

Distributed Embedded Control Systems

南山大学大学院

数理情報研究科

M2012MM002

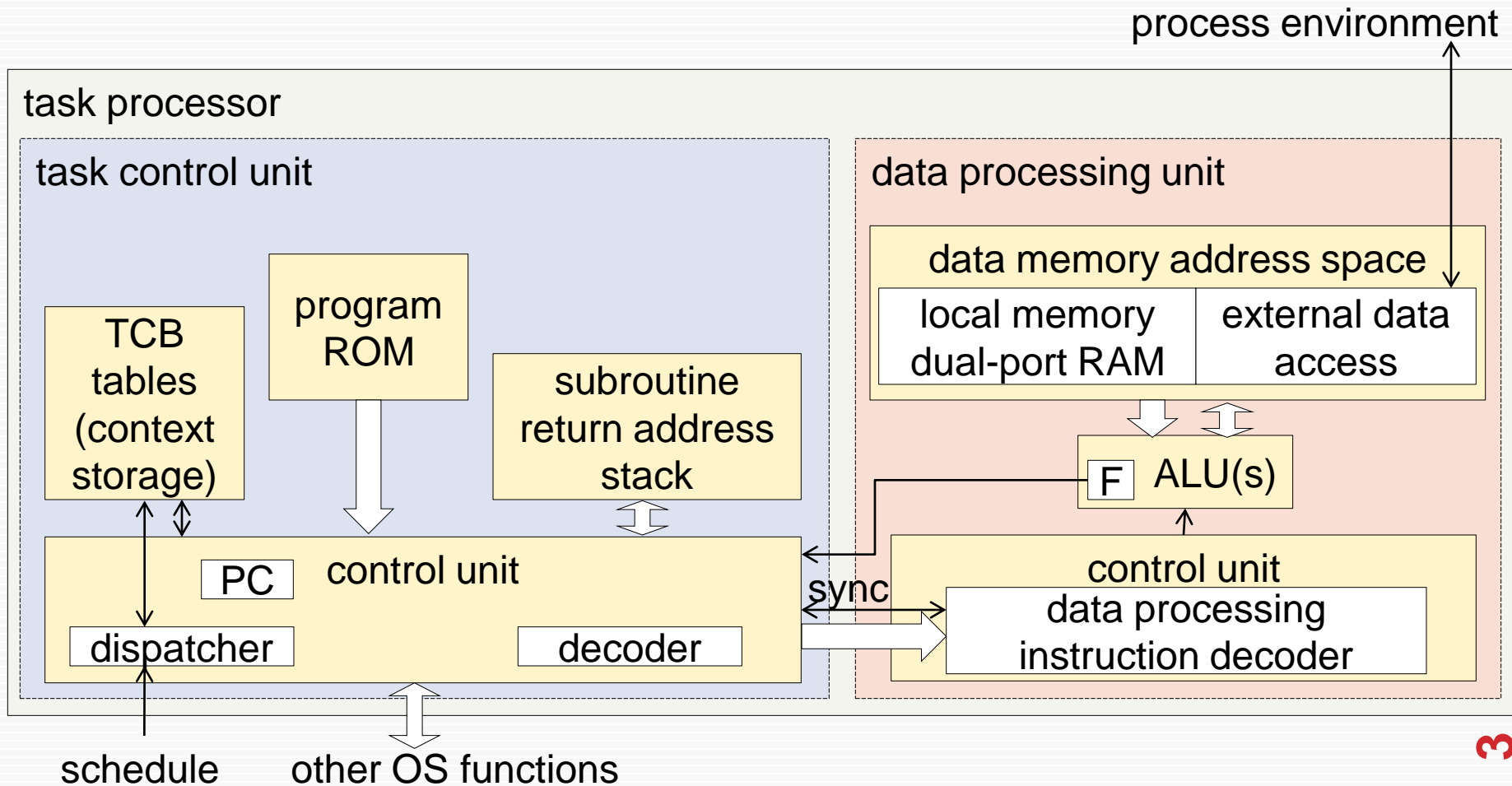
朝倉知也

シナリオ

- タスクプロセッサ
- アーキテクチャモデルの実現
- 周辺インタフェースの改善

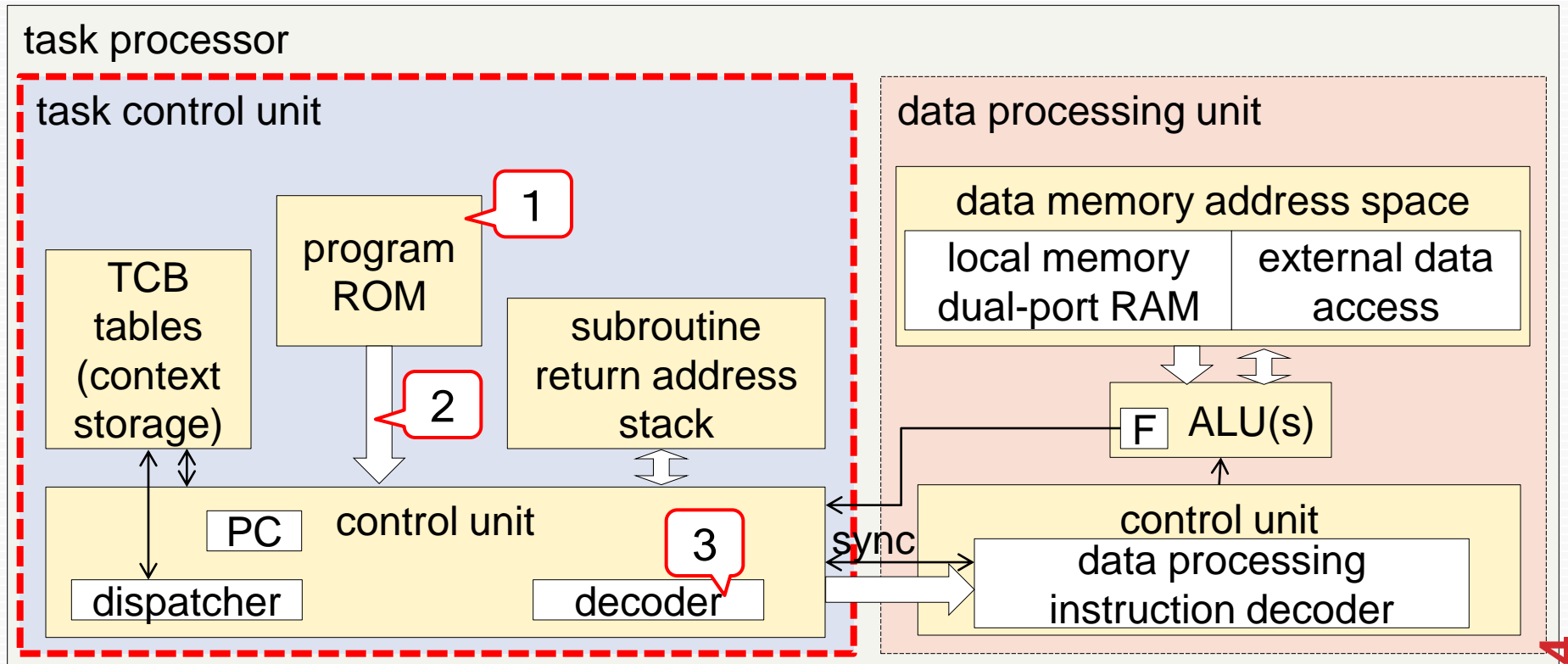
タスクプロセッサ

- ハードウェアアーキテクチャ: プログラムとデータメモリを分割
task control unit : プログラムにアクセス
data processing unit : データメモリにアクセス



タスクプロセッサ

- task control unit
 - プログラムの書換え防止のためROMに保存
 - プログラムメモリ読み込み
 - プログラムフロー制御とデータ処理を判別するためデコード



