

**2016年度(平成28年度)
広島市大・卒業研究**

実現するアルゴリズムの証明に 注目したASIPのシステム検証

**広島市立大学・情報科学部・情報工学科
錦織光輝 (高橋隆一指導)**

Mitsuki Nishikori

研究背景

1990年代にはVerilog HDLによって仕様を記述し、
論理合成によって回路を実現するスタイルが普及した

検証技術が論理合成に続く技術として期待されている

満たすべき性質をアサーションとして記述することによる
シミュレーションでの検証(ABV)が普及しつつある

特定用途向け専用プロセッサ(ASIP)が注目されている

アサーションの記述方法は必ずしも確立されていない

本研究では、実現するアルゴリズムに注目した
ASIPのABVを模索する

高橋研究室での従来研究

アサーションの構成方法

事例

アルゴリズムで普遍的に成り立っている性質に注目

x と y の拡張ユークリッドの互除法において余りは常に $ax+by$ の形であることを検証

処理対象の性質に注目

バブルソートにおいて連続する3つの要素に対する操作を検証

データ構造(スタック)に注目

クイックソートにおいてスタックポインタが下がる前にはPOP命令が読まれていることなどを検証

データ構造(ヒープ)に注目

ヒープソートにおいて親のノードの値が子のノードの値よりも小さいことなど検証

データ構造(キュー)に注目

幅優先探索におけるENQUEUE、DEQUEUEなどを検証

背景のマトロイドに注目

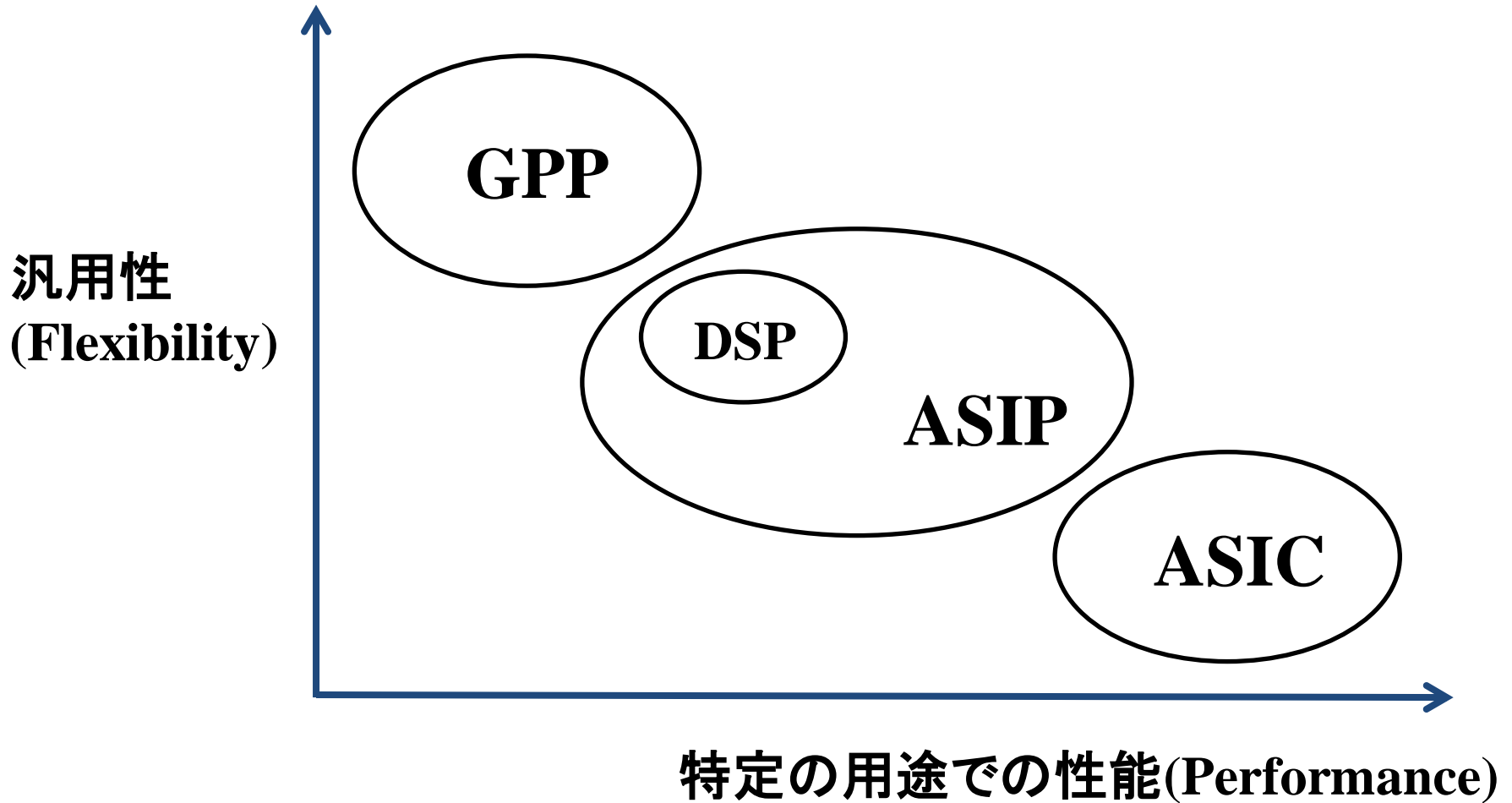
クラスカルのアルゴリズムにおいて連結成分に閉路がないことを検証

特定応用分野向きプロセッサ

application specific instruction-set processor(ASIP)

- ・特定の応用分野に適したアーキテクチャを持つプロセッサ
- ・ハードウェア積和演算器 (Multiplier accumulator (MAC))
を内蔵したデジタル信号処理プロセッサ
(digital signal processor(DSP))はASIPの代表格
- ・汎用プロセッサ (general purpose processor(GPP))
と特定用途向き集積回路
(application specific integrated circuit (ASIC))
の間を埋める手段

