

論理回路の組込み自己診断に関する提案

○香川 敬祐, 矢野 郁也, 王 森レイ, 樋上 喜信, 高橋 寛
(愛媛大学 大学院 理工学研究科)

大竹 哲史(大分大学 工学部)

発表概要

- 研究背景・研究目的
- 組込み自己診断(BISD)についての先行研究
- 提案するBISD機構
 - ・ 診断用署名生成器
 - ・ 診断用署名に基づく故障候補の指摘器
- 予備実験結果
 - ・ 故障診断能力の評価結果
 - ・ 被疑故障署名を格納するメモリサイズの試算結果
- まとめ・今後の課題

研究背景

- 先進自動運転技術などの高度なシステムの実現が急
→自動車分野における集積回路の役割の拡大
- 先進自動運転を備えた自動車の機能安全の確保が急務
→機能安全を確保できる車載用の集積回路の開発



- 解決法: パワーオンセルフテスト(Power On Self Test: POST)
向けのテスト技術の提案[1]

[1]S. Wang et al., “ Structure-Based Methods for Selecting Fault-Detection-Strengthened FF under Multi-Cycle Test with Sequential Observation ,” Proc. ATS, 2016.

研究背景

- POSTにおいてフィールド上で長期の機能安全を確保



- 故障検出(故障の有無)だけでなく故障診断(故障箇所の推定)



- 組込み自己テストから
組込み自己診断(Built-In Self Diagnosis:BISD)の開発へ

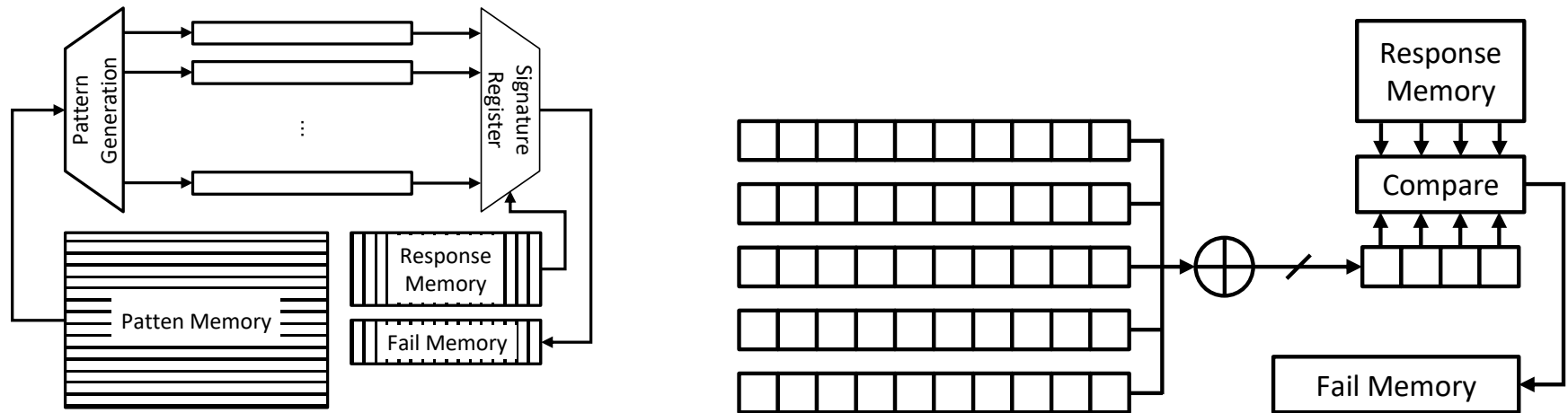
研究目的

- 組込み自己診断機構の課題の整理
- 組込み自己診断機構を提案
 - ・ 診断用署名生成器
 - ・ 診断用署名に基づく故障候補の指摘器
- 組込み自己診断機構に関する予備実験
 - ・ パス/フェイル情報の圧縮と故障診断能力の評価
 - ・ 被疑故障署名を格納するためのメモリサイズの試算

BISDについての先行研究[4](1)