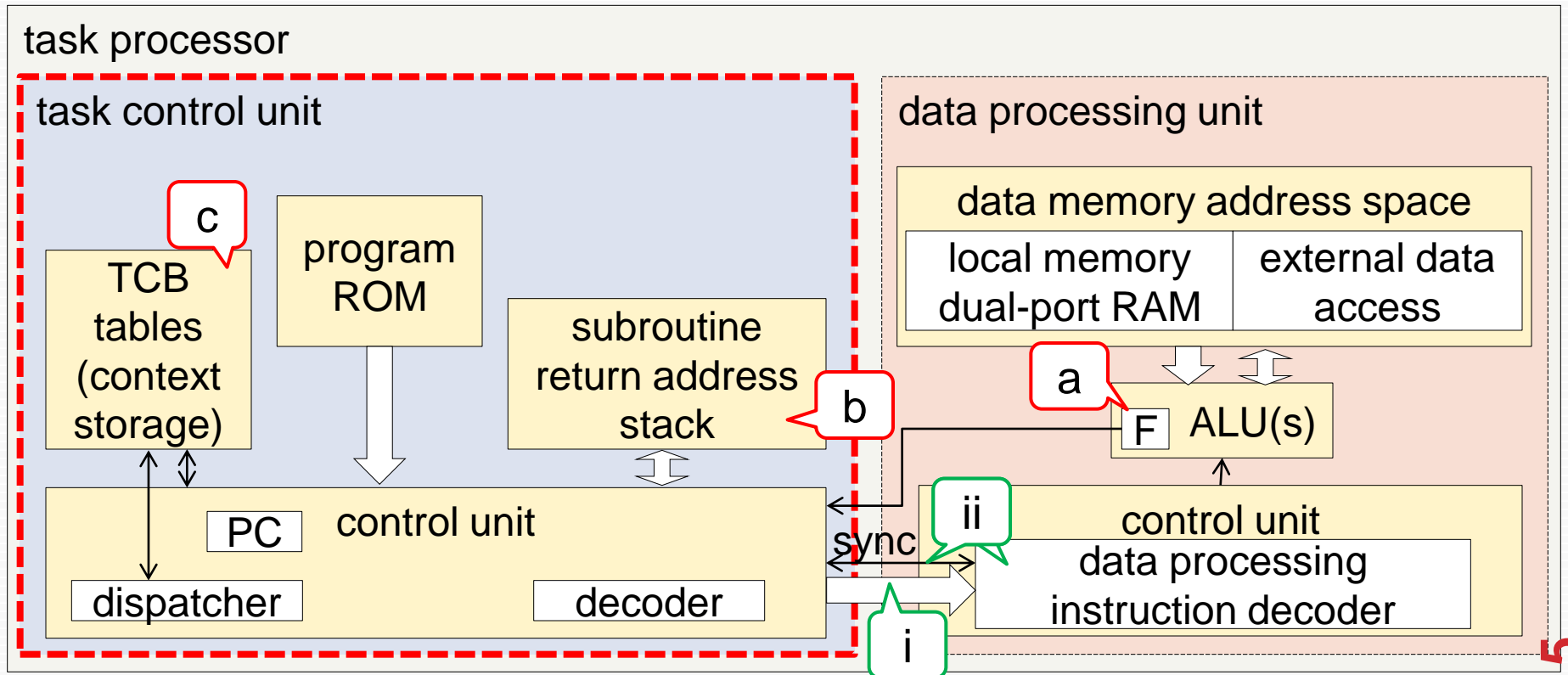
タスクプロセッサ

プログラムフロー制御命令

- 条件分岐判定にdata processing unit内のstates flagを利用
- タスクのサブルーチンリターンアドレスをstackに保存
- タスクごとのスタックポインタをTCB tablesに保存

