



# OpenCL/SYCLを中心とする ソフトウェア開発環境エコシステムへの期待

西村 成司

---

19<sup>th</sup> Nov., 2021

Semiconductor IP R&D Unit, NISTEXE

# アジェンダ

## 1. 会社紹介

2. NSITEXE次世代DFP向けSDKの構想
3. アクセラレータ開発環境エコシステムへの期待
4. AIアクセラレータへの応用例
5. まとめ

# 会社紹介

Create the future by our semiconductor IP

- **半導体IP、半導体ソリューションカンパニー**

- 2017年 (株)デンソーの出資により設立
- 世界各地から集まった60名以上のエンジニアが在籍

- **エンベデッドシステム向け高効率半導体IP (DFP) を開発**

- 2020年1月 最初のDFP製品DR1000Cをリリース
- RISC-Vプロセッサ、AIエンジン、セキュリティサブシステム等も開発

- **エコシステム**

- 自動運転等の自動車向けシステム、ロボティクス、FA、IoT等、次世代を担う様々な分野のアプリケーションをサポート
- HW/SW、デザインメソドロジ、ソリューション各社との幅広いパートナーシップ



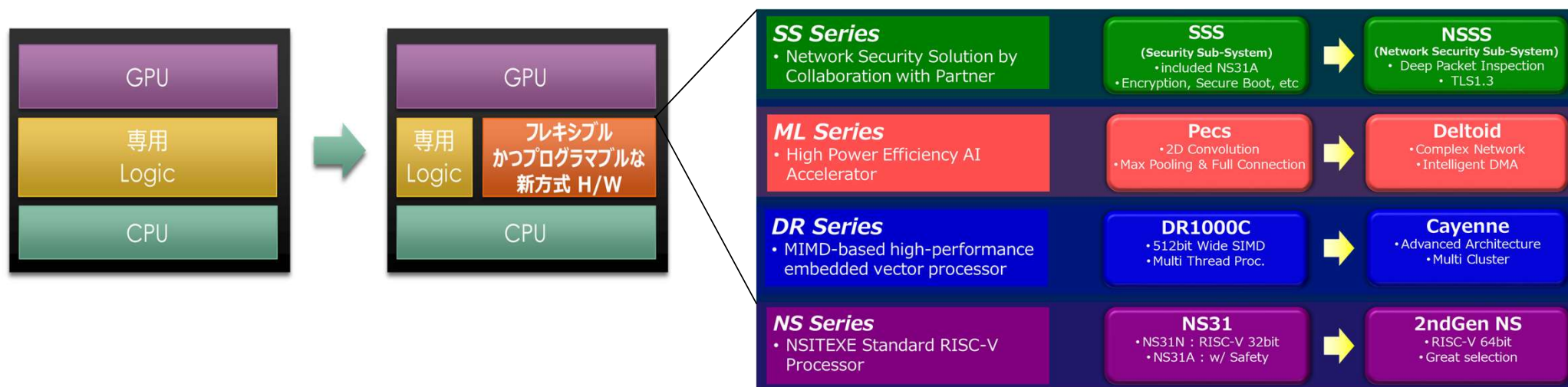
# アジェンダ

1. 会社紹介
- 2. NSITEXE次世代DFP向けSDKの構想**
3. アクセラレータ開発環境エコシステムへの期待
4. AIアクセラレータへの応用例
5. まとめ

# 我々の目指すところ

エコシステムの穴を埋める（例え儲けにくくても…）

→ 既存領域の延長線上の領域は他社にお任せし、特に専用 Logic からプロセッサへの移行が難しい部分をシステム・プロセッサ両知見を活かし推進



## OpenCL

を中核に据え、高位インターフェースとして

## SYCLとOpenMPオフローディング

の提供を検討

# NSITEXE次世代DFP向けSDKの構想

OpenMP

OpenMPディレティブが挿入された標準Cコード

```
#pragma omp target map(to:A) map(from:B)
while (error > tol && ++iter < iter_max) {
    error = 0.0;
#pragma omp parallel for collapse(2) reduction(max:error)
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n; j++) {
            B[i][j] = 0.25 * (A[i-1][j] + A[i][j-1] +
                            A[i][j+1] + A[i+1][j]);
            error = max(error, fabs(B[i][j] - A[i][j]));
        }
    }
    ...
}
```