● BISD機構

- 擬似ランダムパターンを利用したBIST機構を拡張
- より高い故障検出率を得るため決定的パターンを追加

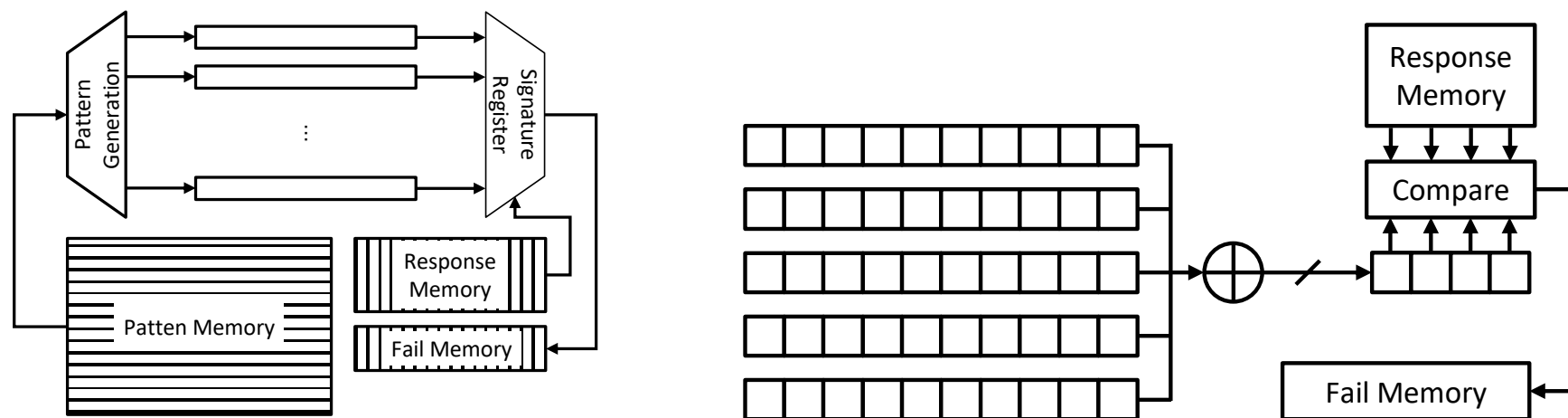


[4]M.Elm, H.Wunderlich, “BISD:Scan-Based Built-In Self-Diagnosis,” Proc. DATE, pp. 1243–1248, 2010.

BISDについての先行研究[4](2)

● BISD機構

- 縮退故障: f , テストパターン: $p \in T$
- 参照署名: $s_{ref}(f, p)$, $S_{ref}(f) = \bigcup_{p \in T} \{s_{ref}(f, p)\}$
- 応答署名: S_{obs}



BISDに関する課題の整理(1)

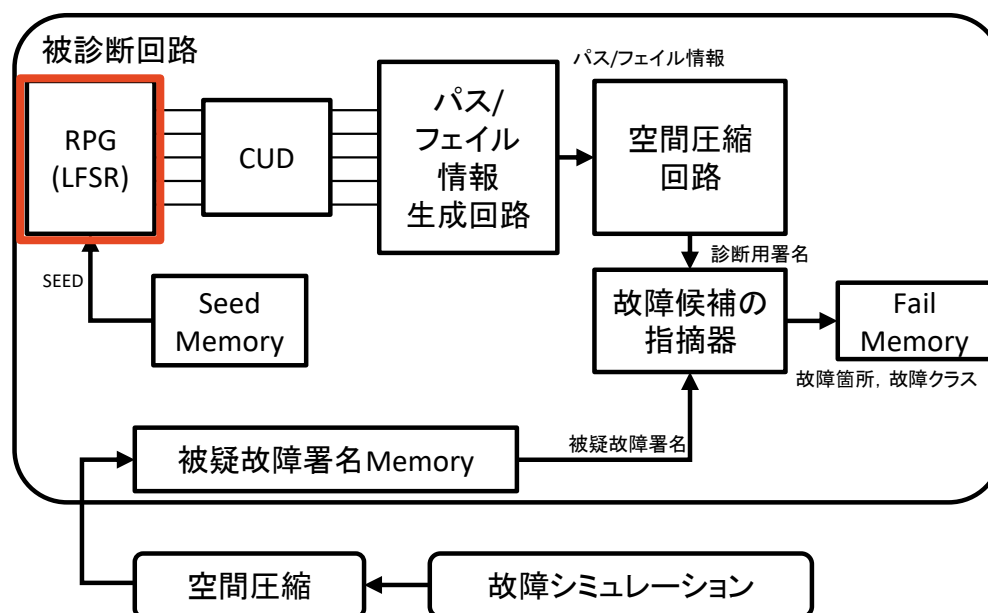
1. テストパターン系列に対する被検査回路のパス/フェイル情報を生成するための仕組みが必要
2. テストパターンに対する期待値署名を記憶するためのメモリのサイズに対する考慮が必要
3. 被検査回路のパス/フェイル情報に基づいて診断用署名を生成するための仕組みが必要

BISDに関する課題の整理(2)

4. 被検査回路から得られた診断用署名と
予め格納された被疑故障署名を比較して、
故障箇所を推定する仕組みが必要
5. 圧縮後の診断用署名と故障診断分解能の精度の関係を
解明しなければならない

