初歩からのHDLテストベンチ

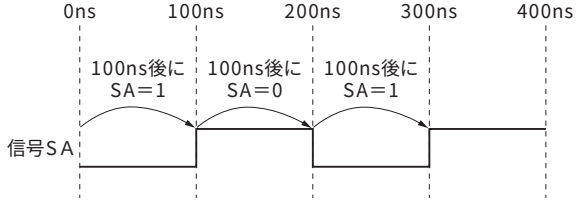
第3回　絶対遅延とソフトウェア風テストベンチの文法

安岡貴志

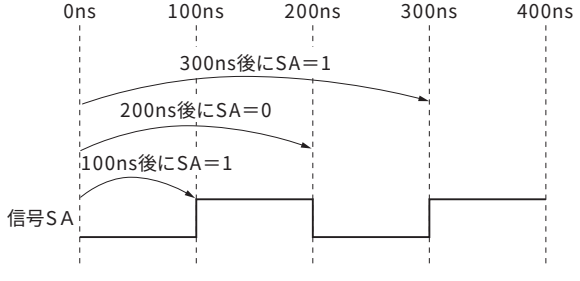
デバイスの記事　ビギナーズ

HDLで回路を記述できるようになったばかりで，これからテストベンチを書こうとしてる方を対象とした連載の第3回である．始めに絶対時間で遅延を記述する方法を解説し，サンプル回路の検証事例をもとに，遅延の効率的な記述法を説明する．（筆者）

前回（本誌2007年8月号，pp.116-124）まではテスト入力（検証対象回路の入力ポートに与える信号）を作るのに相対遅延を使っていました．今回は絶対遅延による表現を解説します．また，第1回（本誌2007年5月号，pp.70-79の特集1 第4章編集部注）で，テストベンチではHDLのすべての文法を，それこそC言語によるソフトウェア・プログラムのように使えると言いました．今回はその中からfor文によるループ表現を説明します．



（a）遅延を前の信号の変化からの相対遅延で表現



（b）遅延をシミュレーション開始時からの絶対遅延で表現

図1　遅延の表現

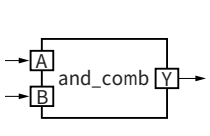
前の信号の変化からの相対関係で表現する相対遅延と，常に開始時点からの時間で表現する絶対遅延がある．

1．相対遅延と絶対遅延

相対遅延とは，前の信号の変化から何nsで代入を実行するというように，遅延を相対関係で表現する手法です［**図1（a）**］．前回までのテストベンチはすべてこの手法で書いてきました．これに対して絶対遅延とは，シミュレーション開始時点（シミュレーション時間0）から何nsで代入するというような，絶対時間で表現する手法です［**図1（b）**］．

ここでは，第1回で示した**図2**の仕様をもつ検証対象の回路に対して，**図3（a）**のテストベンチから，**図3（b）**の信号SAと信号SBのようなテスト入力を与える記述法を，相対遅延，絶対遅延を使って解説します．

編集部注：本連載の第1回は，本誌2007年5月号，pp.70-79の特集1 第4章「テストベンチの書き方を身に付ける」として掲載した．



（a）ブロック図

A	B	Y
0	0	0
1	0	0
0	1	0
1	1	1

ポートA，Bへの入力信号がいずれも‘0’のときポートYからの出力信号は‘0’

