

Wind River Simics

目次

製品ライフサイクル全体を最適化… 2	ターゲットRTOSサポート …… 4
プラットフォームエンジニアリング… 2	機能 …… 5
アプリケーション開発 …… 2	モデリング …… 5
システム統合とテスト… 3	ホストサポート …… 5
プロジェクト管理 …… 3	ターゲットサポート …… 5
迅速かつ効率的なモデル化 …… 4	ターゲットデバイス …… 5
特徴と機能 …… 4	ターゲットCPUアーキテクチャ… 5
スケーラビリティ …… 4	SoCファミリ …… 6
コネクティビティ/インタフェース… 4	商用ボード仮想プラットフォーム… 6
ネットワークとバス… 4	

Wind River Simics は、製品チームに、物理的なシステムハードウェアをワークステーションや PC で稼働する Simics 仮想プラットフォームに置き換えるという開発手法の採用を可能にします。この仮想プラットフォームは、物理ハードウェアで

実行するものと同じバイナリソフトウェアを実行でき、十分な速度を提供するため、ソフトウェア開発やテストで使う物理ハードウェアの代替として使用することができます。Simics 仮想プラットフォームはユニークです。高速かつ正確なため、ハイパー

バイザからアプリケーションまでの完全なソフトウェアスタックを実行でき、ソフトウェアの反復実行、仮想ターゲットハードウェアの完全な視覚化と制御、真の逆実行が保証されます。Simics により、企業は製品開発ライフサイクル全体にわたって技術リスク、スケジュールリスク、開発コストを削減しながら、製品の機能と品質を向上させることができるため、競争上の優位性をさらに高めることが可能になります。

Simics 仮想プラットフォームは、実際の物理ハードウェアと同じ複雑さやシンプルさを実現できます。マルチコアプロセッサ、複数のプロセッサボード、複数ボードのシステムなども含めることが可能です。そのため、部品ごとのデバッグだけではなく、システム全体としてデバッグできます。Simics 仮想プラットフォームでは、混合エンディアンアーキテクチャを含め、異なるターゲットハードウェアアーキテクチャを混在させることができます。Simics 仮想プラットフォームは、利用可能なターゲット数が不十分、プロトタイプハードウェアが不安定、ハードウェアの入手遅延など、製品開発に物理ハードウェアを使用することで付随する問題を排除します。

Simics は、真の反復型開発をサポートし、実践的なシステムアーキテクチャ調査ができ、ハードウェアの入手可能性に左右されずにソフトウェア開発を開始できるため、プロジェクトのワークフローを改善します。開発プロセス全体にわたる一貫したプラットフォームとして Simics を活用することで、製品ライフサイクル全体の改善が可能となり、市場投入までの時間が短縮され、顧客による採用が加速されます。全体的な費用が減少し、市場に投入される製品の品質が上がるため、顧客満足度も向上します。