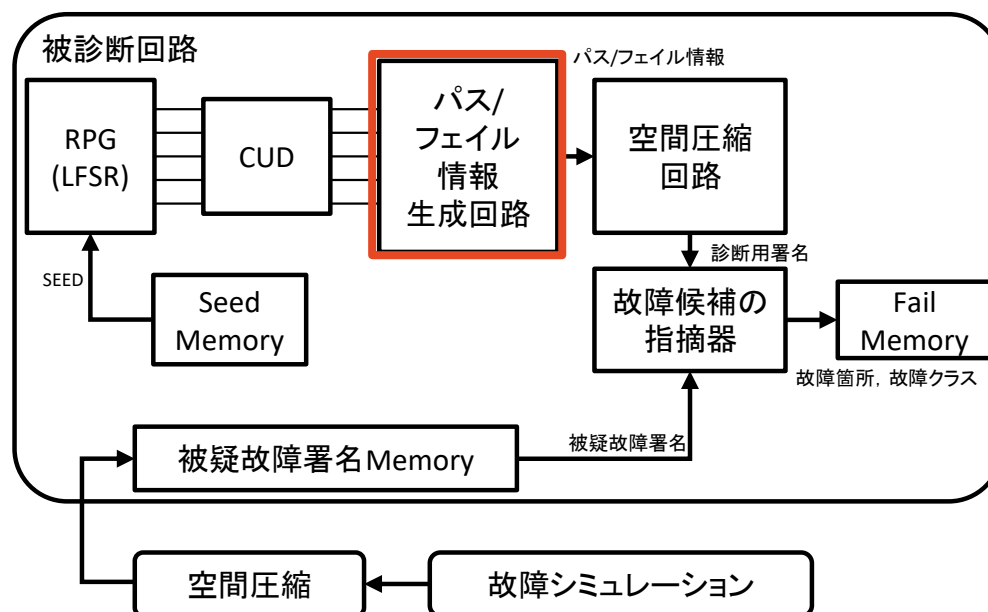
提案するBISD機構(1.テスト生成回路)

- 故障診断能力を向上させるための
ランダムパターン系列を生成 (文献[5]で提案)



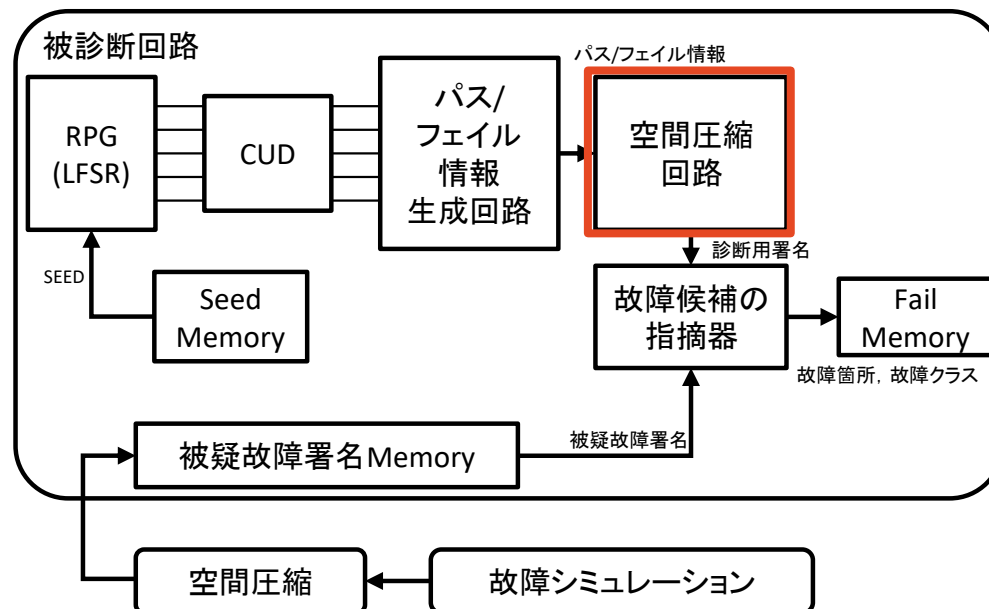
提案するBISD機構(2.パス/フェイル情報生成回路)

- テストパターンごとにCUDの故障検出情報であるパス/フェイル情報を生成(特許申請準備中)



