Reading Assignment

- This lecture: 4.5, 4.6
- Next lecture: 5.1, 5.2
The Single Stuck-Fault Model

Structural Equivalence

Checkpoints

The Multiple Stuck-Fault Model
The Single Stuck-Fault (SSF) Model

- AKA the classical or standard fault model.
- SSF represents many different physical faults.
- SSF is independent of technology, and can be applied to any structural model.
- Tests that detect SSFs detect many non-classical faults.
- The number of SSFs in a circuit is comparably small, and can be reduced by fault-collapsing techniques.
- SSFs can be used to model other type of faults.
The Single Stuck-Fault (SSF) Model

- AKA the classical or standard fault model.
- SSF represents many different physical faults.
  - SSF is independent of technology, and can be applied to any structural model.
- Tests that detect SSFs detect many non-classical faults.
- The number of SSFs in a circuit is comparably small, and can be reduced by fault-collapsing techniques.
- SSFs can be used to model other type of faults.
The Single Stuck-Fault (SSF) Model

- AKA the classical or standard fault model.
- SSF represents many different physical faults.
- SSF is independent of technology, and can be applied to any structural model.
  - Tests that detect SSFs detect many non-classical faults.
  - The number of SSFs in a circuit is comparably small, and can be reduced by fault-collapsing techniques.
  - SSFs can be used to model other type of faults.
The Single Stuck-Fault (SSF) Model

- AKA the classical or standard fault model.
- SSF represents many different physical faults.
- SSF is independent of technology, and can be applied to any structural model.
- Tests that detect SSFs detect many non-classical faults.
- The number of SSFs in a circuit is comparably small, and can be reduced by fault-collapsing techniques.
- SSFs can be used to model other type of faults.
The Single Stuck-Fault (SSF) Model

- AKA the classical or standard fault model.
- SSF represents many different physical faults.
- SSF is independent of technology, and can be applied to any structural model.
- Tests that detect SSFs detect many non-classical faults.
- The number of SSFs in a circuit is comparably small, and can be reduced by fault-collapsing techniques.
- SSFs can be used to model other type of faults.
The Single Stuck-Fault (SSF) Model

- AKA the classical or standard fault model.
- SSF represents many different physical faults.
- SSF is independent of technology, and can be applied to any structural model.
- Tests that detect SSFs detect many non-classical faults.
- The number of SSFs in a circuit is comparably small, and can be reduced by fault-collapsing techniques.
- SSFs can be used to model other type of faults.
Fault Universe of SSF

- SSF is an explicit fault model.
  - Faults are individually identified and can be explicitly enumerated – their number matters.
- For $n$ lines, there are $2^n$ possible SSFs.
- In a gate-level model, consider a signal source $i$ (gate output or primary input) with $f_i$ fanouts.
  - If $f_i = 1$, then $i$ contributes 1 line.
  - Otherwise, $i$ contributes $f_i + 1$ line.
- We have $n = (G + I)(f + 1 - q)$.
  - $G$ and $I$: number of gates and primary inputs.
  - $f$: average fanout count.
  - $q$: fraction of signal sources with one fanout ($f_i = 1$).
  - $n \approx Gf$ as usually $G \gg I$ and $q > 0.5$. 
Fault Universe of SSF

- SSF is an explicit fault model.
  - Faults are individually identified and can be explicitly enumerated – their number matters.
- For $n$ lines, there are $2^n$ possible SSFs.
- In a gate-level model, consider a signal source $i$ (gate output or primary input) with $f_i$ fanouts.
  - If $f_i = 1$, then $i$ contributes 1 line.
  - Otherwise, $i$ contributes $f_i + 1$ line.
- We have $n = (G + I)(f + 1 - q)$.
  - $G$ and $I$: number of gates and primary inputs.
  - $f$: average fanout count.
  - $q$: fraction of signal sources with one fanout ($f_i = 1$).
  - $n \approx Gf$ as usually $G \gg I$ and $q > 0.5$. 
Fault Universe of SSF

