ディペンダブルネットワークオンチップ プラットフォームの構築

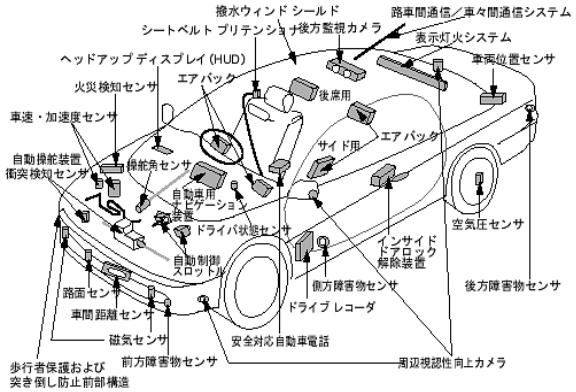
戦略的創造研究推進事業
「ディペンダブルVLSIシステムの基盤技術」

研究代表者　米田友洋（国立情報学研究所）
主たる共同研究者　今井　雅（東京大学）
松本　敦（東北大学）
齋藤　寛（会津大学）

National Institute of Informatics
国立情報学研究所

背景と目的

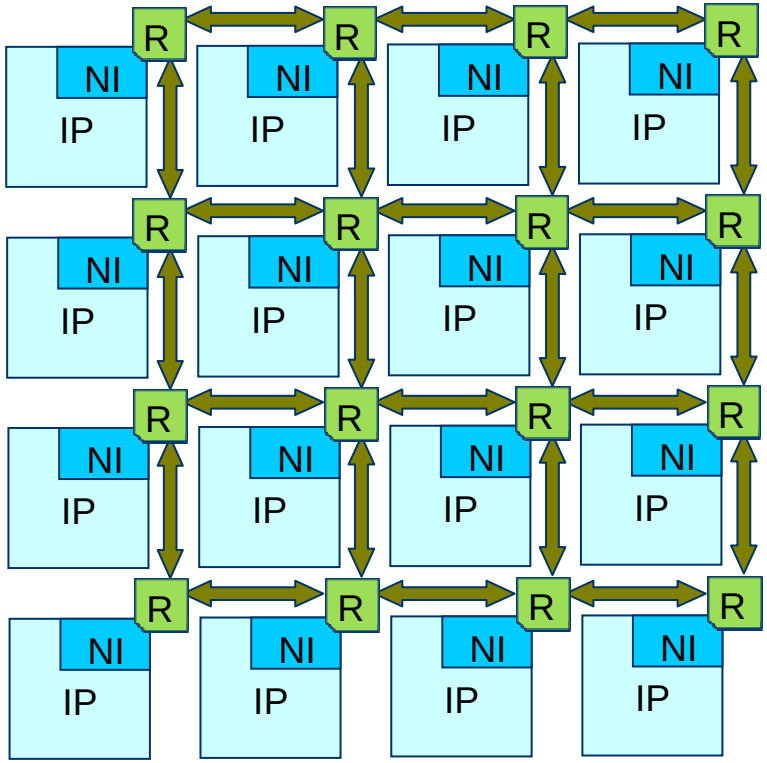
- 一つのVLSIに実装が求められるコア数
  - 近い将来に必ず急激に増加する
- （例）車載制御系システム
  - さまざまなタイプのECUが50個以上も混在
    - 実装方法にいくつかの課題
  - 新しいアプローチ
    - 集中型ECU
      - 各ECUを統合
    - X-by-Wire方式
      - センサー・アクチュエータ間を光ファイバなどで接続



「多数のコアが適応的に協調動作して異種多様なタスクを効率よく高信頼に実行できるプラットフォーム」を開発

アプローチ

- ネットワークオンチップ(NoC)

