Everywhere you imagine. RENESAS

SDF2007
検証言語としてのSystemVerilog
適用事例

株式会社ルネサス テクノロジ
設計技術統括部　システム設計技術開発部
鎌田　丈良夫

2007/1/26　SDF2007 検証言語としてのSystemVerilog適用事例

©2006. Renesas Technology Corp., All rights reserved.

Agenda

- はじめに
- SystemVerilogによる検証環境の構成
- 適用結果
- まとめ

©2007. Renesas Technology Corp., All rights reserved.

Everywhere you imagine. RENESAS

はじめに

- ルネサスは2005年8月よりSystemVerilog導入を開始

1. SVAチェッカ、機能カバレッジ測定
   - チェッカ作成容易化の為、PSLから移行中
   - 検証進捗状況把握の自動化

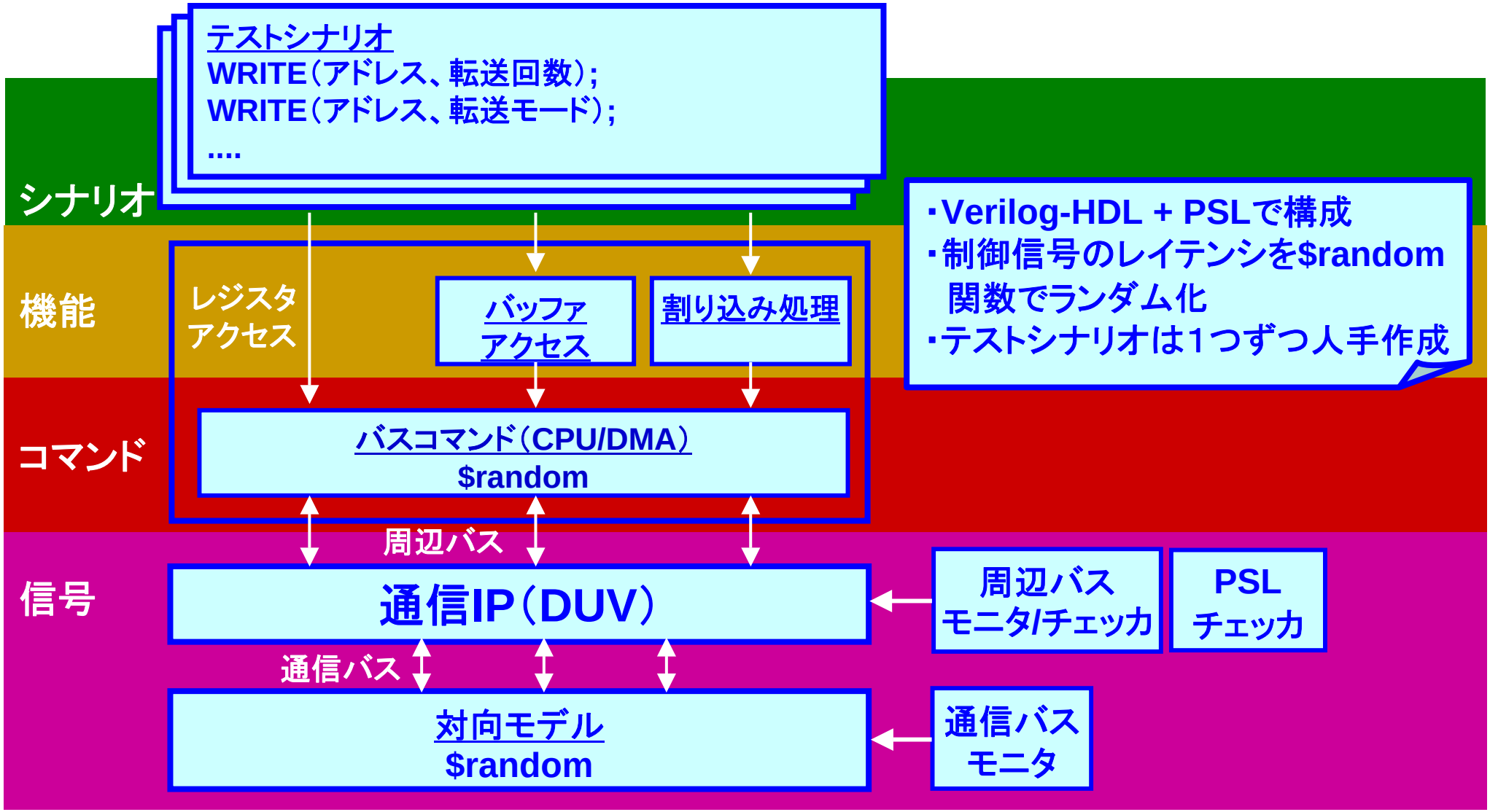
2. 社内共通バスチェッカ
   - チェッカ・カバレッジ測定の高機能化
   - 保守容易化

3. IP単体検証環境
   - テストシナリオ作成効率向上

本発表では3. に関して紹介

©2007. Renesas Technology Corp., All rights reserved.

