

CAE Plus社製品による システムASICプロトタイピング

1999年1月

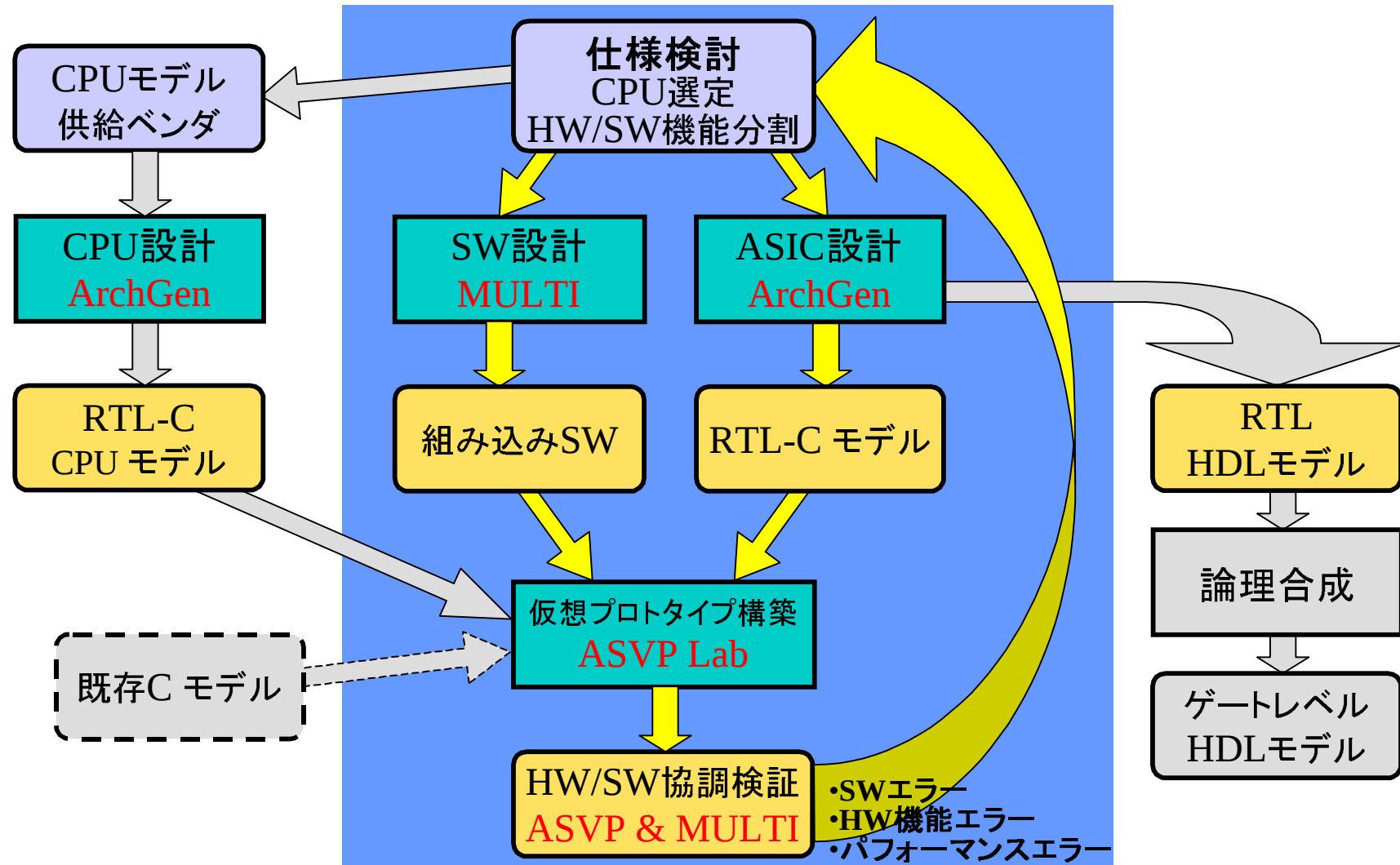
Advanced Data Controls Corp.
株式会社アドバンスデータコントロールズ

システムASIC (SoC) 開発での壁

- より早い市場投入(Time to Market)
 - シリコンにする前にソフトウェアを検証
 - 確実なHW/SW同時設計
 - コアとソフトウェアの効果的な再利用
- HW/SWパフォーマンス改善
 - チップサイズの低減
 - パーティショニング解析 → HW構成をコンパクトに
 - 低消費電力化
 - SWを実行しながらダイナミックなデータアクセス特性解析
 - スループット改善
 - FIFO、共有メモリなどリアルタイム性能解析

