



エミュレーション：全てのデスクトップで実行可能に

ケーススタディ：AXI Switchのデスクトップエミュレーション

概要

FPGAの集積度が高くかつコストが下がり続けており、デザインの規模や複雑さが増してシミュレーションの実行性が失われつつある中、SoCやIPベリフィケーションの広範に渡ってエミュレーションがメインストリームとして開発されてきました。エミュレーションをデスクトップでも使用可能なメインストリームにするためには何がキーとなるのでしょうか？

このホワイトペーパーでは、第1世代のエミュレーションシステム、次世代のソリューションのキーコンポーネント、そしてAXI switchのパフォーマンスバリデーションのための次世代のデスクトップエミュレーションを使用したケーススタディを示します。

目次：

- インTRODクシヨン
- 第1世代のエミュレーションの問題
- 次世代のエミュレーションのソリューション
- ケーススタディ：AXI switchのパフォーマンスバリデーション
- エミュレーションのボトルネックの管理
- 利用可能なテクノロジー：コ・エミュレーションスタック
- 結論

第1世代のエミュレーションの問題

- 高価
- セットアップと管理が複雑
- テストベンチとDUT(Design Under Test：テスト対象のデザイン)の間のボトルネックを取り除くことが困難

次世代のエミュレーションのソリューション

- 手軽な費用：汎用のFPGAボードと高速なシリアル/パラレルインタフェースを利用可能
- シンプル：新しいテクノロジーが簡単に利用可能
- 標準のコ・エミュレーションAPI(SCE-MI)
- 物理的なインタフェースの仮想化
- 高性能：トランザクタと合成可能なテストベンチがテストベンチとDUT間のボトルネックを排除

最適なエミュレーションパフォーマンスを達成するためのキー

- エミュレーションハードウェアにテストベンチコンポーネントを格納することで、コ・エミュレーションインタフェースとシミュレーション時間を可能な限り取り除く
- ハイレイテンシのコ・エミュレーションインタフェースでは特に、コ・エミュレーションインタフェース間のアーキテクチャは、トランザクションレベルのテクニックを使用することでレイテンシに無関係になる
- 適切に構築されたコ・エミュレーションスタックは、ハードウェア/ソフトウェアパーティショニングをよりフレキシブルにし、異なるアプリケーションのサポートを容易に、可搬性を高める

イントロダクション：シミュレーションはシステムの複雑さの増加ペースについていけない

改善のための第一歩であるにも拘らず、チップベリフィケーションは多くのリソースや時間を必要とします。デザインベリフィケーションを実行するツールシミュレーションが基本的なボトルネックになっていました。それは巨大で複雑なシステムを検証するためのパフォーマンスを全く提供できていません。このボトルネックを取り除くことを約束する一方で、エミュレーションはハイエンドでマーケットサイズが大きく、膨大な労力が必要であるプロセッサやグラフィックチップで、専用のエミュレーションを実行するエンジニアがいるような開発において使用されてきました。

しかし現在、FPGAの集積度が高くかつコストが下がり続けており、デザインの規模や複雑さが増してシミュレーションの実行性が失われつつある中、SoCやIPベリフィケーションの広範に渡ってエミュレーションがメインストリームとして開発されてきました(表)。ハードウェアのエミュレーションのセットアップと運用をドラスティックにシンプルにするための転換点として何が必要でしょうか？エミュレーションを簡単に低予算で実行できるメインストリームにするためのキーポイントは：

- エミュレーションのミドルウェアの可用性：
 - シミュレーションからFPGAをそのまま使用できる
 - 汎用のFPGAボードと標準のハイスピードシリアルインタフェースを使用できる
 - 特定のエミュレーションプラットフォームに固定されない
- 迅速なテストベンチの作り直しのための合成可能なテストベンチとトランザクタ

このホワイトペーパーは第1世代のエミュレーションシステムの問題点と、次世代のエミュレーションシステムを実現するキーコンポーネントをレビューし、AXI switchのパフォーマンスバリデーションのための次世代デスクトップエミュレーションを使用したケーススタディを記載します。