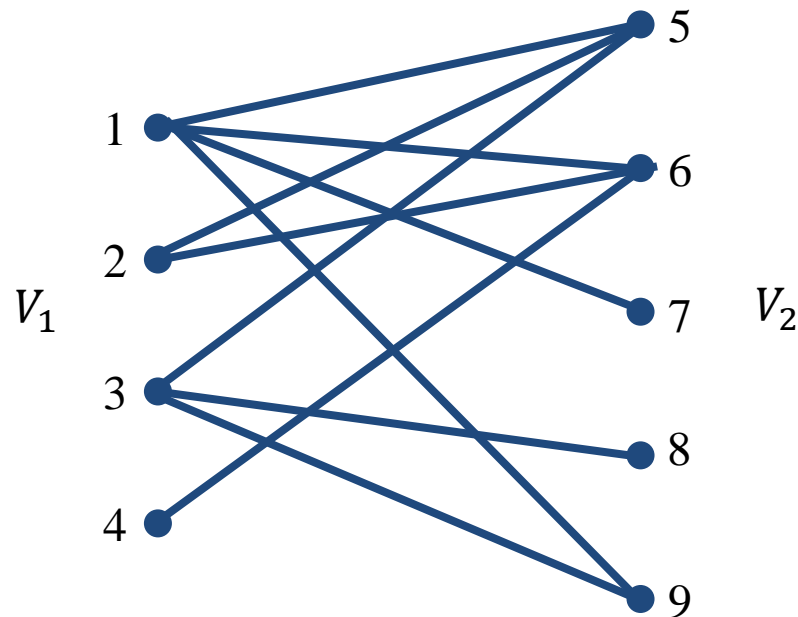
ASIPの位置付け



デジタル信号処理プロセッサ(DSP)はASIPの代表格

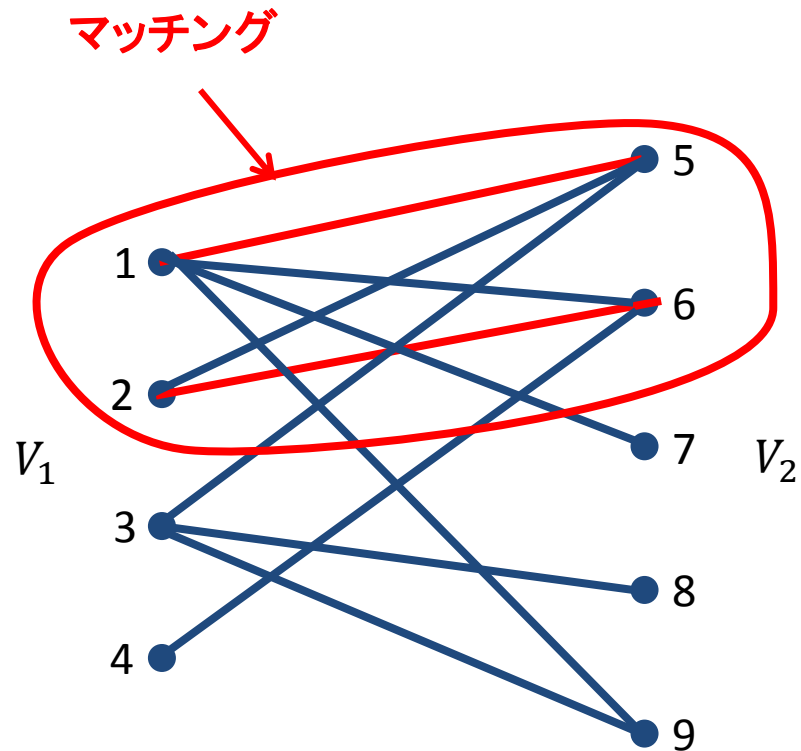
二部グラフ

頂点集合を V_1, V_2 に分割したとき,すべての枝が V_1 の頂点と V_2 の頂点を結ぶようにできるグラフを二部グラフという