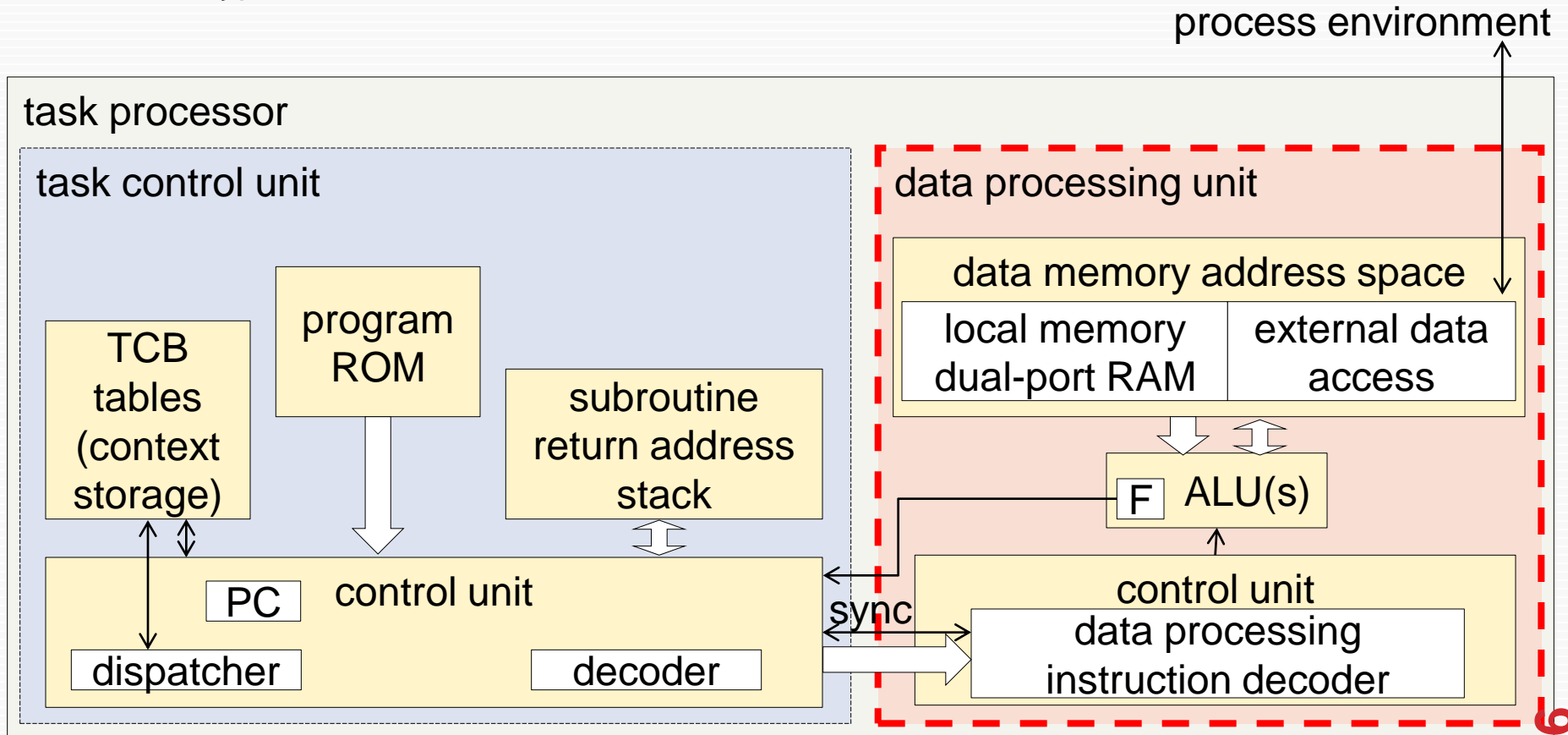
データ処理命令

- data processing unitに送信
- 同期により、データ処理中は更なるデータ処理命令の送信を防ぐ



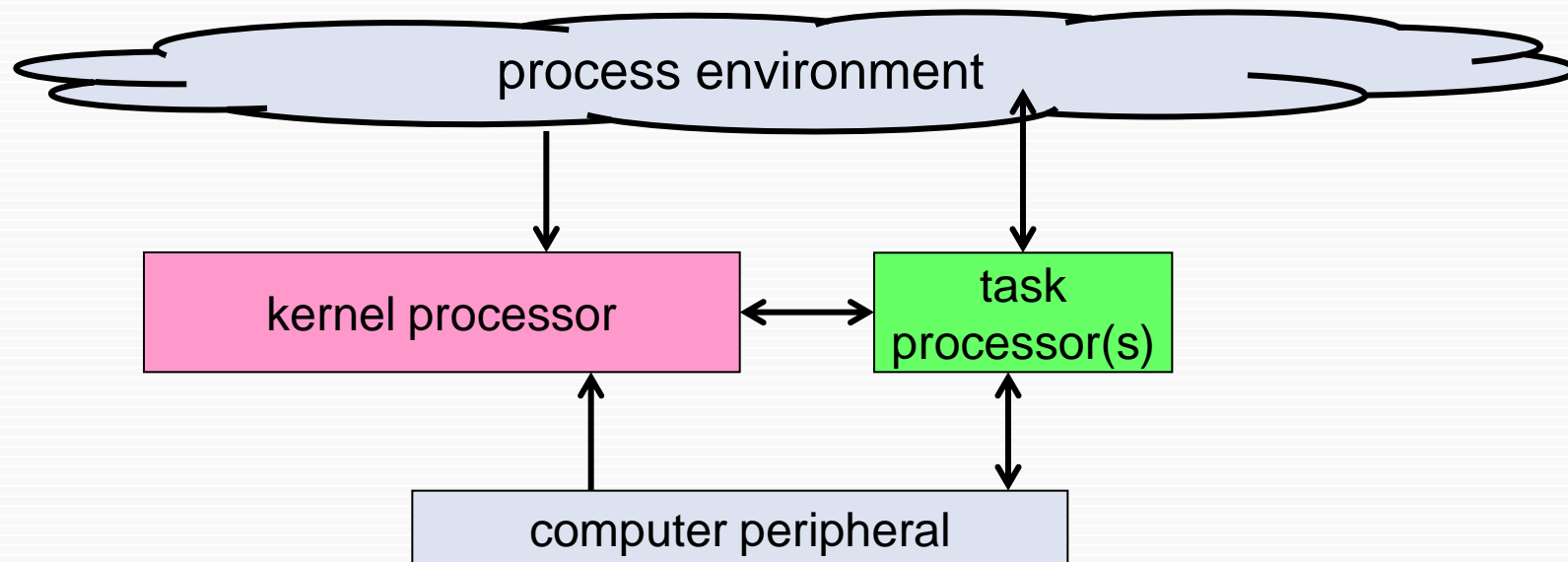
タスクプロセッサ

- Data processing unit
 - data memory address space
 - local memory : 変数と定数を保持
 - external data : process environmentと通信
 - ALU : 算術処理