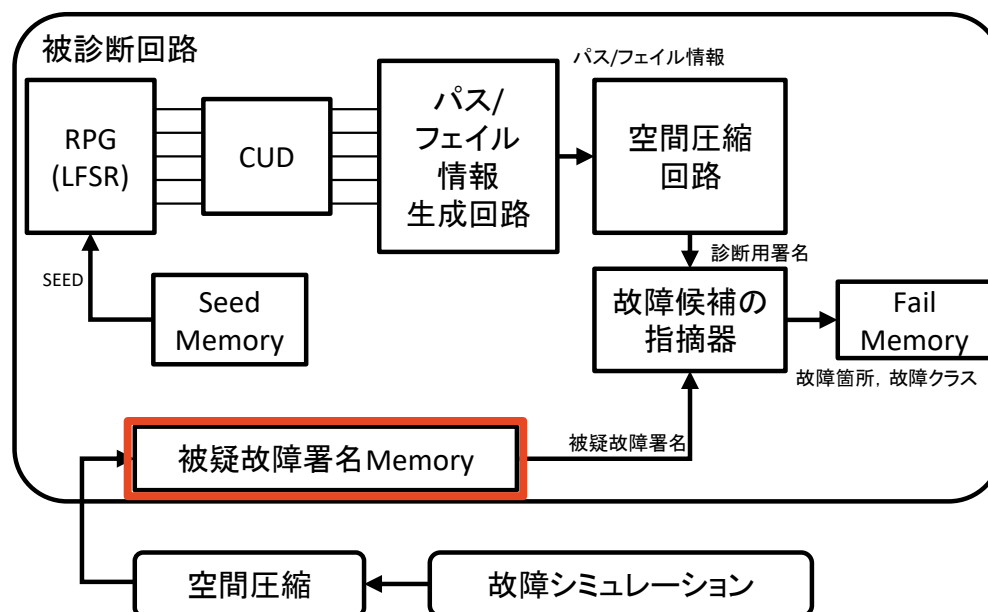
提案するBISD機構(3.空間圧縮回路)

- テストパターンごとのパス/フェイル情報を圧縮して
テストパターン系列に対する診断用署名を生成



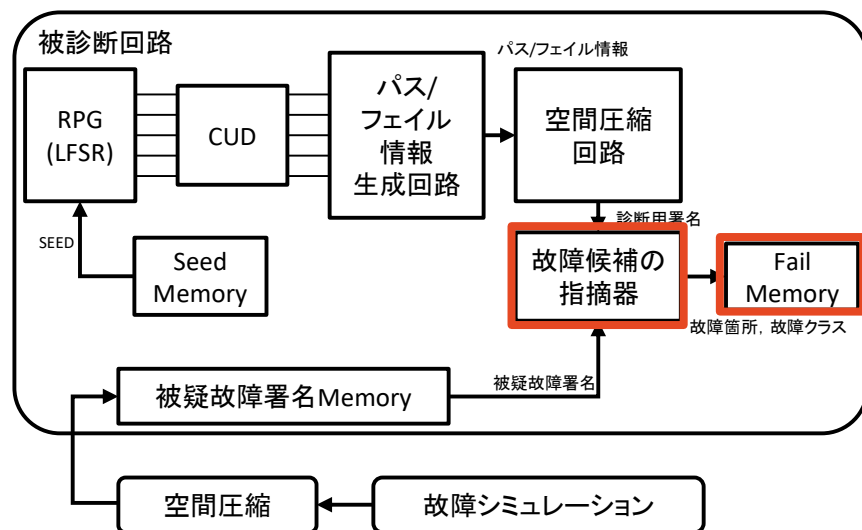
提案するBISD機構(4.被疑故障署名メモリ)

- 被疑故障ごとの診断用署名を格納



提案するBISD機構(5.故障候補の指摘器)

- 診断用署名と被疑故障署名を比較して
署名が一致したアドレス情報から被疑故障を推定



| テスト パターン 系列 T | パス/ フェイル 情報 |
|-----------------------|-------------------|
| \vdots | \vdots |
| t_i | 1 |
| t_j | 1 |
| t_k | 0 |
| \vdots | \vdots |