問題解決には RTL-C モデルによる仮想プロトタイプが最適

仮想プロトタイプ開発フロー



上位設計フロー シングルソースSoC設計フロー 既存設計フロー

Behavior

フロー図作成とアーキテクチャ最適化
ArchGen

組み込みSWとシステムパフォーマンスの改善

デバッガを使った組込ソフトの検証
MULTI

RTL

RTLモデル生成
ArchGen

VCD
タイミング
データ

C ← Verilog
コンバータ
Verilog2C

論理合成可能なRTLレベル設計
Verilog

(開発中)

RTL-C
モデル

デバッグIFとモデルのリンク
ASVP Lab

仮想
プロトタイプ
ASVP

タイミング
解析ツール
Signalscan等

論理合成用ロジック
論理合成ツール

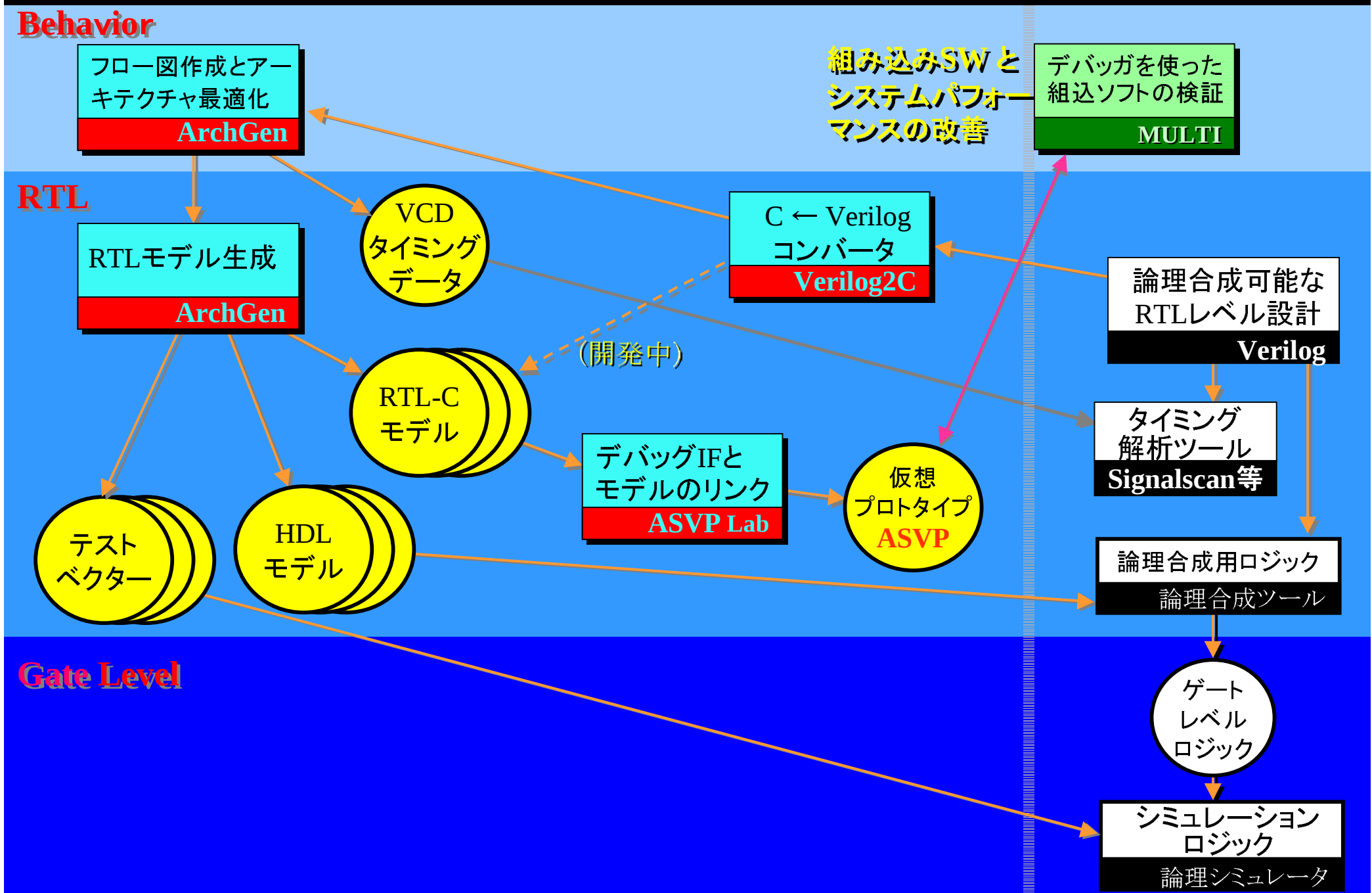
Gate Level

テスト
ベクター

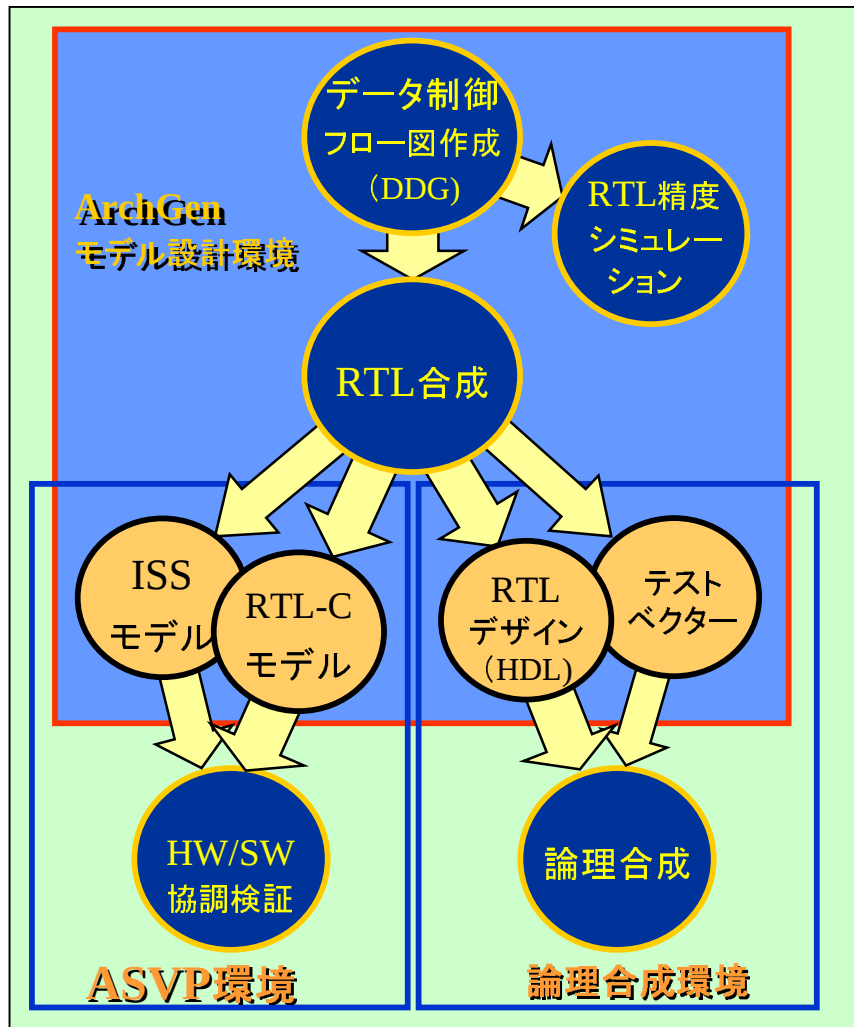
HDL
モデル

ゲート
レベル
ロジック

シミュレーション
ロジック
論理シミュレータ

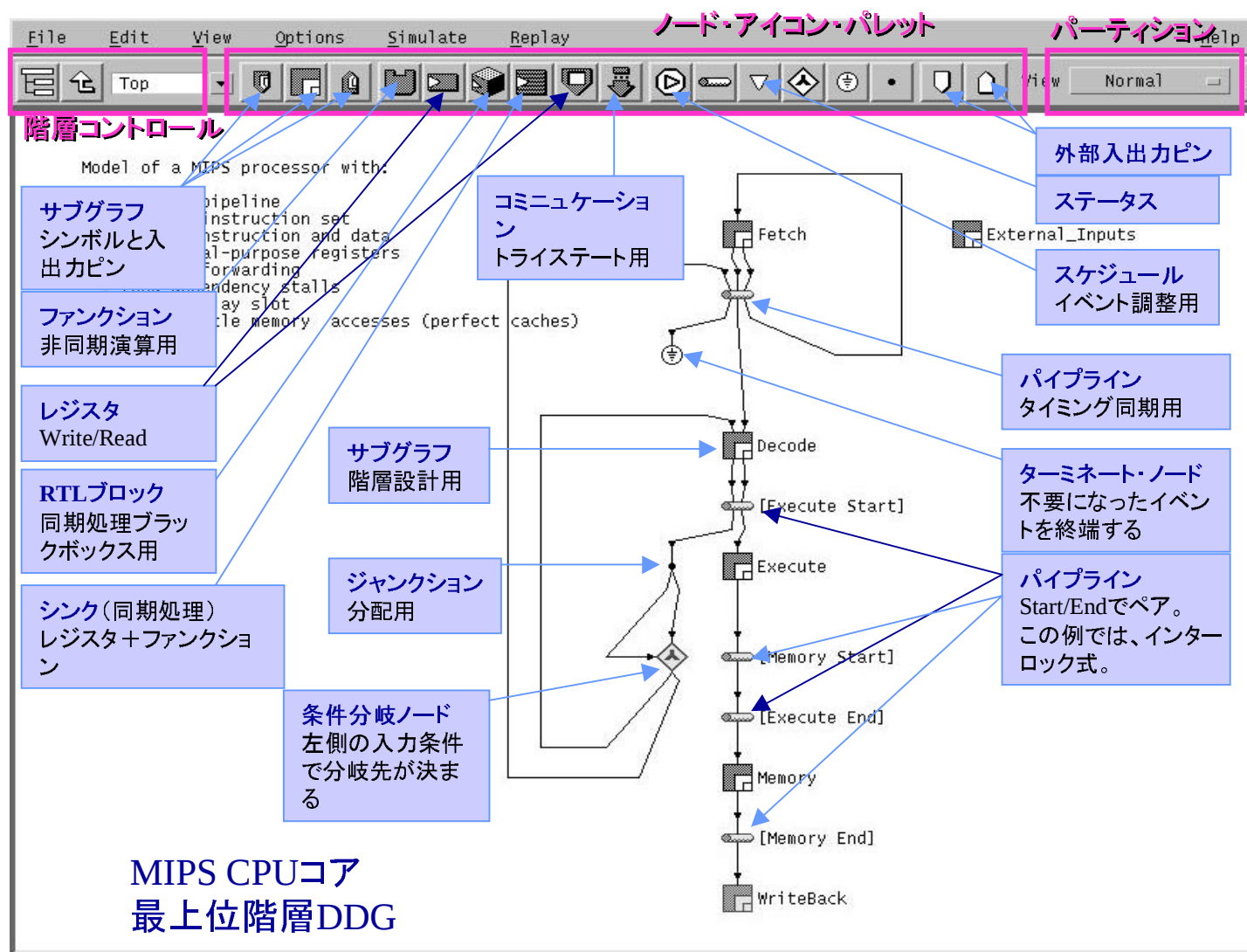


ArchGen 概要



- ArchGenはHDLモデル開発ツール
 - 1994年、ビヘイビア・レベルのHDLモデル設計環境として開発
- 視覚的にデータ・フローを把握し易いデザイン・キャプチャ
- 有限ステートマシンを自動設計
 - ステート制御ロジック設計は不要
- 論理シミュレータ内臓