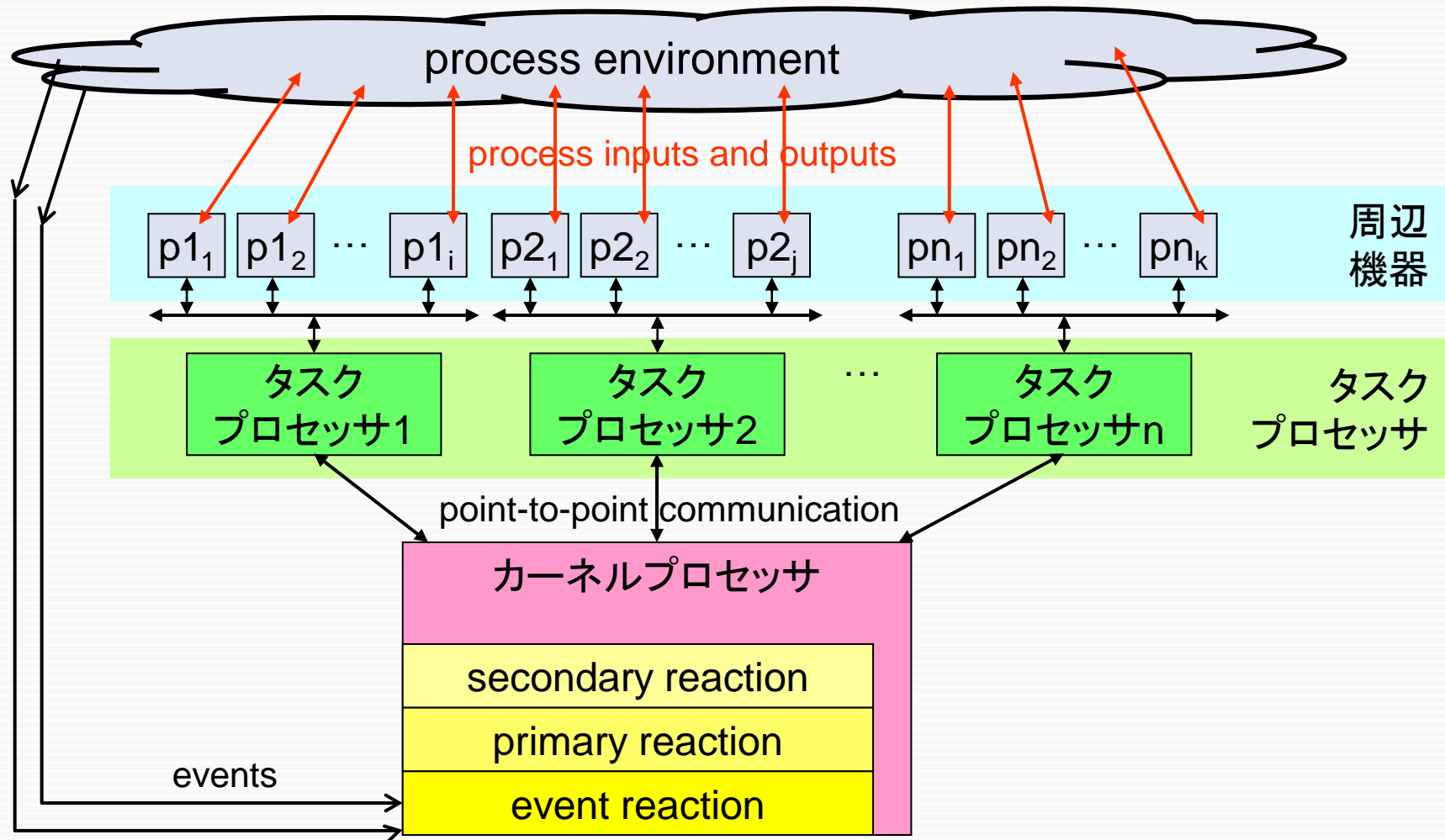


アーキテクチャモデルの実現

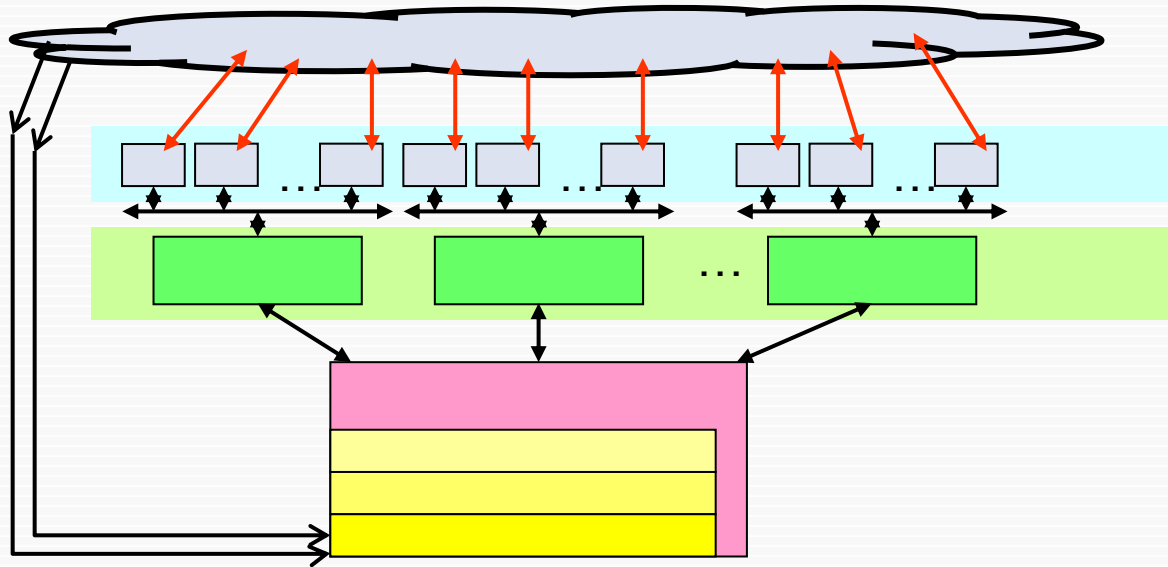
- アーキテクチャの実装
 - ✓開発と改善の歴史を3ステップで説明
 - ✓利点と問題点から、次ステップのヒントを得る



