システムLSI設計における形式的手法への期待

1．システムLSIとは？
2．システムLSIの設計の流れ
3．形式的検証
4．上流設計と形式的手法
5．システムLSIの経済学
6．Quality, Reliability and Security

設計とは？

- 実現したい機能・性能を実現するために与えられた設計パラメータの値を決定する作業。

  設計パラメータとして何を準備するか？

  設計パラメータの指定の方法（記述言語）

  設計パラメータと機能や性能の関係の評価法

  （シミュレーション、評価・検証技術）

  設計パラメータの最適化法（合成，最適化）

設計パラメータの影響

詳細
複雑

概略
単純

細かな最適化
高品質の設計
設計時間の増大
設計コストの増大
設計者のスキル
プロセスへの依存性大

グローバルな最適化
大幅な設計改善
設計時間短縮
設計コスト低減
システム設計の視点
ソフトウェアとの関係

新しいパラメータは？

- システムアーキテクチャ
  - » プラットフォームのアーキテクチャ
  - » マルチプロセッサの相互結合
  - » メモリアーキテクチャ
- プロセッサのアーキテクチャ
  - » 命令セット、データ語長、メモリ構成など
- ソフトウェア
- 電源電圧
  - » 可変電源電圧プロセッサ
- 計算の精度

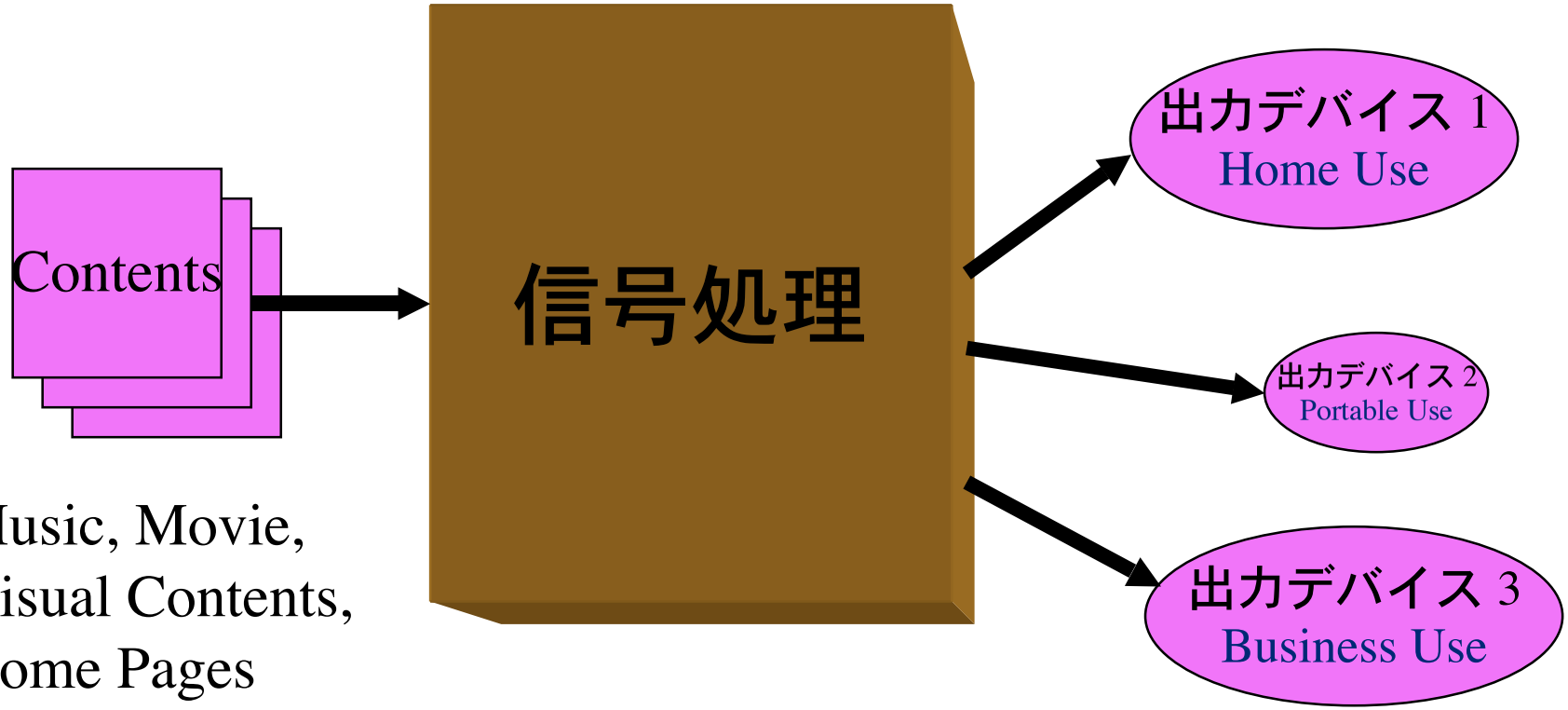
パラメータの５Ｗ１Ｈ

- なぜ最適化が必要か？
- 何をパラメータとするか？
- どのようにパラメータを最適化するか？
- いつパラメータを最適化するか？
- だれがどこでパラメータを最適化するか？

Why?

- コスト（面積）最小化
  - » 回路，プロセッサ，メモリ
- 性能最大化
  - » 回路，プロセッサ，メモリ，I/O
- 消費電力最小化
  - » 回路，プロセッサ，メモリ，I/O
- 信頼性や計算品質のチューン

Quality Driven Design



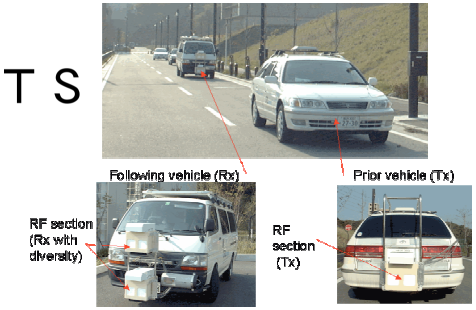
Music, Movie,
Visual Contents,
Home Pages
標準フォーマット

個別の品質

多様化する設計要求

- 多様なシステムのデジタル化により、デジタルシステムの設計が多様化している。
- システムの**出力の品質**に対する設計要求
  - システムの出力の品質を変更しないコスト・消費電力最小化が必要な設計