圧縮

| テスト パターン 系列 | 診断用 署名 |
|-------------------|-----------|
| T | S_{f_i} |

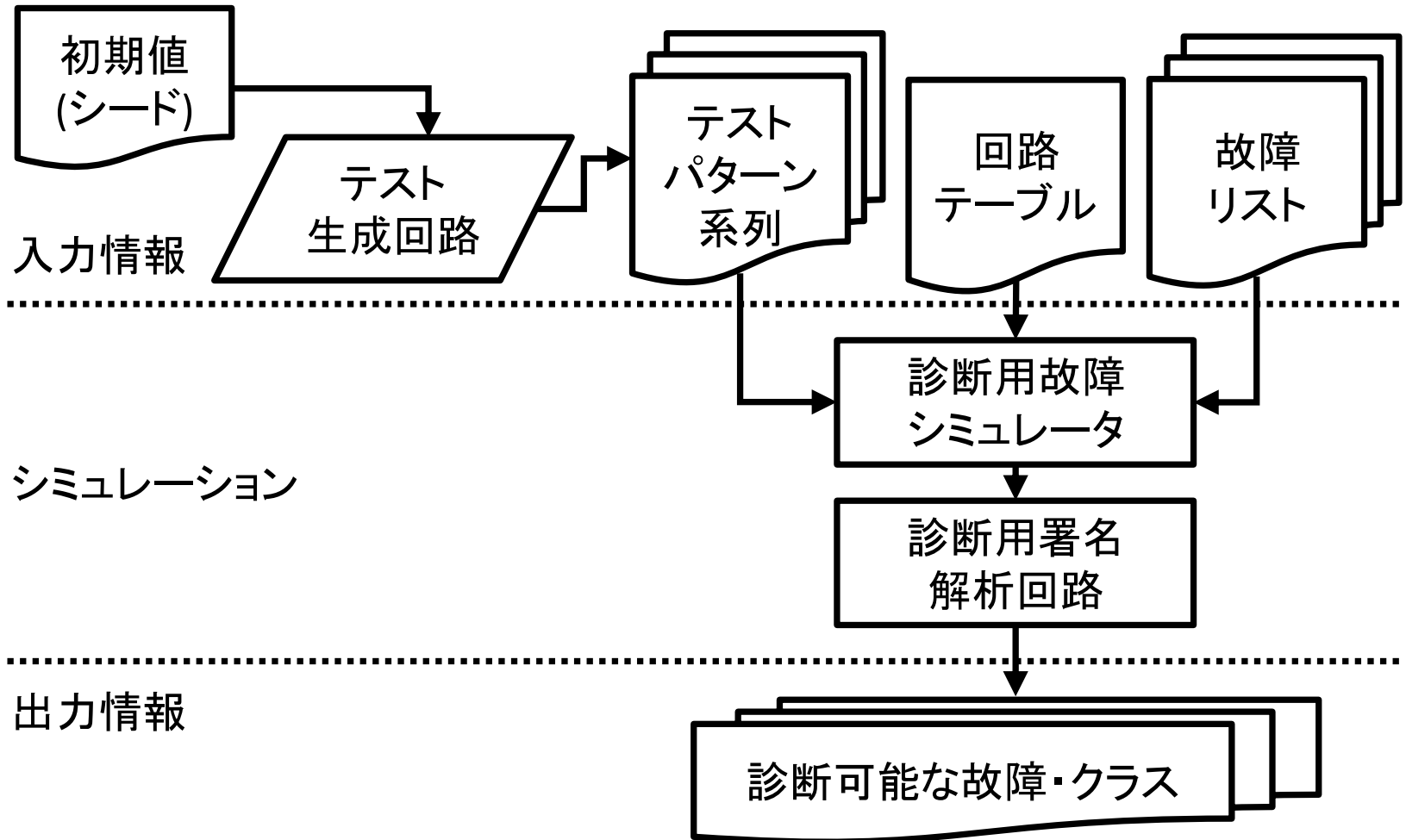
比較

故障箇所
 f_j

1:検出
0:非検出

| 被疑故障 F | \cdots | f_i | f_j | f_k | \cdots |
|------------|----------|-----------|-----------|-----------|----------|
| 被疑故障署名 S | \cdots | S_{f_i} | S_{f_j} | S_{f_k} | \cdots |

BISD機構のシミュレーションモデル



BISD機構の故障診断能力(診断可能な故障)

- 診断用署名(S_*)がほかの故障と比べて唯一
- 診断可能な故障数=3

| | f_1 | f_2 | f_3 | f_4 | f_5 | f_6 | f_7 | f_8 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| t_1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| t_2 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| t_3 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| t_4 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| t_5 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| | S_1 | S_2 | S_3 | S_3 | S_4 | S_2 | S_5 | S_3 |

1:検出
0:非検出

BISD機構の故障診断能力(クラス)

- 診断用署名(S_*)が同一な故障をクラス化
- クラス数=2, クラス内の最大故障数=3

| | f_1 | f_2 | f_3 | f_4 | f_5 | f_6 | f_7 | f_8 |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| t_1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| t_2 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| t_3 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| t_4 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| t_5 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| | S_1 | S_2 | S_3 | S_3 | S_4 | S_2 | S_5 | S_3 |