OpenCLを核に据える、ベンダニュートラルで標準的なプログラミング環境を実現

ユーザのアプリケーション資産への投資を活かすためにも標準準拠が重要

SYCL™

ホストコードとカーネルコードが同一ソースコード上で記述されるC++環境

OpenCL™

アクセラレータ基盤

RVV命令を備える次世代DFP

KRONOS GROUP

RISC-V

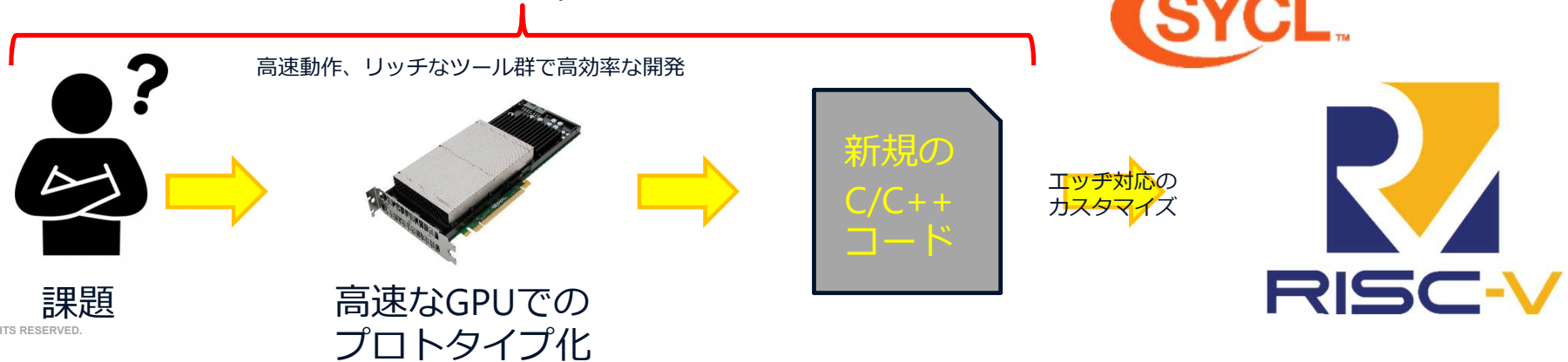
# 高位インターフェースの想定されるユースケース

- 検証済みのコード資産の活用



- スクラッチから書く新しいアプリケーションへの適用

特定ベンダにロックオンされないオープン標準かつ  
マルチプラットフォームなC/C++ベースの開発環境





# NSITEXE次世代DFP向けSDK

OpenMP™

SYCL™

多様なユーザニーズに合わせて  
高レベルAPIから低レイヤAPIまでカバー

アクセラレータ基盤

低レイヤ高性能

RVV命令を備える  
次世代DFP

RISC-V

# NSITEXE次世代DFP向けSDKの開発

- 京都マイクロコンピュータ（株）、Codeplay Ltd.をパートナーとして、商用化を前提としてSDKの開発をスタートさせています
- OpenCLをバックエンドとするSYCL処理系
- OpenCLレイヤでのRVV命令への対応
- RVV命令対応LLVM/clangコンパイラ



～認知度が高まっているRISC-Vプロセッサに、OpenCL™とSYCL™を実装することで、既存のHPCおよびAIソフトウェアの組み込みシステムへの移植を容易にします～

2020年10月30日  
株式会社エヌエスアイテクス

2020年10月30日 - 革新的なプロセッサIPを開発する株式会社エヌエスアイテクス（以下、NSITEXE）は、本日、NSITEXEと京都マイクロコンピュータ株式会社（以下、KMC）が参加する日本の新エネルギー・産業技術総合開発機構（以下、NEDO）のプロジェクトにより組み込みシステム向けのHPCやAIに取り組むソフトウェア開発者がRISC-Vアーキテクチャ上でThe Khronos Groupの業界定義のオープンスタンダードを利用できるようになると発表しました。