- RTL-Cモデル及び論理合成可能なRTLモデル (Verilog-HDL, VHDL) とテストベクタを自動生成
- Synopsys, BuildGates, AutoLogic 各々に最適なHDLコードを生成
- ISSモデル組み込み可能

ArchGen デザイン・キャプチャ



ArchGen シミュレーション

The image displays the ArchGen simulation environment. The main window shows a circuit diagram with various components and signals. A red box highlights the 'Time' field in the top toolbar, showing '33320' and 'Cycles' as '833+0'. A red arrow points to a signal in the circuit diagram, labeled '現在伝播中のデータ。アークがハイライトされる。' (Data currently being propagated. The arc is highlighted).

On the right, there are two xterm windows. The top one shows a Verilog module definition for an ALU. The bottom one shows a Verilog function for the ALU.

At the bottom right, there is a 'DAI Signalscan DX 5.3 - Waveform Window:1' window. It displays a list of signals and their values at a specific time (33,320 ns). The signals are grouped into 'A' and 'B'. The values are shown in hexadecimal and decimal formats.

Verilogファンクション

```
module ALU(c,a,b,ctrl);
    output [31:0] c; // AG_OUTPUT_DELAY 16
    input [31:0] a,b;
    input ctrl; // AG_CNTPORT sub=0,add=1,AG_INACTIV
    reg [31:0] c;

    always @(a or b or ctrl)
    begin
        if (ctrl)
            c=a+b;
        else
            c=0;
    end
endmodule
```

Verilogファンクション

```
function void coprocessor_v2c:ALU
    param ctrl
    param b
    param a
    delay 16
endfunction
```

(Signalscan)