1:検出
0:非検出

診断用署名生成器

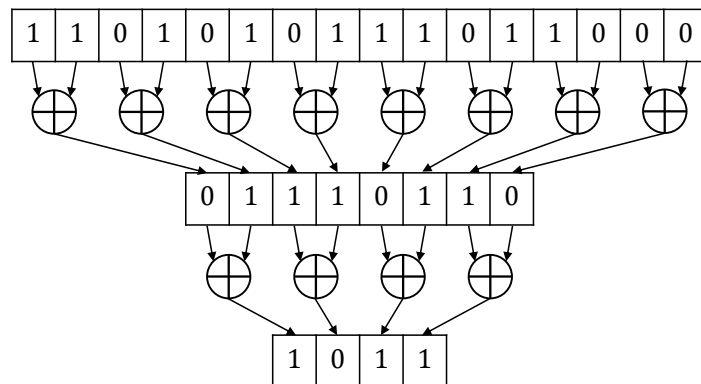
- 3種類の診断用署名生成器を考案

1. XORゲートを利用したアルゴリズムを実装する回路
2. ハッシュ関数を利用したアルゴリズムを実装する回路
3. XORゲートとハッシュ関数を併用する回路

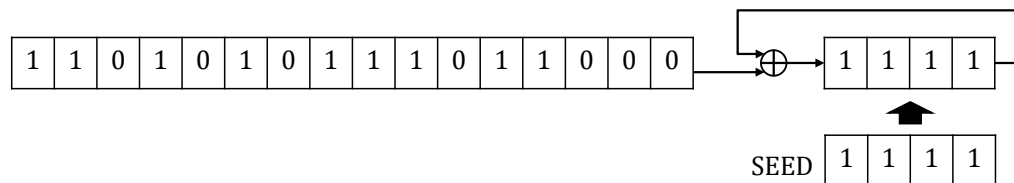
診断用署名生成器

1. XORゲートを利用したアルゴリズムを実装する回路

- XOR_1



- XOR_2



診断用署名生成器

2. ハッシュ関数を利用したアルゴリズムを実装する回路

- ハッシュ関数
 - ✓ 入力に対して固定長の擬似乱数列(ハッシュ値)を生成
- SHA-256
 - ✓ 256bit($32\text{bit} \times 8$)のハッシュ値を生成
- SHA-512
 - ✓ 512bit($64\text{bit} \times 8$)のハッシュ値を生成

診断用署名生成器

2. ハッシュ関数を利用したアルゴリズムを実装する回路

- ハッシュ関数の特徴

- ✓ 任意のメッセージから固定長のハッシュ値を計算可能

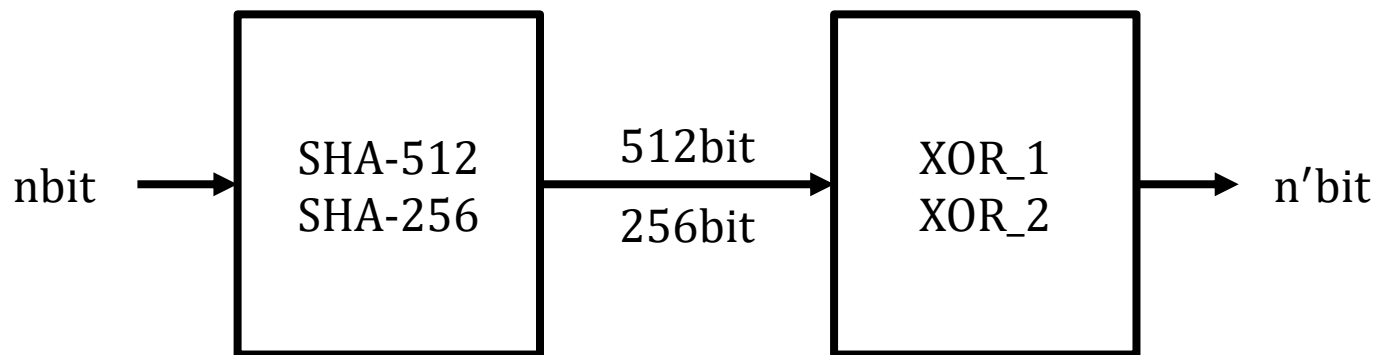
- ✓ 高速にハッシュを計算可能:
高速に計算(圧縮)できることは
オンチップで空間圧縮を行うために重要

