

EDAと離散構造処理

九州大学大学院システム情報科学研究
院

松永 裕介

(matsunaga@ait.kyushu-u.ac.jp)



BDDとの関わり

- 1989: BDDを用いた多段論理簡単化アルゴリズムの開発
- 1993: BDDを用いた再帰的な推移的閉包計算アルゴリズムの開発
- 1996: 大規模組み合わせ回路論理回路の等価検証アルゴリズムの開発
- 1998: BDDを用いた効率的な単純直交分解アルゴリズムの開発

そういえば、最近、BDDをつかってないな～

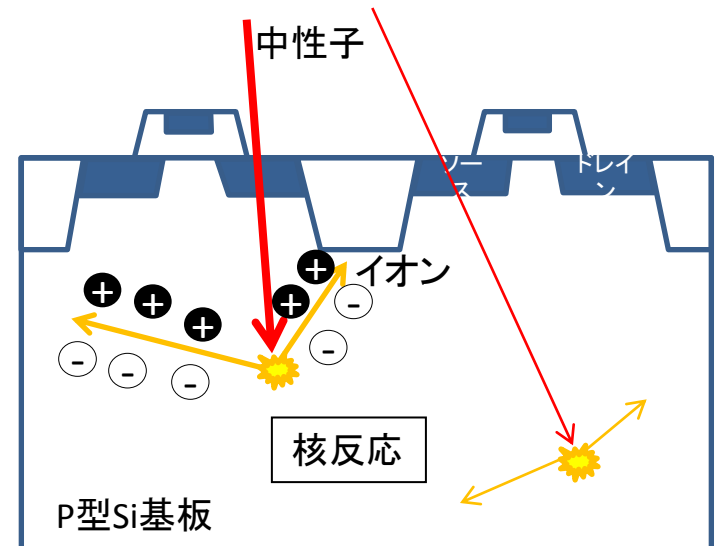
ソフトウェア

放射線によってLSIの動作不良が引き起こされる現象

- 放射線による、LSI内の論理ゲートの異常パルスの発生やレジスタのビット反転(エラー)は、誤った最終出力(Failure)を発生させる可能性がある
- ハードエラー(LSIそのものの不良による恒久的な故障)とは異なる対策と評価手法が必要となる
- トランジスタの寸法縮小, 低消費電力化による臨界電荷量の低下による**エラーの増加**(2004年の報告:1GBのメモリで1~5日に一度の頻度で発生)

