

設計記述言語の意味論と シミュレーションの精度について

淡海功二 安浦寛人 田丸啓吉

京都大学 工学部 電子工学科

あらまし 論理シミュレーションの精度を議論するための枠組みを提案する。ハードウェア設計記述言語の自然な意味論を定義し、非決定性イベント系列モデル (NES モデル) の考え方に基づいてシミュレーション結果と標準的な意味論との違いを定量的に比較する手法を提案する。この手法を用いて、現在一般的に使われているシミュレーションアルゴリズムの精度を調べ、その精度を左右する原因について整理する。本研究の成果は、今後の設計言語の標準化の動きの中でのシミュレーションの在り方に一つの方向を与えるものである。

Semantics of Hardware Description Languages and Accuracy of Logic Simulation

Koji TANKAI, Hiroto YASUURA and Keikichi TAMARU

*Department of Electronics, Faculty of Engineering, Kyoto University
Kyoto 606, Japan*

Abstract We propose a definition of accuracy measure for logic simulation. Introducing a natural semantics for hardware design languages, we propose a method to evaluate accuracy of simulation by a comparison between the simulation result and the natural semantics based on the idea of the Nondeterministic Event Sequence model. We apply the method to simulation algorithms which are popular in practical logic simulation and analyze causes of the variations of accuracy. The result of this paper points out a direction of the research of simulators in the era of the standard design languages.

1 はじめに

機能 / 論理シミュレーションの性能は、主として速度と精度の2つに大別され、速度と精度がトレードオフの関係にあることはよく知られている [1]。速度性能に関しては、従来から単位時間あたりのイベントの評価数やゲートの処理数などの定量的な指標が利用されてきた。シミュレータの性能に関してまず関心をもたれるのはこの速度性能であると言っても過言ではないだろう。一方、シミュレーションの精度も重要な性能の1つであるが、一般的な定義が確立しているとはいえない。その原因は、精度を議論するための基準、および基準と対象アルゴリズムの比較手法が不明確であったからと考えられる。

定量的な尺度について議論するためには、機能 / 論理シミュレーションの精度に限らず一般に、基準のモデリング及び基準と対象の比較手法が明確に定義されていなければならない。従来では、実際の回路そのものの動作を基準として思い浮かべ、それをもとに議論することが多かったと考えられる [1]。しかし実際の回路動作を基準とすることは、次に挙げるような理由から厳密さに欠けるおそれがある。

- (1) 実際の回路では、製造上のばらつきや動作環境あるいは温度などの変化によって回路動作自体にばらつきが生じ、ある回路の厳密な動作を特定することは困難であること
- (2) たとえばばらつきがないと仮定した場合でも、物理量の実測値には大なり小なり必ず測定誤差が含まれること

などがその理由である。また、現実の世界とモデル化された世界を比較する際においても、解釈の方法が多様であり、議論する人によって異なっていたため、基準だけでなく比較手法も曖昧であったと考えられる。あるモデリングされた回路動作を基準にした比較の例 [1] も見られるが、多くの場合は、(例えば論理 / フリップフロップ / レジスタトランスファレベルの違いや、標準 / 最大最小 / 慣性遅延の違いのような) モデリング自体の違いに起因する精度である。これらはシミュレータやシミュレーションアルゴリズムの精度を示しているものではないため、本稿の目的とは異なる。いづれにせよ、精度の基準、比較法、あるいは定量法を厳密に定義したものはいずれも見あたらない。

以上の議論から、シミュレーションの精度を定量化するためには、基準となる回路動作のモデリングを行い、そのモデルに対するシミュレーションの精度を議論する

手法が現実的であると考えられる。ここで、そのモデルの定義、および比較(解釈)手法の定義を明確な形で示すことが重要な問題である。モデルと解釈手法の定義は、言い換えれば回路動作の意味付けの問題であり、意味論が厳密に定義される必要がある。

一方、近年、VHDL[2]、Conlan[3]、UDL/I[4]などのハードウェア記述言語(以下HDLと記す)の標準化に関する作業、研究が盛んに行われている。その中で、言語の意味論(semantics)の重要性が顕在化してきている [5] が、特に本稿において重要である事柄を要約すると、以下の2点になる。

