CSE412
Selected Topics in Computer Engineering
Sheet 1

1. Why is Verification important in the SOC design flow?
   a. It helps the designer to don't care about what he does during design process
   b. It helps ensuring that design meets specification and makes sure that SOC has less bugs.
   c. It is just fun to do
   d. It makes the whole design flow looks complete.

تخيل معايا لو مش هعمل verification, هيبقى مطلوب مني اصنع دايرة على الsilicon
وبعدها شهور بقى في المصانع وعقبال ما تجيلي الchip, وأمسكها واقعد اجرب الfeatures
على الchip شخصيًا.. ده الحل البديل للverification, غير عقلاني صح؟

---

2. What is the most commonly used Hardware description language for Verification in industry?
   a. C++
   b. VHDL
   c. System Verilog
   d. Vera
   e. System C

الC++ كانت بتستخدم زمان في الverification قبل ظهور لغات الـHVL ولكن في النهاية هي مش لغة مخصوصة بالhardware
فكان لازم تظهر لغات تانية معنية اكتر بالverification وبس
الـVHDL والVerilog كانوا بيُستخدموا في الverification بس في الأصل هما لغات HDL
وبعدين ظهرت Vera وهي يعتبر اول لغة HVL ولكنها كانت closed-source ولازم licenses وحوارات

وفي الأخير ظهرت الSystemVerilog وهي نتاج دمج الfeatures الموجودة في الverilog وvera وهي لغة مخصوصة بالأساس للverification واللي منها ظهر الUVM كstandard framework
كل الناس ماشية بيه في الverification حاليًا

---

3. What happens if a critical functional bug escapes the chip design flow into silicon?
   a. It is easy to just fix the bug in silicon just like fixing a software code
   b. A re-spin of chip must be done to fix the bug and fabricate again
   c. Just ignore and continue using the chip

طلما وصلت للsilicon, يبقى مفيش قدامي حل غير إني ارميها وأعيد الflow كله من أول وجديد
بعد ما اصلح الbug دي في الdesign

---

4. What is meant by inheritance in object-oriented programming?
   a. It is the concept that allows a class to instantiate another class as its member (ملهاش اسم)
   b. It is a way in which a class can hide all its properties and attributes being accessed from outside **(Encapsulation)**
   c. It is a concept in which same object can exist in multiple forms **(Polymorephism)**
   d. It is a way in which a class can extend all properties and methods of a base class **(Inheritance)**

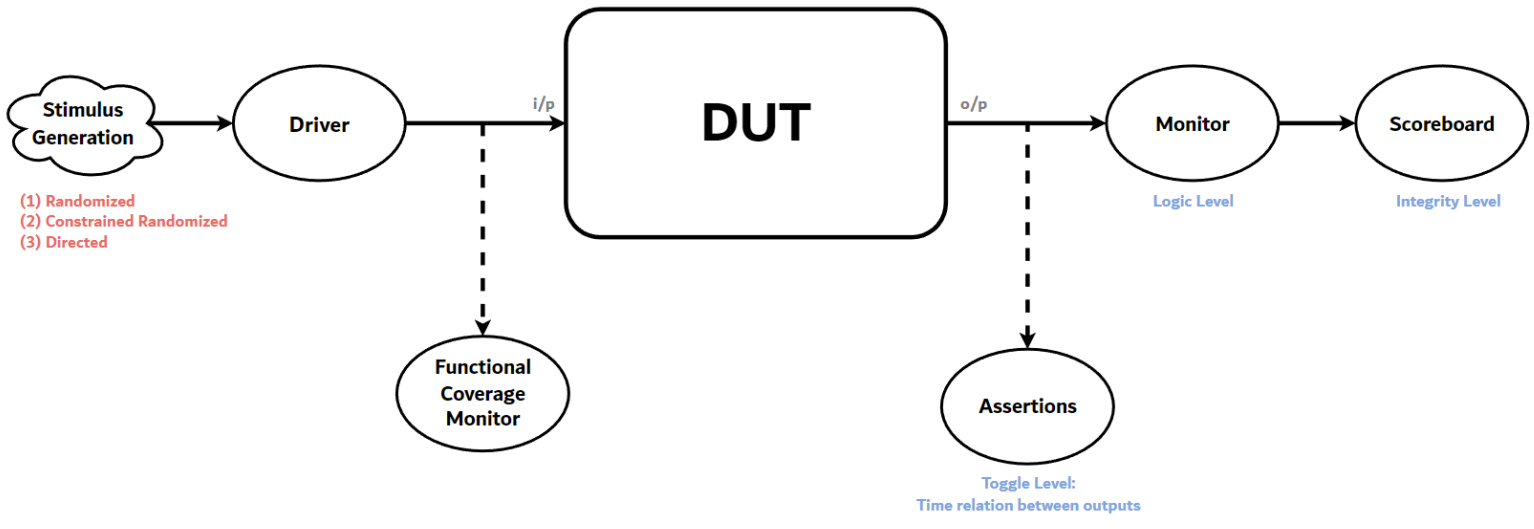
It's a concept involves simplifying complex systems by focusing on the essential characteristics while hiding unnecessary details. (Abstraction)

---

5. Which of the Following is not captured in a verification plan?
   a. Scenarios to be verified for ensuring design correctness
   b. Methodology for stimulus generation for exercising design
   c. Verification effort to complete Verification
   d. Design implementation details like Flip Flops, State machines etc

(Verification Plan)

Verify That	Check That	Testcase/config	Priority	Status	Notes
Scenarios to be verified for ensuring design correctness	Verification effort to complete Verification	Methodology for stimulus generation for exercising design			



6. Which of the following statement related to assertions is not correct?
   a. Assertions are a way to generate stimulus for verification
   b. Assertions helps in debugging and checking of design correctness
   c. Assertions are useful in both Simulation and Formal Verificaiton
   d. Assertions are useful to judge quality of stimulus used for verification

- An Assertion is a statement about a design's intended behavior, which must be verified, or in other words, it is a way of capturing the design intention or part of design specification in the form of a property

- This property can be used along with your normal dynamic simulation or along the formal verification to make sure that specific intention is being met or not met.

