

# 超ディペンダブルVLSIへの道

CREST-DVLSI

- Fundamental Technologies for Dependable VLSI Systems -

---

坂井修一

東京大学

# ディペンダビリティとは何か？

## ■ ディペンダビリティの環状モデル

### ● どこでディペンダビリティは定義されるべきか？

△ (～100% 設計) \* (～100% ハードウェア) \* (～100% ソフトウェア)

◎ **ユーザインタフェース**

### ● ディペンダビリティのために何をすべきか？

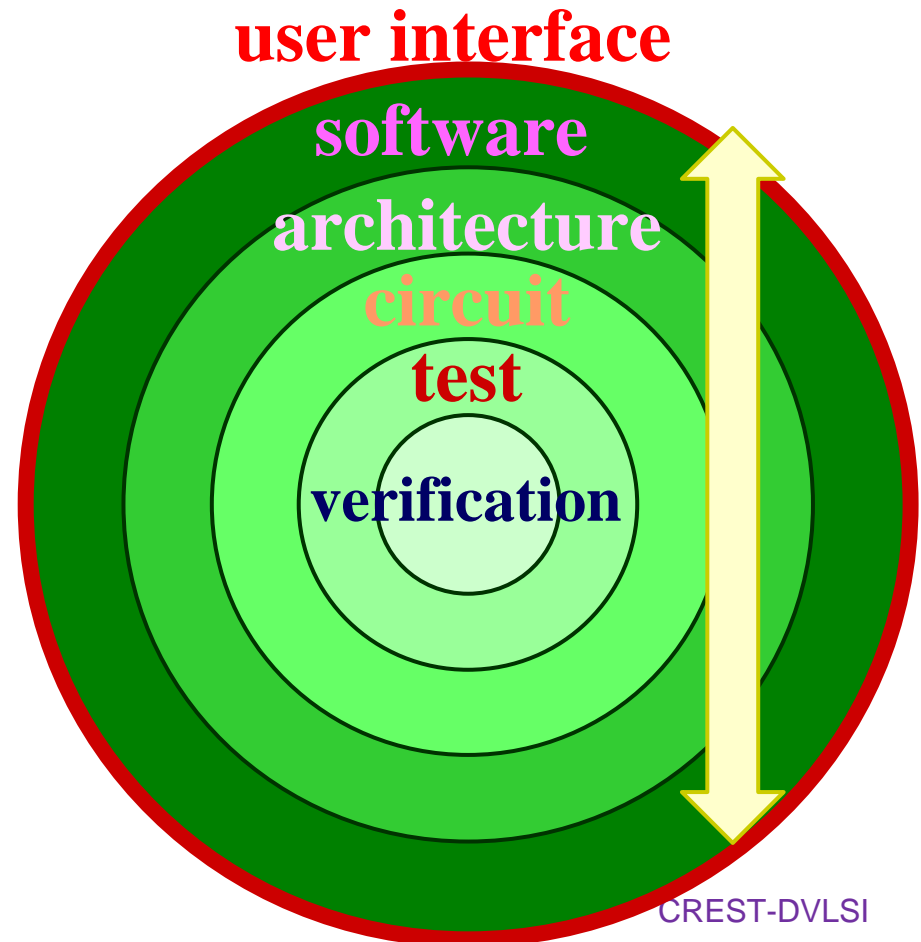
1. それぞれの層で提供

2. 層の間の調整

- 補完
- 重複除去
- 最適化

### ● IEC61508 / ISO26262

- 機能安全
- 決定論的 / 確率的



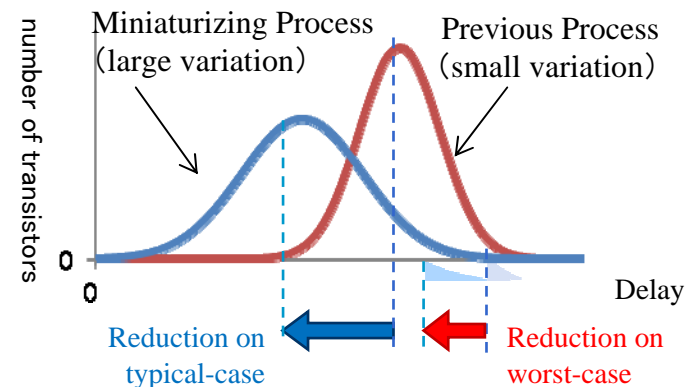
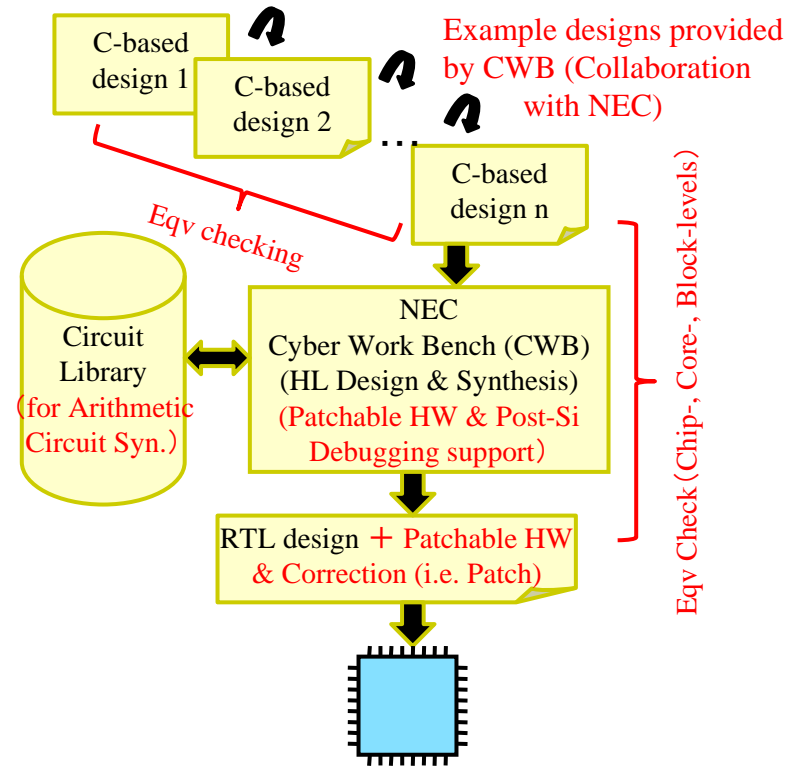
# 新しいデザインバランス

## ■ 形式検証 VS シミュレーション

- 藤田G  
-> 大規模形式検証
- 若林G  
-> 出口！

## ■ ワーストケース設計 VS ベストエフォート設計

- 坂井・五島G -> ベストエフォート設計
- 吉瀬G
- Fault Tolerant メニコア



# ベストエフォート設計＋実行時リカバリー

---

- 複雑さ / デザインばらつき
  - デザインコスト増大
- ベストエフォート設計
  - ノミナル設計に近い設計も可能に
- 実行時最適化
  - 故障検知
  - アーキテクチャ、ソフトウェアによるリカバリー
- ユーザレベルディペンダビリティの確保
  - エラー回避
  - 時間制約の充足



# 課題

---

1. 正しいか？
  - VLSI設計ミス / ソフトウェアバグ = 決定論的？
  - ソフトエラー、ハードエラー = 確率的？
2. ソフトウェアはベストエフォート …
  - デザインは ？
  - 回路は ？
3. 層間最適化は誰がやるか？
  - 水平分業が前提
  - アーキテクチャ、ソフトウェアは標準品が前提
4. 一番大切なものは何か？
  - 産業
  - 大学
  - ユーザ (社会)