- (1) HDLが不特定多数のユーザ、シミュレータ設計者に対して公開されるため、形式的な意味論が明確、厳密、かつ理解し易い形で与えられる必要がある。また1つのHDLに対し、複数の処理系が作成され得る。
- (2) 今後HDLはシミュレータだけでなく、他の検証ツールや合成ツールなどのCADツールの入力としても用いられる可能性がある。そのため、HDLの文法および意味論が様々なCADツールにとって扱いやすい形で定義されていなければならない。

これらは、標準HDLが公開され各ツールから中立的な立場に立つことにより、従来のような、シミュレータの意味すなわち言語の意味、という状況が今後変化していくことを示している。また、合成系に対する仕様を厳密に記述するためには、不確定性を含む記述ができなければならない。なぜならこの段階では、製造上や動作環境に起因するハードウェア動作の不確定性だけでなく、設計自体にも不確定性が存在するからである。すなわち、設計の中に未指定の情報(don't care)が少なからず含まれており、設計の進行にともなってこれらの未指定の情報が確定される。このような状況では、シミュレータが基準HDLの意味を完全に模擬するのは困難であり、また必ずしもその必要はない。

上述の議論を踏まえ、基準のモデリングに関し、基準HDLの表す回路の動作が自然現象のモデリングであり、シミュレータは記述された回路モデルを近似的に模擬するものである、という立場に立つのが望ましいと考える。この意味において、シミュレーションの精度とは基準言語の意味論における回路モデルの動作とシミュレータの動作との違いである。信号値や時間の離散化にともなう誤差、つまり物理現象とモデリングされた基準との間の誤差については、ここでは取り扱わない。

本稿ではこのような前提を基に、精度の定式化を行う。2節で基準となるモデルを厳密に定義し、3節で精度の定式化を行う。4節では精度を決定する要因を、具体例を示しながら解説する。

2 自然な意味論と NES モデル

2.1 基本方針

定量的な尺度について議論するためには、基準のモデリング及び基準と対象の比較手法の定義が重要な問題である。

まず基準のモデリングに関して本稿では、前節でも述べたように、基準 HDL の動作が自然現象のモデリングであり、シミュレータは記述された回路モデルを近似的に模擬するものである、という立場に立つ。この理由としては、以下のものが挙げられる。

- (1) 基準を自然現象でなく対象と同じデジタルのモデリングの中に設定することにより、精度の比較が容易になる。
- (2) HDL では、形式的に厳密な意味付けが可能である。言い換えれば、基準および対象回路の動作の解釈が厳密に行うことができる。

次に、基準モデルと対象モデルの比較法に関しては

- (1) アルゴリズムの解析的な比較
- (2) ベンチマーク (特定回路, 特定入力) に対する入出力関係の比較

が考えられる。(1) はアルゴリズムの正当性にもかかわる問題であり、公理的な手法が要求されると考えられる。しかし一般にシミュレータのアルゴリズムは非常に複雑であり、アルゴリズムを静的に解析して比較することは現在のところ困難な状況である。(2) はそのアルゴリズムにおけるベンチマークに対する動作を比較する手法であり、実現は比較的容易であると考えられる。しかし回路構造や入力波形などの入力が与えられて初めて精度が定量化される点がデメリットである。理想的には全ての回路構造および入力パターンを考慮することにより、そのシミュレーションアルゴリズムの精度が与えられるが、全ての回路構造と入力パターンを与えることは現実的に不可能であると言わざるを得ない。しかしこの点に関しては、精度の違いが反映されるような特徴的なベンチマーク回路を用いることにより、近似的にそのアルゴリズム自身の精度を推し量ることが可能であると考えられる。また少なくとも、アルゴリズムの性質を知ることができる。

2.2 自然な意味論

基準モデルの意味論を、本稿では自然な意味論 (Natural Semantics: 以後 NS) と呼ぶ。基準モデルとしての NS に求められる本質的に重要な点は、意味論が自然な形で厳密に定義されることである。自然な形とはハードウェアの物理的な不確定性や設計の不確定性、あるいは信号値の解釈などを設計者の感覚に近い形で表現できるということであり、これらの現象を曖昧さなく厳密に表せるべきであると考えられる。

