	1	1	0	x	x	
x	0	x	x	x	x	x	1	x	x	x	x	x	x	x	
z	0	x	x	x	z	x	1	x	x	z	x	x	x	x	

and	0	1	x	z	or	0	1	x	z	xor	0	1	x	z	(28.3.2)
0	0	0	0	0	0	0	1	x	<u>z</u>	0	0	1	x	<u>z</u>	
1	0	1	x	<u>z</u>	1	1	1	1	1	1	1	0	<u>z</u>	x	
x	0	x	x	<u>0</u>	x	x	1	x	<u>1</u>	x	x	<u>z</u>	<u>0</u>	<u>1</u>	
z	0	<u>z</u>	<u>0</u>	<u>z</u>	z	<u>z</u>	1	<u>1</u>	<u>z</u>	z	<u>z</u>	x	<u>1</u>	<u>0</u>	

See: perso.telecom-paristech.fr/guilley/ENS/20161206/TP/tp_syn/doc/IEEE_VHDL_1164-1993.pdf

VHDL MODEL INTEROPERABILITY (Std_logic_1164_1993) (1164-1993.1)

```
-- truth table for "not" function
CONSTANT not_table: stdlogic_1d :=
-----
-- U    X    0    1    Z    W    L    H    -
-----
( 'U', 'X', '1', '0', 'X', 'X', '1', '0', 'X' );

-- truth table for "and" function
```

```

CONSTANT and_table : stdlogic_table := (
-----
-- U      X      0      1      Z      W      L      H      -
-----
( 'U', 'U', '0', 'U', 'U', 'U', '0', 'U', 'U' ), -- | U |
( 'U', 'X', '0', 'X', 'X', 'X', '0', 'X', 'X' ), -- | X |
( '0', '0', '0', '0', '0', '0', '0', '0', '0' ), -- | 0 |
( 'U', 'X', '0', '1', 'X', 'X', '0', '1', 'X' ), -- | 1 |
( 'U', 'X', '0', 'X', 'X', 'X', '0', 'X', 'X' ), -- | Z |
( 'U', 'X', '0', 'X', 'X', 'X', '0', 'X', 'X' ), -- | W |
( '0', '0', '0', '0', '0', '0', '0', '0', '0' ), -- | L |
( 'U', 'X', '0', '1', 'X', 'X', '0', '1', 'X' ), -- | H |
( 'U', 'X', '0', 'X', 'X', 'X', '0', 'X', 'X' ) -- | - | );

```

```

-- truth table for "or" function
CONSTANT or_table : stdlogic_table := (
-----
-- U      X      0      1      Z      W      L      H      -
-----
( 'U', 'U', 'U', '1', 'U', 'U', 'U', '1', 'U' ), -- | U |
( 'U', 'X', 'X', '1', 'X', 'X', 'X', '1', 'X' ), -- | X |
( 'U', 'X', '0', '1', 'X', 'X', '0', '1', 'X' ), -- | 0 |
( '1', '1', '1', '1', '1', '1', '1', '1', '1' ), -- | 1 |
( 'U', 'X', 'X', '1', 'X', 'X', 'X', '1', 'X' ), -- | Z |
( 'U', 'X', 'X', '1', 'X', 'X', 'X', '1', 'X' ), -- | W |
( 'U', 'X', '0', '1', 'X', 'X', '0', '1', 'X' ), -- | L |
( '1', '1', '1', '1', '1', '1', '1', '1', '1' ), -- | H |
( 'U', 'X', 'X', '1', 'X', 'X', 'X', '1', 'X' ) -- | - | );

```

```

-- truth table for "xor" function
CONSTANT xor_table : stdlogic_table := (
-----
-- U      X      0      1      Z      W      L      H      -
-----
( 'U', 'U', 'U', 'U', 'U', 'U', 'U', 'U', 'U' ), -- | U |
( 'U', 'X', 'X', 'X', 'X', 'X', 'X', 'X', 'X' ), -- | X |
( 'U', 'X', '0', '1', 'X', 'X', '0', '1', 'X' ), -- | 0 |
( 'U', 'X', '1', '0', 'X', 'X', '1', '0', 'X' ), -- | 1 |
( 'U', 'X', 'X', 'X', 'X', 'X', 'X', 'X', 'X' ), -- | Z |
( 'U', 'X', 'X', 'X', 'X', 'X', 'X', 'X', 'X' ), -- | W |
( 'U', 'X', '0', '1', 'X', 'X', '0', '1', 'X' ), -- | L |
( 'U', 'X', '1', '0', 'X', 'X', '1', '0', 'X' ), -- | H |
( 'U', 'X', 'X', 'X', 'X', 'X', 'X', 'X', 'X' ) -- | - | );

