Software Driven Verification

テストプログラムはC言語で! SystemVerilog DPI-Cを使えば、こんなに便利に!

2011年9月30日

富士ゼロックス株式会社
コントローラ開発本部コントローラプラットフォーム第五開発部　宮下晴信

FUJI xerox

この資料で使用するシステム名、製品名等は一般にメーカーや
団体の登録商標などになっているものもあります

なお、この資料の中では、トレードマーク、コピーライト等の表示は
明記しておりません

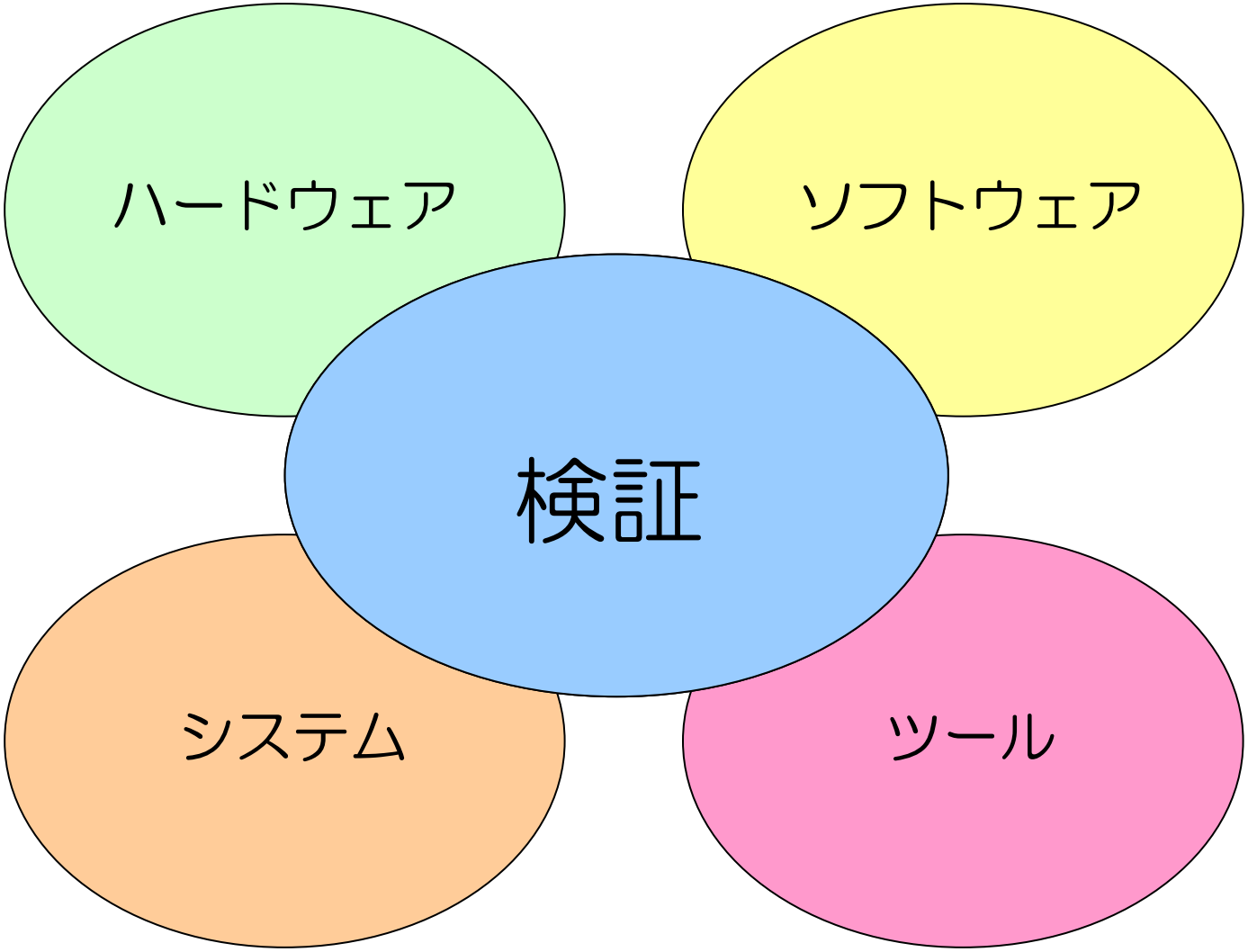
FUJI xerox

目次

FUJI xerox

検証という仕事

検証という仕事



仕事の詳細は？

ハードウェア

モデル作成 (モデリング)

ソフトウェア

シナリオ作成 (テストプログラム)

プログラム　(C、C++、Java、HVL、HDL)

スクリプト　(簡易言語、汎用言語 : Perl/Ruby/Python)

システム

検証仕様書

検証項目リスト作成、テストベンチ作成、データ/期待値作成

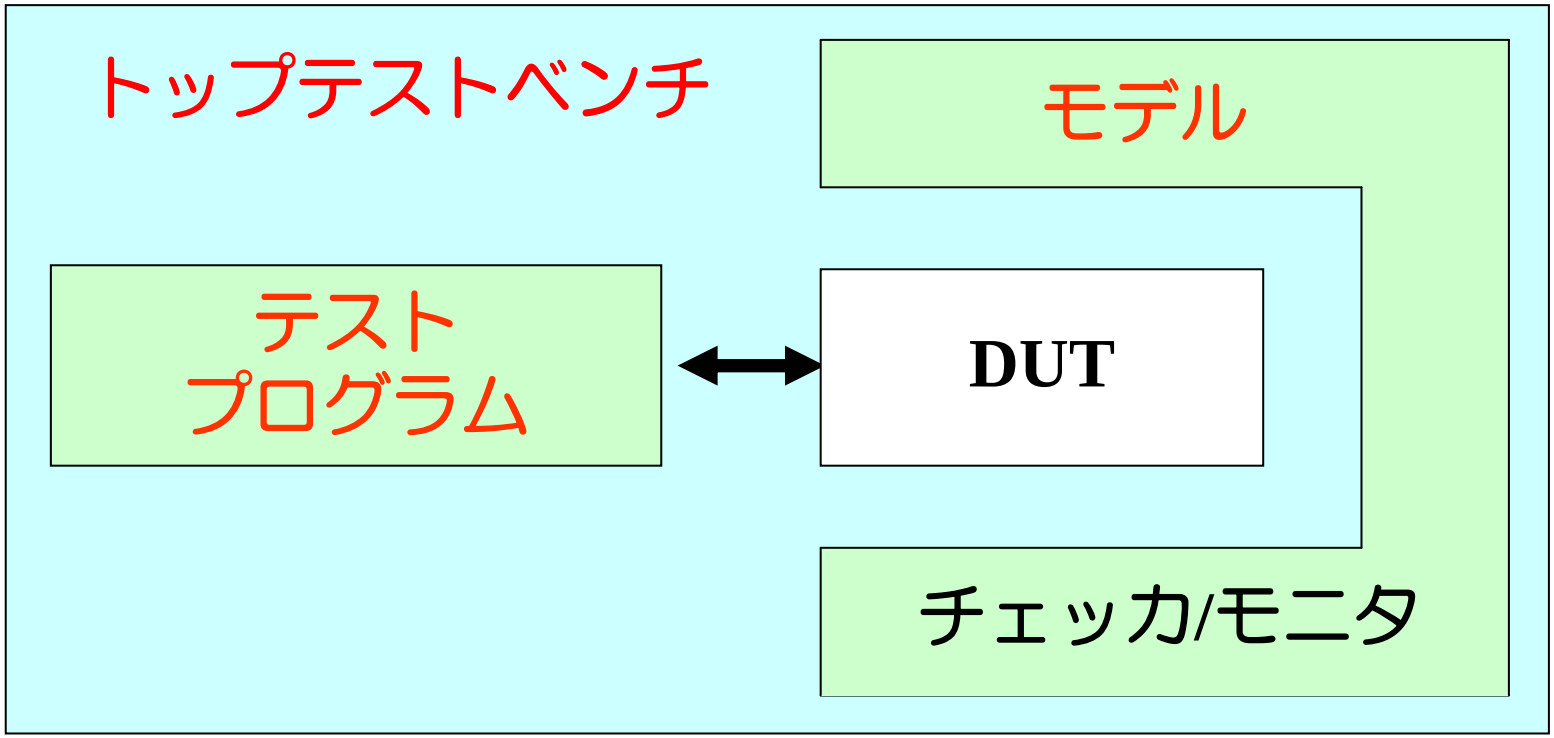
ツール

ツール作成

自動化(期待値生成、シミュレーション、期待値チェック)

FUJI xerox

3つの要素



言語：1990年代

	設計		検証	
	実装	アサーション	モデリング	ﾃｽﾄﾌﾟﾛｸﾞﾗﾑ
HDL 前半 ‘90 – ‘94	VHDL Verilog-HDL	VHDL Verilog-HDL	VHDL Verilog-HDL	VHDL Verilog-HDL
HVL 後半 ‘95 – ‘99	VHDL Verilog-HDL	e Vera	e Vera	e Vera

