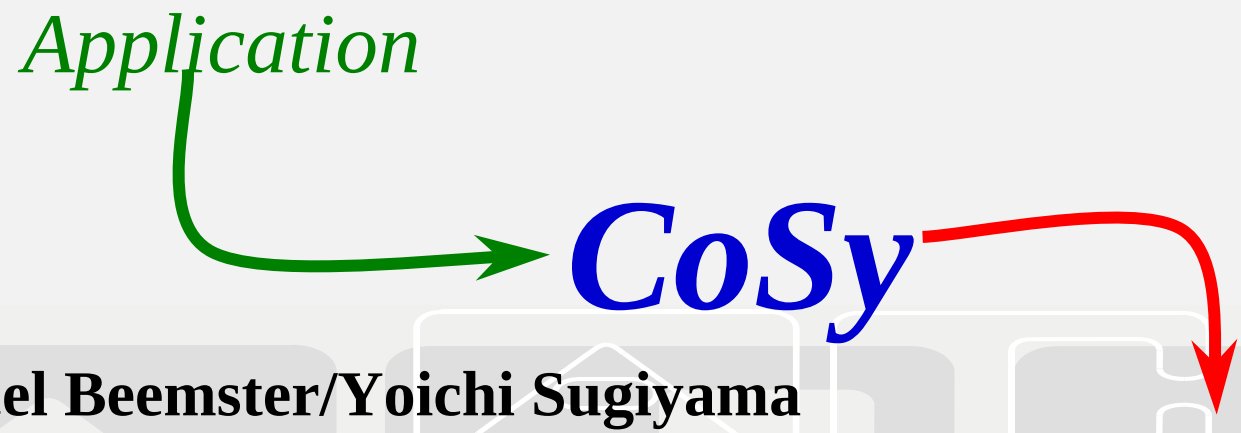
CoSyコンパイラ開発システムによる組み込み用プロセッサの並列実行のサポート



Marcel Beemster/Yoichi Sugiyama
ACE Associated Compiler Experts &
Japan Novel Corporation
contact: yo_sugi@jnovel.co.jp

JNOVEL 組み込みシステムにおける並列実行の必要性

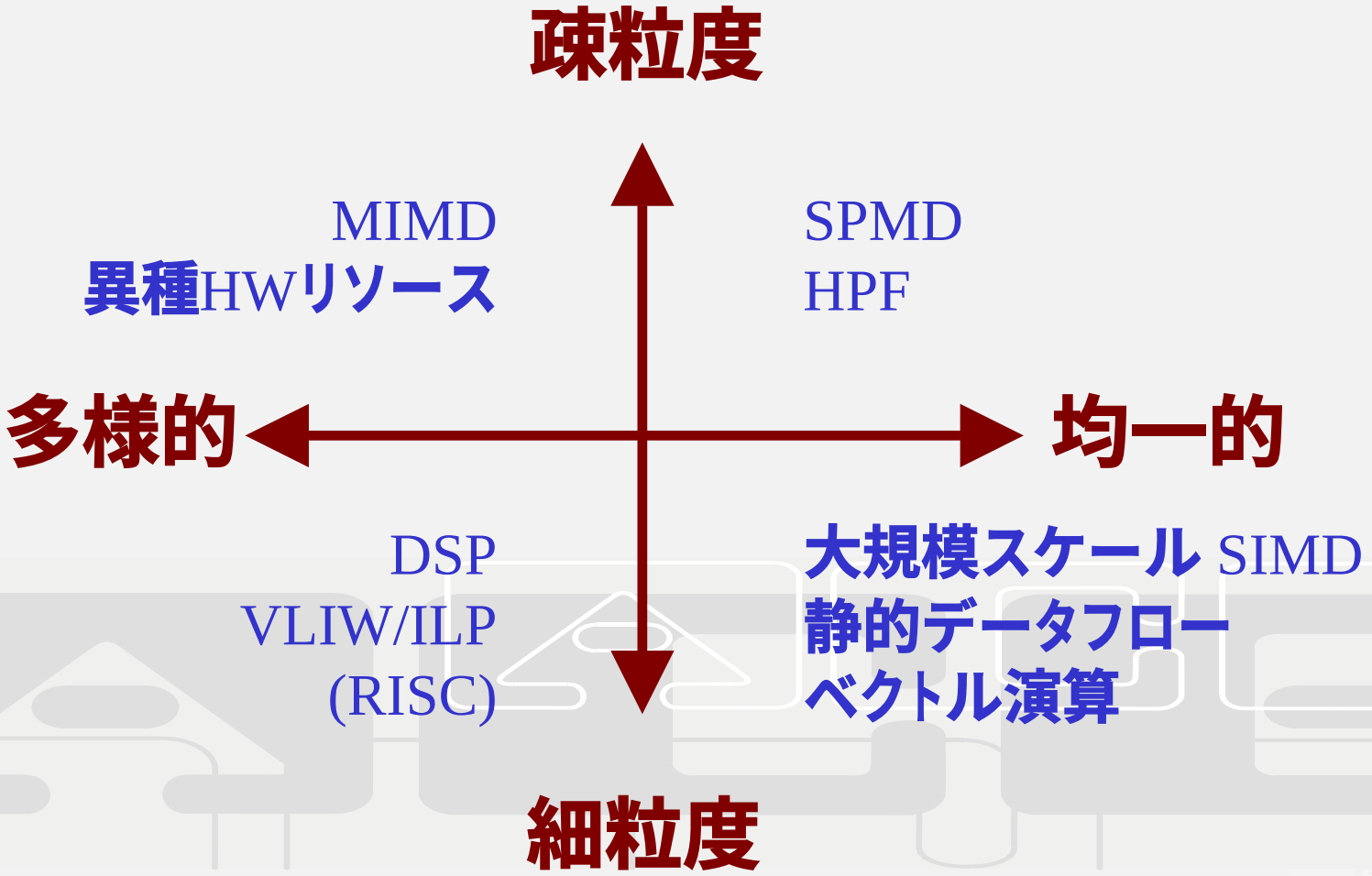
- 増大する演算能力に対する要求
- 消費電力 (電池持続時間、冷却) 問題： 高速になるほど消費電力は増大
- 既存の演算モジュールの使用
- 並列実行に適したアプリケーションの存在