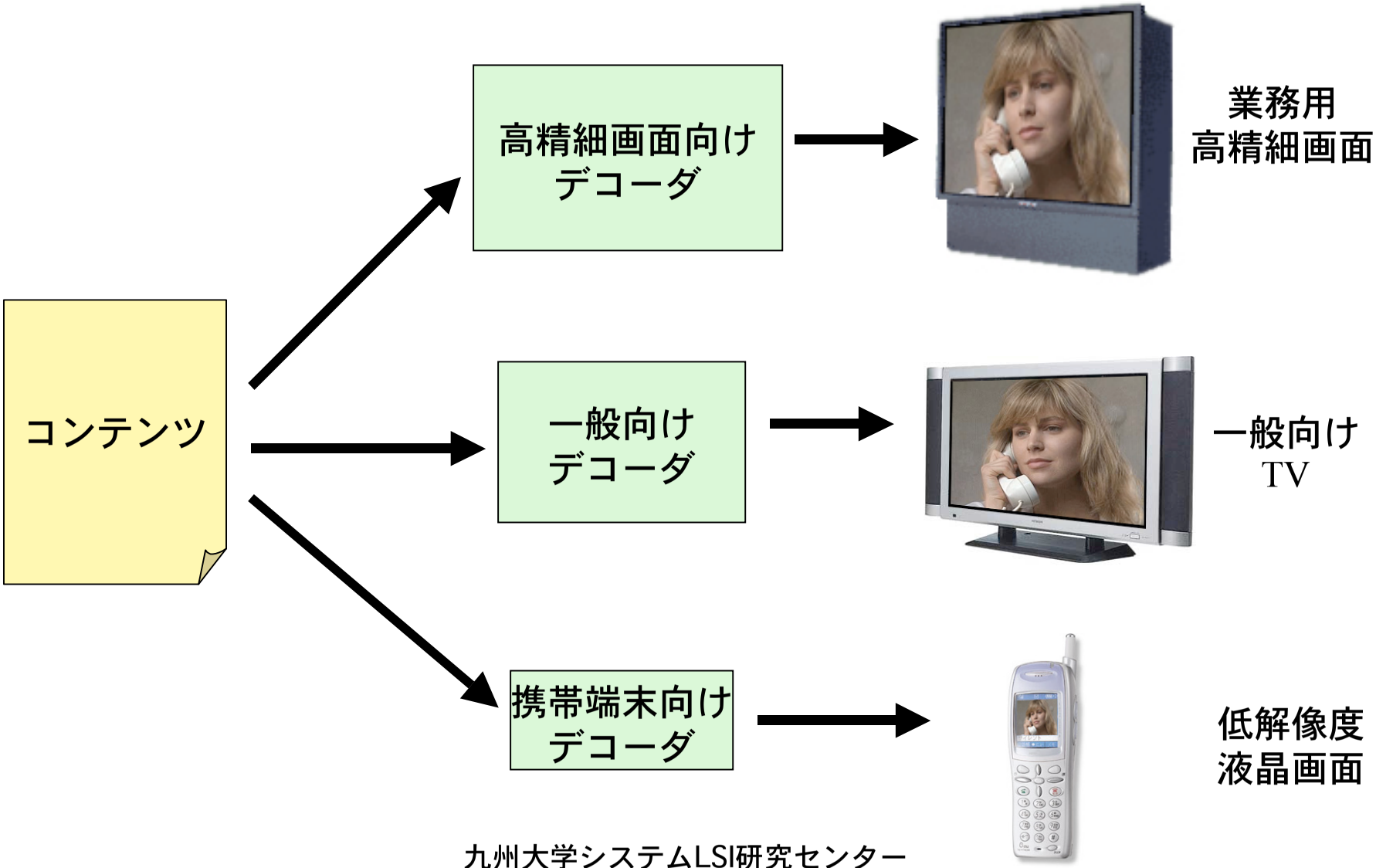
ＩＴＳ



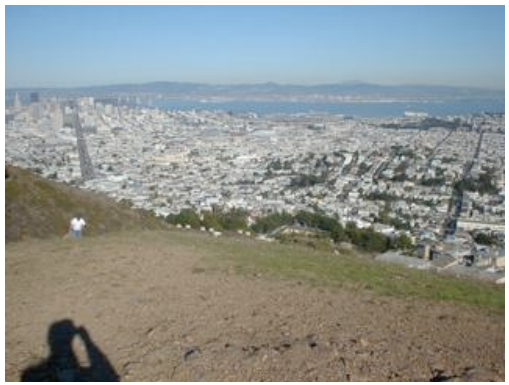
ＣＴスキャン

  - システムの出力の品質と消費電力・コストとのトレードオフを考慮した設計

動画デコーダの設計


コンテンツ
高精細画面向け
デコーダ
一般向け
デコーダ
携帯端末向け
デコーダ
業務用
高精細画面
一般向け
TV
低解像度
液晶画面

MPEG2のデコード結果



ソース画像

出力画像 b　マスク4bits
PSNR = 33.65 [dB]

出力画像 c　マスク8bits
PSNR = 29.80 [dB]

出力画像 d　マスク12bits
PSNR = 11.02 [dB]

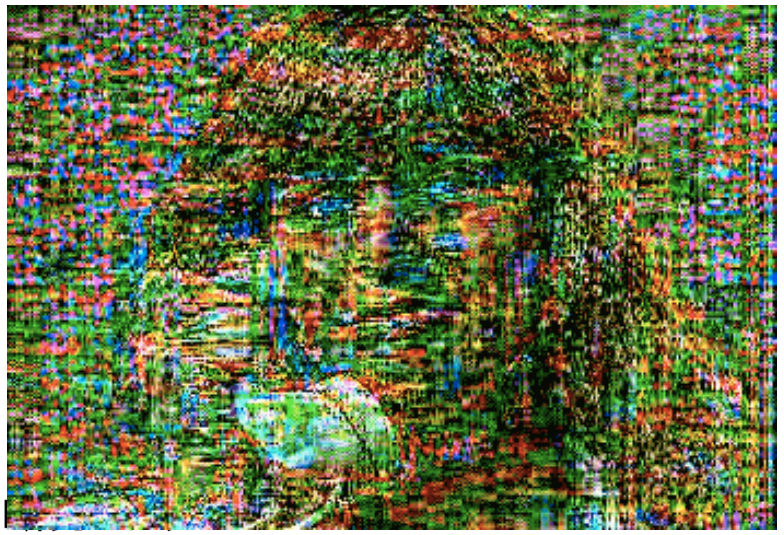
実験結果 (1)

16bit

12bit

8bit

4bit



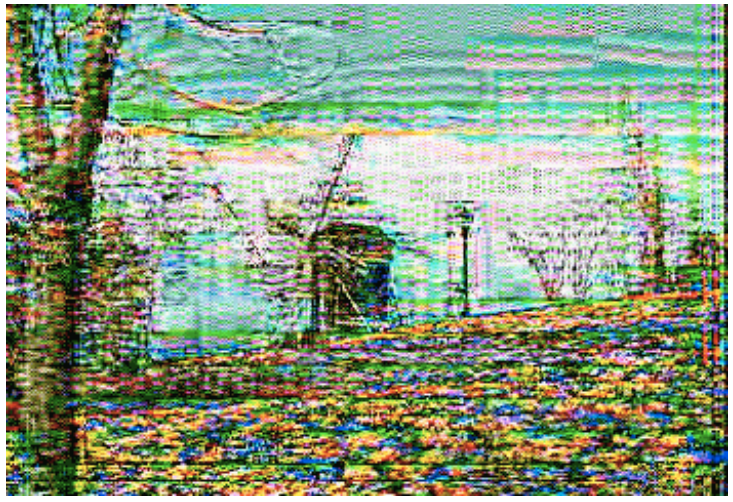
実験結果 (2)

16bit

12bit

8bit

4bit



実験結果 (3)