- SSF is an explicit fault model.
  - Faults are individually identified and can be explicitly enumerated – their number matters.
- For $n$ lines, there are $2^n$ possible SSFs.
- In a gate-level model, consider a signal source $i$ (gate output or primary input) with $f_i$ fanouts.
  - If $f_i = 1$, then $i$ contributes 1 line.
  - Otherwise, $i$ contributes $f_i + 1$ line.
- We have $n = (G + l)(f + 1 - q)$.
  - $G$ and $l$: number of gates and primary inputs.
  - $f$: average fanout count.
  - $q$: fraction of signal sources with one fanout ($f_i = 1$).
  - $n \approx Gf$ as usually $G >> l$ and $q > 0.5$. 
Fault Universe of SSF

- SSF is an explicit fault model.
  - Faults are individually identified and can be explicitly enumerated – their number matters.
- For \( n \) lines, there are \( 2^n \) possible SSFs.
- In a gate-level model, consider a signal source \( i \) (gate output or primary input) with \( f_i \) fanouts.
  - If \( f_i = 1 \), then \( i \) contributes 1 line.
  - Otherwise, \( i \) contributes \( f_i + 1 \) line.
- We have \( n = (G + I)(f + 1 - q) \).
  - \( G \) and \( I \): number of gates and primary inputs.
  - \( f \): average fanout count.
  - \( q \): fraction of signal sources with one fanout (\( f_i = 1 \)).
  - \( n \approx Gf \) as usually \( G >> I \) and \( q > 0.5 \).
Outline

The Single Stuck-Fault Model

Structural Equivalence

Checkpoints

The Multiple Stuck-Fault Model
In theory, equivalence and dominance relations can be exploited to reduce the SSFs that need to be explicitly analyzed.

In practice, to determine such relations among two arbitrary faults are computationally prohibitive.

- As hard as SAT (NPC).

Structural equivalence: we only consider faults that are structurally related.

- e.g. as seen in the previous lecture for inputs/outputs of AND/NAND/OR/NOR gates.
- Equivalence can be obtained via local analysis, which is computationally feasible.
In theory, equivalence and dominance relations can be exploited to reduce the SSFs that need to be explicitly analyzed.

In practice, to determine such relations among two arbitrary faults are computationally prohibitive.

- As hard as SAT (NPC).

Structural equivalence: we only consider faults that are structurally related.

- e.g. as seen in the previous lecture for inputs/outputs of AND/NAND/OR/NOR gates.
- Equivalence can be obtained via local analysis, which is computationally feasible.
Structural Equivalence

▶ In theory, equivalence and dominance relations can be exploited to reduce the SSFs that need to be explicitly analyzed.
▶ In practice, to determine such relations among two arbitrary faults are computationally prohibitive.
  ▶ As hard as SAT (NPC).
▶ Structural equivalence: we only consider faults that are structurally related.
  ▶ e.g. as seen in the previous lecture for inputs/outputs of AND/NAND/OR/NOR gates.
  ▶ Equivalence can be obtained via local analysis, which is computationally feasible.
Define $S(N_f)$ be a simplification of $N$ with the fault $f$.

- Propagate all constants due to $f$.
- Remove all gates not eventually driving a primary output.

Two faults $f$ and $g$ are structurally equivalent iff $S(N_f)$ and $S(N_g)$ are identical.

It is much easier to prove two circuits are identical than to prove they are functionally equivalent.
Structural Equivalence Example

Define $S(N_f)$ be a simplification of $N$ with the fault $f$.

- Propagate all constants due to $f$.
- Remove all gates not eventually driving a primary output.

Two faults $f$ and $g$ are structurally equivalent iff $S(N_f)$ and $S(N_g)$ are identical.

- It is much easier to prove two circuits are identical than to prove they are functionally equivalent.