回路動作のモデリングやシミュレーションアルゴリズムにおいて、解釈が多様であり処理系によって扱いが異なるため、しばしば問題として挙げられる点について以下に示す。ただし本稿では、時間の概念に関しては離散時間を仮定する。

- (1) 信号値
- (2) 遅延モデル
- (3) イベントハンドリング
- (4) 単位時間

これらの項目に対して自然で厳密な解釈を与えることが、NS を定義することであると考えられる。逆に言えば、これがシミュレーションの精度に関わる要因であるということもできる。

(1) の信号値に関しては、シミュレータ毎に様々な信号値体系が存在する。しかし現実の値である $\{0, 1, Z\}$ 以外の値に関しては、多くは値的な不確定性を表す不定値 (X, U, \dots) であったり、時間的な不確定性である遷移状態を表す値 (R, F, \dots) などである。これらの値は実際の回路内に発生する値ではなく、シミュレーション用の信号値であり、いわば架空の値である。これらの値は実際にはある単位時間内では $\{0, 1, Z\}$ のいずれかの決定的な値をとっていると考えられる。このような理由から NS における信号値体系は $V_{NS} = \{0, 1, Z\}$ とし、 X のような不定値などの値は V_{NS} 上の系列集合によって表されると考えるのが自然である。この変換規則については、2.4 節で説明する。

(2) の遅延モデルについて、NS では、遅延は最小値と最大値の組によって表される区間内のいずれかの値をとる、という非決定的な解釈が自然であると思われる。実際の回路素子の遅延は、製造条件や動作環境によってばらつきを生じるが、このばらつきを自然に表現できる。また、設計初期の段階では一般に遅延は確定していないが、大まかな幅を持った曖昧な遅延を厳密に表すことができ、設計者の意図を正確に反映できると考えられる。

(3) のイベントハンドリングの問題は、0 遅延の計算順序の問題である。これは 2.3 節で述べるように全ての順序関係を考慮するのが自然である。

(4) については、回路モデル中の素子に現れる遅延の最小桁を NS の単位時間とする。単位時間がこの値より大きいと丸め誤差が生じる可能性がある。またこの値より小さくすることは無意味である。

以上の議論より、本稿では NS の基本的な計算モデルとして、NES モデル (Nondeterministic Event Sequence Model)[6] を用いる。NES モデルを用いることにより、本節で述べた非決定的な動作を正確かつ厳密に定義できる。次節で NES モデルについて簡単に説明を加える。

2.3 NES モデル

NES モデル [6] の動作は、機能 / 論理シミュレーションにおいて従来から広く用いられてきたイベントドリブン法の一般化という形で特徴付けられる。イベント e は信号線 (ブレースと呼ぶ) $p \in P$ と信号値 $v \in V$ の組 $e \in E = P \times V$ であり、ブレースの信号値の変化を表す。信号値の時系列は、イベントの一次元系列によってモデル化され、そのイベントの順序が時間的発生順序を表す。つまり 1 つのイベント系列は 1 つの時間的なイベントの全順序関係を表している。一般にはシミュレーションの出力は、イベント間の時間的半順序関係すなわちイベントの同時発生を含んでいる。しかし NES モデルでは、半順序関係はそれと矛盾しない全ての全順序関係の集合として表現される。

時間のモデリングは、時刻の経過を表す特別なイベントによって行われる。時間イベントは $ps, ns, \mu s, ck$ などの時間単位とその値の組である。時間をイベントとして扱うことにより、時間の概念と独立した因果関係の表現や、複数の時間単位およびクロック時間の混在が可能になっている。