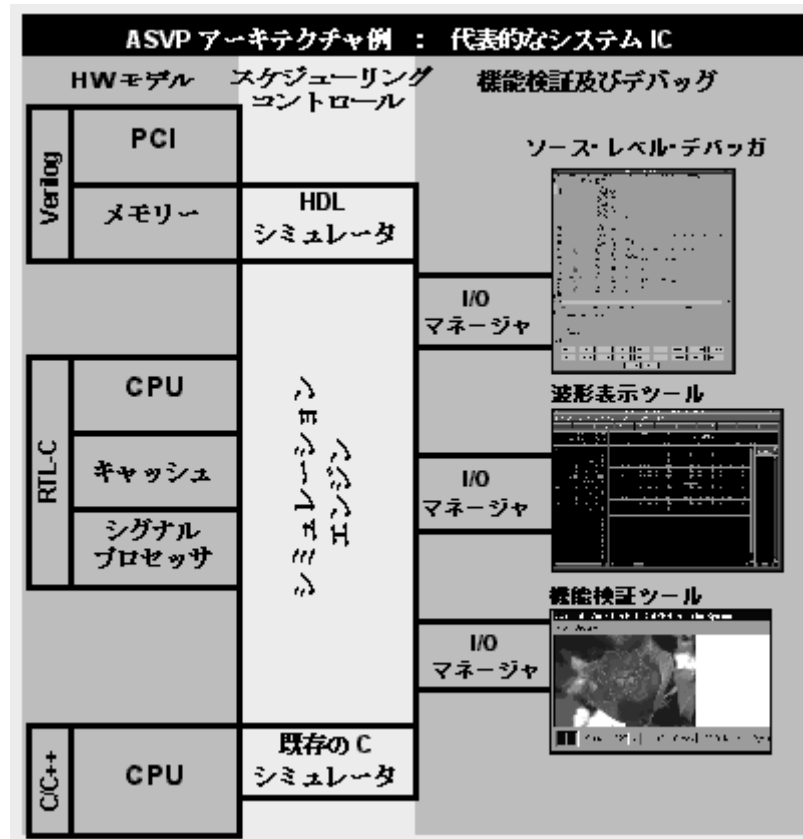
Signal	Value
X1[31:0]	000001BC
X2[31:0]	000001A7
Y1[31:0]	00000108
Y2[31:0]	0000018E
Convex_Rdy[0]	0

ArchGenの優位性

- RTL設計時間の短縮
 - 既存RTLデザイン(HDL)をビヘイビア・レベルに拡張
 - グラフィカルなデータ・フロー図によりドキュメント性の向上
 - 自動的にシンセサイザブルなRTLコード生成
 - HDLでのコーディングと比較して、4倍の効率で仕様をキャプチャ、設計時間を10倍以上短縮
- RTLの最適化
 - グラフにより、並行処理の検証が容易
 - 各種動作解析レポートによりボトルネック発見が容易
- CとHDLモデル双方を単一ソースで設計可
- 既存HW設計、SW設計ツールとの連携がスムーズ

ASVP Lab 仮想プロトタイプ・ビルダー

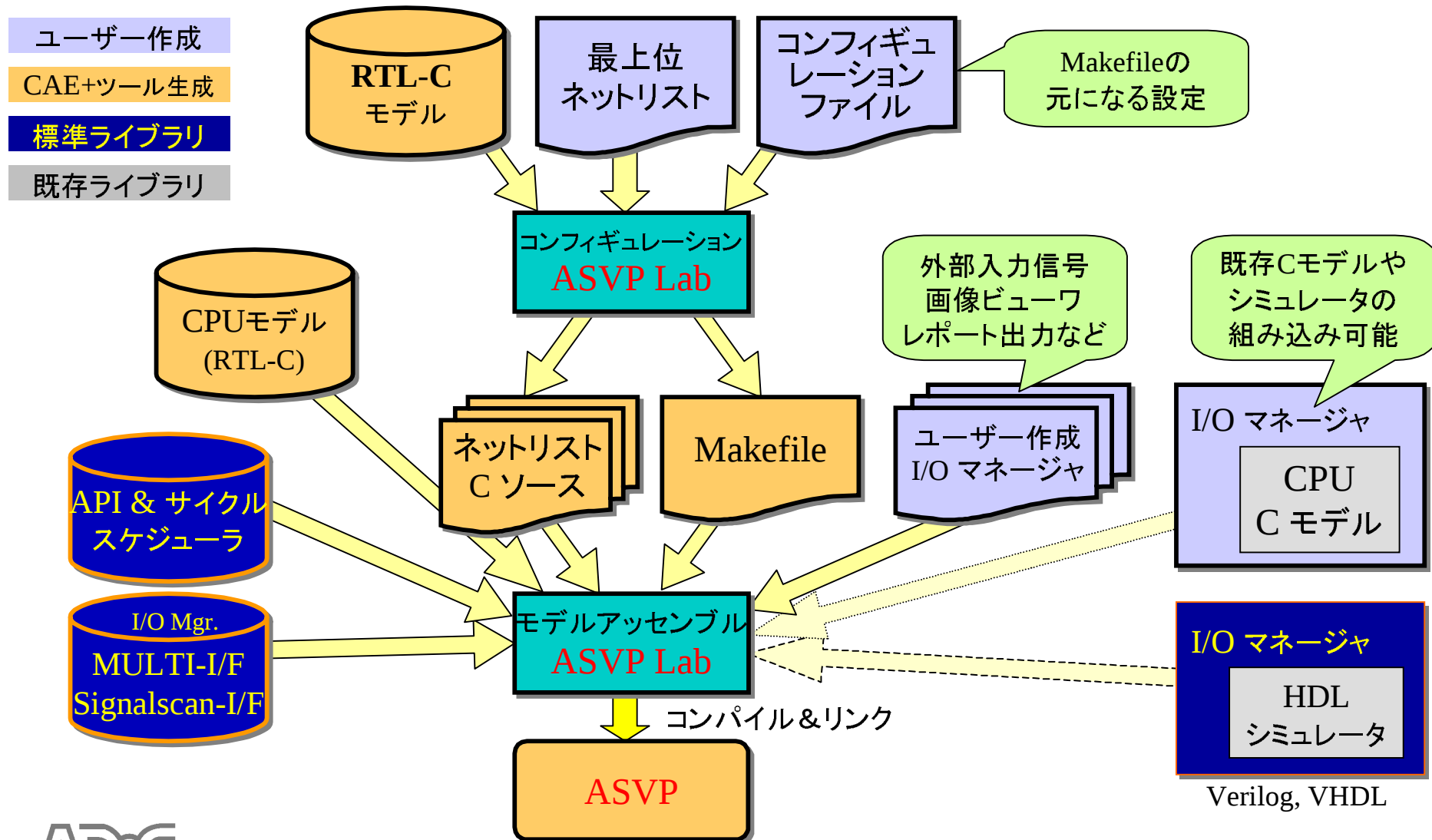
(Application-Specific Virtual Prototype)