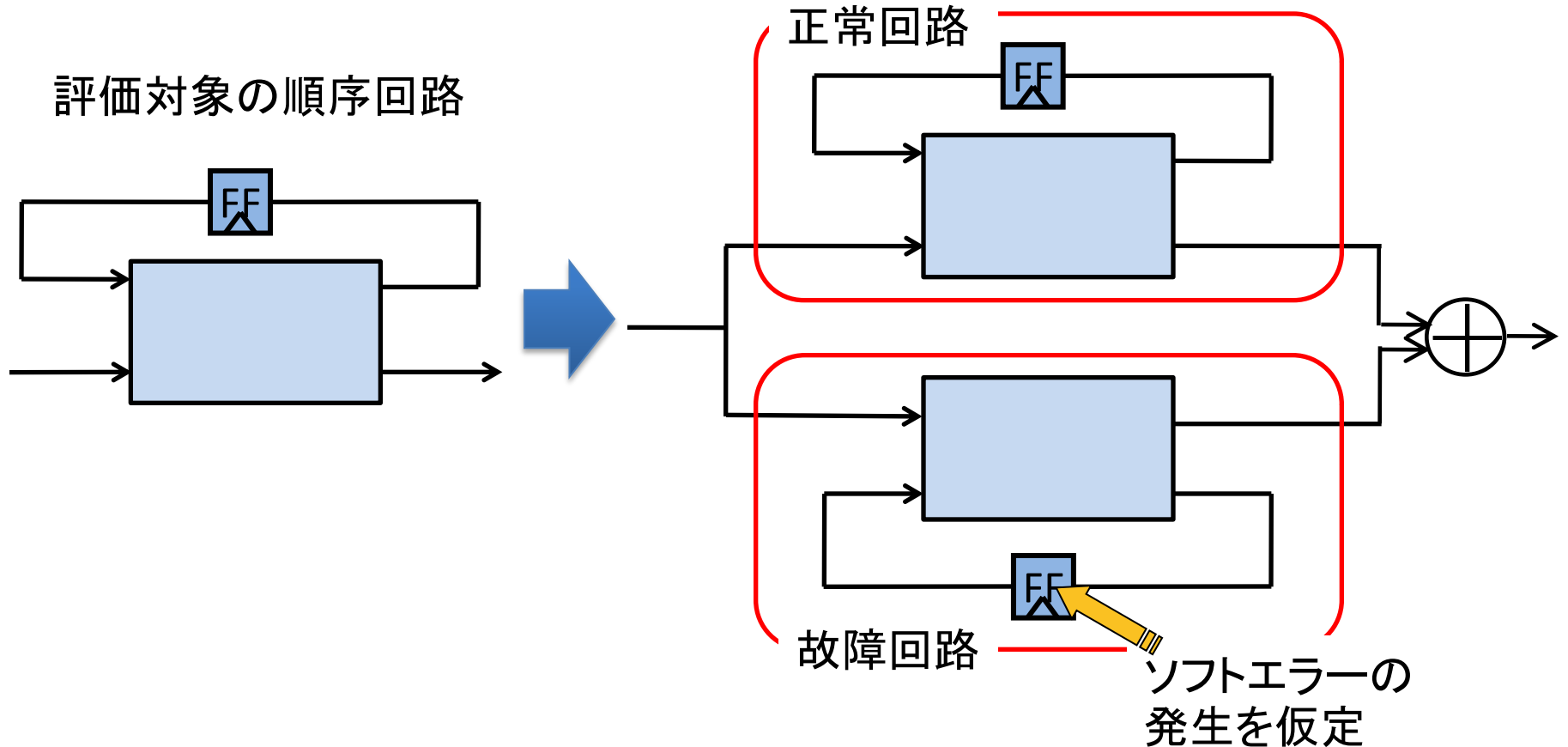


Failureの増加



回路の設計者は回路の信頼性を保つため、ソフトウェアから回路を守る手法を考える必要がある

正常な振る舞いと エラーを含む振る舞いの比較



評価対象の回路を複製し、**回路対**とする

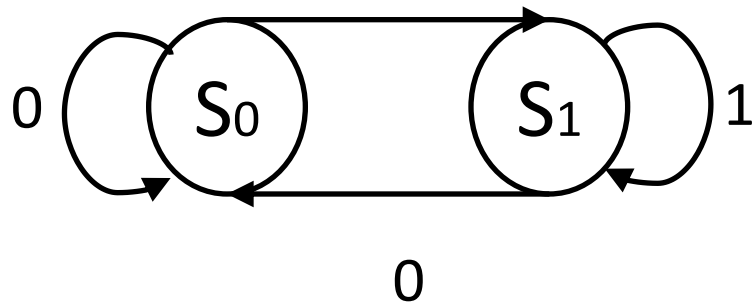
- 正常回路と故障回路の出力値が異なるとき、(回路対の出力のXORの値が1のとき)エラーが外部出力に伝搬したといえる

