Corso Luigi Einaudi, 55/B - Torino

Appunti universitari
Tesi di laurea
Cartoleria e cancelleria
Stampa file e fotocopie
Print on demand
Rilegature

NUMERO: 2339A

ANNO: 2018

A P P U N T I

STUDENTE: Caldera Filippo

MATERIA: Testing and fault tollerance - Prof. Sonza

Il presente lavoro nasce dall'impegno dell'autore ed è distribuito in accordo con il Centro Appunti.
Tutti i diritti sono riservati. È vietata qualsiasi riproduzione, copia totale o parziale, dei contenuti inseriti nel presente volume, ivi inclusa la memorizzazione, rielaborazione, diffusione o distribuzione dei contenuti stessi mediante qualunque supporto magnetico o cartaceo, piattaforma tecnologica o rete telematica, senza previa autorizzazione scritta dell'autore.

ATTENZIONE: QUESTI APPUNTI SONO FATTI DA STUDENTI E NON SONO STATI VISIONATI DAL DOCENTE.
IL NOME DEL PROFESSORE, SERVE SOLO PER IDENTIFICARE IL CORSO.

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 3 di 180

TFT

Testing and Fault Tolerance

- M. Sonza -

FILIPPO
CALDERA

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 3 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 5 di 180

L1 10/06/17

TEST = detect the presence of a fault.

Establish the probability of faults ⇒ Dependability
=
characteristic of a system where a fault may produce a critical behavior

1. Estimate P[failure]
2. Act to make the system work by reducing P["failure"] lower

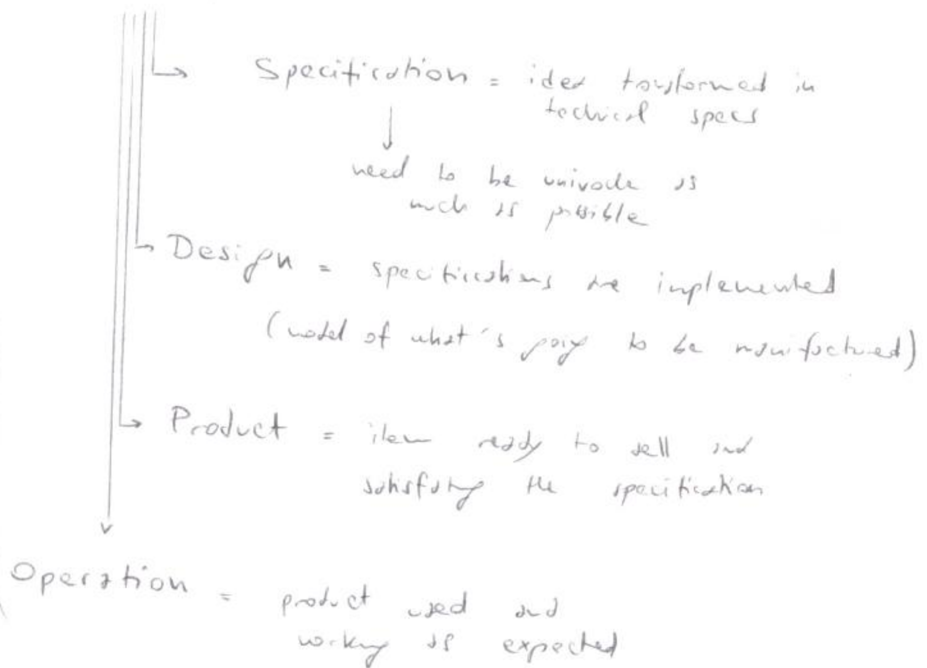
© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 5 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 7 di 180

Product specification = list of what the product is supposed to do.
↓
ALWAYS WRITTEN

MISSION = description of the function in the environment and their duration.

From the first idea to the ready-to-use product there is a long process = Life Cycle
↓
divided in phases



phases duration depend by the product

EX.

Car ECU

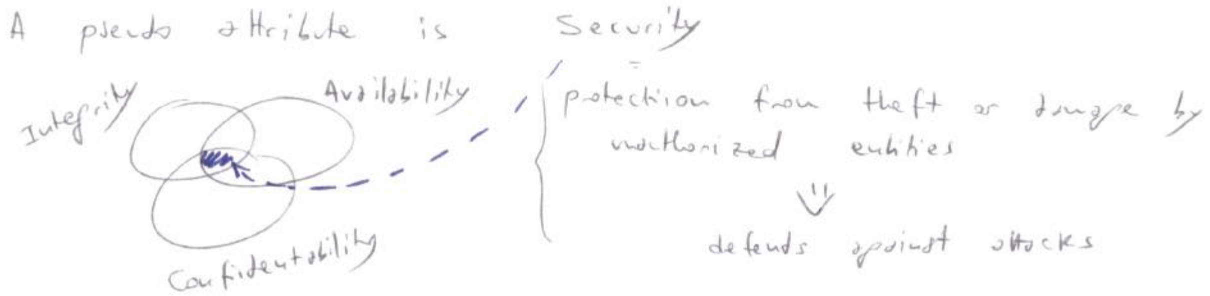
Design last years → Operation last decade

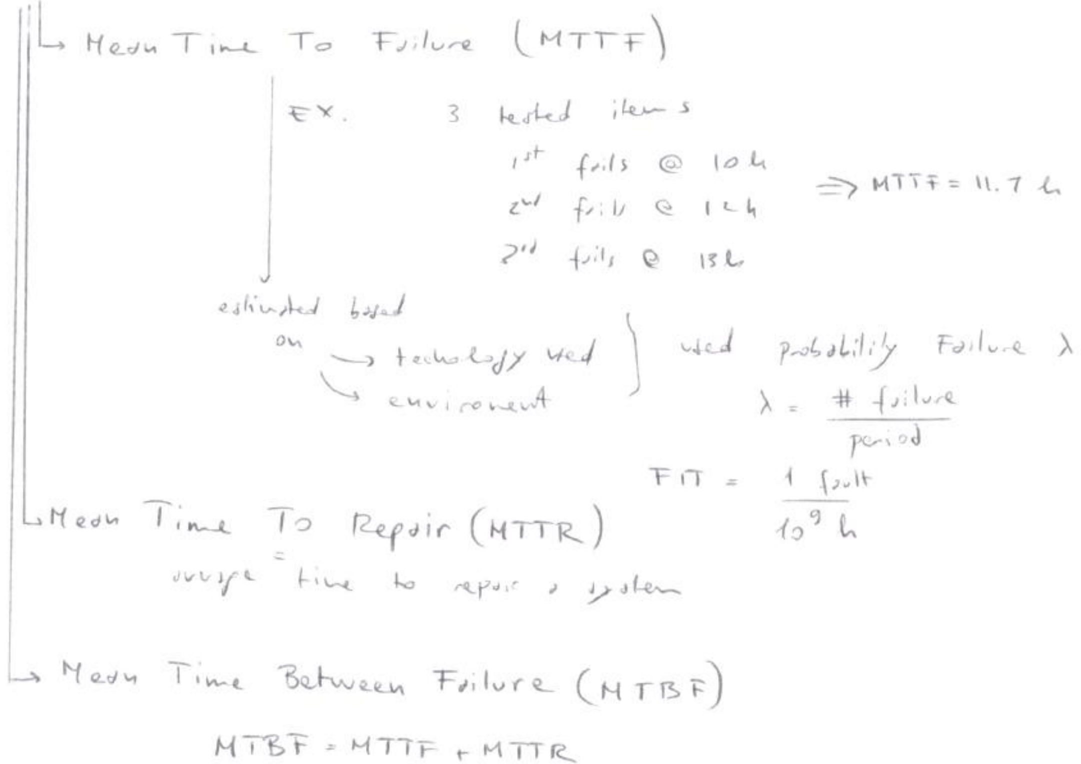
Space Launcher

Design lasts years → Operation takes 30 min

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 7 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 9 di 180

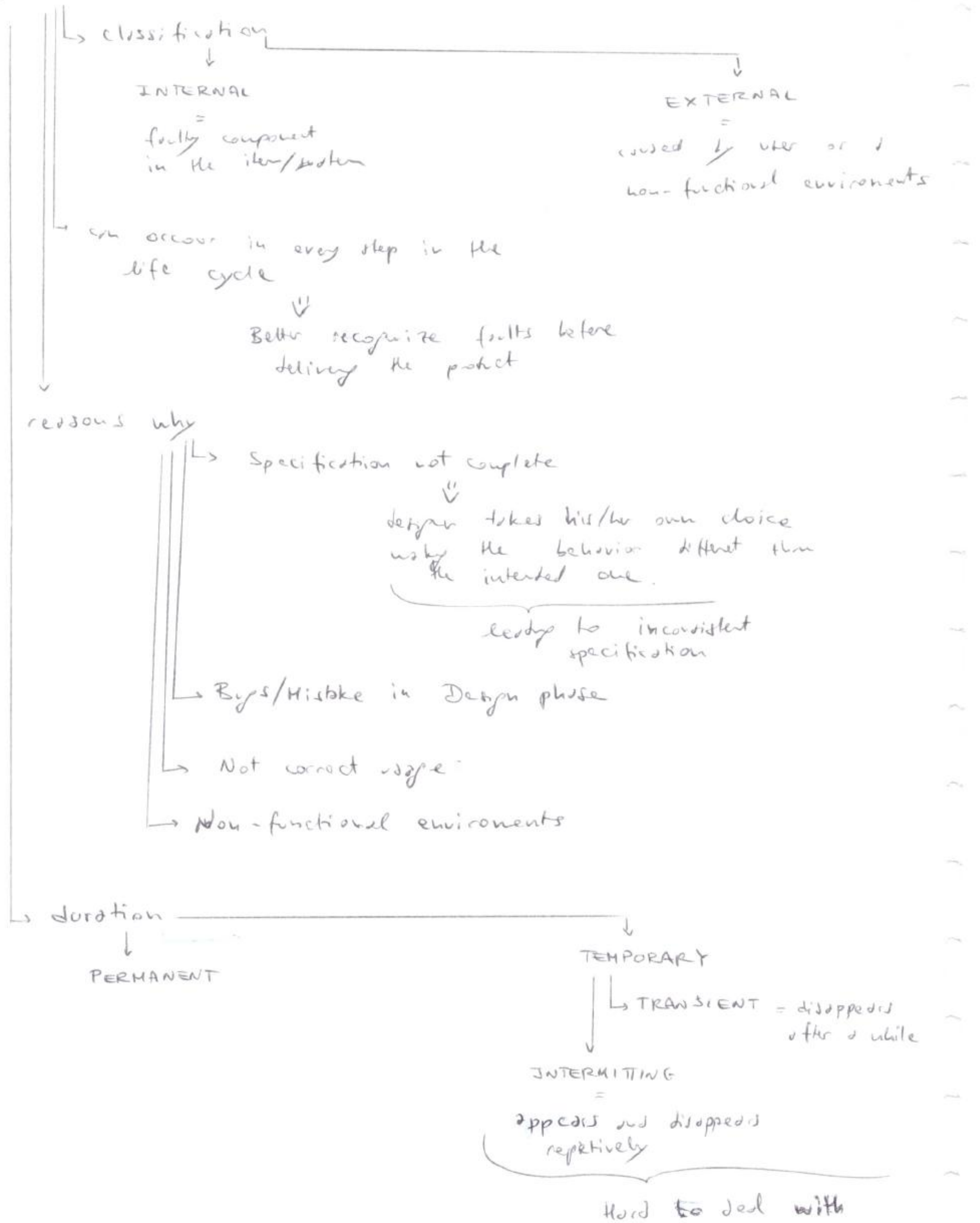




THREATS = fault causes

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 9 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 11 di 180



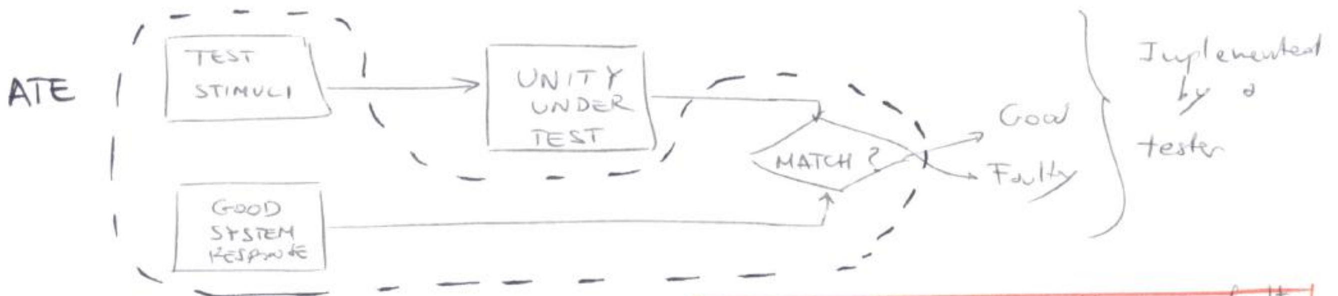
© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 11 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 13 di 180

L3 10/13/17

TEST = process to identify faults in projects
- very important, expecially in the design phase
- uses Faults Models = typical Faults = fault models
  ⇓
  makes testing easier ⇐ Faults are generalized in classes of models

Testing Base scheme is



stimuli must be close in order to provoke fault, if presents ⇐ In order to recognize a fault it need to produce a failure

V.I.

