REASONING ABOUT THE VHDL STANDARD LOGIC PACKAGE SIGNAL DATA TYPE

J. W. Gambles[a] and P. J. Windley[b]

[a]Microelectronics Research Center, Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, New Mexico 87131

[b]Laboratory for Applied Logic, Department of Computer Science, University of Idaho, Moscow, Idaho 83843

Abstract

Formal verification methods provide a way to prove that a circuit structure correctly implements its specification. Low level gate and transistor logic circuits can be verified using methods such as symbolic simulation. Higher level circuits can be verified using theorem–proving methods, providing a multi-leveled approach to complex device verification. Correct modeling of many VLSI circuits requires a signal value data type that includes some degree of strength indeterminacy. The VHDL Standard Logic Package includes such a signal value data type, **t_wlogic**, that includes 46 unique values. In this paper we provide a foundation for the **t_wlogic** values and their resolution function that provides a possible link between simulation and theorem-proving. A formalization of the low level signal values used by design tools also facilitates reasoning about CAD tools.

Keyword Codes:
Keywords:

1 Motivation

Many VLSI design techniques involve the use of varying sized transistors for interconnected device outputs. Under certain circuit conditions the output of one device will overdrive the other connected outputs. In order to correctly simulate these circuits, a signal value model that combines three signal states, 1, 0, and X with multiple signal strengths, in such a way that there is a degree of strength indeterminacy is required [4]. The VHDL Standard Logic Package (SLP) defines such a signal value data type, **t_wlogic**, that includes 46 unique values [5]. In this paper, we provide a foundation for the **t_wlogic** values and resolution function based on an extension of Bryant's lattice-theoretic approach. This foundation facilitates reasoning about the **t_wlogic** signal value model. The imperative style signal value resolution algorithms and look-up tables (LUTs) used in many logic simulation programs are not appropriate for reasoning. The ability to manipulate these lower level signal values in a theorem–proving environment offers a more flexible interface to lower level circuit verification methods such as symbolic simulation.

2 Modeling Signal Values

Few modern VLSI circuits are designed using only classical logic gates [4]. In designs using pass-transistor, tri-state, and pre-charge logic, it is common for circuit nodes to be driven from more than one circuit device. These multiple drivers are designed to have differing drive strengths in order for one to dominate over another in cases of contention. The drive strength is considered to be closely related to current drive or charge sourcing/sinking capability [3, 9]. These design techniques dictate that digital logic circuit models and simulations must be capable of including devices down to the transistor level. Transistors, at the logic level, are modeled as simple switches, with electrical properties such as threshold voltage abstracted away. The physical attributes such as transistor size can be represented by the strength of the signal that the transistor drives.

2.1 The Lattice Model of Signals

Bryant developed a *lattice theoretic* data type for his original MOS switch-level simulator, MOSSIM, where the members of the signal value data type represent the combination of *logic state*, from the set {1, 0, X}, and *drive strength* [2, 9, 11]. The signal values form a partially ordered set (poset), with their order based on strength dominance when circuit signal values are interconnected.

A lattice structure is described through the notion of immediate superiors or *covers*. For two members, a and b, of a poset, *a covers b* if and only if $a > b$ and there exists no member x of the poset such that $a > x > b$ [1]. A lattice can be represented in graph form. The vertices of the graph are the members of the poset and the edges of the graph represent the covers. If the graph is drawn such that whenever x covers y then the vertex x is *higher* than the vertex y it is called a *Hasse diagram* [1]. Figure 1 shows an example of the kind of signal value lattice Bryant defined [9, 11]. This example has three strengths; "strong", "weak", and "disconnect". A strong true is depicted as "S1", a strong false is "S0", a weak true is "W1", a weak false is "W0". A strongly driven unknown is "SX", a weakly driven unknown is "WX", and "Z" represents circuit "disconnect". By definition, the logic state of disconnect is unknown. The least-upper-bound (*join*) of two or more signal values provides a signal resolution function that determines the result of the interconnection of nodes driven toward those signal values. For example, the result of interconnecting signals S1 and S0 is a signal of value SX. Similarly, the join of the two signals W0 and S1 is S1.