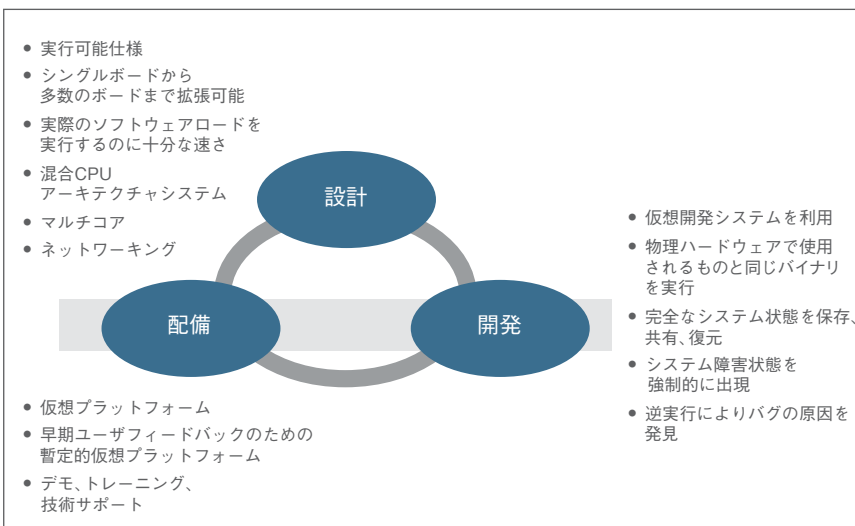


図1: Wind River Simicsは製品のライフサイクル全体をサポート

製品ライフサイクル全体を最適化

仮想プラットフォームを利用できるのは、ハードウェア設計者、ハードウェアデザイナー、ソフトウェアデザイナー、ソフトウェアテスト技術者、システムインテグレータ、システムテスト技術者で構成される製品開発チーム全体です。また、技術サポート用に顧客ごとに固有のハードウェアを維持しなければならない場合に、仮想プラットフォームを利用すると、サポート活動のコストを低減できます。

さらに、仮想プラットフォームを使用して、販売・マーケティングチームが新しい製品機能のデモを行ったり、顧客がトレーニング目的で活用したりすることも可能です。

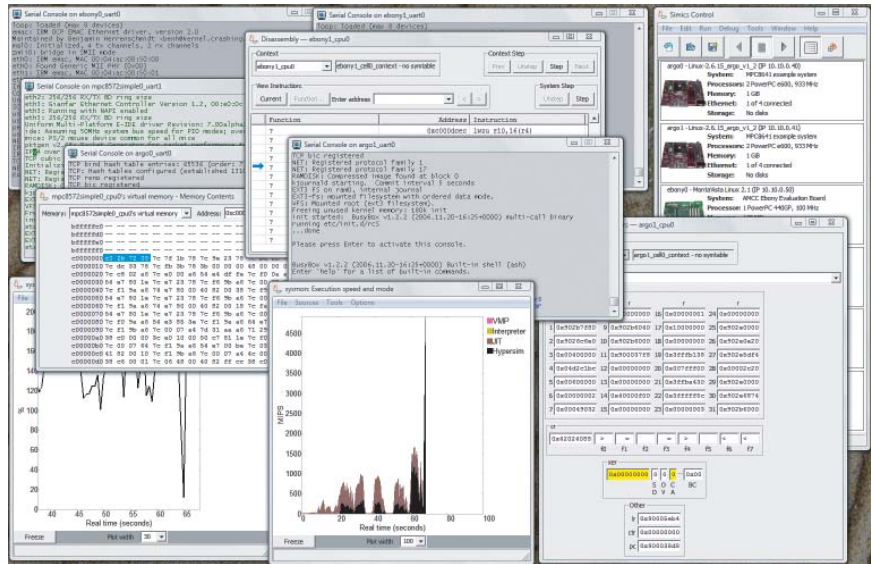


図2: Wind River Simicsを使ったソフトウェアのデバッグ

Simicsでできること

システムアーキテクチャの設計

- 文書から離れて「実行可能な仕様」を作成
- 設計の決定: CPU数は? DSPとGPPのどちらにするか? キャッシュサイズは? どのソフトウェア最適化を利用するか?

ソフトウェアの開発

- 利用可能なターゲットハードウェア数が不十分であることによるスケジューリング上の問題を排除
- 「不可能な」デバッグを実行
- ビット単位で正確なシステムのスナップショットを別の開発者に送信し、共同作業を実施
- あらゆるバグが再現可能
- ビッグバン統合を排除: 早期に開始し徐々に統合
- ハードウェアのコーナーケースをテストすることによる品質向上
- 異種OS環境やハードウェアアーキテクチャにわたりソフトウェアアプリケーションを開発

仮想プラットフォームの配備

- 仮想プラットフォームの配備により現場チームをサポート
- ラップトップでシステムをデモンストレーション
- 新しい顧客のコンフィギュレーションを容易かつ安価にサポート

プラットフォームエンジニアリング

プラットフォームエンジニアリングチームは、ハードウェア設計者、ハードウェアデザイナー、ファームウェアプログラミングを行うソフトウェア開発者で構成されます。プラットフォームエンジニアリングチームは、製品のハードウェアアーキテクチャ、システムデザイン、基盤ソフトウェアの設計と作成を行います。

従来、アーキテクチャの段階で開発ノートが使われてきたことは対照的に、仮想プラットフォームでは、以下のような技術的な問題の実践的解決が可能になります。

- マルチコアとアクセラレータのどちらを使うか、または両方使用するか?
- GPPとDSPのどちらで機能を実行するか?
- メモリのサイズと速度はどうするか?