2 phases
- TEST STIMULI GENERATION
  =
  chosing the right stimuli in order to discover as much as possible any possible faults
  - done before manufacturing
  - cost depends by → UUT → Constrains
- TEST APPLICATION
  =
  provide as input the stimuli chosen and check for failures
  ⇓
  compare system output with the expected ones } Accomplished by Automatic Test Equipment (ATE)
  - costly instruments ⇕
  - precise

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 13 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 15 di 180

DEFECT LEVEL = % of faulty devices that pass the test.

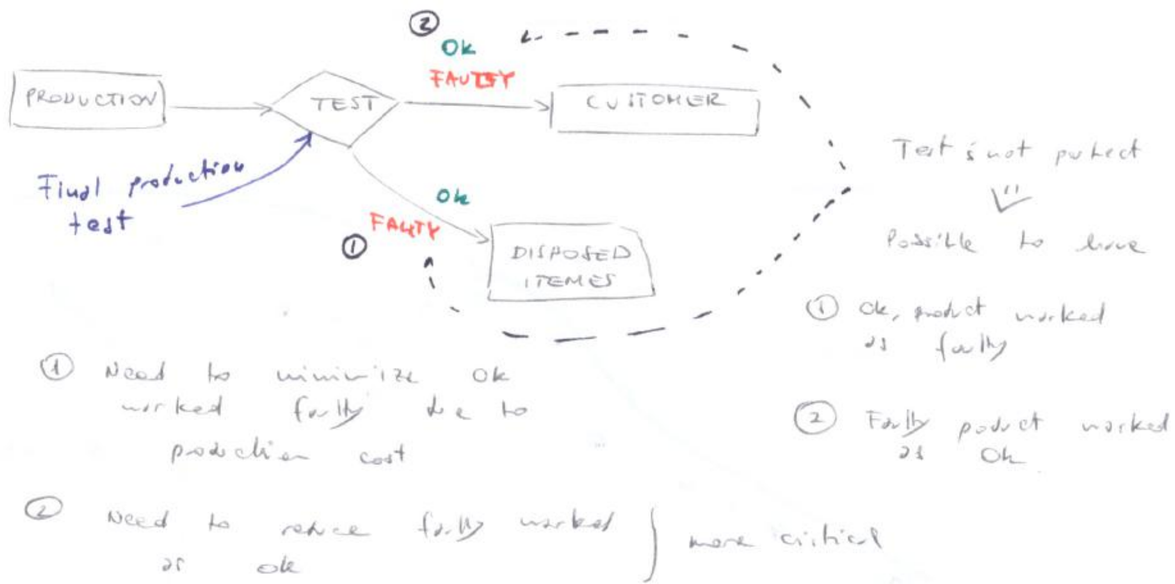
- measured in ppm
- adaptive to product dependability

  ex.

  Product for Aerospace ⇒ low DF

  " " consumer electronics ⇒ decent DF is acceptable

- scheme



Test is not perfect

⇓

Possible to have

① Ok, product marked as faulty

② Faulty product marked as Ok

① Need to minimize Ok marked faulty due to production cost

② Need to reduce faulty marked as ok } more critical

Computation methods → Analitic ⇒ Estimation

$$DL = 1 - Y^{1-T}$$ = faulty products passing the test

Y = manufacturing yield
= % of good product

T = Fault Coverage
= how strick the test is
= % of faults detected over the total faults
→ depends on fault-model

↓

Experimental

= Faulty products / Total products

(takes long time; return for the field after enough long time)

After all the production

⇓

No a priori estimation

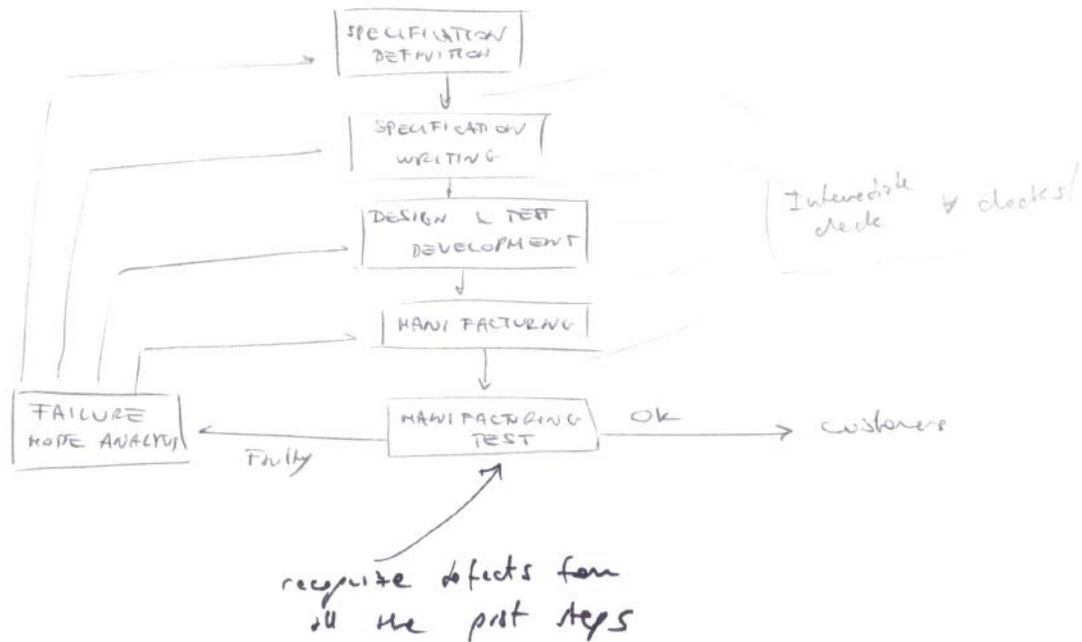
Computation of DL depends by company and models used.

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 15 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 17 di 180

TESTING IN DESIGN AND PRODUCTION

Faults happen for many reasons ⇒ Faults due to → production line
→ design phase ← solved before production starts



© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 17 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 19 di 180

INCOMING INSPECTION - tests performed by customers to make sure items work before using it

Customers don't know the design behind the item ⇒ Test performed with limited infos.

ON-LINE TEST - item tested when it is used in the working product

Electronics usage deteriorates the components ⇒ checking behavior during the working phase

⇓

- No tester used
- Device not removed from the environment

⇒ Need a design for testability

Performed in order to guarantee or check the system reliability

EX. Car ECU

Each few ms a test is performed to check the correct behavior

2 types

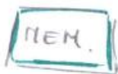
- **CONCURRENT** = system tested without stopping operations
- **NON-CONCURRENT** = system tested during idle times or stopping operation.

DESIGN FOR TESTABILITY = design unit in order to have easy access to test it.

EX.

Memory Test

Memory is designed | also a test circuit is designed with memory

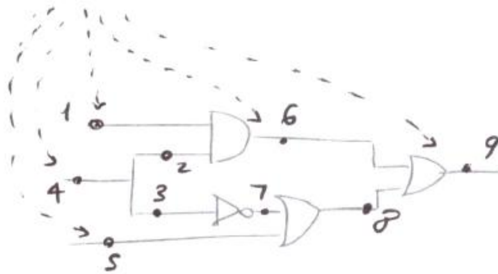


Circuits performs a R/W sequence when required and provide a boolean output reporting the test outcome (OK, FAULTY)

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 19 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 21 di 180

- Stuck-at location : point in the circuit where stuck-at can occurs.



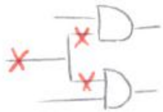
For any stuck-at location we have 2 faults case: s-a-0, s-a-1

⇓

We have

2 m = # Faults possible

m = # checkpoints = # stuck-at-locations

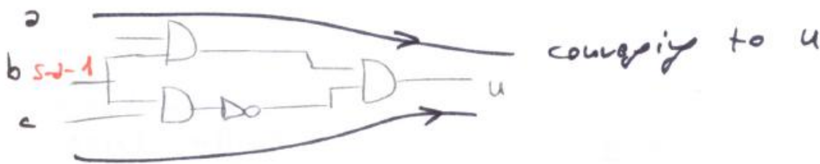


Defect with the same condition but with different observation condition ⇒ Different Test vectors

When the faults converge to a single fanout

RECONVERGENT FANOUT

ex

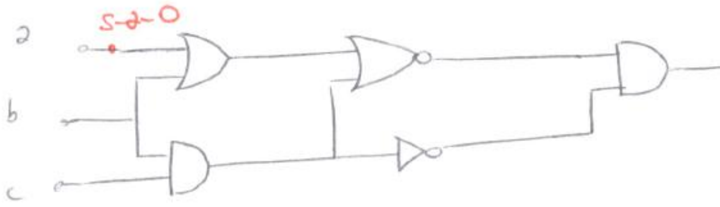


© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 21 di 180

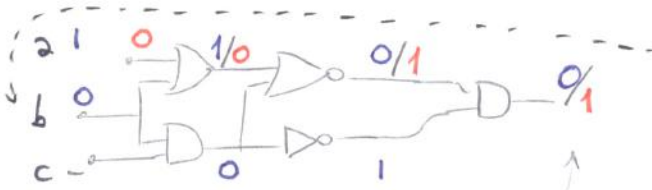
© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 23 di 180

Ex 3

$n = 12 \Rightarrow$ # Total Faults $= 24$



a s-a-0 $\Rightarrow$ a = 1



b = 0 in order to make OR gate transparent to the fault

c = - since 0 and 1 are both ok

$0 \neq 1 \Rightarrow$ Failure $\Rightarrow$ {1, 0, -} is a test vector

winning strategy is to make gates transparent to faults

Let's use "transparent rules"

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 23 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 25 di 180

For each fault we need to create a failure.

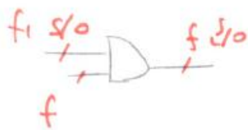
In practice the device is subject to multiple inputs during times

TEST SET = set of input vectors detecting a fault.

EX.

$$T_f = \begin{Bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{Bmatrix}$$ ← All these vectors can provoke a failure due to S/1

Sometimes a single vector can detect more than a fault.