```

Remark 1164-1993.2: Areas amended are shaded in gray with modifications underlined.

```

-- truth table for "not" function
CONSTANT not_table: stdlogic_1d :=
-----

```

(1164-1993.2)

```

--      U      X      0      1      Z      W      L      H      -
-----
( 'U', 'X', '1', '0', 'X', 'X', '1', '0', 'X' ); = not[...]
( 'U', 'X', '1', '0', 'X', 'X', '0', '1', 'X' ); = Not( not[...])

```

```

-- truth table for "and" function
CONSTANT and_table : stdlogic_table := (

```

```

-----
-- U      X      0      1      Z      W      L      H      -
-----
( 'U', 'U', '0', 'U', 'U', 'U', '0', 'U', 'U' ), -- | U |
( 'U', 'X', '0', '0', 'X', 'X', '0', 'X', 'X' ), -- | X |
( '0', '0', '0', '0', '0', '0', '0', '0', '0' ), -- | 0 |
( 'U', 'X', '0', 'Z', 'X', 'X', '0', '1', 'X' ), -- | 1 |
( 'U', '0', '0', 'Z', 'Z', 'X', '0', 'X', 'X' ), -- | Z |
( 'U', 'X', '0', 'X', 'X', 'X', '0', 'X', 'X' ), -- | W |
( '0', '0', '0', '0', '0', '0', '0', '0', '0' ), -- | L |
( 'U', 'X', '0', '1', 'X', 'X', '0', '1', 'X' ), -- | H |
( 'U', 'X', '0', 'X', 'X', 'X', '0', 'X', 'X' ) -- | - |);

```

```

-- truth table for "or" function
CONSTANT or_table : stdlogic_table := (

```

```

-----
-- U      X      0      1      Z      W      L      H      -
-----
( 'U', 'U', 'U', '1', 'U', 'U', 'U', '1', 'U' ), -- | U |
( 'U', 'X', 'X', '1', '1', 'X', 'X', '1', 'X' ), -- | X |
( 'U', 'X', '0', '1', 'Z', 'X', '0', '1', 'X' ), -- | 0 |
( '1', '1', '1', '1', '1', '1', '1', '1', '1' ), -- | 1 |
( 'U', '1', 'Z', '1', 'Z', 'X', 'X', '1', 'X' ), -- | Z |
( 'U', 'X', 'X', '1', 'X', 'X', 'X', '1', 'X' ), -- | W |
( 'U', 'X', '0', '1', 'X', 'X', '0', '1', 'X' ), -- | L |
( '1', '1', '1', '1', '1', '1', '1', '1', '1' ), -- | H |
( 'U', 'X', 'X', '1', 'X', 'X', 'X', '1', 'X' ) -- | - |);

```

```

-- truth table for "xor" function
CONSTANT xor_table : stdlogic_table := (

```

```

-----
-- U      X      0      1      Z      W      L      H      -
-----
( 'U', 'U', 'U', 'U', 'U', 'U', 'U', 'U', 'U' ), -- | U |
( 'U', '0', 'X', 'Z', '1', 'X', 'X', 'X', 'X' ), -- | X |
( 'U', 'X', '0', '1', 'Z', 'X', '0', '1', 'X' ), -- | 0 |
( 'U', 'Z', '1', '0', 'X', 'X', '1', '0', 'X' ), -- | 1 |
( 'U', '1', 'Z', 'X', '0', 'X', 'X', 'X', 'X' ), -- | Z |
( 'U', 'X', 'X', 'X', 'X', 'X', 'X', 'X', 'X' ), -- | W |
( 'U', 'X', '0', '1', 'X', 'X', '0', '1', 'X' ), -- | L |
( 'U', 'X', '1', '0', 'X', 'X', '1', '0', 'X' ), -- | H |
( 'U', 'X', 'X', 'X', 'X', 'X', 'X', 'X', 'X' ) -- | - |);

```