- しかしながら、並列実行は容易ではない・・・

並列実行の多様性

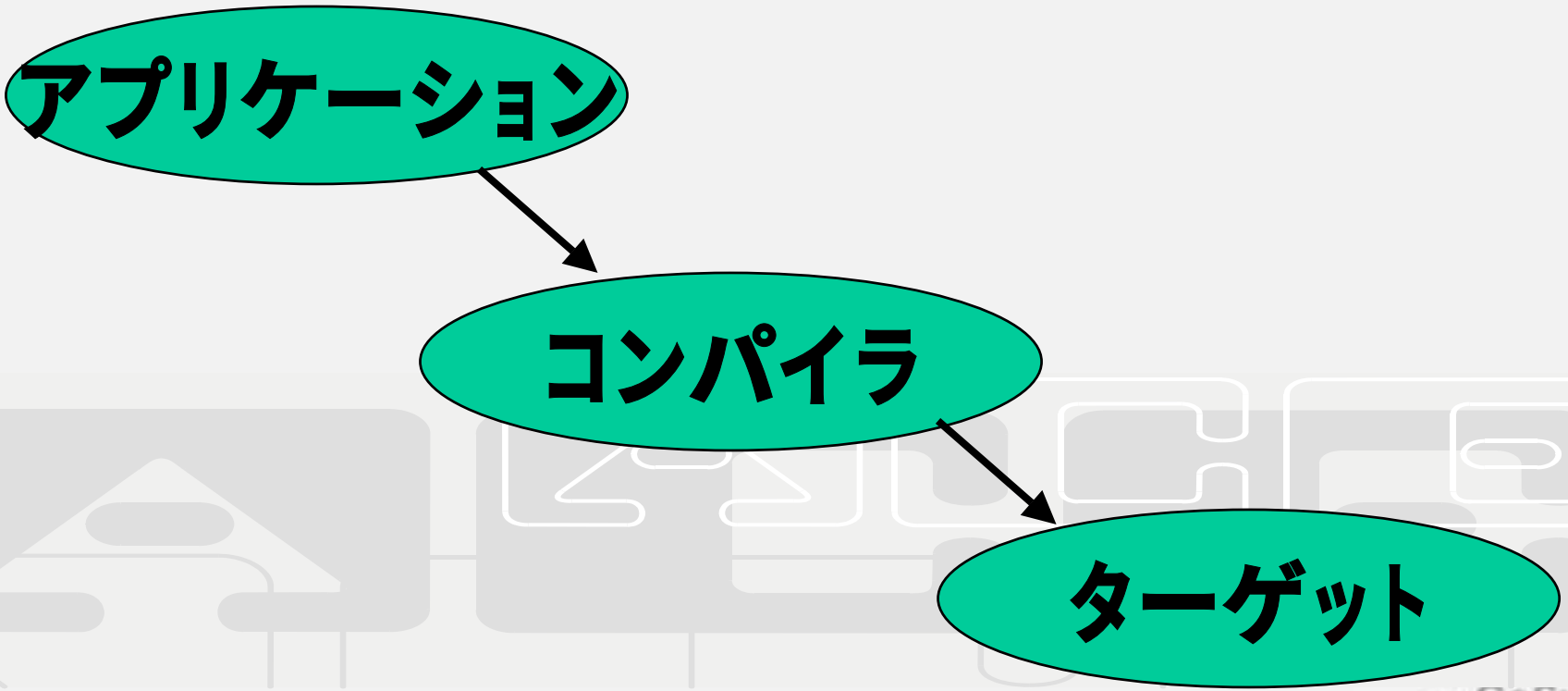
- 密結合と疎結合
  - パイプライン、VLIW、SIMD、ベクトル、静的データフロー、MIMD、等
- 自動的または明示的並列実行
- 並列実行のタイプによりサポートするツールに対する要求内容が変わる、特にコンパイラが顕著