2.2 The Unknown State

The unknown state can be the result of a node connected to two outputs, with one driving toward true and the other driving toward false, neither one having sufficient strength to dominate the other, resulting in an invalid logic level. In many logic simulators, the unknown state is also used to model a node whose voltage is not yet known. Combining

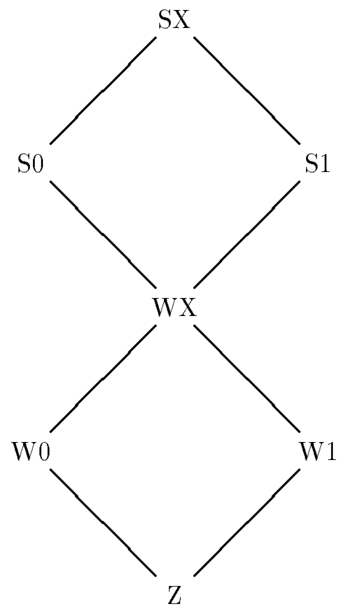

Figure 1: Bryant's Signal Lattice for Two Strengths

the cases of *invalid* logic level and *valid but not known* into a single unknown state simplifies the simulation algorithm. A rigorous modeling of the effects of unknown but valid states would require determining Boolean satisfiability, an NP-complete problem [3]. To avoid an exponential algorithm, many simulators do not attempt to distinguish between valid but not known and invalid logic levels. This approach can make the simulator pessimistic since it may propagate the unknown state when resolving other circuit nodes [3].

2.3 Extending the Lattice Model

Bryant abandoned the logic state/strength combination approach to signal values for the MOSSIM II simulator, stating "while this approach at first seems very elegant, it cannot adequately describe the effects of transistors in the X state [3]." Cameron and Shovic demonstrated that the problem with the X state can be overcome by extending the signal strengths to include a possible *range*, giving the signal values some degree of strength indeterminacy [4]. Solving the problem of the unknown state by extending the signal values to include strength indeterminacy may have an advantage over the switch graph approach in that the latter *requires* working at the transistor switch-level. The indeterminate strength approach supports accurate modeling at the switch-level but also allows higher level models, including wired-logic, tri-state buses, as well as combining system blocks implemented with different technologies such as TTL, NMOS, CMOS, and ECL [5].

2.4 Related Work

Cameron and Shovic called their signal value data type STATE. A four strength, 21 value, STATE signal model has become the basis for the BOLT HDL and NOVA logic simulator [10]. We have developed an extended lattice representation for STATEs [8, 7] and used the lattice and resolution function (join) to demonstrate an automatic translation and verification of circuits described in BOLT using the HOL theorem proving system [6]. The commutative, associative, and idempotent properties of the join operation pass on to the signal value resolution function the desired properties of order invariance and the validity of replacing a subset of interconnected driver values by their resolved value. This last property can serve to prevent the resolution function from over pessimistic propagation of the unknow logic state. In addition to providing a mathematical foundation to the STATE resolution function, the Hasse diagram of the STATE lattice provides a useful, quick, visual understanding of the interconnection of signals and has now been included in the NOVA manual [10].

2.5 The t_wlogic Data Type

The VHDL Standard Logic Package **t_wlogic** data type defines five fully ordered strengths:

- Forced (F)
- Strong resistive (R)
- Weak resistive (W)
- High impedance (Z)
- Disconnect (D)

and represents a further extension of the STATE data type:

1. The indeterminate strengths are allowed to range over *all* possible strengths, including Disconnect.
2. An uninitialized value is included to representand valid but not known signals that is distinct from the invalid logic level.

The true and false **t_wlogic** values that include a degree of strength indeterminacy are represented by the triple **bdK** where:

bd represents a *range* of strengths:

b is the strongest possible strength (F $\geq$ **b** $\geq$ D) which sets a lower bound on the strength of a signal that can overdrive this signal;

d is the weakest possible strength (**b** $>$ **d** $\geq$ D) which sets a upper bound on the strength of a signal that this signal can overdrive.

K is 1 or 0;

For the cases where $\mathbf{b} = \mathbf{d}$ (i.e. there is no strength indeterminacy) the value is represented by the double **bK**.

The **t_wlogic** values corresponding to the unknown state are represented as a triple **pq**X where:

p is the strongest possible strength driving toward 0 (F $\geq$ **p** $\geq$ D) which sets a lower bound on the strength of a signal that can overdrive this state to a 1;

q is the strongest possible strength driving toward 1 (F $\geq$ **q** $\geq$ D) which sets a lower bound on the strength of a signal that can overdrive this state to a 0.

For the cases where $\mathbf{p} = \mathbf{q}$ the value is represented by the double **p**X.

There are two single character values:

1. The value U is included to represent the valid but not known (uninitialized) value.
2. The value D represents the unknown logic state at the disconnect strength.

