DIGITAL SYSTEM TESTING
COE -545

Lecture – 11

Random Test Generation & Signal Probabilities

Fault-Independent Test Generation
Random Test Generation

- **Random** Test Generation → **Does not Target Particular Fault**
- **Random Test Vectors Are Applied and One Hopes to Detect As Many Faults As Possible**

***Advantages*:**

- **Low Test Generation Cost**
- **Test Vectors Usually Generated On- The- Fly (*No Need for Test Vector Storage*)**

***Disadvantages*:**

- **Long Test Sequence (*Approx. 10 Times Longer* Than Deterministically Generated Tests)**
- **High Test Application Cost**

COE – KFUPM *Dr. Alaaeldin Amin (COE 545)* *Slide Number 2*

Fault-Independent Test Generation Random Test Generation

Total Test Cost = Cost of Test Generation + Cost of Test Application

Cost of Test Generation

- **Random** Test Generation of TVs → **Inexpensive**
- **Deterministic** Fault-Oriented Test Generation → **Expensive**
- For High Fault Coverage, A lot more Randomly generated TVs are Required

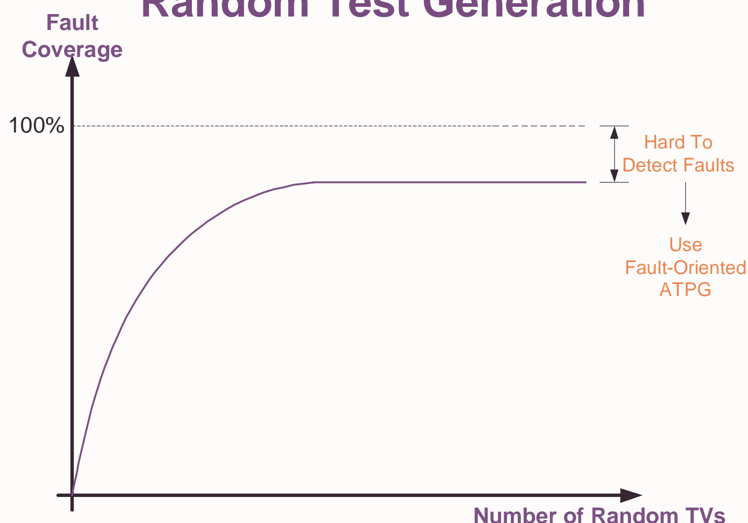
Cost of Test Application (*Test Time on the ATE*)

- Depends on the Number of TVs in the Test Set
- Random Test Generation is usually combined with deterministic Fault-Oriented testing in a hybrid approach

  1. Use **Random** testing to Detect "*easy*" faults
  2. Use **Deterministic** Fault-Oriented ATPG algorithm (PODEM, FAN, etc.) to Detect the residual "*hard*" faults

COE – KFUPM *Dr. Alaaeldin Amin (COE 545)* *Slide Number 3*

Fault-Independent Test Generation Random Test Generation



COE – KFUPM *Dr. Alaaeldin Amin (COE 545)* *Slide Number 4*

Random Test Generation

Quality Measures

Testing quality T_N: Probability that all detectable SSL faults are detected by N random vectors

N-Step Detection Probability ($d_N^{\,f}$) – the Probability to Detect Fault f after applying N TVs

Detection quality d_N : Probability that the *hardest-to-detect* SSL fault is detected by N random vectors.

$$d_N = \min d_N^{\,f} \text{ over all faults } \{f\}$$

- If T_f is the set of all faults that detect fault f in an n-input circuit, the probability that a random vector detects f is

$$d_f = |T_f| / 2^n$$

Dr. Alaaeldin Amin (COE 545) Slide Number 5

Random Test Generation

Quality Measures

Estimate of the number N_t of random tests needed to achieve testing quality of at least c [Savir and Bardell 1984]:

$$N_t = \left\lceil \frac{\ln(1-c) - \ln(k)}{\ln(1-d_{min})} \right\rceil$$

where k is the number of faults with detection probability between d_{min} and $2d_{min}$

- *Lower Bound on d_{min}* $\quad d_{min} \geq 1/2^{n_{max}}$
  - *where n_{max} = The Max number of PIs feeding a PO*

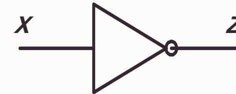
Dr. Alaaeldin Amin (COE 545) Slide Number 6