FUJI xerox

言語：2000年代

	設計		検証	
	実装	アサーション	モデリング	ﾃｽﾄﾌﾟﾛｸﾞﾗﾑ
C 前半 ‘00 – ‘04	SystemC ＋ VHDL Verilog-HDL	PSL e OVA	SystemC ＋ e OpenVera	SystemC ＋ e OpenVera
SV 後半 ‘05 – ‘09	SystemC ＋ SystemVerilog	SVA	SystemC ＋ SystemVerilog	SystemC ＋ SystemVerilog DPI-C

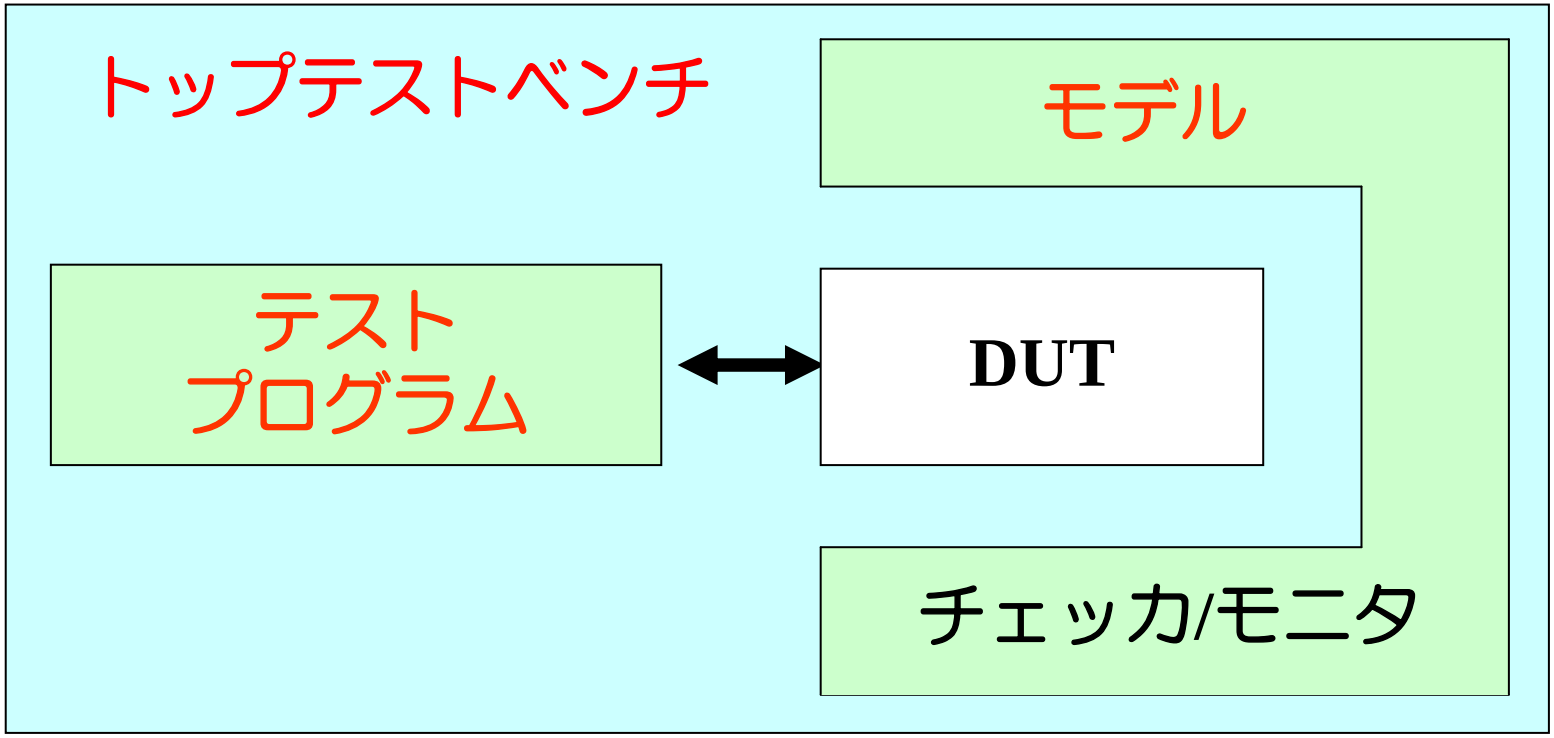
FUJI xerox

Software Driven Verification