アーキテクチャの特性

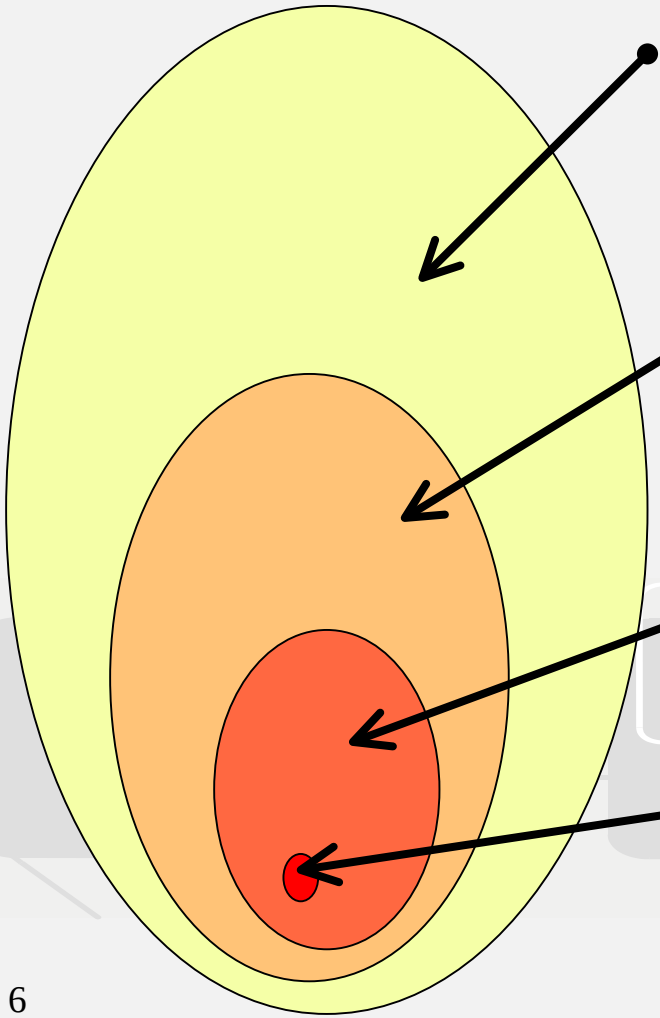

疎粒度
MIMD
異種HWリソース
SPMD
HPF
多様的
均一的
DSP
VLIW/ILP
(RISC)
大規模スケール SIMD
静的データフロー
ベクトル演算
細粒度