アプリケーション	テスト	サイクル	シミュレーション	エミュレーション
Video Codec	D1 H.264 の 1 秒間のエンコード	600M	15 時間	10 秒
Embedded RISC	Linux ブート初期化	16M	140 時間	16 秒
ベースバンド復調	偏重された入力信号	1188M	27 時間	24 秒
AXI switch	バンド幅とレイテンシテスト	50M	9.9 時間	1 秒
TCP/IP 負荷	1Gb/s 信号の 1 秒間	300M	7 時間	20 秒
ネットワークプロセッサ	10Gb/s 信号の 1 秒間	200M	185 時間	20 秒

表 1. シミュレーションボトルネック

第 1 世代のエミュレーションの問題

図 1 は、テストベンチがワークステーション内のシミュレータで稼動し、エミュレータ内のDUTと適切なコ・エミュレーションを介して接続されている第1世代のエミュレーションシステムを示しています。

エミュレーションハードウェアの高価格に加えて、複雑さが広範に使用されることを阻害する基本的な要因でした。セットアップは時間を浪費し、リソースを集中させる必要があります。コ・エミュレーションのリンクのAPIは退屈なビットレベルのデータバックとタイミング同期を必要とします。リンクとシミュレータのボトルネックは解決されず、パフォーマンスが達成されません。バグを特定したり解決することは非常に困難でした。単純なバグを含むDUTはシステムリセットを引き起こし、問題の原因を解決できないためにプロジェクトでエミュレーションを放棄する必要性を引き起こします。

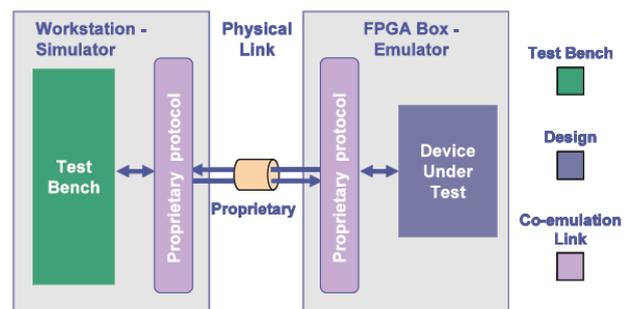


図 1. 第 1 世代のエミュレーションシステム

次世代のエミュレーションソリューション

図2は仮想化された物理的なリンクと標準のコ・エミュレーションプロトコル、トランザクタ、合成可能なテストベンチによる次世代のエミュレーションシステムを示しています。これらのコンポーネントはセットアップや簡便な使用、リンクとシミュレータ間のボトルネックの問題を解決します。