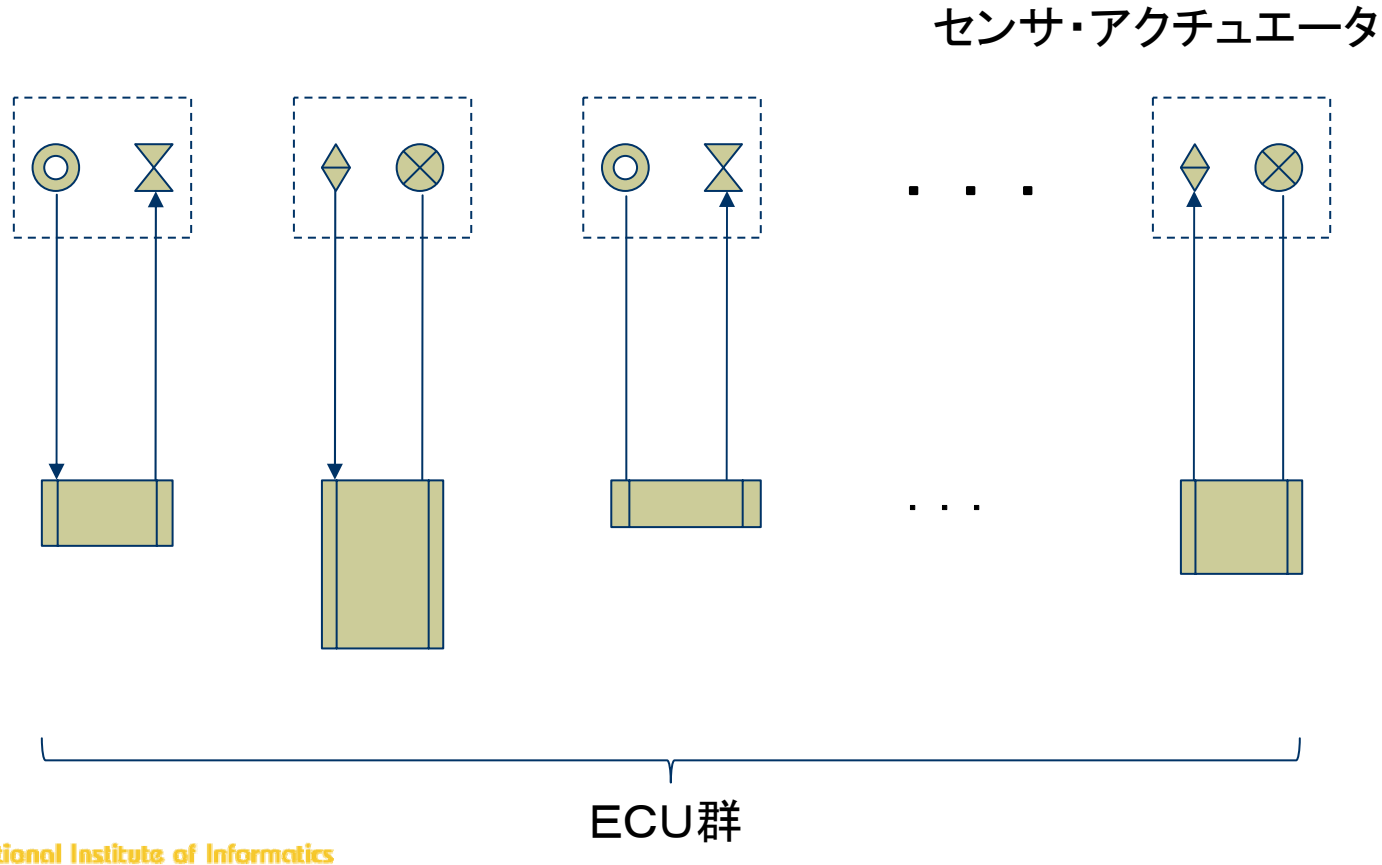


IP：CPUコアやアクセラレータ・メモリ等
NI: ネットワークインターフェース
R: ルータ

- 大域非同期局所同期(GALS)方式

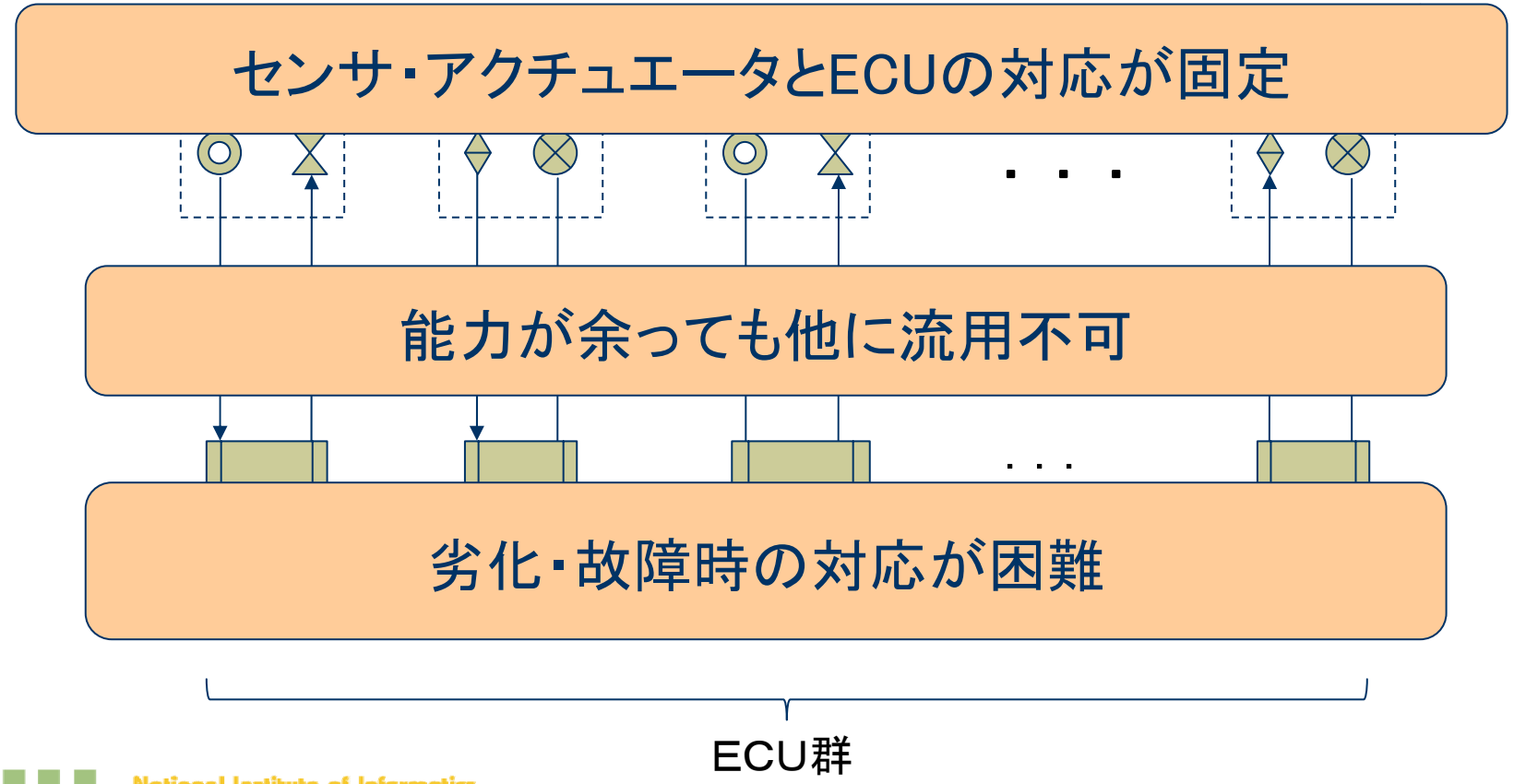
実証のイメージ

- 従来手法



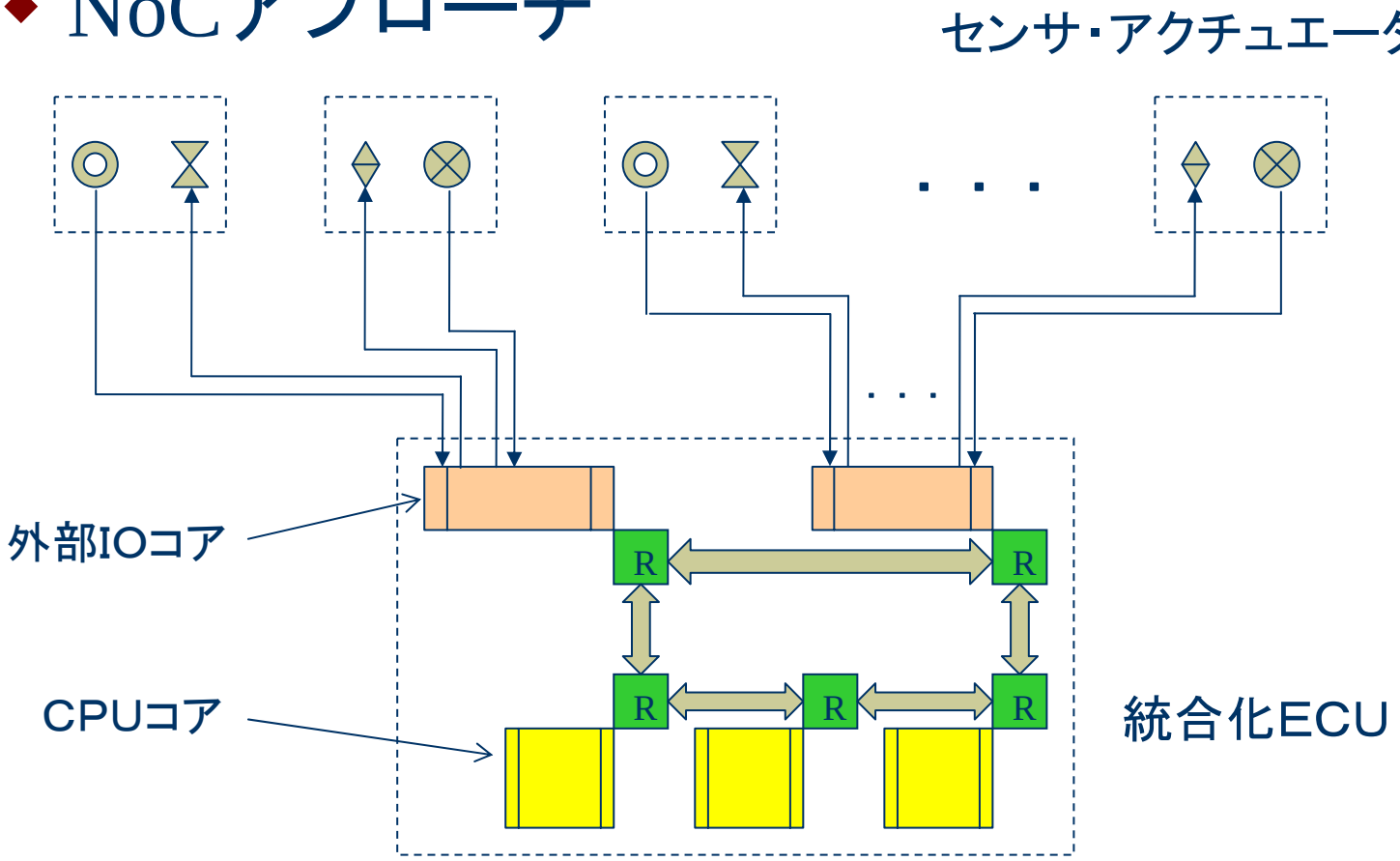
実証のイメージ

- 従来手法



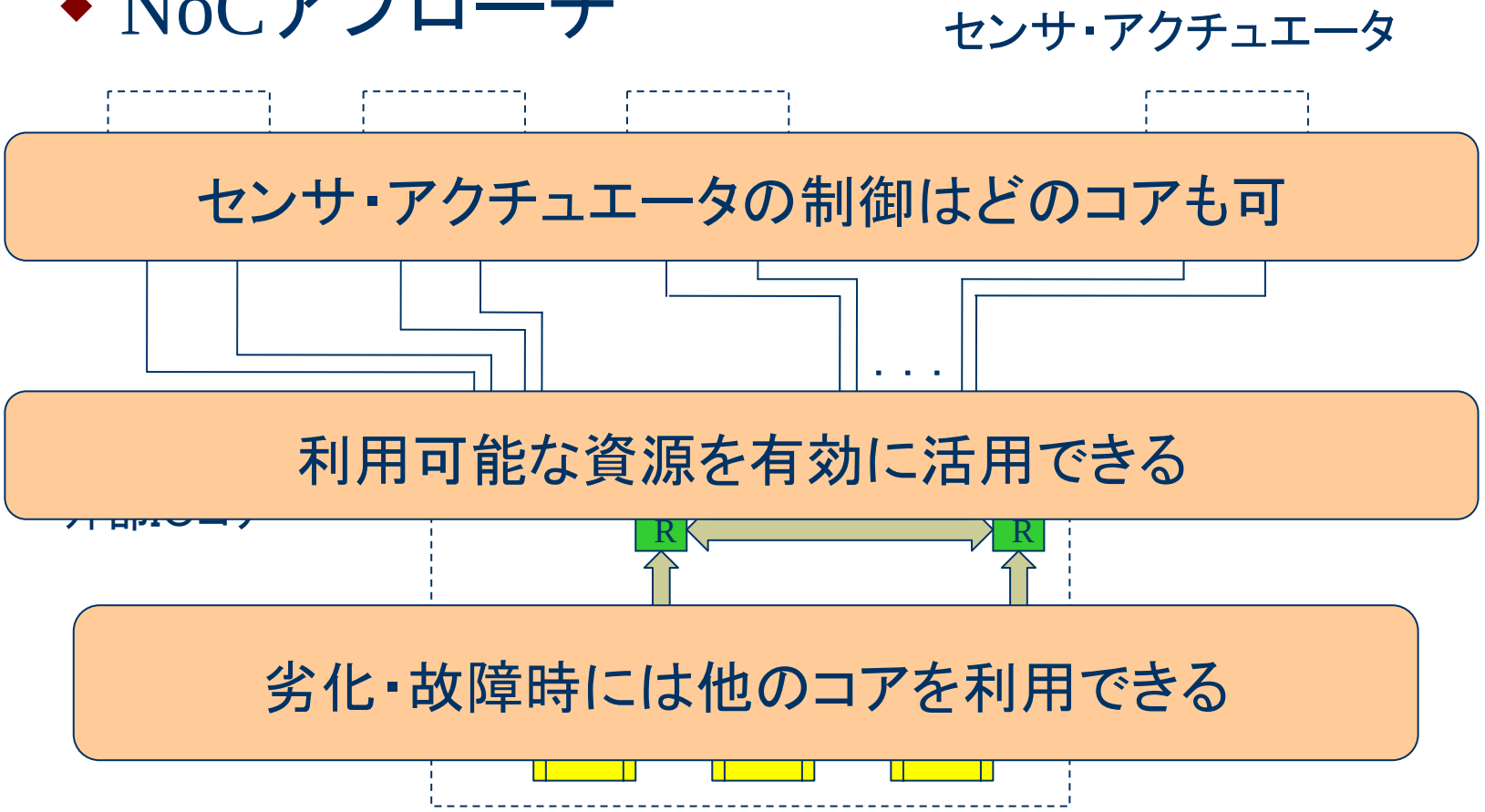
実証のイメージ

- NoCアプローチ



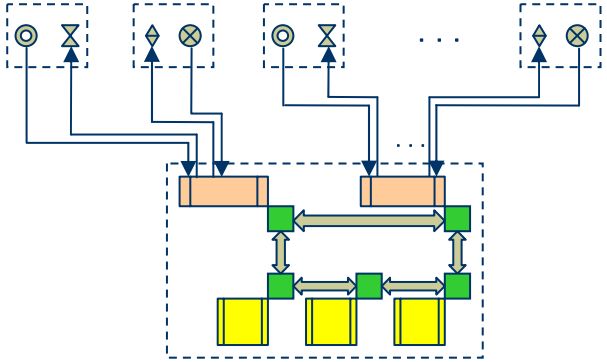
実証のイメージ

- NoCアプローチ

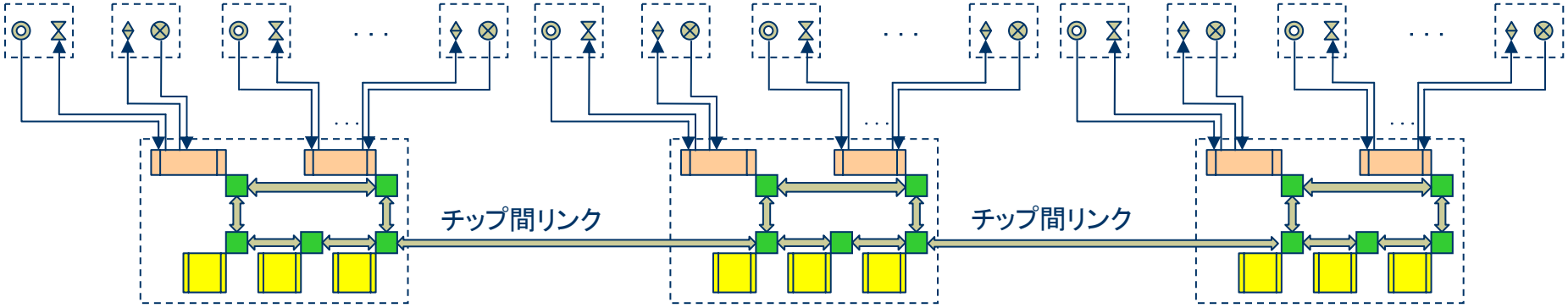


実証のイメージ

