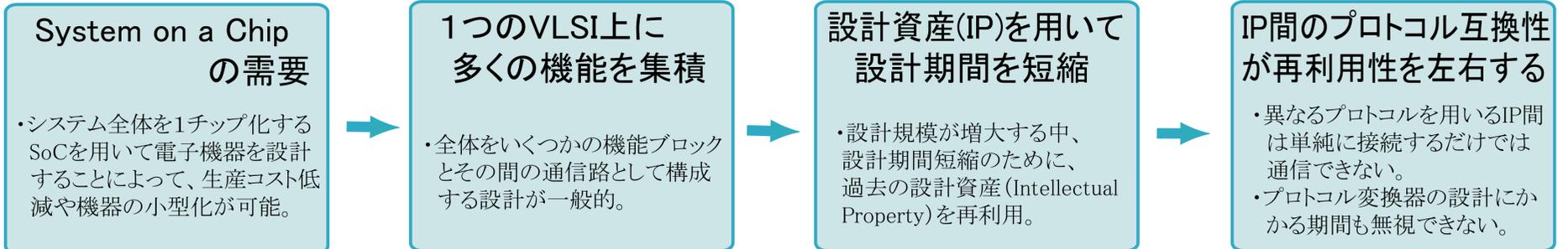


## 研究の背景

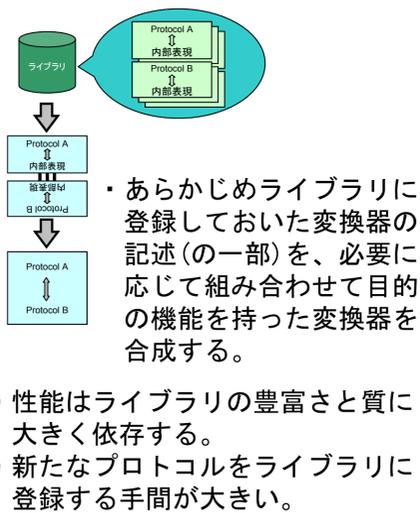


## プロトコル変換器の自動合成技術が鍵！

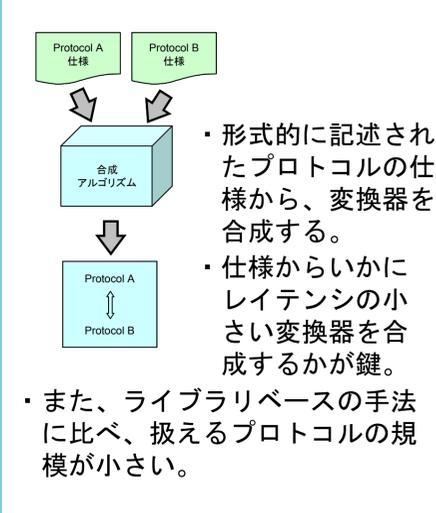
### プロトコル変換器の自動合成手法

- 大きく分けて2つのアプローチが存在する。

#### ライブラリベース



#### 直接合成



### 本研究の目的と既存手法の問題点

- 本研究では、直接合成アプローチを用いる合成手法であるPasseroneらの手法<sup>\*1</sup>をベースとする。
- 彼らの合成手法は、2つのプロトコルを表すオートマトンの状態を依存関係に違反しないように融合し、変換器のFSMを探索する。
- この手法を拡張し、扱えるプロトコルの規模を拡大する。
- Passeroneらの手法では、通信路上を流れるデータの量が順番が合成時に既知であると仮定している。その為、無限に続く通信は探索空間が無限になり、扱うことができない。
- また、計算量が $O(M \times N)$ であるため、大規模なプロトコルに適用するのが難しい。