Figure 4.18 Illustration for structural fault equivalence

(Abramovici et al., 1990)
Simple Rules for Structural Equivalence

(Fig. 4.19, Abramovici et al., 1990)

- Introduce 2 faults on every signal source and destination.
- Simple rules:
  - For each gate, if an input s-a-c controls the output as $c \oplus i$, then the input s-a-c is equivalent to the output s-a-c $\oplus i$.
  - For each signal with one fanout, the fault at source is equivalent to the corresponding fault at destination.
  - A fault at signal source for a signal with multiple fanouts is structurally equivalent to a multiple fault at all destinations and thus CANNOT be used for our analysis.
Simple Rules for Structural Equivalence

- Introduce 2 faults on every signal source and destination.
- Simple rules:
  - For each gate, if an input s-a-c controls the output as $c \oplus i$, then the input s-a-c is equivalent to the output s-a-c $\oplus i$.
  - For each signal with one fanout, the fault at source is equivalent to the corresponding fault at destination.
  - A fault at signal source for a signal with multiple fanouts is structurally equivalent to a multiple fault at all destinations and thus CANNOT be used for our analysis.
For simple rules, no fault is structurally equivalent to $f \ s-a-0$.

- A three-valued logic simulation (with $u$) indicates 0 at $b$ will result in 0 at $f$.
  - So $f \ s-a-0$ is structurally equivalent to $b \ s-a-0$.

In practice, the gain of such complex rules may be small and does not justify their cost.
Complex Rules for Structural Equivalence

(Fig. 4.20, Abramovici et al., 1990)

▶ For simple rules, no fault is structurally equivalent to $f \ s-a-0$.
▶ A three-valued logic simulation (with $u$) indicates 0 at $b$ will result in 0 at $f$.
   ▶ So $f \ s-a-0$ is structurally equivalent to $b \ s-a-0$.
▶ In practice, the gain of such complex rules may be small and does not justify their cost.
For simple rules, no fault is structurally equivalent to $f$ s-a-0.

A three-valued logic simulation (with $u$) indicates 0 at $b$ will result in 0 at $f$.

- So $f$ s-a-0 is structurally equivalent to $b$ s-a-0.

In practice, the gain of such complex rules may be small and does not justify their cost.
Recall for a gate-level model, there are $2n = 2(G + I)(f + 1 - q)$ SSFs.

- $G$ and $I$: number of gates and primary inputs.
- $f$: average fanout count.
- $q$: fraction of signal sources with one fanout ($f_i = 1$).

We may eliminate $m = G(g + p)$ SSFs by structural equivalence.

- Assume all gates with more than one input are among AND/NAND/OR/NOR so we can eliminate half of the faults at gate inputs for them.
- $g$: average fanin count.
- $p$: fraction of gates with only one input where we can eliminate both faults at gate inputs.

- $g \approx f$, half SSFs are eliminated.
- Can you do better?
Recall for a gate-level model, there are 
\[ 2n = 2(G + I)(f + 1 - q) \] SSFs.

- \( G \) and \( I \): number of gates and primary inputs.
- \( f \): average fanout count.
- \( q \): fraction of signal sources with one fanout \((f_i = 1)\).

We may eliminate \( m = G(g + p) \) SSFs by structural equivalence.

- Assume all gates with more than one input are among AND/NAND/OR/NOR so we can eliminate half of the faults at gate inputs for them.
- \( g \): average fanin count.
- \( p \): fraction of gates with only one input where we can eliminate both faults at gate inputs.

- \( g \approx f \), half SSFs are eliminated.
- Can you do better?
Fault Collapsing via Structural Equivalence

- Recall for a gate-level model, there are $2n = 2(G + I)(f + 1 - q)$ SSFs.
  - $G$ and $I$: number of gates and primary inputs.
  - $f$: average fanout count.
  - $q$: fraction of signal sources with one fanout ($f_i = 1$).
- We may eliminate $m = G(g + p)$ SSFs by structural equivalence.
  - Assume all gates with more than one input are among AND/NAND/OR/NOR so we can eliminate half of the faults at gate inputs for them.
  - $g$: average fanin count.
  - $p$: fraction of gates with only one input where we can eliminate both faults at gate inputs.
- $g \approx f$, half SSFs are eliminated.
- Can you do better?
Recall for a gate-level model, there are
\[ 2n = 2(G + I)(f + 1 - q) \] SSFs.
- \( G \) and \( I \): number of gates and primary inputs.
- \( f \): average fanout count.
- \( q \): fraction of signal sources with one fanout \((f_i = 1)\).