16bit

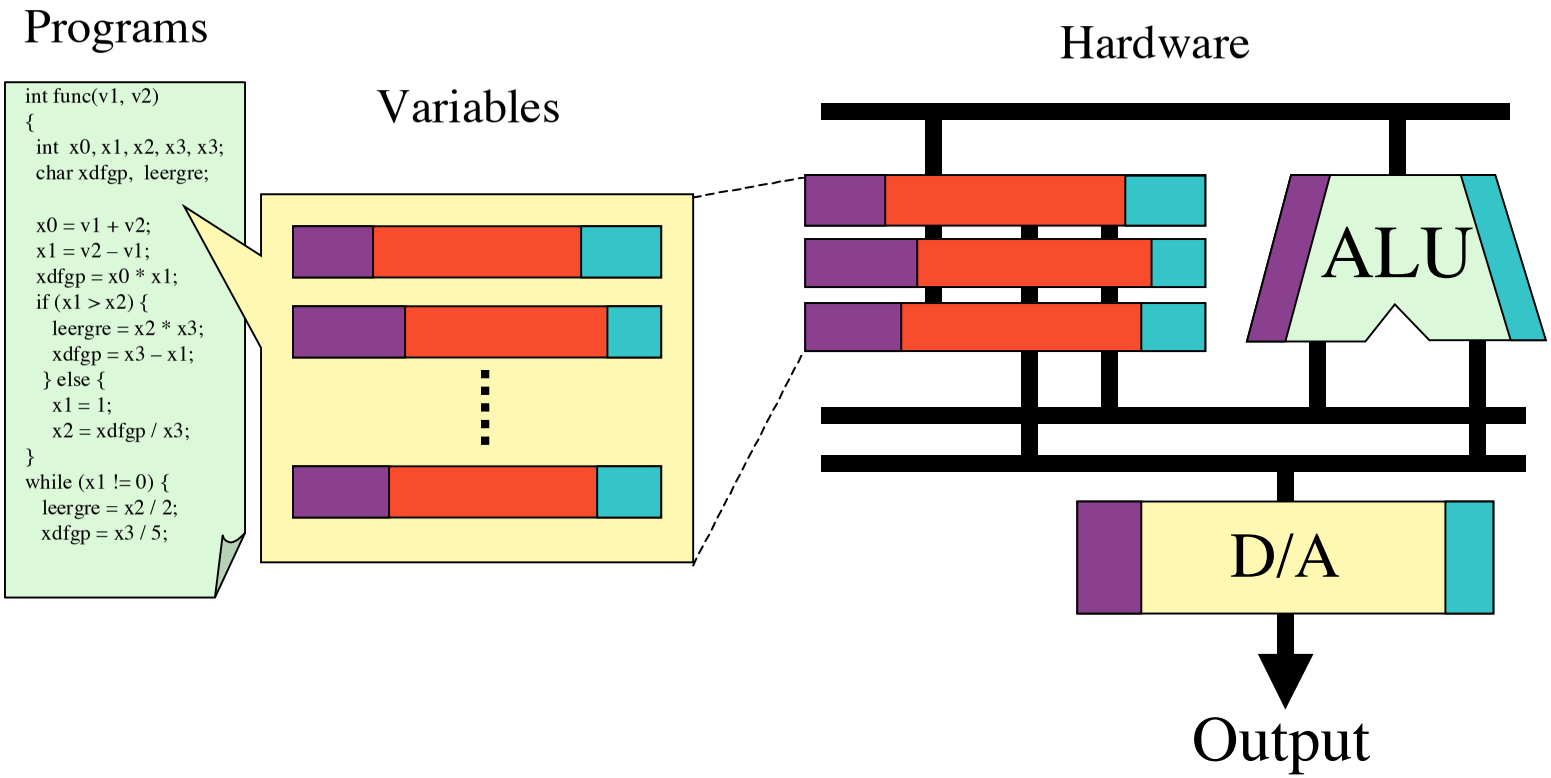
12bit

8bit

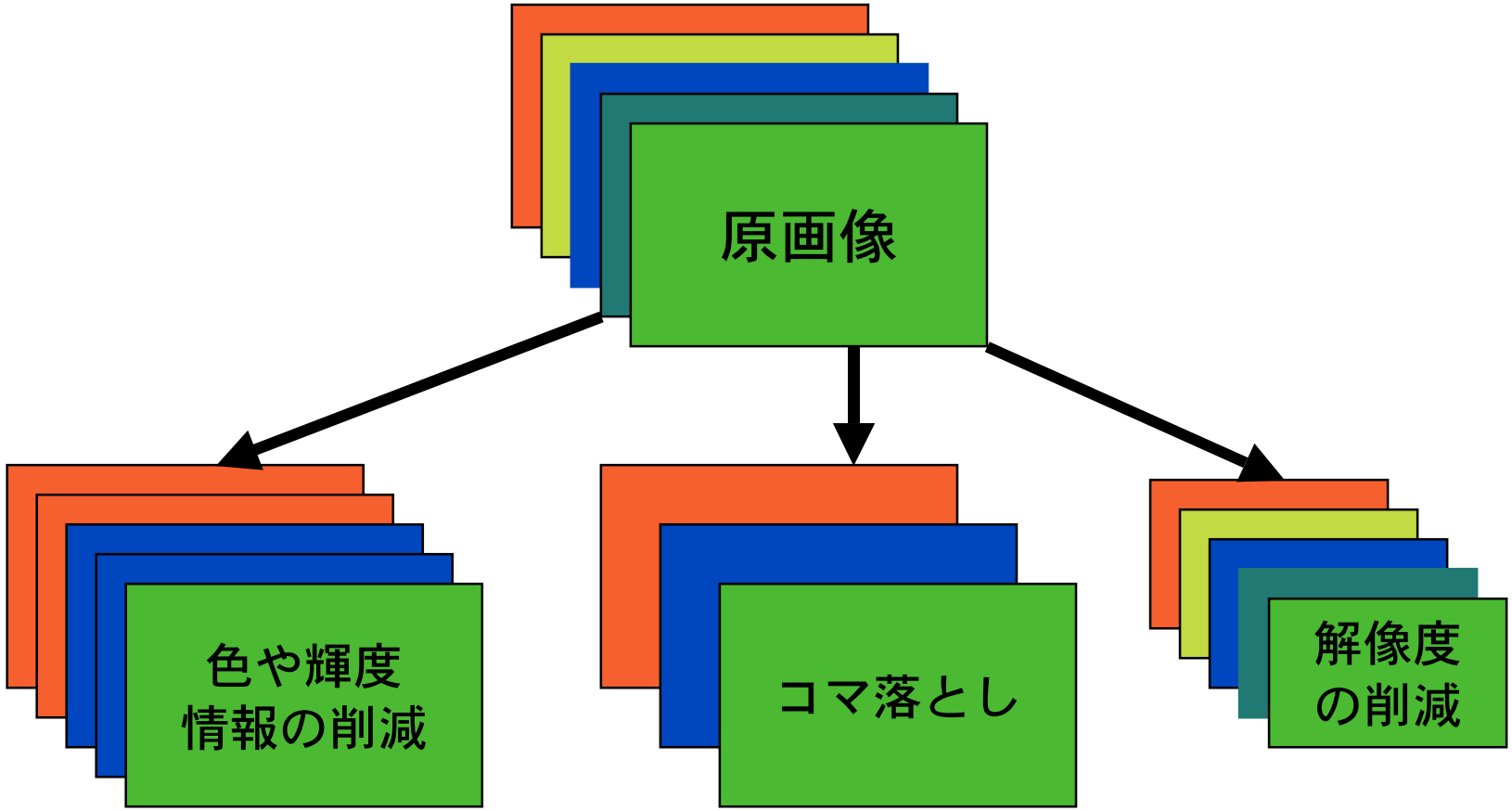
4bit

無駄な計算の省略

要求される出力精度を得るのに必要な
計算に必要最低限のデータパスを用意する

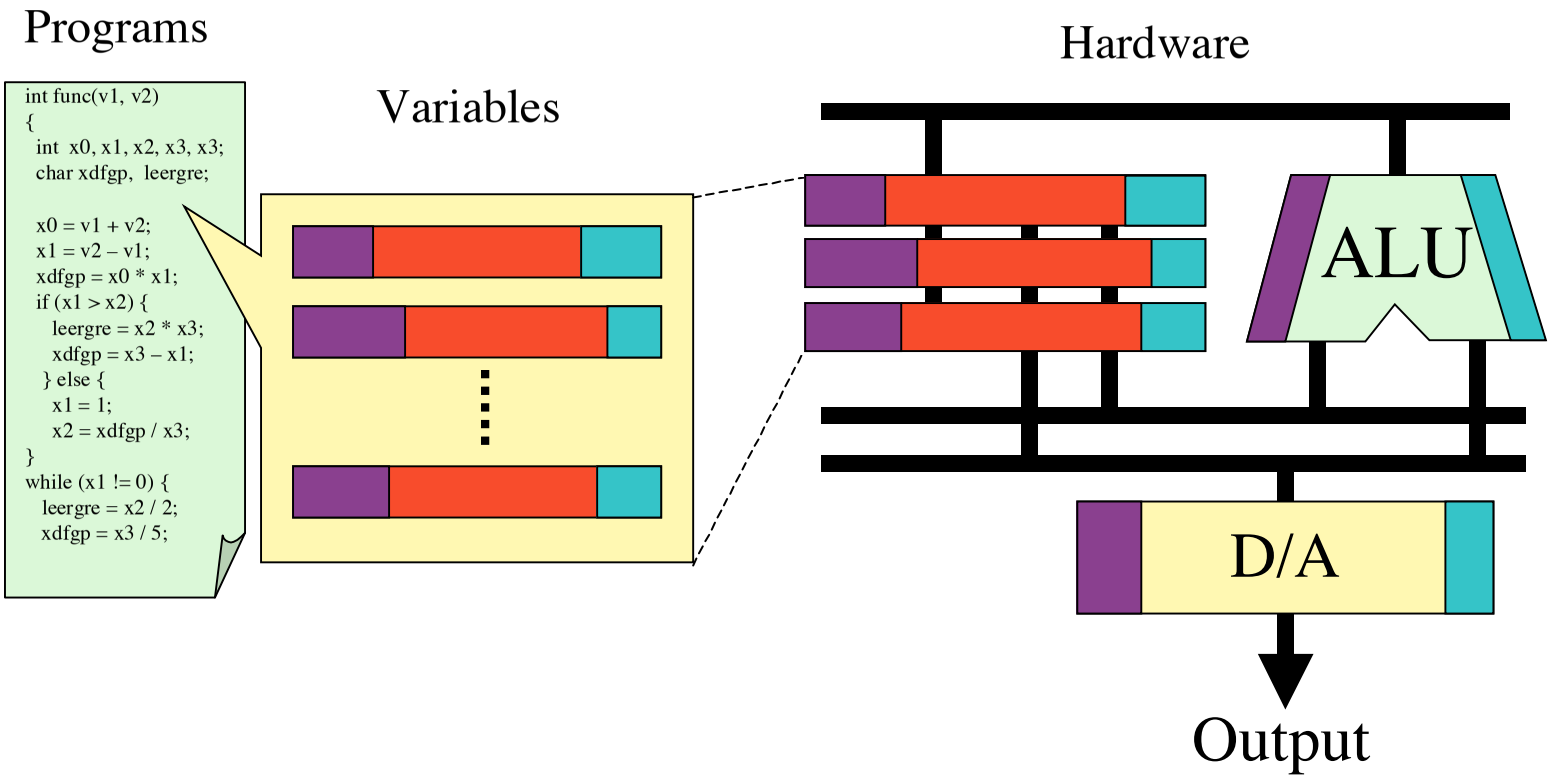


動画像に対する設計例



無駄な計算の省略

要求される出力精度を得るのに必要な
計算に必要最低限のデータパスを用意する



形式的手法への期待 1

- 要求される出力の品質から必要となる内部の計算精度を算出する技術.
- 要求される信頼性や安全性に対して, 必要となるパラメータ値を決定する技術.

What?

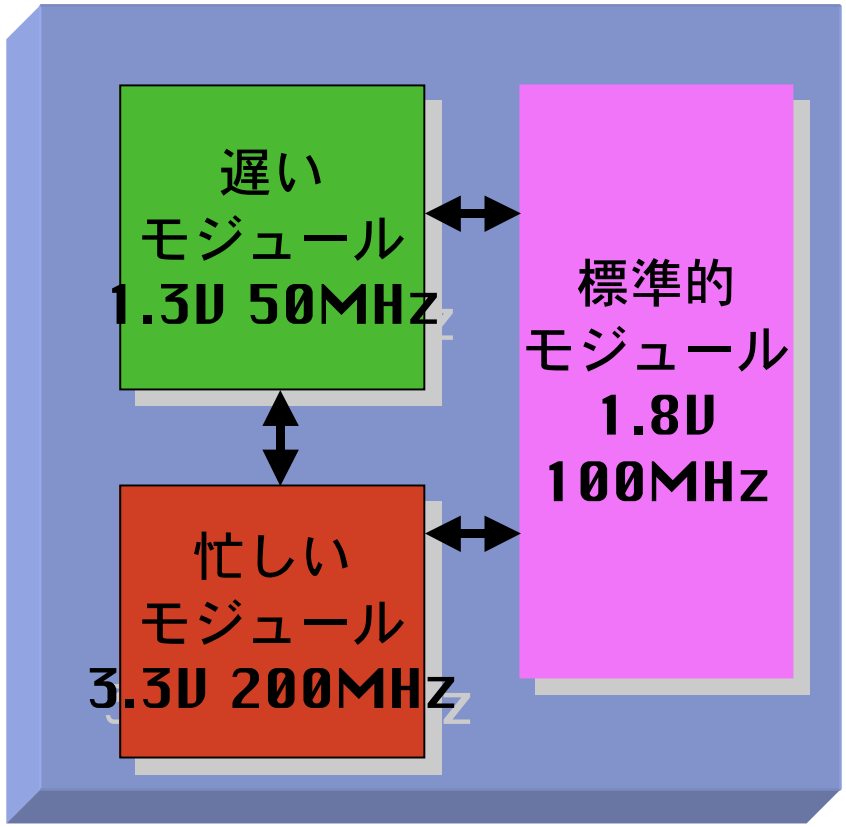
- 資源量（プロセッサ、メモリ、バス、演算器、レジスタなど）
- 資源同士の接続法