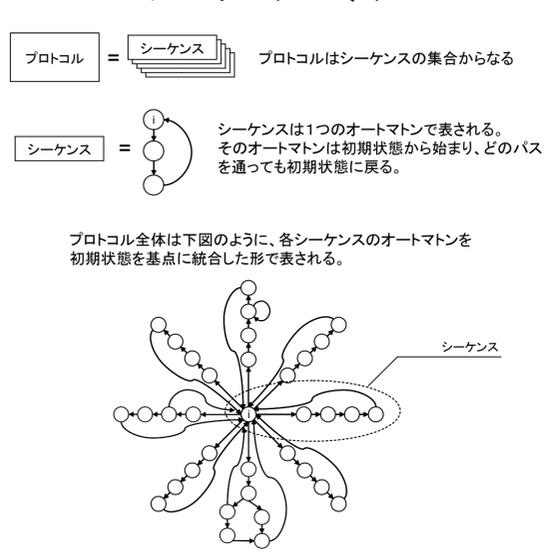
\*1 R.Passerone, J.A.Rowson, A.Sangoivanni-Vincentelli, "Automatic Synthesis of Interfaces between Incompatible Protocols", DAC'98.

## 提案手法によるプロトコル変換器自動合成

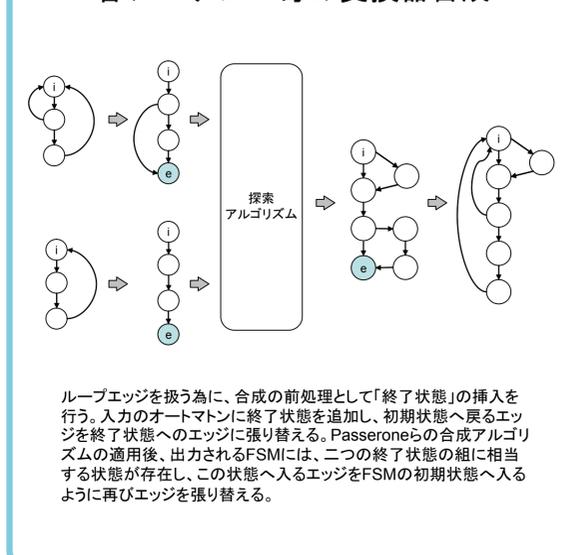
我々はPasseroneらの手法をベースに、下図で紹介する3つの要素技術を拡張し、プロトコル変換を行えるプロトコルの規模を拡大した。当研究室ではこの手法を用いて、Single R/W, Burst R/Wシーケンスを扱うOCP⇔AMBA変換器を合成することができる事を確認している。

また、この手法では、シーケンス毎にプロトコル変換器の合成を独立に行う事ができるため、計算量を $O(M+N)$ に留める事ができる。

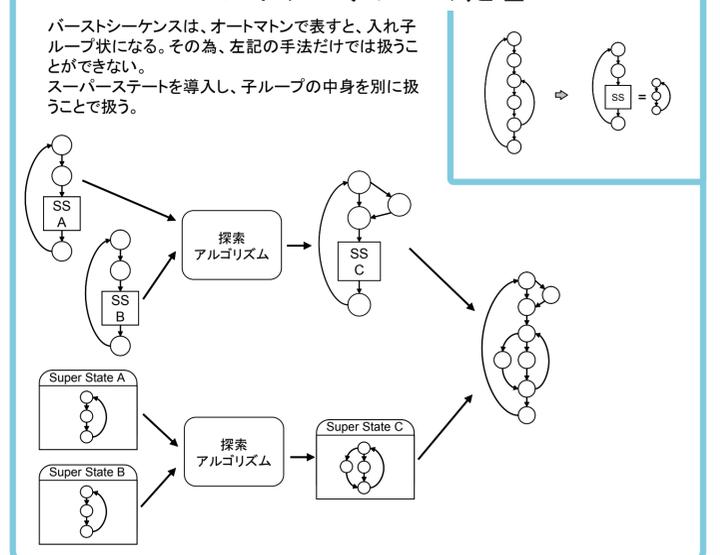
#### プロトコルモデル



#### 各シーケンス毎の変換器合成



#### バーストシーケンスの処理



右図に合成したAMBA→OCP変換器のFSMを示す。シーケンス毎に変換器を合成する事により、1回の合成で探索する解空間が小さくなった為、合成にかかった時間は1.32[sec]と短かった。今後は、仕様書からオートマトン表現に変換する作業の省力化を目指す。