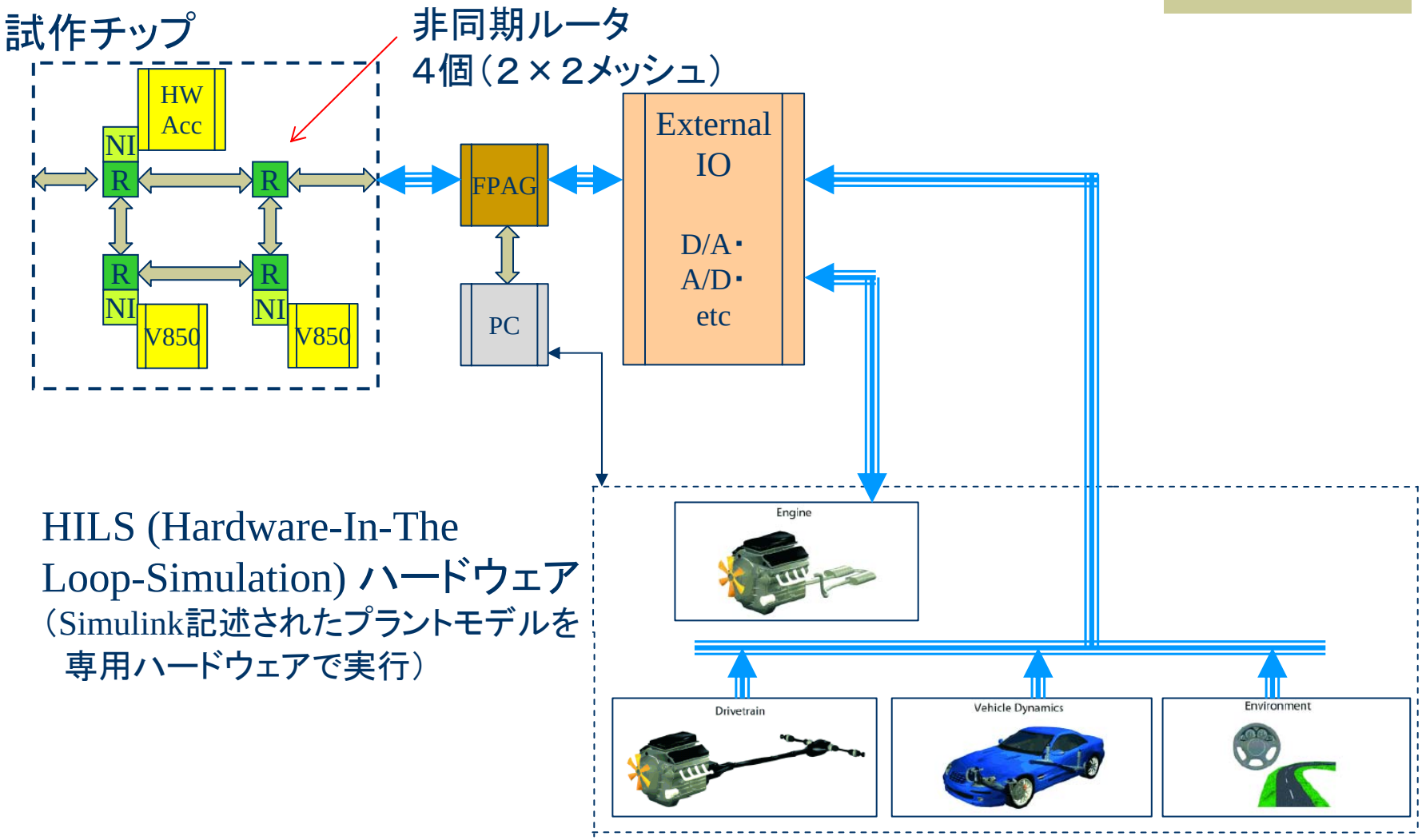
- 小型車



- 高級車

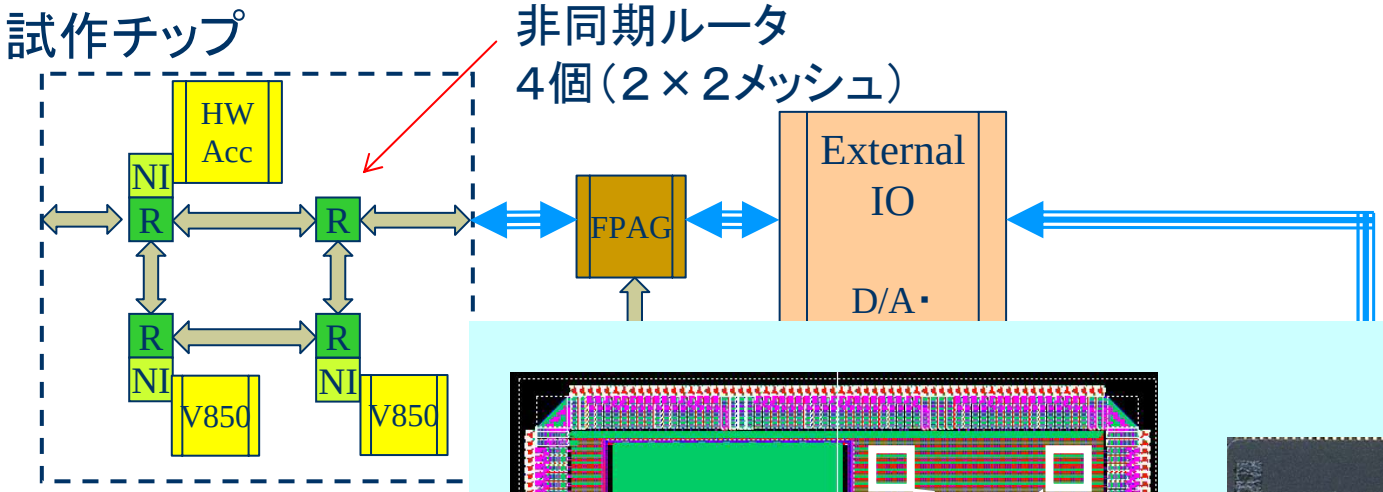


第一次実証用モデル

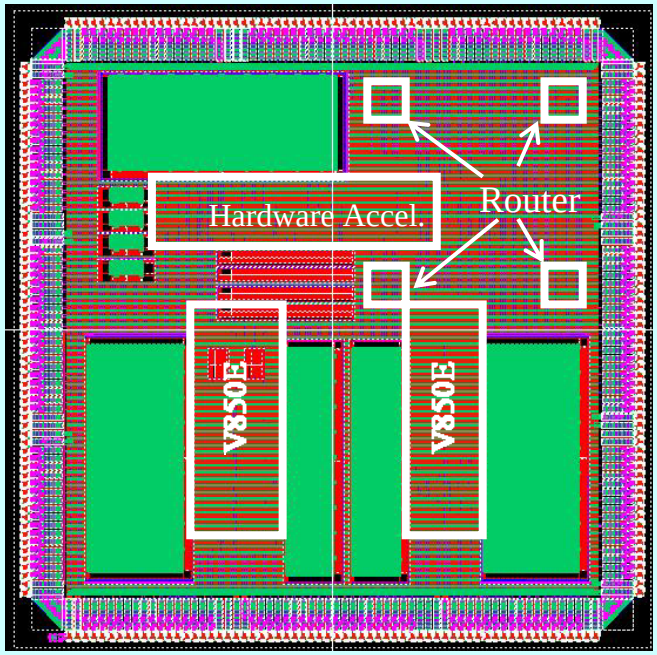


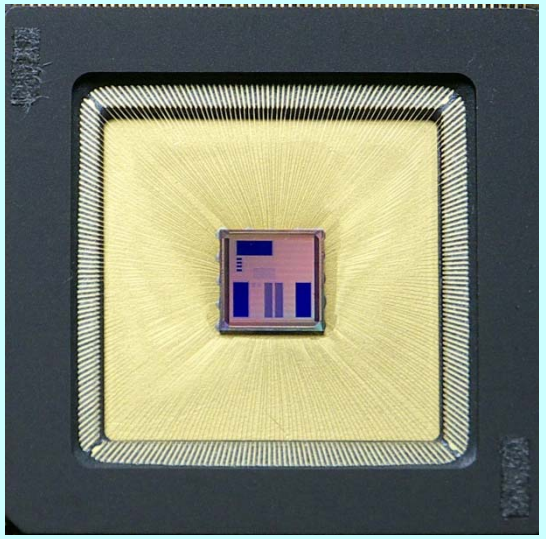
HILS (Hardware-In-The Loop-Simulation) ハードウェア
（Simulink記述されたプラントモデルを
専用ハードウェアで実行）

第一次実証用モデル

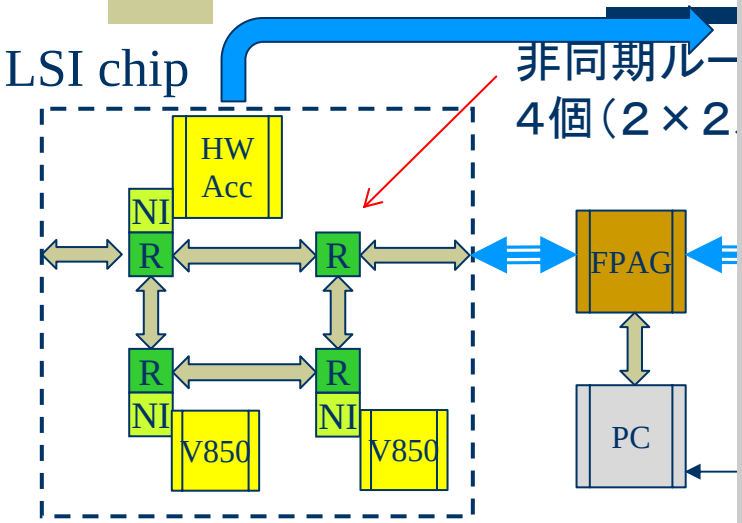


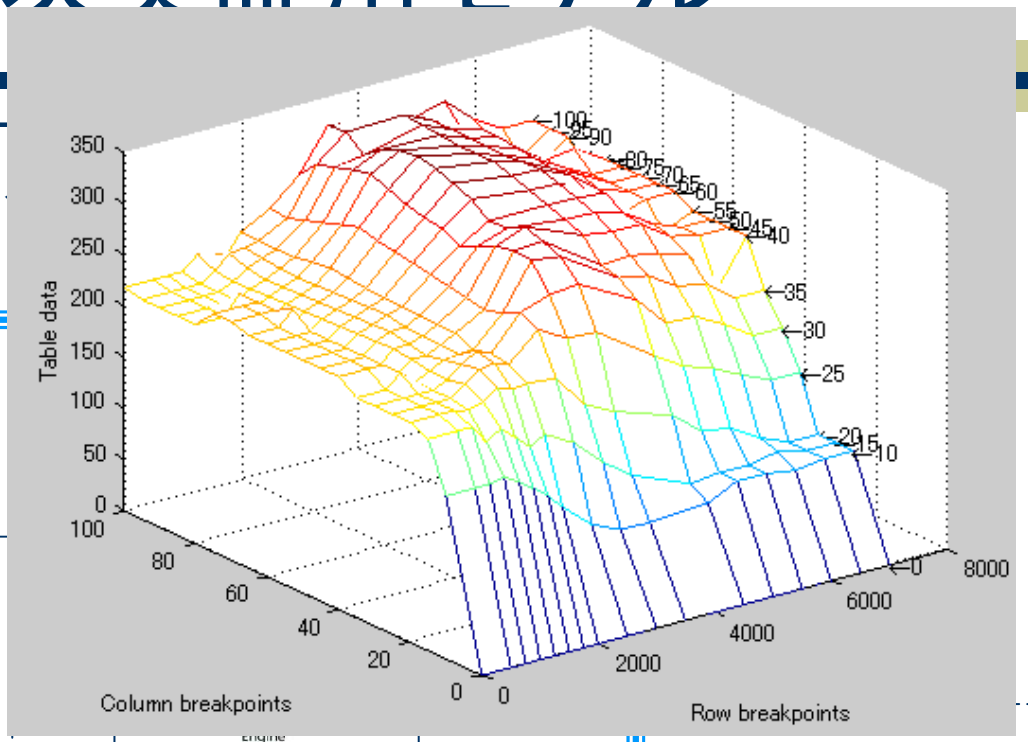
HILS (Hardware-In
Loop-Simulation) ノ
（Simulink記述されたプ
専用ハードウェアで実