Signal Probabilities:

Let P_Z be the Probability of Signal Z to have a Logic 1 Value

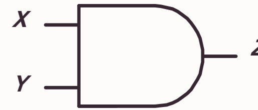
- For PI's $P_{PI} = 1/2$

Inverter : (Input x and Output Z)



$P_Z = 1 - P_x$

AND Gate: *Assuming x & Y are Independent* (Do Not Depend on Common PI's)

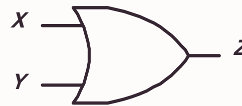


$$P_Z = P_x P_y$$

COE – KFUPM   Dr. Alaaeldin Amin (COE 545)   Slide Number 7

Signal Probabilities:

OR Gate: *Assuming x & Y are Independent* (Do Not Depend on Common PI's)



$$P_Z = P_x + P_y - P_x P_y$$

NAND Gate: $1 - P_Z = P_x P_y \rightarrow P_Z = 1 - P_x P_y$

NOR Gate:

$$1 - P_Z = P_x + P_y - P_x P_y \rightarrow P_Z = 1 - (P_x + P_y - P_x P_y)$$

COE – KFUPM   Dr. Alaaeldin Amin (COE 545)   Slide Number 8

Random Test Generation

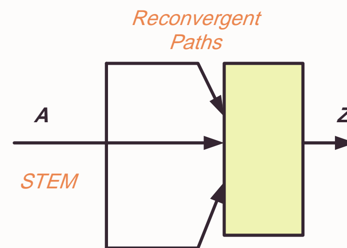
For a Stem A with Reconvergent Fanout:

$P_Z = (P_Z/A=0).\ P_{A=0} + (P_Z/A=1).\ P_A$

$= (P_Z/A=0).\ (1- P_A) + (P_Z/A=1).\ P_A$

To Compute $(P_Z/A=0)$ && $(P_Z/A=1)$ **in some Circuit N**

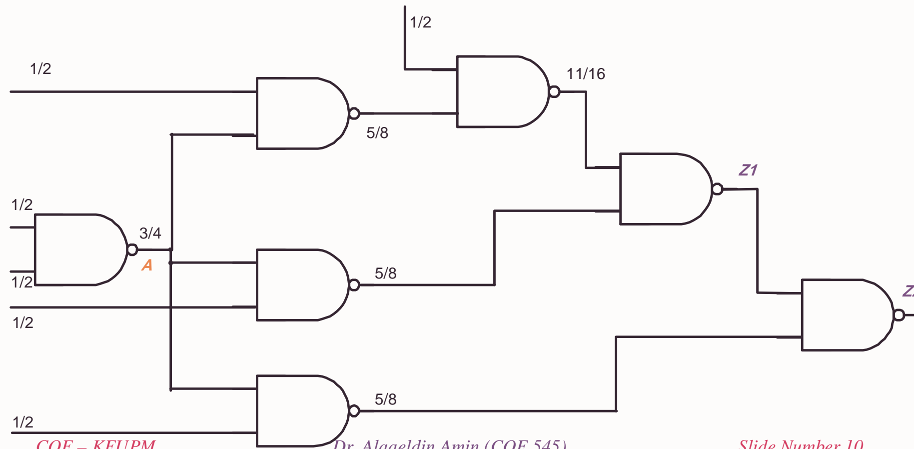
- Derive N^0 and N^1, where $N^0 = N(A=0)$ and $N^1 = N(A=1)$
- $(P_Z/A=0)$ is P_Z in N^0
- $(P_Z/A=1)$ is P_Z in N^1