- アルゴリズム
- 計算精度，データ語長，データパス幅
- 基本命令・基本演算
- データの配置（特にメモリ上），メモリ構成
- 電源電圧，クロック周波数

新しいパラメータの例

- 可変電源電圧アーキテクチャ
  - » 電源電圧，クロック周波数
- ソフトコアプロセッサ
  - » データ語長，レジスタ数，メモリサイズ，基本命令，データ配置
- キャッシュメモリアーキテクチャ
  - » ラインサイズ，タグ比較アルゴリズム，予測アルゴリズム，データ配置

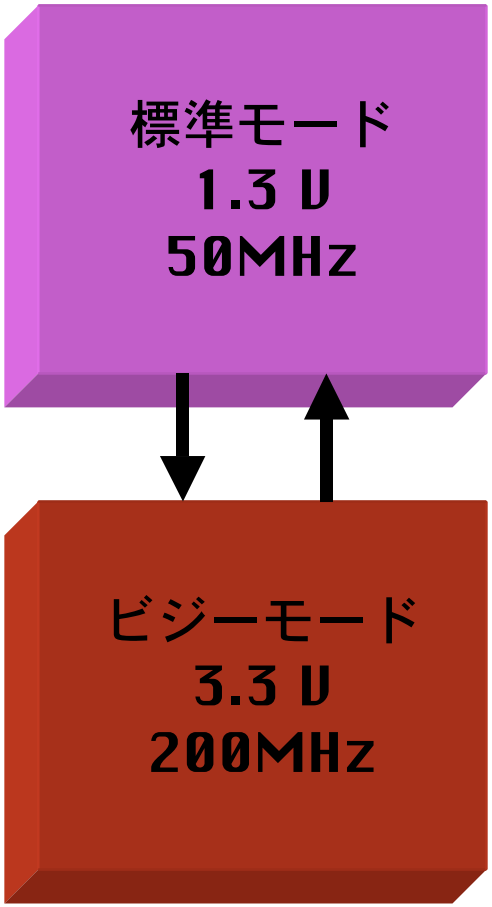
電源電圧の空間的最適化

- 各部分は必ずしも同じ忙しさではない。
- 暇な部分回路は低い電圧でゆっくり動かす。



電源電圧の時間的最適化

- 要求される性能は時間的に変化する。
- 忙しいときだけ高い電圧で高速処理する。
- 暇なときは低い電圧で低速処理する。
- 動物の血圧
- Transmeta社 etc.