アーキテクチャモデルの実現

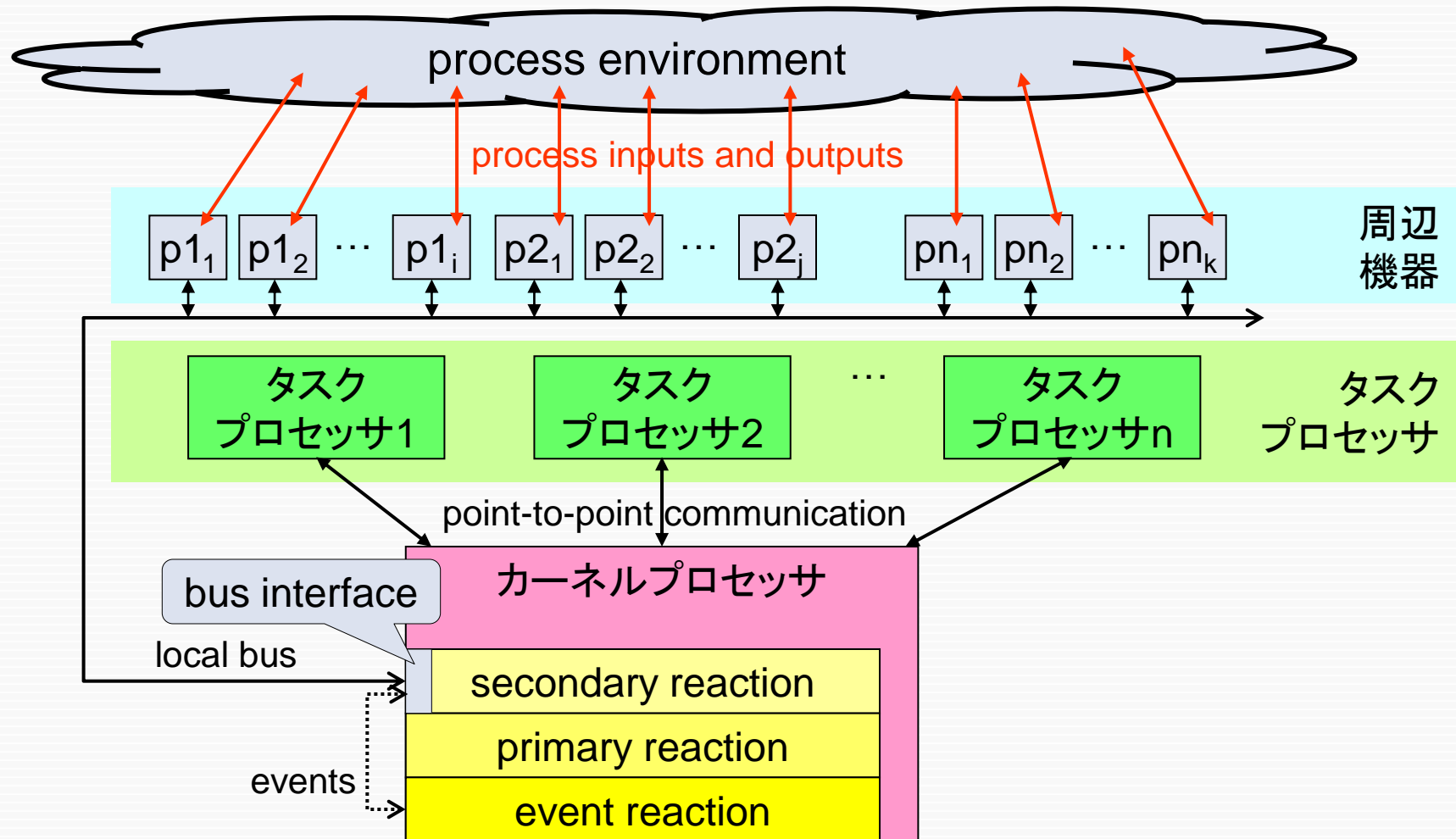


アーキテクチャモデルの実現

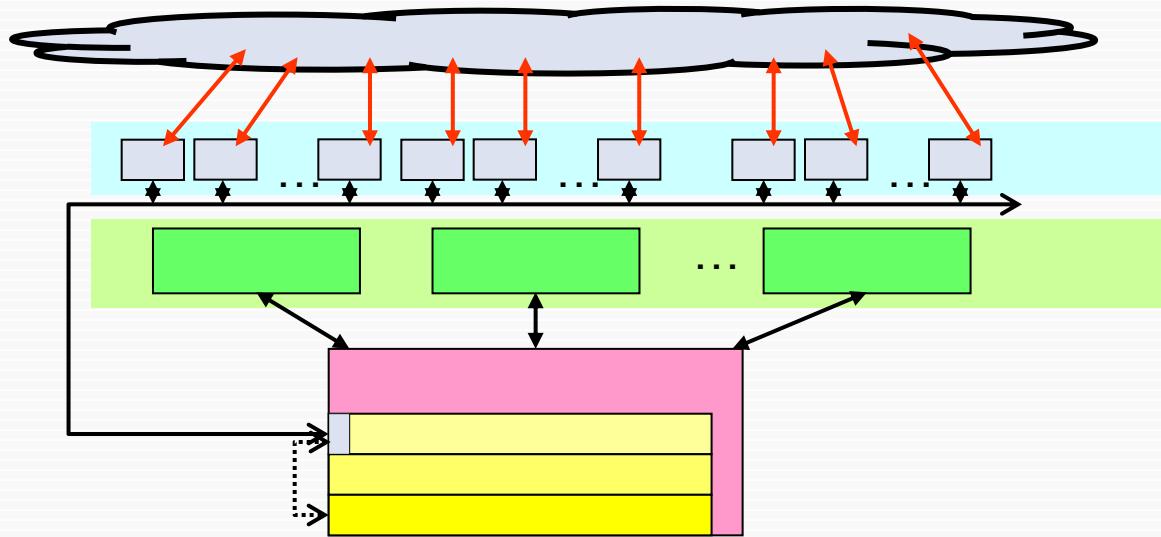


- タスクプロセッサ - カーネルプロセッサで競合が発生しない
 - ✓ タスクプロセッサは複数の周辺機器を制御
 - ✓ カーネルプロセッサとタスクプロセッサ: Point-to-Point
- process environmentからevent reaction layerまで,
平行してイベント通知を行わなければならない