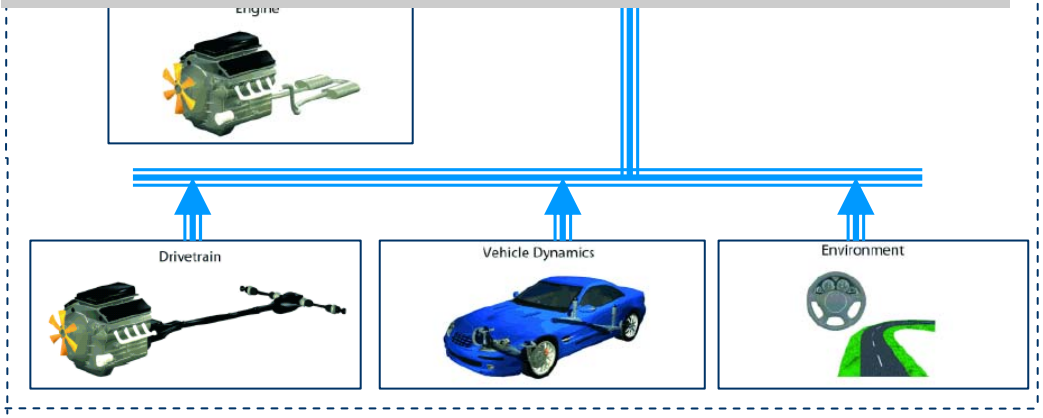


第一次実証用モデル



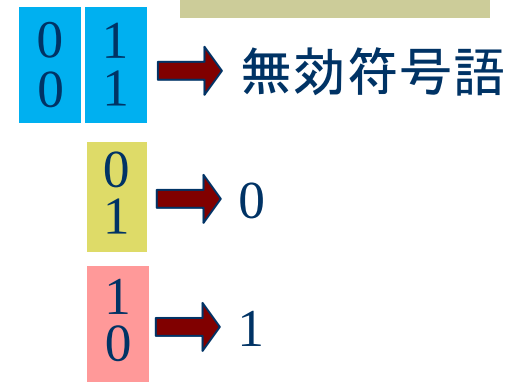


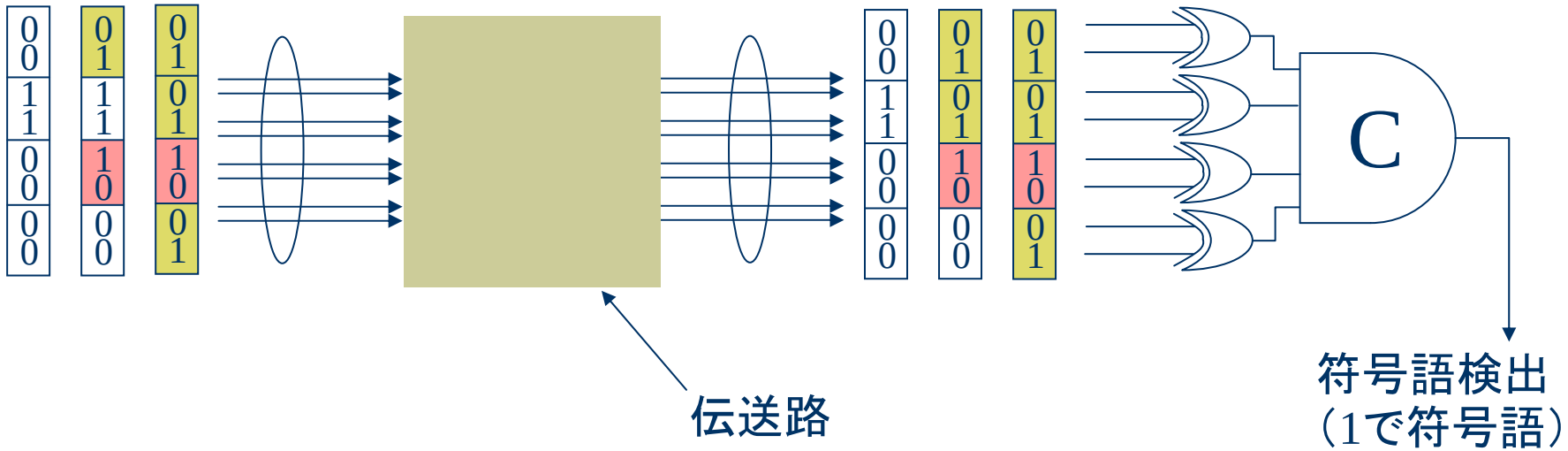
HILS (Hardware-In-The Loop-Simulation) ハードウェア
（Simulink記述されたプラントモデルを
専用ハードウェアで実行）



第一次実証用モデル

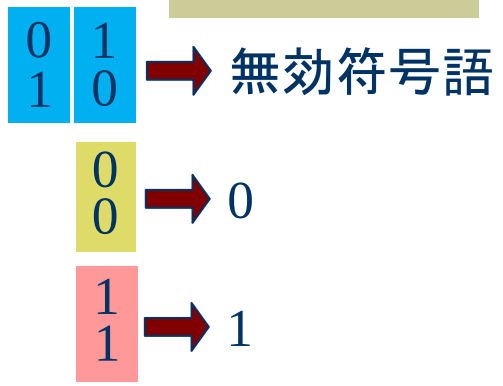
- リンク符号化
  - 2相式符号化方式(奇数相)

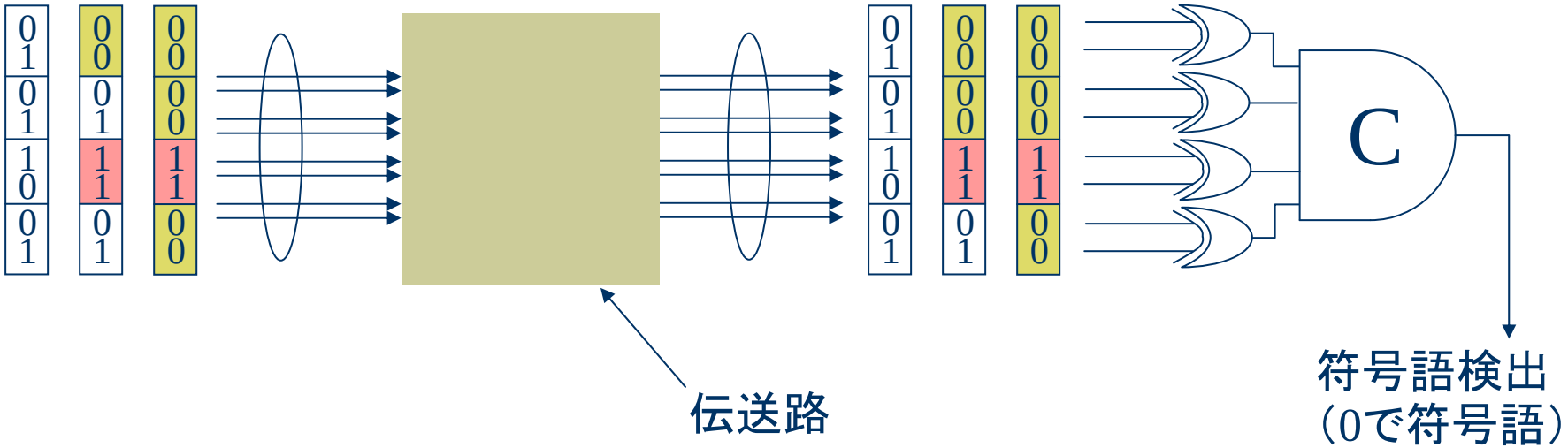




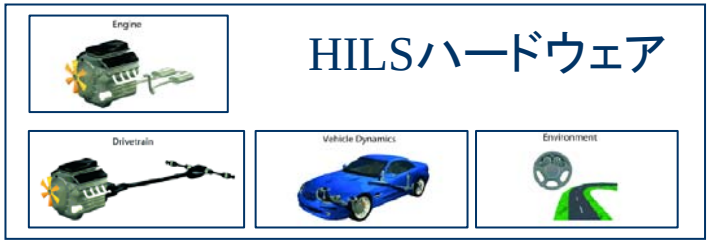
第一次実証用モデル

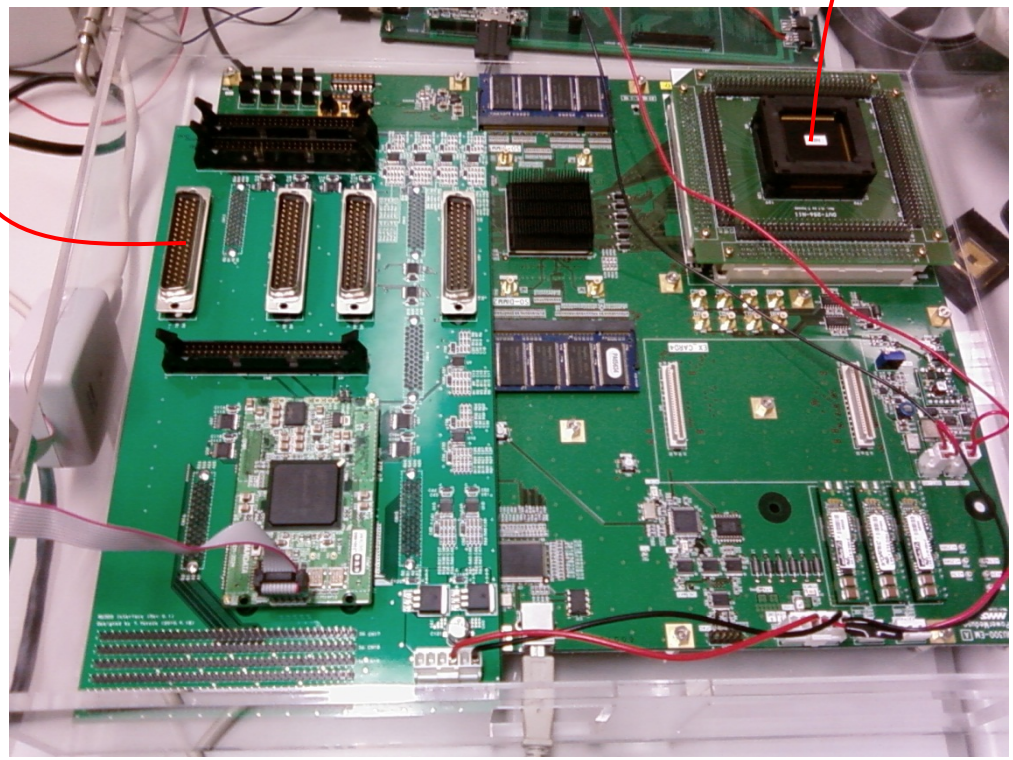
- リンク符号化
  - 2相式符号化方式(偶数相)