# アジェンダ

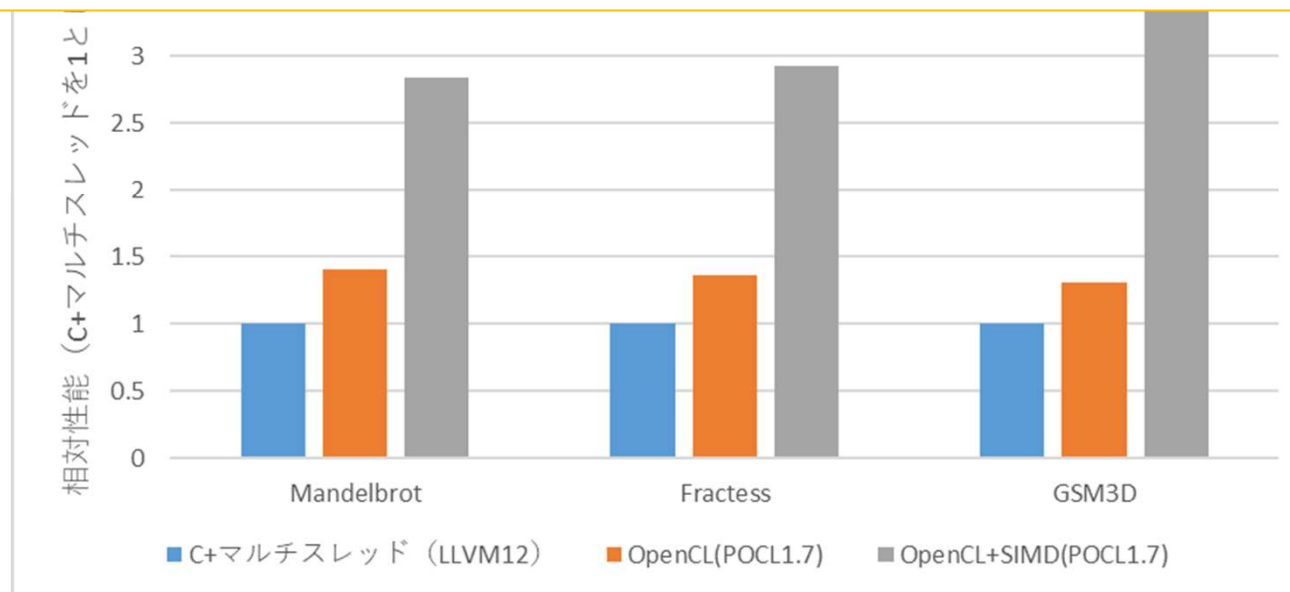
1. 会社紹介
2. NSITEXE次世代DFP向けSDKの構想
- 3. アクセラレータ開発環境エコシステムへの期待**
4. AIアクセラレータへの応用例
5. まとめ

