

# Progetti a.a. 2020-21

M. Favalli

Engineering Department in Ferrara

# Realizzazione di procedure per ABC

- ▶ ABC ha una buona interfaccia di programmazione
- ▶ Vedi programming notes in <https://people.eecs.berkeley.edu/~alanmi/abc/>, dove c'è anche un tutorial che guida alla realizzazione di nuovi comandi per ABC
- ▶ In questo ambito sono disponibili diversi progetti fra cui
  - ▶ scrittura dell'output in formato VHDL dataflow
  - ▶ realizzazione di un semplice simulatore logico
  - ▶ interfaccia di navigazione nella rete (path tracing)
- ▶ Alcuni di questi progetti possono essere fatti anche senza interfacciarsi ad ABC (in questo caso però bisogna realizzare un semplice parser per il formato BLIF)

## ATPG per circuiti sequenziali

- ▶ Realizzazione di un semplice ATPG per guasti di tipo stuck-at singolo in circuiti sequenziali
- ▶ L'ATPG sarebbe basato su time frame expansion (statica) e l'utilizzo di Boolean Satisfiability
- ▶ Lo strumento deve contenere anche un simulatore di guasto
- ▶ Il formato di ingresso é quello blif

## Cell-aware transition faults

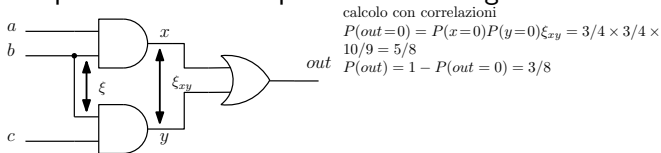
- ▶ Le celle logiche basate sui transistori FinFET possono essere piú complesse di quelle basate sui MOSFET
- ▶ I guasti contenuti al loro interno sono difficilmente rappresentabili dagli stuck-at
- ▶ Si possono utilizzare modelli di guasto che per ogni guasto interno mettono a disposizione l'elenco di condizioni che lo attivano producendo un errore sull'uscita
  - ▶ tabella con i vettori di test come righe e i guasti come colonne
- ▶ Si può fare un ATPG basato su SAT per questo tipo di guasti nel caso in cui il guasto non sia attivato da un singolo vettore, ma necessiti di un'inizializzazione

# Probabilità dei segnali

- ▶ Numerose applicazioni necessitano di stime rapide della probabilità che i segnali in circuito digitale assumano il valore 1 o il valore 0:
  - ▶ stima del consumo di potenza (probabilità di transizione dei segnali);
  - ▶ stime di affidabilità (probabilità di errore);
  - ▶ collaudo (misure di collaudabilità).
- ▶ In una rete ad albero questa operazione può essere svolta in maniera molto veloce (COP) In una rete con riconvergenze questo approccio non è esatto.
- ▶ In alternativa è stato sviluppato un metodo approssimato che tiene conto delle correlazioni fra i segnali.

## Probabilità dei segnali

- ▶ Si supponga di voler calcolare  $p(y) = p(a \wedge b) = p(a)p(b|a)$  che é diverso da  $p(a)p(b)$
- ▶ Si può però scrivere  $p(y) = p(a)p(b)c_{ab}$  ove  $c_{ab}$  rappresenta la correlazione fra  $a$  e  $b$
- ▶ Regole per propagare tali coefficienti consentendo così una stima piú accurata delle probabilità dei segnali.



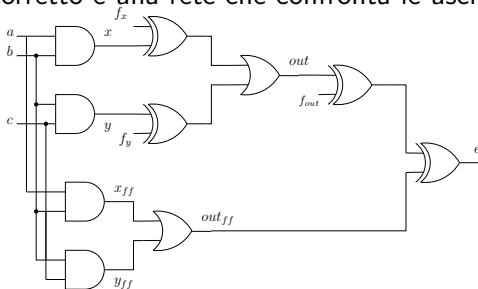
- ▶ Il programma riceve in ingresso
  - ▶ la rete combinatoria descritta utilizzando un blif
  - ▶ le probabilità degli ingressi
  - ▶ Produce in uscita la probabilità di ogni segnale nel circuito.

## Calcolo della probabilità di errore

- ▶ Le tecnologie che forse sostituiranno quelle attuali presentano livelli ridotti di affidabilità
- ▶ I gate presentano una probabilità di produrre un uscita errata
- ▶ Nel caso di una rete combinatoria, si vuole conoscere la probabilità di errore media in uscita al circuito

## Calcolo della probabilità di errore

- ▶ Si aggiungono alla rete dei gate controllati da variabili addizionali che consentono di iniettare gli errori
- ▶ Poi si costruisce un miter che contiene anche una copia del circuito corretto e una rete che confronta le uscite



- ▶ Quindi data una probabilità per le variabili addizionali di valere 1 (che é uguale alla probabilità di errore dei gate), vogliamo conoscere la probabilità di errore, ovvero di avere  $e = 1$ .



# Calcolo della probabilità di errore

- ▶ Tecniche utilizzabili allo scopo:
  1. simulazione
  2. metodi probabilistici
  3. BDD
  4. metodi basati su Boolean SAT
- ▶ Il SAT solver dovrebbe essere applicato al circuito in figura per trovare tutte le possibili soluzioni per cui  $e = 1$
- ▶ Il compito é computazionalmente molto pesante e si ricorre quindi a tecniche di approximate model counting che trovano limiti inferiori per il numero di soluzioni