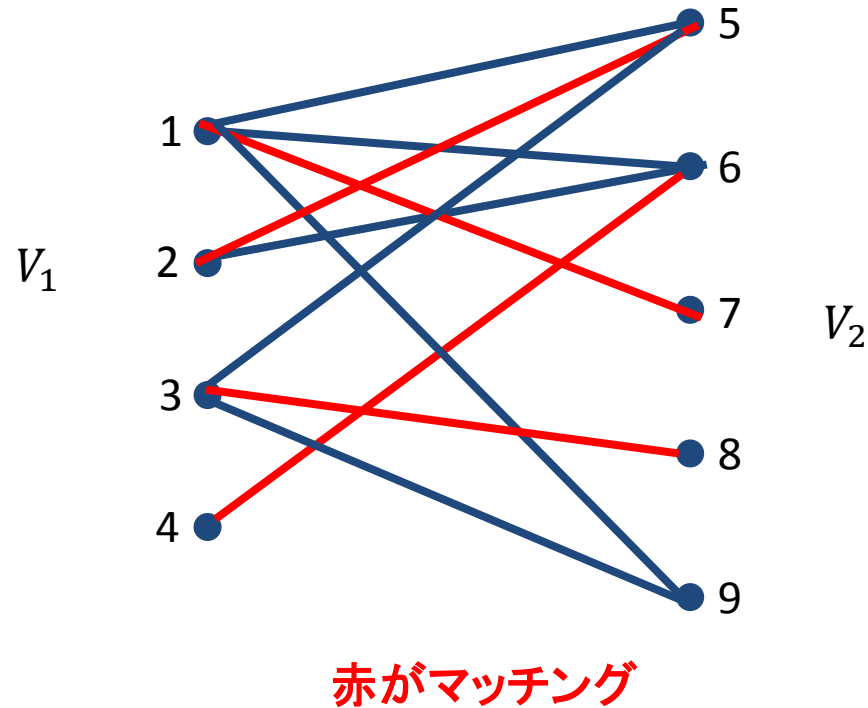
二部グラフにおけるマッチング

枝集合で、互いに端点を共有しない組



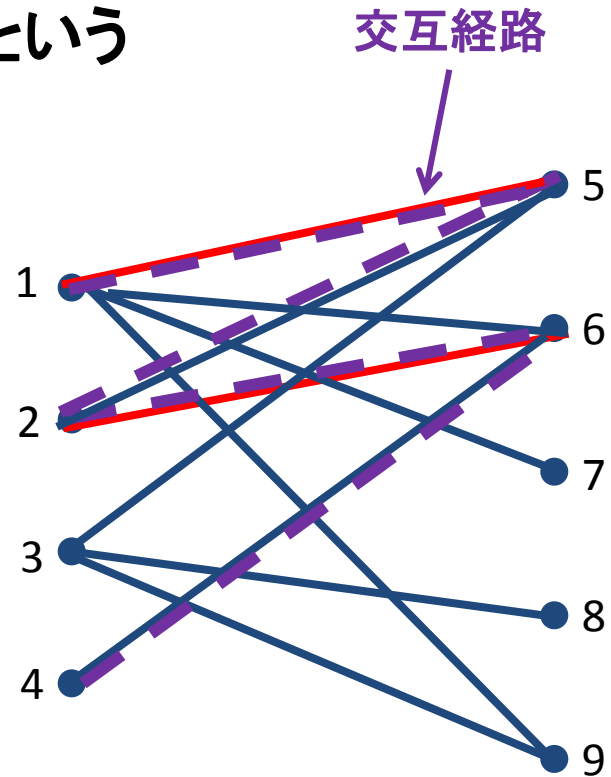
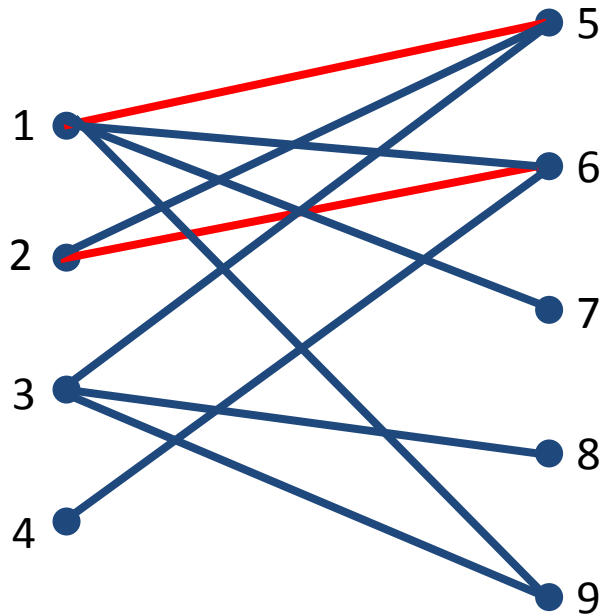
二部グラフにおける最大マッチング