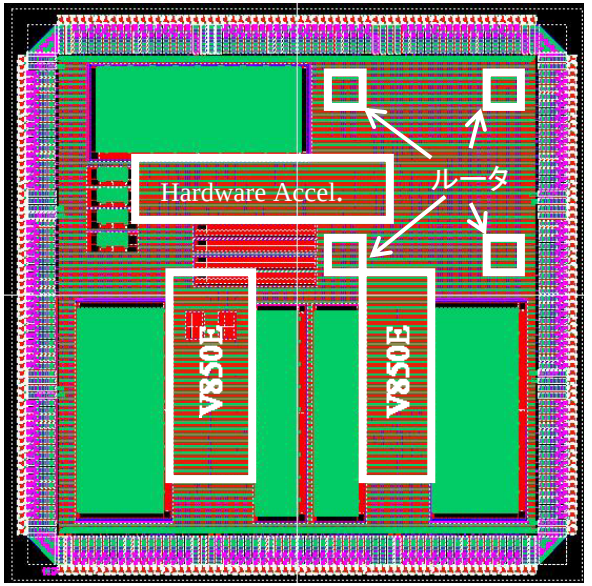




第一次実証用モデル


Engine
HILSハードウェア
Drivetrain
Vehicle Dynamics
Environment




Hardware Accel.
ルータ
V850E
V850E

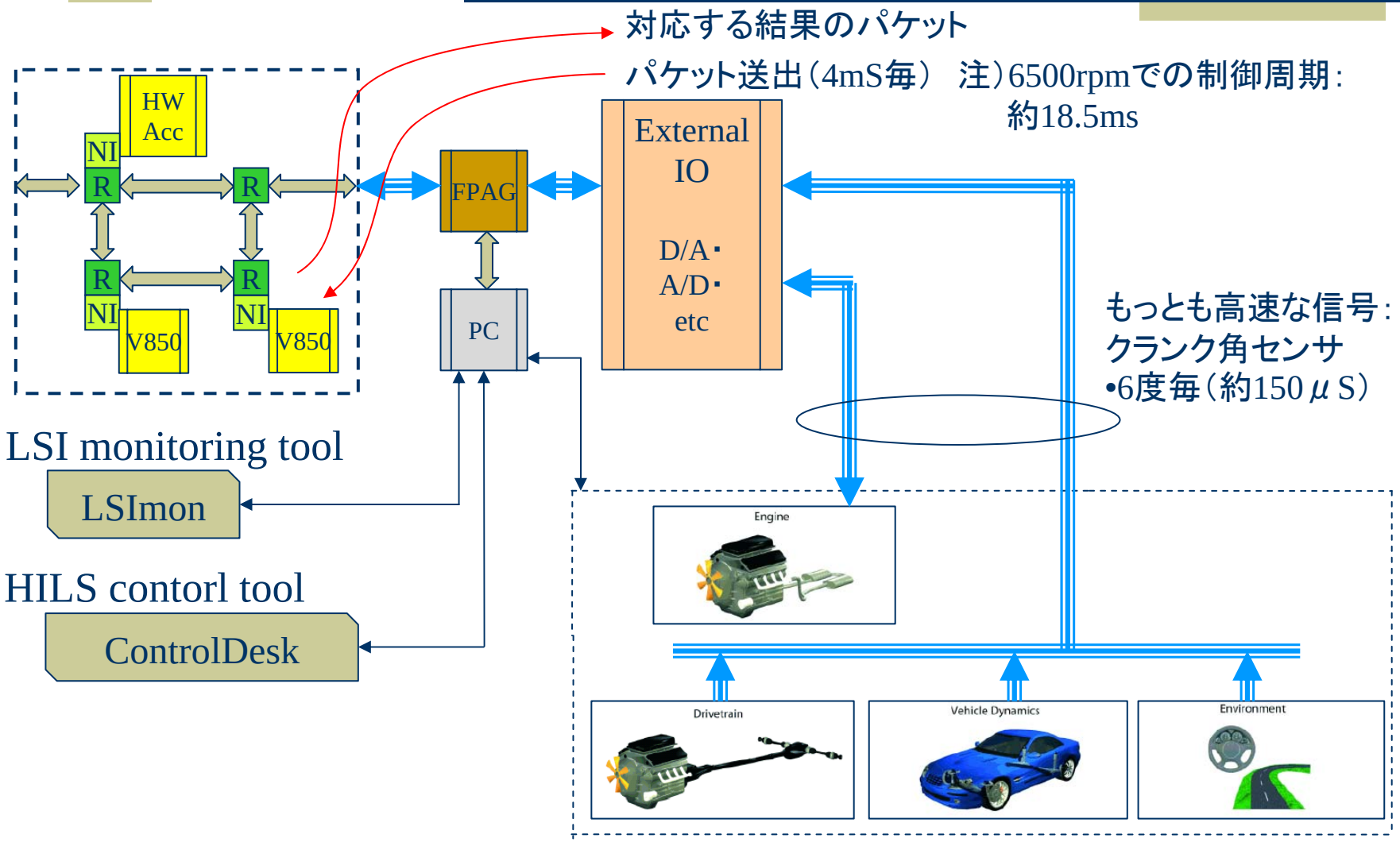
第一次実証用モデル

- V850Eコアで行っている制御
  - 燃料噴射制御（各気筒ごと）
    - 燃料噴射時間(us)
    - インジェクションタイミング(before TDC:度)
  - 点火制御（各気筒ごと）
    - 点火タイミング(before TDC:ラジアン)
  - Simulinkから変換したCプログラムがCPUコア上で動作

第一次実証用モデル

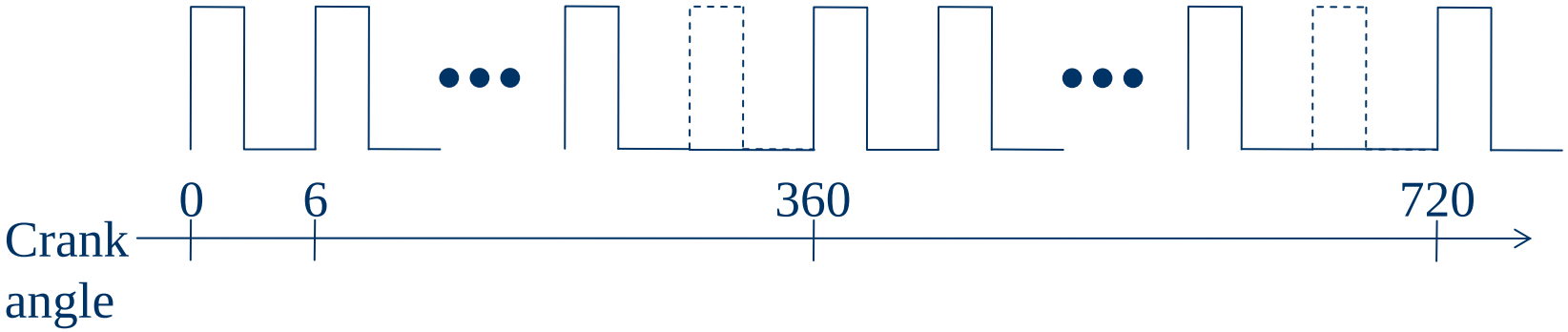
- 現在の実装
  - OSなし
  - 外部IOコア，浮動小数点コアは故障しないと仮定
  - 外部IOコア
    - センサ信号をAD変換器，タイマ等により数値化
    - パケットに格納してCPUコアに送る
    - CPUコアから送られてきたパケット内の数値を，DA変換，パルス発生器により信号化
  - ２つのCPUコアにはすべてのタスクを予めロード
    - センサ情報をパケットにて受信
    - 計算
    - 結果をパケットにて送出

第一次実証用モデル

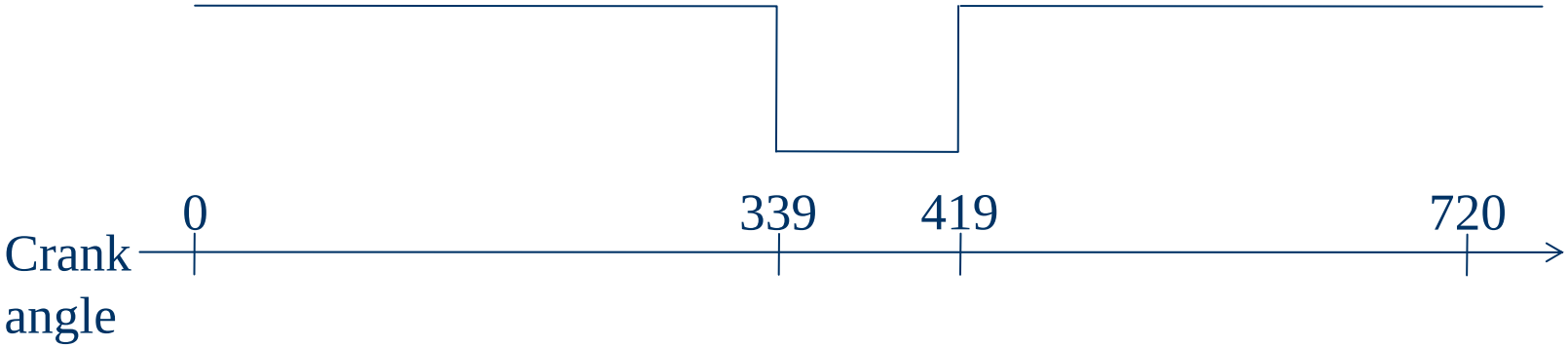

対応する結果のパケット
パケット送出（4mS毎）
注）6500rpmでの制御周期：
約18.5ms
HW
Acc
NI
R
R
R
NI
R
NI
V850
V850
FPAG
PC
External
IO
D/A・
A/D・
etc
もっとも高速な信号：
クランク角センサ
•6度毎（約150μS）
LSI monitoring tool
LSImon
HILS contorl tool
ControlDesk
Engine
Drivetrain
Vehicle Dynamics
Environment

第一次実証用モデル

- Crank angle sensor output



- Cam shaft sensor output

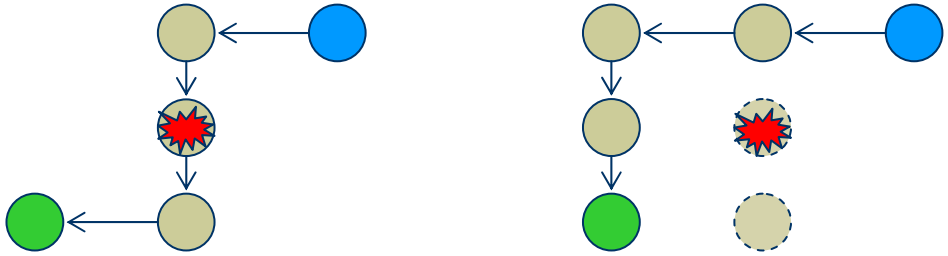


第一次実証用モデル

- 簡単な耐故障性の実現
  - 外部IOコアが主導
    - 初期状態では，パケットをCPUコア0に送る
    - 4mS毎のパケットに対して，次のタイミングまでに結果パケットが返ってこない場合，そのCPUコアを故障と見なす
    - CPUコアの故障を検出した場合は，それ以後，もう一つのCPUコアにパケットを送る
    - 現在の実装では，CPUコアは内部状態を持たないので，CPUコアの切り替えが容易に実現

オンチップネットワークの耐故障化

- ディペンダブルなオンチップネットワーク
  - 非同期式実現による耐劣化性
  - 故障ルータを自律的に避けてパケットを伝送するディペンダブルルーティング



  - 符号化リンクに適した新しいタイムアウト検出手法（特許出願）
  - ソフトエラー等への耐性を高めた制御回路構成法

CPUコアの故障検出と再構成

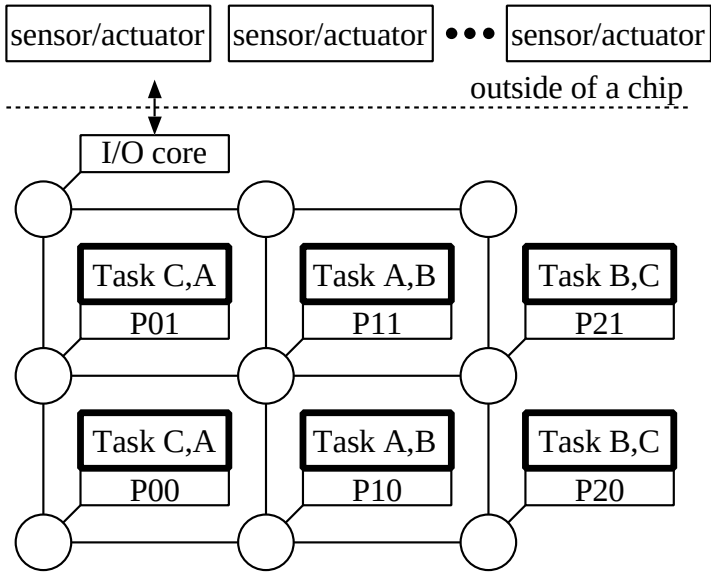
- プロセスの二重化による故障検出とTMR(Triple Module Redundancy)等による故障判定とシステム再構成
  - 通常時(replication and comparison phase)
    - タスクをコアのペアで実行し、結果を比較する
    - 不一致が生じなければ次のタスクを実行、不一致が生じた場合は故障判定・再構成時モードに移行
  - 故障判定・再構成時(retry and decision phase)
    - 不一致が生じたタスクを再度実行
      - 予め冗長にタスクを割り当てておいた他のコアと不一致を生じたペアのコアを用いてTMRを構成
      - 再度不一致の場合、永久故障と判定して故障コアを同定し、システムを再構成して通常時モードに移行
      - 不一致が生じなければ、一過性故障と判定して通常時モードに移行