状態遷移の確率的な解析

順序回路の状態遷移をマルコフモデルで表わす
⇒回路の状態遷移の振る舞いを確率的に解析する

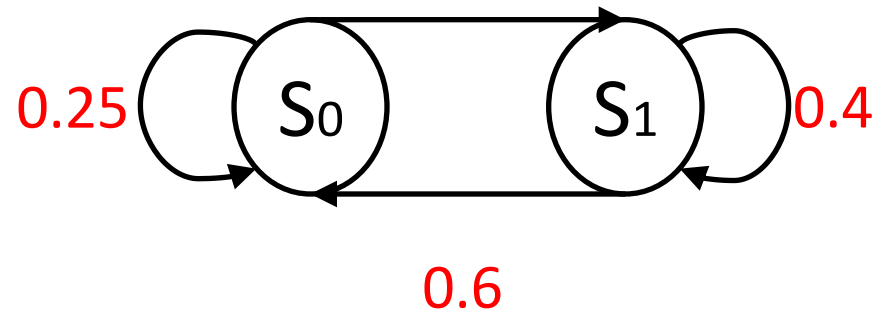
決定的な状態遷移モデル

入力:1

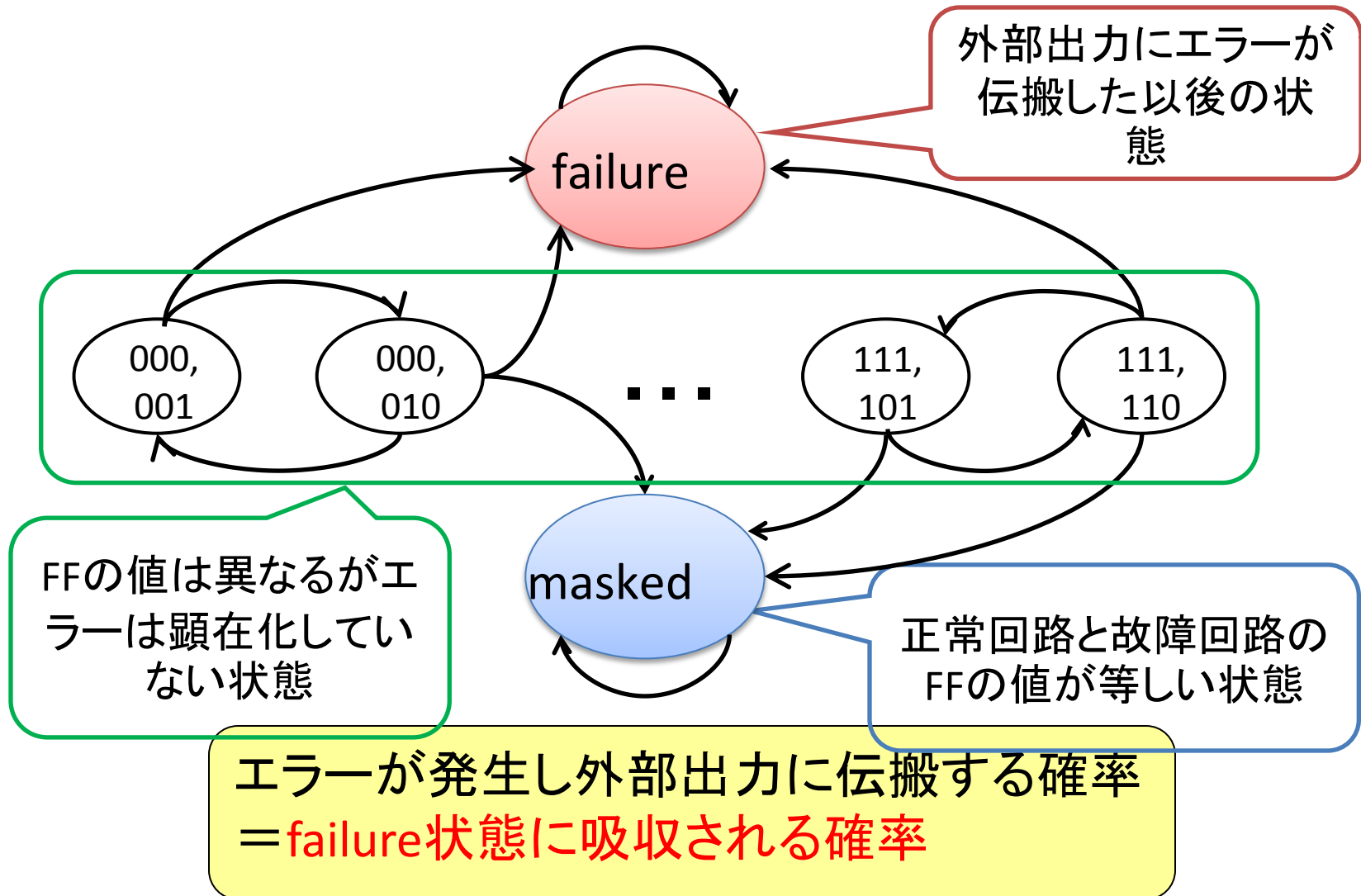


確率的な状態遷移モデル

遷移確率:0.75



回路対の状態遷移のマルコフモデル

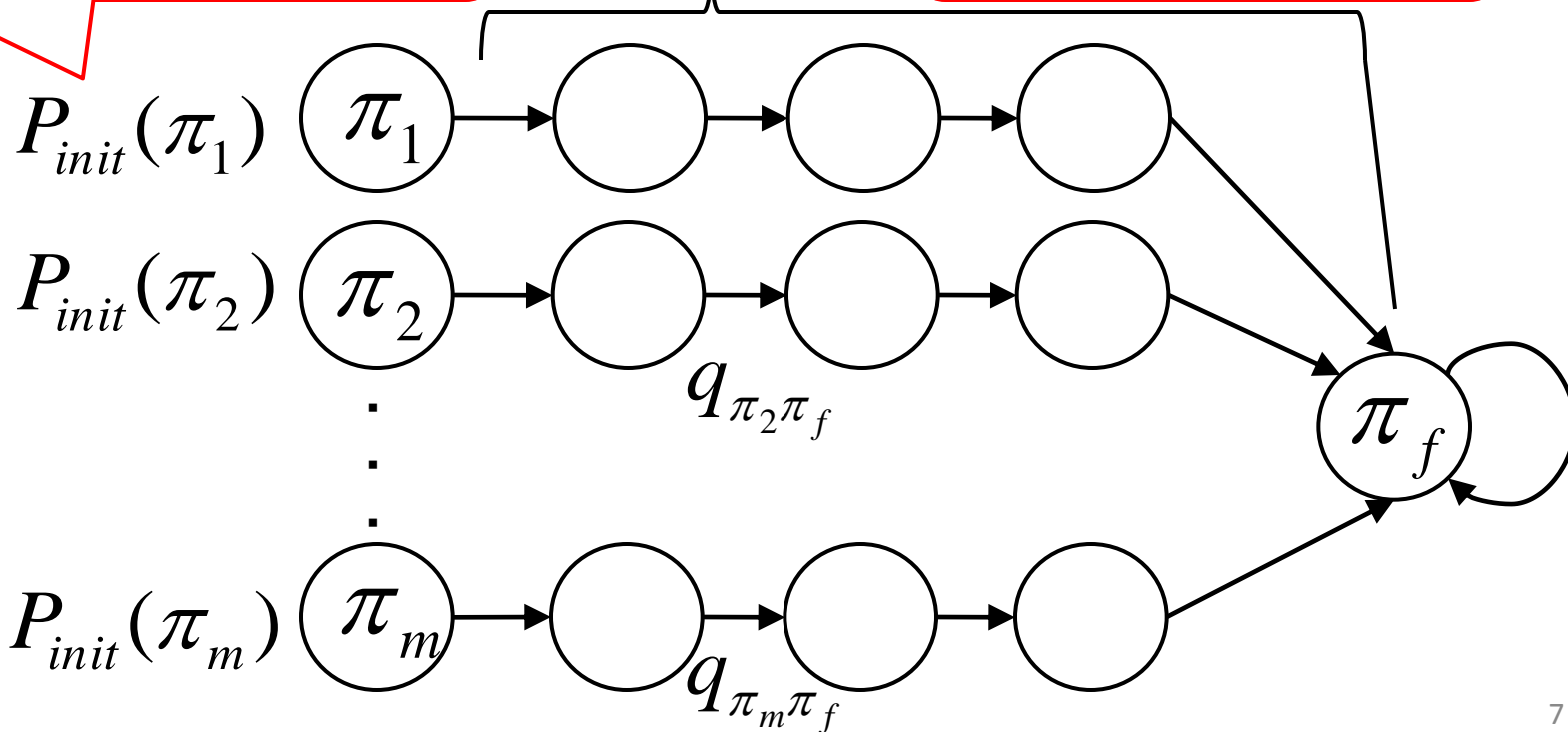