We may eliminate \( m = G(g + p) \) SSFs by structural equivalence.
- Assume all gates with more than one input are among AND/NAND/OR/NOR so we can eliminate half of the faults at gate inputs for them.
- \( g \): average fanin count.
- \( p \): fraction of gates with only one input where we can eliminate both faults at gate inputs.

\[ g \approx f, \] half SSFs are eliminated.

Can you do better?
Outline

The Single Stuck-Fault Model

Structural Equivalence

Checkpoints

The Multiple Stuck-Fault Model
We may extend the idea of structural equivalence to discover dominance relations structurally.

For each gate, if an input s-a-c controls the output as $c \oplus i$, then the output s-a-c $\oplus i$ dominates the input s-a-$\overline{c}$.

It helps to eliminate the faults not eliminated by the structural equivalence if fault location is not of concern.
We may extend the idea of structural equivalence to discover dominance relations structurally.

For each gate, if an input \( s-a-c \) controls the output as \( c \oplus i \), then the output \( s-a-c \oplus i \) dominates the input \( s-a-\bar{c} \).

It helps to eliminate the faults not eliminated by the structural equivalence if fault location is not of concern.
We may extend the idea of structural equivalence to discover dominance relations structurally.

For each gate, if an input $s-a-c$ controls the output as $c \oplus i$, then the output $s-a-c \oplus i$ dominates the input $s-a-\overline{c}$.

It helps to eliminate the faults not eliminated by the structural equivalence if fault location is not of concern.
Checkpoints

Checkpoints: primary inputs and the fanout branches.
- $a, b, c, d, e, g, h$

A test set that detecting all SSFs on the checkpoints of a circuit detects all SSFs in it.

Faults can be further eliminated via structural equivalence.
- Collapsing due to equivalent checkpoint SSFs: $a$ s-a-0 and $b$ s-a-0, $d$ s-a-0 and $h$ s-a-0.
- Collapsing due to equivalent non-checkpoint SSFs and dominance: $g$ s-a-1 and $f$ s-a-1, $e$ s-a-1 and $i$ s-a-1.
Checkpoints

- Checkpoints: primary inputs and the fanout branches.
  - $a, b, c, d, e, g, h$.

- A test set that detecting all SSFs on the checkpoints of a circuit detects all SSFs in it.

- Faults can be further eliminated via structural equivalence.
  - Collapsing due to equivalent checkpoint SSFs: $a$ s-a-0 and $b$ s-a-0, $d$ s-a-0 and $h$ s-a-0.
  - Collapsing due to equivalent non-checkpoint SSFs and dominance: $g$ s-a-1 and $f$ s-a-1, $e$ s-a-1 and $i$ s-a-1.

(Abramovici et al., 1990)
Checkpoints

- Checkpoints: primary inputs and the fanout branches.
  - $a, b, c, d, e, g, h$.
- A test set that detecting all SSFs on the checkpoints of a circuit detects all SSFs in it.
- Faults can be further eliminated via structural equivalence.
  - Collapsing due to equivalent checkpoint SSFs: $a$ s-a-0 and $b$ s-a-0, $d$ s-a-0 and $h$ s-a-0.
  - Collapsing due to equivalent non-checkpoint SSFs and dominance: $g$ s-a-1 and $f$ s-a-1. $e$ s-a-1 and $i$ s-a-1.