CPUコアの故障検出と再構成

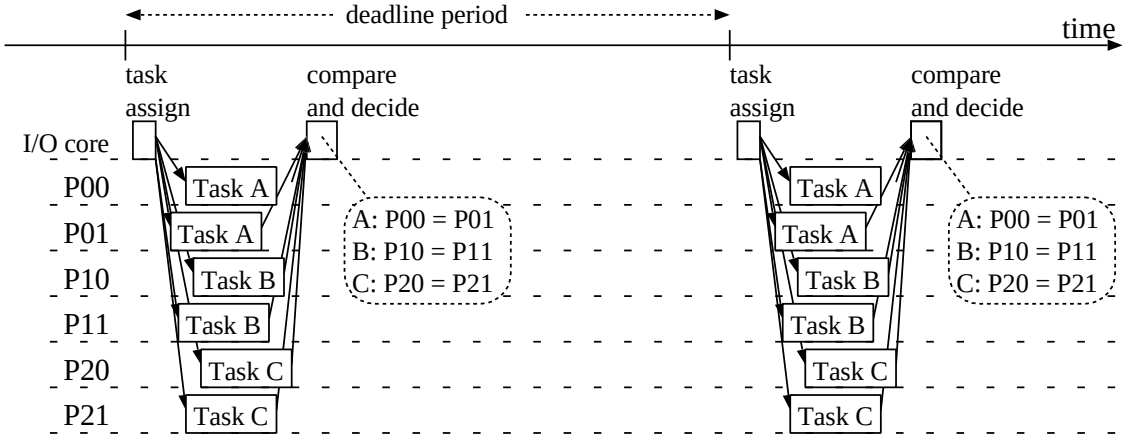
- フォールトモデル
  - 単一コア故障
    - 故障は一度に一つのコアでしか発生しない
    - 故障の発生頻度は比較の頻度に対して十分小さく、故障診断の間、他の故障は発生しない
  - 故障は一過性故障と永久故障の両方を仮定
    - 永久故障の場合、故障コアを同定して使用を停止
    - 一過性故障の場合、使用を継続

CPUコアの故障検出と再構成

- タスクの静的4重化割り当て
  - 例：繰り返しタスクA,B,C、コア数6
    - 各コアがプライベートメモリを持つ形式のNoC-SoCを想定
    - 各メモリにはタスクを2つずつ保持
    - I/Oコアがタスク割り当て、計算結果の一致判定を行う（I/Oコアは無故障と仮定）

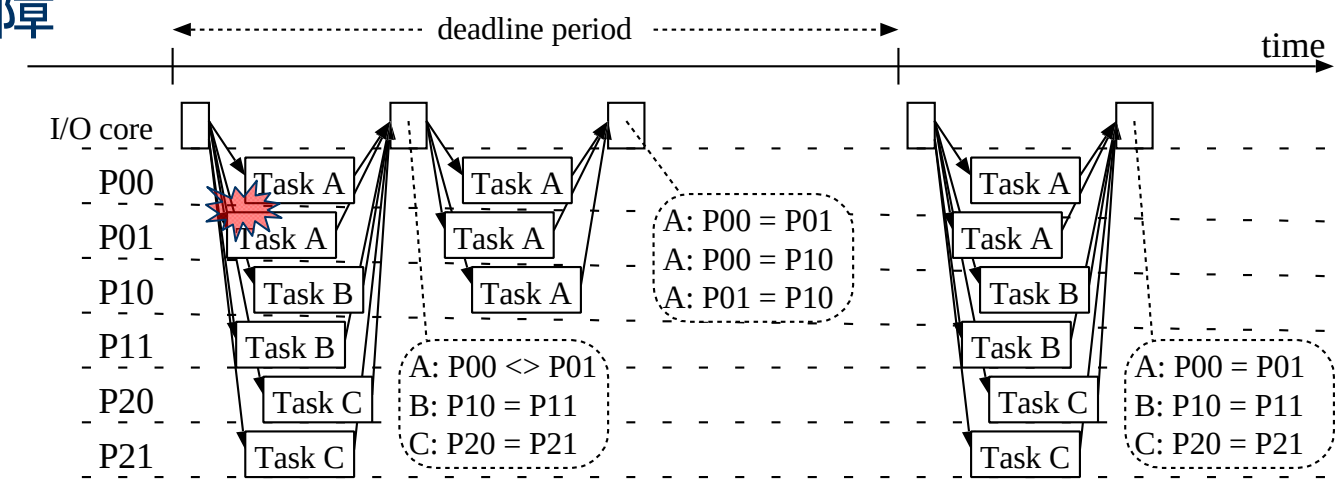


- 無故障時

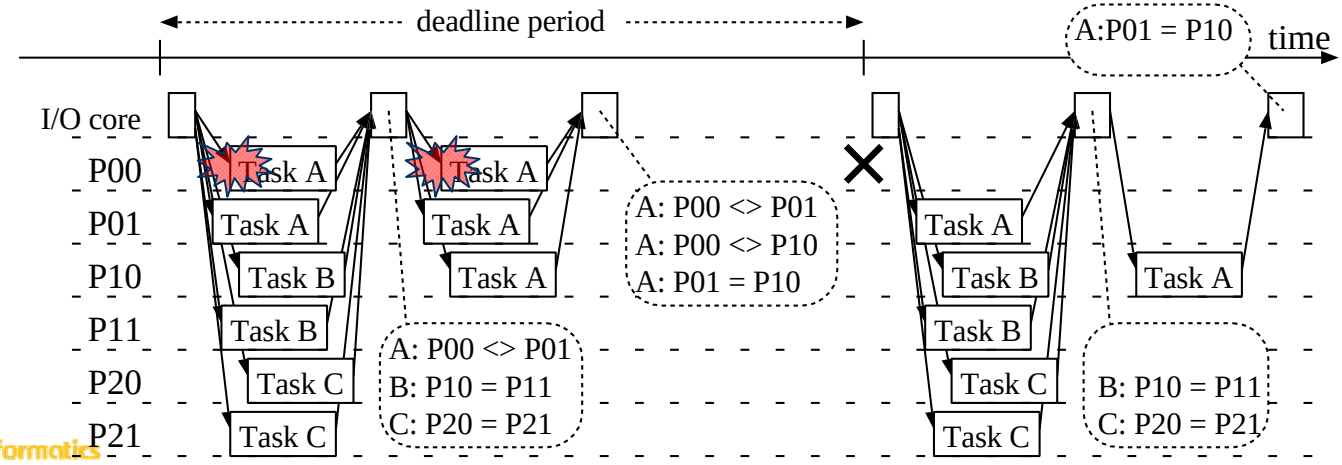


CPUコアの故障検出と再構成

- 一過性故障

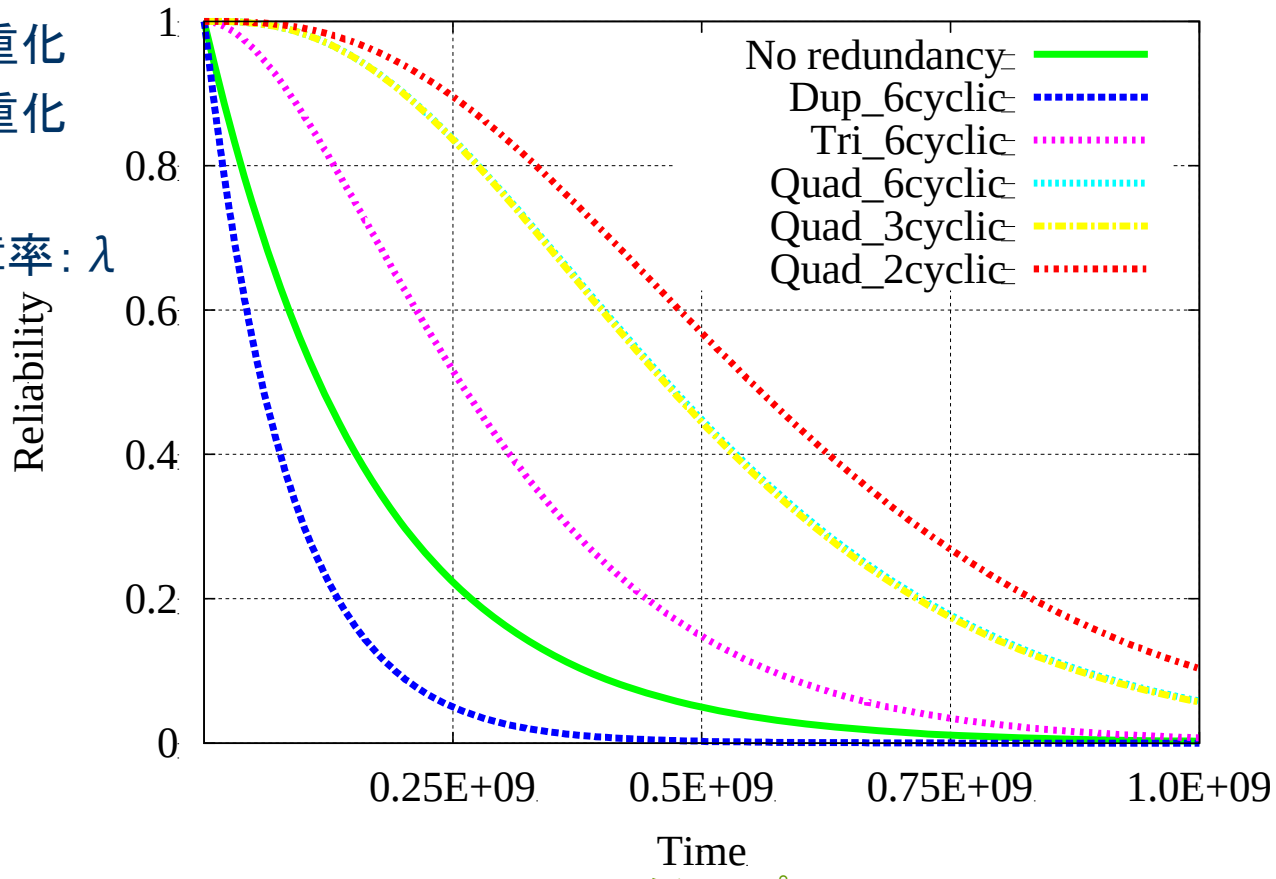


- 永久故障



CPUコアの故障検出と再構成

- Reliabilityの比較（6タスク/12コアの場合）
  - 冗長化無し（6コアのみ）：故障判定不可能
  - タスクの静的2重化
  - タスクの静的3重化
  - タスクの静的4重化
  - コアの永久故障率：λ
    - λ = 1E-09



