Simulazione di guasto

M. Favalli

Engineering Department in Ferrara

Introduzione

Simulazione di guasto

- Simulazione nei circuiti digitali
- Problemi e applicazioni
- Algoritmi di simulazione di guasto
  - Seriale
  - Parallelo (fault parallel, test parallel)
  - Deduttivo
  - Concorrente

Simulazione dei circuiti digitali

- La simulazione dei circuiti digitali puó avvenire a diversi livelli:
  - fisico
  - elettrico
  - logico
- Nel caso del collaudo i livelli fisico ed elettrico sono utilizzati esclusivamente per la validazione e analisi dei modelli di guasto
- Al livello logico la simulazione puó avvenire ai seguenti livelli
  - switch
  - gate
  - RTL
  - behavioral

Simulazione di guasto

- Dati
  - un circuito
  - una sequenza di vettori di test
  - un insieme di modelli di guasto
- Determina:
  - la copertura di guasto (frazione di guasti modellati rivelati come errori dai test applicati)
  - l'insieme dei guasti non rivelati
- Applicazioni:
  - stima della qualitá del collaudo (defect level)
  - miglioramento delle sequenze di collaudo e del progetto stesso
  - aiuto computazionale alla test generation

Utilizzo della simulazione di guasto per la stima dell'affidabilitá

- La simulazione di guasto consente di stimare l'affidabilitá dei circuiti digitali
- Vengono costruite copie guaste del circuito secondo una qualche distribuzione di probabilitá dei guasti stessi
- Il circuito viene simulato con una sequenza di vettori di ingresso rappresentativa del normale carico di lavoro
- Viene valutata la probabilitá di errore sulle uscite
- Il target é dato dai guasti che possono manifestarsi durante il normale funzionamento del circuito (stuck-at, guasti transitori)
- Elevato interesse industriale (standard in campo aeronautico, automobilistico)

Simulazione di guasto al livello logico

- Modello del circuito
  - mixed level: gate e switch per parti custom del circuito
  - modelli behavioral con guasti su ingressi e uscite
- Modello dei segnali
  - $\{0, 1\}$ per reti combinatorie, $\{0, 1, X\}$ per reti sequenziali e $\{0, 1, X, Z\}$ per reti con componenti ad alta impedenza
  - modelli con considerazioni analogiche (transistor stuck-on e bridging)
- Modello delle temporizzazioni
  - zero delay per reti combinatorie
  - unit delay o cycle accurate per reti sincrone
  - propagation delay per reti asincrone

Simulazione per diversi modelli di guasto

- Stuck-at con fault collapsing
- Transistor stuck-on e open
- Bridging
- Transition faults, gate delay faults e path delay faults
- Fault dropping: un guasto rivelato non viene ulteriormente simulato
- Fault sampling: si simula solo un campione di guasti

Algoritmi di simulazione di guasto

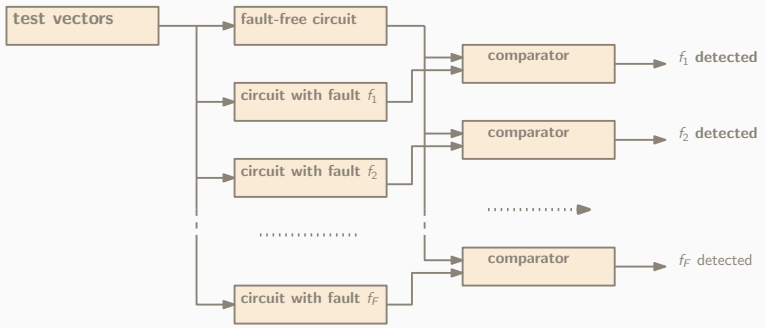
- Li vedremo nel caso piú semplice, ovvero per i guasti di tipo stuck-at
  - Algoritmo seriale
  - Algoritmo parallelo
  - Algoritmo deduttivo
  - Algoritmo concorrente
- Complessitá computazionale $O(N \times F \times T)$ dove N é il numero di gate, F é il numero di guasti e T é il numero di test
- In una rete combinatoria si puó assumere che F e T siano proporzionali a $N \Rightarrow O(N^3)$

