汎用テストベンチ作成ツール
を用いた
シミュレーションからテストまで

京都工芸繊維大学
小林和淑

概要と内容

Perlを用いた自作EDAツールSTの紹介

- 単一の記述からさまざまなシミュレータ、テスタへのテストベンチが生成可能

内容

- 既存の環境とSTによる環境
- STを使った汎用テストベンチ記述
- Verilog, Spiceでの使用例
- LSIテスタでの使用例
- 結論、今後の課題

ST is a solution for sim & test

- Now, you have to write various test benches for all simulators:
  - HDL, Netlist level
  - Circuit, SPICE level
- ST is a solution!!
  - You can write your test bench with PERL!!
  - Your testbench can be converted to
    - » Verilog
    - » SPICE
    - » LSI testers

LSI設計時のシミュレーション

- LSI設計
  - 様々なレベルでのシミュレーションが必須
- ネットリストはツールから自動的に.
  - HDL, ネットリスト, 回路レベル...
- But テストベンチは自分で.
  - ツール、シミュレータ毎に書き方を覚えないと駄目

ST

単一の記述から様々なシミュレータのテストベンチに変換

既存のテストベンチ記述, テスタインタフェース環境

- テストベンチ環境
  - Cadenceのstl(simulation & test language)
  - 単一の記述から変換可
- LSIテスタへのベクタ変換
  - TSSI社のツール
  - stlと接続.
  - サポート費用高額(300万/year)
  - もう一つ使いやすくない

商用変換ツールの問題点

- 大学:　設計者＝＝テスト屋さん
- シミュレーション結果からの変換は不便
  - 双方向ピン 入力?出力?
  - 不定値≠ドントケア(x)
  - ピンの設定が必要
- シミュレーションしないとテスト不可
  - Verilogでシミュレーションできないものはテストできない.
- ちょっとしたテストをするには不便.