CPUコアの故障検出と再構成

- MTTF(Mean Time To Failure)の評価（6タスク/12コアの場合）

Configuration	System MTTF	Memory size
2重化	0.0833/ λ （1.00）	1.0
3重化	0.311/ λ （3.73）	1.5
4重化	0.596/ λ （7.15）	2.0

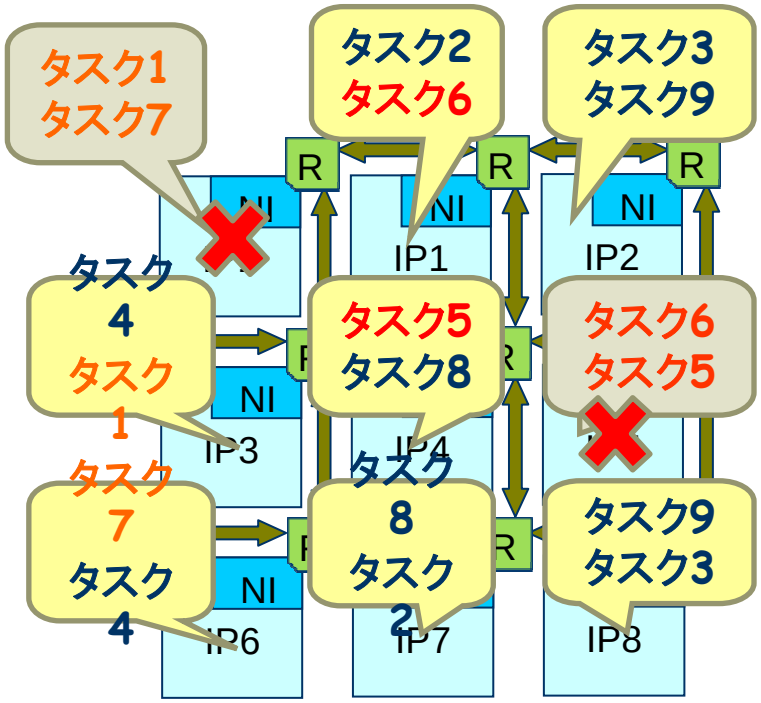
  - λ :コアの永久故障率
  - 2重化と比較して、4重化は必要メモリサイズは2倍に抑えつつ、システムのMTTFを7倍以上のばせる

コアの故障を考慮したタスク割り当て

- SimulinkでモデリングされたNoC車載アプリを実装
- 目標
  - タスク割り当てを支援する設計支援ツールセットの開発
    - 多くの故障パターンに対応できるようタスクの割り当てを行う
      - 同一タスクを多重に割り当てる
    - 時間制約の下、故障パターン毎にタスクスケジューリングと使用するタスクを決める
      - 同一タスクが複数あるため、実際にどれをつかうのかを決める
- 方針
  - コアの故障数、メモリサイズの下、タスクを各メモリに多重に割り当て
    - 通信時間の考慮
    - 多重化の制限
  - タスク割り当てと時間制約の下、故障パターン毎にタスクスケジューリングと実際に使用するタスクを決定

コアの故障を考慮したタスク割り当て

- メモリサイズや故障数の下、あらかじめ同一のタスクを多重に割り当て、実行できる故障パターンを最大に
  - 多重故障にも対応可



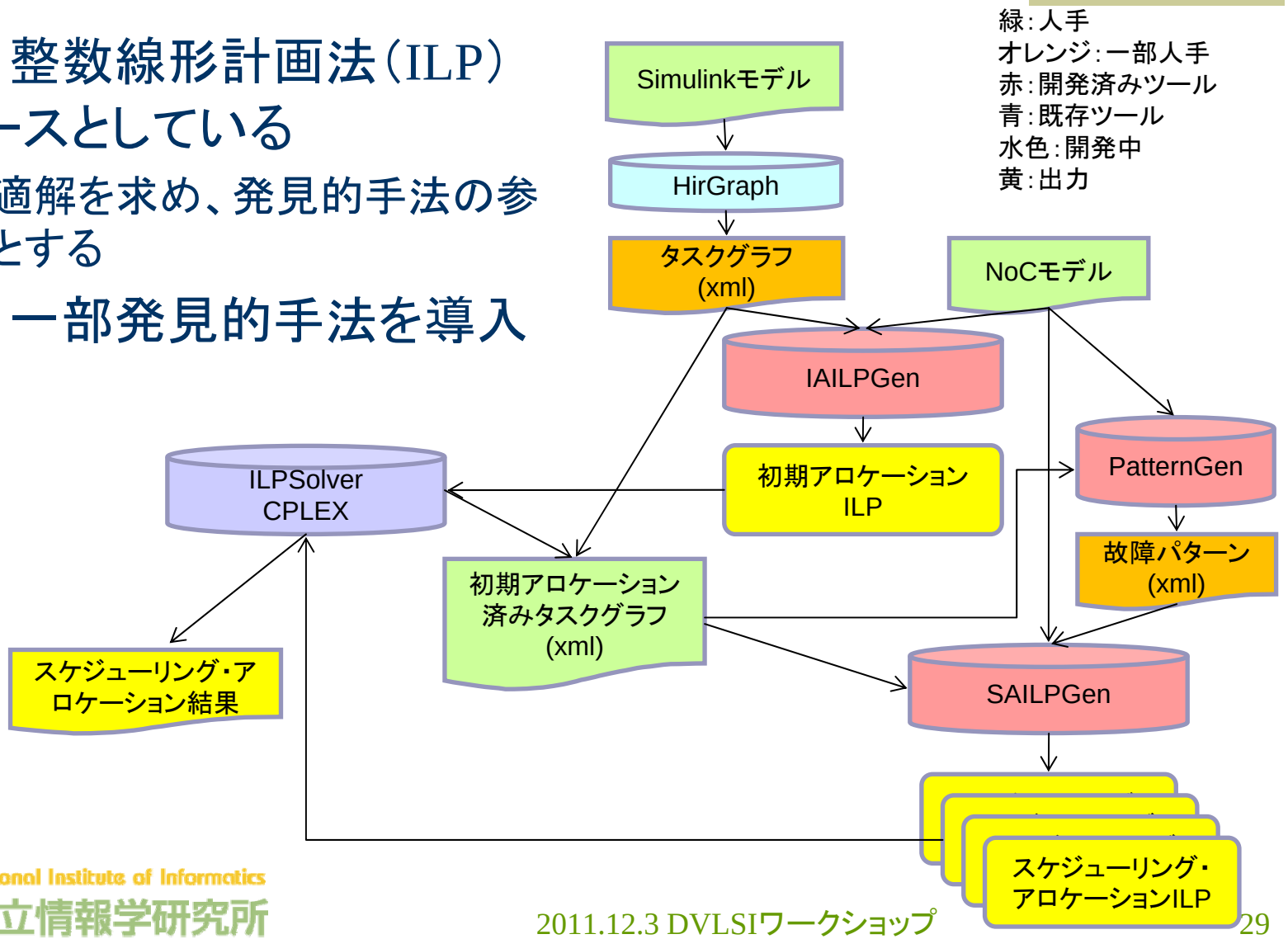
故障パターンA:IP0とIP5の故障

IP0のタスクはIP3とIP6にも存在
IP5のタスクはIP1とIP4にも存在

故障パターンAは動作可能

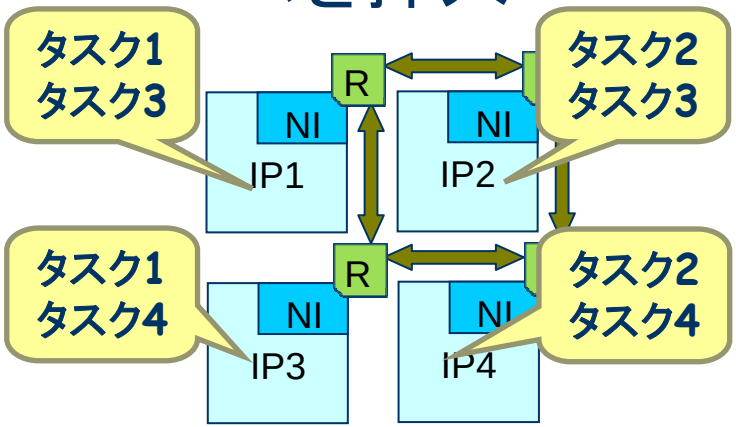
コアの故障を考慮したタスク割り当て

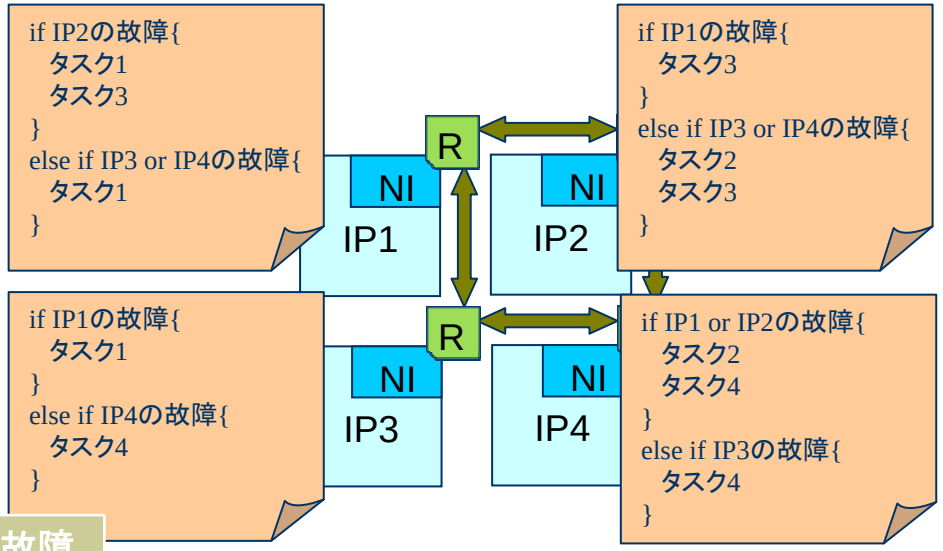
- 現在：整数線形計画法（ILP）をベースとしている
  - 最適解を求め、発見的手法の参考とする
- 今後：一部発見的手法を導入