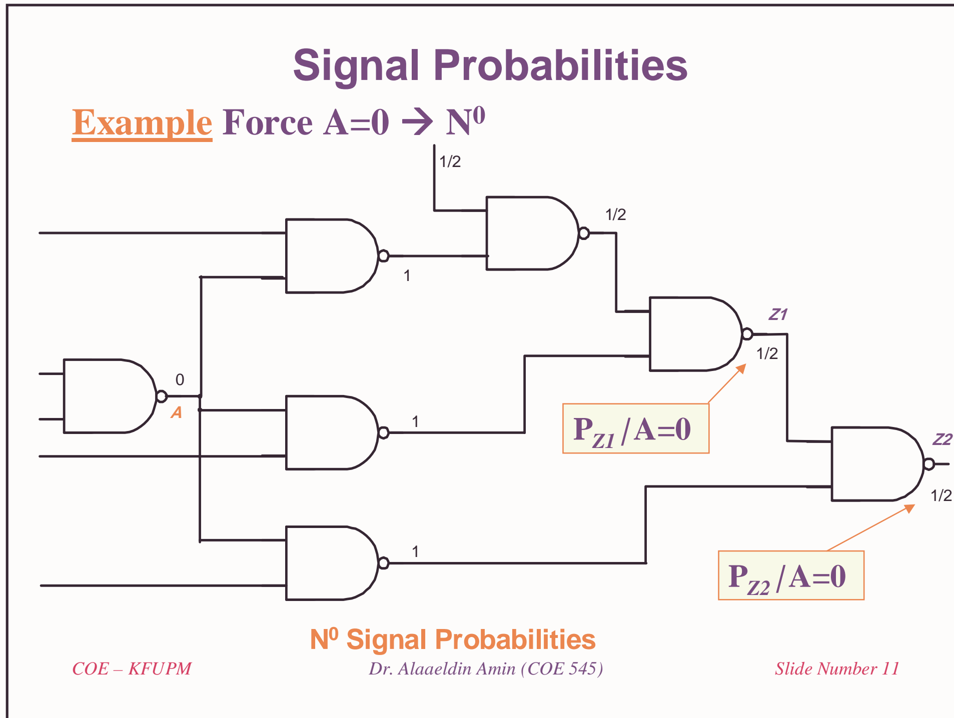


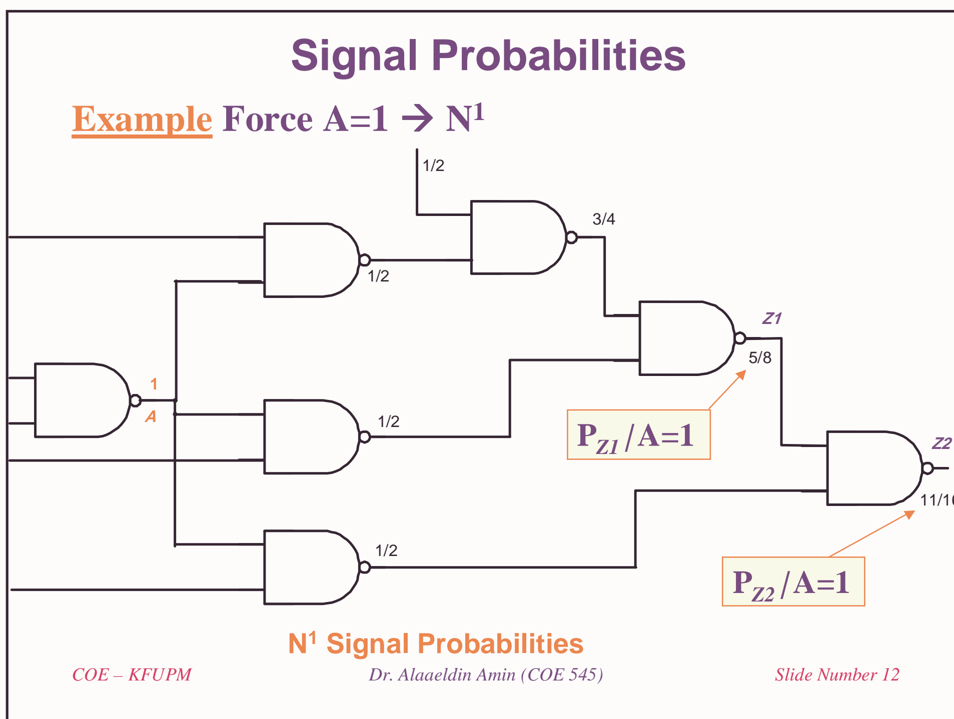
Signal Probabilities

Example

- Z_1 & Z_2 are Computed From Reconvergent Fanout Branches
- Signal Probabilities for Lines Oher than Z_1&Z_2 are Shown




Signal Probabilities
Example Force A=0 → N0
1/2
1/2
1
Z1
1/2
0
A
1
PZ1 / A=0
Z2
1/2
1
PZ2 / A=0
N0 Signal Probabilities
COE – KFUPM
Dr. Alaaeldin Amin (COE 545)
Slide Number 11