枝数が最大のマッチング(枝の集合)



二部グラフにおける交互経路の例

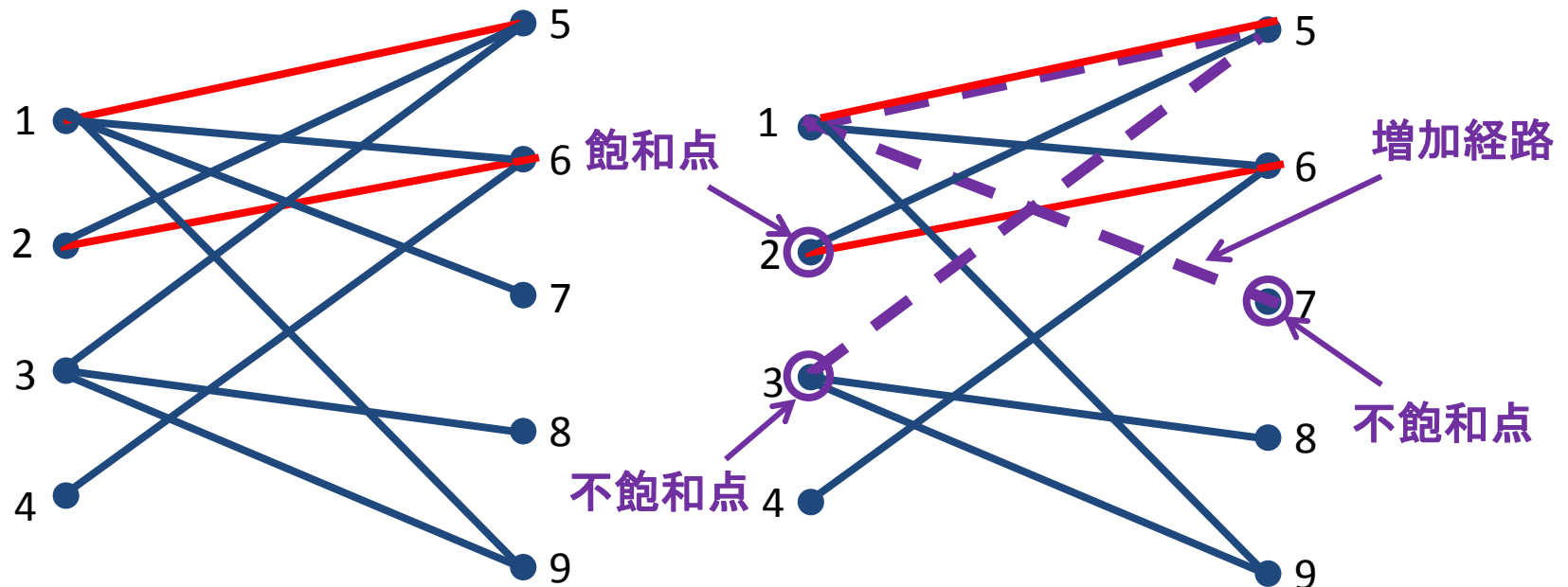
マッチングの枝とマッチングでない枝が交互に現れる経路を交互経路という



二部グラフにおける増加経路の例

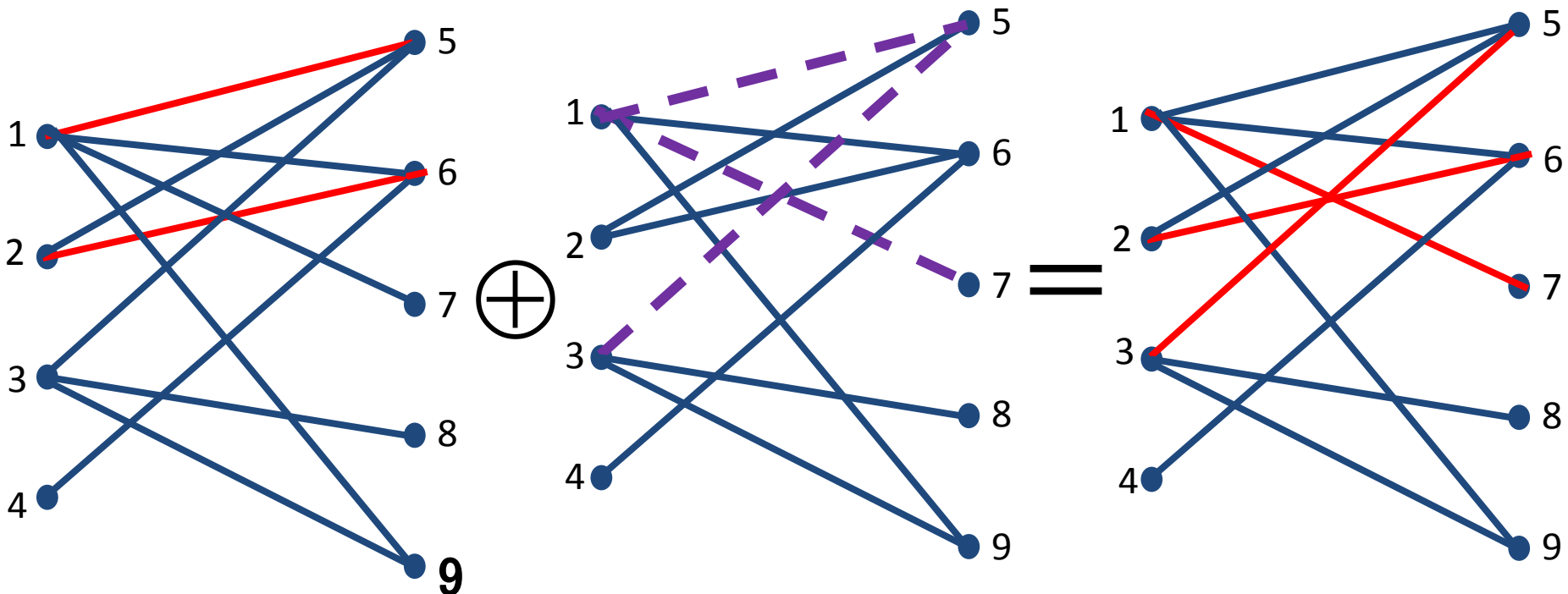
マッチングの枝の端点を飽和点,
マッチングでない枝の端点を不飽和点という

両端点が不飽和点である交互経路を増加経路という



現在のマッチングと増加経路の対称差

現在のマッチングMと増加経路の対称差をとることで
枝の増えた新たなマッチングM'が得られる



現在のマッチングM

増加経路

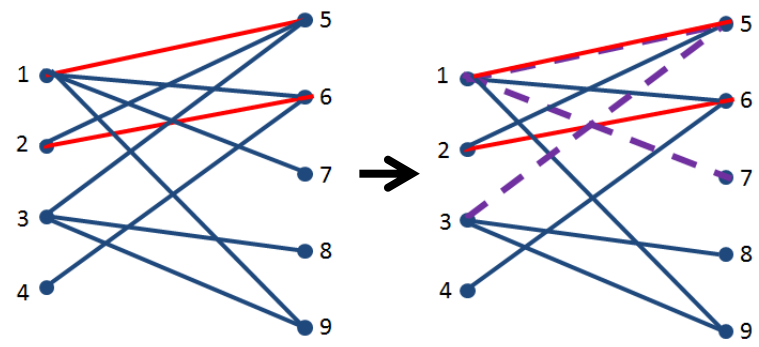
新たなマッチングM'