コアの故障を考慮したタスク割り当て

- 故障パターン毎に使うタスクが決まる
  - コア毎に故障が起こった時に切り替えるプログラムを挿入

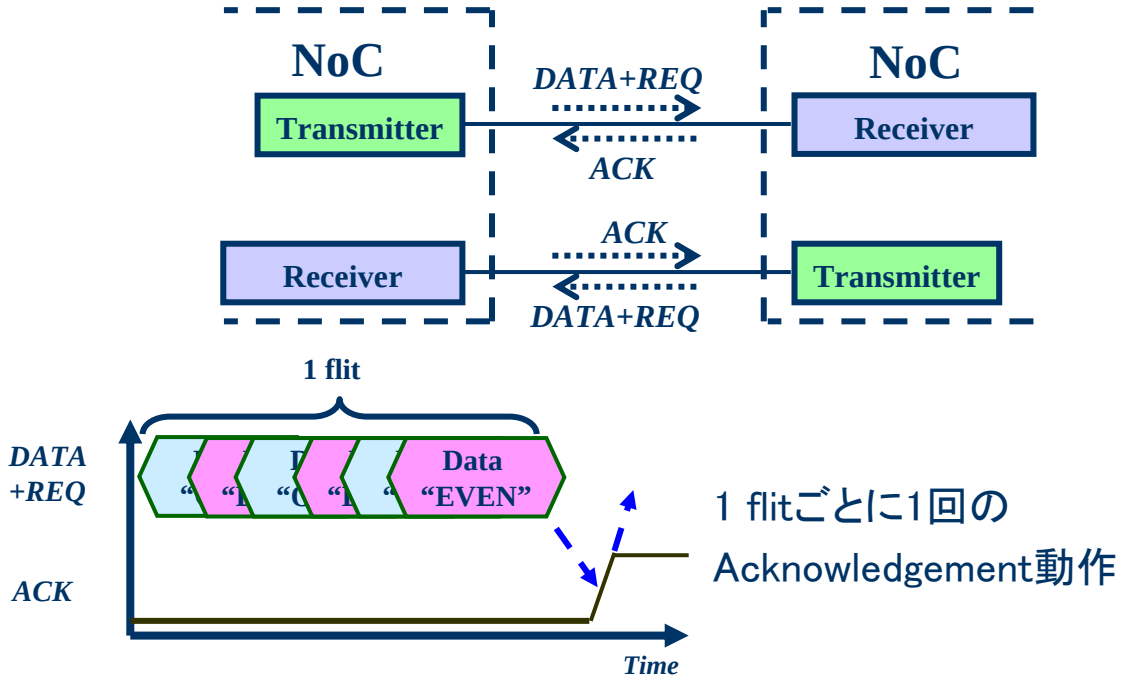




	IP1の故障	IP2の故障	IP3の故障	IP4の故障
タスク1	IP3	IP1	IP1	IP1
タスク2	IP4	IP4	IP2	IP2
タスク3	IP2	IP1	IP2	IP2
タスク4	IP4	IP4	IP4	IP3

高効率チップ間通信

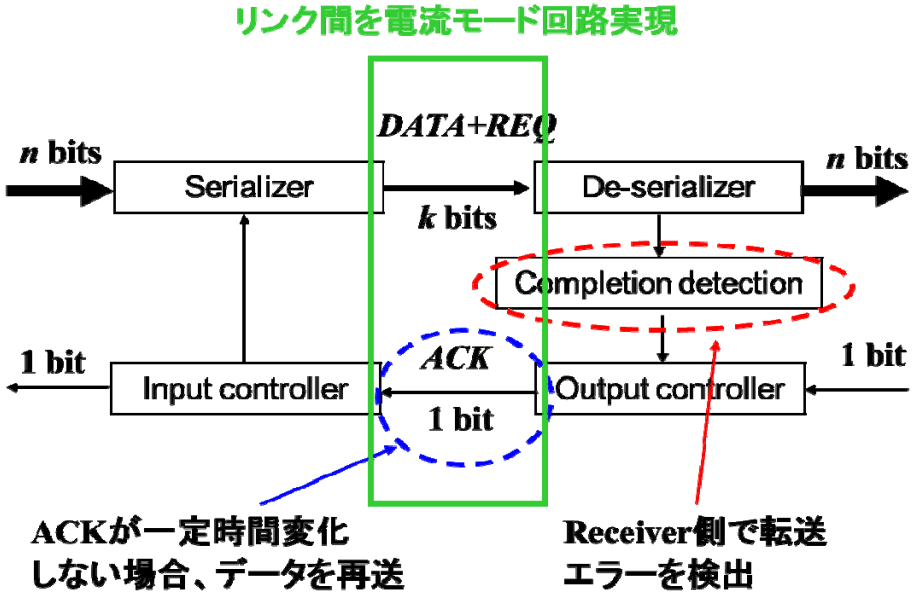
- 非同期バースト転送方式
  - ハンドシェイクに要するコスト削減→スループットの向上



Acknowledgementの回数を減らすことで高速転送

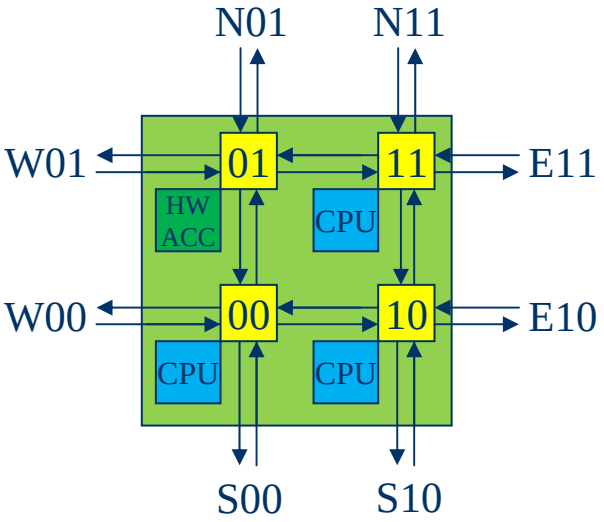
高効率チップ間通信

- 転送ミスの検出・データ再送機構の実現
  - ノイズ，追い越しによる転送データの消失に対応
  - 符号語検出部が規定時間内に完了しない場合
    - 転送符号の消失と見なして送信側に再送を要求

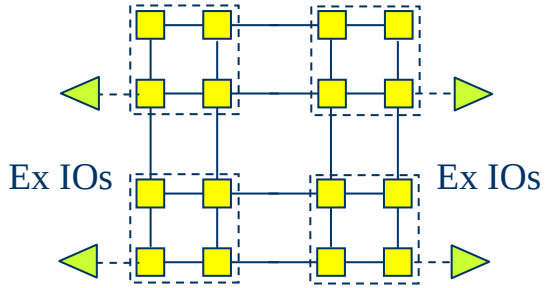


第二次実証用モデル（予定）

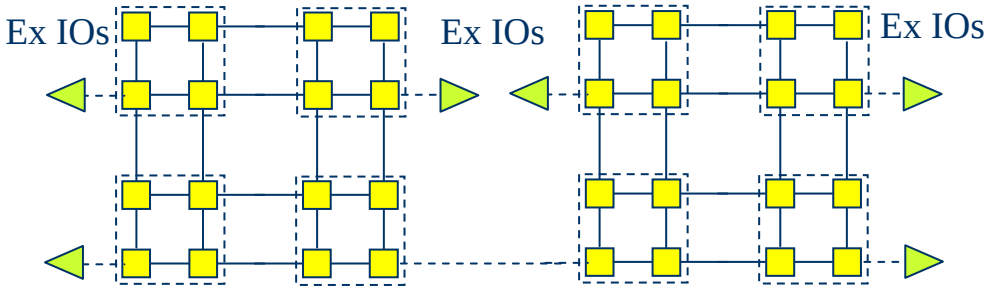
- ASICによる試作予定
  - 4コア（V850×3, FPU×1）
  - チップ間通信を用いて拡張



16 Cores
(12CPU, 4FPU)
Tight coupling
4 Ex IOs



32 Cores (24CPU, 8FPU)
Loose coupling 6 Ex IOs

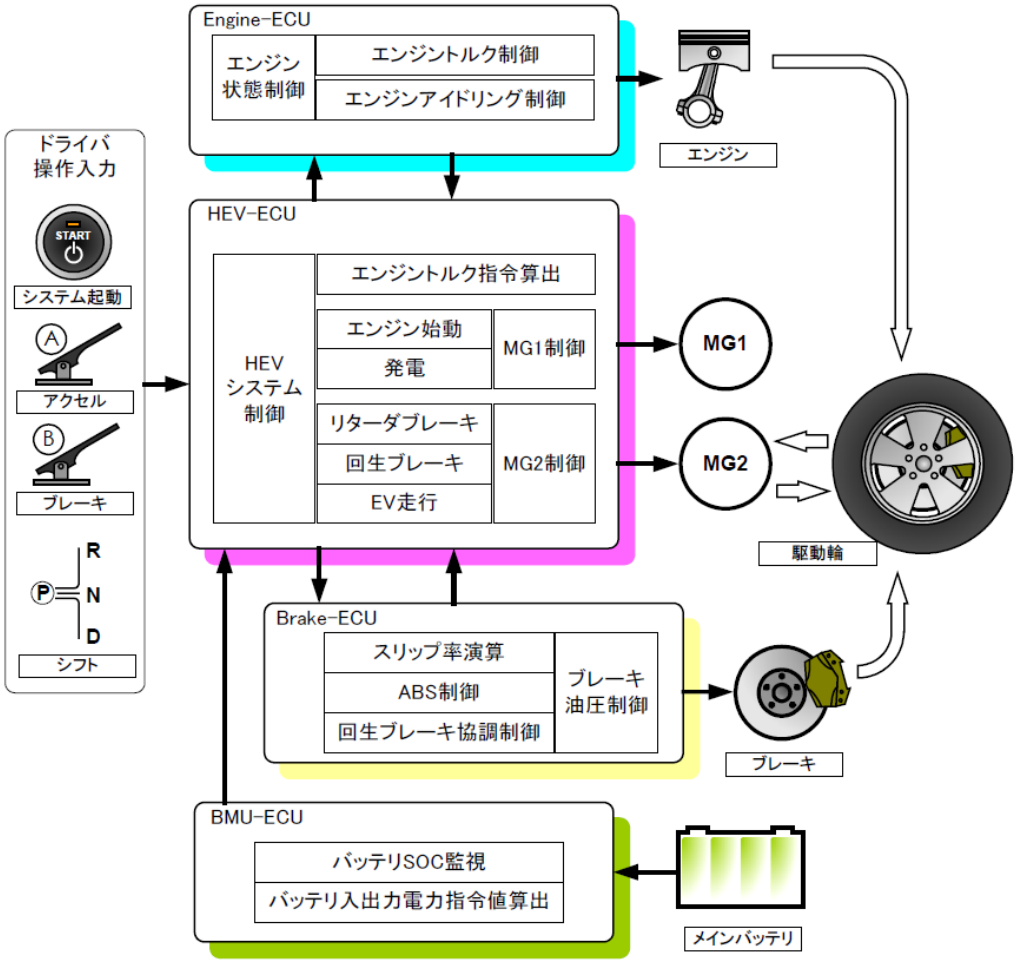


第二次実証用モデル(予定)

- 実証用アプリケーション
  - ガソリンエンジン制御(点火タイミング，燃料噴射量，燃料噴射タイミング等の計算)
  - 動力用モータ制御(必要なトルクを生成するための制御入力計算)
  - 発電機・スタータ制御(必要なトルクを生成するための制御入力計算)
  - バッテリ制御(放電量，充電量計算)
  - 駆動輪スリップ制御(ブレーキ液圧およびエンジン・駆動用モータトルク値計算)
  - 上記の連携(アクセル／ブレーキペダル値，車速から決められた状態遷移に基づきエンジン・動力用モータのトルク制御，発電・放電制御等を行う)

第二次実証用モデル(予定)

- 実証用アプリケーション



まとめ

- ディペンダブルNoCプラットフォームの特徴
  - ルータ、リンク故障
    - ディペンダブルルーティングにより迂回
  - 故障ルータに接続のコア、故障コア
    - 冗長なタスク割り当てにより、他コアで実行
  - チップ間通信によるシームレスな拡張
    - 小さなコストでさまざまな規模に対応
  - さらに非同期オンチップネットワークでは
    - ばらつきや劣化に強い
    - 複雑なルーティングアルゴリズムに対しても性能低下が小さい
    - シンクロナイザによるオーバヘッドなし