ハードウェアを構成する機能ブロックの動作は、入力イベント系列の集合から、入力・出力イベント系列の集合を計算する抽象機械によってモデル化される。抽象機械はイベント系列中の 1 つのイベントを指すポインタを持ち、各イベント系列を左から右へ方向に走査する。イベント系列に対する操作は、ポインタ位置のイベントの読み取りとポインタの直後 (右) へのイベントの挿入のみが可能である。つまり、入力イベント系列の中に出力イベント系列が挿入されることによって、出力イベント系列 (の集合) が生成される。抽象機械 M は $M = (Q, I, \delta)$ によって表される。ただし、 Q は状態の集合、 I は初期

状態の集合、 $\delta : Q \times E^* \rightarrow 2^{Q \times E^*}$ は状態遷移関数である。

2.4 信号値の解釈

前節で NS における信号値体系は $\{0, 1, Z\}$ であることを述べた。しかし現実には、シミュレータ毎に様々な信号値体系が存在する。例えば、不定値を表す X, U, D 、信号の立上り / 立下りの遷移状態を表す R, F 、静的・動的ハザードを表す $0^*, 1^*, R^*, F^*$ [7]、ストレングスの概念などがある。これらは、その意味、表記法ともに多種多様であるため、精度の議論を目的として NS との比較を行うためには、信号値の解釈、すなわち上記の信号値から NS の信号値への変換規則を定義する必要がある。ここではその変換規則を解釈関数と呼ぶことにする。

上記の信号値は、単にある単位時間内の値を表すものだけでなく、信号値の時間的な変化を表している場合がある。そのため解釈関数を定義するには、信号値の時間的な変化を考慮しなければならない。例えば、立ち上がりの遷移期間を表す信号値 R について、 $\{0RR1\}$ の解釈を示すと、

$$\{0RR1\} \rightarrow \{0001, 0011, 0111\}$$

つまり、 $0101 \notin \{0001, 0011, 0111\}$ と考えるのが、一般的であろう。そのため、単純に (前後の時刻における値を考慮せずに) $R = \{0, 1\}$ とすることはできない。このような理由から、解釈関数は信号値系列 (の集合) から信号値系列の集合への変換規則として定義されなければならない。すなわち、NS における信号値の集合を V_{NS} 、比較対象となるシミュレータの信号値の集合を V_S とすると、解釈関数 f は一般に

$$f : V_S^* \rightarrow V_{NS}^*$$

という形で表される。

以下では一例として、一般の機能 / 論理シミュレーションでよく頻繁に用いられている不定値 X の本稿での解釈、およびその解釈関数について説明する。ただしここでは簡単のため、 $V_{NS} = \{0, 1\}$ とする。

不定値 X は、0 か 1 か (あるいは Z か) 分からない値として、機能 / 論理シミュレーションでは頻繁に用いられる値であるが、その解釈は多様である。例えばある信号線が値 X をとり続けている場合、値は分からないが安定しているのか、各単位時間毎に変化する可能性を持っているのか、あるいは発振しているのか等、様々な解釈

が有り得る [7]. ここでは離散時間を仮定しているため、単位時間内での変化を含む発振などは考慮しない。また、安定かどうかの解釈については、次のように考える。 X が安定かどうかという性質は不定値 X の発生する回路構造やモデリングに依存するが、得られた出力から識別するのは一般に困難である。そのためより悲観的に、各単位時間毎に変化する可能性がある、という立場をとる。すなわちこの解釈関数 f_X は、 $V_X = \{0, 1, X\}$, $V_{NS} = \{0, 1\}$ として、 $f_X : V_X^* \rightarrow V_{NS}^*$ である。このとき f_X は

$$f_X(v_1 v_2 \cdots v_n) = \begin{cases} \{v_1\} \circ f_X(v_2 v_3 \cdots v_n) & \text{if } v_1 \in V_{NS} \\ V_{NS} \circ f_X(v_2 v_3 \cdots v_n) & \text{if } v_1 = X \end{cases}$$

$$f_X(\varepsilon) = \{\varepsilon\}$$

で与えられる。ただし、 $v_1, v_2, \dots, v_n \in V_X$ であり、二項演算子 \circ は、系列またはその接続 $a_i, b_j \in \Sigma^*$ から成る集合に対し、

$$\{a_1, \dots, a_n\} \circ \{b_1, \dots, b_m\} = \{a_1 b_1, \dots, a_1 b_m, \dots, a_n b_1, \dots, a_n b_m\}$$