Software Driven Verification

DUTにソフトウェア(テストプログラム)を使って、検証する。

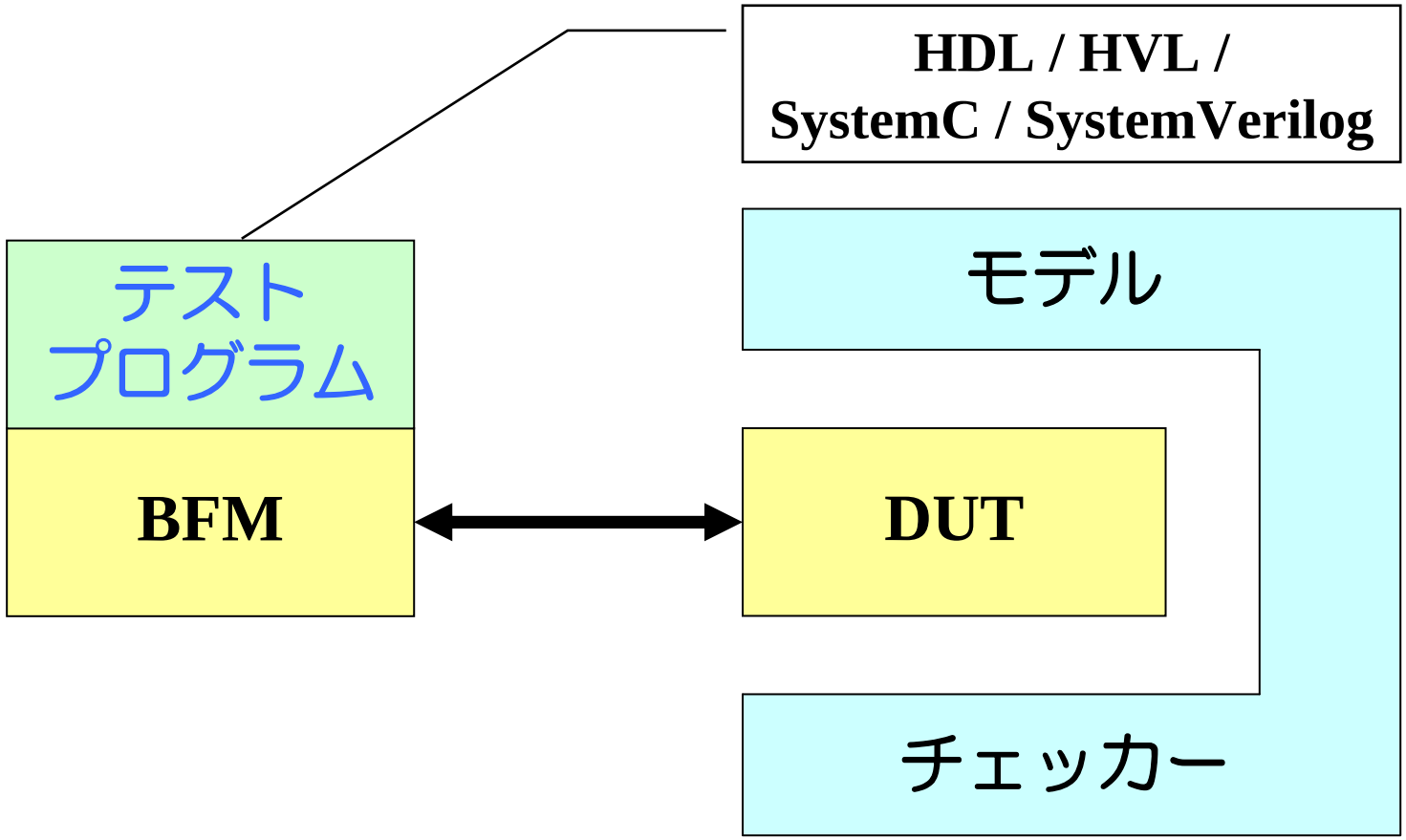
そこで、テストプログラムを何で書いている？



FUJI xerox

BFMによるテストプログラムの実行

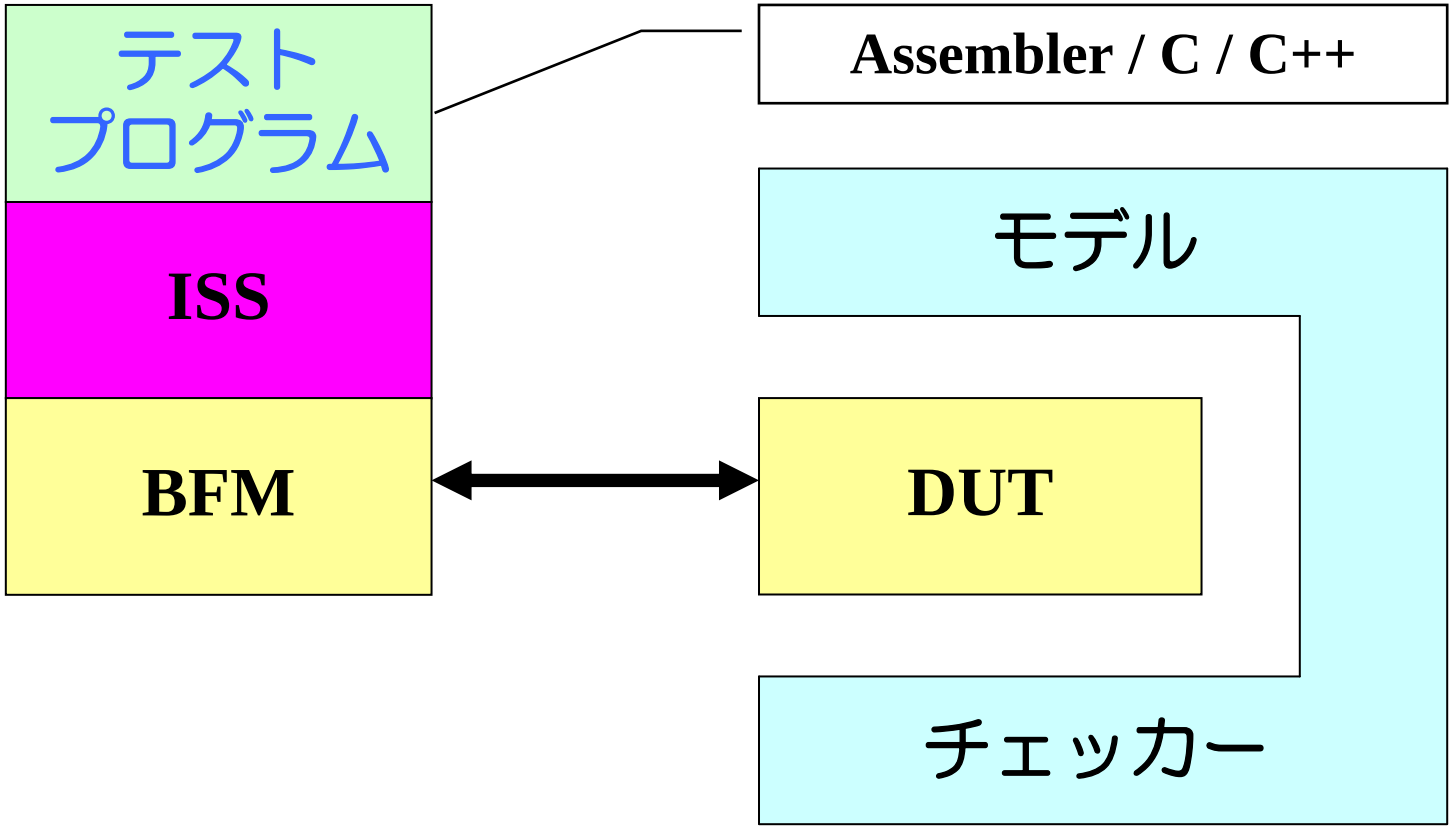
HDL / HVLが利用できる
Logic Simulatorのみで実行可能である



FUJI xerox

ISSによるテストプログラムの実行

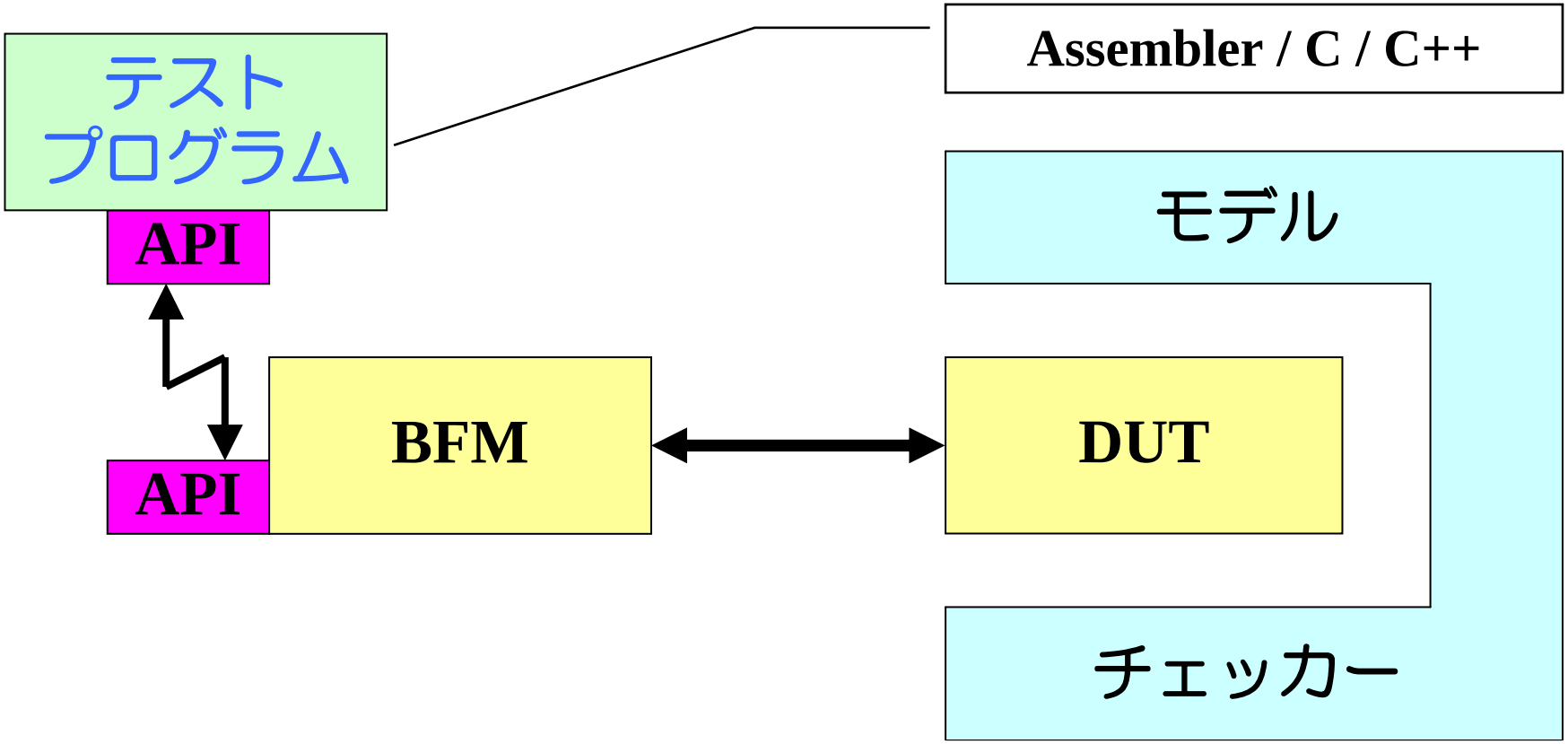
実機と同じプログラムが利用できる
ISSが重いと、全体のシミュレーション速度が低下する