ファームウェア、ブートコード、オペレーティングシステム、デバイスドライバといったシステムの基盤ソフトウェアの作成では、以下のような状況で問題が生じます。

- ハードウェアがまだ入手できない、またはプロトタイプの信頼性が低い
- ハードウェア仕様が頻繁に変更される
- 重要なデバイスレジスタにアクセスできない

Simics 仮想プラットフォームはハードウェアデザイン設計とともに進展し、開発者は物理ハードウェアを利用することなく生産性を維持することが可能になります。仮想プラットフォームはすべてソフトウェアで構成可能なため、変更への対応は物理ハードウェアよりもはるかに速くなります。そのため、従来の開発方法に比べより早期に、アーキテクチャ調査と基盤ソフトウェアを完了させることができます。

アプリケーション開発

アプリケーション開発チームは、ソフトウェアの設計、作成、最適化、テストを実施します。アプリケーション開発では、以下のような状況で問題が生じます。

- アプリケーションの重要な部分の開発を別々のチームが行う
- 試作ハードウェア、デバッグツール、計器へのアクセスが制限されている
- 単一のアプリケーションをデバッグするために全システムへのアクセスが必要
- 全システムを実行したときにしか視認できないバグがある
- 以前見つかったバグの再現が容易でない
- マルチコアアプリケーションを開発中である

Simics によるアプリケーション開発は、50以上のボードと700のプロセッサで構成され、マルチコアシステム、異種設計、マルチOS、ネットワーク分散システムなどを含むシミュレートされたシステムにも対応します。Simicsは、ハードウェアと時間の完全な制御、一見不可能に思われるデバッグシナリオのサポート、完全なrun-to-run再現性を実現します。これにより、開発者やアプリケーション開発チームの各メンバーは、システムスナップショットを共有し、物理ターゲットハードウェアで使い慣れたコンパイル/リンク/デバッグツールを使って作業できるため、より効率的に仕事を進められます。