$T_{f_1} = \{1, 1\}$
$T_{f_2} = \{1, 1\}$ ⇒ Same test vectors ⇒ Faults are equivalent
$T_{f_3} = \{1, 1\}$

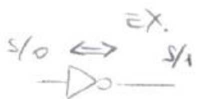
allows to reduce the test list

Another way to reduce the test list and make the test easier is use TRANSITIVITY

↳ for each gate is possible to reduce 2 faults based on I/O gate relation.

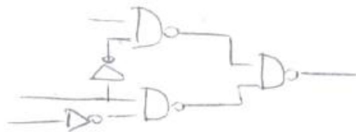
EX.

S/0 ⇔ S/1



other gates

EX.



$n = 10 \Rightarrow$ Faults $= 20$

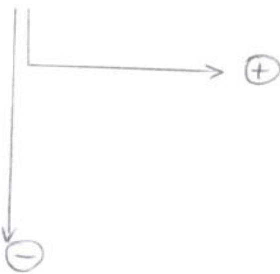
Gate $= 5 \Rightarrow$ Deletable Faults $= 10$

Fault to Test $= 20 - 10 = 10$

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 25 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 27 di 180

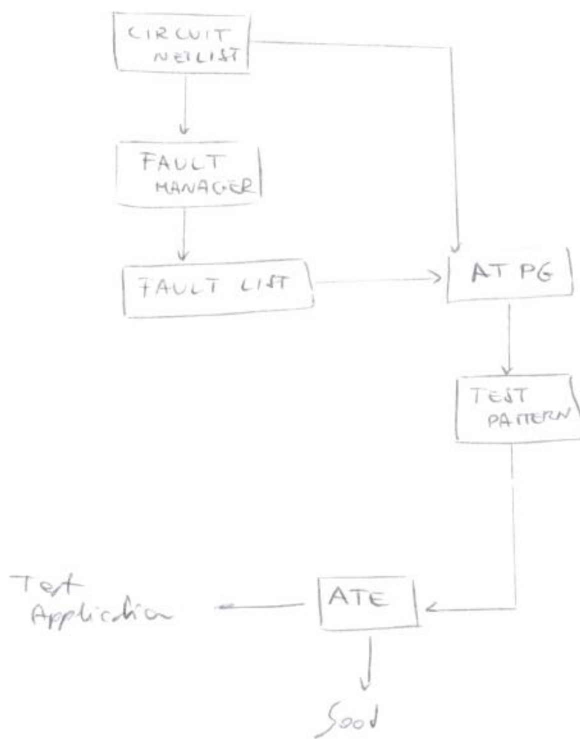
STUCK-AT model is always used in industry



- defects behaves like the model
  (like a way connection to Vss or GND)
- Supported by CAD tools

- Can't model open circuits } solved by pattern test.

In general the test procedure is this:



© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 27 di 180

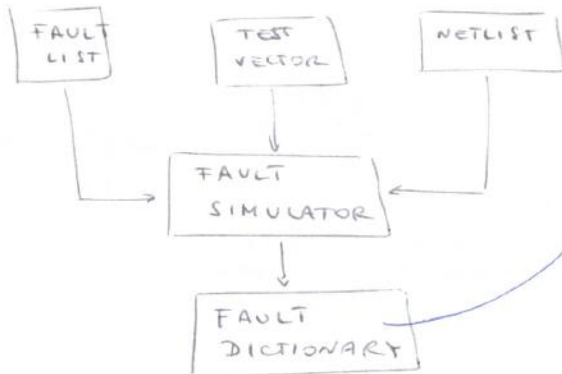
© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 29 di 180

L5 10/20/17

TEST EQUIVALENCE
=
two faults have the same test set

FUNCTIONAL EQUIVALENCE
=
two faults have the same functional equation
=
$\delta f_1 = \delta f_2$

If Functional Equivalence => Test Equivalent
⇍



Stores faulty circuit behavior given set of TV

n faults => n output sets.

Comparing output from test and scanning the fault dictionary is possible to distinguish the fault.

If
f_{k_1}, f_{k_2} are functionally equivalent => $\delta f_{k_1} = \delta f_{k_2}$ = faults modify behavior in the same way
⇓
Hard to distinguish the faults.

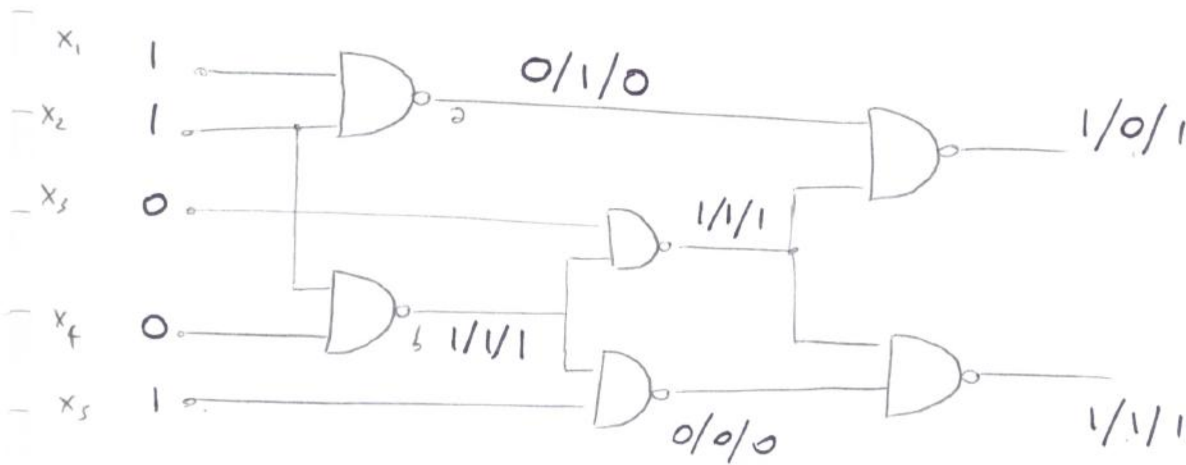
If f_{w_1}, f_{w_2} have same output behavior =>
- f_{w_1}, f_{w_2} functionally equivalent
- f_{w_1}, f_{w_2} not distinguishable since ∄ T.V. able to produce different outputs

Some time, improving TV is possible to distinguish HW fault if not funcionally equivalent } = Distinguishability
↓
needed for localize faults in a circuit

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 29 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 31 di 180

Ex Identify test vectors to distinguish a/1 and b/1



Good C / Circuit with a/1 / Circuit with b/1

If applying $\{1, 1, 0, 0, 1\}$

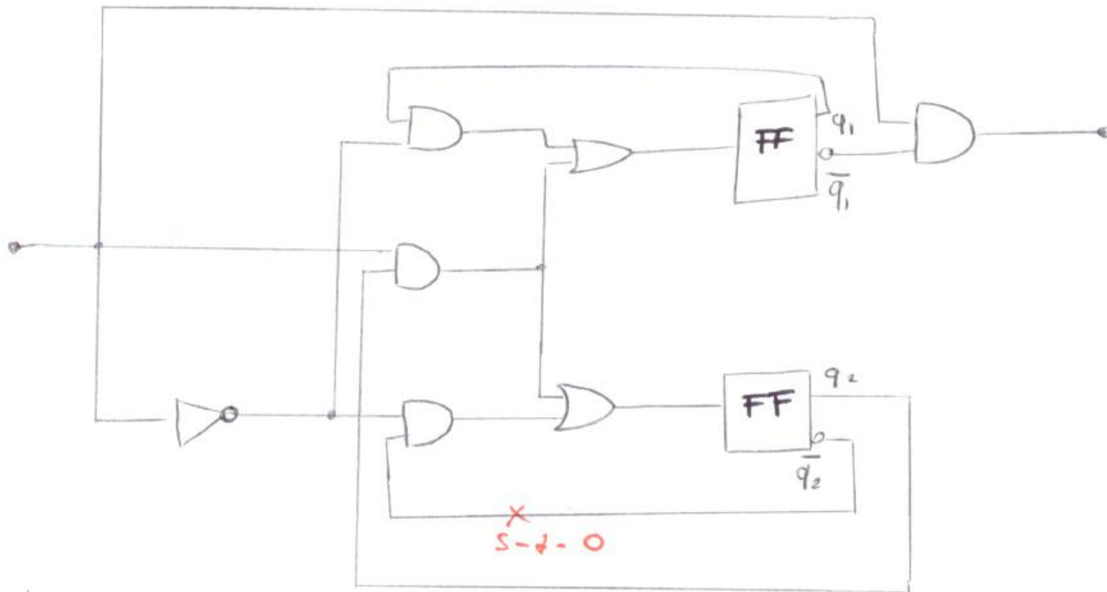
$$\text{if} \quad u = \begin{cases} \{0, 1\} \Rightarrow \text{fault separable is } a/1 \\ \{1, 1\} \Rightarrow \text{fault separable is } b/1 \end{cases}$$

Some tools provides $TFC = \dfrac{\#\text{ detected Faults}}{\#\text{ Tot} - \#\text{ undetectable}}$

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 31 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 33 di 180

EX



@ start $q_1 = q_2 = 0$

let's assume S-a-0

STEP 1: GET $\bar{q}_2 = 1$ TO EXCITING FAULT

$q_2 = 0 \Rightarrow \bar{q}_2 = 1 \Rightarrow$ fault excited already

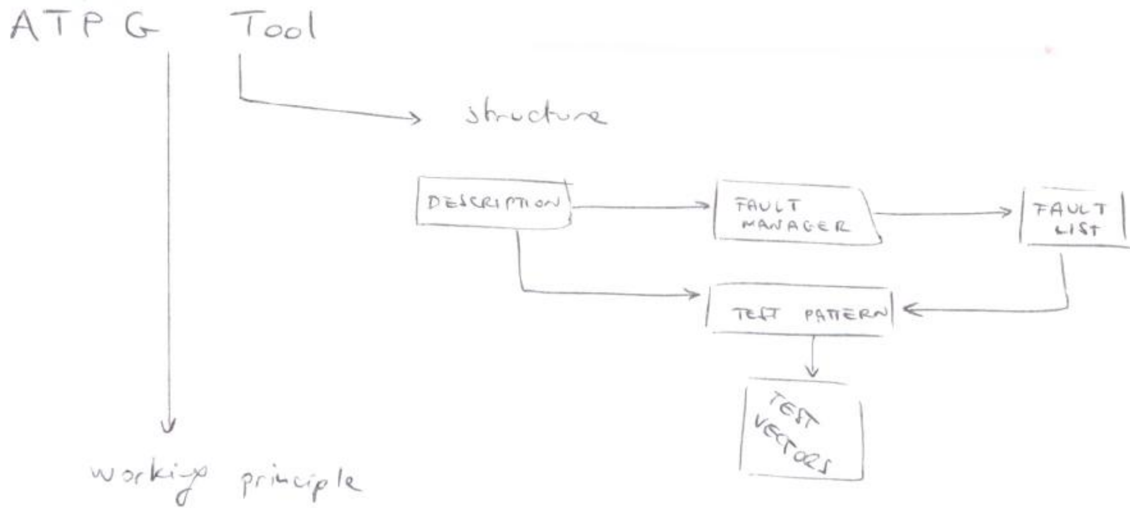
STEP 2:

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 33 di 180

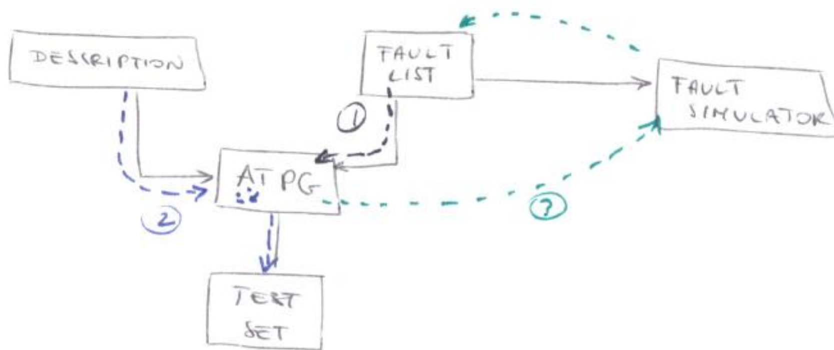
© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 35 di 180

ATPG Tool

→ structure



→ working principle



STEP 1: ATPG reads fault from FAULT LIST

STEP 2: ATPG generate a test vector for that fault using DESCRIPTION

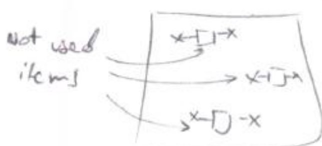
STEP 3: ATPG asks to FAULT SIMULATOR if that test vector simulate the fault or other faults. FAULT SIM. sets a flag on the simulated fault in the FAULT LIST } helps FAULT DROPPING =

⇓

simulating untested faults only ⟹ makes simulation faster

In order to make testing faster a circuit analysis is performed before start simulation.