ソフトコアプロセッサ

- パラメータ化されたプロセッサ
  - » 基本的な可変パラメータ
    - – データパス幅
    - – レジスタ数
    - – 命令セット
    - – データメモリ空間
- 論理合成，レイアウト合成の環境
- コンパイラ生成

  →カスタム化可能なコアプロセッサのIP
- ARM, Tensilicaなど

MPEG-2 Video Decoder

- Size of Program : 6275 lines (written in C)

Variable size analysis result :

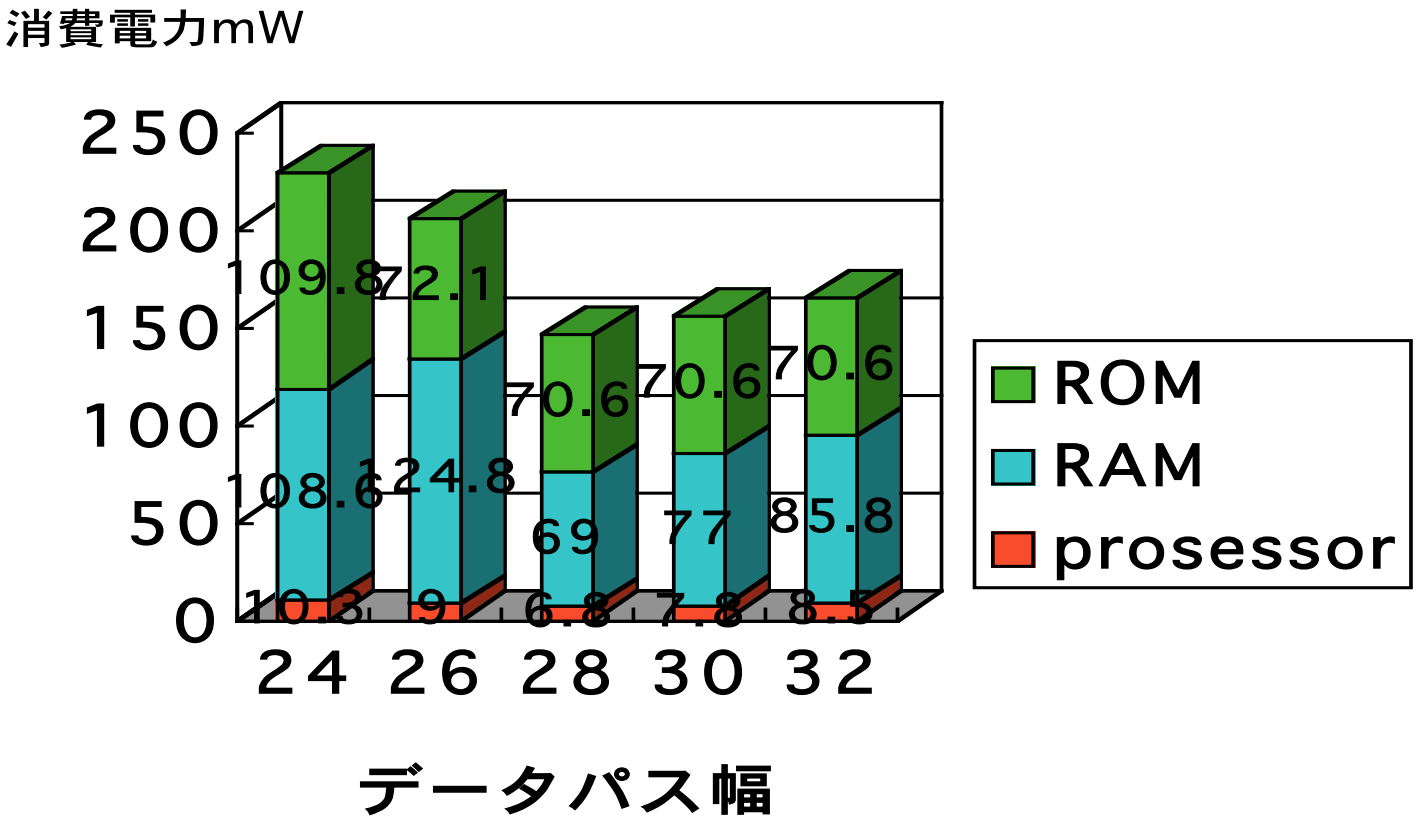
Bit Width	# of variables
1 bits	50
2 bits	17
3 bits	11
4 bits	11
10 bits	3
11 bits	6
12 bits	17
13 bits	0
14 bits	46
15 bits	2
16 bits	39

Bit Width	# of variables
17 bits	2
18 bits	3
19 bits	0
20 bits	6
26 bits	2
27 bits	4
28 bits	3
29 bits	3
30 bits	7
31 bits	0
32 bits	5

Bit Width	Arrays
1 bits	9 * 4
3 bits	10 * 1 20 * 1
4 bits	4 * 1

Variable Size Analysis Results / 32 bits Variables * 100 = 35%

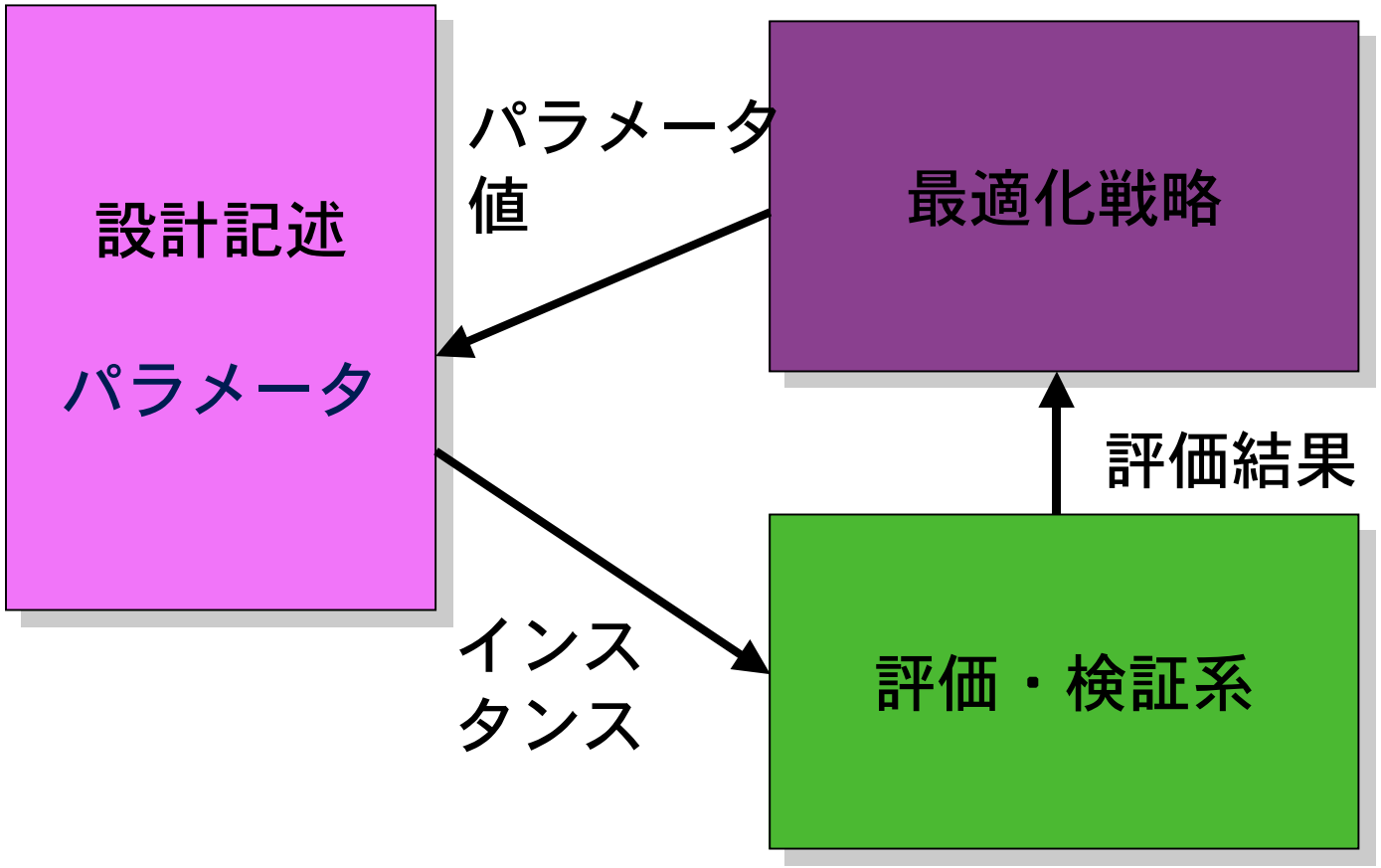
消費電力の削減効果 (MPEG2 decoder)



形式的手法への期待

- いろいろな設計パラメータに対する対応
  - » モデル化の方法
  - » 評価関数の柔軟な定義
    - – 機能以外の評価基準
    - – 性能，消費電力，計算品質，信頼性，安全性

How?