# 汎用プロセッサでのOpenCLの有効性

素朴な疑問：汎用プロセッサでOpenCLは有効？

X86 (AVX2) での性能比較例

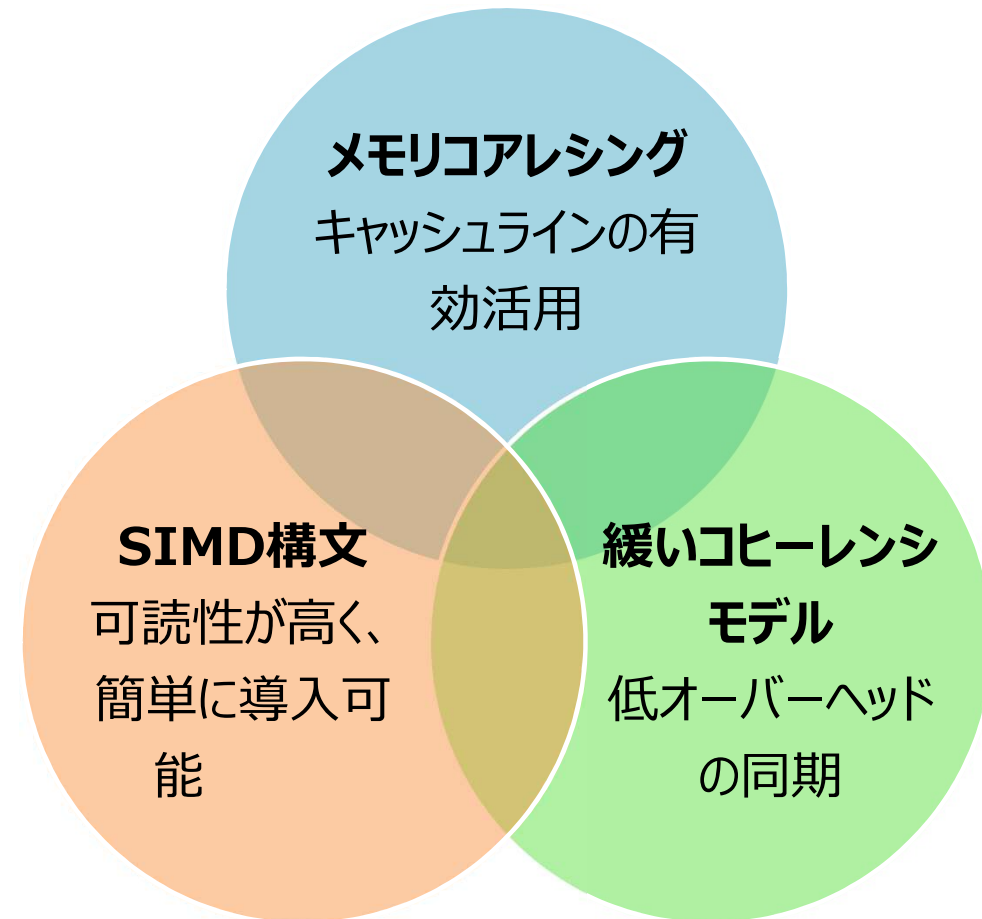
**結論：汎用プロセッサでもOpenCLは十分有用である**



Mandelbrot, Fractessは画像生成、GSM3Dは三次元差分法陽解法

# 何故、OpenCLが速いのか？

- 主に、右に挙げる三つの要因が考えられる
- …しかし、低レベルAPIであり、ユーザ視点から見ると煩雑な点も多い
  - デバイスの初期化
  - カーネル関数の呼出し



# アクセラレータプログラミングの課題

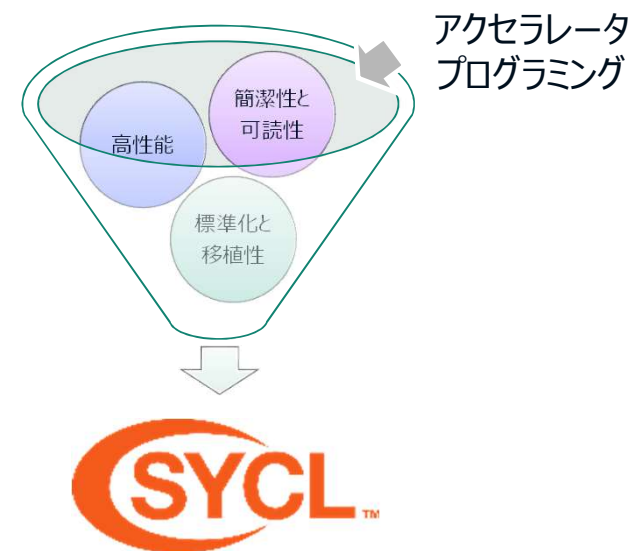
- システムが複雑化するにつれ、アプリケーションの構造も複雑化の一途
- **ユーザは「本質的な処理」に集中**し、隠蔽できるものは可能な限り隠蔽したい
  - 同一ソースコードでのホストコードとデバイスカーネル関数の記述
  - C++オブジェクトによる処理の隠蔽
    - デバイスの初期化
    - ホスト-デバイス間のデータ通信