従来のIP検証環境の構成例



©2007. Renesas Technology Corp., All rights reserved.
Everywhere you imagine. RENESAS

IP検証環境の要件と問題点

- 要件
  - 標準的な言語を利用して、シミュレータ間の移植性を確保
  - 検証環境の構築が容易
  - 検証環境のメンテナンス・展開が容易

- 従来環境での問題点
  - ツール非依存かつ柔軟な制約付きランダム生成が困難
  - テストシナリオはパラメータの組み合わせの数だけ人手作成が必要
  - テストシナリオのバリエーション作成工数大

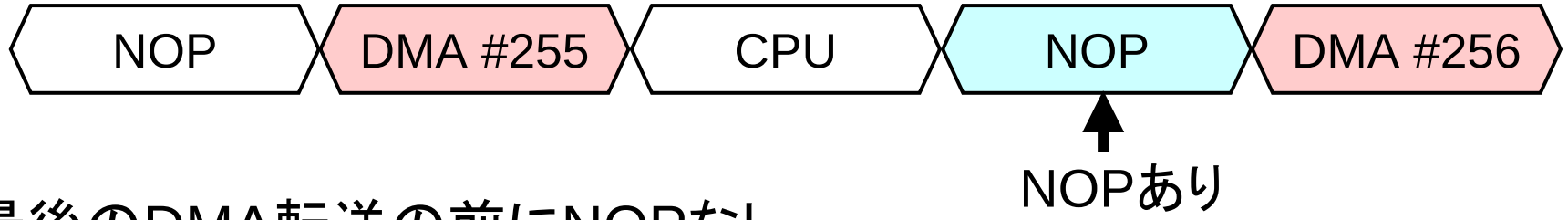
上記要件を満たしつつ、問題点を解決したい

©2007. Renesas Technology Corp., All rights reserved.

