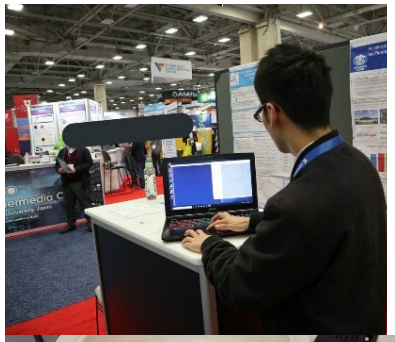
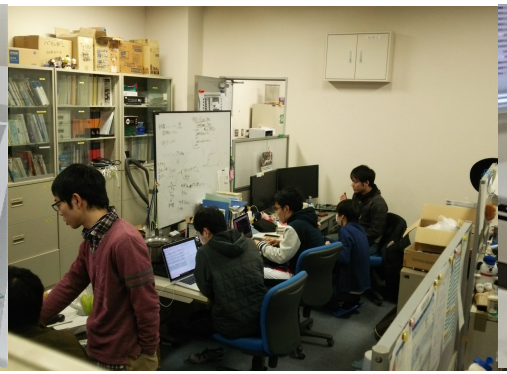
コンピュータアーキテクチャ研究グループ

弘中哲夫(教授)，川端英之（准教授），谷川一哉(講師)，
児島彰(助教)，窪田昌史（助教）

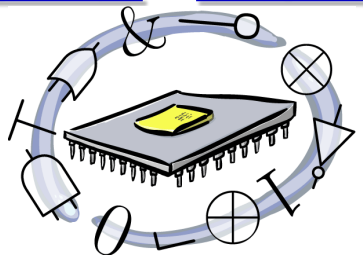


情報科学部棟5F(511,512,513,541,543)
http://www.ca.info.hiroshima-cu.ac.jp/



目標は革新的コンピューティング技術の開拓

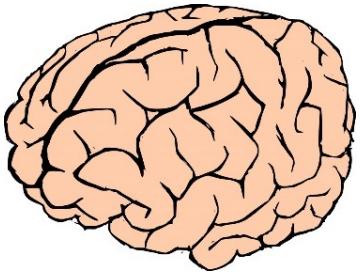
来る超スマート社会に対応できる高並列，高信頼，高精度，省電力の次世代コンピューティングと次世代プログラミング環境の実現を目指す



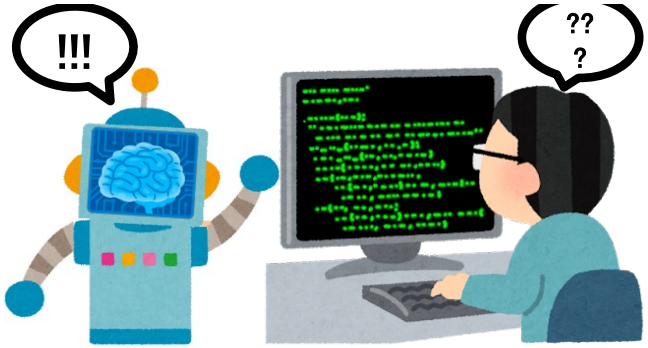
リコンフィギャラブル
プロセッサ，LSI技術

コンピュータシステム
（マルチプロセッサ，GPU等）

コンピュータアーキテクチャ



人工知能技術応用
（新しい計算処理法）



新しいプログラミング技術

さらにいろいろな
企業との連携も

ハードウェアのポテンシャルを引き出す
ソフトウェア技術に関する研究

プログラミング言語処理系

コンピュータと楽しく会話したい！

JITコンパイル環境，各種最適化，数値計算向け言語設計，
高機能インタプリタ，モバイル機器向けの言語処理系

高品質なソフトウェアに支えられたコンピューティング環境

ちゃんと動くモノ作りには欠かせません！

ユーザ支援環境

高機能エディタ，各種検証器，
デバッグ支援環境，
データ整形ツール，GUI

ズバ抜けた性能を追求！

ライブラリ

高精度計算の実現，数値シミュレーションの基本要素のチューニング，特殊ハード用ライブラリ

CやJavaだけじゃない！

色々なプログラミング言語で，
色々なプログラミングスタイルで，
時には便利なツールを設計したり，
時にはプログラムの正しさを証明したり，・・・

C言語が好きな人も，嫌いな人も，Welcome！

テーマ例「お望みの精度で計算してさしあげます」

「コンピュータは計算が得意なはずなのに，計算誤差に気をつけなくちゃならないなんて・・・」

「正確な計算結果が欲しいけど，複雑なプログラミングが必要だとすると嫌だなぁ・・・」

✓ 「実数計算」を実現するライブラリ

プログラミングの負担を大幅軽減！
ワンランク上の計算精度保証の実現！

(計算例)