コンパイラの役割

- コンパイラがアプリケーションコードの並列実行要素をターゲットのアーキテクチャにマッピング



自動的並列化が常に可能とは限らない



アプリケーション

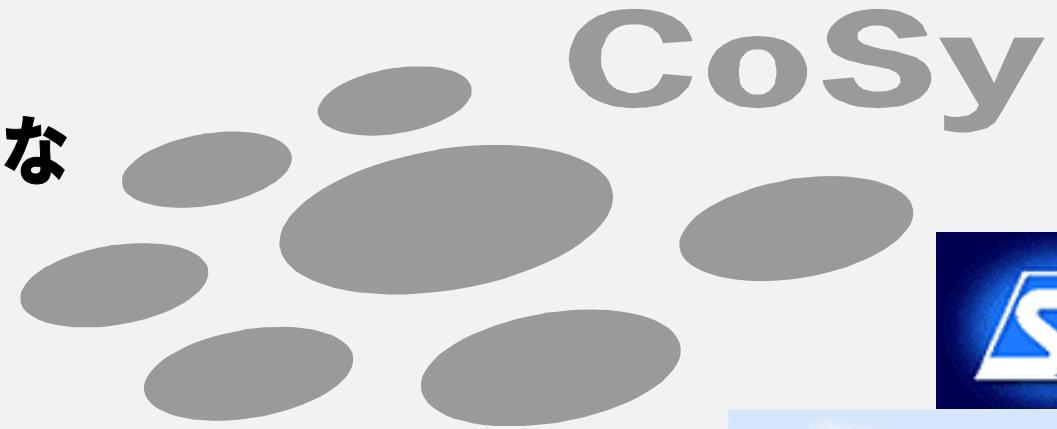
人手で並列化可能なコード

明示的アノテーションにより
並列化可能なアプリケーション

自動的並列化が可能な順次処理
アプリケーション

CoSyについて：