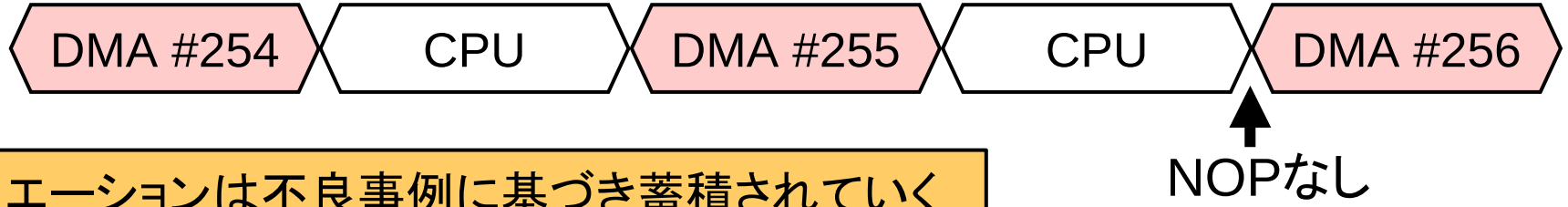
テストシナリオのバリエーション

- DMA転送のテストシナリオバリエーション例
  サイクルスチールモード、256回のDMAアクセス実行時
  最後のDMAアクセスの前にNOP挿入あり／なし

最後のDMA転送の前にNOPあり



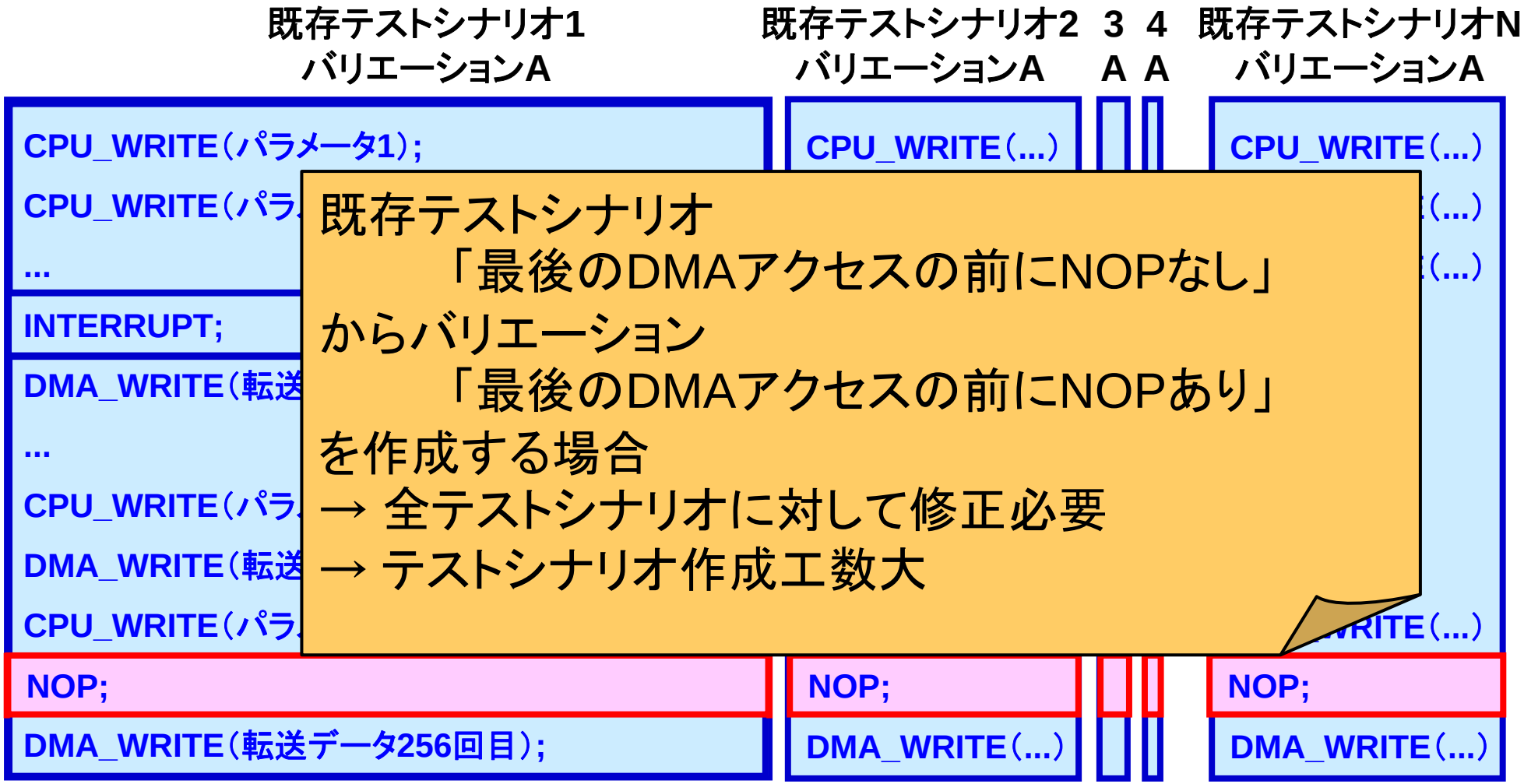
最後のDMA転送の前にNOPなし



バリエーションは不良事例に基づき蓄積されていく
検証上の資産として、関連開発に活用したい

©2007. Renesas Technology Corp., All rights reserved.

既存テストシナリオからのバリエーション作成（従来）



©2007. Renesas Technology Corp., All rights reserved.
Everywhere you imagine. RENESAS

既存テストシナリオからのバリエーション作成の効率化

- 既存テストシナリオの一部を修正してテストシナリオのバリエーションを作成する場合、全テストシナリオを修正せずに済ませたい

- テストシナリオのバリエーション作成に関する既知ノウハウ増加に伴い
  → ノウハウ収集／テストシナリオ作成工数も増加していく
  → テストシナリオのバリエーション作成に関する既知ノウハウを効率よく活用したい

©2007. Renesas Technology Corp., All rights reserved.

Agenda

- はじめに
- SystemVerilogによる検証環境の構成
- 適用結果
- まとめ

©2007. Renesas Technology Corp., All rights reserved.