ASVP Lab はオープン環境

- C言語ベースのツールやモデル、シミュレータなど統合可能
- Green Hills社のMultiデバッガ I/Fを標準装備
- タイミング解析用にDAI社のSignalscan I/Fを標準装備
- HDLシミュレータとのI/Fを用意
- 公開されたAPIをベースにカスタムI/Oマネージャの作成結合が容易

ASVP作成フロー



ASVPの優位性

- 早期に、ハードウェアを伴ったソフトウェア検証
 - デバイス設計前に検証
- HW/SWの信頼できる最適化
 - パフォーマンス検証
- 高精度な HW/SW協調検証
 - RTLレベルの精度で → RTL-accurate Cモデル
- 高速な HW/SW協調検証
 - HDL最速シミュレータと比較して10倍以上の速度
 - ISS と RTL-C を切り替え可能
- 既存HW設計、SW設計ツールとの連携がスムーズ

成功事例

- Motorola
 - ArchGenを使ってVHDL を生成しシリコン(CopperGlod内のFFTなどの機能ブロック)を作成→エラーのないVHDL生成の実証
- Hughes Network Systems
 - TX3904コアの提供を受け、それに独自設計の周辺ブロックを結合しシステムICの仮想プロトタイプ作成→HW/SWの共同開発と協調検証の実現
- NEC(US)
 - 3ヶ月未満でV851のコアモデル(クロック精度)を開発し、UltraSparc 2で80,000cpsというシミュレーション速度を実現

HW/SW Co-Verification用 論理シミュレータとHWモデル

シミュレータの現状(1)