# SYCLへの期待

SYCL=アクセラレータプログラミング+  
可読性+高性能+標準化

- C++標準規格化と連動し、将来的にC++へ統合される計画
- タスクグラフに基づく並列性の抽出
- 特定のベンダに依らないオープン規格
  - コードの移植性を確保
- ユーザが本質的な処理に集中できる



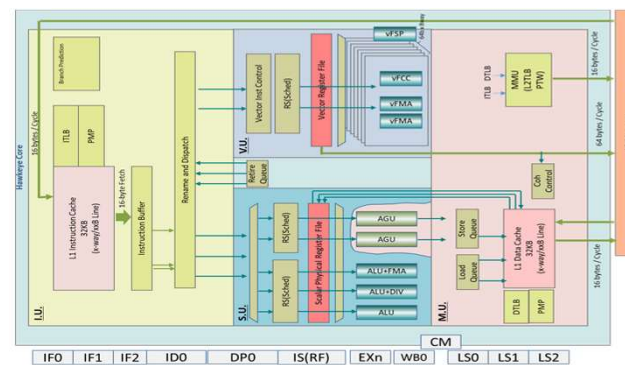
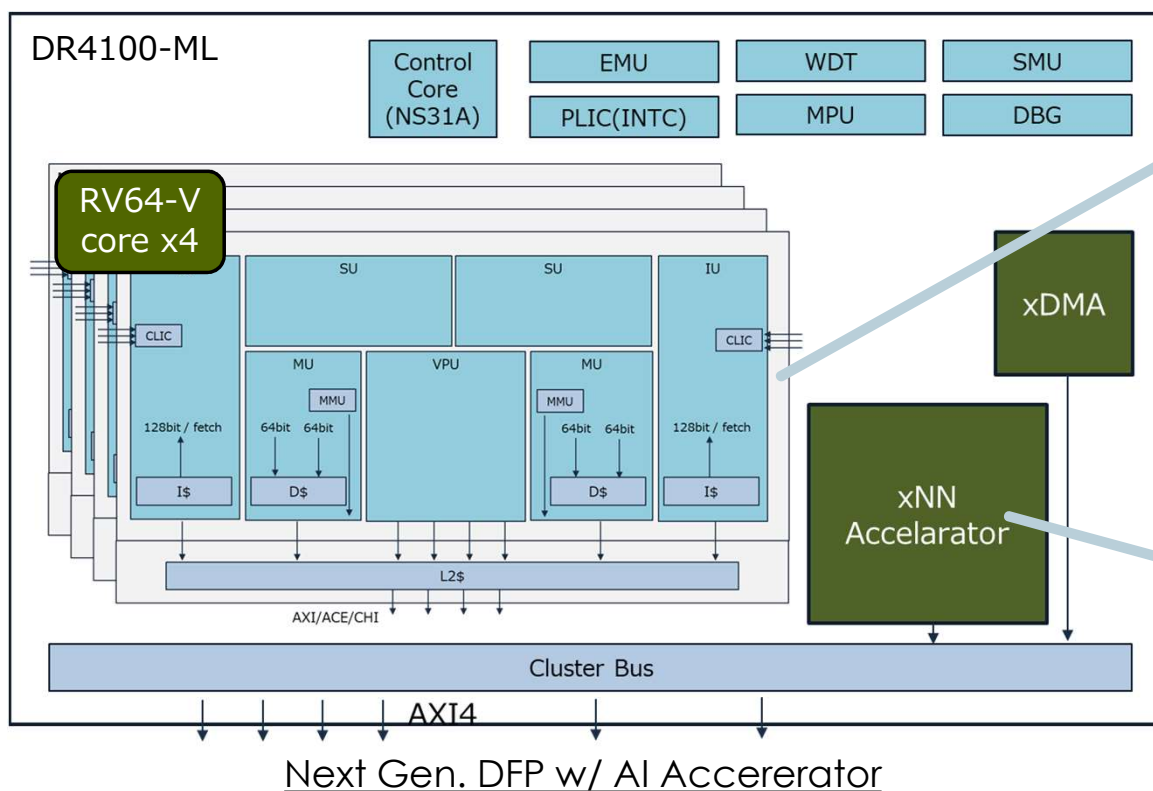
# アジェンダ