FUJI xerox

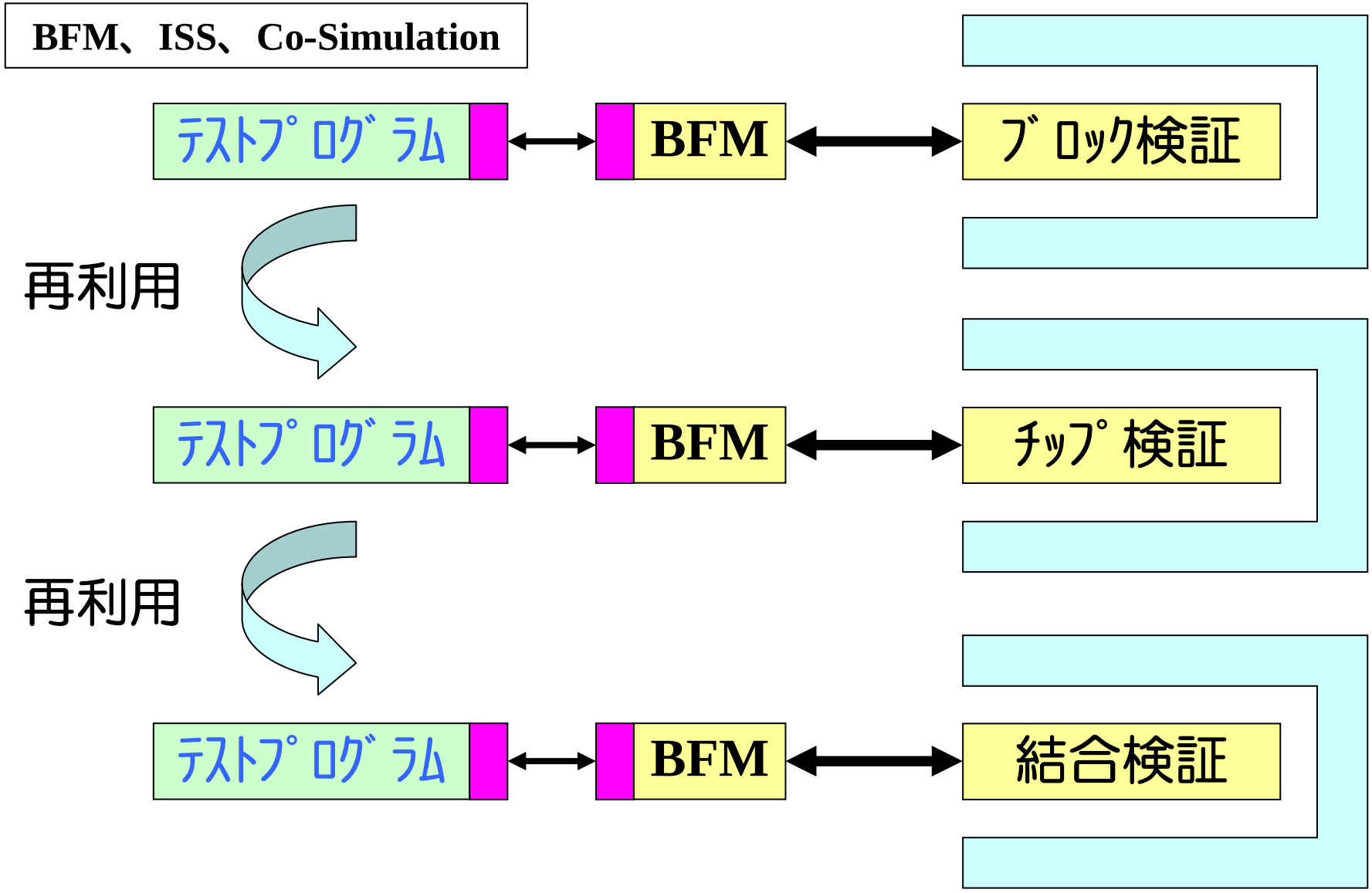
Co-Simulationによるテストプログラムの実行

C/C++などのソフトウェアでの言語が利用できる
テストプログラムを別のCPUで実行すれば、
シミュレーション速度の低下は少ない



FUJI xerox

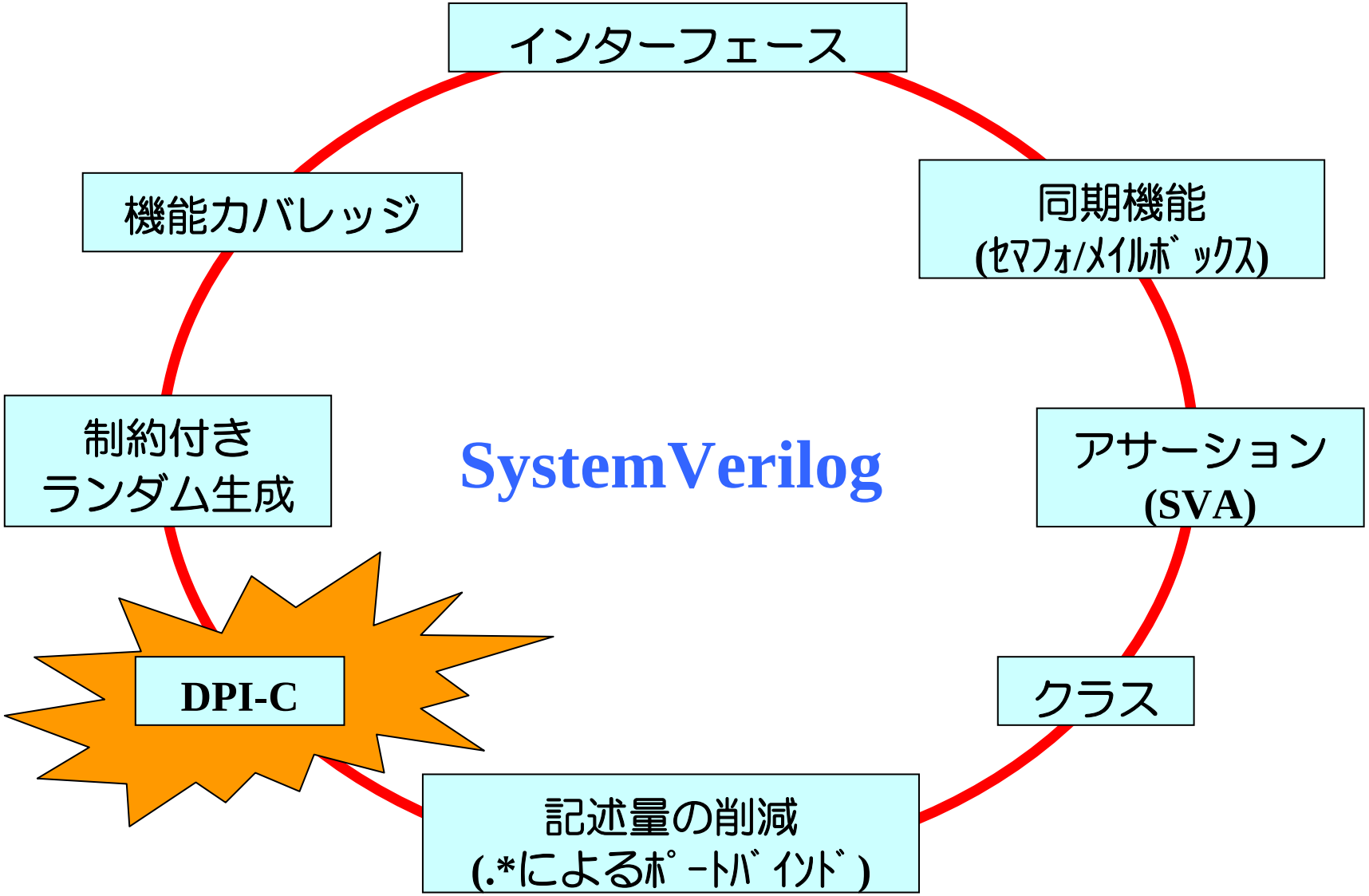
テストプログラムの再利用



FUJI xerox

SystemVerilog DPI-Cを利用してみる

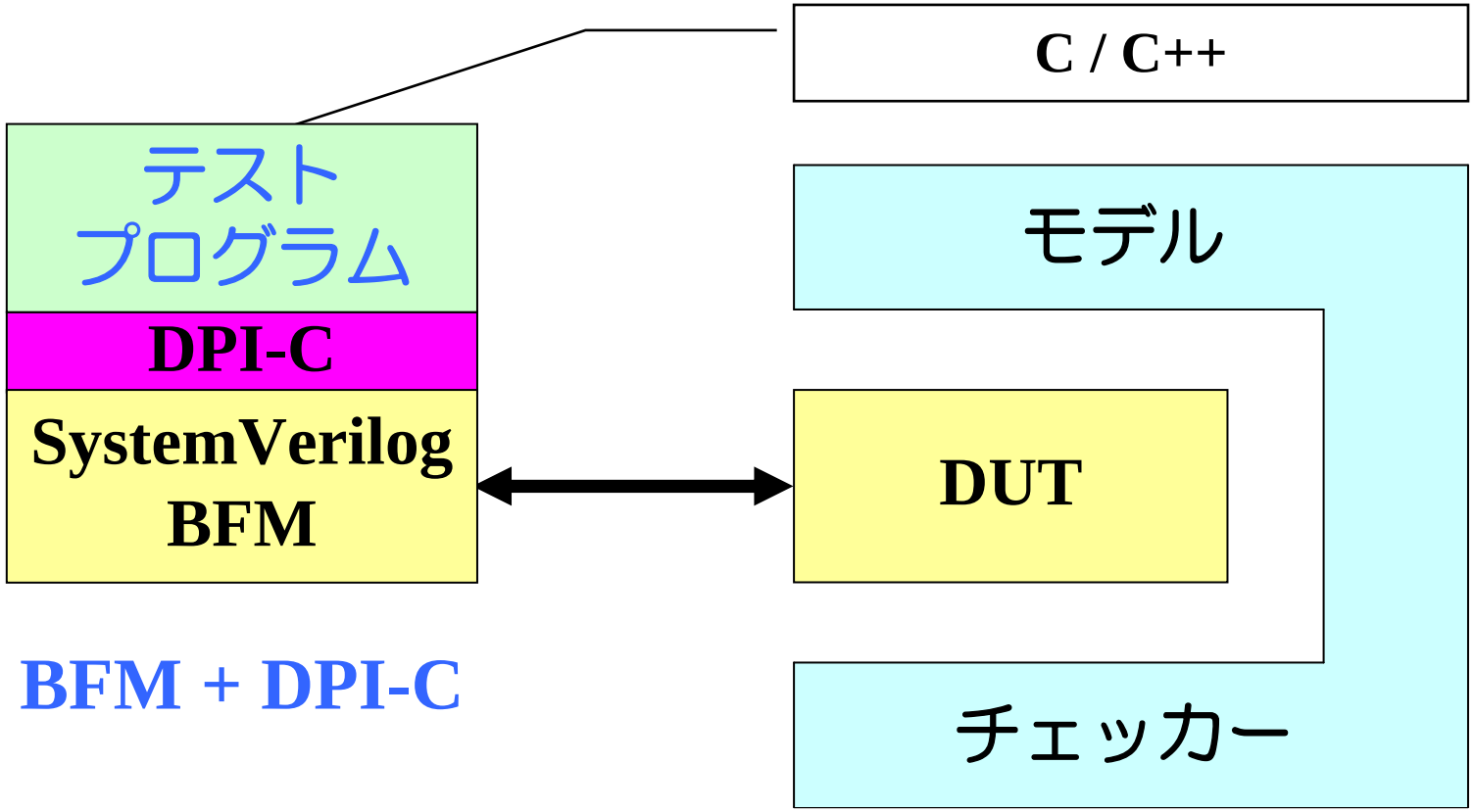
SystemVerilogのいろいろな機能



FUJI xerox

SystemVerilog DPI-Cを利用すると

基本的には、
Co-Simulationによるテストプログラムの実行と同じ
特別な仕組み(道具は必要なし)