Simics 仮想プラットフォームでは、以下により、バグの検出、バグの発見、再現、隔離、修正がより容易になります。

- 物理ターゲットで動作するものと同じバイナリを実行
- Simicsスナップショットのフリーズ、保存、復元

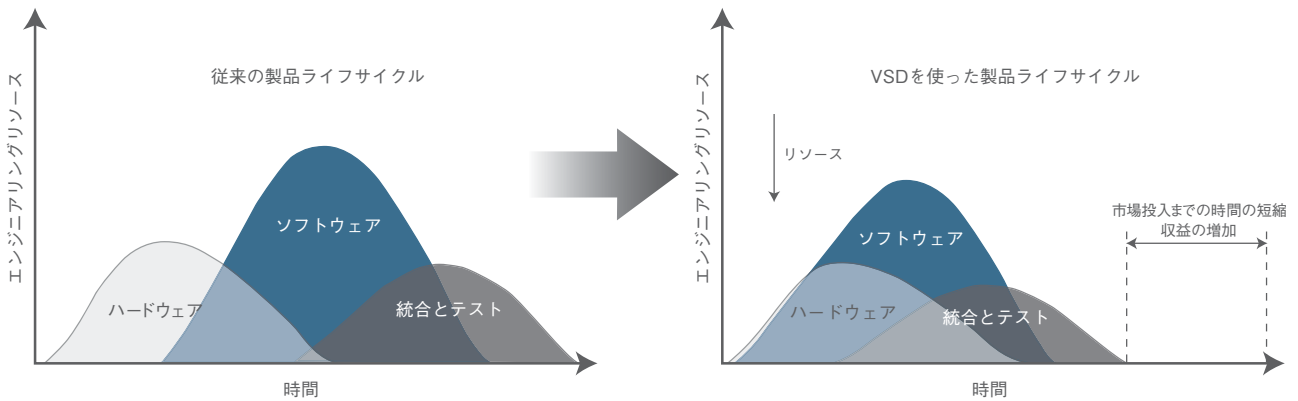


図3: Wind River Simicsは従来の製品開発に比べて市場投入までの時間を短縮

- 物理ハードウェアでは不可視であっても、全システムの個々のデバイス、レジスタ、通信インターフェース（イーサネット、仮想マシン環境、RapidIO、PCI、Expressなど）の完全な視認性と制御を活用可能
- レジスタの完全な可観測性やブレークポイント機能を使いながら、物理ハードウェアと同様のソフトウェアの順実行、またはSimics独自の逆実行を実施
- 完全なシステムデバッグのために、通常のソフトウェアパラメータまたは任意のハードウェアパラメータ（時間、命令、デバイス、メモリアクセスなど）でブレーク
- ソースやバイナリの変更を行わず、ハードウェアツール（JTAGなど）も必要とせず、トレース、コードカバレッジ、プロファイル、その他の情報を収集

システム統合とテスト

システム統合とテストは、次のような状況で、製品を問題なく納品することへのリスクを生じます。

- ハードウェアに遅れが生じているが納期の変更はない
- ターゲットハードウェアがないためにテストを完了できない
- ハードウェアエラーによりテストが中断し、再始動や再実行に時間がかかる
- 物理的にハードウェアを壊さないとシステム障害を強制的に再現できない
- run-to-runでバグを再現できない

仮想プラットフォームを使用すると、開発プロセスのかなり早い段階でシステム統合を開始することができます。その場合の「統合」は、ガントチャートで目立つ単一のタスクから、システムソフトウェアが入手可能になるにしたがって進展する比較的低い活動へと移行します。この漸進的な統合方法は、従来の方法よりもはるかに早期にシステムの問題を識別できるため、その解決も早期に迅速に行うことができ、他の方法よりも著しくコストを下げる事が可能になります。

Simicsでは、各エンジニアが開発、テス

ト、または統合タスク用の仮想ハードウェアを自分専用で使えます。Simics 仮想プラットフォームは、完全なシステム可観測性と、いつでもシステムを停止し検査できる機能を併せ持っているため、エンジニアは、破損したバケットへの応答、範囲外の読み込み、ディスクの不良セクター、推測されるハードウェア障害など、ハードウェアやI/Oの故障状態について包括的なテストを実行できます。このテスト中にバグが検出された場合、バグの確実な複製とそれに続く修正を行うためにテストエンジニアがすべきことは、システムをフリーズさせて、チェックポイントファイルをソフトウェアエンジニアリングチームに送信するだけです。

プロジェクト管理

プロジェクト管理者は、製品の完成と納品を確実にするために、プログラムの高レベルの側面に注目します。

- 設計審査
- ハードウェア/ソフトウェアチームの協働と意思疎通
- 品質向上
- 納期
- リスク軽減

従来の開発方法では、ハードウェアチームが次のプロジェクトに移行するのはプラットフォームの引き渡し次第になりますが、技術文書が100%完璧または正確であることはまれなため、スケジュールの遅延、機能の損失、手直しといった結果になる可能性があります。エンジニアは「その場に到達してみなければ、何が本当なのか実際にはよくわからない」ことがよくあるため、重大な設計問題が何度も行ったり来たりしうるので。

これとはきわめて対照的に、Simics を使えば、ハードウェア、ソフトウェア、システムの各エンジニア相互間の意思疎通と協働に改善がもたらされます。なぜなら、仮想プラットフォームは、ハードウェアの役割を果たしソフトウェアを実行できるもの

ですが、これを使用して製品仕様を決定したり、潜在的なシステムの問題を、別の方法で対処するよりもかなり早い段階で発見し解決することが可能になるからです。仮想プラットフォームがあれば、ハードウェアとソフトウェアの開発者は並行して作業を行い、「適切に対処するにはタイミングが遅すぎる」事態となるよりもずっと早いタイミングで問題を識別し総体的に解決できるようになります。実製品が入手可能になるより相当前のタイミングで、仮想モデルによる製品のデモ、議論、修正を実施できると、設計審査はまったく新たな様相を示します。