- ✓ メッセージが異なればハッシュ値も異なる:
高い衝突耐性は圧縮による情報のエイリアシングを
防ぐために重要

診断用署名生成器

3. ハッシュ関数とXORゲートを併用する回路

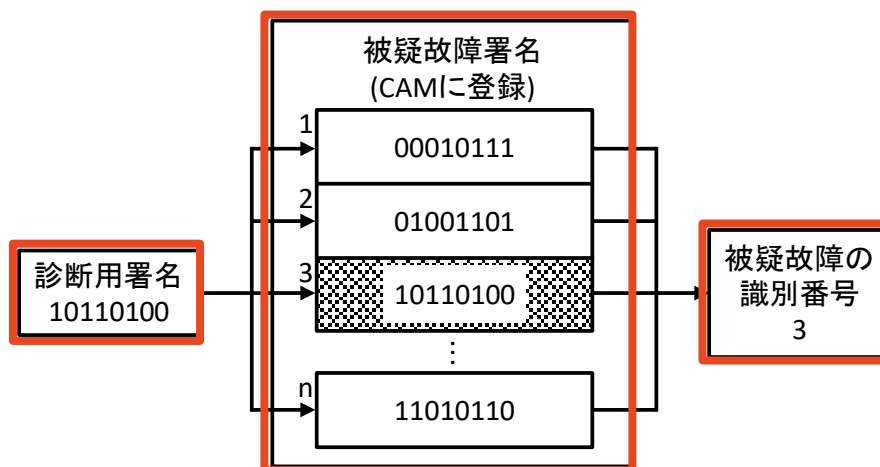
- ハッシュ値はビット幅が固定かつ大きい(256/512[bit])
- XORゲートを利用することで圧縮の効果を高める



診断用署名に基づく故障候補の指摘器

- 故障候補の指摘器の仕様

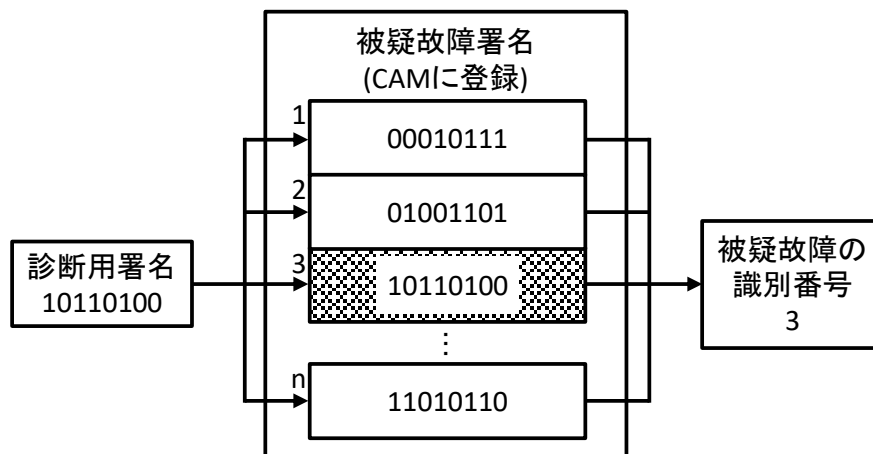
- 入力: 被検査回路から得られた診断用署名
- 出力: 診断用署名と一致した被疑故障署名の識別番号
- 記憶装置: 連想メモリ(Content Addressable Memory: CAM)



診断用署名に基づく故障候補の指摘器

● 故障候補の指摘器に必要な機能

1. 被疑故障署名の識別番号をアドレスとしてその被疑故障の被疑故障署名を格納
2. 被検査回路から得られた診断用署名と一致する被疑故障署名を探索



故障診断能力の評価結果

● 故障診断能力の評価に利用した回路の諸元

| 回路名 | ゲート数 | FF数 | | |
|-----------------------|--------------------|--------------|-------|----------------|
| cs9234 (ISCAS' 89) | 5,597 | 228 | | |
| 対象 故障数 | テスト パターン 系列数 | 診断可能な 故障数 | クラス数 | クラス内の 最大故障数 |
| 7,756 | 2,048 | 659 | 1,443 | 45 |

故障診断能力の評価結果

● XORゲートを利用した診断用署名生成器

- ・ 圧縮ビットの幅を短縮すると故障診断能力が低下

(診断可能故障数/659*100 %)

| 圧縮 ビット幅 [bit] | XOR_1 | | | XOR_2 | | |
|---------------------|--------------|-------|----------------|--------------|-------|----------------|
| | 診断可能な 故障数 | クラス数 | クラス内の 最大故障数 | 診断可能な 故障数 | クラス数 | クラス内の 最大故障数 |
| 256 | 629(95.45%) | 1,243 | 48 | 640(97.12%) | 1,246 | 48 |
| 128 | 627(95.14%) | 1,240 | 48 | 635(96.36%) | 1,245 | 48 |
| 64 | 624(94.69%) | 1,232 | 48 | 630(95.60%) | 1,243 | 48 |
| 32 | 608(92.26%) | 1,222 | 48 | 624(94.69%) | 1,231 | 48 |
| 16 | 571(86.65%) | 1,182 | 72 | 583(88.47%) | 1,195 | 52 |