failure状態に吸収される確率

$$P_{abs}(\pi_f) = \sum_{\pi \in \Pi} P_{init}(\pi) \cdot q_{\pi\pi_f}$$

ソフトウェア発生直後の
状態が π_1 である確率

π_1 を出発して π_f に
吸収される確率

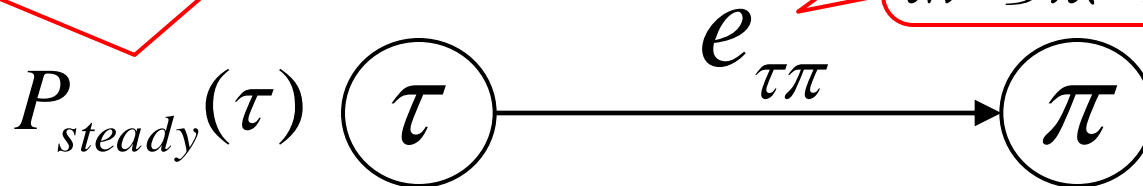


初期状態確率の計算

$$P_{init}(\pi) = \sum_{\tau \in T} P_{steady}(\tau) \cdot e_{\tau\pi}$$

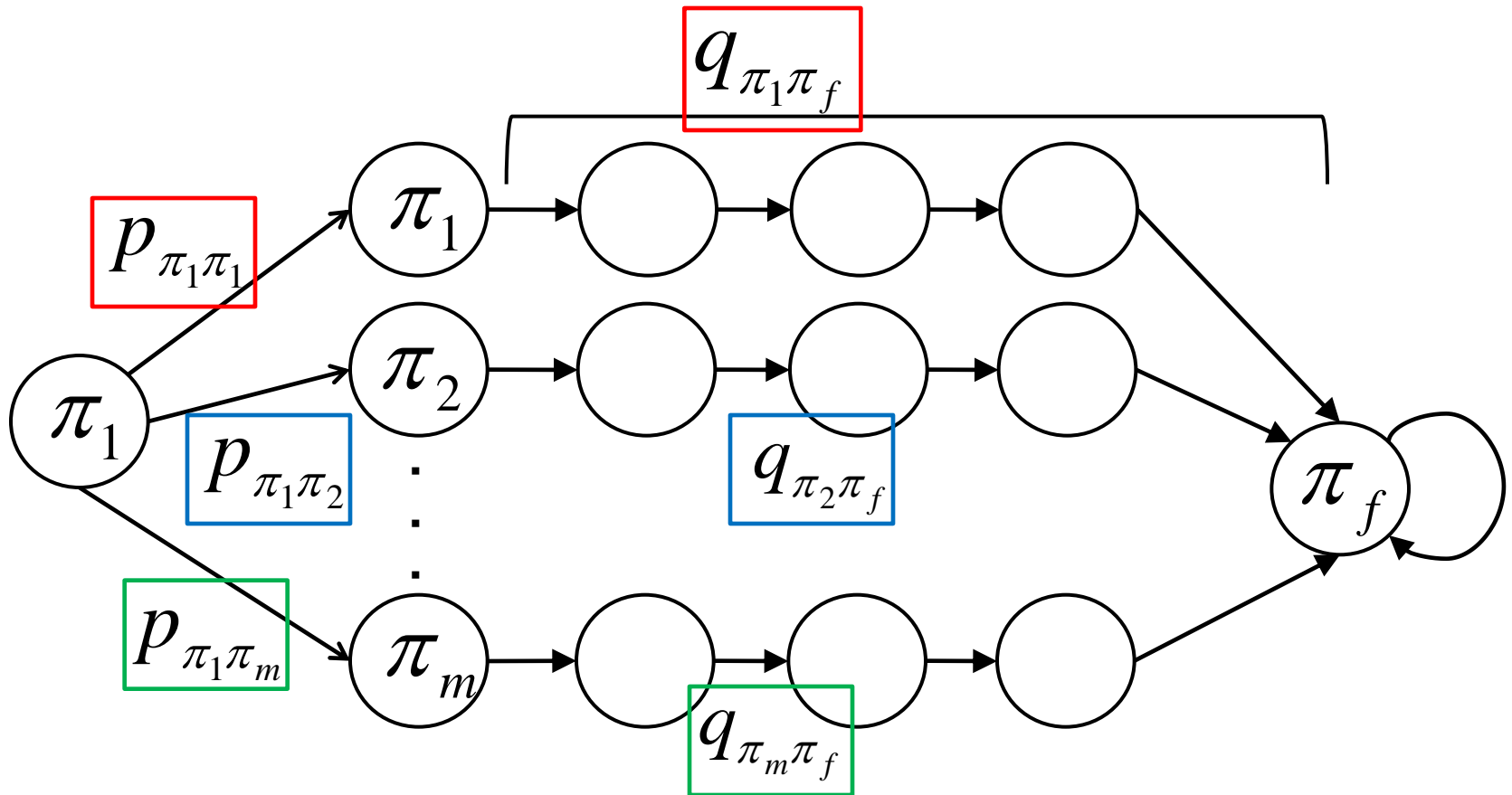
ソフトウェア発生以前の
ある時刻の状態が τ である確率

ソフトウェアにより状態が
 τ から π に遷移する確率



- $P_{steady}(\tau)$
 - ソフトエラー発生以前の各状態間の遷移確率が既知であれば連立方程式を解くことにより計算可能
(遷移確率は論理シミュレーションにより計算)
- $e_{\tau\pi}$ ソフトエラーが発生する確率(given)から計算可能