協調検証用論理シミュレータに求められるものは、「スピード」と「精度」。既存のツールでそれを求めると

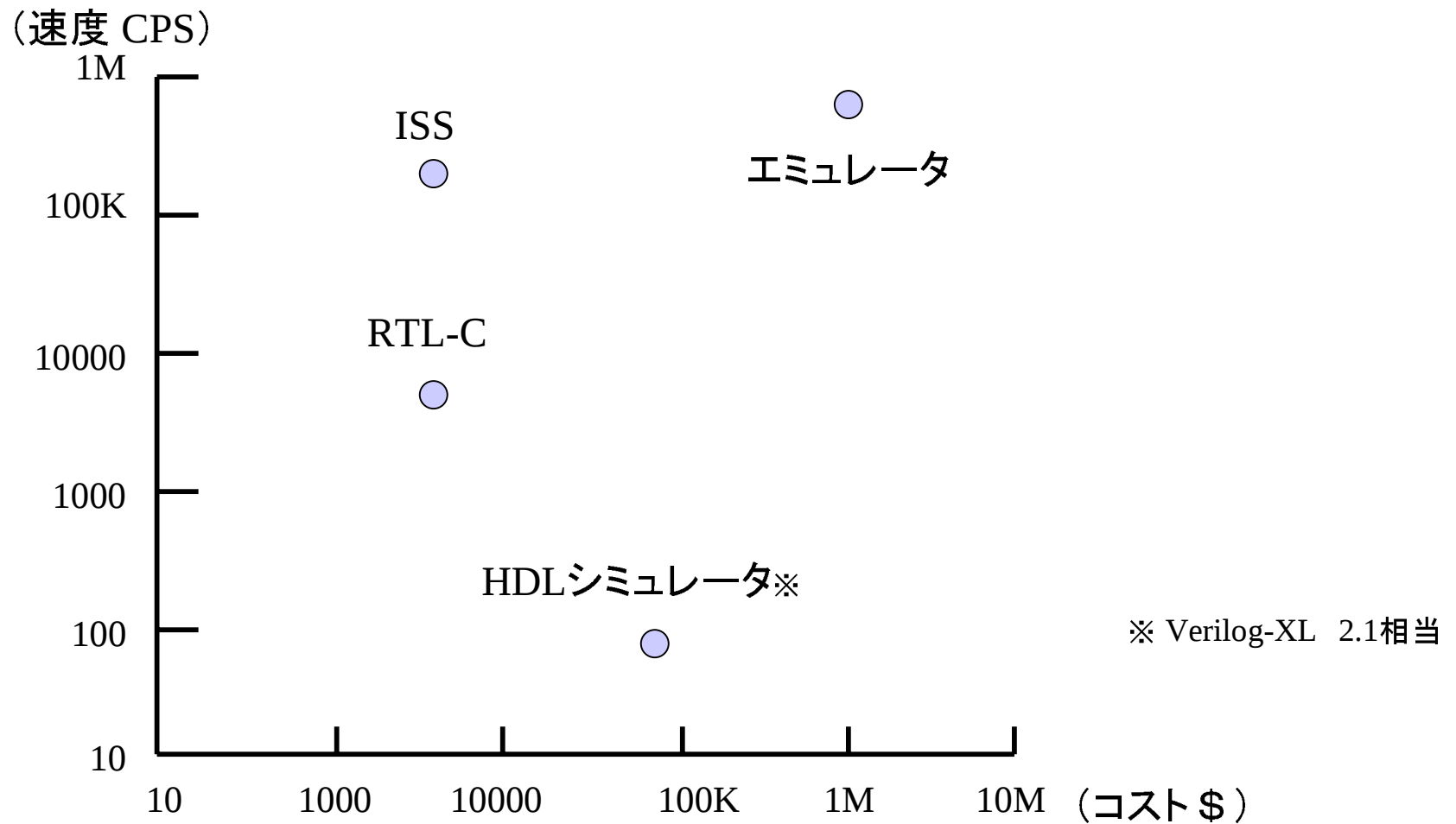
- HWエミュレータ
→最高速を誇るが、数千万円～億単位の投資
- 最新テクノロジーを採用したHDLシミュレータを導入
→投資が増える割には、劇的なスピードアップにつながらない
- HDLシミュレータ＋C言語(PLI)を利用した高速化
→Cファンクション部分の時間精度がとれない

シミュレータの現状(2)

- ISSの利用
 - 周辺ブロックが無いため、CPUとメモリ程度の検証
 - 時間精度／ビット精度がとれない
- ISS＋HDLシミュレータ
 - 時間同期したシミュレーションが難しい

このような背景から、精度を保ちつつ高速化可能なC言語ベースの論理シミュレータが必要になってきた。

シミュレータ性能比較(1)



シミュレータ性能比較(2)

機能	RTL-C	HDL	パフォーマンス	ISS
シミュレータ	不要	必要	必要	不要
シミュレーション速度	15,000cps	15cps	150cps	100,000ips
パフォーマンス検証	○		○	
ハードウェア検証	○	○		
ソフトウェア検証	○			○
協調検証	○			
コンカレンシー	○	○	○	
タイミグ	○	○	○	一部
ビットレベルサポート	○	○		
可能なデータ型	Cの型	整数	Cの型	Cの型
シミュレーションベース	サイクル	イベント/サイクル	イベント	

C言語ベースHWモデルの問題点

- 処理がシーケンシャルである
 - HWは全てパラレル
- 実時間の概念がない
 - 遅延を計算できない
- ビット(バス)精度を出すのに手間がかかる
 - C言語は型依存(char, short, long, int)
ビット接続や分離のようなHW寄りの記述が苦手
- HDLデザインに流用できない