FUJI xerox

BFMとBFM+DPI-Cの違い(その1)

各テストプログラムを使って、シミュレーションを実行するとき

■ **BFM**の場合

テストベンチのコンパイル/エラボレーションが必要

■ **BFM+DPI-C**の場合

テストプログラムは**C**コンパイラでコンパイルし、
シミュレーション時にオブジェクトファイルを
-sv_libオプションで指定するだけ
(HDL側のコンパイル/エラボレーションは必要ない**)**

BFMとBFM+DPI-Cの違い(その2)

テストプログラムのライブラリ化

- **BFM**の場合

  HDLコードでライブラリを構築し、

  includeディレクティブでの取り込みになる

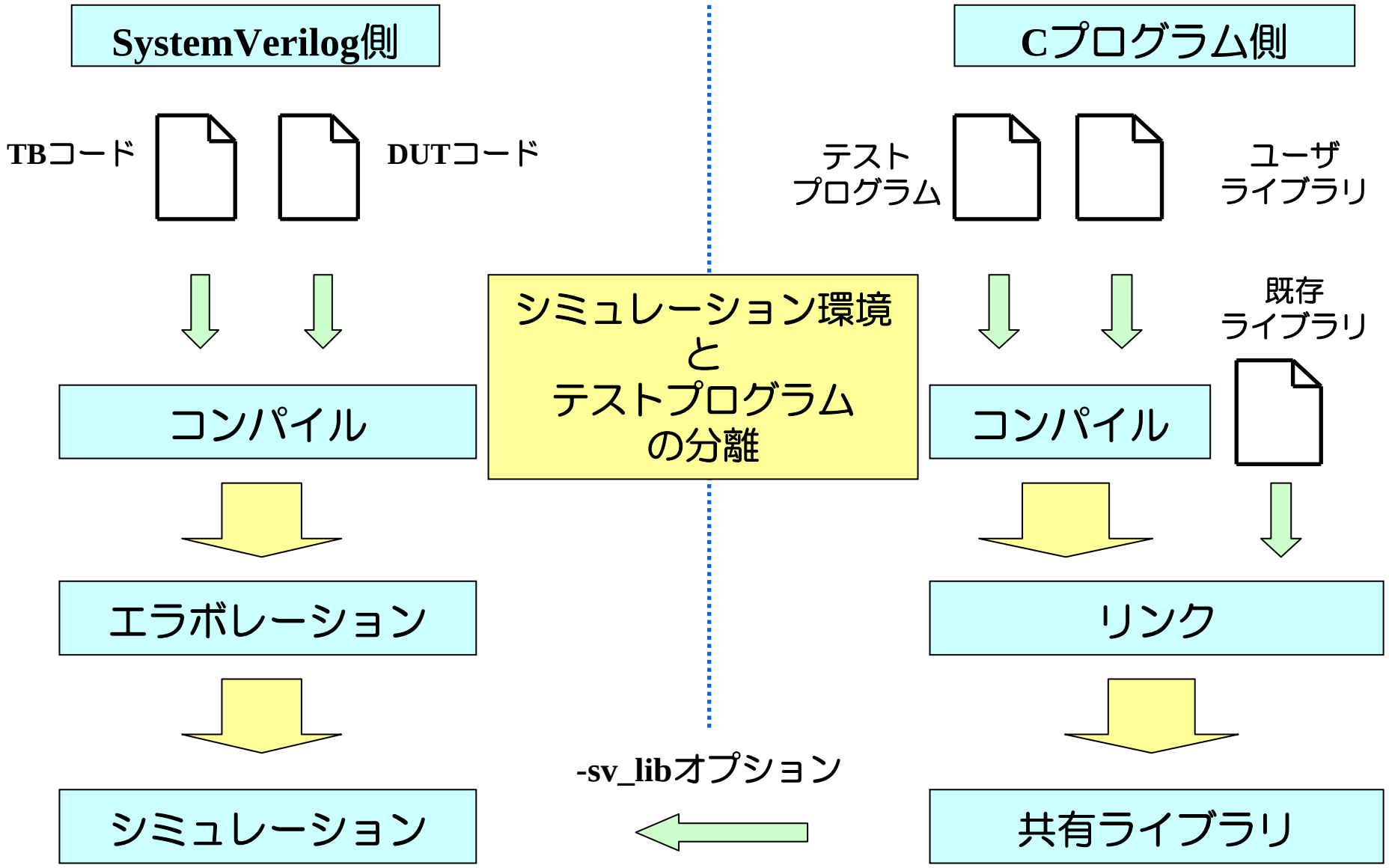
- **BFM**＋**DPI-C**の場合

  C/C++でライブラリを構築し、

  既存ライブラリなどのリンク可能

FUJI xerox

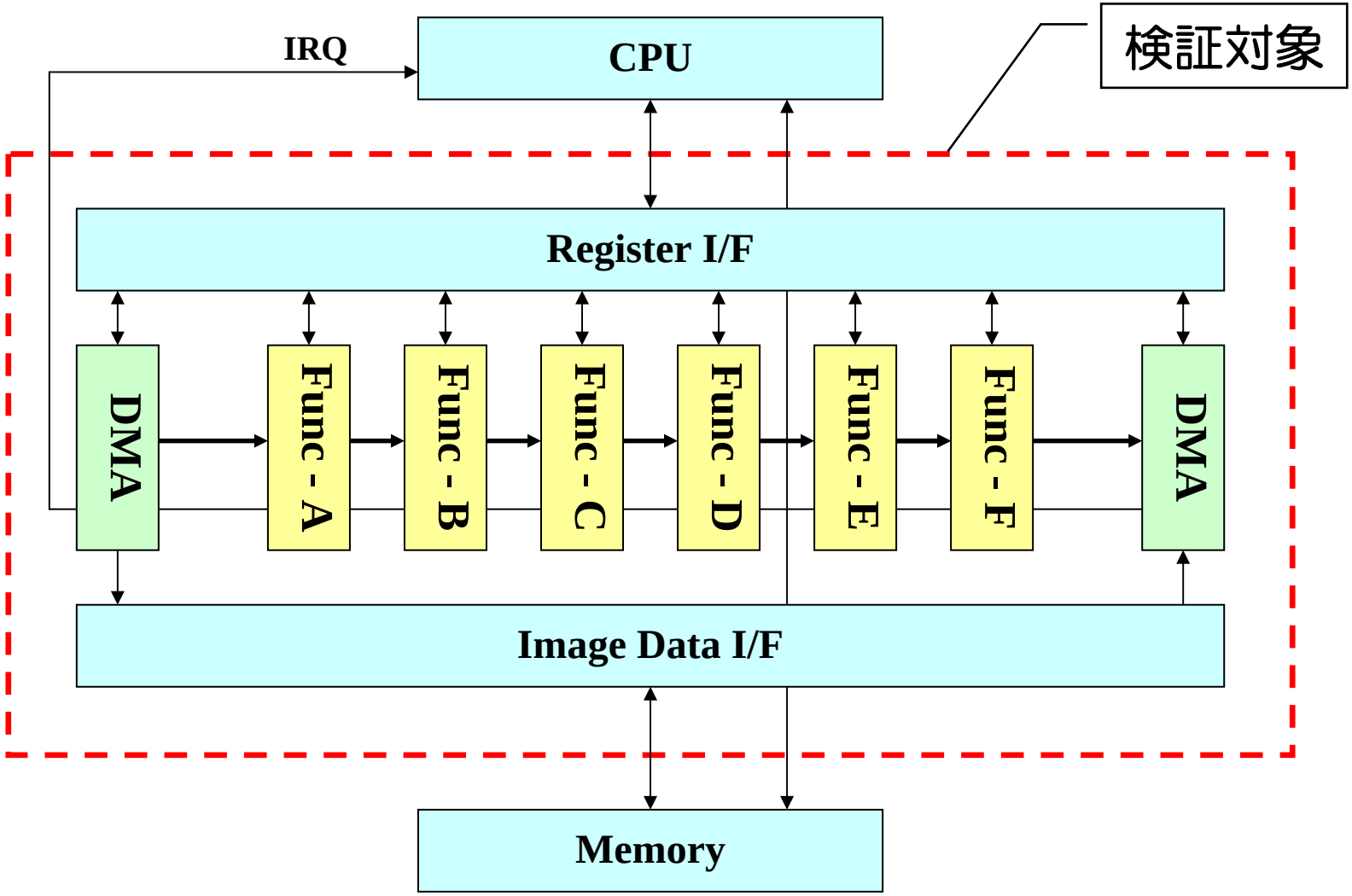
DPI-Cによるテストプログラム



FUJI xerox

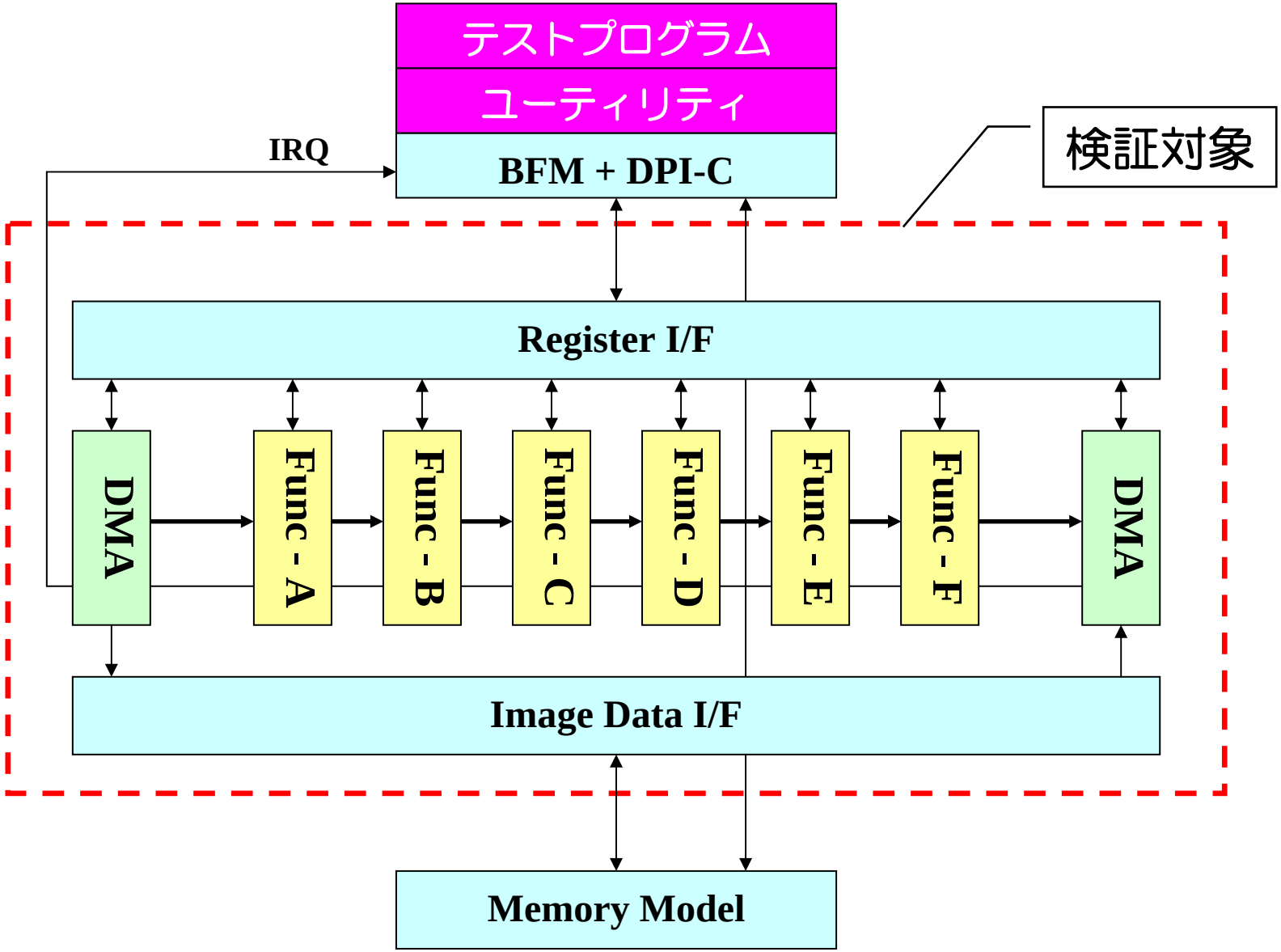
開発事例

開発事例：画像処理部の検証



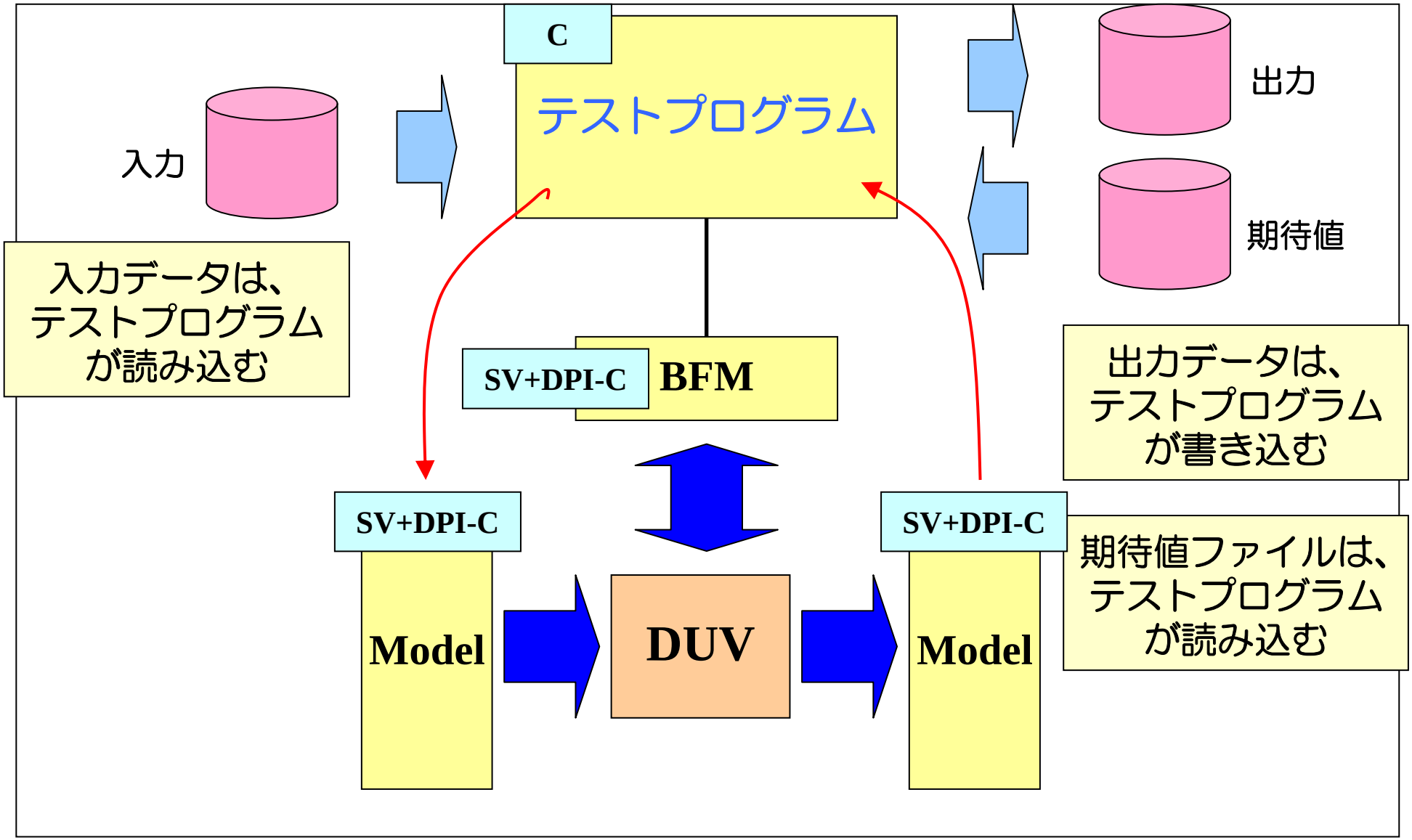
FUJI xerox

SystemVerilog DPI-Cを使った検証環境



FUJI xerox

プログラムと検証環境の関係



FUJI xerox

DPI-Cによるテストプログラム(その1)

Bus Functional Model

DPI-Cで、
C言語とのインターフェースを定義する

```
module BFM( ... );

  import "DPI-C" context task c_main () ;
  export "DPI-C" task Bus_Load, Bus_Store, Bus_Wait;

  initial begin
      c_main();   // C言語部
  end

  task Bus_Store( .. );
  endtask : Bus_Store

  task Bus_Wait( .. );
  endtask : Bus_Wait

  task Bus_Load( .. );
  endtask : Bus_Load

endmodule : test_prog
```

test_prog.c

```
int c_main(void)
{
    // ここに
    // テストプログラムを
    // 書く



        return 0;
}
```