π を出発し π_f に吸収される確率の計算

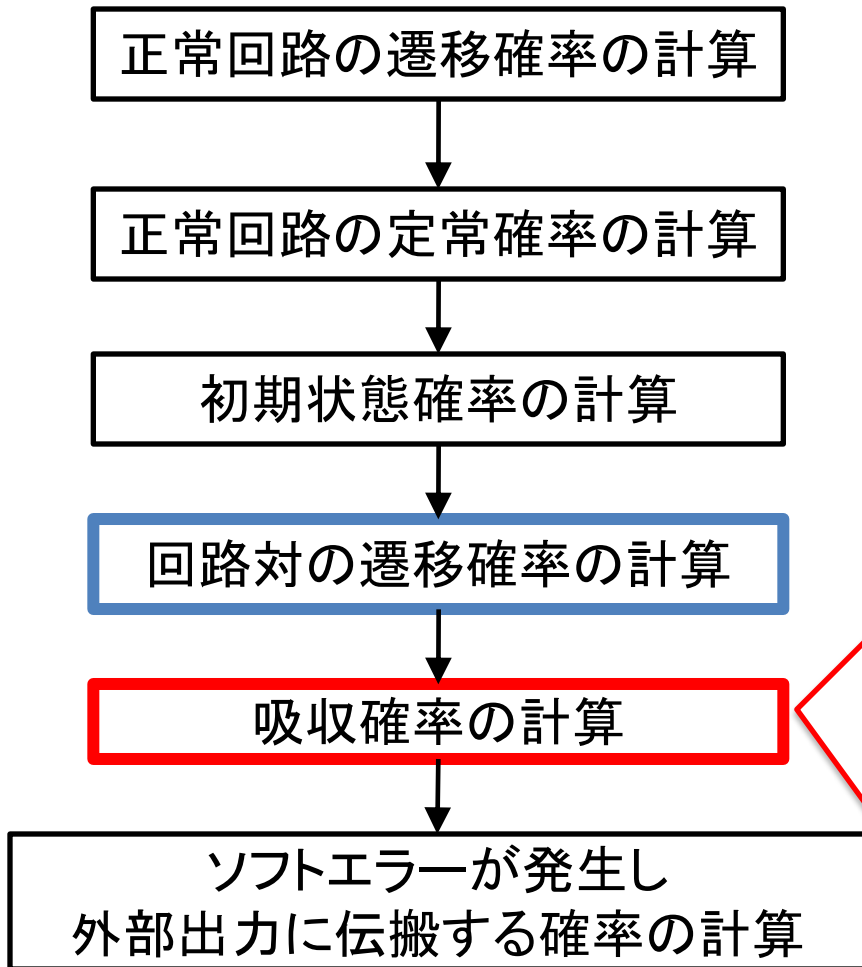


$$q_{\pi_1\pi_f} = p_{\pi_1\pi_1} \cdot q_{\pi_1\pi_f} + p_{\pi_1\pi_2} \cdot q_{\pi_2\pi_f} + \dots + p_{\pi_1\pi_m} \cdot q_{\pi_m\pi_f} + p_{\pi_1\pi_f}$$