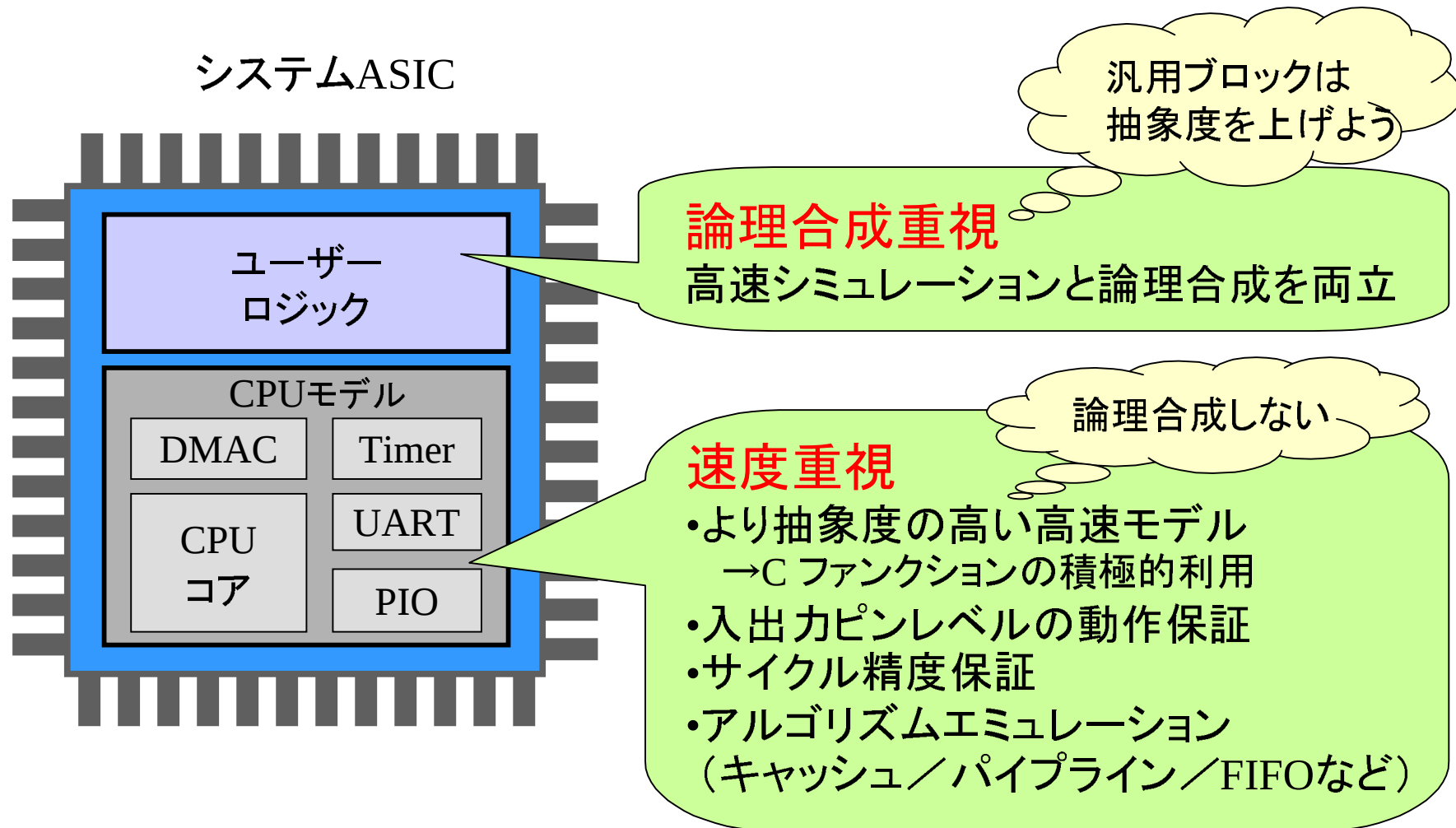
CAE Plus社のArchGen/ASVPは、これらの問題を解決する、C言語(RTL-C)HWモデル作成ツール

RTL-Cモデル

(RTLレベルの動作記述が可能なC)

- 単独で実行、シミュレータ不要
- 高速シミュレーション(20,000～5,000cps)
- パフォーマンス、HW/SW検証に利用可能
- コンカレンシー(並行性)
- サイクルベースの正確さ
 - 外部IOはサイクルベースでインターフェースするが、HWモデル内部はサイクル内の遅延も考慮する。
- HDLシミュレータやHWEミュレータより低コスト
- ISSモデルの組み込みが可能

協調検証用途HWモデルの二面性



ユーザーロジックのHWモデル

- 高速シミュレーションと論理合成を両立できないと効率が悪い
 - シングル・ソース
コンバータによるHDLからCモデルへの変換
 - 論理合成可能なHDLコードの生成
 - 協調検証用高速モデルの生成
- 既存HDL-IP転用の可能性

デバイス設計用HDLモデルは検証速度を追求してない。
故に、協調検証用途としての最上の選択では無い。

 - アルゴリズムを変えずにC言語モデルに変換
(開発中:条件付で、かなりの部分が変換できるのでは?)

HW/SW協調検証ツールに求められる 拡張性

- 既存クロスコンパイラ／デバッガと容易に接続
 - GUIなど、既存ツールと同一が望ましい
 - ISS / ICE / HWシミュレータ毎に違うようでは使いにくい
- デバッグ用外部信号やモニタとの接続
 - ユーザー・プログラムとのI/Fライブラリが必要
 - 同期／非同期信号、ON/OFF制御、画像ビューワ等
- HWのタイミング波形のプローブ
 - HWシミュレータを用いることで、チェックは容易になる
 - 実機では観測不能な場所も可
 - 高価なロジアナや低容量プローブ不要