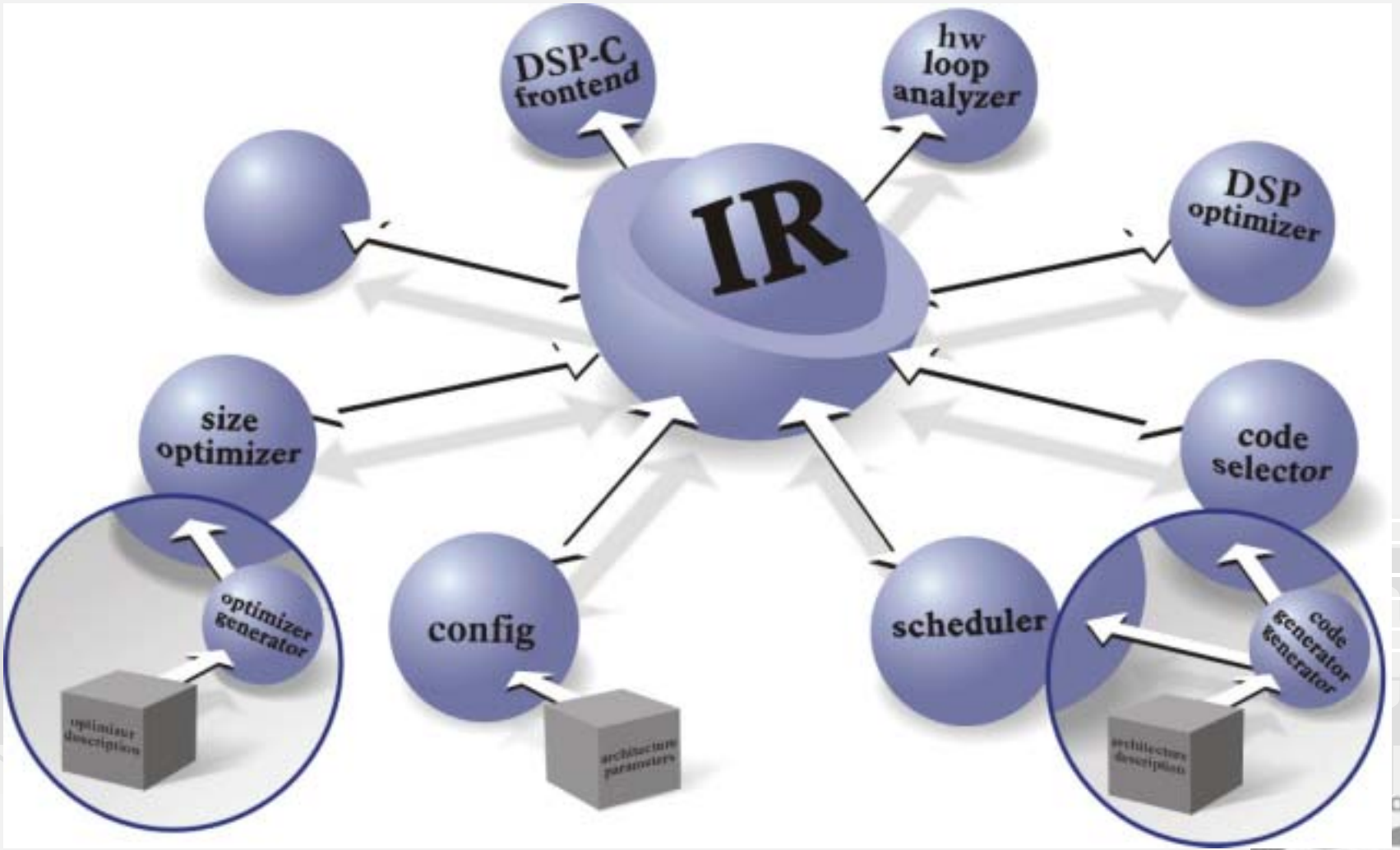
- 世界で一番先進的なコンパイラ開発システム
- 全世界の主要な企業で使用



CoSyの品質

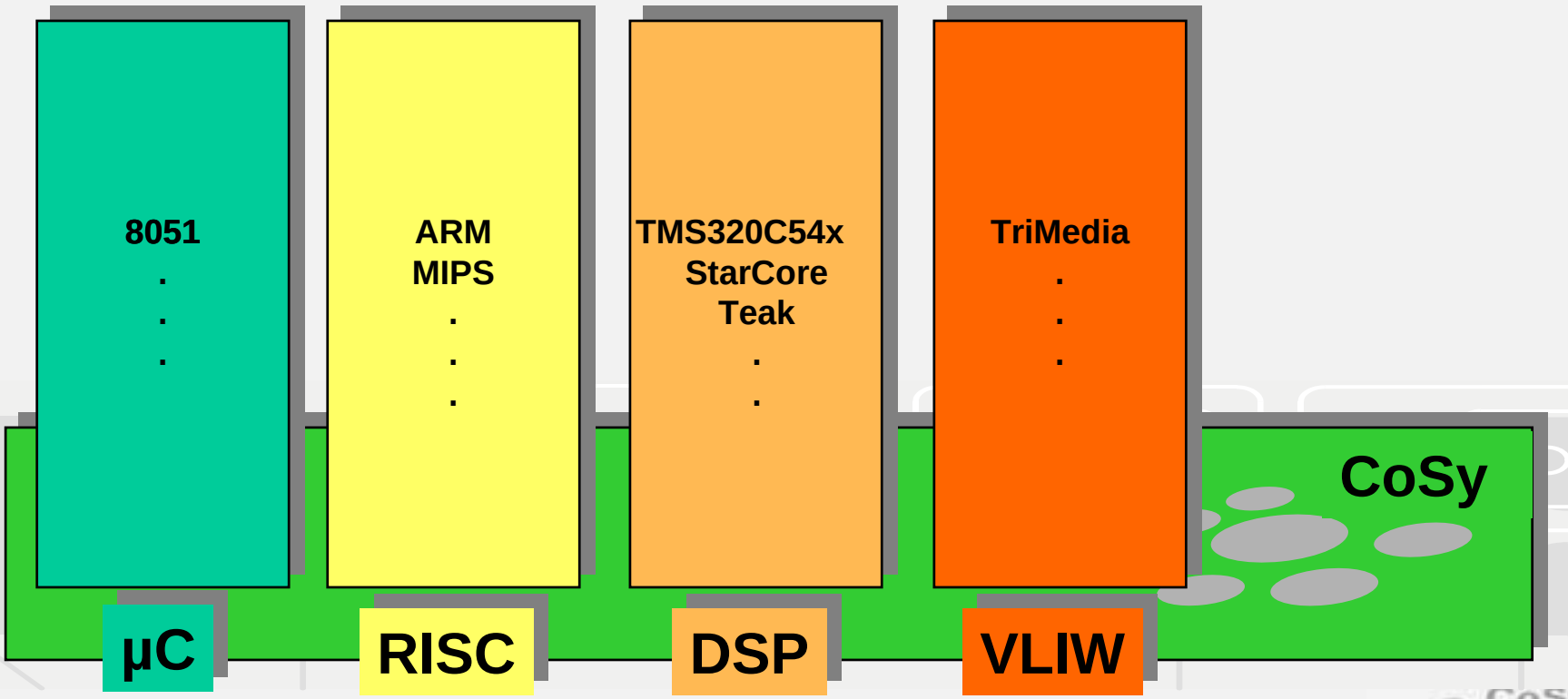
- コンパイラ生成システム
- モジュール構造
- 再構築可能
- リターゲット可能
- 豊富な機能
- 拡張可能
- 高品質
- 最適化されたシステム
- ACE社より製品とサポートを提供
- 日本では日本ノーベルよりサポート