$$333.75b^6 + a^2(11a^2b^2 - b^6 - 121b^4 - 2) + 5.5b^8 + a/(2b) \quad \textbf{where} \quad a = 77617, b = 33096$$

普通の方式： -1.180591620717411 3e21 （倍精度演算だと全くデタラメな値になる）

我々の方式： -0.827396059946821368141165095479816291 9 ... (何桁でもOK)

テーマ例「プログラミングを楽にするプログラミング」

「高性能を手に入れるには，新しい言語やライブラリを頑張って学ぶしかないのかなぁ・・・」

「昔からある有名なライブラリは性能はいいけど決して使い易くはないんだよなぁ・・・」

- FFIによる言語を超えた連携
- トランスレータ，DSL
- 組み込み環境向けシステム
- GUI

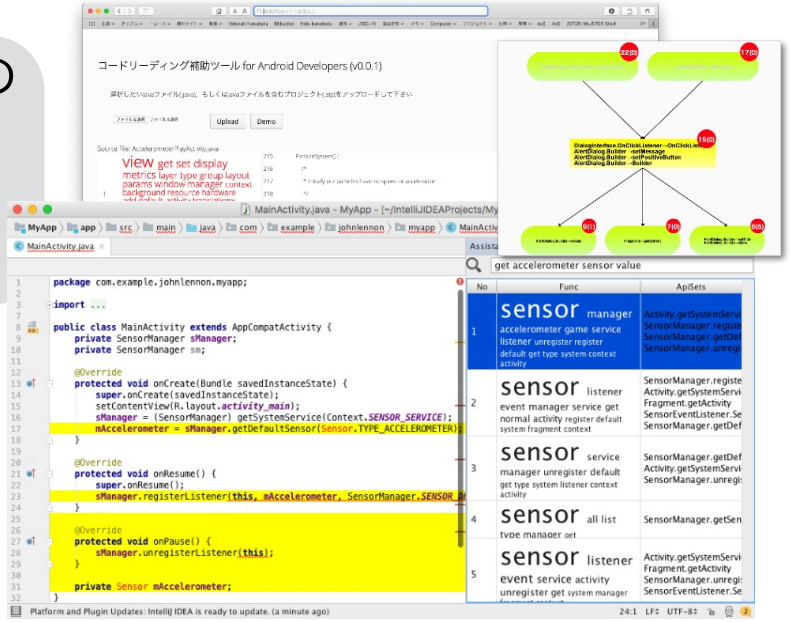
✓ 既存の道具や言語にとらわれないプログラミング環境を目指して

高性能ライブラリのバインディング
➡ 既存の言語でできることを増やす！

いつものツールを高性能化
➡ コノ環境でアレができる時代がきた！

テーマ例「ソフトウェア開発手法の未来形を追求」

「本格的なアプリ開発をしてると，開発中のコードの規模に圧倒されて，埋もれちゃう気分・・・」
「新しいフレームワークがどんどん出て来るから，最新情報を追いかけるのが大変なんだよね・・・」