A Hasse diagram of the 46 values and 73 covers of the extended **t_wlogic** lattice is shown in Figure 2. Given the lattice represented by the Hasse diagram, the least upper bound of any subset of members of **t_wlogic** represents the resolution of contending circuit elements being driven toward those values. As an indication of the expressive power of this lattice treatment, the information expressed in Figure 2 is presented in a 46 X 46 look-up table, covering nearly five pages in *The* VHDL *Handbook* [5]. In addition to providing a framework for reasoning about **t_wlogic** signal values and their resolution function, the Hasse diagram also provides a useful, quick, visual understanding of the interconnection of signals during manual analysis of circuits.

3 Example

A static RAM cell, implemented with gate level and pass transistor primitives, can be used to demonstrate the resolution function (Figure 3). This circuit makes an interesting example because without a signal value representation and resolution function that supports output dominance this circuit cannot be correctly modeled. Fundamental to the operation of this circuit is that the output strength of pass-transistor `M1` dominates the output of feedback inverter `Inv2` to force node `n1` to the state of the input `d` while signal `wr` is 1 (high voltage). The feedback inverter `Inv2` then acts to store the state after the gate of `M1` goes to 0, turning the transistor off. Assuming that the strength of the signal at `d` and the size of `M1` are such that `M1` passes a forced signal (e.g. F0, F1, or FX) and that `Inv2` is sized such that its output strength is strong resistive (e.g. R0, R1, or RX) a resolution function based on the least-upper-bound of values from the lattice of Figure 2 does produce the desired result:

1. The join of an input F0 and a feedback R1 is F0.

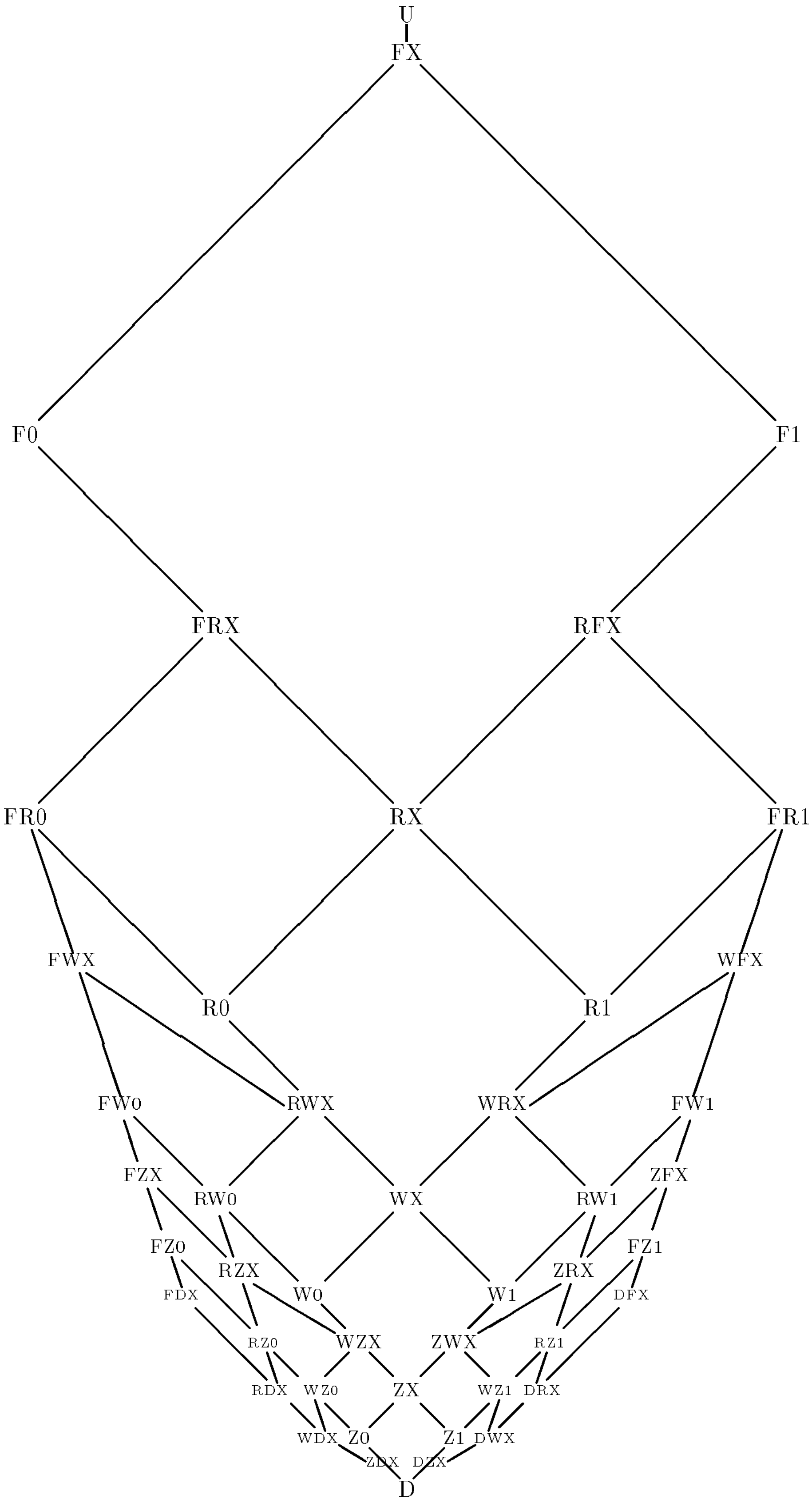

Figure 2: Signal Lattice for VHDL Standard Logic Package

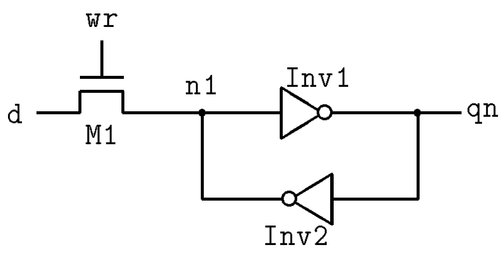

Figure 3: Example RAM cell

2. The join of an input F0 and a feedback R0 is F0.
3. The join of an input F0 and a feedback RX is F0.
4. The join of an input F1 and a feedback R1 is F1.
5. The join of an input F1 and a feedback R0 is F1.
6. The join of an input F1 and a feedback RX is F1.

Note that the presence of unknown values does not require the propagation of an unknow result. A complete verification of this circuit using the STATES data type is included in [6].

4 Conclusion

In this paper we have provided a foundation for the VHDL Standard Logic Package **t_wlogic** signal data type. This foundation provides the basis for formal reasoning about these values. The lattice development is necessary because neither the previous algorithm or LUTs used for logic simulation programs are suitable for reasoning about such signals. This is important because it provides for the interface between verification through theorem-proving and lower level methods to possibly extend below the point where all signals are represented strictly as booleans. This may include such structures as high level blocks that are interconnected via tri-state busses. The ability to reason about the structures and models used by the CAD tools also enables reasoning about those tools.

5 Acknowledgements

This research was supported in part by NASA under Space Engineering Research Grant NAGW-1406 and by the NSF under Research Initiation Grant MIP-9109618.

References

[1] G. Birkhoff. *Lattice Theory.* American Mathematical Society, 1948.

[2] R. E. Bryant. "MOSSIM: A Switch-Level Simulator for MOS LSI". In *Proceedings of the 18th ACM Design Automation Conference*, pages 786–790, 1981.

[3] R. E. Bryant. "A Switch-Level Model and Simulator for MOS Digital Systems". *IEEE Transactions on Computers*, Vol. C-33:160–177, February 1984.

[4] K. B. Cameron and J. C. Shovic. "Calculating Minimum Logic State Requirements for Multi-Strength Multi-Value MOS Logic Simulators". In *1987 IEEE International Conference on Computer Design: VLSI in Computers & Processors*, pages 672–675, Rye Brook, New York, October 1987. IEEE Computer Society Press.

[5] D. R. Coelho. *The VHDL Handbook*. Kluwer Academic Publishers, Boston, 1989.

[6] J. W. Gambles. Incorporating formal verification into VLSI design methodology. Master's thesis, University of Idaho, May 1992.

[7] J. W. Gambles and P. J. Windley. "A Verification Logic Representation of Indeterministic Signal States". In *Third NASA Symposium on VLSI Design*, pages 10.2.1–10.2.12, Moscow, Idaho, October 1991. NASA Space Engineering Research Center, University of Idaho.

[8] J. W. Gambles and P. J. Windley. "An HOL Theory for Logic States with Indeterminate Strengths". In P. J. Windley, M. Archer, K. N. Levitt, and J. J. Joyce, editors, *The Proceedings of the International Tutorial and Workshop on the HOL Theorem Proving System and its Applications*. IEEE Computer Society Press, February 1992. Also Technical Report LAL-91-08, University of Idaho Laboratory for Applied Logic, October 1991.

[9] J. P. Hayes. "A Unified Switching Theory with Applications to VLSI Design". *Proceedings of the IEEE*, Vol. 70(No. 10):1140–1151, October 1982.

[10] NASA Space Engineering Research Center for VLSI Systems Design, University of Idaho. *NOVA User's Manual*, July 1991.

[11] J. D. Ullman. *Computational Aspects of VLSI*. Computer Science Press, 1984.