CoSy の構造


DSP-C frontend
hw loop analyzer
IR
DSP optimizer
size optimizer
code selector
optimizer generator
config
scheduler
code generator generator

CoSy 2003

- CoSy はフレキシブルなコンパイラ開発環境として多様なアーキテクチャに対応



CoSy技術：パイプライン RISC アーキテクチャ

- メモリ読み込み遅延挿入 (スケジューラ)
- 分岐遅延スロット挿入 (スケジューラ)
- レジスタ割付 (レジスタアロケータ)

CoSy技術：DSPアーキテクチャ

- 複数メモリ読み込みサポート (スケジューラ)
- ポストインクリメントアドレシング (スケジューラ)
- 特別レジスタの効率的使用 (レジスタアロケータ)
- ゼロオーバーヘッドループのサポート

CoSy技術：
VLIW アーキテクチャ

- リソースとレイテンシを考慮した命令の集約化（スケジューラ）
- 述語付き命令実行
- インライン化
- ループ展開
- ソフトウェアパイプライニング

ソフトウェアパイプライニング例

```
void
func(float * restrict p, float * q, float * r)
{
   int i;

   for (i = 0; i < 10; i++) {
      *p++ = *q++ * *r++;
   }
}
```

```
func:
        save    %sp,-104,%sp
        add     %g0,10,%l5
.L1:
! --- cycle 0 --------------
        ld      [%i2+0],%f0
        ld      [%i1+0],%f1
        add     %i2,4,%i2
! --- cycle 1 --------------
        add     %i1,4,%i1
! --- cycle 3 --------------
        fmuls   %f0,%f1,%f0
! --- cycle 7 --------------
        st      %f0,[%i0+0]
        add     %i0,4,%i0
        subcc   %l5,1,%l5
        bne     .L1
        nop
! --- cycle 0 --------------
        ret
```