FUJI xerox

DPI-Cによるテストプログラム(その2)

SystemVerilog側

```
initial begin
    c_main();
end


task Bus_Store( .. );
endtask : Bus_Store

task Bus_Wait( .. );
endtask : Bus_Wait

task Bus_Load( .. );
endtask : Bus_Load
```

Cプログラム側

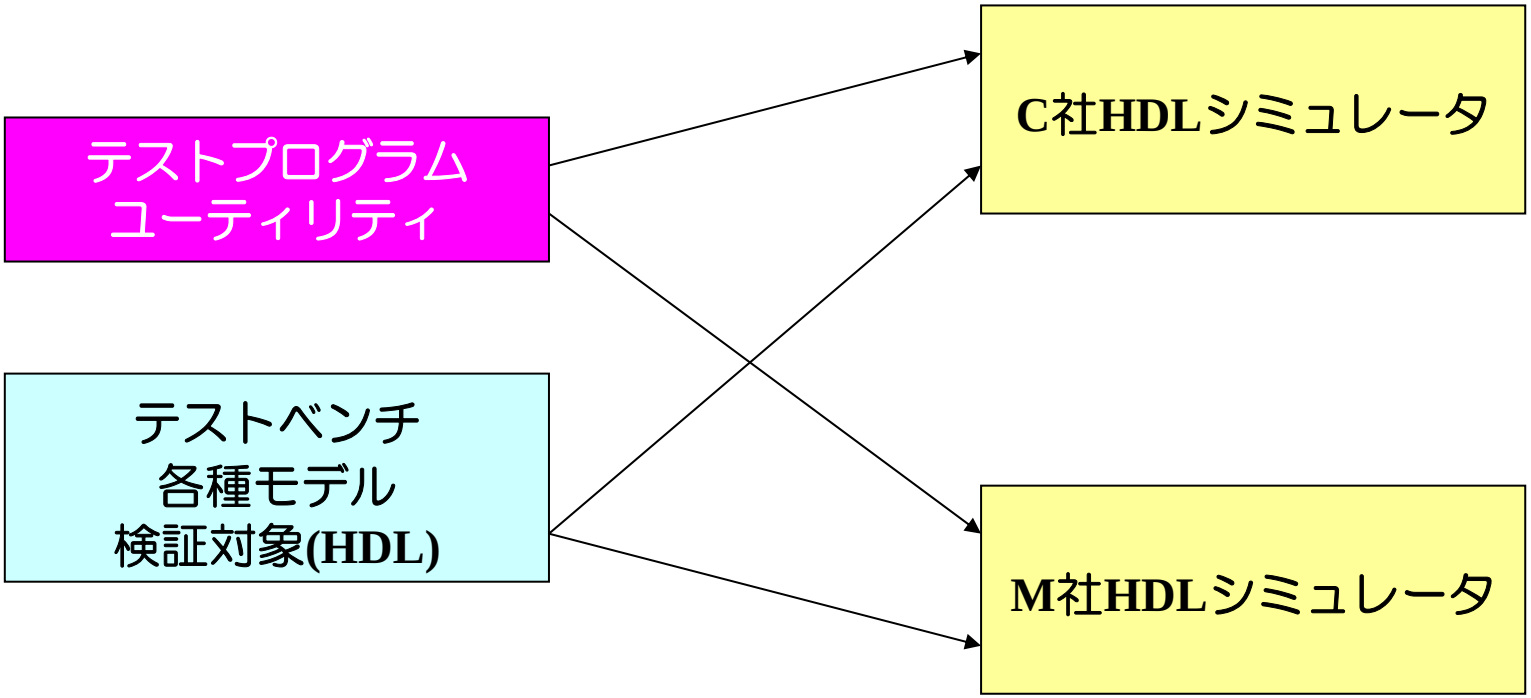
```
int c_main() {
  u_int32 addr, data;

  addr = 0x00000004;
  data = 0x12345678;

  Bus_Store( addr,  data );
  Bus_Wait( 10 );
  Bus_Load(  addr, &data );
  if( data != 0x12345678 )
    printf("Error¥n");
  return 0;
}
```