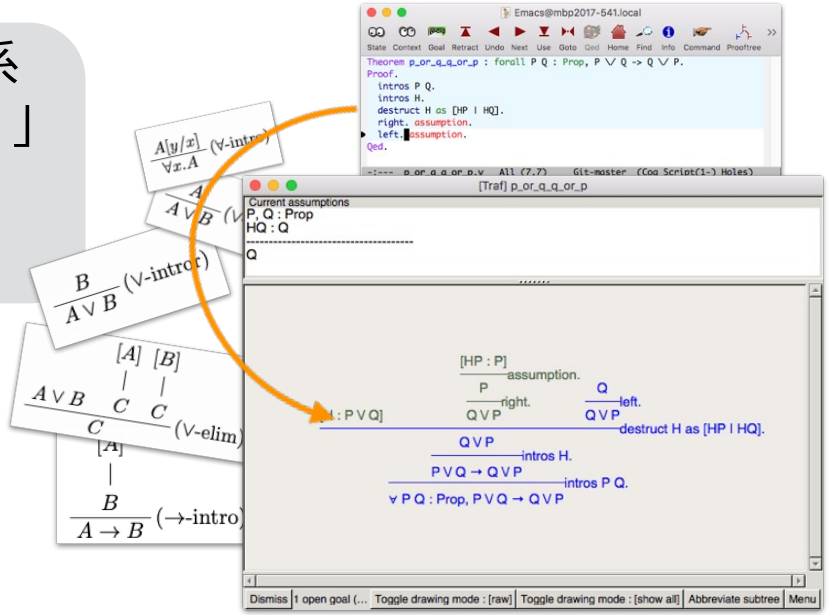


✓ ひらめきベースプログラミング環境

面倒はツールに任せて人はアイデアに集中！
➡ 検索＆合成おまかせ，マニュアルいらず！

テーマ例「定理証明のお手伝いをいたしましょう」

「ソフトウェアの仕様検証のために定理証明支援系が使われつつあるとは聞いたけど，難しい・・・」
「既存の定理証明支援系は，高機能だけれど，使いにくい・・・」



✓ 高機能な証明支援系GUIを実現

証明支援系Coqのインターフェース
数理論理学やプログラミング言語理論入門に

研究テーマ：高速化したい！

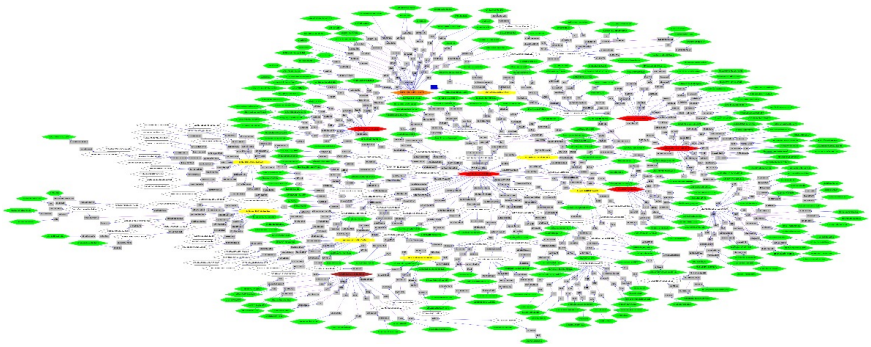
- コロナ対策においてシミュレーションは大活躍！（理研の坪倉先生のスライド）
- これらのシミュレーションができたのは，スパコン「富岳」の高い性能のおかげ！

「高速化」は研究テーマの1つ
「何を」高速化するか？
「どうやって」高速化するか？

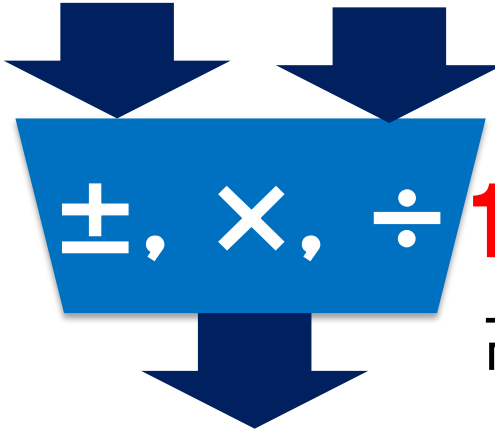
「何を」高速化するか？

- グラフ探索
  - ニッチ，でも，今後重要性が高まりそう！
  - 量子コンピューティングの手助けになるかも！
- 高精度数値演算＋区間演算
  - 超ニッチ，でも，間違いなく需要はある！
- 量子コンピューティング
  - 最先端のコンピューティング．30年後の主流!?