Usually a circuit present unused items due to development evolution



Gates are alone and not connected ⟹ Fault there would be untestable

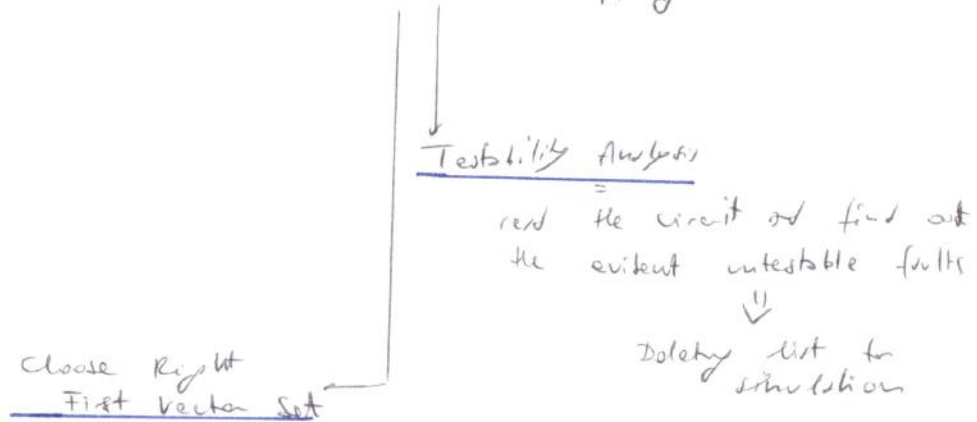
⇓

Time Saved { FAULT MANAGER recognize these faults and doesn't allow to simulate them

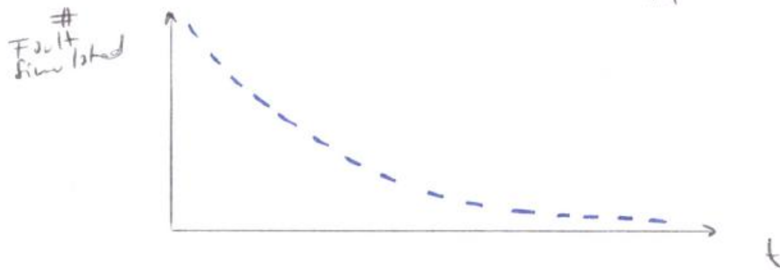
© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 35 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 37 di 180

There are some Tricks for Fault Dropping



If there is fault Dropping ⇒ t_{SIM} reduced



Another good practice is **TEST SET COMPACTION** → ways:

Reverse Order Fault Simulation

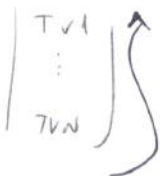
ATPG provides TVs

TV1	f0 → detect 4 faults
TV2	f1 → detect 3 faults
⋮	⋮
TVN	

more we go on less faults would be detected due to Fault Dropping.

⇓

Possible to reverse the TVs scanning.



If a test vector recognize fault which was already recognized by another TV then it's skipped (no sense simulate them)

Exploiting Don't Care

00_ = 001 and 000

ATPG, however deals with 0,1 only

⇓

usually we set 00 Random(0,1)

- Random value 0/1 is place @ "_" → NOT SO GOOD

set 0 or 1 based on the most efficient way ⇐ BETTER

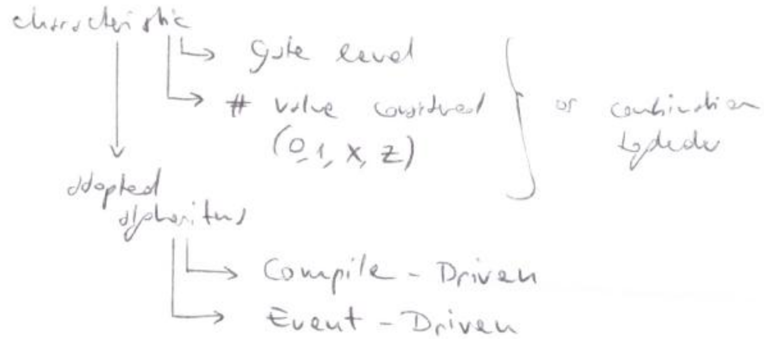
© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 37 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 39 di 180

LOGIC SIMULATION

L7 10/30/2017

= sim of faulty-free circuit at gate level

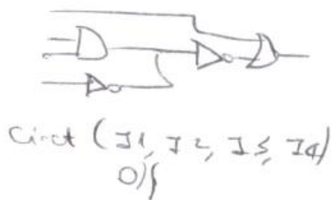


characteristic
- gate level
- # value considered (0, 1, X, Z)

} or combination together

adopted algorithms
- Compile - Driven
- Event - Driven

- **Compile - Driven** = read netlist and write code representing the circuit



Circt (I1, I2, I3, I4) O/s

⋮ // Instruction

}

⇓

- Instruction for any gate
- Variables for I/O and lines

Usually the variable used are char in order to contain also X, Z.

} for 0,1 simulation is possible to exploit the 8-bit char tipe in order to have multiple test vector applyed

=

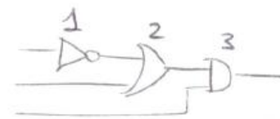
PARALLEL GATE SIMULATION

#PIs = 253
#POs = 372
#Gates = 123.318 ⇒ 123.318 instructions

To solve is needed a ~~good~~ correct execution order (need gate Inputs for computing its output)

⇓

Levelization Needed
=
compile low level first to provide the next level the needed inputs



(+) Very Fast simulation

(−) . Low portability due to Assembly
No delay simulation
. For each simulation all gates are re-computed

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 39 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 41 di 180

STRATEGY TO SHORT SIMULATION TIME

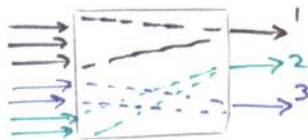
Rare cases:

- **Exhaustive Testing**

  If PIs < 20 ⇒ All possible combination are still good for computation
  ⇓
  Very Good Test Quality ⇐ All combinations tested

- **Pseudo Exhaustive**

  If PIs > 20 ⇒ possible to apply E.T. on some subsets of inputs



1 depends only by black PIs
2 " " " Green PIs
3 " " " Blue PIs

Let's apply E.T. for all the subsets
⇓

$\# \text{Comb} = 2^3 + 2^2 + 2^2 < 2^7$

(2^3 by 1, 2^2 by 2, 2^2 by 3; 2^7 ← case of ET applied for all the input)

ATPG Based:

- **Algebraic Methods**

  Use boolean function of both faulty and faulty-free circuit to get the test set

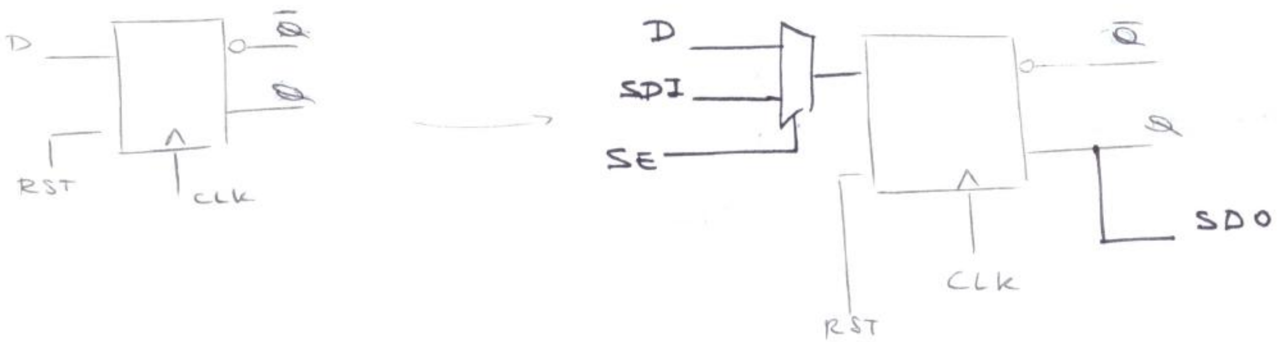
- **Topological Methods**

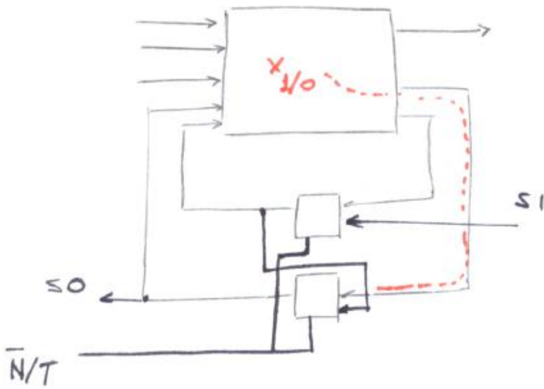
  ↳ based on branch-and-bound
  =
  Test vector obtained combining fault excitation and observability conditions.

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 41 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 43 di 180

In order to turn FFs into Shift Register a little HW modification is needed





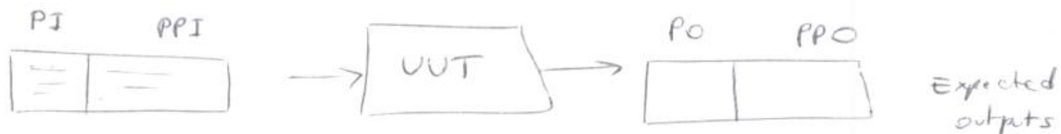
When there is a stuck-at fault it must be propagated till the FF then insert it in the FF and check the fault observing SO

Sequence is:

- Enter Test Mode ⇒ $\overline{N}/T = 1$
- Shift S1
- Apply Test Vector on PIs
- Enter Normal Mode ⇒ $\overline{N}/T = 0$
- wait a CLK
- Read SO

} ∀ test vectors

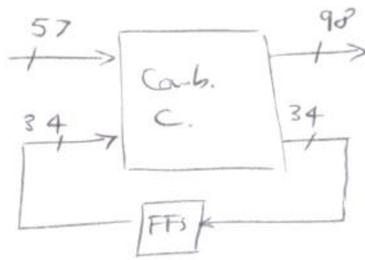
Test set is generated by ATPG



© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 43 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 45 di 180

EX.