Simics 仮想プラットフォームは、お客様のターゲットハードウェアに取って代わり、ターゲットソフトウェアスタックすべてを修正なしに実行できます。さらに、ターゲットの構築とデバッグで使用したものと同一ソフトウェアツールを、仮想プラットフォームで継続して使用できます。

Simics は階層的にモデルを構築します。どの仮想プラットフォームでも、その最下層には、マイクロプロセッサ、メモリ、ネットワークインターフェース、バスコントローラなどの物理ターゲットハードウェアの個々のモデルがあります。これらの低いレイヤーのデバイスモデルがボードにまとめられ、ボードはシャーシに、シャーシはラックに、ラックはネットワーク接続されたサブシステムにまとめられていきます。このモジュール方式により、新しいシステムの作成は確実に簡単明瞭になります。

モデルは、ウインドリバーの汎用品の利用、または、ウインドリバー、お客様自身、ウインドリバーのエコシステム パートナー企業による開発も可能です。

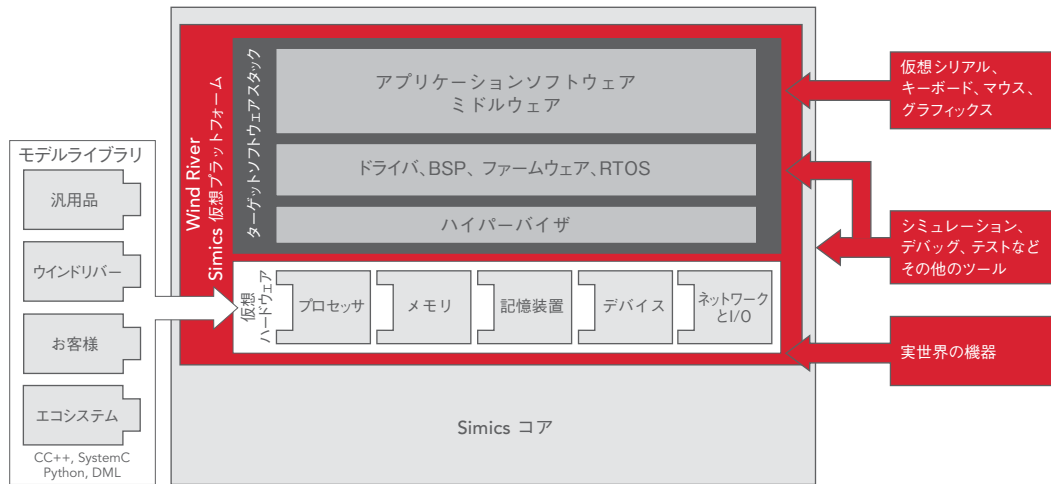


図4: Wind River Simicsブロック図

迅速かつ効率的なモデル化

Simics は、1つのコンポーネントはある言語で作成され、もう1つのコンポーネントは別の言語で作成されているといった、言語の混在するモデルをサポートします。これには、SystemC や TLM 2.0 などの標準的な言語や、そのほかに C、C++、Python、DML (Wind River Simics で利用可能なモデル化ツール) などの言語のサポートが含まれます。DML は、デバイス状態検査、チェックポイント、逆実行などの機能を自動的にサポートする専用の C ベースの構文により、デバイスモデルの効率的な作成を実現できるように調整されています。また Simics を使えば、DML で、他のサポートされるモデル化言語の中でその言語に応じて、または IP-XACT レジスタ定義ファイルを直接インポートすることによって、デバイスのプログラミングレジスタを直接定義することが簡単になります。

Simics では、Python スクリプトを使用して個々のモデルを結びつけることにより、完全な仮想プラットフォームを組み立てられます。この方法は、より困難なモデル構築タスクをシステムコンフィギュレーションタスクから切り離します。また、仮想システムのバリエーションや組み合わせを、迅速かつ容易に作成して管理することを可能にします。