Figure 4.21
(Abramovici et al., 1990)
Discussions

- There are $r = I + (G + I)(f - q)$ checkpoints.
  - $G$ and $I$: number of gates and primary inputs.
  - $f$: average fanout count.
  - $q$: fraction of signal sources with one fanout ($f_i = 1$).
- $2r \approx 2(G + I)(f - q)$.
  - About 40% reduction comparing to $2n = 2(G + I)(f + 1 - q)$ for typical $f = 2.5$ and $q = 0.7$.
  - Possible further reduction via structural equivalence.
- Overall algorithm complexity considering checkpoints and structural equivalence.
  - Space: $O(n)$.
  - Time: $O(n)$. 
There are $r = I + (G + I)(f - q)$ checkpoints.
- $G$ and $I$: number of gates and primary inputs.
- $f$: average fanout count.
- $q$: fraction of signal sources with one fanout ($f_i = 1$).

$2r \approx 2(G + I)(f - q)$.
- About 40% reduction comparing to $2n = 2(G + I)(f + 1 - q)$ for typical $f = 2.5$ and $q = 0.7$.
- Possible further reduction via structural equivalence.

Overall algorithm complexity considering checkpoints and structural equivalence.
- Space: $O(n)$.
- Time: $O(n)$. 
There are \( r = I + (G + I)(f - q) \) checkpoints.
- \( G \) and \( I \): number of gates and primary inputs.
- \( f \): average fanout count.
- \( q \): fraction of signal sources with one fanout \((f_i = 1)\).

\[ 2r \approx 2(G + I)(f - q). \]
- About 40% reduction comparing to \( 2n = 2(G + I)(f + 1 - q) \) for typical \( f = 2.5 \) and \( q = 0.7 \).
- Possible further reduction via structural equivalence.

Overall algorithm complexity considering checkpoints and structural equivalence.
- Space: \( O(n) \).
- Time: \( O(n) \).
Outline

The Single Stuck-Fault Model

Structural Equivalence

Checkpoints

The Multiple Stuck-Fault Model
The Multiple Stuck-Fault (MSF) Model

- For \( n \) possible SSF sites, there are totally \( 3^n - 1 \) MSFs.
  - 3 possibilities per site.
  - Include all SSFs but not the fault-free circuit.
- There are \( \sum_{i=1}^{k} \binom{n}{i} 2^i \) MSFs with at most \( k \) sticks.
  - Usually too large to allow us to deal explicitly.
The Multiple Stuck-Fault (MSF) Model

- For $n$ possible SSF sites, there are totally $3^n - 1$ MSFs.
  - 3 possibilities per site.
  - Include all SSFs but not the fault-free circuit.
- There are $\sum_{i=1}^{k} \binom{n}{i} 2^i$ MSFs with at most $k$ stuck.
  - Usually too large to allow us to deal explicitly.
A fault $f$ *functionally masks* the fault $g$ iff the multiple fault \{f, g\} cannot be detected by $T_g$.

For example, a s-a-1 masks c s-a-0.

- 011 is the only test detecting c s-a-1.
- But it won't detect \{c s-a-0, a s-a-1\}.

Tests detecting SSFs may omit MSFs, but usually can cover most of them in practice.
Masking Relations

A fault $f$ functionally masks the fault $g$ iff the multiple fault $\{f, g\}$ cannot be detected by $T_g$.

For example, $a$ s-a-1 masks $c$ s-a-0.

- 011 is the only test detecting $c$ s-a-1.
- But it won't detect $\{c$ s-a-0, $a$ s-a-1\}.

Tests detecting SSFs may omit MSFs, but usually can cover most of them in practice.
A fault \( f \) \textit{functionally masks} the fault \( g \) iff the multiple fault \( \{f, g\} \) cannot be detected by \( T_g \).

For example, \( a \text{-s-a-1} \) masks \( c \text{-s-a-0} \).

- 011 is the only test detecting \( c \text{-s-a-1} \).
- But it won't detect \( \{c \text{-s-a-0}, a \text{-s-a-1}\} \).

Tests detecting SSFs may omit MSFs, but usually can cover most of them in practice.
Summary

▶ Structural properties can be exploited to quickly eliminate the majority of SSFs.
▶ Masking relations may prevent SSF tests to detect MSFs, though in practice, SSF tests are usually suffice.