\# TVs = 315

Test Vector Lenght = 57 + 34 = 91 bits

Expected Output Vector Lenght = 98 + 34 = 132 bits

Test Duration for a single TV (CLK) = 34 + 1 + 34 = 69 CLK

(34: Shift IN; 1: Load PI; 34: Shift OUT)

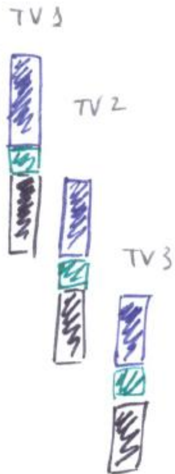
Total Test Duration = 69 CLK · 315 = .. high = Costly Test

Need to make test faster ← Costly Test

=

Let's exploit overlapping between next Shift IN and past Shift OUT

let's unload the FFs while loading the next Scan In



CLK

(legend) = Scan out; = Normal mode; = Scan In

← while loading next value the old one are put to Scan out ⇒ Total time = (34+1) 315 + 34

Exploiting overlapping test duration is

Test Duration = $M_{TV}(M_{FF}+1) + M_{FF}$

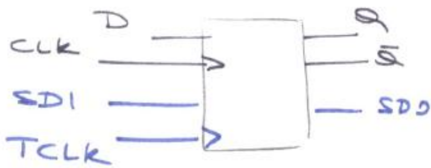
$T_{TEST} \neq T_{CLK} T_{TEST}|_{CLK}$ ⇐ In test mode the circuit is slower

$T_{CLK}|_{TEST} > T_{CLK}|_{OPERATIONAL}$

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 45 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 47 di 180

An alternative way to scanning is implementing FFs with dual clock



Operational mode
Testing mode

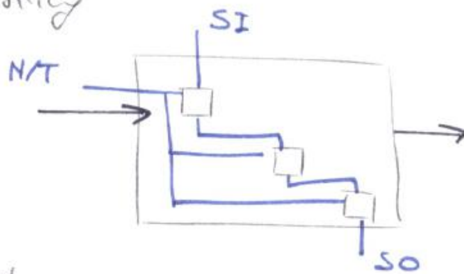
Basically are two FFs in one

⇓

No T/N but 2 CLK signal ← Not very appreciated by designers

usually not used

Scan testing



Stuck-at can occur also on the scan-chain ⇒ before test, test the chain

SHIFT TEST performed <

↳ steps

Teskeption

contribution to test time:

$M_{FF} + 4$

1. Enter Test-Mode
2. Force SI in sequences of 1 and 0.
3. Scan-out and check if SI SO sequences match

Overall time:

Shift-Time $= M_{FF} + 4$
ATPG Test $= (M_{FF}+1)M_T + M_{FF}$

⇓

Total Test Time $= (M_{FF}+1)M_T + M_{FF} + M_{FF} + 4 =$

$= O(M_T M_{FF})$

↑

= Long Time

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 47 di 180

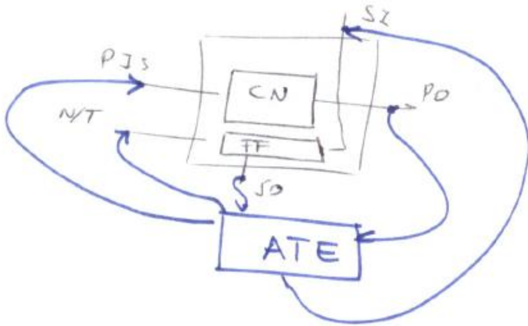
© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 49 di 180

EX

PIs = 107 # FF = 2145 Using scan test with only one chain.
POs = 212 # TV = 82

1. How is memory size for test?
2. How long does the test take?

1. The situation is:



Automatic Tool Equipment has to control the test pin and store the I/O data to check them later.

∀ TV are needed : 107 + 2145 + 212 + 2145 ~ 5000 bits
(PIs, SI, POs, SO)

$M_{TV} = 82$

⟹ $Mem_{TOTAL} = (107 + 2145 + 212 + 2145) \cdot 82 \approx$

$\sim 500'000$ bits

⇓

Memory Needed = 500 kbits

2.

$T_{TEST} \simeq M_T \cdot M_{FF} = 82 \cdot 2145 = 175890$ ← Approx

$T_{TEST} = \underbrace{M_T (M_{FF} + 1) + M_{FF}}_{ATPG} + \underbrace{(M_{FF} + 4)}_{Shift-time} = 180266$ ← Real Tck

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 49 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 51 di 180

SCAN TEST DERIVATIONS

PARTIAL SCAN

= not all FFs are in the scan-chain
⇓
No possible to observe those FFs values

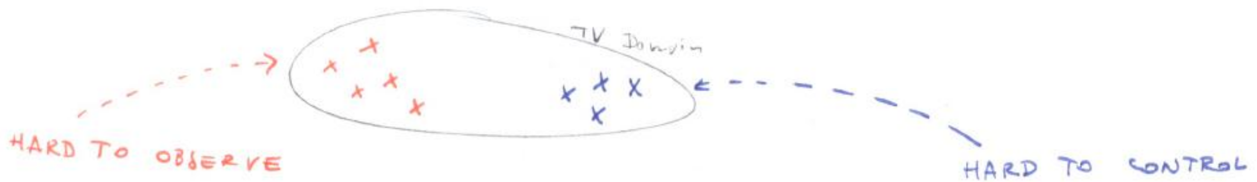
performed because in some FFs the value is easy to obtain or possible to reach in other ways

Not so used, not worth it.

TEST POINTS IN DESIGN

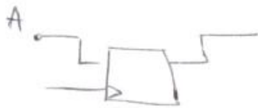
If the circuit is large and difficult ⇒ Testing Time is very long
An aspect of testing is to reduce the testing time.
⇓
TV must be reduced.

Analyzing the circuit is possible to recognize some test points



HARD TO OBSERVE

A FF is introduced in the scan-chain.
Used only in Test Mode making the critical fault observable



HARD TO CONTROL

Node is far from PIs
⇓
Difficult to obtain the wanted result starting from PIs

To solve, a FFs inserted to scan-chain, control the hard point in the circuit.

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 51 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 53 di 180

There are additional fault models, not only the stuck-at **L9** 11/10/17

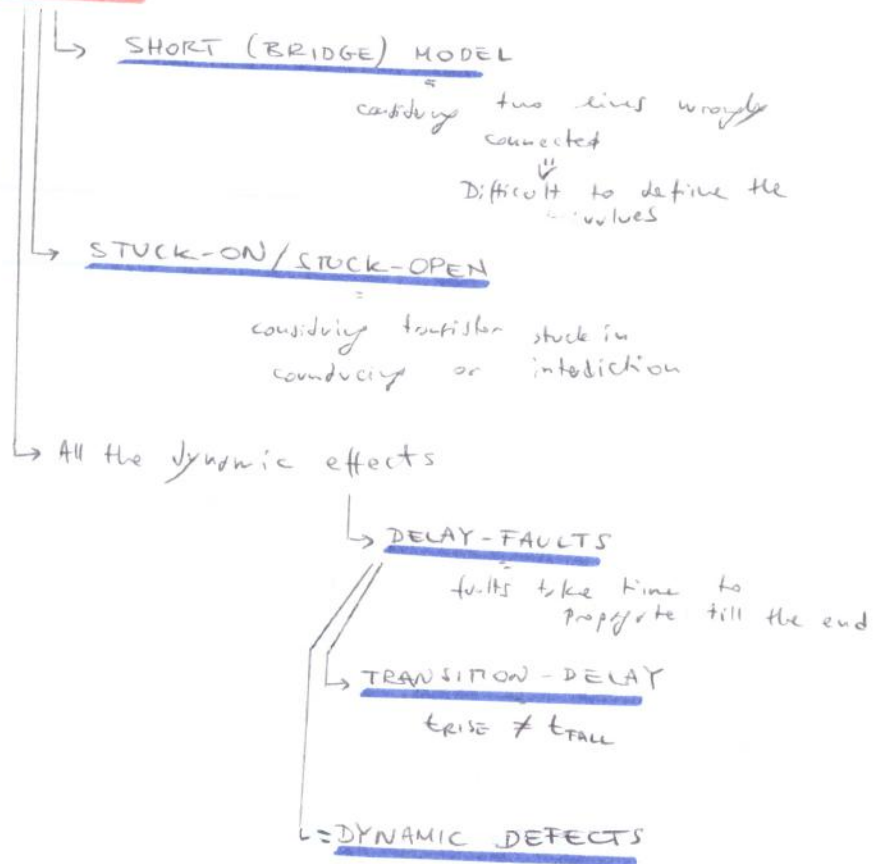
The main goal of testing is produce a given defect level

To have the wanted F.C.
an "endurance" procedure is used
=
Testing with stuck-at first

FC ≥ WANTED — Y → Test Done = Stuck-at is sufficient

N → Need to test using others faults models combined with the stuck-at

OTHER FAULT MODELS



© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 53 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 55 di 180

- **STUCK-ON / STUCK-OPEN** = transistors stuck at conducting or interdiction
  ⇓
  looking at gate level

Possible to simulate transistors switching affirming that a transistor is stuck in a configuration.

Independently on V_{gs} there are 2 configuration
- stuck-on ⇒ conducting
- stuck-open ⇒ interdiction

In a circuit there are million/bilion of transistors ⇒ Test them all is gonna be hard

CELL-AWARE TEST = Usually only a small cell of the circuit is considered for this test

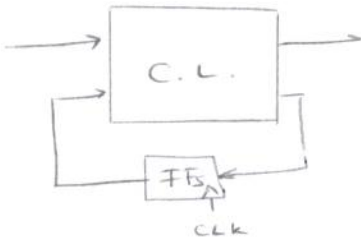
Considering only a cell where transistor level is known

Generate T.V. to the cell and check the output.

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 55 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 57 di 180

- **DYNAMIC DEFECTS**
  ↳ affects the timing required by the circuit to reach steady state



f_{CLK} is related to combinational logic delay

Usually $\frac{1}{f} > \Delta$ in order to have the outputs stable.

If there is a delay in the network it could produce a delayed output leading to an error.

$$\Delta\Big|_{DEFECT} > \frac{1}{f}$$

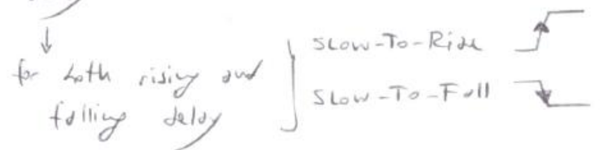
⇓

To test the circuit we need to test it @ $f_{CLK\ MAX}$

If test is performed @ $f < f_{MAX}$ ⇒ possible to not detect any faults

Models

- **TRANSITION DELAY** = model defects as delay
  ↓
  for both rising and falling delay



When new input is applied to a gate it takes a bit to have the output result updated

we could have

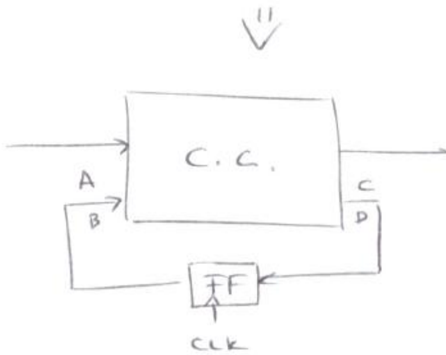
δ_R, δ_F } Provided by library used
Could change due to fault, but changes are unknown.

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 57 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 59 di 180

- Assuming the circuit is a sequential circuit

⇓



Assuming that:

$\delta = 1\,ns \;\; \forall$ gates

$\delta_F = \delta_R = 1\,ns$

let's assume 4 gates $\Rightarrow \Delta_{MAX} = 4 \cdot 1\,ns = 4\,ns$

$\Delta_M = 4\,ns \Rightarrow f_M = \frac{1}{\Delta_M} = 250\,MHz$

let's run circuit @ $f_{CLK} = 200MHz$, considering slack

$t_{SLACK} = 0.02\,\mu s$

$f = 200MHz \Rightarrow T = 5\,ns \Rightarrow$ outputs C, D sampled @ 5 ns

Assuming defect in A causing a 1ns delayed transition in C.