検証言語としてのSystemVerilog採用理由

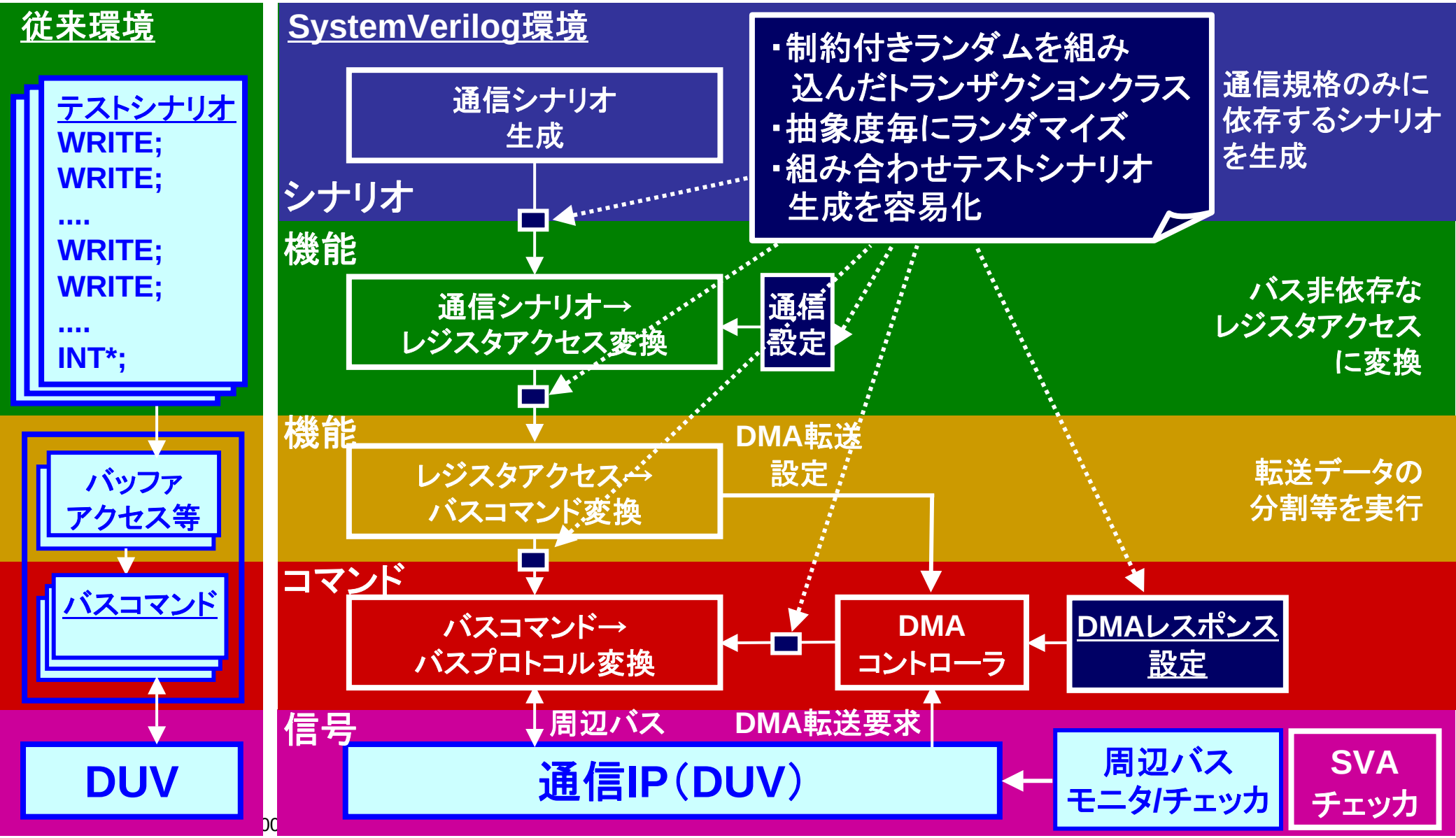
- 他の検証言語やHDLと比較して、下記のメリットがある
  - 複数のシミュレータ間で移植が容易
  - Verilog-HDL習得者にとって学習が容易
  - テストベンチのデバッグが容易
  - 組み込みの制約付きランダム生成をサポート
  - オブジェクト指向言語により再利用性や拡張性が向上
  - 機能カバレッジ、高機能なアサーション言語、等々

以上を総合的に判断して、SystemVerilogでの環境構築を採用した。

©2007. Renesas Technology Corp., All rights reserved.

