ネットワークオンチップアーキテクチャに基づく車載制御システムの実現

Implementing automotive control systems based on Network-on-Chip architecture

米田　友洋（国立情報学研究所），今井 雅（東京大学），松本 敦（東北大学），齋藤 寛（会津大学）
Tomohiro YONEDA(NII), Masashi IMAI(Univ. of Tokyo), Atsushi MATSUMOTO(Tohoku Univ.), Hiroshi SAITO(Univ. of Aizu)

どんな研究？

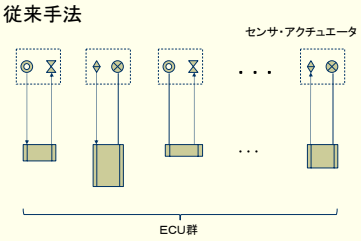
半導体プロセス技術の進歩に伴い、大規模でディペンダブルなVLSIを実現する上で、今までにないタイプの故障が問題となりつつあります。この研究は、このような問題を解決し、多数のユニットが適応的に協調動作できる、高ディペンダブルなネットワークオンチッププラットフォームを開発し、それに基づく車載制御系システム実現しようというものです。

何が特徴？

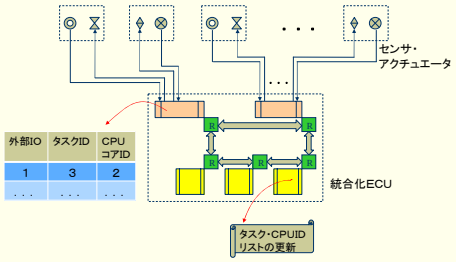
車載制御系システムでは、さまざまなタイプのECUが多数混在し分散的に配置されているため、製造コストや耐故障性などに関してさまざまな問題が指摘されています。そこで、センサやアクチュエータのみを必要な場所に残し、各ECUを統合した、集中型ECUをネットワークオンチップアーキテクチャを用いて高信頼に実現する手法を研究します。

基本的アイデア

- 従来手法



- NoC(Network on Chip)アプローチ



- 従来手法
  - センサ・アクチュエータとECUの対応が固定
    - 能力が余っても他に流用不可
    - 劣化・故障時の対応が困難
- NoCアプローチ
  - センサ・アクチュエータの制御はどのコアも可
    - 利用可能な資源を有効に活用できる
    - 劣化・故障時には他のコアを利用できる

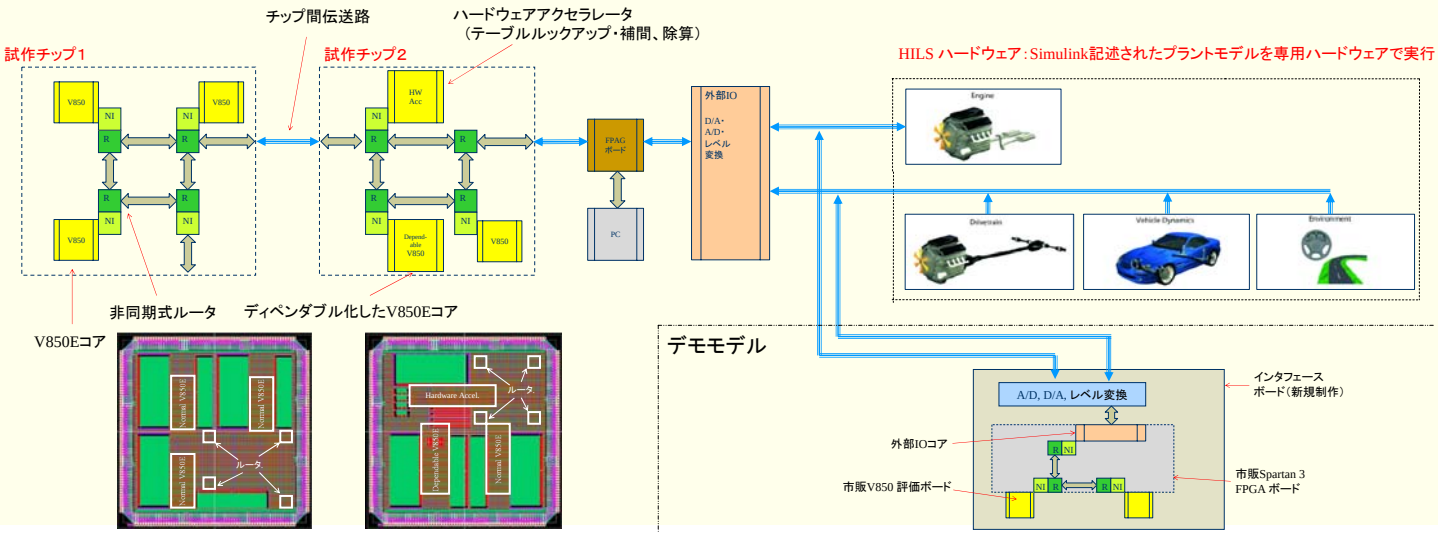
- 目的
  - CPUコアの各時点での能力に応じて，タスクの割り当てを，動的かつ自律的に行う
    - CPUコアの劣化や停止故障が起こると，タスクの実行中に，実行するCPUコアが自律的に切り替わる
  - 実現するために必要な事項
    - CPUコアの劣化や停止故障の検出
    - タスクを実行するCPUコアを動的に切り替えるしくみ
    - 自律的な制御方法
- 枠組み
  - 複数のCPUコア，外部IOコア，メモリコア