⇓

using correct sampling would get wrong values and transition won't appears.

To "see" the values FFs are used

TRANSIENT DEFECTS are difficult to detect and propagate

If there is a general defect in an area affecting multiple gates = Distributed Defects → can't be used with transition fault model

↙

requires PATH FAULT MODEL

- PATH FAULT MODEL = considering all possible path

$\forall$ path all possible transitions must be considered

⇓

check timing of the transitions (both rising and falling)

} in this way all faults are detected

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 59 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 61 di 180

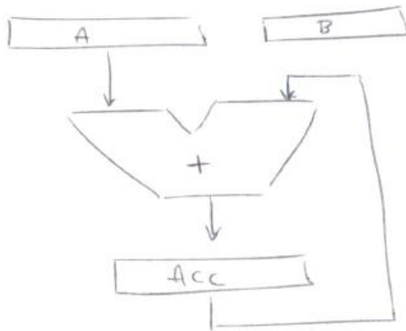
UNTESTABLE DELAY FAULTS

11/13/17

2 types
- **Structurally untestable** = No possible to generate couple of vectors just looking the combinational logic
- **Functionally untestable** = No access to I/O. Faults can't be tested acting on I/O

EX

Considering multiplier A×B using only adders
⇓
A×B = A+A+... (B times)
=



A, B 8 bits
Acc 16 bits

If we would like to consider a delay fault we need to list all patterns from input until output } looking for critical path

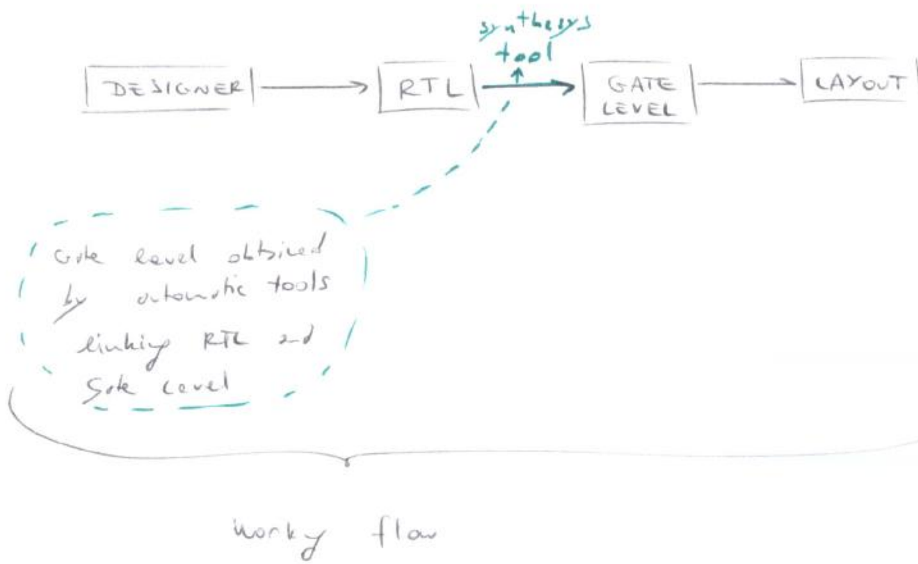
∀ fault need a pair of T.V. ⇒ Needed V_1, V_2 ⇒ Need values for those Test Vectors

Assuming
V_1 = { - - - - - - - - , 00000000 }
V_2 = { - - - - - - - - , 00000001 }

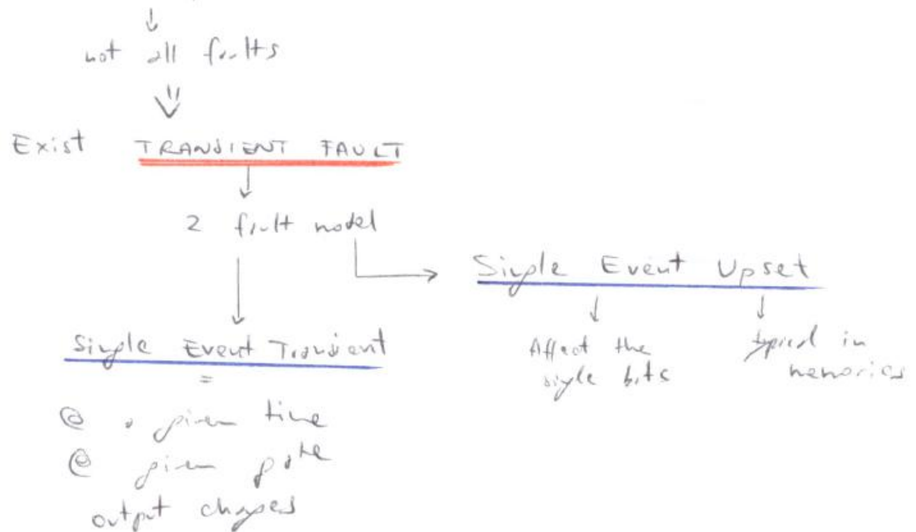
At a given time adder input would pass to 00000000 → 00000001

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 61 di 180

Designer works at the RT level

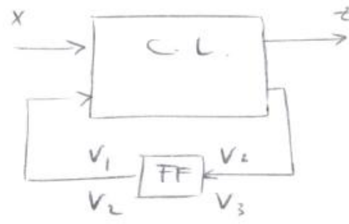


For now, only permanent faults has been discussed



© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 65 di 180

LAUNCH-ON CAPTURE



Procedure

STEP 1: V_1 scanned in and applied to C.L.
⇓
CL produces V_2 as output
↓
stored in FFs

STEP 2: V_2 applied to C.L.
⇓
CL. produces V_3 as output
↓
stored in FF

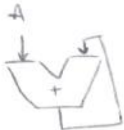
STEP 3: V_3 shifted out

- V_1 freely selected
- V_2 is the response of CL to V_1
  ⇓
  No possible to apply any vector because V_2 depends on V_1, X.

Risk of overtesting due to V_1 freely chosen but risk is lower.

If sure V_1 is reached in normal behavior ⇒ Possible to avoid overtesting

EX



Before shift in a value need it to reach adder inputs

SHIFT-IN V_1 ⇒ TEST MODE ⇒ lower f

CAPTURE ⇒ V_1 applyed to CL ⇒ V_2 produced on output @ NORMAL MODE

CAPTURE II ⇒ V_2 applied to CL ⇒ V_3 produced on output @ NORMAL MODE

SHIFT-OUT ⇒ V_3 shifted-out @ TEST MODE

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 65 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 67 di 180

FUNCTIONAL TEST

11/17/17

= Testing without design for testability just apply test stimuli and check the outputs

2 definitions

① "Test performed on a product without relying on design for testability"
⇓
Using Functional I/O
Depends on
- How to apply the test
- Gray Box model

≠

② "Test based only on the function it is expected to provide"
Black-Box model
Relates to how the test is generated
Used in industries

Roles

→ IN-FIELD TEST
Test performed periodically when the item is used
Based on Definition ②
Ex. Safety Application when faults can anytime

→ INCOMING INSPECTION
Making sure that an item, acquired by third parts, is working before integrate it on the design
⇓
Buying the item ⇒ No idea about the structure

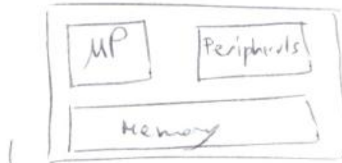
→ END-MANIFACTURING
Testing an item when is just made in the factory
⇓
Test performed by factory who made and designing the item
⇓
Item structure is known

(Incoming inspection and end-manufacturing: done using ATE)

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 67 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 69 di 180

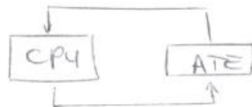
- Assuming a perform functional Test on a SoC
- Functional Test is often used in SoC testing
- A SoC



To perform a test μP execute programs involving Peripherals and Memory and observes the results

Basically a μP realize if there is a fault or not.

SoC Testing @ END MANUFACTURING



Device is on the tester and a certain program is executed

SoC Testing @ IN-FIELD-TEST

In field using a tester is hard ⇒ CPU performs some test program looking for mismatch.

Program is stored in a flash memory

only solution is to run a test program executing the operations producing data stored in memory Then reading the same memory and look for a mismatch.

If a mismatch is recognized ⇒ notification to O.S. making aware of faulty circuit.

working principle of an Software-Based Self-Test

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 69 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 71 di 180

FIFO Testing Steps

STEP 1: INITIALIZE

initial status unknown ⇒ Initialize the FIFO ⇒ leading to a known status

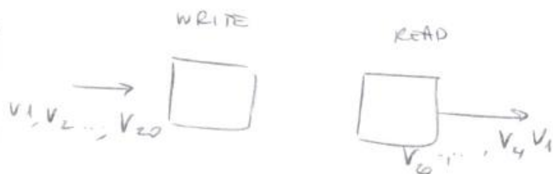
STEP 2: WRITE

Write 20 known values in the buffer or
write values until the FULL signal turns high } write different values

STEP 3: READ

Read 'till EMPTY signal is activated.

STEP 4: CHECK I/O



Need to have

$V_{I_j} = V_{O_j}$ } same output otherwise we have a fault.

In the functional testing there is no blueprint to perform a test.
No guarantee to have the entire fault coverage possible since there is a black box.

Test vectors usually are larger

Testing Time larger

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 71 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 73 di 180

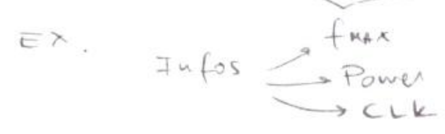
ATE for IC Structure

- testing device when it's placed into the package
- Parts
  - **TEST HEAD** = location where DUT is placed
    - ⇓ takes care of the connections between ATE and DUT.
    - has a **PROBE CARD**
    - need to know the correct pins map to apply the test.
  - **HOST** = user interface
    - executing test user is willing to perform.
    - possible to configure the reaction to fault detection. like stop test or perform diagnosis
  - **LOADING SYSTEM** = system putting the DUT on the Test Head
    - In an industry there are lot of device produced in the line. To be tested a system to place them on the ATE is needed
    - communicate with the probe card to understand the correct position and connection (since Tester are for many devices)
      - ⇓ Probe card should resist to mechanical stress
    - mount the DUT before the test and dismount it after the test is completed
  - **CENTRAL SYSTEM** = stores input stimuli and output
    - stores DUT I/O
    - compare output with the expected ones.

Tester produces set of log files containing test infos and results.

↓

can't contain all the info due to long test = long log files.



© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 73 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 75 di 180

Let's image to test a memory circuit

Memory Test = Perform W/R operations

Memory uses addresses to select where perform the operation

⇓

Need a tester storing the memory address } Feature in tester for memories

Before use the tester is better to test it

Again fault/test inception (yes, like the Movie) = "what if tester is faulty?" "How to test a thing designed to test?"

Solved (or partially) using some **Test operation Sequences** → List

Check V, I on pins in order to check the system = Making sure all pins are connected } = **Steady State Pin Parametrics** (1)

Test correct behavior of the tester before using it } = **Test Logic Verification** (2)

Apply vectors and combinations of → Scan Test → Functional Test } **DC Logic Stuck-at** (3)

Check if memory elements are able to store informations = How much the memory is able to keep those values } **DC Logic Retention** (4)

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 75 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 77 di 180

Test Program = programming the tester in order to perform the wanted operation during the testing phase

↓ needs infos like
- → Start/Stop Time
- → Input values to be applied
- → Expected output values to be applied

} Written using specific languages compatible with CAD tools.