（b）真理値表

図2　検査対象の回路

回路の名前は，and-combである．1ビットの入力ポートA，Bと1ビットの出力ポートYを持つ．記憶素子（フリップフロップなど）を含まない組み合わせ回路である．

KeyWord　テストベンチ，テスト入力，絶対時間，相対時間，fork，wait，after，assert文，エンコーダ，for文

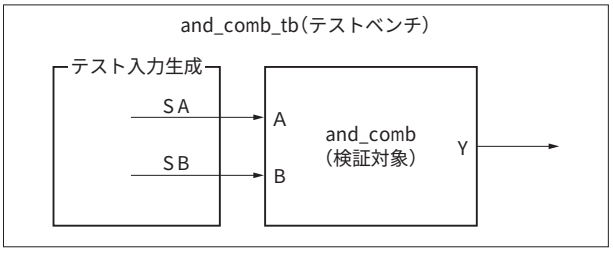
Verilog HDL

リスト1(a)は，第1回で示したテスト入力の記述例です．ここで記述された信号SA，SBの変化のタイミングは**図3(b)**と同じです[注1]．

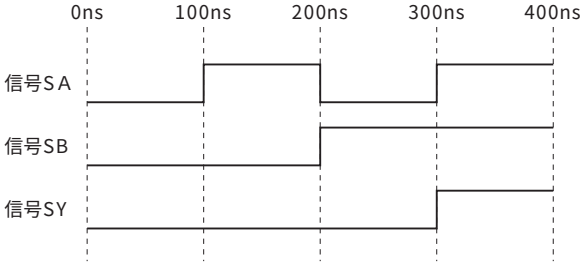
リスト1(b)は，**リスト1(a)**の記述を1行当たり一つの式だけの記述に直したものです．式の実行タイミングは，式の間の相対遅延で記述されています．**図4**のように，begin～endの間に記述された式は上から順番に実行され，その間の遅延は累積されます(式2は必ず式1の後に実行され，式3は必ず式2の後に実行される)．

図5はfork～joinという文法の書式を表しています．fork～joinは，begin～endと違い，式の間で遅延は累積しません．それぞれの式がシミュレーション開始時から独自の遅延値に従って並行に実行されます．つまり，各式の遅延値は絶対遅延となります．

リスト2は，**リスト1**とまったく同じ変化タイミングで信号SAとSBのテスト入力を記述したものです．各代入式の#に続く遅延値は絶対遅延となり，式の間で累積しません．



(a) ブロック図



(b) タイミング・チャート

図3　AND回路の検証
(a)のようなテストベンチから，(b)のテスト入力を与える．

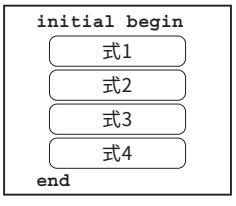


図4
Verilog HDLによる相対遅延の書式
begin～endの間の式は，上から順に実行され，遅延が累積する．

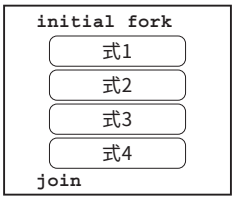


図5
Verilog HDLによる絶対遅延の書式
fork～joinの間の式は，シミュレーション開始時点からの独自の遅延値に従って並行して実行され，式の間で遅延は累積しない．

VHDL

リスト3(a)は，第1回で示したテスト入力の記述例です．ここで記述された信号SA，SBの変化のタイミングは，**図3**と同じです．

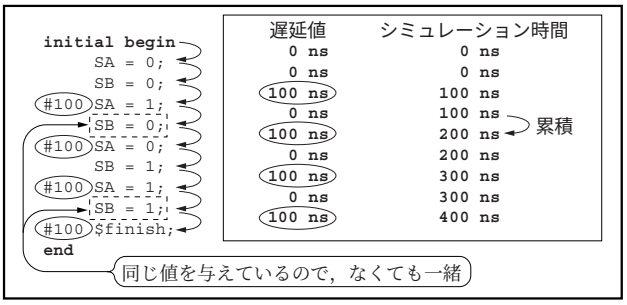
リスト3(b)は，**リスト3(a)**の記述を1行当たり一つの式だけの記述に直したものです．式の実行タイミングは，wait文ごとに累積する相対遅延で記述されています．**図6**のように，wait文を使った記述では，式2は式1実行後，間のwait文で指定した遅延経過後に実行され，式3は式2実行

注1　本稿のVerilog HDLのすべての解説では，シミュレータのシミュレーション時間の単位が，nsに設定されているものとして解説しています．

リスト1　Verilog HDLによるAND回路のテストベンチ

```
initial begin
      SA = 0; SB = 0;
#100 SA = 1; SB = 0;
#100 SA = 0; SB = 1;
#100 SA = 1; SB = 1;
#100 $finish;
end
```

(a) 記述例1



(b) 記述例2

リスト2　Verilog HDLによる絶対遅延を使ったテストベンチ