- 各CPUコアは他の劣化状況を監視
- 劣化・停止コアを検出したら
  - あるアルゴリズムに従い，(task_id, CPU_id)のテーブルを更新
    - そのアルゴリズムを実行するCPU: 例えば，一番アイドル時間の長い正常CPU
    - 結果を各CPUコア，外部IOコアに送出
  - 外部IOコアは新しい(task_id, CPU_id)リストに基づいてパケットの送受
  - CPUコアは状態（少量）の引き継ぎが必要
    - 参照・更新する頻度が少なければメモリコアに置いておくと引き継ぎ不要

10年先の目指すもの

- マルチコアシステムのための必須プラットフォームの提供
  - ユニットをつなぎ足すだけで、小型車から超高級車まで対応
  - システムの状況（故障・劣化状況等）と環境の状況から各時点における最適なタスク割当を自動的に実行
    - スリップ回避のための4輪ブレーキ制御時にいくつかのコアが故障している場合はカーナビを切る等
  - ハードウェア・ソフトウェアのための開発環境提供

第一次実証用モデル



NII 連絡先：米田　友洋（Tomohiro YONEDA）／ 国立情報学研究所　アーキテクチャ科学研究系　教授
TEL：03-4212-2557　FAX：03-4212-2693　Email：yoneda@nii.ac.jp

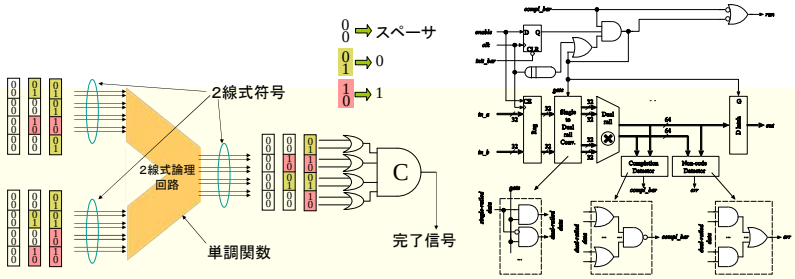
ネットワークオンチップアーキテクチャに基づく車載制御システムの実現

Implementing automotive control systems based on Network-on-Chip architecture

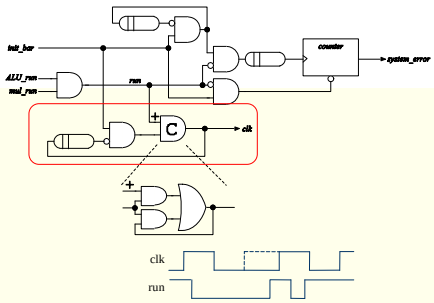
米田　友洋（国立情報学研究所），今井 雅（東京大学），松本 敦（東北大学），齋藤 寛（会津大学）
Tomohiro YONEDA(NII), Masashi IMAI(Univ. of Tokyo), Atsushi MATSUMOTO(Tohoku Univ.), Hiroshi SAITO(Univ. of Aizu)