Signal Probabilities
Example Force A=1 → N1
1/2
3/4
1/2
Z1
5/8
1
A
1/2
PZ1 / A=1
Z2
11/16
1/2
PZ2 / A=1
N1 Signal Probabilities
COE – KFUPM
Dr. Alaaeldin Amin (COE 545)
Slide Number 12

Signal Probabilities

Example

$P_{Z1} = (P_{Z1}/A=0).(1- P_A) + (P_{Z1}/A=1).P_A$

$= (1/2).(1 - 3/4) + (5/8).(3/4) = 19/32$

$P_{Z2} = (P_{Z2}/A=0).(1- P_A) + (P_{Z2}/A=1).P_A$

$= (1/2).(1 - 3/4) + (11/16).(3/4) = 41/64$

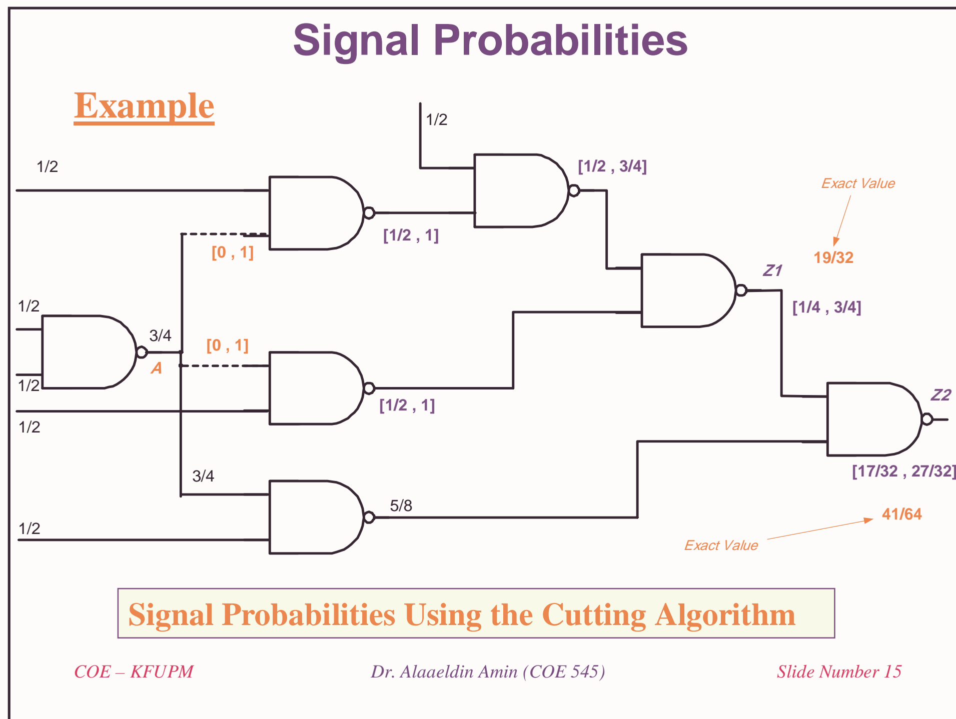
- **For a Circuit N with k Stems with Reconvergent Fanout, This Approach Requires** ***2^k*** **Copies of N with Signal Probabilities → *Impractical***
- ***Approximate*** **Algorithms Are Used, e.g. Cutting Algorithm**

COE – KFUPM Dr. Alaaeldin Amin (COE 545) Slide Number 13

Signal Probabilities
Cutting Algorithm [Savir, et. Al. 1984]

- Does Not Yield Exact Signal Probability P_x *(Approximate Solution)*
- Gives a *Range* for Signal Probabilities.
- For a Stem with ***k*** *Reconvergent* Fanout Branches, ***k-1*** Branches Are ***Cut*** and handled as PI's, except that they Are Assigned a Range [0 , 1] of Signal Probability.
- Only *Reconvergent* Fanout Branches Are *Cut*
- Resulting Circuit is Free of *Reconvergent* Fanout Branches