SystemVerilogによる検証環境の構成

■：制約付きランダム生成を組み込んだトランザクションクラス



©2007. Renesas Technology Corp., All rights reserved.

Everywhere you imagine. RENESAS

制約付きランダム生成

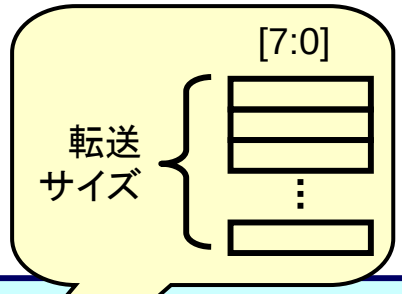
- 制約付きランダム生成を適用したパラメータ

ランダマイズ対象クラス	ランダマイズパラメータ（パラメータ数）	
	従来	新規追加分
通信シナリオ	なし	転送モード（2）、転送回数（1～N）、転送サイズ（1～N）、リード/ライト（2）アドレス、データ
通信設定	なし	通信バス幅（3）、CPU/DMA転送（2）、通信クロック（1～N）、転送コマンド（N）等
バスコマンド	NOPサイクル数（1～N）	周辺バス幅（3）
DMAレスポンス設定	DMA要求受信-実行レイテンシ等	追加なし

- SystemVerilogの制約付きランダムにより柔軟な乱数重み付けが容易に変更可能になった

©2007. Renesas Technology Corp., All rights reserved.

制約付きランダム生成



SystemVerilogによる実装

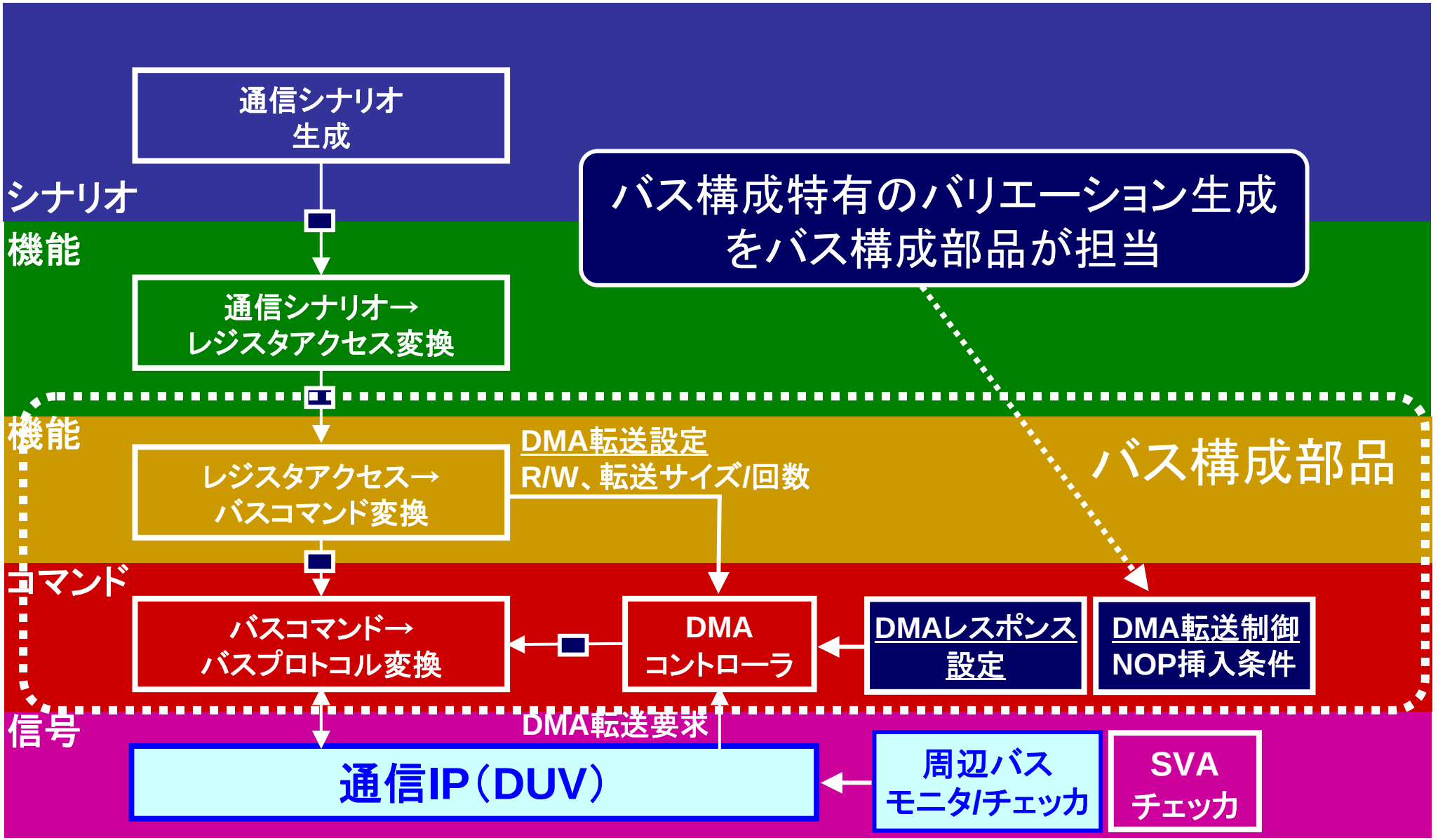
```
class communication_transaction;
   rand operation_e                operation;          リード、ライト
   rand transfer_mode_e            transfer_mode;      // 転送モードA、転送モードB
   rand int unsigned               transfer_size       // 転送サイズ
   rand int unsigned               transfer_count;     // 転送回数
   rand bit [31:0]                 address;            // アドレス
   rand bit [7:0]                  data[];             // データ

constraint transfer_count_constraint {
if(transfer_mode == モードA)      transfer_count == モードA転送回数(固定値);
else if(transfer_mode == モードB) transfer_count >= 1 && transfer_count <= 最大転送回数; }

constraint transfer_size_constraint {
if (transfer_mode == モードA)      transfer_size >= 1 && transfer_size <= 最大転送サイズ;
else if (transfer_mode == モードB) transfer_size == 最大転送サイズ;  }
```