システム設計の記述への要求

- できるだけ上流で
  - » HWもSWも含んだシステムレベルで
- できるだけ正確に
  - » 曖昧さのない記述で厳密に
- できるだけ分かりやすく
  - » 読みやすく，書きやすく，検証しやすく
- どのようなパラメータも扱えるように
  - » いろいろなパラメータに対応できるように

形式的手法からの解決策への期待

記述言語の方向性

- できるだけ上流で
  - » システムレベルでの記述，要求仕様の記述
- できるだけ正確に
  - » 形式的な言語による記述
- できるだけ分かりやすく
  - » グラフィック言語
  - » 各種のシンタクス的工夫
  - » 検証・評価ツールとの連携
- どのようなパラメータも扱えるように
  - » 自由度を持った言語

When?

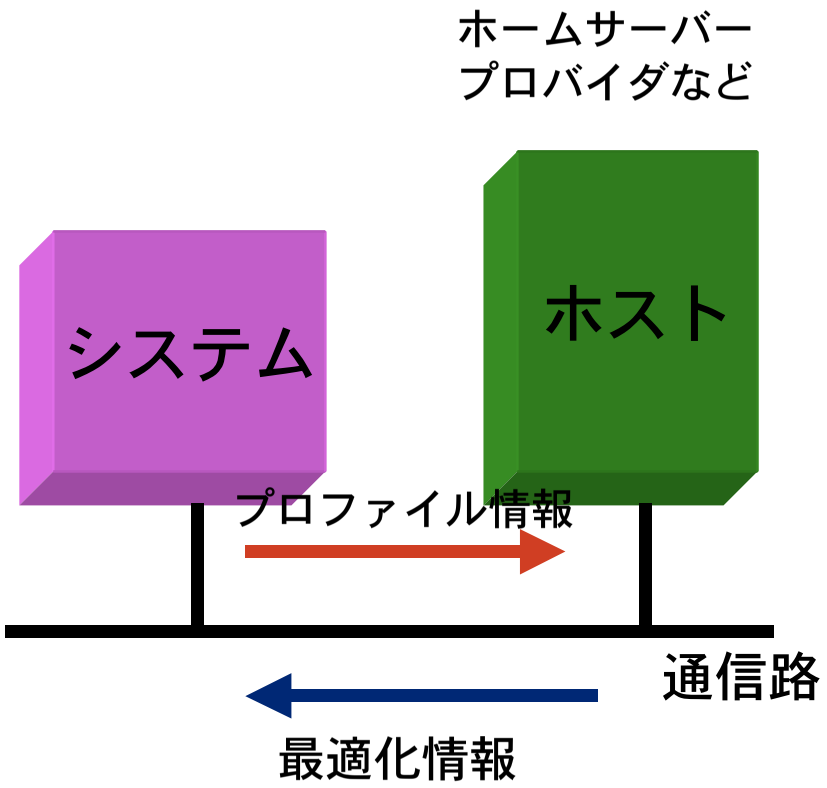
- 出荷前(Optimization in Design)
  - » チップ設計時
  - » チップ設計後
    - – プログラムROM
- 出荷後（ユーザに依存した最適化）
  - » 実行時(Optimization in Operation)
  - » 休止時(Optimization in Sleep)
    - – Sleep時（例えば充電中）に最適化

Who and Where?

- Who
  - » 設計者
  - » システムの利用者
  - » Optimization Service Provider
- Where
  - » 設計環境
  - » 対象システム内
  - » 最適化用ホスト

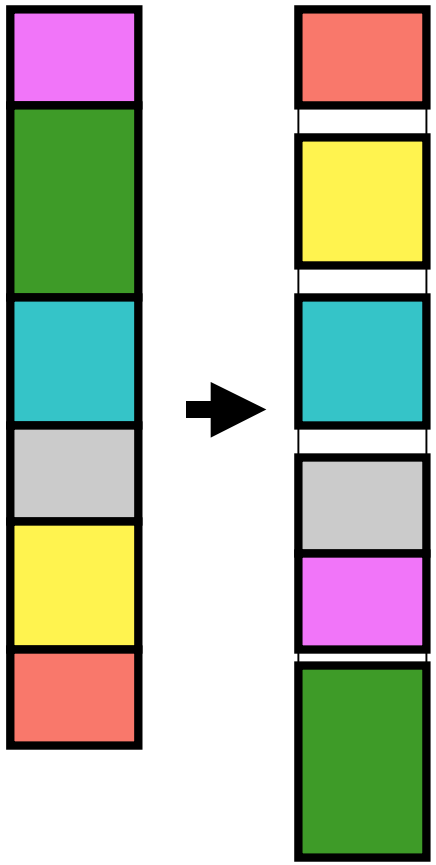
Optimization in Sleep

- 個人用携帯情報機器はほとんど休止している.
- システムの複雑化により，使われる機能は個人差が大きい.
- システム内に最適化機能を持たせるのはコストがかかる.
- 充電時などにプロファイルデータをホストに送り，ホスト側で最適化し，最適化情報をシステムに戻す.
- システムは，プロファイル収集機能のみでよい.
- 最適化はホストで時間とエネルギーをかけて行える.

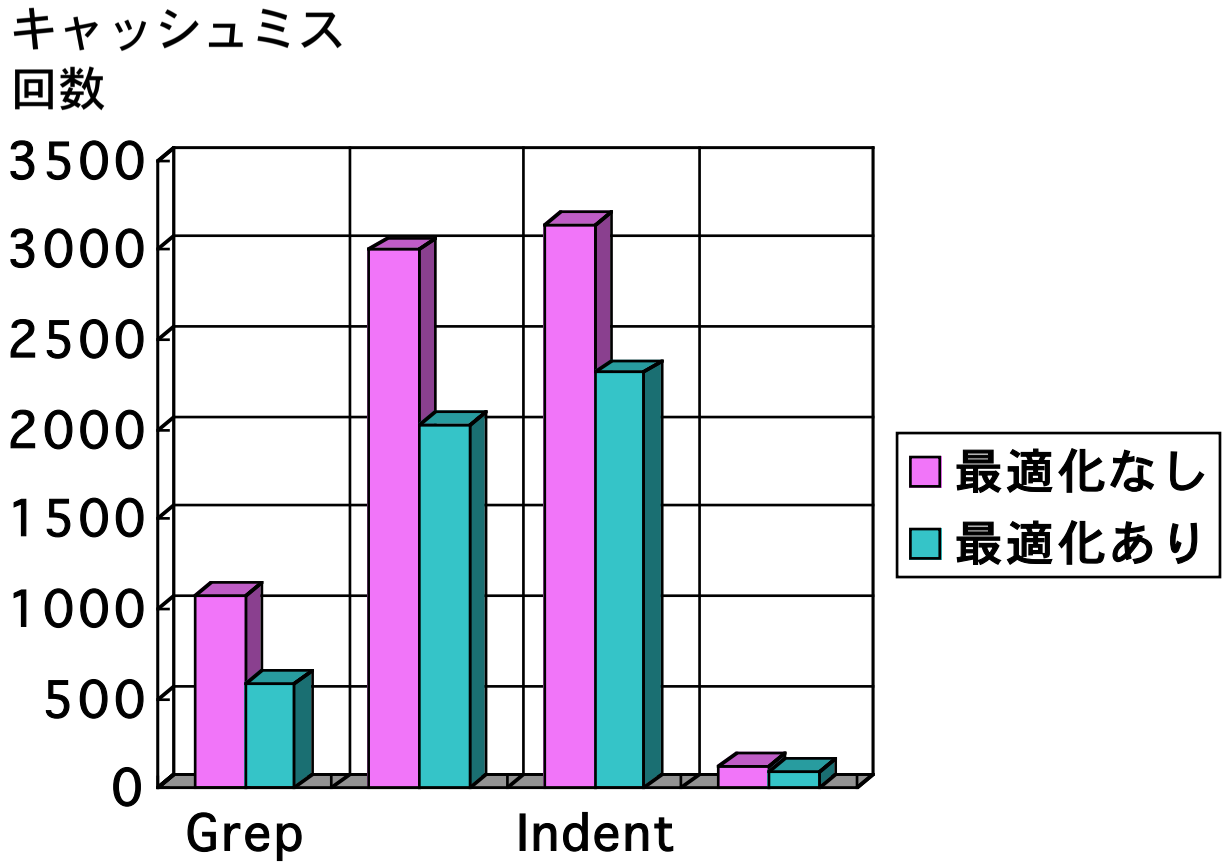


キャッシュミスの削減

- プログラムのプロファイルをとり、最も頻繁に実行される命令系列に対してキャッシュミスが少なくなるように主記憶上の基本ブロックの配置を最適化する。



主記憶配置の最適化の効果



Cache Size
1KB
Direct Mapping

Opt. in Sleepへの技術課題

- 可変部分（パラメータ）を何にするか？
- 可変部分をどう実現するか？
- プロファイルデータの集め方
  - » データ量
  - » 評価関数（性能，消費電力）に対する感度
  - » 観測のしやすさ
- ビジネスモデル
  - » Optimization Provider
  - » Home Server
  - » Security