アーキテクチャモデルの実現

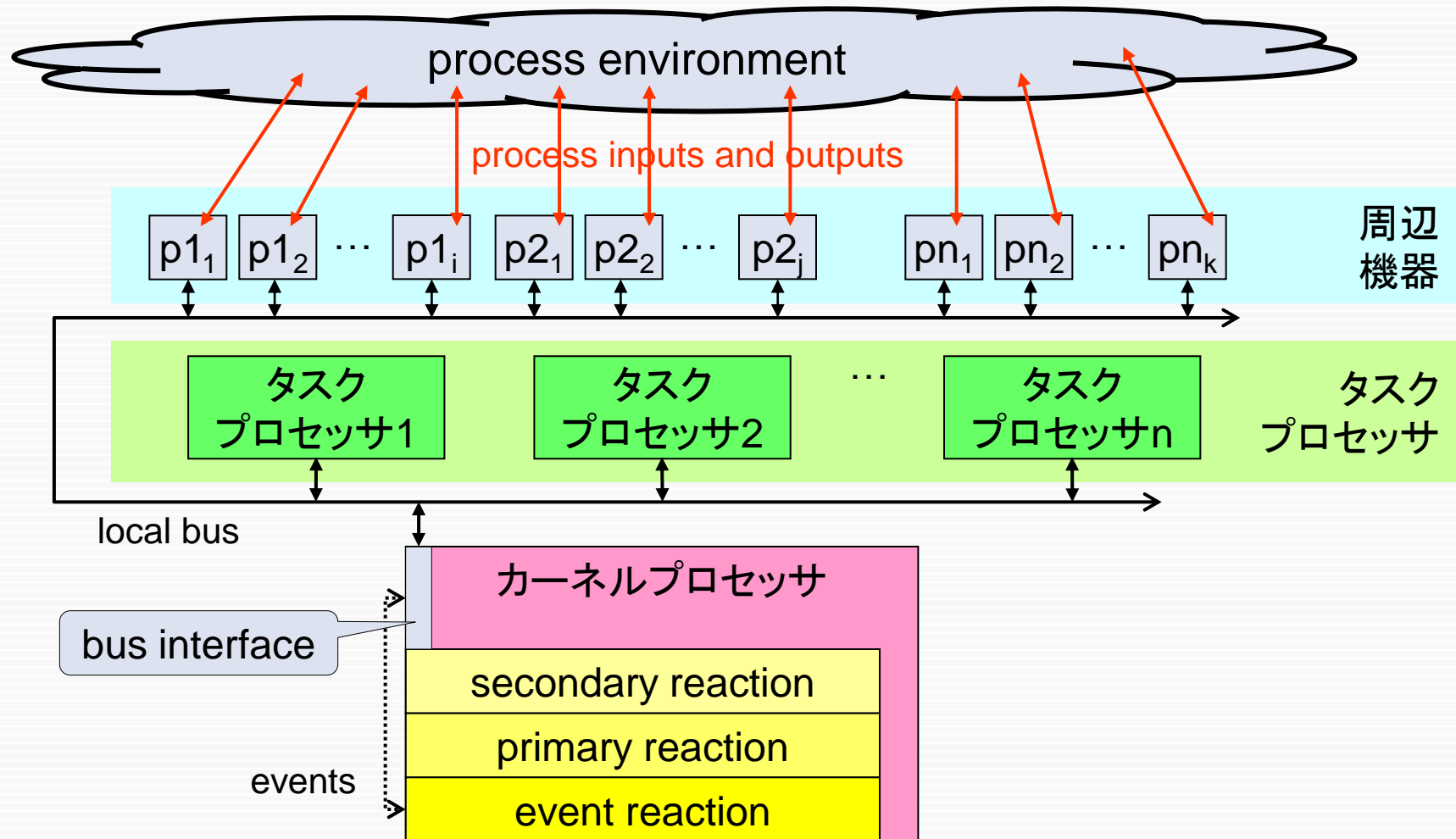


アーキテクチャモデルの実現

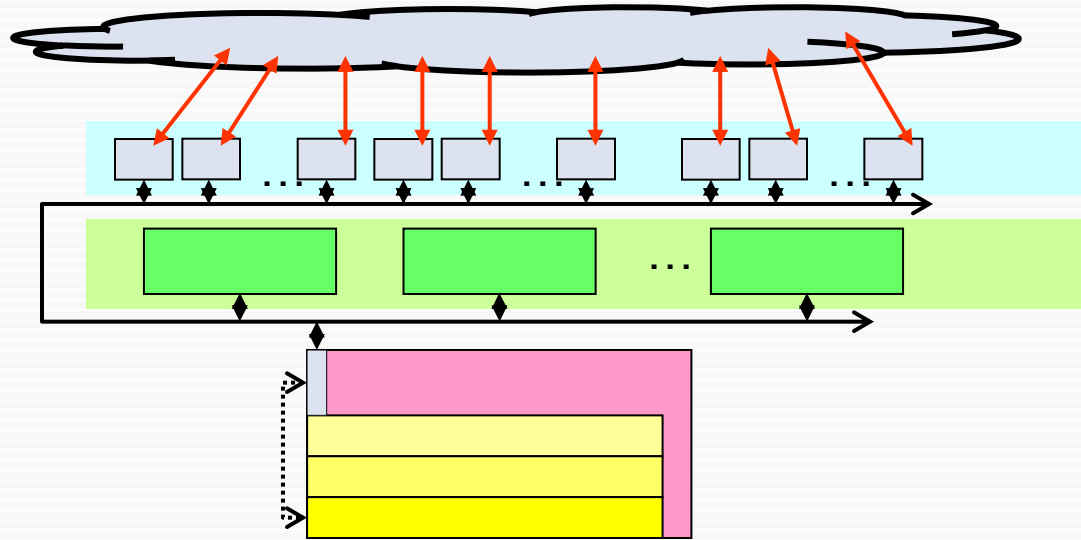


- タスクプロセッサと周辺機器はカーネルプロセッサを通じて通信
 - ✓ 多少のオーバーヘッド ⇒ 長いメッセージを送信しないので問題なし
- カーネルプロセッサによって制御
 - ⇒ 競合が発生しない
- フォールトトレランス: 問題の発生した周辺機器を動的に再構成可能
 - ⇒ しかし, **タスクプロセッサは再構成不可能**

アーキテクチャモデルの実現



アーキテクチャモデルの実現



- 周辺機器とタスクプロセッサをローカルネットワークに配置
⇒ 周辺機器とタスクプロセッサを動的に再構成可能
- ユニットの故障時に, スタンバイ状態の他のユニットが代わりに処理を行う