と定義される。例えば n 単位時間連続する X は、 2^n 個の信号値系列の集合で表される。

3 精度の定式化

全ての回路を C , C 中のある回路を $c \in C$, 全ての入力系列の集合を I , c に対する入力系列全ての集合を $I \subset I$, ある入力系列を $i \in I$ とする。 C 中の全ての信号線 (入力信号線を含む) を T , 回路 c におけるシミュレーションの観測点の集合 $T \subset T$ を定める。あるシミュレータ (シミュレーションアルゴリズム) S および NS を用いて、回路 c に対し i を入力としてシミュレーションした結果をそれぞれ $O(S, c, T, i)$, $O(NS, c, T, i)$ と表す。

ここで $O(S, c, T, i)$ は、得られたシミュレーション結果において前節で示したように信号値 $V_S - V_{NS} = V_S \cap \overline{V_{NS}}$ について解釈を与え (すなわち $V_S^* \rightarrow V_{NS}^*$ の変換を行い), NS 上のイベント系列の集合に変換したものである。また $O(NS, c, T, i)$ は、 NS による出力結果に関して T についてのみ注目したものである。この理由は、 NS によるシミュレーション結果が回路記述内の全ての信号線に関するイベントを含んでいるからである。具体的には全ての出力イベント系列から T に含まれない信号線に関するイベントを取り除き、同じイベント系列はすべてマージするという操作が行われた結果を表す。

上記の表記法を用いると、回路 c と入力系列 i の観測

点 T に関するシミュレーションアルゴリズム S の精度は

$$f(O(S, c, T, i), O(NS, c, T, i)) \quad (1)$$

で与えられる。全ての回路構造および全ての入力系列に対する S の精度 (一般化精度と呼ぶ) は

$$f(O(S, C, T, I), O(NS, C, T, I))$$

で与えられるが、ここでは取り扱わない。 f として考えられるものに、集合の要素数の比、集合の包含関係などが挙げられる。 f を変えることによって、異なった側面から見た精度の定義が可能である。

次に (1) 式の f を具体的に定義することによって、精度を定式化する。シミュレーション出力の性質に関し、一般的によく用いられる表現として、「楽観的、悲観的」というものがある。この性質は、 $O(S, c, T, i)$, $O(NS, c, T, i)$ をそれぞれ O_S, O_{NS} と略記すると

$$O_S \subset O_{NS} \quad (2)$$

のときは楽観的、

$$O_S \supset O_{NS} \quad (3)$$

のときは悲観的、という形で表すことができる。そのため、このようないずれの性質も反映する f の定義が必要であると考え。そこで本稿では、次のような2種類の精度を定義する。

・過小評価精度

$$A_{\text{under}} = \frac{|O(NS, c, T, i) - O(S, c, T, i)|}{|O(NS, c, T, i)|} \quad (4)$$

・過剰評価精度

$$A_{\text{over}} = \frac{|O(S, c, T, i) - O(NS, c, T, i)|}{|O(S, c, T, i)|} \quad (5)$$

ただし、 $A - B = A \cap \overline{B}$, $|A| = \text{number}(A)$ 直感的には、過小評価精度は基準モデルにおける解の中で、 S では得られなかった解の割合であり、過剰評価精度は S によって得られた結果の中で、基準モデルに含まれないものの割合である。(4),(5)式を用いると、(2),(3)式で示される性質は、それぞれ次のように表すことができる。

$$A_{\text{under}} \neq 0, A_{\text{over}} = 0 \quad (6)$$

$$A_{\text{under}} = 0, A_{\text{over}} \neq 0 \quad (7)$$

(6)式が成立する時は楽観的、(7)式の時は悲観的である。例えば X を用いたシミュレーションにおいては、シミュレーション結果が(7)式で表される悲観的な場合に含まれることはよく知られている [7].