特徴と機能

Simics は、ソフトウェアとシステム開発者のニーズに合ったユニークな特徴と機能の組み合わせを提供します。開発者を満足させるのに十分な正確さと速度を保ちながら、混合 CPU アーキテクチャや OS、マルチコアプロセッサ、ネットワーク接続されたプラットフォームから成る複雑で大規模なシステムモデル上で、製品版のフルソフトウェアスタックを実行できるのは Simics だけです。

スケーラビリティ

- SoC (システムオンチップ)
- シングルボード
- マルチボード、マルチコア、マルチプロセッサ
- 混合アーキテクチャシステム
- 分散型およびラックベースのシステム
- ネットワーク接続システム
- 共有メモリおよびローカルメモリのシステム

コネクティビティ/インタフェース

- ウインドリバー
- Eclipse プラグイン
- コマンドライン、バッチ、Telnet を介したリモート
- GDB
- ケイデンス
- クリティカルブルー
- エニア
- フリースケール
- IBM ラショナル
- ポリコア・ソフトウェア
- 大部分の主要商用デバッグ

ネットワークとバス

- イーサネット、AFDX、ATM、シリアル
- MIL-STD-1553、ARINC 429、SpaceWire、IEEE 1394 (FireWire)
- RapidIO、PCI および PCI Express、USB、I2C
- 共有メモリ型ラック バックプレーン

ターゲット RTOS サポート

- VxWorks
- Wind River Linux
- Wind River Hypervisor
- エニア
- エクスプレス・ロジック
- グリーンヒルズ
- IBM AIX

- Linux
- マイクロソフト
- モンタビスタ
- NetBSD および FreeBSD
- QNX ソフトウェアシステムズ
- サン Solaris
- 社内製作および大部分の商用リアルタイム OS (RTOS)

Wind River Simics製品ファミリ

- **Wind River Simics Hindsight:** ユーザインタフェース、シミュレーションフレームワーク、デバッグ
- **Wind River Simics Model Builder:** 新規デバイスモデルとマシン コンフィギュレーション
- **Wind River Simics Extension Builder:** お客様によるカスタム機能 (プラグイン) を使った Simics の拡張や、追加ワークフローへの接続を可能にする Simics シミュレータとのオープンインタフェース
- **Wind River Simics Analyzer:** あまり複雑でないシステムだけでなく、大規模な分散システムや異種システム用ソフトウェアアプリケーションの機能の分析、コードカバレッジ、デバッグ
- **Wind River Simics Ethernet Networking:** 仮想世界と実世界のネットワーク接続性を提供
- **Wind River Simics Accelerator:** マルチプロセッサとマルチマシンのホストを利用した大規模なシミュレーション
- **Wind River Simics Virtual Platform:** 汎用ハードウェアボードからフルカスタムのシステムまで、物理ターゲットシステムのモデル

機能

- 完全な実行制御とデバッグ
- メモリ、時間、I/O にブレークポイントを設定
- あらゆるハードウェア状態を検査し修正
- コンポーネントとコンフィギュレーションの制御
- プロファイル、トレース、ログ (データ、命令、レジスタ)
- ノンインタラクティブ コードカバレッジを利用
- メモリとレジスタの検査と編集
- 速度の制御
- システムの順実行または逆実行
- キャッシュ分析
- スクリプトによるテストの自動化
- パフォーマンスと同期の制御
- ネットワークのシミュレーション
- ターゲット OS アウェアネスの利用
- Simics とモデルの完全な制御
- プロセスとスレッドのグラフィカルな全システム実行タイムライン
- ユーザ定義のモジュールを使った拡張

モデリング

- OSCI SystemC TLM-2
- SPIRIT コンソーシアムの IP-XACT インポートおよびエクスポート
- C、C++ および Python API
- DML (デバイスモデル化言語)
- パラメータ化された階層型システムモデル

ホストサポート

- Linux (x86、x86-64)
- Windows (x86)