形式的手法への期待３

- 動的最適化機構の安全性の保証
- パラメータ変更の安全性
  - 機能の維持
  - 性能の変化
  - 消費電力の変化

何をシステムLSIで作るか？

- 集積回路は出版と同じ
  - » 数が出ればコストが下がる．Million Sellerが稼ぐ．
- 従来のLSIは部品
  - » メモリはいろいろな所（パソコン，ゲーム機，家電製品など）で使われるので，同じものがたくさん売れる（年間数億から数十億個）
- システムLSIは製品の個数しか売れない．
  - » 世界の人口は６０億人
  - » 一人がたくさんシステムLSIを使ってもらうには？
- 生活必需品・社会基盤としての利用

半導体産業の転換期

- 部品産業からの脱却
  - » 大量生産の論理の破綻
  - » メモリ依存の破綻
  - » チップ数＝製品数
- 個人消費依存の限界
  - » コスト主導の市場競争
  - » PC、ゲーム機、携帯電話
  - » 世界の人口は６０億人
  - » 製品サイクルの短縮
  - » 過当競争と価格の下落

- 社会システムへの展開
  - システム構想からLSIへ
  - 大量のLSIのネットワーク
  - 数百個／人・年
- 社会基盤としての技術
  - 生命、財産、プライバシーが対象
  - 信頼性と安全性
  - 長期的な利用
  - 経済性
  - 社会資本としての蓄積

基本的な疑問

- 人の生命，財産，プライバシーにかかわるシステムを現在のパソコン，ゲーム機，携帯電話の技術で作ってよいのか？
  - » 失敗が許されないシステム
    - – 偽電子マネー
    - – プライバシ情報の漏洩
    - – 生命の危険を引き起こす事故

社会基盤としての情報技術

- Quality　　ー品質
- Reliability　　ー信頼性
- Security and Stability
  ー安全性と安定性

もちろんCostとPerformanceも重要

社会基盤は、社会資本である！

社会と技術

- 情報技術は社会構造や文化を変える.
- 情報技術によってどのような社会を築くか？

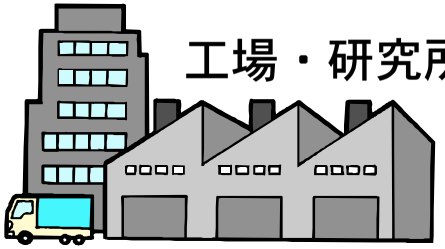
  公害や薬害のような社会問題を未然に防ぐ
  - » 使う人にわかりやすいシステム
  - » 一般人が原理を理解できるシステム
  - » 採用の可否を社会が決定できること

システムLSIの応用

コンピュータ

医療応用

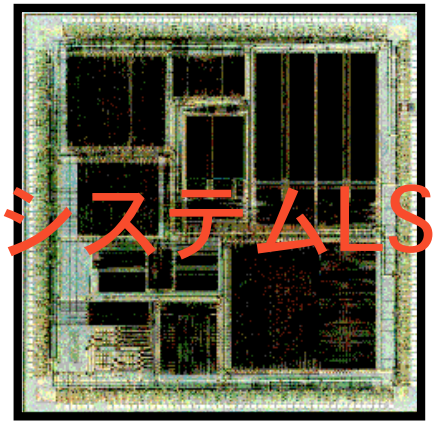
都市基盤



工場・研究所

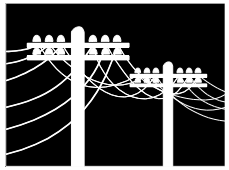
宇宙開発

航空機



システムLSI

家電製品

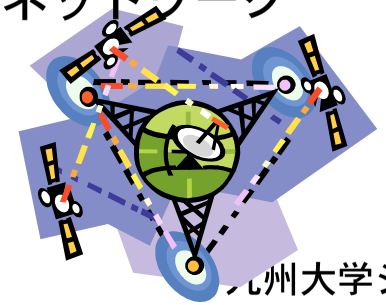


エネルギー産業

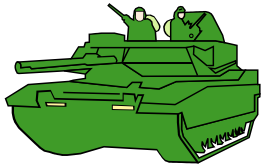
通信機器

自動車

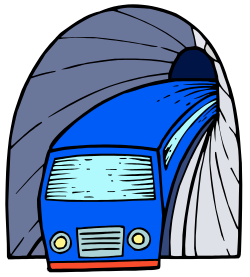
通信ネットワーク



経済・商取引



軍事技術



鉄道

システムLSI研究センターとは

発足：平成１３年４月

目的：高度情報化社会の基盤技術としてのシステムLSI技術の総合的研究.

学問体系の確立と社会の中でのこの技術の利用の方向を明確化.

21世紀の社会のデザインに要素技術の側面から指針.

特徴：九州大学の研究院制度の下での始めての学内共同利用センター.

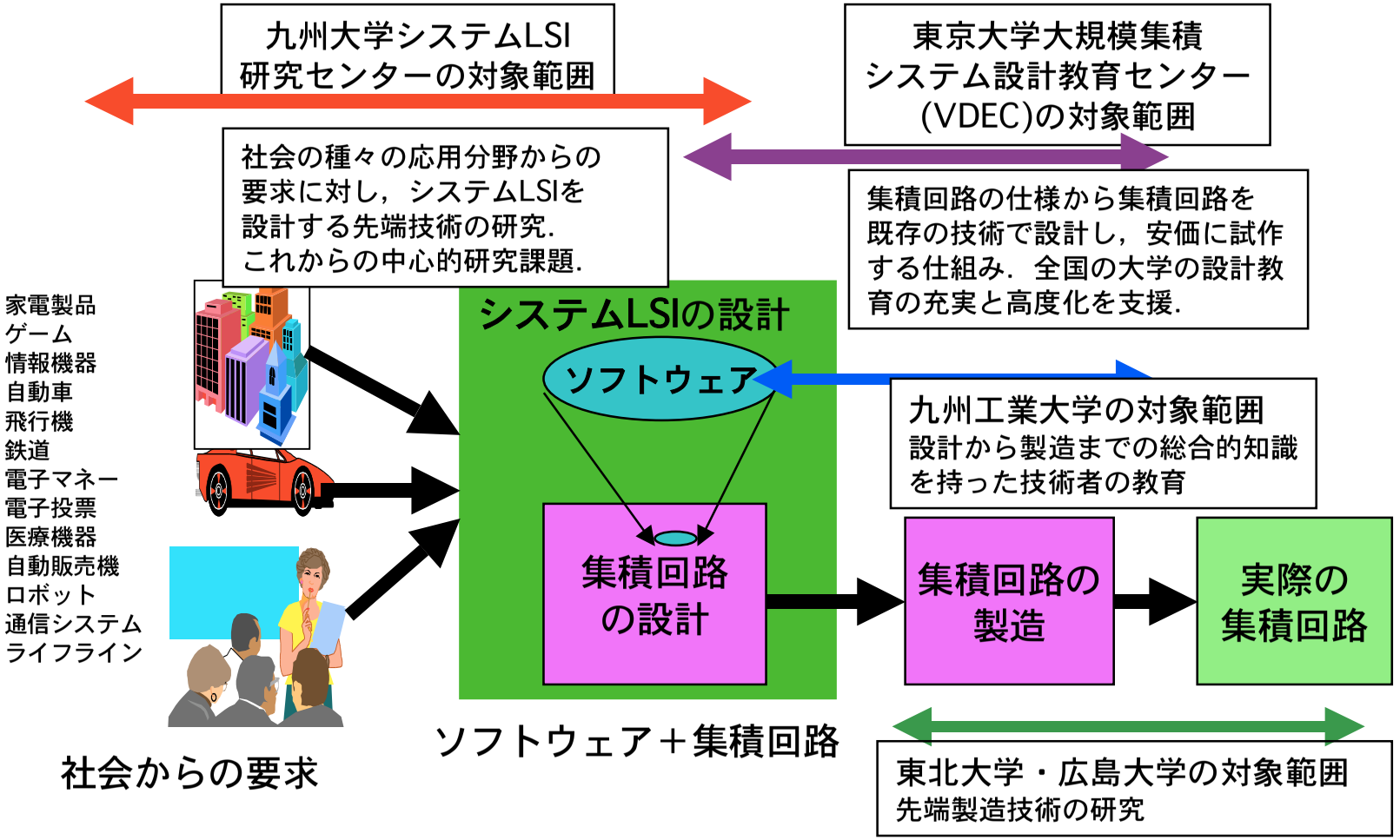
各研究院からの研究者による学際的研究プロジェクト.