Ex.
STIL = language used to describe vector
STDF = language describing test program.

Pin Timing = synchronize operations according to CLK
↳ specified ∀ pin by test program

In theory CLK is unique and single one ≠ In practice
- CLK enters test unit generating others and different clock signal
- Sometime Tester required the DUT fCLK.

Test Plan = how test is going to evolve

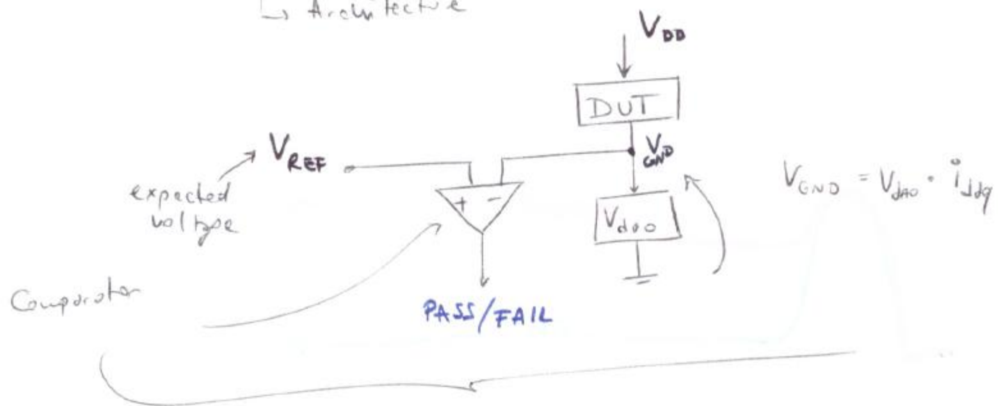
Test complex => Better understand the test phases order deciding the steps to use

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 77 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 79 di 180

Parameters tuning
- → Theshold Current = value selected in interval $1\mu A < I_{ddq} < 20\mu A$ (Depending on experience)
- → Observation Time

To measure Current in Test Phase a good Tester is needed
└ = Built-In Current Sensor
  → Architecture



looking at voltage ⇒ Simpler kind of design for testability

Experimentally HP performs 3 testing type
- → Full Scan
- → I_{ddq} Test
- → Functional Vectors

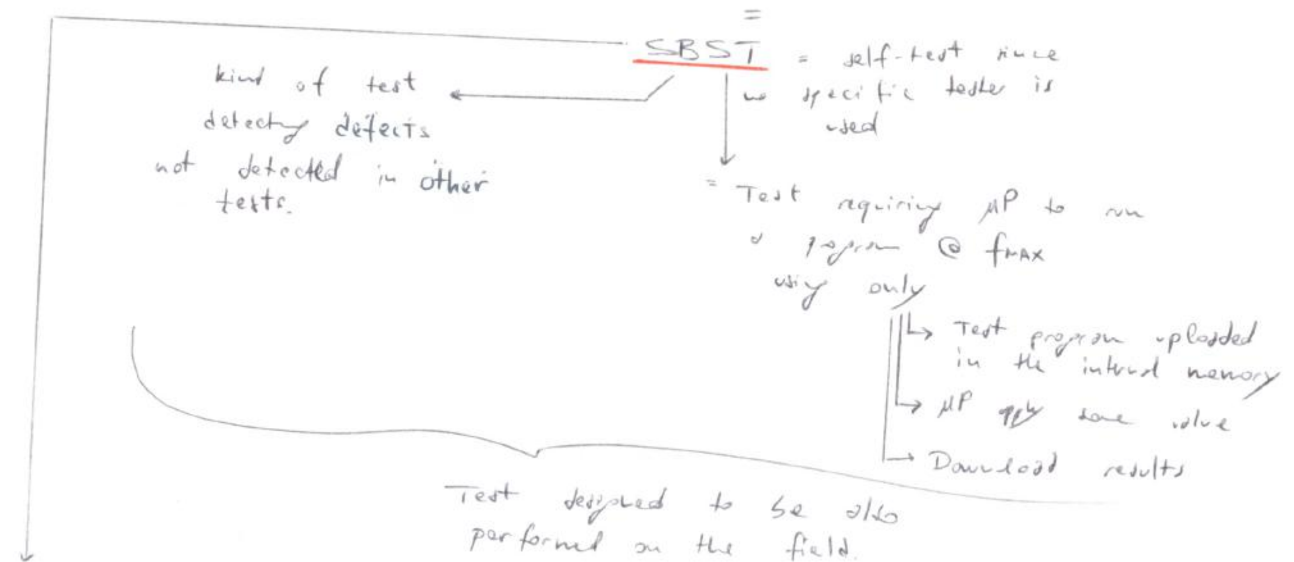
© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 79 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 81 di 180

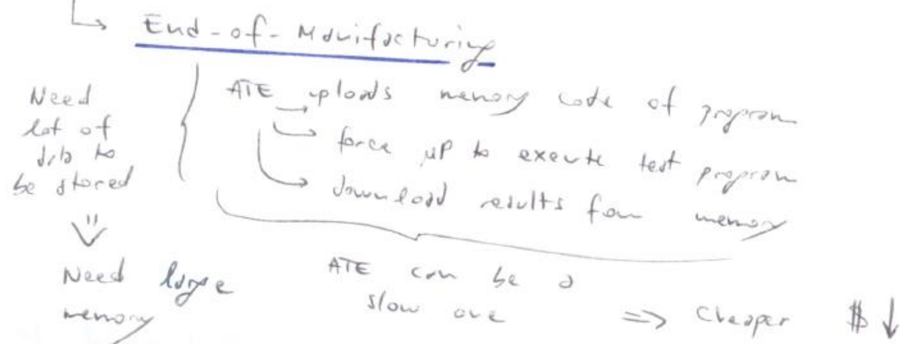
11/20/17

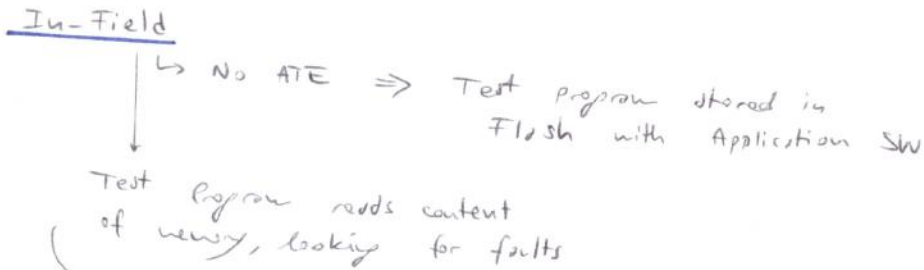
Test μP, SoC using Functional test ⇒ Use the CPU to execute a test program and check if results are ok or not

=



used in 2 scenarios





Test Program is completely autonomous

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 81 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 83 di 180

```
initialize (&signature)
for (i=0; i<256; i++){
    for (j=0; j<256; j++){
        signature = compact(z=i+j);
    }
}
if (signature!= exp_signature){
    error();
}
```

Apply all the possible values for i, j and perform addition.
⇓
Save result in z

store values and compare with the expected one takes TOO MUCH space

Better compute a SIGNATURE
→ ALU returns a value; signature tells if there are faults.
⇓
Just need to check the final signature ≠ Read whole memory for checking values

way to compute:
- EXOR — just perform z EXOR SIGNATURE
- ASSEMBLY

```
int opd1 [N], opd2 [N], res [N];
int adder_test (void){
    int i, x;
    for (i=0; i<N; i++){
        x = opd1 [i] + opd2 [i];
        if ( x != res[i]) {
            ERROR()
        }
}
```

opt1, opt2 two vectors where values to be applied are stored

- Reading values from operand and store results in the memory
- Scan vectors to detect values

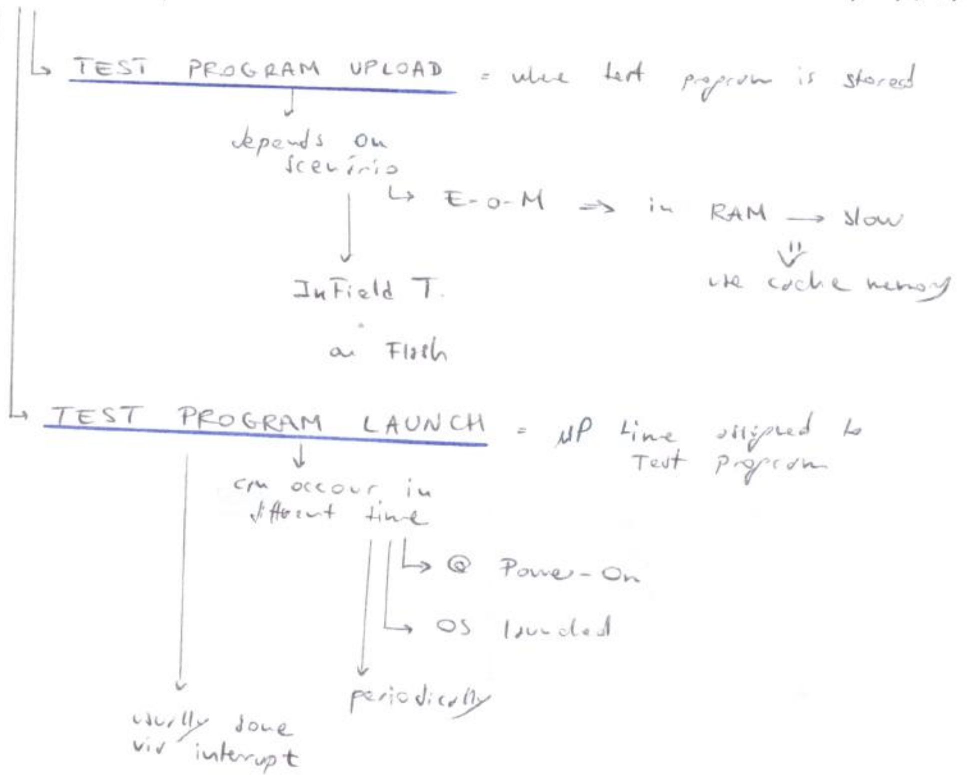
If ALU structure is known ⇒ Isolate ALU and run ATPG ⇒ Test vector to be applied need to be stored in opt1, opt2

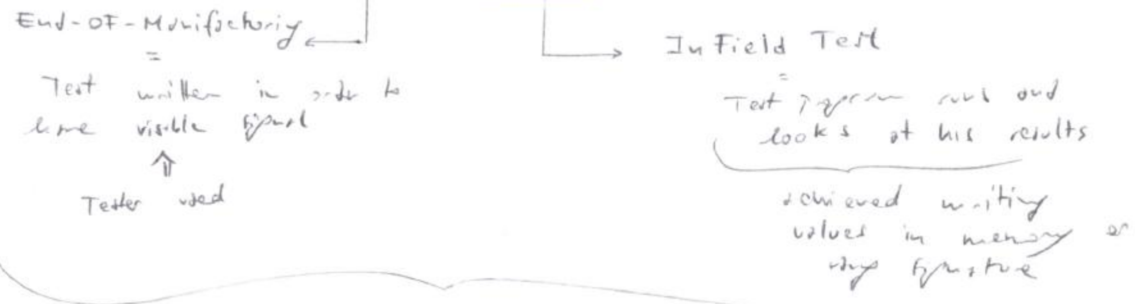
∀ iteration 8 instructions has to be executed ⇒ TEST DURATION = 8 · $N_{VECTORS}$

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 83 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 85 di 180