数時間以上かかる計算を数分以内にしたい

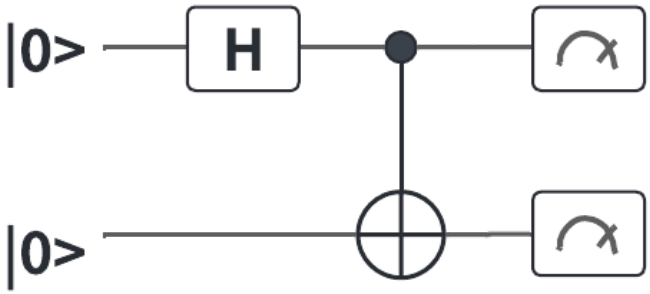


グラフ探索



1024 bit

高精度演算



量子回路の例

「どうやって」高速化するか？

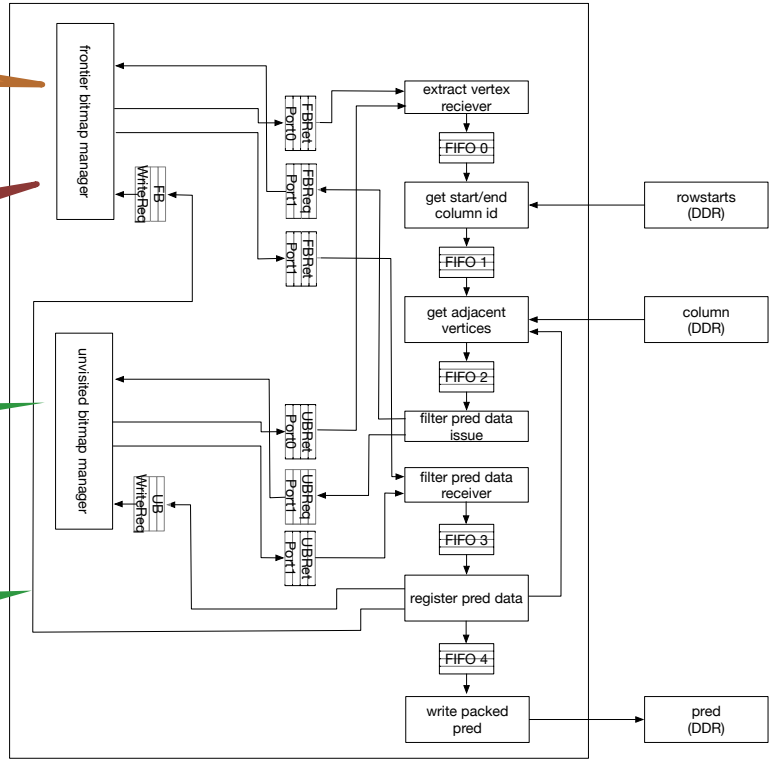
- FPGA上に「専用回路」を作成することで実現
  （FPGA: 論理回路をソフト的に実現）

アルゴリズム理解

ボトルネック解析

キャッシュの応用

パイプライン
並列化



幅優先探索専用ハードウェア

アルゴリズムからハードウェアまで
コンピュータの仕組みを熟知したエンジニアになれる！

プログラムの実行の高速化・低消費電力化

アプリケーション
プログラム

シミュレーション
自動運転
機械学習
etc…

アーキテクチャに適した
プログラムの特徴は?

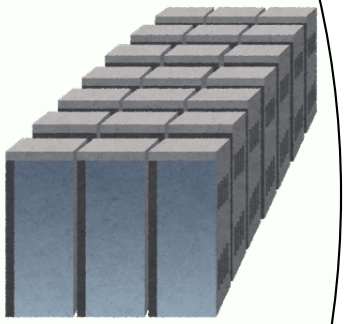
高速化・
低消費電力化

アーキテクチャ間の
プログラムの書き換えは?