COE – KFUPM Dr. Alaaeldin Amin (COE 545) Slide Number 14



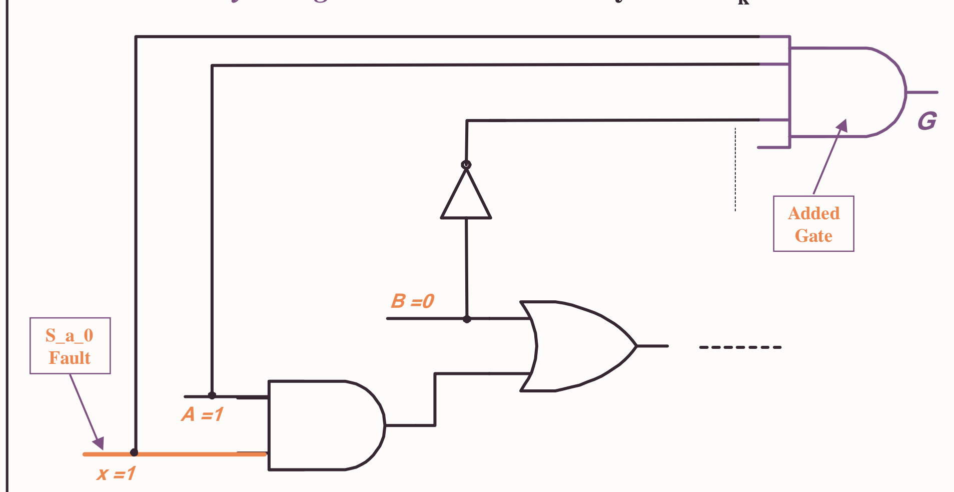
Detection Probability Based on Signal Probabilities

- Assume a CUT with Single PO
- Assume a Fault where Only *One Path* Exists from Fault to the only PO
- An Auxiliary AND gate Can be Added to Allow Computation of the Detectability of the Fault

COE – KFUPM *Dr. Alaaeldin Amin (COE 545)* *Slide Number 16*

Detection Probability Based on Signal Probabilities

- **Fault Detection Probability =** $d_f = P_x P_A P_{B=0} \ldots = P_G$
- **For Multi-Path, Multi PO CUT,** $d_f \geq P_{Gk}$ **is the Fault** *Detectability along Path k* **with Auxiliary Gate** G_k



ATPG Systems

Requirements:

- **High Fault coverage .**
- **Low Test Generation Cost**
- **Small Test Set Size**

Structure:

- **2-Phase System**

Phase1: **Fault Independent Test Generation (50-80% Fault Coverage)**

Phase2: **Fault Oriented Test Generation**

COE – KFUPM *Dr. Alaaeldin Amin (COE 545)* *Slide Number 18*

ATPG Systems

```
repeat
Generate_ test ( t );
fault simulate t
v = value(t)  -- measure of how good the Generated TV is
if acceptable( v) then add t to the Test Set
until endphase1();
```

```
repeat
select a new target fault f   /* Selection Criteria ? */
try to generate a new test t for f
IF   Successful   then
        Add t to the Test Set
        fault simulate t
        Discard the faults detected by t
End IF
until endphase2();
```

/ Test scheduling? */*

/ redundancy elimination ? */*

Stopping Criteria

1. *Exhausted Fault List*
2. *Too Many TVs*
3. *Achieved Prespecified Fault Coverage*

ATPG Systems

- Phase 1 issues:
  - When to stop phase 1 and switch to phase 2
    - Continue as long as many new faults are detected
    - Do not keep a test that does not detect many new faults
    - When many consecutive vectors have been discarded
    - Exhausted all Faults in the Fault List
    - A Prespecified Fault Coverage is Achieved
    - Test Set is too Large

ATPG Systems

- Phase 2 issues (during deterministic test generation):
  - When to Stop
  - What fault to pick next
  - Choice of backtrack limit
  - Switch heuristics
  - Interleave test generation and fault simulation
  - Identify untestable faults by other methods

COE – KFUPM Dr. Alaaeldin Amin (COE 545) Slide Number 21

ATPG Systems

- **When To Stop**
  - Exhausted Fault List
  - A Prespecified Fault Coverage is Achieved
  - Test Set is too Large

COE – KFUPM Dr. Alaaeldin Amin (COE 545) Slide Number 22

ATPG Systems

- *What fault to pick next*
  - Target to generate tests for easy faults first
    - Hard faults may get detected with no extra effort
  - Target to generate tests for hard faults first
    - Easy faults will be detected any way, why waste time
  - Target faults near PIs – Sensitize Long Paths
  - Target faults near POs -- Sensitize Long Paths