SPARC ループ

← **処理前: 8 サイクル**

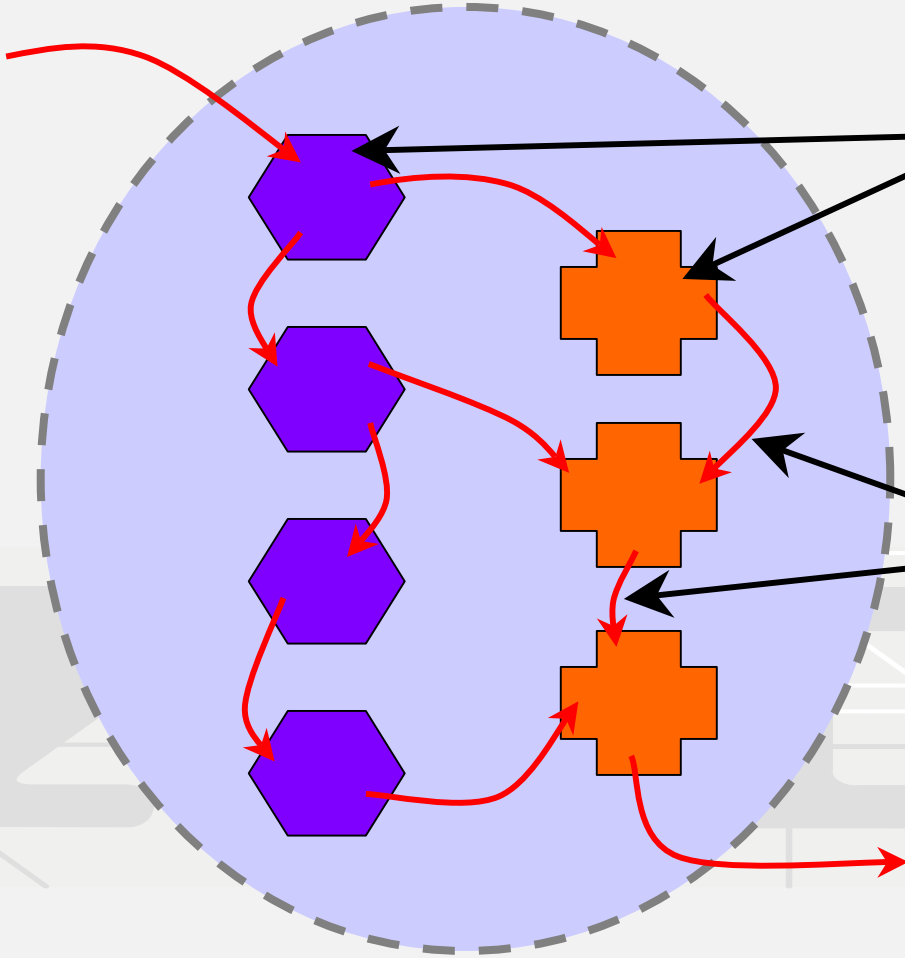
→ **処理後: 4 サイクル**

```
func:
        save    %sp,-104,%sp
        ld      [%i1+0],%f1
        ld      [%i2+0],%f0
! --- cycle 1 --------------
        add     %i1,4,%i1
        add     %i2,4,%i2
! --- cycle 3 --------------
        fmuls   %f0,%f1,%f2
        add     %g0,9,%l5
.L1:
! --- cycle 0 --------------
        ld      [%i1+0],%f1
        ld      [%i2+0],%f0
! --- cycle 2 --------------
        add     %i1,4,%i1
        add     %i2,4,%i2
! --- cycle 3 --------------
        st      %f2,[%i0+0]
        fmuls   %f0,%f1,%f2
        add     %i0,4,%i0
        subcc   %l5,1,%l5
        bne     .L1
        nop
! --- cycle 0 --------------
        st      %f2,[%i0+0]
        add     %i0,4,%i0
        ret
```

CoSy技術：SIMD及びベクトル演算プロセッサ

- データ依存性の解析
- ベクトル演算自動処理 (開発中)
- アラインメント解析 (SIMD向け)
- ダイナミックイントリンシック (コンパイラ既知ファンクション) のサポートとスケジューリング (FPGAも可能)
- コンパイラ既知データ (例：XYZw, RGBa 構造)

CoSy技術：
再構成可能な静的データフローマシン



SDFにおけるストリーム構造の抽出

- ネステッドループプログラムに対応
- メモリの入出力コマンドを抽出
- データフローにおける依存性の抽出
- 同期的ストリームプログラムの生成
  - 再構成可能アーキテクチャにマッピング可能
  - FPGAへのマッピングが可能
  - ハードウェアへの直接マッピングが可能
  - ベクトル演算/SIMDアーキテクチャへのマッピングが可能

メモリーI/Oの解析

マトリクス演算より:

```
for (i=0;i<N;i++){
   for (j=0;j<N;j++){
      for (k=0;k<N;k++){
         .. = .. a2[k][j] ..;
```

アドレス変換適用:

```
StreamInStream3( (int*)a2, N, 0,
                 N, 1,
                 N, N );
```

例題：配列ベース DCT

```
for (block = 0; block < NBLOCKS; block++) {
    for (y = 0; y < SIZE; y++) {
        for (x = 0; x < SIZE; x++) {
            Result[block][y][x] = 0;
            for (v = 0; v < SIZE; v++) {
                for (u = 0; u < SIZE; u++) {
                    int32 tmp;
                    int32 t1 = cosines[x][u];
                    int32 t2 = cosines[y][v];
                    tmp = MUL(t1, t2);
                    tmp = UNSCALE(tmp);
                    tmp = MUL(tmp, inData[block][v][u]);
                    Result[block][y][x] += tmp;
            }   }
            Result[block][y][x] = (Result[block][y][x] >> 2) + SCALE(128);
            Result[block][y][x] = UNSCALE(Result[block][y][x]);
            Result[block][y][x] = (Result[block][y][x]);
            if (Result[block][y][x] > 255)
                Result[block][y][x] = 255;
            else if (Result[block][y][x] < 0)
                Result[block][y][x] = 0;
}   }   }
```