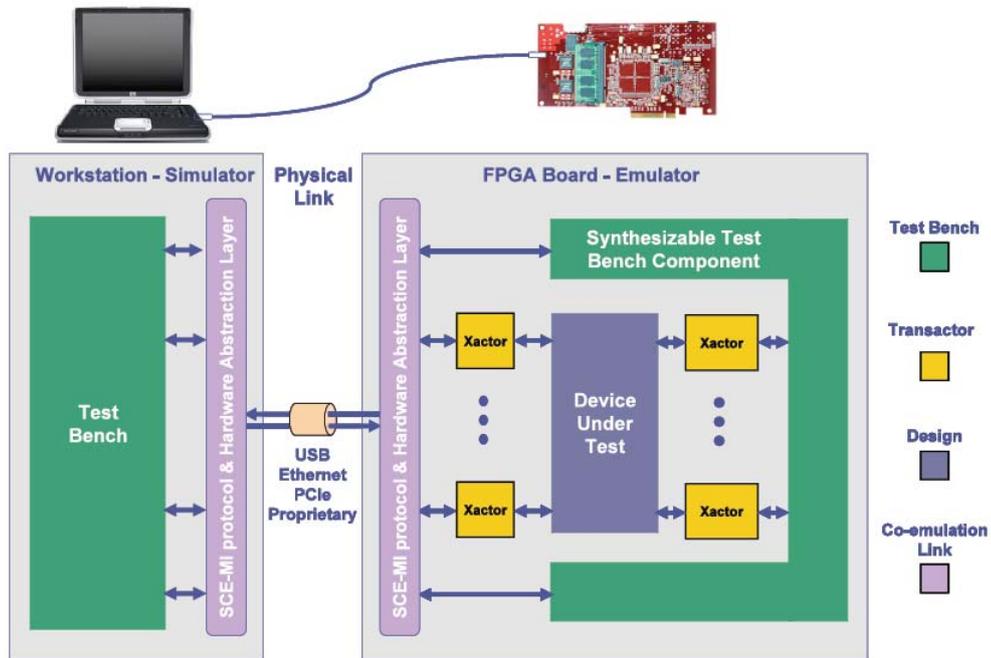


図 2. 次世代のエミュレーションシステム

ハードウェア抽象レイヤ

ユーザのアプリケーションから物理的なリンクを引き離します。リンクは独自のものや標準のもの(例えばEthernet、USB、PCI-eなど)、シリアル、パラレルを問いません。ユーザはアプリケーションに合わせたハイレベルのデータ構造を定義することができます。全てのローレベルのリンクデータやタイミングの詳細は、リンクで提供される抽象レイヤのハードウェアに隠されています。

SCE-MI

SCE-MIはソフトウェアとハードウェアの両方のサイドの標準のAPIでシミュレーションとエミュレーションコンポーネントの相互運用を確実にし、エミュレーションの全てのパフォーマンスを利用するための通信プロトコルを提供します。

トランザクタ

トランザクタはデザインの外部インタフェースの仮想ターミナルを通してテストベンチの迅速なセットアップをもたらします。

合成可能なテストベンチコンポーネント

合成可能なテストベンチコンポーネントはシミュレータとコ・エミュレーションリンクの他のボトルネックを解消します。これによってDUTのすぐ近くにデバック機能を配置し、DUTのパフォーマンスを低下させることはありません。

メーカー	ボード	容量(ASIC ゲート相当)
Hi Tech Global	V5IP-7000	2M
Nu Horizons	TB-LX330-EX	2M
EVE	PCI-card	5M
Synplicity	HAPS-51	2M
The Dini Group	DN9002K	4M

表 1 FPGA/エミュレーションボードの例

ケーススタディ：AXI switch パフォーマンスバリデーション

図3は汎用のML507 Virtex-5ボードとUSBケーブルでPCと接続されているAXI switch DUTの単純かつパワフルなエミュレーションシステムを示しています。