COE – KFUPM Dr. Alaaeldin Amin (COE 545) Slide Number 23

ATPG Systems

- Choice of backtrack limit (High backtrack ⇒ more time)
  - It has been observed that *as the number of backtrack increases the success rate goes down.* Thus we may wish to keep low backtrack limit.
  - Some faults may not get detected due to lack of time spent on them
  - Could start with low limit and increase it when necessary or in second round (often used heuristic)

COE – KFUPM Dr. Alaaeldin Amin (COE 545) Slide Number 24

ATPG Systems

- Switch heuristics
  - ➢Switch between heuristics during backtrace as well as during backtrack
- Interleave Test generation and Fault Simulation
  - ➢Drop Detected Faults after Generation of each Test
    - ❑This has higher switching cost but generally works well
    - ❑Sequential tests may not have other options and this may be the only practical option in some cases

COE – KFUPM Dr. Alaaeldin Amin (COE 545) Slide Number 25

ATPG Systems

- Identify Untestable / Redundant faults by other methods
  - ➢If the goal is to identify only Untestable faults as opposed to find a test, other methods may do a better job – example of such techniques are:
    - ❑Recursive learning
    - ❑Testability evaluations

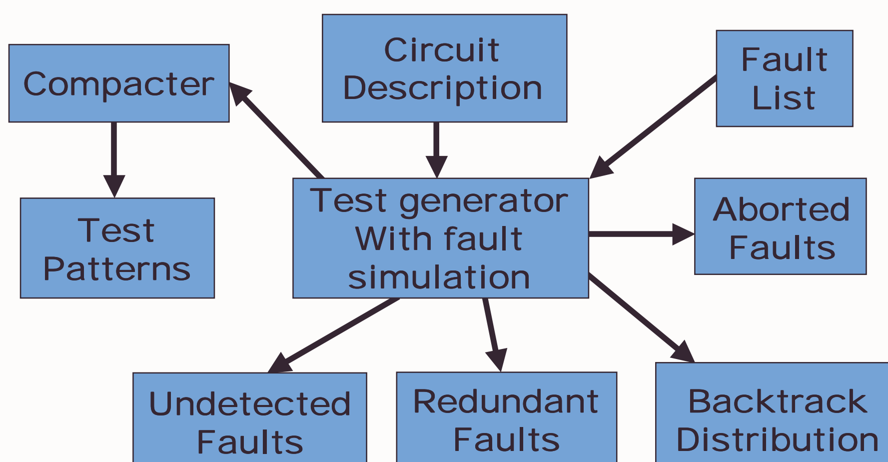
COE – KFUPM Dr. Alaaeldin Amin (COE 545) Slide Number 26

Fault Coverage and Efficiency

$$\textit{Fault coverage} = \frac{\text{\# of detected faults}}{\text{Total \# faults}}$$

$$\textit{Fault efficiency} = \frac{\text{\# of detected faults}}{\text{Total \# faults -- \# undetectable faults}}$$

ATPG Systems



ATPG Systems -- Compaction of TVs

- *Static compaction* (*Post Processing*)
  - **ATPG should leave unassigned inputs as X**
  - **Two patterns *compatible* – if no conflicting values for any PI**
  - **Combine two tests t_a and t_b into one test** $t_{ab} = t_a \cap t_b$ **using D-intersection**
  - **Detects *union* of faults detected by t_a & t_b**
- *Dynamic compaction*
  - **Process every partially-done ATPG vector immediately**
  - **Assign 0 or 1 to PIs to test additional faults**

Compaction Example

- $t_1 = 0\ 1\ X$ $t_2 = 0\ X\ 1$
  $t_3 = 0\ X\ 0$ $t_4 = X\ 0\ 1$
- **Combine t_1 and t_3, then t_2 and t_4**
- **Obtain:**
  - $t_{13} = 0\ 1\ 0$ $t_{24} = 0\ 0\ 1$
- ***Test Length* shortened from 4 to 2**

Test Compaction
Reverse-Order Fault Simulation

- **Fault simulate test patterns in** ***reverse order of generation***
  - **Fault-Oriented patterns go first**
  - **Randomly-generated patterns go last (because they may have less coverage)**
  - **When coverage reaches 100%, drop remaining patterns (which are the useless random ones)**
  - **Significantly shortens test sequence – economic cost reduction**

COE – KFUPM *Dr. Alaaeldin Amin (COE 545)* *Slide Number 31*