ターゲットサポート

ウインドリバーは、ボード、デバイス、アーキテクチャの広範なライブラリを提供します。最新リストは、http://www.virtutech.com/products/simics_model_library.html でご確認ください。

ターゲットデバイス

- メモリとシステムコントローラ
- 割り込みおよび DMA コントローラ
- イーサネットコントローラ
- PCI および PCI Express
- シリアルポート
- USB デバイスとディスク
- SCSI コントローラとデバイス
- I2C コントローラとデバイス
- RapidIO コントローラとデバイス
- FireWire や SpaceWire 用など、さまざまな通信デバイス

ターゲット CPU アーキテクチャ

ARM アーキテクチャ

- StrongARM
- XScale
- ARM7 (v4)
- ARM9 (v5)
- ARM11 (v6)

インテルおよび AMD

- インテル 386
- インテル 486
- インテル Pentium
- インテル Pentium MMX
- インテル Pentium Pro
- インテル Pentium II
- インテル Pentium III
- インテル Pentium 4
- インテル Pentium 4E
- インテル Pentium M
- インテル Core
- インテル Core 2
- インテル Core i7
- インテル Xeon バリエーション
- AMD Athlon

- AMD Athlon 64
- AMD Opteron

MIPS アーキテクチャ

- MIPS 4K
- MIPS 5K
- PMC RM7000
- PMC E9000
- キャビウム cnMIPS 64
- RMI XLR MIPS64

Power アーキテクチャ

- フリースケール e300
- フリースケール e500 v1
- フリースケール e500 v2
- フリースケール e500 mc
- フリースケール e600
- フリースケール MPC603e
- フリースケール MPC750、MPC755 (“G3”)
- フリースケール MPC74xx (“G4”)
- IBM PowerPC 405
- IBM PowerPC 403
- IBM PowerPC 440
- IBM PowerPC 464FP
- IBM PowerPC 750 (FX/GX)
- IBM PowerPC 970、970MP
- IBM POWER6
- IBM Cell SPE

ルネサス

- H8S
- H8/300
- SuperH SH-4

SPARC アーキテクチャ

- SPARC-V8
- SPARC-V9
- Gaisler LEON2

テンシリカ

- Xtensa

テキサス・インスツルメンツ

- TMS320C64x DSP ジェネレーション
- TMS320C64x+ DSP ジェネレーション

SoCファミリ

フリースケール

- QorIQ P1/P2、P4ファミリ (P40x0 および P20x0 を含む)
- PowerQUICC II (MPC82xx)
- PowerQUICC II Pro (MPC83xx)
- PowerQUICC III (MPC85xx)
- MPC8641/D

IBM

- Cell BE

ルネサス

- SH7756
- H8S/2164、H8/3867

テキサス・インスツルメンツ

- TI C6414
- TI C6455

キャビウム

- OCTEON 38xx/58xx

商用ボード仮想プラットフォーム

- BAEシステムズ RAD750 3U/6U
- カーチス・ライト SVME-183
- カーチス・ライト CHAMP-AV6
- GEファナック VG5
- サン SPARCサーバ
- x86 PC
- さまざまな SBC開発ボード

ウインドリバーは組み込みソフトウェアとモバイルソフトウェアのリーディングカンパニーです。
企業がデバイスソフトウェアを、より早く高品質かつ低コスト、かつ高信頼性で開発、運用、管理することを可能にします。

WIND RIVER ウインドリバー株式会社

東京本社

〒150-0012 東京都渋谷区広尾 1-1-39 恵比寿プライムスクエアタワー
TEL.03-5778-6001 (代表)

大阪営業所

〒532-0011 大阪市淀川区西中島 7-5-25 新大阪ドイビル
TEL.06-6100-5760 (代表)

www.windriver.co.jp

© 2010 Wind River Systems, Inc. Wind River ロゴは、Wind River Systems, Inc. の商標です。Wind River、および VxWorks は、Wind River Systems, Inc. の登録商標です。記載されているその他の商標は、各所有者に帰属します。詳細：www.windriver.com/company/terms/trademark.html Rev.04/2010

■販売代理店