$$M \oplus \text{AUGMENT_PATH} = M'$$

二部グラフの最大マッチングを求める アルゴリズム

開始

現在のマッチングにおける
増加経路を探す



現在のマッチング

増加経路

増加経路が
存在するか

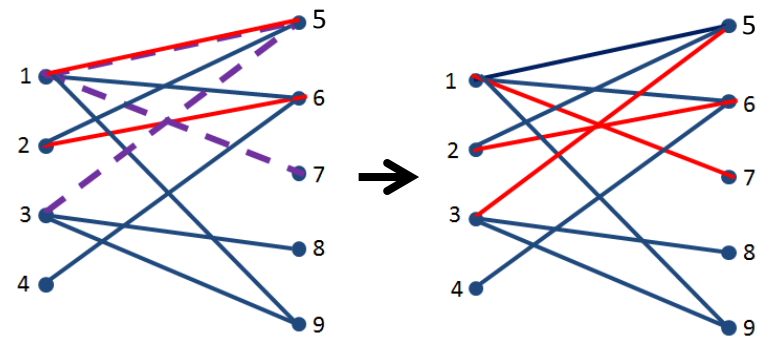
false

終了

新たなマッチングを
現在のマッチングとする

現在のマッチングと
増加経路との対称差をとる

true

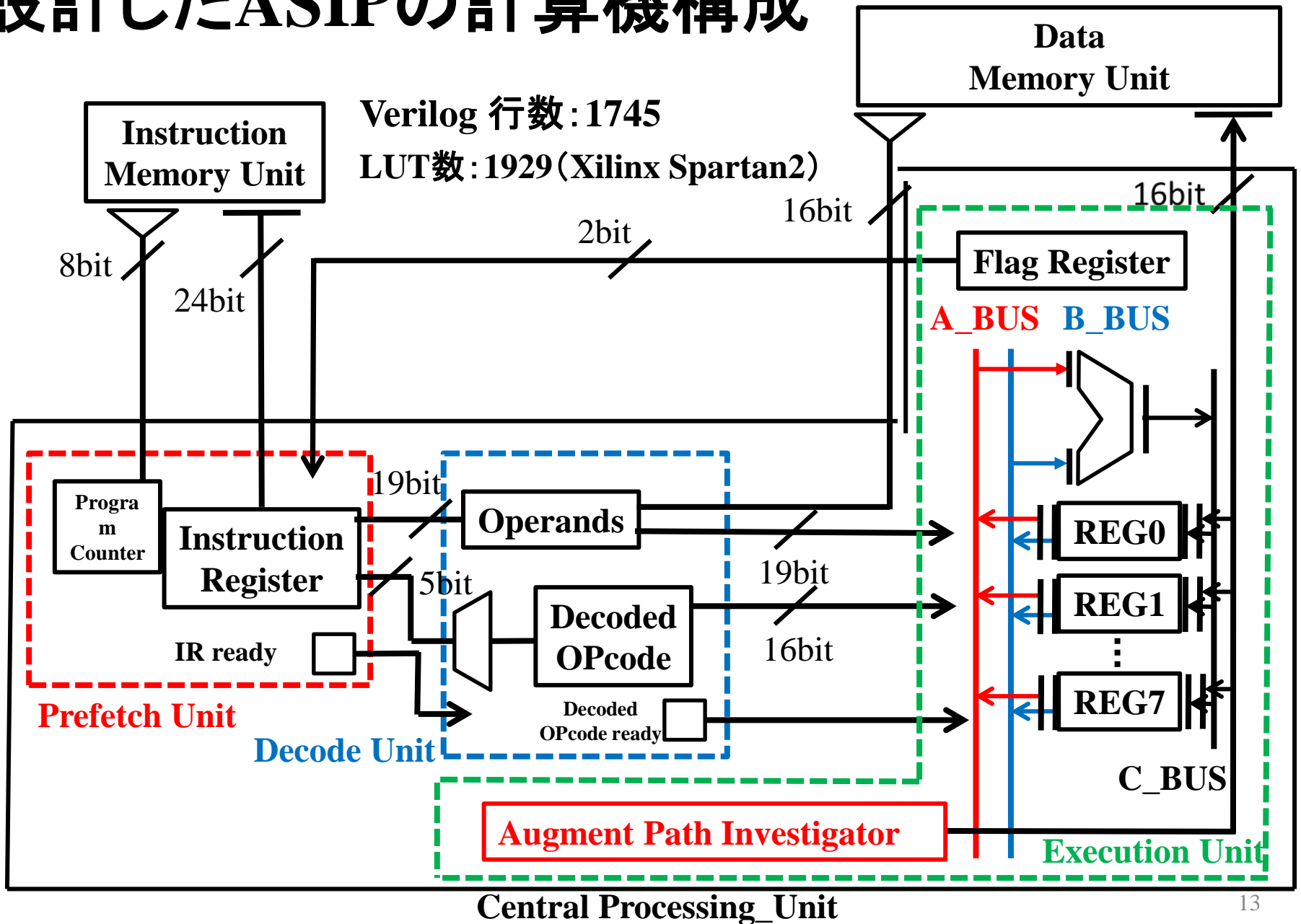


対称差をとる

新たなマッチング

$$M \oplus \text{AUGMENT_PATH} = M'$$

設計したASIPの計算機構成



検証の前提

三段パイプラインの正当性は検証済み

パイプラインインターロックは正しく設計されている
フォワーディングも正しく行われている

フォワーディングに対するアサーションが
満たされている

作りこみ品質は高い

問題点があるとなれば **増加経路の取り扱い**

Augment Path Investigator

アルゴリズムの証明

アルゴリズムの根拠

M: 現在のマッチング

増加経路が存在しない \Leftrightarrow Mは最大マッチングである

必要性 Mは最大マッチングである \Rightarrow 増加経路が存在しない

対偶 増加経路が存在する \Rightarrow Mは最大マッチングでない

自明

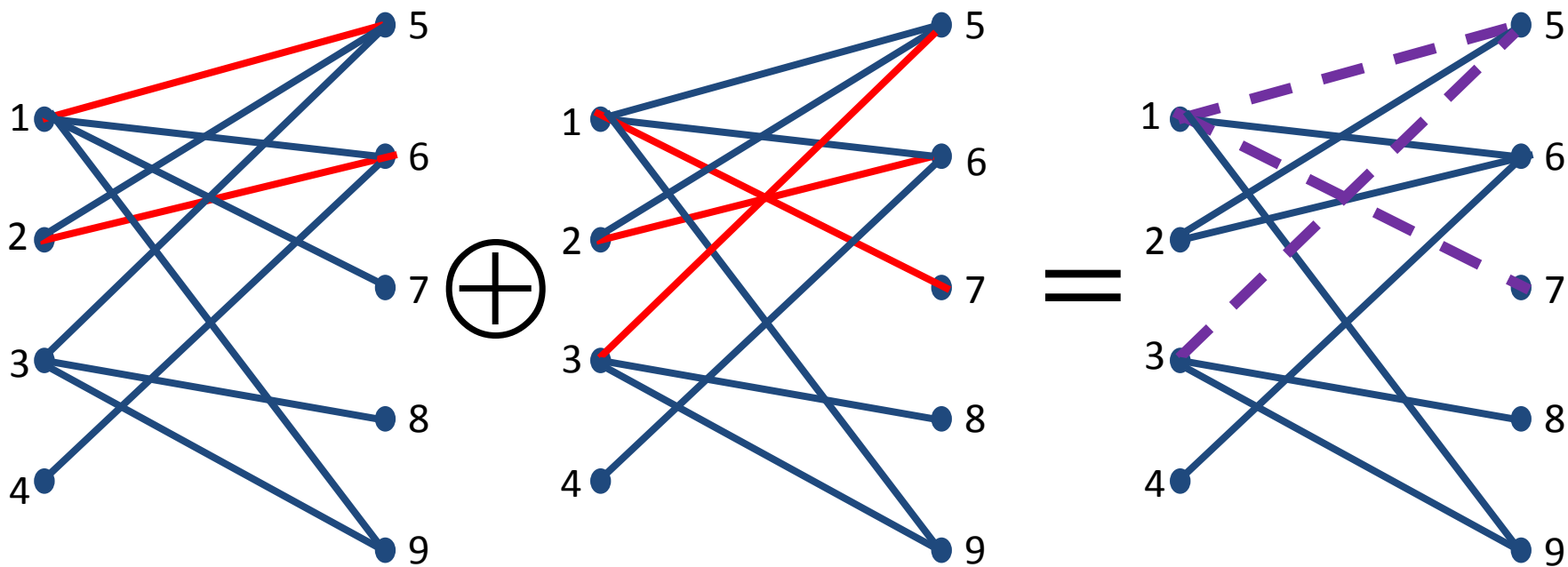