SVからCをCall

CからSVをCall

FUJI xerox

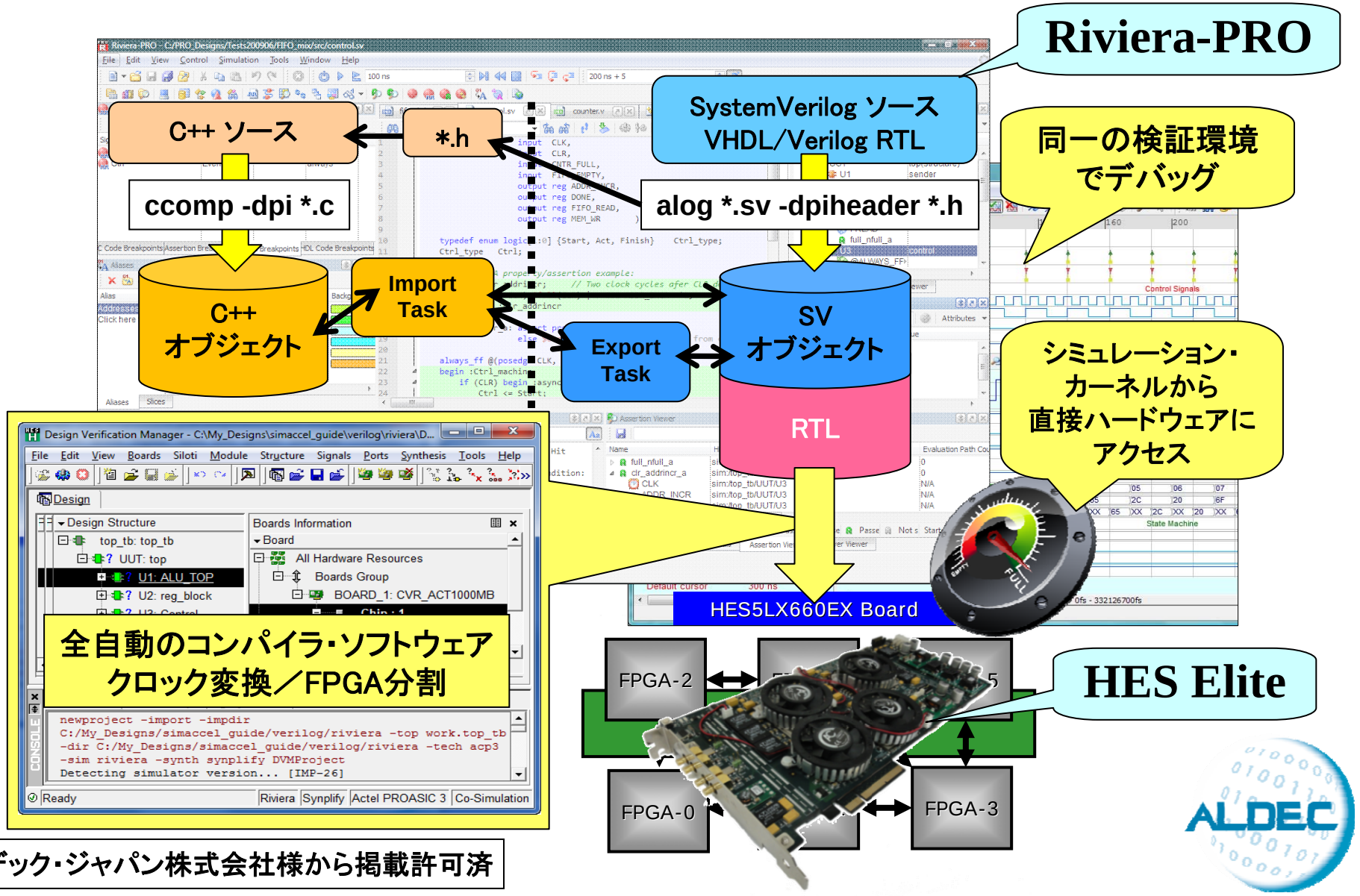
シミュレータでの利用



使用した2社のシミュレータ間では、
テストプログラム/ユーティリティ
の変更無しで動作確認できた。

FUJI xerox

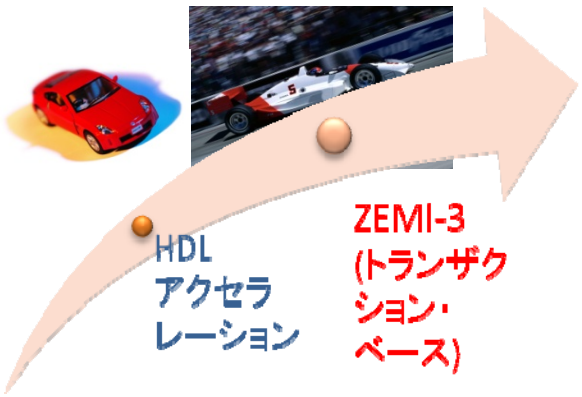
アクセラレータでの利用：Aldec社Riviera-PRO + HES

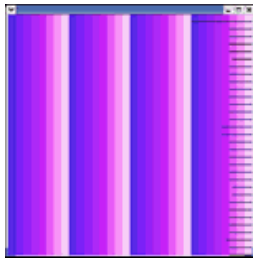


アルデック・ジャパン株式会社様から掲載許可済

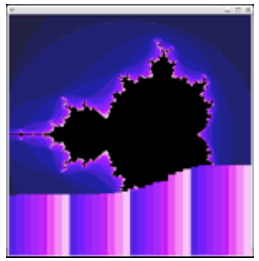
FUJI xerox

エミュレータでの利用：EVE社Zebu-Server



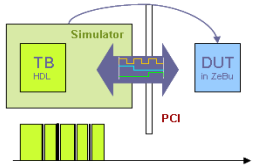


~ 100KHz

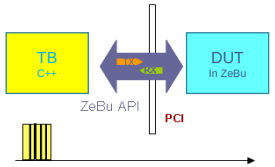


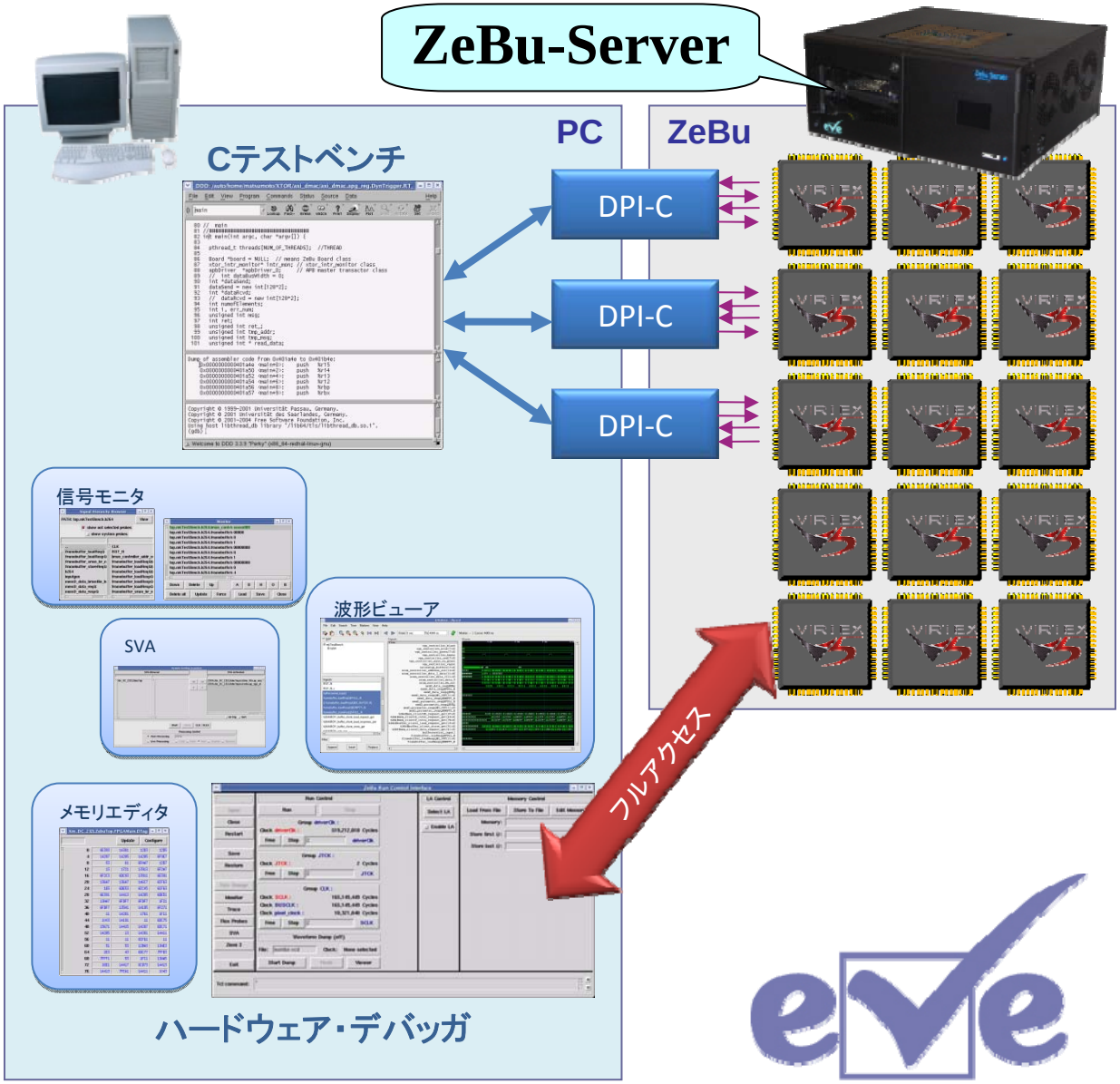
~ 30,000KHz

HDL
アクセラレーション



ZEMI-3

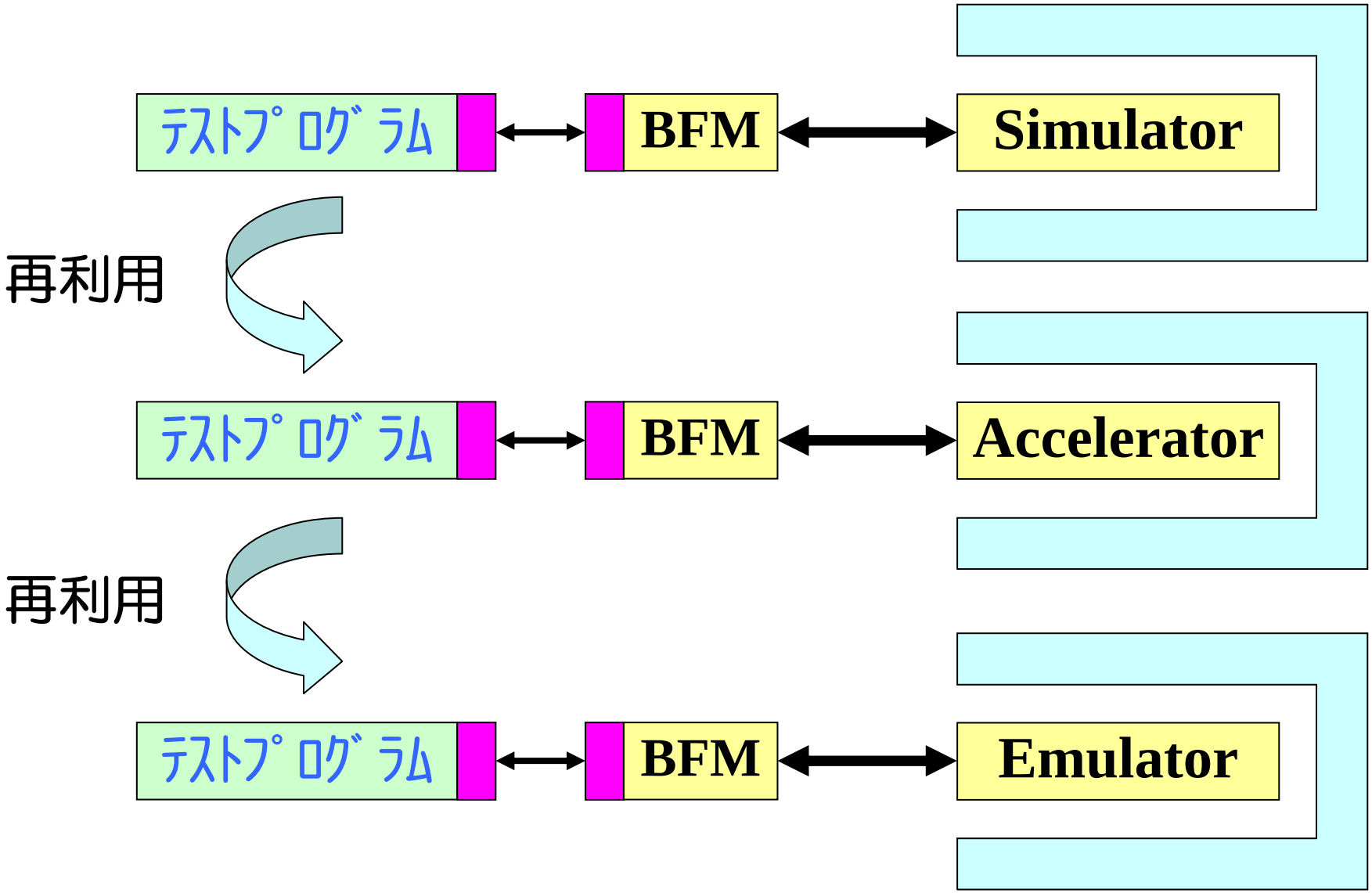




日本イヴ株式会社様から掲載許可済

FUJI xerox

テストプログラムの再利用



FUJI xerox

FUJI xerox