CoSyで生成したDCT用ストリームコード

```
in0 = new in_stream(inData, stride(4,256), stride(8,0),
              stride(8,0), stride(8,32), stride(8,4));
in1 = new in_stream(cosines, stride(4,0), stride(8,0),
                stride(8,32), stride(8,0), stride(8,4));
in2 = new in_stream(cosines, stride(4,0), stride(8,32),
                  stride(8,0), stride(8,4));
calc0 = StreamMultiply(in1, in2)
calc1 = StreamAddition(calc0, 8192)
calc2 = StreamShiftright(calc1, 14)
calc3 = StreamMultiply(in0, calc2)
calc4 = StreamAccumulate(calc3, ?)
calc5 = StreamShiftright(calc4, 2)
calc6 = StreamAddition(calc5, 2097152)
calc7 = StreamAddition(calc6, 8192)
calc8 = StreamShiftright(calc7, 14)
calc9 = StreamSatCeiling(calc8, 255)
calc10 = StreamSatFloor(calc9, 0)
StreamOutStream(calc10, Result, stride(4,256), stride(8,32),
                                                 stride(8,4));
```

CoSy技術：SPMDアーキテクチャ

- 高性能 Fortranコンパイラのフロントエンドから IRへ
- 集合的配列操作のための IRの拡張
- データ部分分割の実施
- コミュニケーションスタブの生成
- プログラム相互の同期化

CoSy技術：MIMD及び異種マルチプロセッサ

- アプリケーションは明示的に分割済みであること
- プラグマ制御によるターゲットの指定
- データモデルの統合化
- 実現されていない機能のエミュレーション（例：RISCにおける固定小数点演算のサポート）
- コミュニケーションスタブの生成
- OS機能の包含(コンパイラ既知ファンクション)

HW/SW協調設計のための CoSy Express

- CoSyベースの OEM製品
- 既構成済みCoSy：コンパイラ生成にはデータモデル及びコード生成用ルールが必要
- 最適化モジュール、ライブラリ、テスト用フレームワークを含む

⇒ コンパイラの生成が非常に高速 (数分で)

⇒ HW/SW協調設計環境におけるコンパイラ開発ツールとして

CoSyに関するその他の特長…

- C-89, C-99, DSP-C, Embedded C, C++, Fortran, GNU拡張に対するフロントエンドの提供
- Dwarf2デバッグ情報の生成
- 強力なループの最適化機構 (ゼロオーバーヘッドループのサポートを含む)
- ビットレベルのターゲット構築が可能
- エミュレータの生成
- 早期開発着手のためのサンプルコンパイラの提供
- 関数呼び出し系列、スタックのレイアウトの設定が可能

ACE社について

- 所在地はオランダのアムステルダム
- 創業30年、社員30名規模
- CoSyコンパイラ開発システムにフォーカス
- コンパイラの自社開発を行う顧客に対して世界的に展開
- CoSy製品及びサポートの提供
- 日本では日本ノーベル社より販売・サポート

Japan Novel and CoSy

- 日本ノーベルはACE社の日本での販売代理店
- 日本ノーベルはソフトウェア製品の品質向上を実現する製品を提供
  - コンパイラ評価サービスの提供
  - 組込み機器の自動テストシステム「QualityCommander」の販売
  - PlumHall社製品販売代理店
- コンパイラ評価サービスはC／C++、Embedded－C、DSP-Cテストを提供
- CoSyコンパイラ開発システムはその高い信頼性から日本の組込みシステム開発に大きく貢献できるものと考える

CoSyで並列実行を実現しよう!

ACE Associated Compiler Experts

Home of CoSy

the Compiler Development System

yo_sugi@jnovel.co.jp/marcel@ace.nl

標準製品の内容 ‘out-of-the-box’

- CoSyコンパイラ開発システム
  - 多数の最適化エンジンを用意
  - バックエンドコード生成モジュール
- サンプルコンパイラ及び使用法
- SuperTest C/C++ テスト･検証スイート
- 標準 C ライブラリ
- CADESE バージョン管理システム
- CoSy サポートサービスプログラム