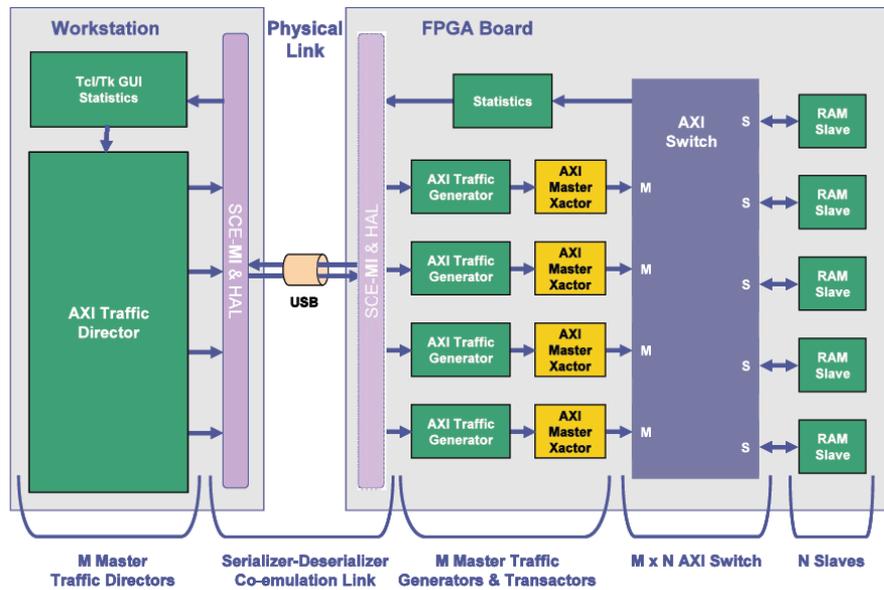


図 3. AXI switch DUT

AXI switchはマスタとスレーブのポート数を調整可能で、ランダムなトラフィックによってスイッチのレイテンシとスループットを検証することが目的です。トラフィックはシミュレータとエミュレータの両方に分割して格納されているテストベンチで生成と解析を行います。シミュレータの一部はTcl/Tkで書かれた設定やトラフィックパタンの解析を行うユーザインタフェースと、トラフィック設定パラメータを転送するAXIトラフィックディレクタをもっています。これらのコンポーネントはCで実行されていますが、容易にSystemCや混在言語RTLシミュレータで実行することもできます。エミュレータのテストベンチコンポーネントはAXIポートにランダムなトラフィックを出すAXIトラフィックジェネレータとメモリマップされたスレーブのRAMモジュールをもっています。

シミュレータとエミュレータ内のDUT間の全てはSCE-MIを経由して通信します。各通信チャンネルはシミュレータとエミュレータを半分ずつもっており、全ての通信はそれぞれの間で1対1で行われます。AXI switchは2つのトランザクタタイプをもち、1つはトラフィックの設定パラメータ(マスタポート毎に1つのインスタンス)で、もう1つはAXIトラフィックの統計を取るためのものです。SCE-MIはシミュレータとエミュレータ間のアンタイムのインプリメントのための一貫したフレームワークを提供し、ハードウェア抽象レイヤがUSBリンクを経由するトランザクションレベル通信をシリアル化、パラレル化を自動で行います。

AXI switchとテストベンチはVirtex-5 FPGA上のエミュレーションで、容易に50MHzで実行できます。そしてシミュレータとエミュレータ間のアンタイム通信の低バンド幅によって、全体のシステムはフルの50Mサイクル/秒で実行されます。これは混在言語RTLシミュレータで同じコンポーネントを実行する際の1.4Kサイクル/秒と比較して35,000倍高速です。これによってAXI switchのパフォーマンスを高信頼性を達成するための多数のスイッチの設定や多数のトラフィックシナリオのテストを実行可能になります。

その他のボトルネック防止

フルのエミュレーション速度で実行するためのキーは、コ・エミュレーションリンクとシミュレータのクリティカルパス以下に抑え、アムダールの法則(システムの一部を改良したときに全体として期待できる性能向上の程度を知るための法則)に気を配ることです。最初に考慮すべきことは物理的なリンクのレイテンシで、USBやEthernetではミリ秒のオーダーです。シミュレータとエミュレータ間の頻繁な同期が起こる場合にはレイテンシが支配的になり、パフォーマンスが1Kサイクル/秒以下に低下します。これがアンタイムで、レイテンシに影響を受けないテストベンチが重要であるという理由です。

次の気を配る必要のあるボトルネックはコ・エミュレーションリンクのバンド幅です。最大データレート480Mbit/秒のUSBは、ポートデータが各サイクルでシミュレーション側から送信されると仮定すると、8個のAXIポート(200bit/ポート)で300Kサイクル/秒程度をサポートします。これは、エミュレータの50Mサイクル/秒に比較して160分の1の速度です。リンクのバンド幅がボトルネックになるとき、これは一般的なケースでは、唯一の解決策は通信負荷を減らすことです。AXI switchの例では通信負荷は、例えばAXIトラフィック設定パラメータや統計情報を全てAXIトラフィックにする代わりにエミュレーション側に全て含むことで、転送データ量を減らすことで軽減されます。エミュレーション側のAXIトラフィックジェネレータが全てのAXIトラフィックを取り扱います。これは合成可能なテストベンチがリンクトラフィックをどのように管理できるかということを表す素晴らしい例です。