故障診断能力の評価結果

- ハッシュ関数を利用した診断用署名生成器
 - ・ 圧縮なしの場合と同じ故障診断能力

| 圧縮 ビット幅 [bit] | SHA-256 | | |
|---------------------|--------------|-------|----------------|
| | 診断可能な 故障数 | クラス数 | クラス内の 最大故障数 |
| 256 | 659(100%) | 1,443 | 45 |

| 圧縮 ビット幅 [bit] | SHA-512 | | |
|---------------------|--------------|-------|----------------|
| | 診断可能な 故障数 | クラス数 | クラス内の 最大故障数 |
| 512 | 659(100%) | 1,443 | 45 |

故障診断能力の評価結果

- ハッシュ関数とXORゲートを併用する診断用署名生成器
 - ・ XORゲートのみを使用した場合より高く,
ハッシュ関数のみを利用した場合とほぼ同等の故障診断能力

| 圧縮 ビット幅 [bit] | SHA-256&XOR_1 | | | SHA-512&XOR_1 | | |
|---------------------|---------------|-------|----------------|---------------|-------|----------------|
| | 診断可能な 故障数 | クラス数 | クラス内の 最大故障数 | 診断可能な 故障数 | クラス数 | クラス内の 最大故障数 |
| 256 | - | - | - | 659(100%) | 1,443 | 45 |
| 128 | 659(100%) | 1,443 | 45 | 659(100%) | 1,443 | 45 |
| 64 | 659(100%) | 1,443 | 45 | 659(100%) | 1,443 | 45 |
| 32 | 659(100%) | 1,443 | 45 | 659(100%) | 1,443 | 45 |
| 16 | 640(97.12%) | 1,224 | 45 | 644(97.72%) | 1,436 | 45 |

被疑故障署名を格納するメモリサイズの試算結果

- メモリ量[bit]=圧縮ビット幅*(診断可能な故障数+クラス数)

| 圧縮器の種類 | 圧縮ビット幅 | 診断可能な故障数 | クラス数 | メモリ量 |
|-----------------------|---------|-----------------|-------|--|
| XOR_1 | 16[bit] | 571 (86.65%) | 1,182 | $16 * (571 + 1,182)[\text{bit}]$ $= 28,048 [\text{bit}]$ $\approx 3.42[\text{kB}]$ |
| SHA-512 & XOR_1 | | 644 (97.72%) | 1,436 | $16 * (644 + 1,436)[\text{bit}]$ $= 33,280 [\text{bit}]$ $\approx 4.06[\text{kB}]$ |

まとめ・今後の課題

●まとめ

- 組込み自己診断機構を提案
 - ✓ 診断用署名生成器
 - ✓ 診断用署名に基づく故障候補の指摘器
- 組込み自己診断機構の予備評価実験
 - ✓ パス/フェイル情報の圧縮と故障診断能力を評価
 - ✓ 被疑故障署名を格納するためのメモリサイズを試算

●今後の課題

- 実チップ上に実装するための詳細なハードウェア設計

BISDについての先行研究[4](3)

- $S_{ref}(f)$ と S_{obs} から評価値を求める
 - σ^f : $S_{ref}(f)$ で不良で S_{obs} でも不良なビットの個数
 - ι^f : $S_{ref}(f)$ では不良だが S_{obs} では正常なビットの個数
 - τ^f : $S_{ref}(f)$ では正常だが S_{obs} では不良なビットの個数
- 以下のように被疑故障の順位を並び替える

$$r(f) > r(f') \Leftrightarrow \begin{cases} \sigma^f > \sigma^{f'} \text{ or} \\ \sigma^f = \sigma^{f'} \wedge \iota^f > \iota^{f'} \text{ or} \\ \sigma^f = \sigma^{f'} \wedge \iota^f = \iota^{f'} \wedge \tau^f > \tau^{f'} \end{cases}$$

BISDについての先行研究4

- 文献[4]で提案されたBISD機構の制約
 1. 決定的テストパターンをオンチップで格納するメモリ(Pattern Memory)が必要
 2. パス/フェイル情報を生成するために正常値を格納するメモリ(Response Memory)が必要
 3. 誤り情報用メモリ(Fail Memory)に格納されたパス/フェイル情報を外部テスト等で解析する必要

診断用署名生成器

2. ハッシュ関数を利用したアルゴリズムを実装する回路

- 実装する際の回路オーバーヘッドの問題

- ✓ BISD機構が組み込まれるチップには
セキュリティ関連のシステムのために
ハッシュ関数が実装される



- ✓ 既存の回路を利用できると仮定