十分性 増加経路が存在しない \Rightarrow Mは最大マッチングである

対偶 Mが最大マッチングでない \Rightarrow 増加経路が存在する

最大でないマッチングと枝数の多いマッチングとの対称差をとることで存在が証明ができる

現在のマッチングと新たなマッチングの対称差

最大ではないマッチングと枝数の多いマッチングの対称差をとると増加経路が現れる



最大ではないマッチング 枝数の多いマッチング

増加経路

証明に注目した検証

現在のマッチングと新たなマッチングとの対称差が
増加経路になることを検証

$M \oplus M' = \text{AUGMENT_PATH}$

M: 現在のマッチング

M': 新たなマッチング

AUGMENT_PATH: 増加経路

```
core_dmu_MMX: assert property
  ( @(posedge CLOCK1)
    (LAST==0)##1(LAST==1)|->CHECKPATH);
```

証明に注目したアサーション
(満たされるべき性質の記述)

検証時のアクセス

assertion_for_DM.sv

```
core_dmu_MMX: assert property  
  ( @(posedge CLOCK1)  
    (LAST==0)##1(LAST==1)|->CHECKPASS);
```

証明に注目したアサーション

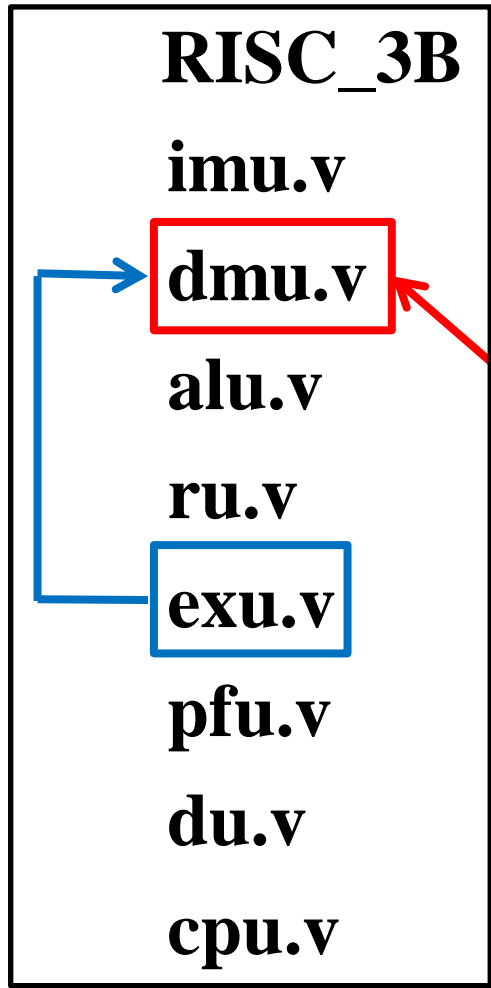
マイクロアーキテクチャはポートを通して
メモリにアクセスする (Verilog HDL)