相関する制約を宣言的に記述できる → 設定が容易

```
constraint byte_data_size_constraint { data.size() == transfer_size; }
endclass
```


©2007. Renesas Technology Corp., All rights reserved.

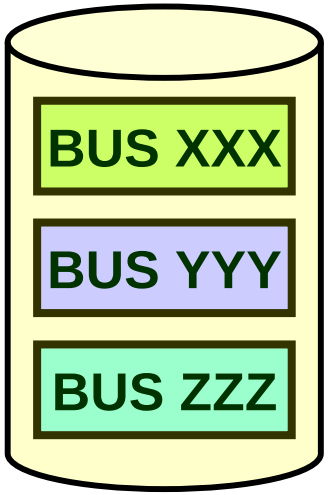
既存テストシナリオからのバリエーション作成（DMA転送）

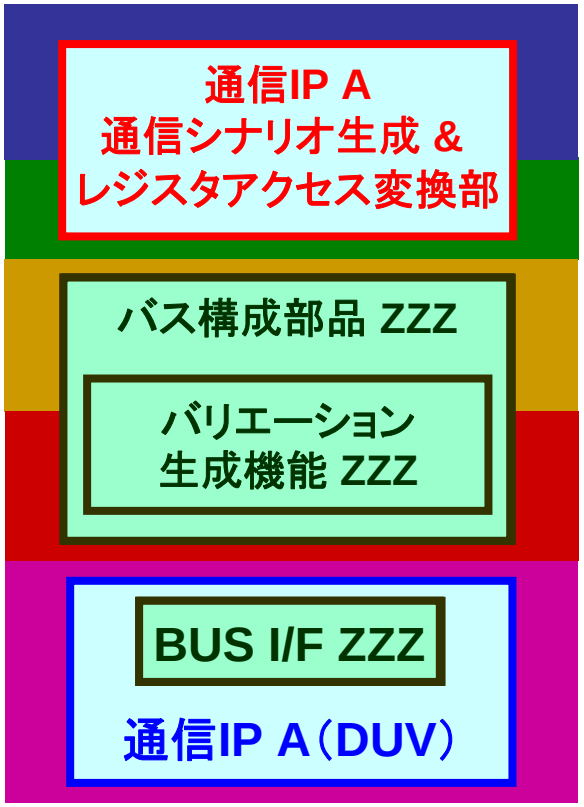


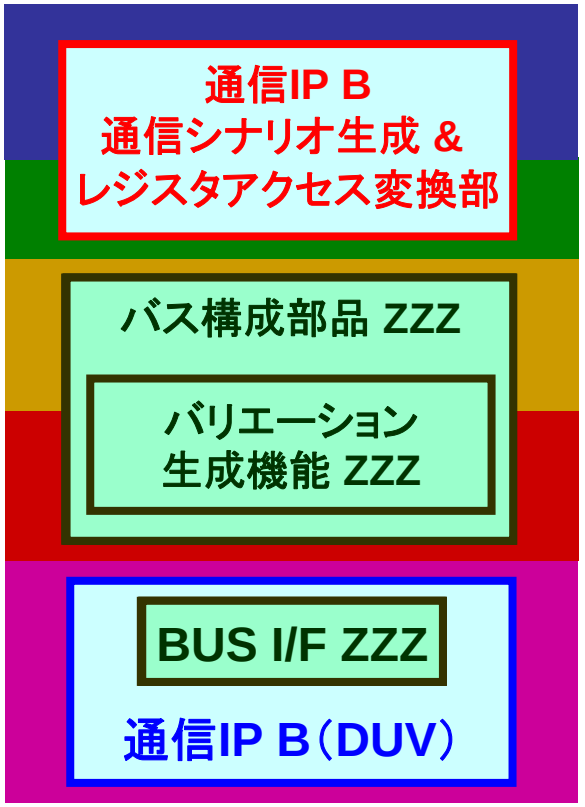
©2007. Renesas Technology Corp., All rights reserved.
Everywhere you imagine. RENESAS

テストベンチ部品の再利用

バス構成部品







DUVのバスI/Fのみ変更、IP機能は共通
→ テストシナリオは変更不要

DUVの機能を変更
→ バスに依存しないテストシナリオを作成

バス構成特有のバリエーション生成はバス構成部品が担当

©2007. Renesas Technology Corp., All rights reserved.

Agenda

- はじめに
- SystemVerilogによる検証環境の構成
- 適用結果
- まとめ

©2007. Renesas Technology Corp., All rights reserved.

適用結果

- テストベンチ部品を抽象度ごとに構築することで、テストシナリオの組合せ生成容易化、制約付きランダム生成の多様化、テストベンチ部品の再利用容易化を実現した
  - テストシナリオの組合せ生成容易化により、テストシナリオ作成工数を大幅に削減可能とした
  - 制約付きランダム生成により、ランダム化可能なパラメータを1→12に多様化し、かつ重み付けを容易に変更可能とした
  - テストベンチ部品の再利用容易化により、既存テストシナリオからのバリエーション作成効率を向上させ、かつ今後の環境構築工数を削減可能とした