ST: Perl Package for Sim & Test

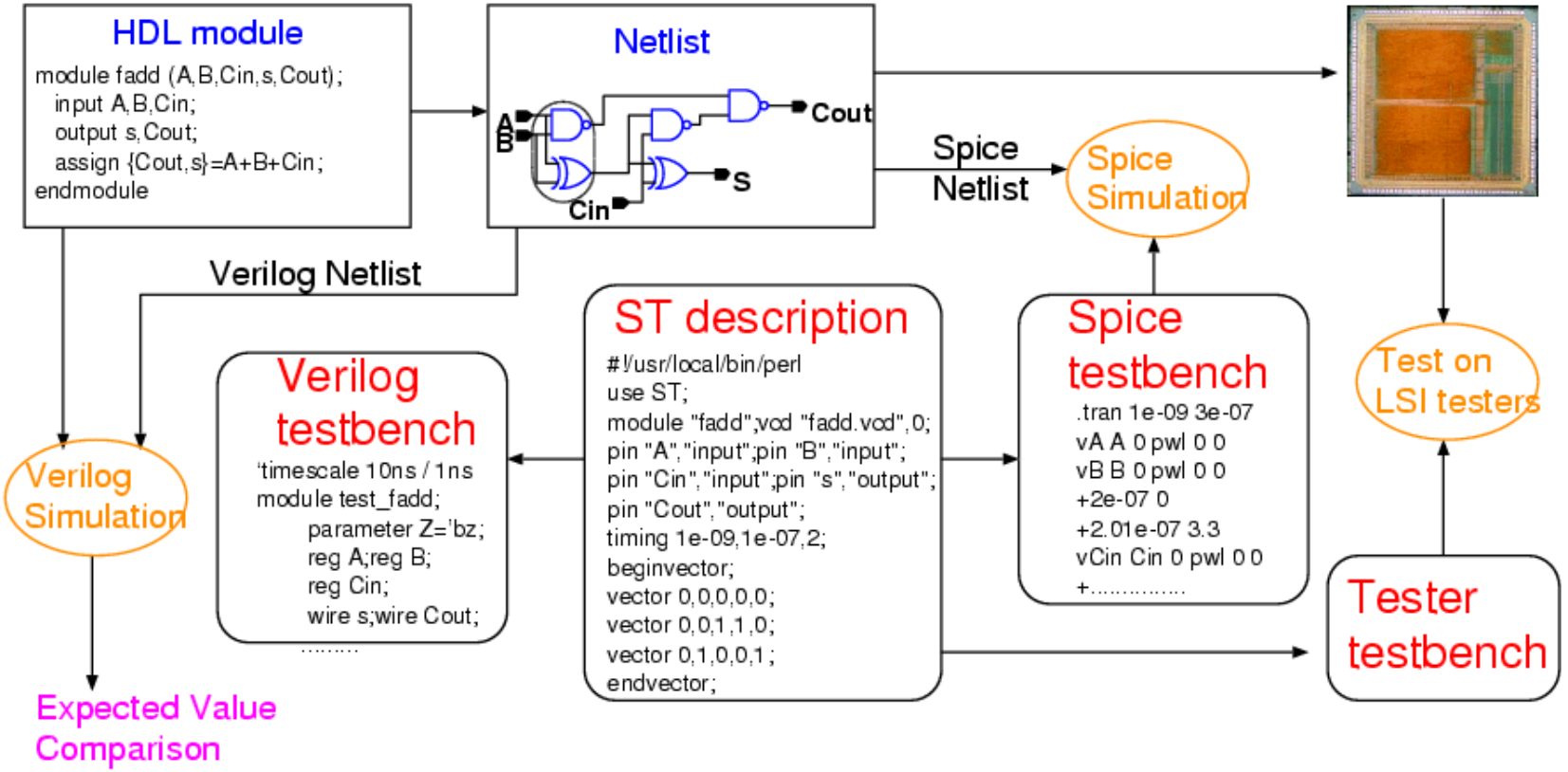
- 設計者が使いやすいテストベクタ記述
- Perlのpackage
- 単一の記述からシミュレータ、テスタへ
  - シミュレータ: Verilog, Spice, VHDL等
  - テスタ: HP83000, hilevel griffin, MMS MU300-EM
  - 電源電圧、クロック周期、ベクタが簡単に変更可
- 対象: サイクルベースのディジタル回路
  - 1サイクル毎の波形を指定
- Verilog, hspice: 期待値比較

Why Perl??

最初の案
- 独自言語仕様を、Perlで処理

Perlそのもので書く
- テストベンチ作成用のサブルーチンをパッケージ化
- 構文解析が非常に簡単
- CPUが高速化
  - インタプリタ型言語でも遅くならない

STを使った検証フロー



設計の詳細度にしたがって、レベルを変更可能

記述例: 4ビット累算回路

```
#!/usr/local/bin/perl
use ST;
target "verilog";
module "fourbitaccum";
vcd "fourbitaccum.vcd",0;
pin "in[3:0]","input";
pin "CLK","clock";
pin "RST","input";
pin "out[7:0]","output";
timing 1e-09,1e-07,10;
clock "CLK","1111100000";
waveform "input","dnrz","%.....","in","RST";
waveform "output","edge",".....%","out";
pinorder "in","RST","out";
beginvector;
vector 0,0,0;
vector 1,1,0;
vector 5,1,1;
endvector;
```

シミュレーション対象の定義

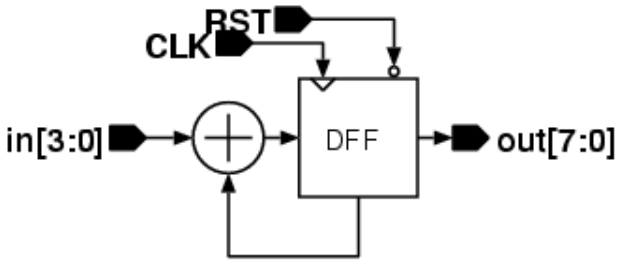
Verilog定義

ピンの定義

タイミングの定義

波形の定義

入力ベクタ、期待値の定義



fourbitaccum

使い方: 設定関係コマンド

target ターゲット名;	シミュレータ, テスタを指定
pin ピン名, ピンの種類, オプション;	ピンの定義
timing 最小時間単位, クロック周期, タイミング分割数;	最小時間単位, 1 サイクルの長さ等を指定
clock クロック信号名, 波形;	クロック波形を定義
waveform ピン方向, 波形タイプ, 波形, ピン名	1 サイクル内の入力波形, 出力比較ポイントを指定
pinorder ピン 1, ピン 2, ピン 3,,,;	vector コマンドの波形順