産学官の共同プロジェクトの中核.

従来に無い機動的運営と新しい研究組織モデル.

大学院生の活用とベンチャーの育成.

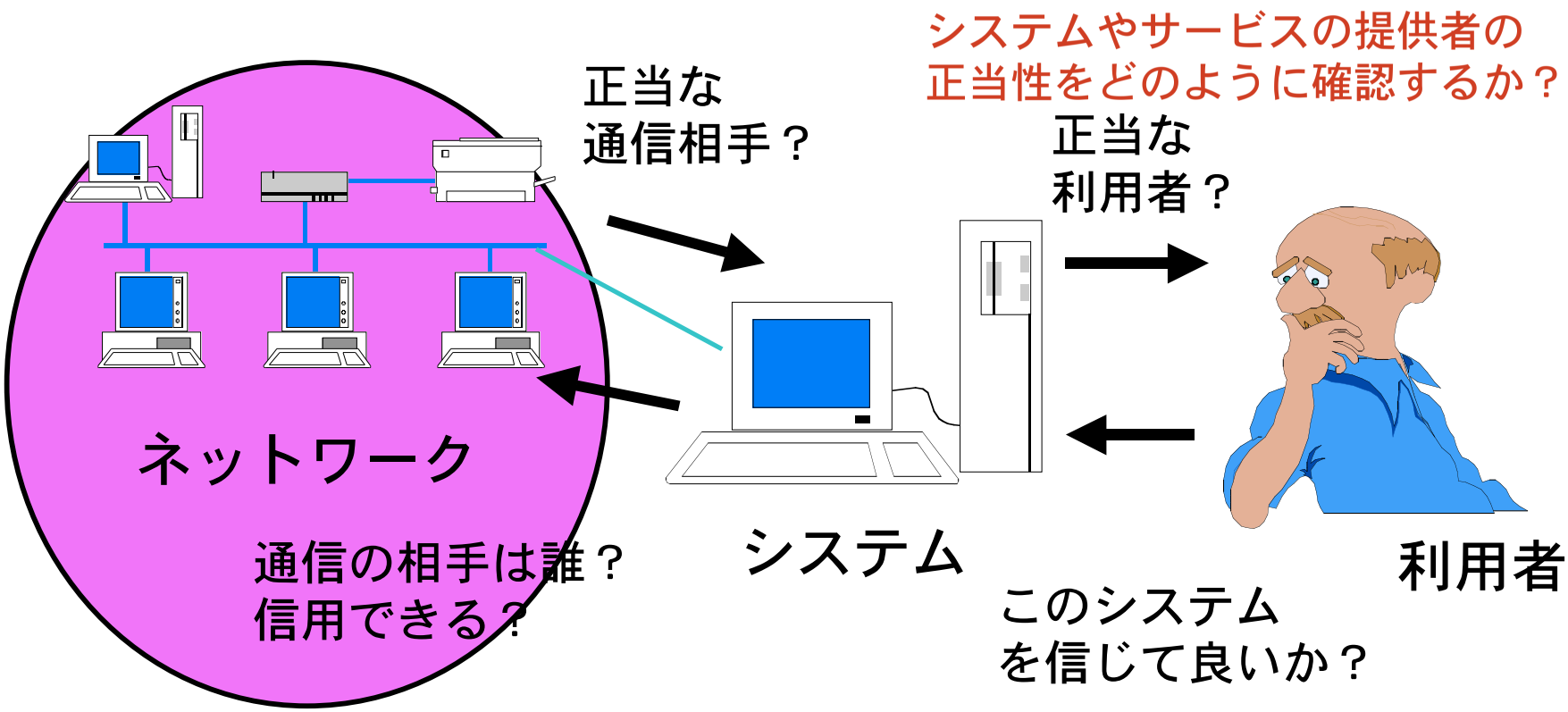
九州大学システムLSI研究センターの役割



社会基盤への条件

- 個人と社会の双方を守るための仕組みでなければならない．（個人の権利と社会の秩序）
- 仕組みは単純で理解しやすいものでなければならない．（弱者も不利にならないしかけ）
- 長期的に安定して運用が可能でなければならない．（信頼性と安定性）
- 技術の進歩に柔軟に対応可能でなければならない．（柔軟性と保守性）
- 攻撃や災害に対して強くかつ復旧が簡単に行えなければならない．（危機対応能力）
- 経済的に成り立たなければならない．（経済性）

相手が見えない情報社会



わかり易い社会基盤を！

- 利用者個人が基本原理を直感的に理解できる「単純な」システムでなければならない。
  - » 弱者を守るシステム
    - – システムを守るための複雑化は本末転倒
    - – 余計なお世話の透明性・利便性は危険を増やす
  - » 個人の責任によるリスク分散
  - » 製造物責任による過度のコスト上昇を防ぐ
- システムの安全性が数学的に証明・保証されている.

電子投票は信用できるか？

- 自分の投票が正しくカウントされることをどうやって確認するか？
- 自分の投票が特定されていないことをどうやって確認するか？
- 投票に疑惑があるとき，チェックが行えるか？（アメリカ大統領選挙のフロリダ州のような場合）

情報化社会の課題

- 通信相手の確認方法は？
- 自分が自分であることの証明は？
- 事実の確認方法は？
- 現実社会の事象と仮想世界の事象の対応は？誰が対応づけるのか？
- 真実とは何か？どうやって確認するか？
- 組織を守るセキュリティ技術から個人を守る技術への転換

これまでの半導体設計

- 半導体集積回路をベースとした同一回路の大量生産
  - » メモリとプロセッサー＞部品の発想
- 微細加工技術をベースとした継続的進化
  - » 高集積化・高速化・低消費電力化
  - » 価格は一定
- システムの集積化と応用分野の拡大

これからの半導体設計

- システム＝デバイス
  - » System LSI→集積回路にシステム開発の発想
- 製造技術の転換
  - » 微細化の限界　微細化する経済的なメリット
  - » ミニファブ　　製造の仕組みの変化
    - – バッチ処理から毎葉処理へ
    - – ファブラインの小型化・廉価化（月産10万個、50億円程度）
    - – 製造期間の短縮（１週間以下）
- 応用分野の変化→Post PC, Post 携帯端末
  - » 設計期間の短縮　市場の流動性
  - » 性能・コスト・電力から品質・信頼性・安全性へ

形式的手法への期待

- システムLSI設計に関する種々の課題への対応
  - 設計パラメータの多様性
  - 評価関数の多様性
  - 信頼性，安全性への対応
  - ソフトウェアとハードウェア双方に対応