Algoritmo seriale

- Si simula il circuito corretto e si memorizzano le risposte
- Per ogni guasto nell'insieme considerato l'algoritmo
  - modifica la netlist iniettando un guasto
  - simula la rete guasta confrontando le risposte con quelle corrette
  - se differiscono dichiara il guasto come rivelato e passa al successivo (fault dropping)
- Vantaggi:
  - semplicitá di implementazione in qualsiasi tipo di simulatore
  - applicabile a diversi modelli di guasto
  - ridotta occupazione di memoria (sequenziali)

Algoritmo seriale

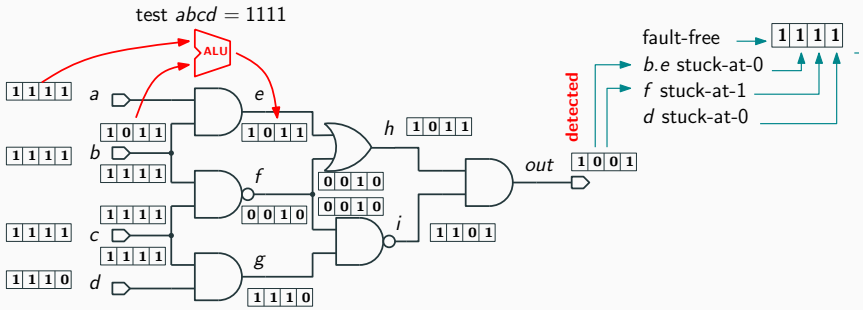
- Lo svantaggio principale é dato dal fatto che vengono ripetute diverse operazioni inutilmente
- Tempi non compatibili con circuiti di grandi dimensioni
- Come alternativa é stata utilizzata la possibilitá di simulare diversi circuiti guasti in maniera concorrente



Simulazione di guasto parallela (nei guasti)

- Sfrutta il parallelismo delle operazioni della ALU rispetto ai bit di una parola di CPU ($w = 64\,\text{bit}$) o di GPU ($> 64bit$)
- Simulazione di guasti in parallelo
  - una parola per ogni segnale
  - il primo bit di una parola rappresenta un segnale in assenza di guasti e rimanenti $w - 1$ bit rappresentano quel segnale in presenza di $w - 1$ guasti diversi
  - a ciascun passo si simulano la macchina prima di guasti e $w - 1$ macchine guaste
  - si puó effettura il fault dropping solo quando tutti i guasti del gruppo corrente sono stati simulati $\Rightarrow$ speed-up $\simeq w - 1$
  - non puó tenere conto dei ritardi di propagazione
- Simulazione di test in parallelo (solo reti combinatorie)

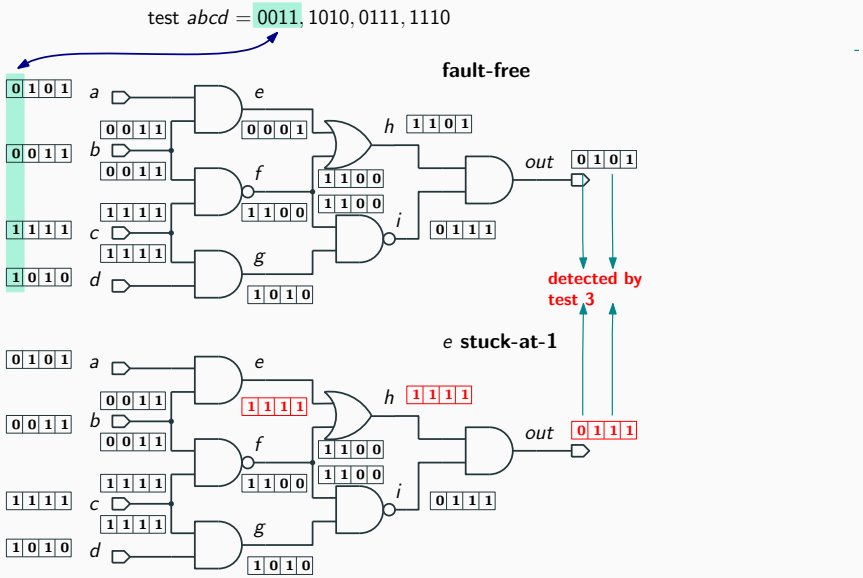
Esempio



Simulazione di guasto parallela (nei test)