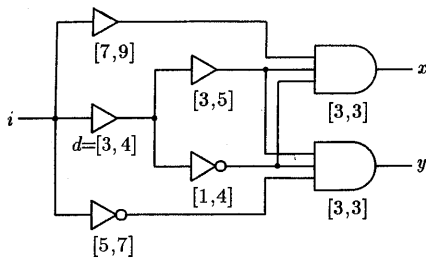


図 1.1 回路 c , 入力系列 $i = \{i\}$, 観測点 $T = \{x, y\}$

4 シミュレーションアルゴリズムと精度

4.1 概説

自然な意味論 NS に基づく回路動作を完全にシミュレートするための計算複雑度は、NP 困難であることが知られている [8][9]。これは換言すれば、ある回路 c とある定数長の入力イベント系列に対し、 c の出力信号線上にハザードが存在するかどうかを求める問題が、NP 完全であるということを示している。そのため NS を完全にシミュレートすることは、現実的とは言えない。一般に、実用されているシミュレータの多くは、回路規模と入力系列長に比例する程度の計算複雑度の中で実行可能なアルゴリズムを用いて、シミュレーションを行っている。

この場合多くのシミュレータが選択するアルゴリズムは、大きく分けて、楽観的なアルゴリズムを選択する方法と悲観的なアルゴリズムによる手法の 2 通りであると考えられる。この 2 つの手法は直感的には次のように説明される。NS の完全なシミュレーションは、非決定的な探索木の全探索に相当する [10]。この探索木のある 1 つの状態において非決定的な遷移先が複数存在する場合を考える。このとき、どれか 1 つ (あるいは一部分) の枝だけを探索するのが楽観的なアルゴリズムであり、全てを探索するのが完全な NS のシミュレーションである。また、複数の可能性をすべて信号値の問題に押し込めたものが悲観的なアルゴリズムであるといえる。この悲観的な場合の例として、2.4 節で述べた不定値 $X(U, D)$ 、遷移期間を表す $0^*, 1^*, R^*, F^*$ などが挙げられる。

4.2 シミュレーション精度の要因による分類

上述の意味において、NS を基準とした場合多くのシミュレータには誤差が含まれていると考えられる。以下にシミュレーションの精度を決定する要因を分類し、まとめる。

1) 信号値体系

前節でも述べたように多くのシミュレータでは、非決定

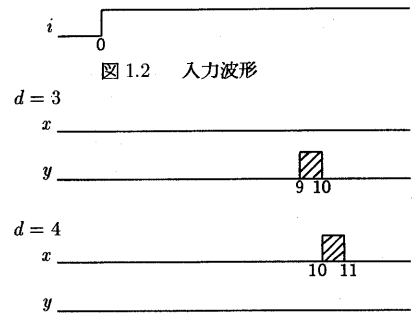


図 1.2 入力波形

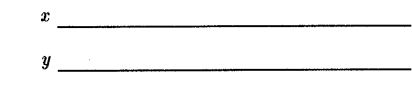


図 1.3 NS による出力波形 $O(NS, c, T, i)$

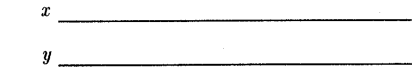


図 1.4 各々のゲートの遅延は最小値: O_1

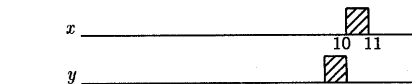


図 1.5 各々のゲートの遅延は最大値: O_2

的な意味を決定的なアルゴリズムでシミュレーションするために、 X, R, F, \dots などの値を用いる。これらの値そのものが現実存在するわけではなく、いわば架空の信号値であり、2.4 節で示したように別の現実の (NS の) 値によって解釈される。これらの値は、ファンアウトによって分かれて異なった径路を伝搬し、再取れんする場合などにおいて、情報損失を生じる。例えば、同じ現実の値を持つ X については、 $X + \bar{X} = 1, X \cdot \bar{X} = 0$ が成立するという情報が失われるのである。なぜなら再取れんによって出会った 2 つの X が同じものかどうか区別がつかないからである。この要因は一般に、シミュレーションを悲観的にする方向に働く [7]。