1. 会社紹介
2. NSITEXE次世代DFP向けSDKの構想
3. アクセラレータ開発環境エコシステムへの期待
- 4. AIアクセラレータへの応用例**
5. まとめ

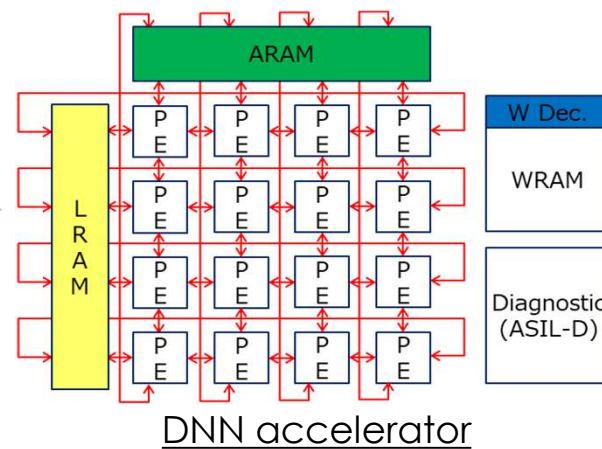


# RISC-V, AI の応用例 – アクセラレータ

組込の特性に向けたアクセラレータも

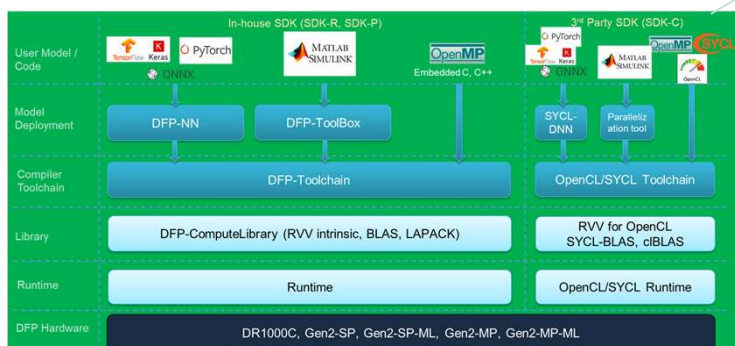


RV64 Processor Core

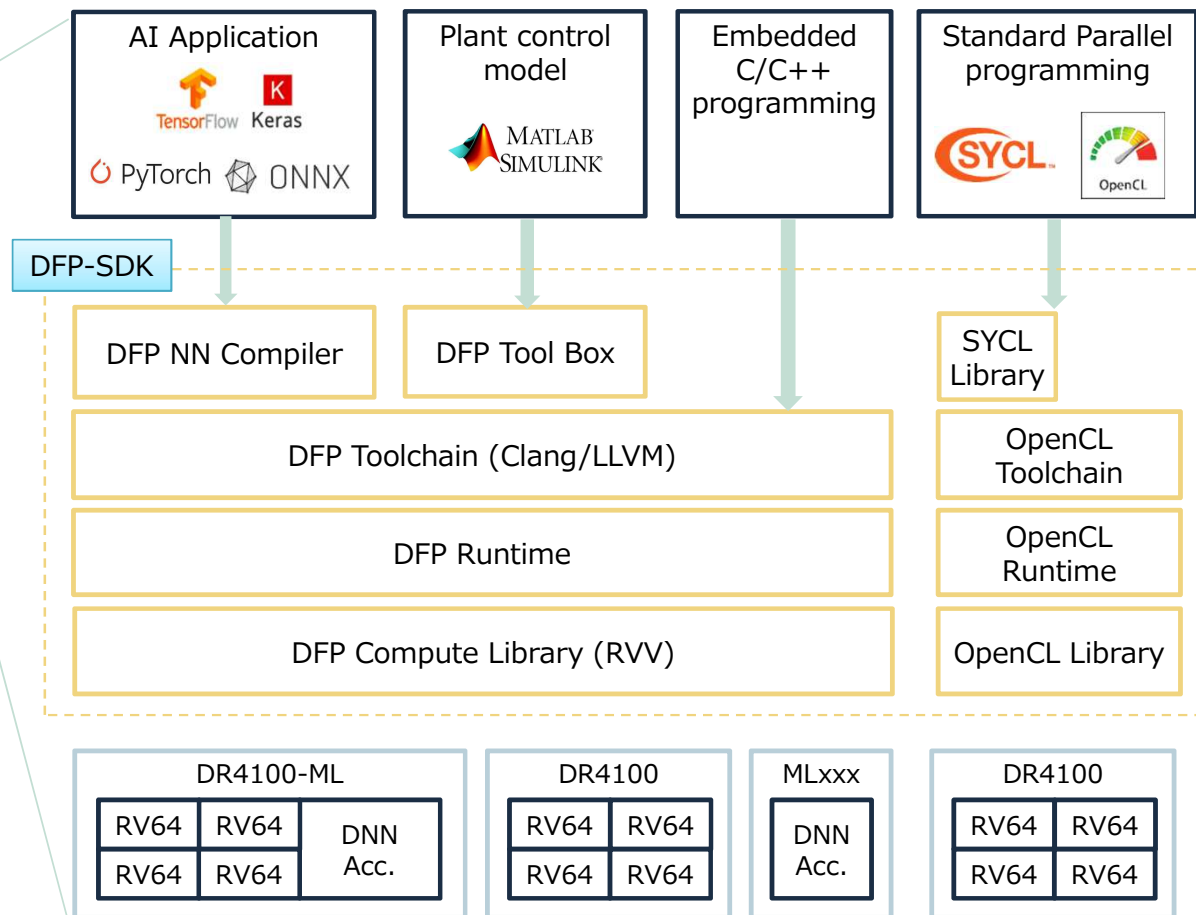


# RISC-V, AI の応用例 – S/W 構成

## 統一 SPF へ組み込み



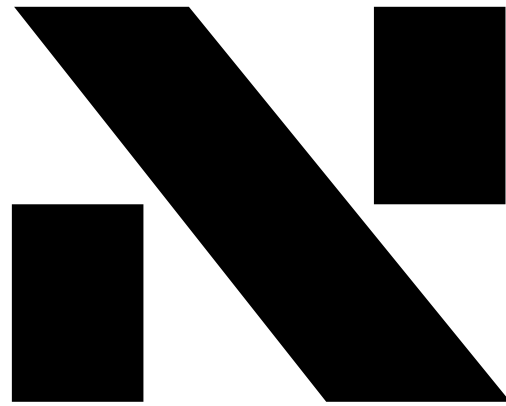
- Highly functional and easy to implement industry-standard AI and plant control models in DFP
- Seamlessly enable PC-based application development with DFP
- Compiler for efficient NN layer fusion
- High performance and reliable real-time thread control
- Automotive quality and functional safety



# アジェンダ

1. 会社紹介
2. NSITEXE次世代DFP向けSDKの構想
3. アクセラレータ開発環境エコシステムへの期待
4. AIアクセラレータへの応用例
- 5. まとめ**

- 次世代DFP向けにOpenCLを中心に据えるSDKを  
構想 ⇒商用化に向けて開発進行中
  - RISC-V Vector extensionに対応するプロセッサの性能を最大  
限引き出すための礎 ⇒やるべき事が数多く残されている！！
  - 高位インターフェースとしてSYCLとOpenMPオフローディングを用意
- OpenCL/SYCLを中心とするソフトウェアエコシステムは  
汎用プロセッサにおいても非常に有望
  - ベンダニュートラルな**標準規格**と高い**可読性**
  - 単純にC+マルチスレッドで記述するよりも**高い性能**が得られる  
可能性



NSI-TEXE