- Limitata ai circuiti combinatori
- Si simula il circuito in assenza di guasti per w vettori di test
- Per ciascun guasto si simula il circuito per gli stessi vettori di test
- Ciascun bit di una parola associata a un segnale rappresenta il valore di tale segnale per un certo vettore di ingresso

Esempio



Simulazione deduttiva

- Per ogni vettore viene simulato il circuito fault-free
- A ciascuna linea j del circuito viene associata la lista dei guasti attivi su j
- Dopo la simulazione fault-free di ciascun vettore, le liste di guasti delle uscite dei gate sono calcolate utilizzando semplici operazioni sugli insiemi a partire dai valori dei segnali (nel circuito corretto) e le liste di guasti agli ingressi dei gate
- Le liste dei guasti ai PO forniscono i dati sulla fault detection
- Limitazioni:
  - regole sugli insiemi non utilizzabili per gate non elementari
  - difficile da usare per i ritardi

Simulazione deduttiva

Calcolo delle liste di uscita per un gate di tipo AND

a	b	output list
0	0	$L_a \cap L_b$
0	1	$L_a \setminus L_b$
1	0	$L_b \setminus L_a$
1	1	$L_a \cup L_b$

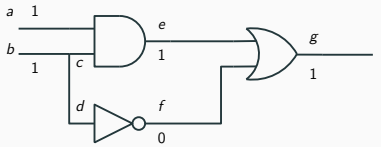
Calcolo delle liste di uscita per un gate EXOR

a	b	output list
-	-	$L_a \cup L_b \setminus L_a \cap L_b$

Esempio

L_k denota la lista di guasti del segnale k, mentre $k_{0/1}$ denota il guasto k stuck-at-0/1

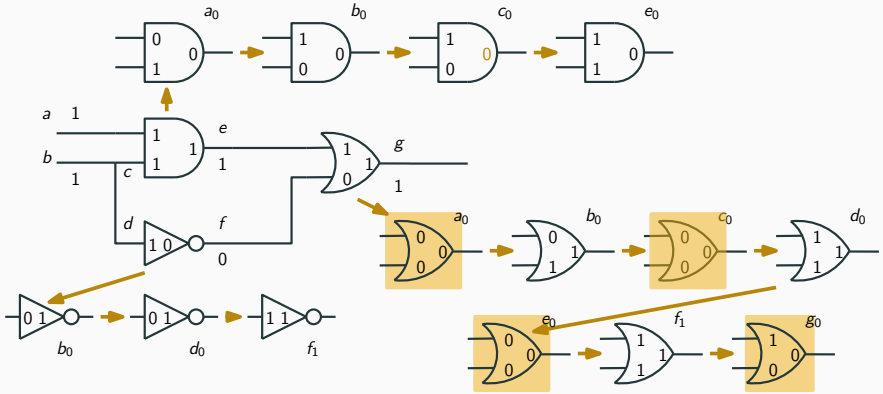
signal	fault list
a	$L_a = \{a_0\}$
b	$L_b = \{b_0\}$
c	$L_c = \{b_0, c_0\}$
d	$L_d = \{b_0, d_0\}$
e	$L_e = L_a \cup L_c \cup \{e_0\}$
	$= \{a_0, b_0, c_0, e_0\}$
f	$L_f = L_d \cup \{f_1\} = \{b_0, d_0, f_1\}$
g	$L_g = (L_e \setminus L_f) \cup \{g_0\} =$
	$= \{a_0, c_0, e_0, g_0\}$



Algoritmo concorrente

- Consistente con la simulazione event-driven e con qualsiasi modello di ritardo e componente da gate a behavioral
- Passi:
  1. simulazione event-driven del circuito fault-free
  2. per ciascun guasto vengono poi simulate quelle parti dei circuiti guasti che differiscono da quello privo di guasti
- Per ciascun gate si tiene una lista con lo stato del gate (ingressi e uscita) in assenza di guasti e con gli stati del gate nei circuiti guasti se tale stato gate é diverso da quello fault-free
- Tutti gli eventi dei circuiti guasti uguali a quelli fault-free sono simulati implicitamente
- Piú veloce degli altri metodi, ma usa troppa memoria

Esempio



Progetti

- Sono disponibili diversi progetti nell'ambito della simulazione di guasto
  - realizzazione di un simulatore di guasto parallelo utilizzando la GPU come accelleratore (CUDA)
  - simulazione di guasti transitori
  - ....