2) 遅延モデル

NS においては、最大 / 最小遅延が基準の意味であることを仮定した。ここで最大 / 最小遅延とは、最小遅延と最大遅延で表された区間内のいずれかの単位時間の整数倍の遅延で信号が伝搬することであり、その全ての場合が可能性としてあり得るという意味である。もしシ

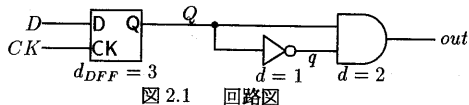


図 2.1 回路図

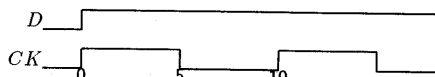


図 2.2 入力波形

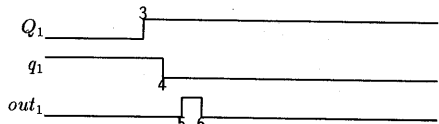


図 2.3 D が先であるとみなした場合の出力波形

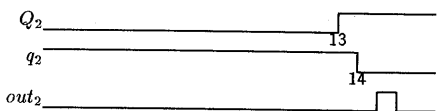


図 2.4 CK が先である場合の出力

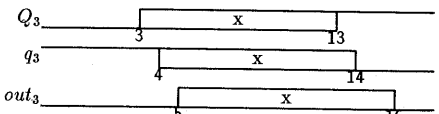


図 2.5 X を用いた場合の出力

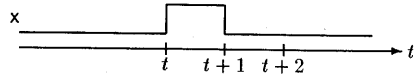
シミュレータがその中の1つを遅延の値として決定的に用いれば、そのシミュレーションは楽観的であるといえる。また仮に、遅延の不確定性を X を用いてシミュレーションしたならば、それは悲観的である。例を図 1 に示す。NS による出力系列は図 1.3 のようになるが、 x, y 同時にハザードが発生することはない。回路中の各々のゲートの遅延に関し、すべて最小値と考慮してシミュレーションした場合を S_1 、すべて最大値とした場合を S_2 とする。このときの過小評価精度 A_{under_1}, A_{under_2} 、過剰評価精度 A_{over_1}, A_{over_2} は、それぞれ式 (4), (5) より

$$A_{under_1} = A_{under_2} = \frac{2}{3}, \quad A_{over_1} = A_{over_2} = 0$$

であり、楽観的である。X を用いてシミュレーションした場合 S_3 の精度は、

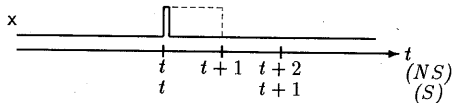
$$A_{under_3} = 0, \quad A_{over_3} = \frac{2}{5}$$

となり、悲観的である。



$$(nano-sec, t)(x, 1)(nano-sec, t+1)(x, 0)(nano-sec, t+2)$$

図 3.1 $T_{NS} = 1, T_S = 1$ の場合の波形とイベント系列表現



$$(nano-sec, t)(x, 1)(x, 0)(nano-sec, t+1)(nano-sec, t+2)$$

$$\text{or} \\ (nano-sec, t)(nano-sec, t+1)(nano-sec, t+2)$$

図 3.2 $T_{NS} = 1, T_S = 2$ の場合の波形およびイベント系列

3) イベントハンドリング

因果関係のないイベントの同時発生は、NS では全ての全順序集合として表される。これらのイベントに、あるシミュレータのアルゴリズムによって何等かの順序付けが行われた場合、言い換えれば NS での全ての全順序集合の中から1つの系列が選択された場合、シミュレーションは楽観的になる。逆に全順序集合の代用として不定値 X を用いた場合は悲観的になる。これらは特に非同期回路においてしばしば問題となることがある。例を図 2 に示す。D のイベントが先であるとみなした場合の出力系列は out_1 、CK が先であるとした場合を out_2 、X を用いた場合を out_3 とする。NS による出力は、集合 $\{out_1, out_2\}$ である。 S_1, S_2 による精度は、

$$A_{under(1)or(2)} = \frac{1}{2}, \quad A_{over(1)or(2)} = 0$$

で楽観的である。X を用いた S_3 による精度は、

$$A_{under_3} = 0, \quad A_{over_3} = \frac{2^{11} - 2}{2^{11}} = 1 - 2^{-10}$$

となり、悲観的である。

4) 単位時間

あるシミュレーションにおける単位時間が NS の意味での単位時間よりも大きい場合に、誤差が生じる可能性がある。これは混合レベルシミュレーションなどにおいてよく見られる現象であり、一般に高位レベルの方が単位時間が大きい。一例を図 3.1, 3.2 に示す。あるシミュレーションにおける単位時間 T_S を NS の単位時間 T_{NS} の 2 倍