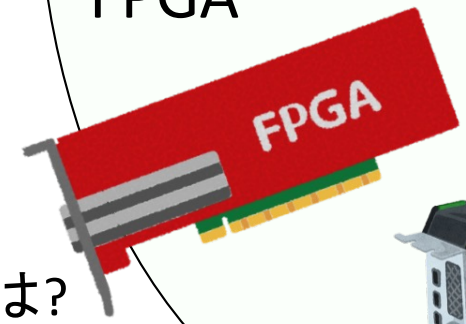
アーキテクチャ

量子
コンピュータ

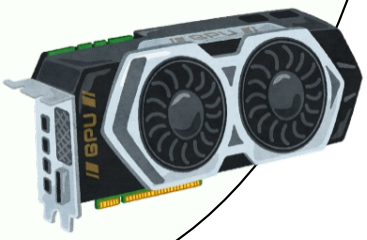
スーパー
コンピュータ



FPGA

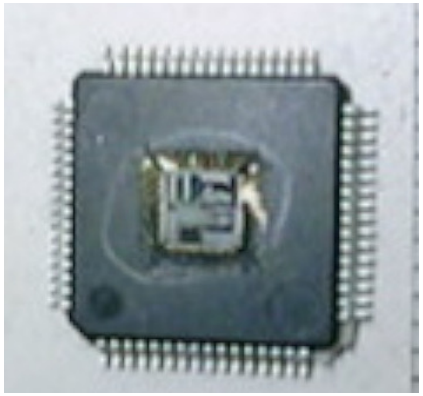


GPU



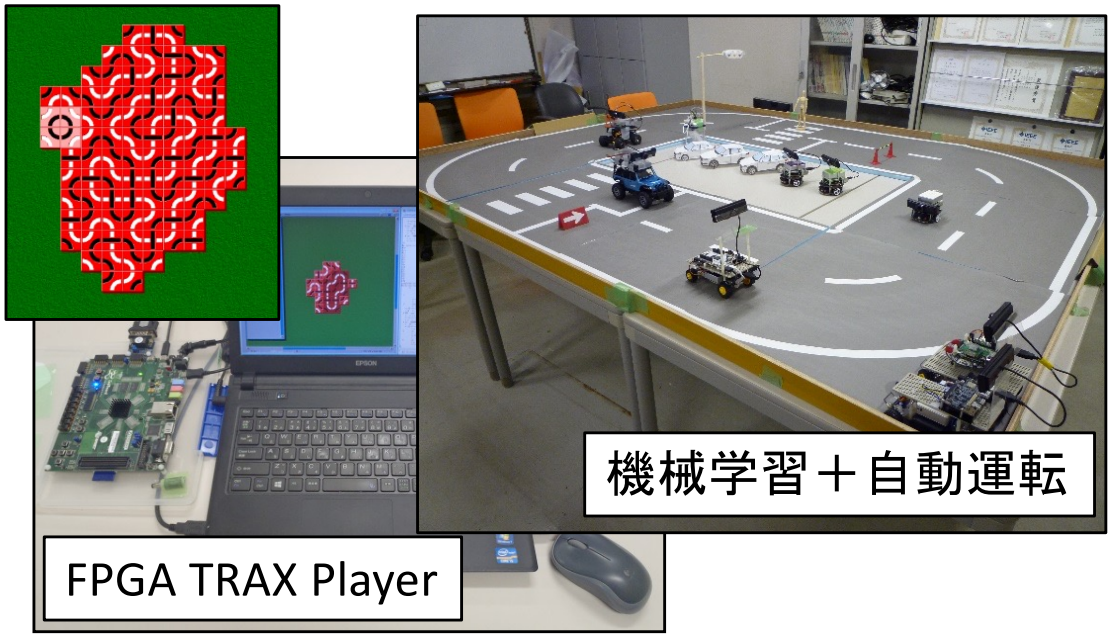
Thanks to いらすとやさん, 咲っとイラストさん, @jun_ando(twitter)さん

求める人材　モノ作り（ソフト，ハード）をしたい人



スペシャルなCPUを設計，チップ化

様々な応用・アプリ開発



ゲームAI・自動運転（ハード・ソフト協調設計）

身につく技量　システム設計・実装能力

ハードウェアとソフトウェアが連携動作する装置の研究開発

並列ハードウェアを動かすシステムソフトウェア／医療機器のデータ処理装置／実時間シミュレータ／ロボットのセンサーシステムと制御システム／機械学習ハードウェア／自動運転ロボットカー

システム開発に必要な基礎体力（ハードもソフトも）

プロトタイピング，デバッギング，モジュール化，仕様検証，テスト，シミュレーション，性能測定・・・．

様々なパラダイムに基づく設計手法の体得

手続き型言語，スクリプト言語，関数プログラミング，オブジェクト指向，・・・．

ハードとソフト両方できるエンジニアに！