アサーションはメモリ内部を直接検証する
(System Verilogでバインド)

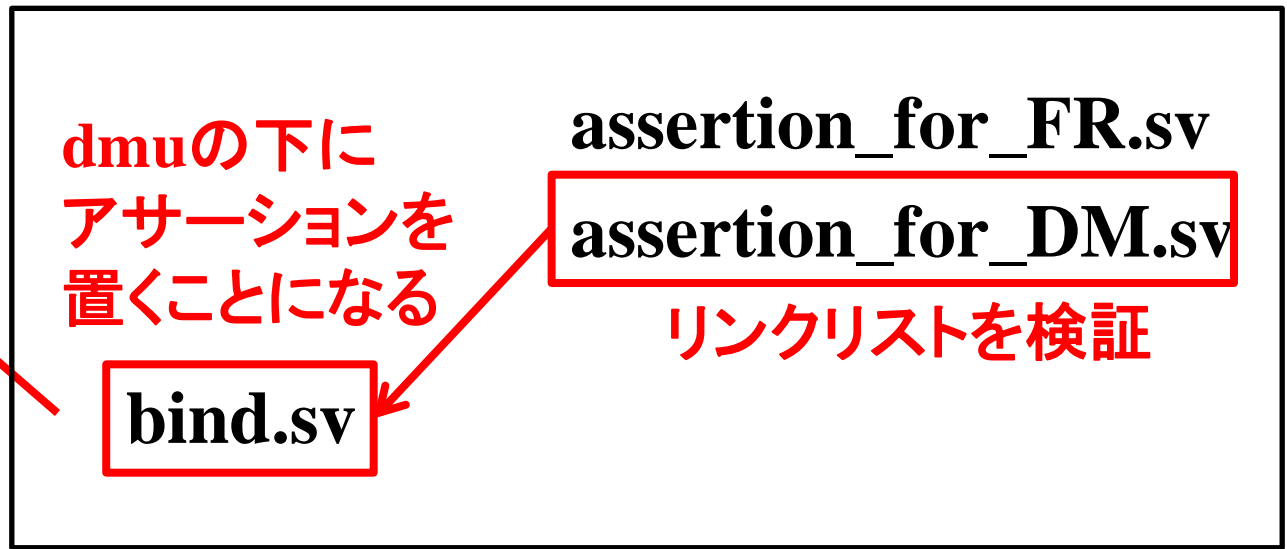
バインド(bind)

アサーションをモジュール内に置く手段

Verilog HDL **dmu:data memory unit**



System Verilog 特有



アサーションはメモリ内部のリンクリストを直接調べる

実行ユニット
execution unit
(exu)

実行ユニットはバスを介してメモリにアクセスする

ABVの結果

$$M \oplus M' = \text{AUGMENT_PATH}$$

Name ▾	Cursor ▾	0ns	3450ns	3460ns	3470ns	3480ns	3490ns	3500ns
core_dmu_MMX	active	active			failed		active	

アサーションは通らなかった
増加経路の取り扱いに問題がある

最大でないマッチングと増加経路の
対称差をとるハードウェアの問題

M: 現在のマッチング M': 新たなマッチング
AUGMENT_PATH: 増加経路

$$M \oplus \text{AUGMENT_PATH} = M'$$

増加経路を求めるハードウェアの問題

バグの特定①

```
core_dmu_NUM: assert property  
  ( @(posedge CLOCK1)  
    (LAST==0)##1(LAST==1)|->CHECKNUM);
```

新たなマッチングの枝数が
現在のマッチングの枝数より多いことを検証

最大でないマッチングと増加経路の
対称差をとるハードウェアに
問題が見出された

バグの特定②

```
core_dmu_V: assert property
  ( @(posedge CLOCK1)
    (LAST==0)##1(LAST==1)|->CHECKV);
```

増加経路の端点が
不飽和点になっていることを検証

増加経路を求めるハードウェアに
問題が見出された

まとめと今後の課題

- **まとめ**

実現するアルゴリズムの証明に注目したABVで設計したASIPのバグを見出せることを示した

- **今後の課題**

他のアルゴリズムに対しても同様な手法が適用できるかを確認すること