最後のボトルネックはシミュレーション時間です。リンクのボトルネックが管理できてもシミュレータのコンピュータに束縛されたブロックがパフォーマンスを低下させる可能性があります。このケースでは、第一段階でDUTをエミュレーションに入れてシミュレーションボトルネックを解消したのと同様に、合成可能なテストベンチをエミュレーションに入れるというパスがあります。例として、図3でエミュレータにあるAXIトラフィックジェネレータをシミュレータに入れると、トラフィック生成レートは良くして1M/秒の AXIのマルチポートトランザクションで、エミュレータのときに達成できる50M/秒の AXIのマルチポートトランザクションに遠く及びません。

コ・エミュレーション IPで気をつけること

AXIの例で示したエミュレーションの利点は、きちんと設計されたコ・エミュレーションスタックを適用することによって幅広いアプリケーションに応用できます。

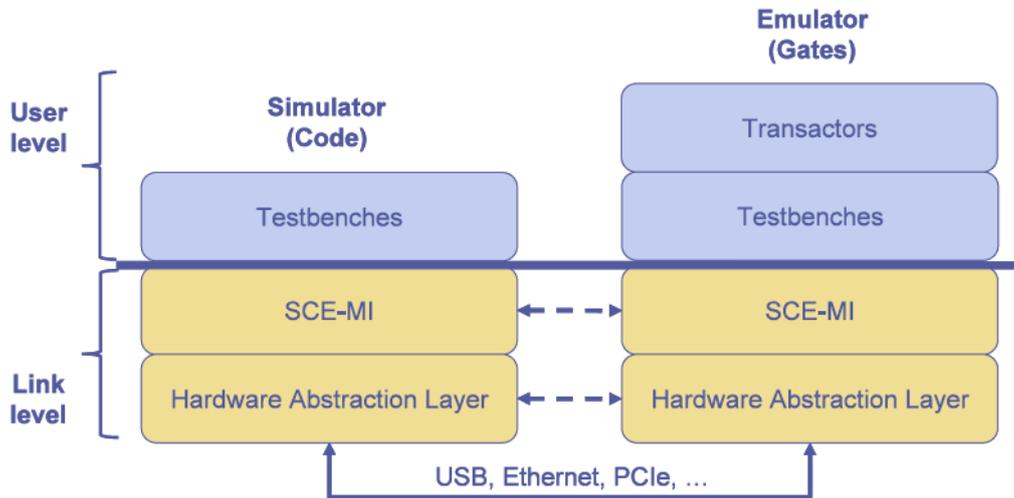


図 4. コ・エミュレーションスタック

図4は下の方の2つのハードウェアで移植可能なコ・エミュレーションリンクのレイヤと、上の方のユーザ固有のトランザクタとテストベンチのレイヤコンポーネントのスタックを示しています。物理的なリンクをいったん決めてポーティングするとリンクレイヤは変更する必要が無く、ユーザレイヤはリンクの両側で実行され、アプリケーションの変更やプロジェクトの進行に応じてシミュレータとエミュレータの両方でシームレスに実行できます。これらのブロックは一般的に高い抽象度で書かれているため、強力なパラメータ化、再利用性、編集性を活かす汎用的で高位の合成可能な言語によってインプリメントされるのが望まれます。この方法でインプリメントすると、図4のスタックは幅広いアプリケーションに適用できるロバスト(強固)なエミュレーション能力を提供できます。スタックのユーザレイヤの考慮すべき投資はエミュレーションハードウェアがもっている将来性をそのまま利用可能です。