にとつた場合(図 3.2), NSでの単位時間幅つまり幅 T_{NS} のパルスは 0 パルスになるか, あるいは消滅する.

4.3 精度に関するまとめ

前節において得られた結果を以下にまとめる.

- (1) 架空の信号値の導入によるシミュレーションの高速化は, 本質的な精度の低下をもたらす.
- (2) 遅延モデルにおける遅延の値の非決定性は, 高精度かつ高速なシミュレーションを絶望的にする [8].
- (3) シミュレータにおけるイベントハンドリングの不注意な操作(プログラミング)が精度に大きな影響を与える.
- (4) 単位時間の変更も, 精度に与える影響がきわめて大きい.

これらの点は従来からも直接/間接的に指摘されてきたことではあるが, 精度を定量化することにより, より鮮明に精度の違いを知ることができる. 今後のシミュレータ設計者は, それぞれのシミュレータの精度を明示することによって, シミュレータユーザの誤解を招かないように努めるべきであろう.

5 おわりに

本稿では, 機能/論理シミュレーションの精度の定量化を目的として, 基準モデルと比較(解釈)手法を定義し, 精度を定式化した.

ここで導入した自然な意味論を, 実用的な時間内で完全にシミュレートすることは困難である. しかし, 回路に対する設計手法を限定すれば, その限定された回路に対しては, 完全に, あるいは高い精度でシミュレーションを行うことが可能である. また, 同期式回路などにおいては, 検証のために必ずしも高い精度のシミュレーションを必要としないことはよく知られている. 大規模な回路の全体を同一精度でシミュレーションを行う必要はなく, 各部分が必要とする精度でシミュレーションを行えばよいと考えられる. 今後設計手法とシミュレーションの精度の関係, 精度が可変なシミュレータなどの研究が重要になると考える.

謝辞 日頃御討論いただく京都大学情報工学教室石浦菜岐佐氏, 並びに田丸研究室の諸氏に感謝致します.

参考文献

- [1] E.Ulrich, D.Herbert, "Speed and Accuracy in Digital Network Simulation Based on Structural Modelling", 19th Design Automation Conference, IEEE 1982

- [2] M.Shadad, et al., "VHSIC Hardware Description Language", *IEEE Computer*, 18(1985), pp94~103
- [3] R.Piloty, D.Borrione "The Conlan Project", *IEEE Computer*, 18(1985), pp81~92
- [4] 唐津修, "LSI 設計用言語標準化委員会の活動について", 電子工業月報, vol.30, No.9, pp.47~55 (Sep.1988)
- [5] 安浦寛人, "LSI 設計用記述言語の標準化における意味の統一について", 信学技報, VLD88-96 (Feb.1989)
- [6] 石浦菜岐佐, 矢島情三, "ハードウェア記述言語の意味付けのための非決定的な動作モデル", 信学技報, VLD89-3 (Apr.1989)
- [7] M.A.Breuer, A.D.Friedman: *Diagnosis & Reliable Design of Digital Systems*, Computer Science Press, Inc.
- [8] 安浦寛人, 石浦菜岐佐, "ハザード検出問題の計算複雑さについて", 信学技報, COMP86-64 (1986)
- [9] H.P.Chang, J.A.Abraham "The Complexity of Accurate Logic Simulation", ICCAD, 1987
- [10] 淡海, 安浦, 田丸, "ハードウェア記述言語の意味定義のための非決定的動作モデルシミュレータ", 信学技報, VLD89-76 (1989)
- [11] 阿都英志: 論理回路シミュレーション, CQ 出版社