使い方：ベクタコマンド

beginvector, endvector;	ベクタの開始と終了
vector 0,1,“X”...;	すべてのピンの波形, 期待値を指定
svector “pin=val”,...;	指定したもののみ変化, 指定しないものはデフォルト値
stimulus “pin=val”,...;	指定したもののみ変化, 指定しないものは前の値のまま

使い方と期待値比較: Verilogインタフェース

vcd “ファイル名”, レベル,...;	vcd 波形ファイルの指定
module モジュール名;	テストするモジュール名を指定

実行結果

期待値との照合結果

Logファイル（VCS）

```
Compiler version 5.2; Runtime version 5.2;  Sep 22 09:36 2000

Some simulation mismatches are detected!

$finish at simulation time                400

           V C S   S i m u l a t i o n   R e p o r t

Time: 400 ns
```

st.log 期待値との相違

```
 cycle(time): pin: exp. val != sim. result
        2(29): out:    2 !=1
        3(39): out:    5 !=6
```

Verilogでの期待値比較

STから出力されたテストベンチ

```
//cycle 3
        in<=0;
        #9
        if(out == 5) ; else begin
                $fdisplay(_f,"        3(%0t): out:     5!=%0d",$realtime,out);
                _error=1;
        end
```

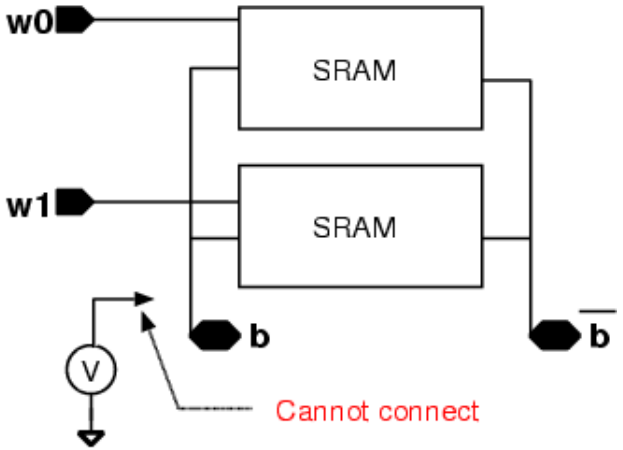
期待値

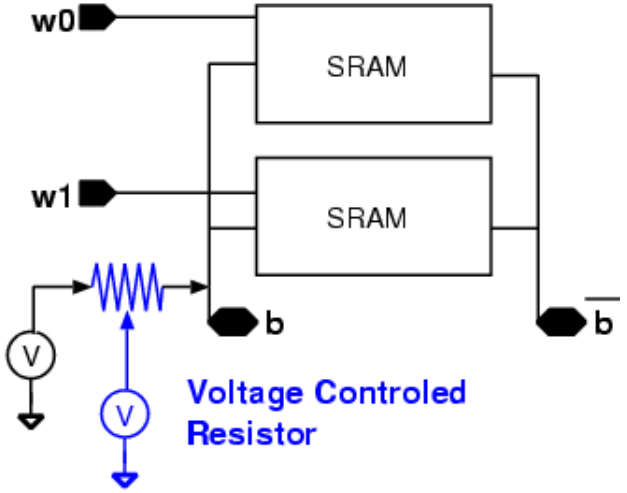
- Verilogそのものの機能により期待値比較
  - PLIを使わなくてよい!
  - シミュレーションだけで結果が判明
- 動作確認シミュレータ
  - Verilog-XL, Ncverilog, VCS等