幅広いアプリケーションとソリューションへの適用

良いトランザクタとテストベンチの高位合成を利用すると、いかなるアプリケーションでもシミュレータとエミュレータの間のアンタイムの通信にマッピングできます。これが図2に示した単純なデスクトップ設定の利用が、ビデオコーデック、無線ベースバンドプロセッサ、DMAコントローラ、メモリコントローラ、エンベデッドプロセッサ、I/Oインタフェースなどのアプリケーションで、シミュレーションのボトルネックを解消することに直接つながります。さらに、FPGAデバイスの利点は単体のボードで数百万ASIC等価ゲートの容量があり、容量は次世代の40nmデバイスでは2倍になることが予測されます。これは、全てではないにしろ、小中規模の多くのSoCの複雑なIPブロックにとって十分な容量です。

まとめ

このホワイトペーパーではデスクトップの利用を含むエミュレーションへの主な適用方法を示しました。FPGAの容量やシリアル/パラレルインタフェースのスピードは単体のFPGAでユニットベリフィケーションを実現できるほどに成長し、チップ全体と経済的にFPGAボードとシステムを使用してエミュレーションできます。導入にいたるキーとなる点は、設定とエミュレーションハードウェアの実行とシミュレーションへの統合を劇的に簡単にすることです。

標準のコ・エミュレーションAPI(SCE-MI)と物理的なリンクの仮想化を活用した、きちんと設計されたコ・エミュレーションスタックは、エミュレーションの使用を主流にするための中心となります。単純で迅速な構築を確実にすることに加えて、コ・エミュレーションスタックはよりフレキシブルなソフトウェア/ハードウェアのパーティショニングを可能にし、異なるアプリケーションを簡単にサポートできるようになり、移植性を高めます。

汎用で高位の合成可能言語、きちんと構築されたコ・エミュレーションスタックは様々な事例へのエミュレーションの適用を可能にします：

- ユニットとシステムベリフィケーション
- ソフトウェア開発
- アーキテクチャ検討とバリデーション
- システムのハードウェアアクセラレーション
- などなど

コ・エミュレーションスタックが構築された新規の大容量のFPGAボードはデスクトップにおける使用を可能にします。使い捨て可能な出来合いの50-100MHzのエミュレーションプラットフォームをもったエンジニアがどれだけ生産性を上げるか想像してみてください。

エミュレーションのためのBluespecのソリューション：全てのデスクトップで実行可能にする

Bluespecはエミュレーションのための本当にユニークな能力を提供します。この能力によってセットアップを容易に、使用を容易に、リンクとシミュレータのボトルネックを解消するなどの利点をもたらし、誰でも使用できるようにします。様々なFPGAプラットフォームで利用できるように、Bluespecのエミュレーションの向けソリューションと製品は以下のものを含んでいます：

- きちんと設計されたコ・エミュレーションスタックは迅速な設定と移植性があり、テストベンチの迅速なパーティショニングを実現できます。また、以下をサポートします：
 - SCE-MI API
 - 開発の柔軟性のための物理的な接続(例えばUSB、Ethernetなど)のハードウェア抽象レイヤ
 - エミュレーションプラットフォーム上のBluespecトランザクタと合成可能なテストベンチのシームレスな統合
- 強力なパラメータ化、高い再利用性、高い編集性のテストベンチ、トランザクタ、DUTを実現できる、汎用の高位の合成可能言語、ツールセット、IPライブラリ。許容できるパフォーマンスを達成するためにリンクとシミュレーション間のボトルネックを解消する。これは必然的にテストベンチコンポーネントをエミュレーションプラットフォームに移動させることを意味し、これは汎用の高位言語による設計と合成によってのみ迅速かつ効果的に行うことができます。

お問い合わせ先：

CYBERNET

サイバネットシステム株式会社
新事業統括部

〒101-0022 東京都千代田区神田練堀町3 富士ソフトビル
Tel: 03-5297-3295 Fax: 03-5297-3637

e-mail: bluespec@cybernet.co.jp
<http://www.cybernet.co.jp/bluespec/>