V850Eのディペンダブル化の試み

- 乗算器の2線式化

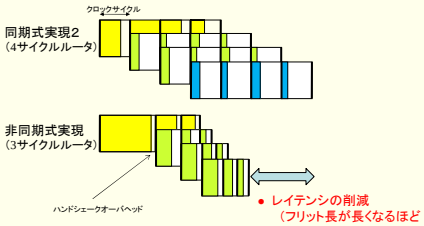


- 完了信号がでるまでクロックの停止



非同期式ルーターの設計

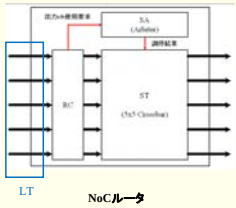
■ 同期式と非同期式の違い



- レイテンシの削減
  （フリット長が長くなるほど顕著になる）
- 通信路を早く解放

■ 1cycleルータを想定
- 入力チャネルと出力チャネルの間にバッファなし
- ヘッダフリットとその他ではサイクル時間が異なる

■ 1cycleの処理
- RC(Routing Computation)
  - フリットの出力ch の計算
- SA(Switch Allocation)
  - 出力chの調停（競合予防）
- ST(Switch Traversal)
  - フリットのクロスバの通過



■ 130nmプロセス、配置配線後のSDFを用いた評価

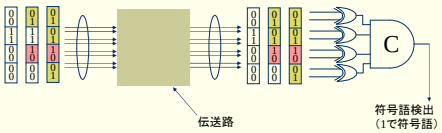
■ 4ルータでパケットを送りあうときの性能
レイテンシ [ns]

フリット数	4	8	16
競合あり	18.71	13.12	9.91
競合なし	13.66	9.68	7.53

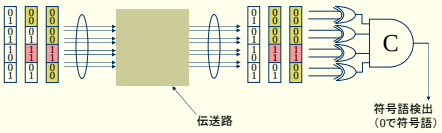
最大スループット：185 [MHz]

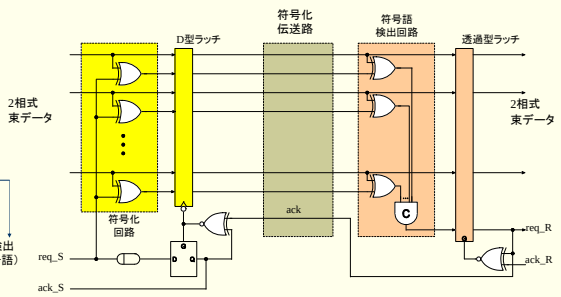
ルーター間の伝送方式（2相式符号化方式）

奇数相 0 1 ⇒ 無効符号語
0 1 ⇒ 0
1 0 ⇒ 1



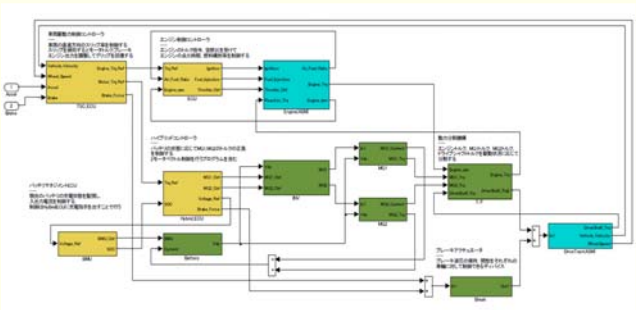
偶数相 0 1 ⇒ 無効符号語
0 0 ⇒ 0
1 1 ⇒ 1

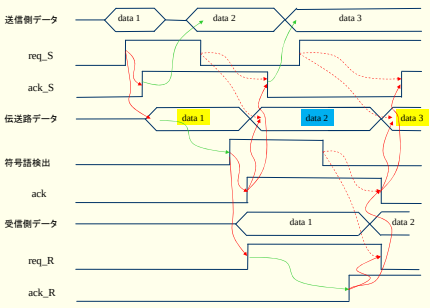




アプリケーションの選定

- ハイブリッドエンジン車のエンジン制御を想定
- 従来4個程度のECUで実現
  - ガソリンエンジン制御ECU
  - ハイブリッドエンジン制御＋駆動／発電モータ制御ECU
  - 駆動力制御用ECU
  - バッテリマネジメントECU
- カー関連メーカと共同開発
- HILSシステムにおいて稼働・評価





連絡先：米田　友洋（Tomohiro YONEDA）／ 国立情報学研究所　アーキテクチャ科学研究系　教授
TEL：03-4212-2557　　FAX：03-4212-2693　　Email：yoneda@nii.ac.jp