使い方: Spiceインタフェース

level ローレベル, ハイ レベル;	電源電圧の指定
probe ピン名;	波形出力の指定 (HSPICE 用)
source 電源名;	電源の指定

- SPICEの電圧制御抵抗(G Element)による双方向端子サポート





使い方: LSIテスタインタフェース

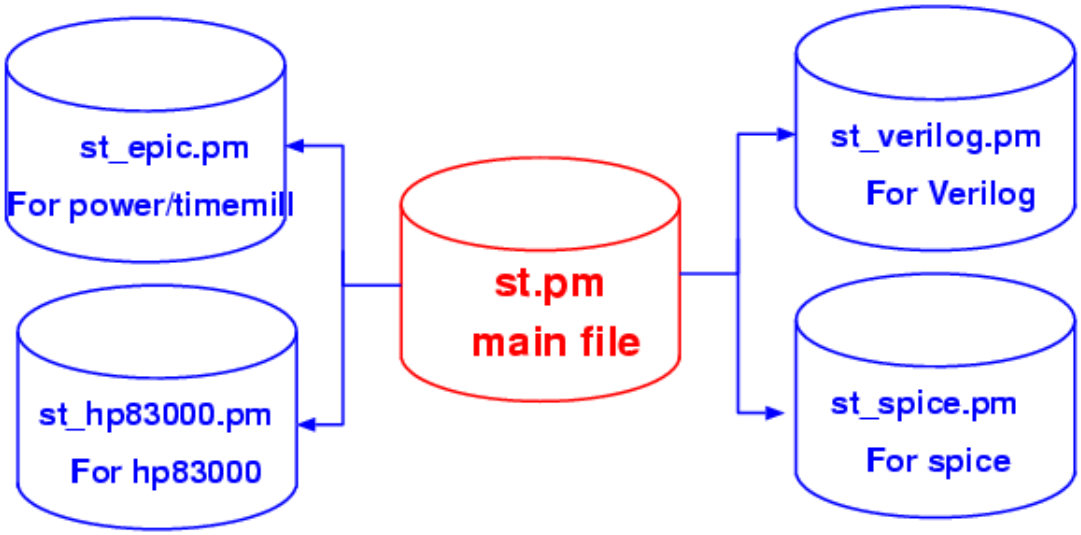
source_vdec ;	VDEC DUT ボードの電源設定
pin “ピン名”,“loc=[1,2,3:5]”;	チップのピン位置指定
channel “ピン”, “チャネル”;	ピン番号とテスタチャネル番号の対応
channel_file ファイル名;	csv より入力
level v_inl,v_inh,v_outl,v_outh;	電源電圧, 比較電圧
trigger;	EB テスタ用トリガの挿入

- テスタに必要なすべてのファイルを出力
- シミュレーションなしでテストベンチ作成

既存のテストベンチより

- 既存のテストベンチからSTのファイルへ
- v2st.pl:
  - verilogのmoduleからstのテンプレート生成
- ターゲット　verilogcomp:
  - 既存のテストベンチより、vectorコマンドを抽出するためのVerilog記述を出力
  - テスタの入力ベクタ変換、spiceでの最終確認

ST内部構造



- module構造
  - 各ターゲット毎のファイルに分離
  - st_シミュレータ名.pmにより対応
  - APIを理解すれば、誰でも書けるはず??

結論

- 汎用テストベンチ記述用PerlパッケージSTを開発
- STを使えば、単一の記述から複数のシミュレータ、テスタのテストベンチが生成可
- Verilog, hspiceでは期待値比較可能
  - 外部プログラム等は一切不要. シミュレータの持つ機能のみで実現
- SPICEでは、双方向端子に対応

STのダウンロード

- GPLで公開しています.
- 何かフィードバックがあればお気軽に

配布先

http://www-vlsi.es.kit.ac.jp/~kobayasi/ST/

今後の課題

- アナログへの対応
  - どうしたら良いでしょうねえ
- マニュアルの整備
  - 英語でのマニュアル記述
  - 古いバージョンはあり
  - 今後は英語だけにするかも….
- 新しいシミュレータ、テスタへの対応