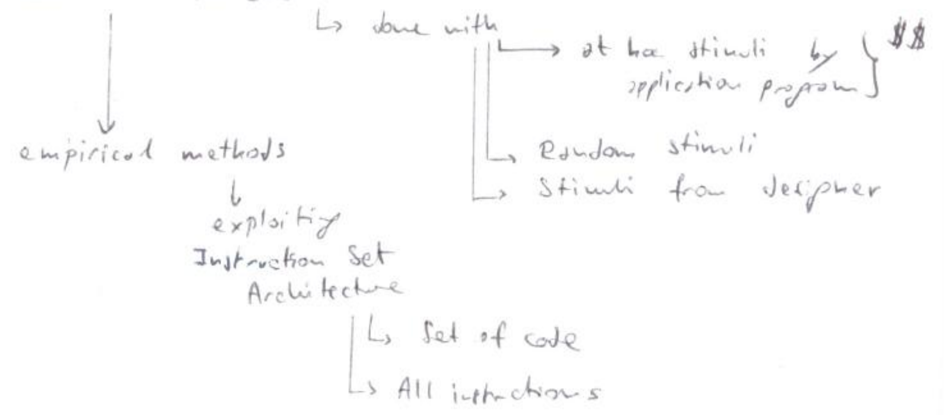
SBST has 3 critical point 11/24/17





All these issues must be solved during test application

Major problem = GENERATING TEST PROGRAM



© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 85 di 180

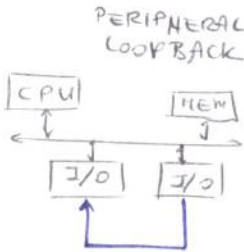
© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 87 di 180

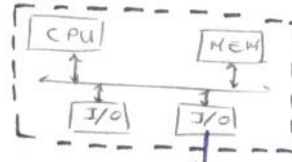
PERIPHERAL

↳ tested using functional test => need to execute/observe peripheral from outside.

2 ways

PERIPHERAL LOOPBACK ← | ↳ ATE





Test without a tester
one I/O generates outputs
while other looks if match
with the corrected one

- $ ↓
- Some stimuli can't be detected

ATE force peripheral to send/receive data

- $ ↑
- Flexible

TEST @ START - testing before the system starts its routine operations

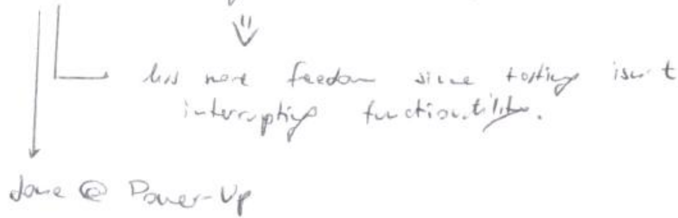
⇓

↳ less more freedom since testing isn't interrupting functionality.

↓

done @ Power-Up

however test should be short in time.



CRITICAL UNITS = are critical in In-Field-Test

↓

list

↳ Address Calculation Unit

∀ memory access μP needs to perform an addition if adder is faulty the access to memory will be wrong.

⇓

Possible to check writing/reading
a portion of memory not reserved
by O.S

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 87 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 89 di 180

11/27/17

BIST Built-In Self Test

= type of design for testability

Embed into the system to test itself without anything from the outside

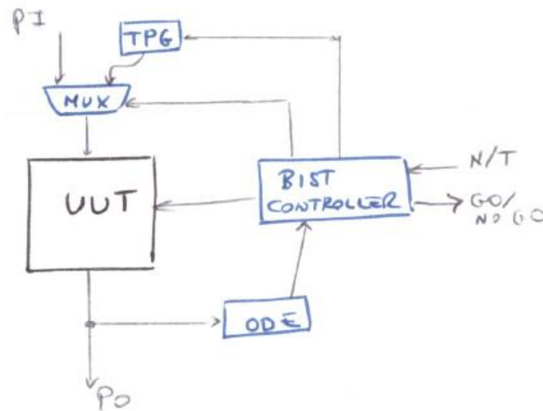
⇓

No need for testers

⇒

Special Circuits setting the whole circuit in "Test-Myself" mode.

↳ System Architecture



TPG = Test Pattern Generation
↳ generate values to perform the test

MUX
↳ select PIs in Normal Mode or T.V. in Test Mode

BIST CONTROLLER
↳ manage different module in both test and normal modes

ODE = Output Data Evaluation
↓
compare the output with the expected values and notify the comparison outcome to BIST CONTROLLER

⇓

Outside testers not needed but just activate test mode ⇐

Assuming UUT is combinational circuit ⇒
- # Input not large ⇒ possible to execute exhaustive test
- Test Pattern Selection is a counter
- ODE computes signature of the output with the expected ones.

↳ Implementation
- ↳ Combinational Circuits
- ↳ Sequential Circuits
- ↳ Memories

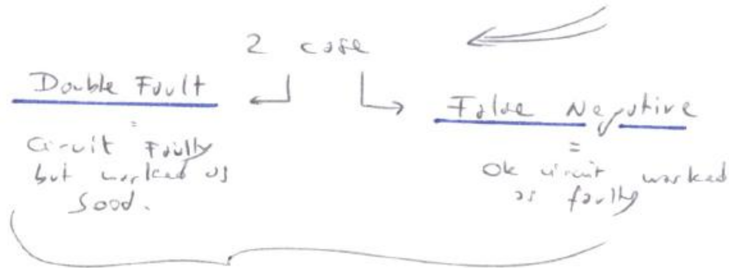
© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 89 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 91 di 180

BIST is based on circuit

⇓

possible faults on it. ⇒ BIST circuit would provide a wrong outcome.



both no good

⇓

Test BIST

↓

Strategies

↳ Duplicate N/T GO/NOGO pins with complementary values.

BIST's Goals

- ↳ Test itself ⇒ Avoid to perform wrong test due to faulty test circuit
- ↳ Minimize
  - ↳ I/O
  - ↳ Test Time
  - ↳ HW.
- ↓ used in InField-Test

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 91 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 93 di 180

Better Describe the behavior using polynomial algebra

Given an $(M+1)$ bit vector $R = [a_m \; a_{m-1} \; \dots \; a_2 \; a_1 \; a_0]$

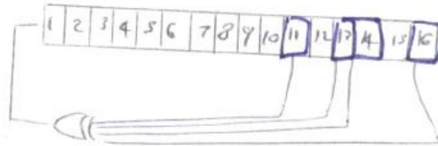
possible to move it into a binary polynomial

$$\begin{cases} P(x) = a_m x^m + a_{m-1} x^{m-1} + \dots + a_1 x + a_0 \\ a_i = \begin{cases} 0 \Rightarrow \text{Open Circuit} \\ 1 \Rightarrow \text{Closed Circuit} \end{cases} \end{cases}$$

possible to create vector depending on where when feedback line exists.

EX

Selecting only $a_{11}, a_{13}, a_{14}, a_{16}$



$\Downarrow$

$P(x) = x^{16} + x^{14} + x^{13} + x^{11} + 1$

TH.

If characteristic polynomial is a primitive polynomial $\Rightarrow$ Possible to generate a LFSR that will evolve through all the states before repeating the sequence

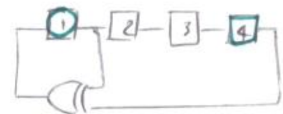
↓

can't be divided in other polynomials

EX

Consider the polynomial of degree 4 $P_{(x)}$

$P_{(x)}: x^{4} + x^{1} + 1 \Rightarrow$

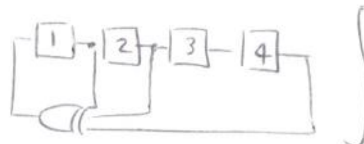


there are many primitives for degree 4

↓

for example

$x^4 + x^2 + x + 1 \Rightarrow$



New connection required = \$\$

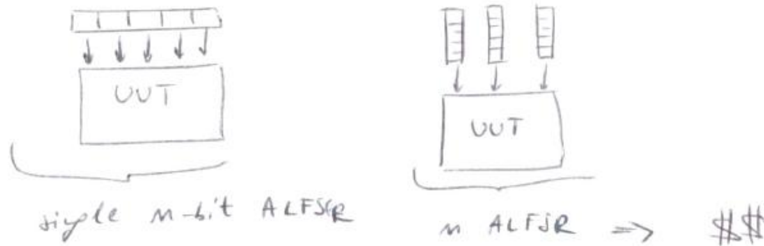
$\Downarrow$

POSSIBLE FOR DEGREE < 20 { Goal is to find high degree requiring few..

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 93 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 95 di 180

ALFS_s used in practice:

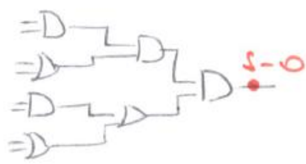


single n-bit ALFSR   n ALFSR => $$

In some cases

Bias the input values => Speed up F.C.

Assuming UUT = Tree gates



← we got a stuck-at-0 I need 1 in all inputs

if using a LFSR need to wait till produce 11...11..

⇐ long time wasted

Better to BIAS the input ⇐

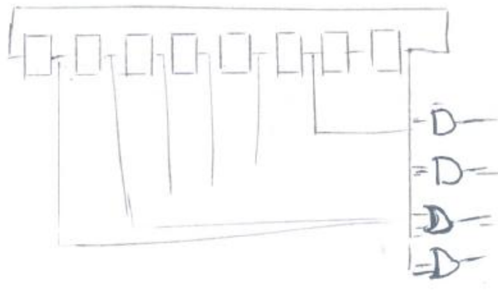
⇓

Need a WEIGHTED (BIASED) RANDOM GENERATOR

↓ faster convergency to ≈ 100%

Biasing probability

Architectures



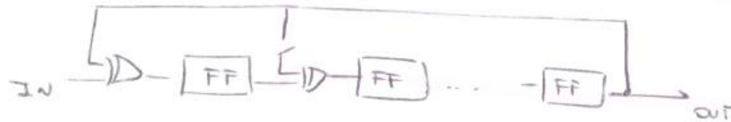
output would be random anyway

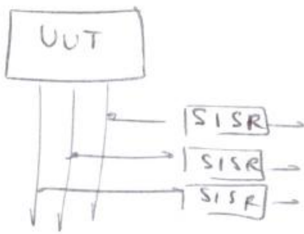
© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 95 di 180

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 97 di 180

ODE implemented using Single Input Shift Register

= LFSR which is similar to modular architecture but EXOR is on input ⇒ Allows to put bit into the circuit





∀ SISR signature is computed

When testing a board ⇒ Apply input stimuli and observe the OUTs

Generally a faulty free board is tested and signature is computed.
↓
used for references.

Low aliasing probability before dell a board

Given 2 input sequences ⇒ ALIASING = P("2 signature are the same")

SISR can be represented using polynomials

Given

$G(x)$ = Input sequence
$Q(x)$ = Output sequence
$P(x)$ = LFSR polynomial

$$\frac{G(x)}{P(x)} = Q(x) + \frac{R(x)}{P(x)}$$

⇒ $R(x)$ = Reminder = Bits in FFs at the end of the experiment.

Goal = find a characteristic polynomial such that reminder of the division is different for the others as much as possible

Better define Polynomial Error $E(x)$

$$E(x) = G'(x) - G(x)$$

$$E(x) = G'(x) - G(x) \Rightarrow R(x) = R'(x) \iff E(x) = m\,P(x)$$

If $P(x)$ is carefully selected ⇒ possible to maximize probability of different signature
↓
MASKING SIGNATURE $P_M = \frac{2^{m-n} - 1}{2^{m-1}}$

© Proprietà riservata dell'autore - Digitalizzazione e distribuzione a cura del CENTRO APPUNTI - Corso Luigi Einaudi, 55 - Torino / Pagina 97 di 180