全ての π に対して式を立て遷移確率が既知であれば
 q を未知数とする連立方程式となる

処理の枠組みと計算時間



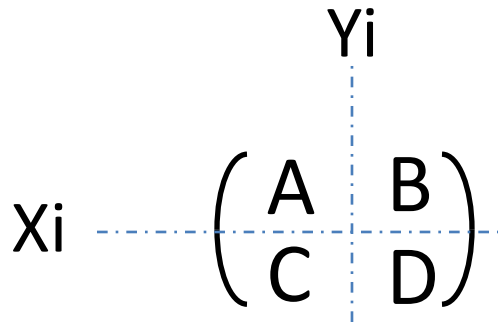
連立方程式を解く

- ガウスの消去法では元数の3乗に比例した時間を要する
- 元数 = 回路対の状態数
- 回路対の状態数 $\leq 2^{2k}$
(k : 正常回路のFF数)
(つまり 2^{6k} に比例した時間)

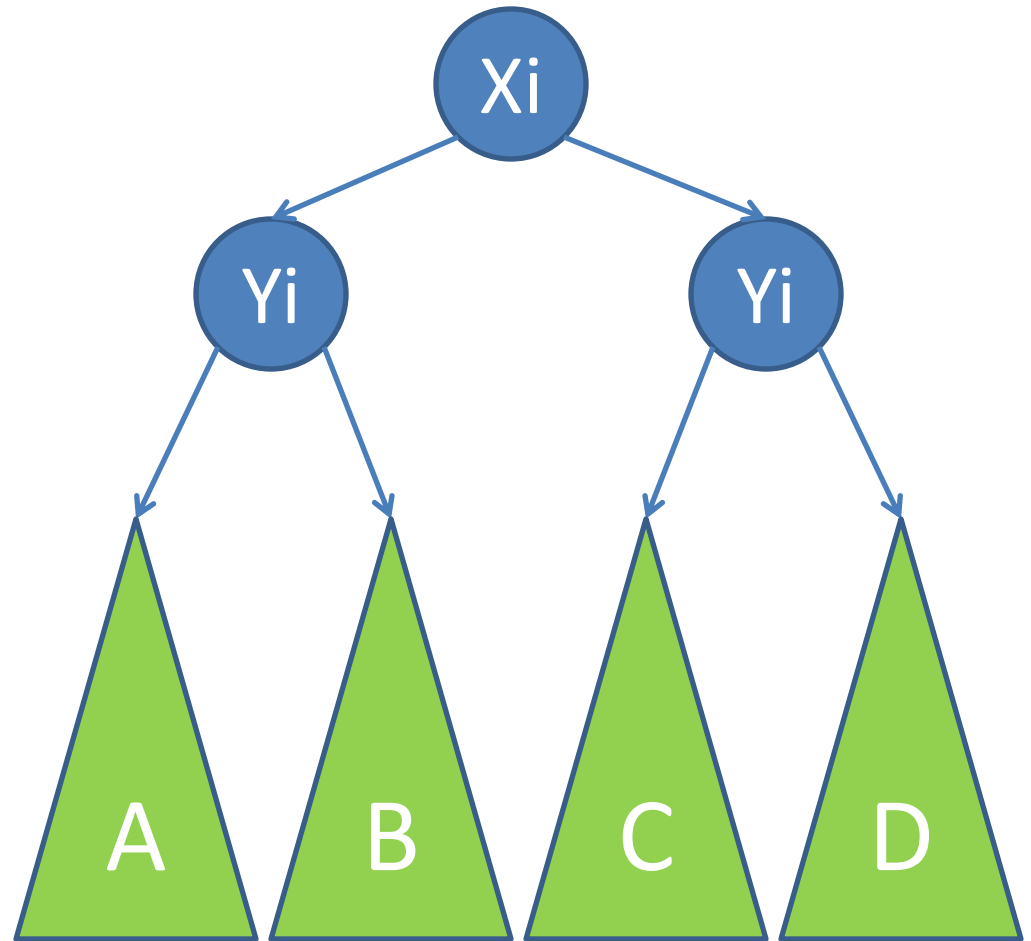
例: s382

- 全体 : 89398秒
(\approx 24時間)
- 吸収確率の計算 : 87834秒

BDDと行列演算

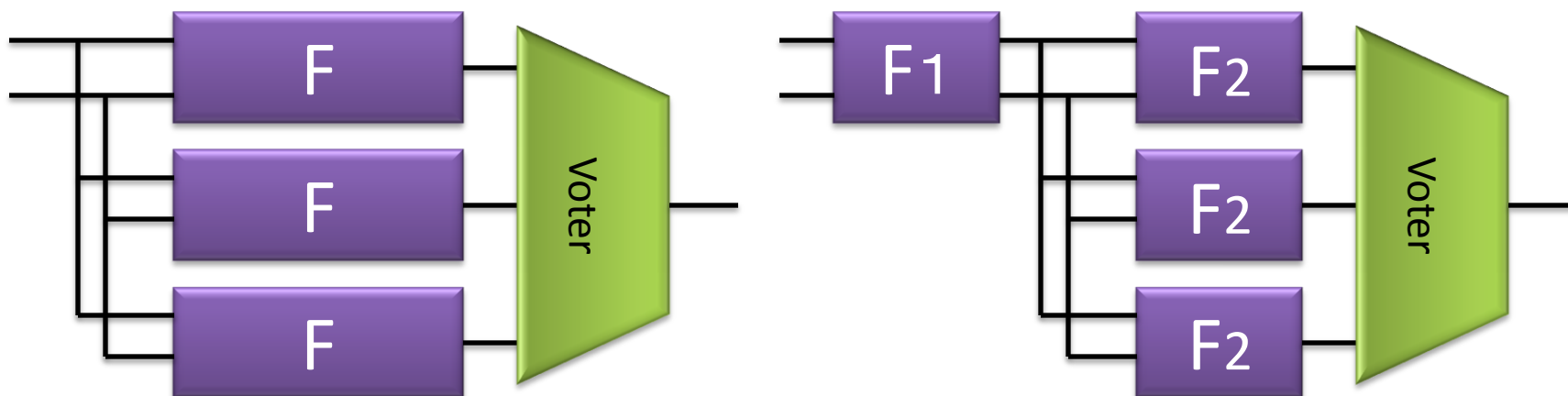


- この分割を再帰的に行う
- \Rightarrow BDDを上からたどる。
- 推移的閉包の効率的な計算アルゴリズム

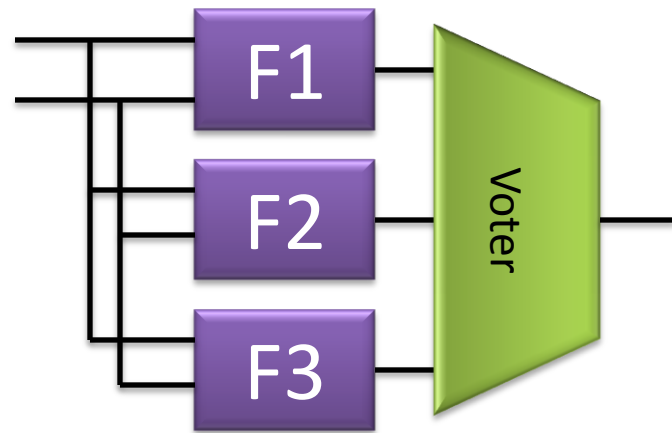


TMR(Triple Modular Redundancy)

- 回路を3重化して出力の多数決をとれば単一の故障はすべて訂正可能
- でも3倍の回路量のオーバーヘッドは大きすぎる
- ⇒部分的に3重化する



関数の近似を用いた部分的なTMR



- 元の関数 F に対して次のような関数 $F1, F2, F3$ を考える。
 - いかなる入力 x に対しても $F1, F2, F3$ の2つ以上の関数が F と等しい値を出力する。
- すると多数決回路の出力は常に F と等しくなる。
- $F1, F2, F3$ の回路が F よりも大幅に小さくできればOK

関数の近似

- とりあえず
– $F1 = F$
– $F2 > F$
– $F3 < F$
- とする。
- 上記の $F2$ と $F3$ の条件を満たすなかで面積の小さな回路をもとめたい。
- 従来の論理合成の目的 \Rightarrow 与えられた論理関数を厳密に実装すること。
- 現在は2段論理(積和形論理式)に対するナイーブな貪欲算法のみ

おわりに？

- Classical な論理合成、論理検証の分野におけるBDDの応用は結構、出尽くした感あり。
 - ⇒結局、BDDで表せれば使えるし、BDDが爆発したら使えない。
- より大規模な回路(論理関数)を近似的に扱うような手法には、理論的にも実用的にも面白そうなテーマがありそう。