Random Testing

- Extensively used for both hardware and software
- Ideally each input is selected randomly. PR (Pseudorandom) schemes approximate random.
- Generally quite effective for moderate coverage.
  - Coverage hard to determine a priori.
  - Ineffective for random-pattern-resistant faults.
  - Coverage tools: Random (functional) followed by Structural testing.

Random Testing: Advantage

- No test generation using structural information needed.
- Test set-up using comparison:

  - Alternative: Is response reasonable? (software testing)
Pseudorandom (PR) Testing

- Unlike true random, reproducible.
- Will not repeat until all combinations applied.
- Generation: usually just-in-time (not stored).
  - Autonomous linear feedback shift register (ALFSR).
  - Cellular automata etc possible.
- Some randomness properties satisfied, but not all.

Coverage Achieved

- Coverage grows fast in the beginning, saturates near end.
- Is it described by $C(L)=1-e^{-aL}$?
  - No, doesn’t fit.
- It is controlled by distribution of detectability of faults.
- Detectability profile (Malaiya & Yang ’84):
  - $H=\{h_1, h_2, \ldots, h_N\}$
    - $N$: total possible vectors
    - $h_k$: number of faults detected by exactly $k$ vectors.
  - Total faults $M=\sum h_k$
  - $h_1$: number of least testable faults
Detectability Profiles: Ex

- CECL Full adder
  Inputs=4 \((N=16)\), \(M=90\)
  \(H=(h_1, h_2, h_3, h_4, h_5, h_6, h_8) = (1,11,2,43,21,4,8)\)

- Schneider’s counterexample:
  Inputs= 4 \((N=16)\), \(M=44\)
  \(H=(h_1, h_2, h_3, h_{14}) = (23,19,1,1)\)

Coverage Obtained by L Vectors

- For PR tests (McClusky 87)
  \[ C(L) = 1 - \sum_{k=1}^{N} \left(1 - \frac{k}{N}\right)^L \frac{C_k}{M} \]
  \[ = 1 - \sum_{k=1}^{N} \left(1 - \frac{k}{N}\right)^L \frac{1}{M} \]
  (for Random)

- For large \(L\), terms with only low \(k\) (i.e. faults that are hard to test) have an impact. Thus only lower elements of \(H\) need to be estimated.

- For CECL Full Adder,
  \[ C(15) = 1 - \{2.1 + 3.1 + 0.07 + 0.1 + 0.0005...\} \times 10^{-7} \]
Detectability Profile: Software

- Software detectability profile is exponential (Adam’s data, IBM).
- Justification: Early testing will find & remove easy-to-test faults.
- Testing methods need to focus on hard-to-find faults.