- 上記の機能増加・再利用性向上にも関わらず、テストベンチ部品のコード量は従来の検証環境と比べてほぼ同量に収まった

©2007. Renesas Technology Corp., All rights reserved.

構築・適用を振り返って

- 学習期間と構築工数
  - SystemVerilog講座を2日間受講
  - 約1人月で構築

- 苦労したこと
  - SystemVerilogの構文エラーデバッグ
    - ・シミュレータ未サポートの構文・記述スタイル or 記述ミスの切り分け
    - ・シミュレータのLint機能が弱い

©2007. Renesas Technology Corp., All rights reserved.

Agenda

- はじめに
- SystemVerilogによる検証環境の構成
- 適用結果
- まとめ

©2007. Renesas Technology Corp., All rights reserved.

まとめ

- SystemVerilogを用いて通信IPの単体検証環境を構築した

- テストベンチ部品を抽象度ごとに構築することで
  - テストシナリオ作成工数を大幅に削減可能とした
  - ランダム化可能なパラメータを1→12に多様化し、かつ重み付けを容易に変更可能とした
  - 既存テストシナリオからのバリエーション作成効率を向上させ、かつ今後の環境構築工数を削減可能とした

を実現した

©2007. Renesas Technology Corp., All rights reserved.

今後の課題

- SystemVerilog検証環境のエンハンス
  - 目視チェック削減のためのスコアボードによる期待値比較
  - 検証進捗状況把握のための機能カバレッジ集計

- 開発環境に関する要望
  - シミュレータのLint機能強化
  - シミュレータのSystemVerilog言語サポート範囲拡大
  - 業界標準SystemVerilogライブラリの早期確立

©2007. Renesas Technology Corp., All rights reserved.

株式会社ルネサス テクノロジ

©2006. Renesas Technology Corp., All rights reserved.