- タスクプロセッサは非同期で動作
⇒ 競合の可能性

周辺インタフェースの改善

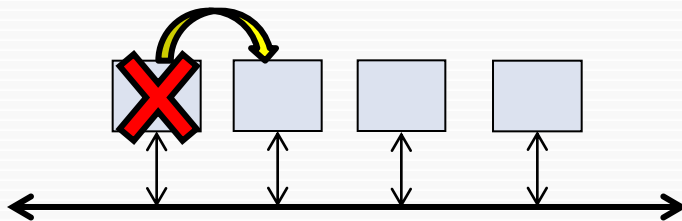
- 周辺インタフェースの改善
 - ✓ 処理の分散
 - ✓ 高水準な情報の作成
 - ✓ データ転送サイズの縮小
- 単純なマイクロコントローラの利用
 - ✓ 特定の機能を実行するようあらかじめプログラミング
 - ✓ 周辺インタフェースやI/Oルーチンの制御

周辺インタフェースの改善

- フォールトトレランス

- ✓ インタフェースで対処されるべき
 - ⇒ システム本体に影響を与えない

- ✓ 不具合が発生した場合, 他の機器に処理を移す



- ✓ 不完全な入力に対処する

⇒ 例: 温度

急激な物体の温度変化, 0K以下の温度

参考文献

- Wayne Wolf, 組込みシステム設計の基礎, 日経BP社