---

7. Which of the following is not correct with respect to formal verification?
   a. Formal Verification uses mathematical models and algorithms to exhaustively verify design intent
   b. Formal Verification needs effort in creating stimulus generator for verifying design
   c. Formal Verification on FSM based models to prove FSM state transitions and behavior is known as model checking
   d. Formal Verification can use assertions to capture design intent

- Static verification, or often called formal verification, is a mathematical proof that the two models (design and verification model) are identical under all conditions

- No stimulus is necessary as all possible stimuli are implicitly considered.

- Formal verification methods can be classifies as:

(1) **Equivalence checking:** Check the logical equivalence, or the functional equivalence between say an RTL implementation versus a final version of the netlist.

(2) **Model checking:** Involves exhaustively exploring all possible states of a finite-state model of the system to determine if there is a state that violates the property to be verified.

---

8. What is meant by constrained random verification approach?
   a. It is a method of creating random tests for verifying a design by hoping that **(Random Testing)** random seeds will cover all design state space verification
   b. It is a method of creating intelligent stimulus using constraints around interesting design points that allows faster and thorough testing of design features under different conditions **(Constrained random testing)**
   c. It is a method in which we create tests for each design feature and working conditions and use those for design verification **(Directed Testing)**

(1) **Directed testing:** Can only cover scenarios that are thought in our planning. During the verification planning all scenarios or features that we can think about has to be tested.

(2) **Random testing:** In the pure random test approach we depend on the random test generated, that can randomly hit all the scenarios. So we need to run for an infinite number of states if we want to make sure all the conditions are covered

(3) **Constrained random testing:** It's In between the two approaches, we don't let the generator to be purely random, but we make the generator directed towards some interesting scenarios in the design.

---

9. Which of the following is not true with respect to HW accelerated simulation or emulation?
   a. It improves the simulation speed and allows to run long tests in shorter time during verification
   b. It improves the overall debug capability and observability in the design
   c. It helps to gain higher confidence in system level verification
   d. It can help in early SW validation and reduce risks of finding bugs after chip is manufactured.

- HW emulation: To further improve the speeds, we can do a full mapping of the hardware to the emulator

- The hardware emulation will look more like a real target system. Speedups can reach like 1000x

- Debug is still a challenge on hardware emulations since we may not have a full visibility of the internals of the design.

- Hardware/software co-verification is how to make sure that an embedded system software works correctly with the hardware, and that the hardware has been properly designed to run the software successfully

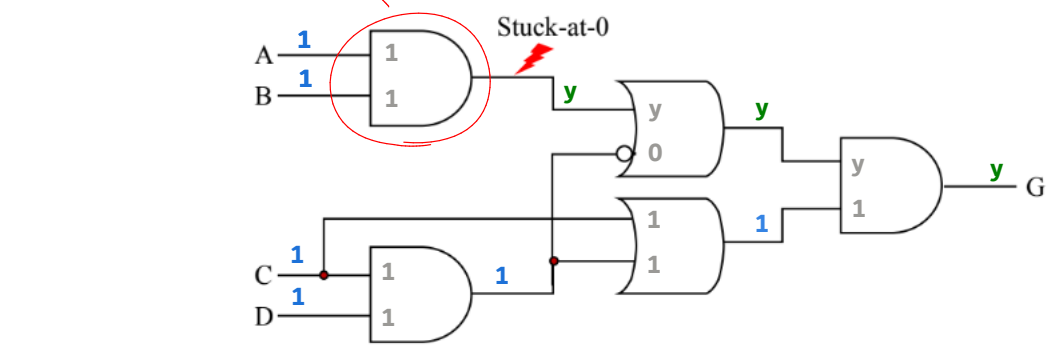
---

10. The following circuit has a stuck-at-0 fault as shown. Find a testing pattern for ABCD such that we can observe the fault at the output G.

عشان اقدر اعرف ان حصلي stuck-at-0 fault, لازم احط test pattern يخلي الnode دي تكون بـ1 كـexpected output, وده اللي أسمه الcontrollability
وفي نفس الوقت انا محتاج اطلع قيمة الnode دي على الoutput عشان اشوف هي صح ولا غلط, وده اللي اسمه observability

كده احنا محتاجين نشوف ازاي نخلي الnode دي تكون بـ1, وايه اللي يخلي قيمة الnode دي هي اللي تحدد قيمة الoutput
لو الnode بـ1 يبقى الoutput بـ1 ولو الnode بـ0 يبقى الoutput بـ0

لازم احط الA والB بـ1
عشان تبقى الnode بـ1



ABCD = 1111 <-- يكون test pattern محتاج احط output عند الnode value الobserve-أقدر عشان

---

11. For a circuit with k lines _____ single stuck-at faults are possible.
a. k b. 2k c. k/2 d. k^2

- Stuck-at-faults: this is the most common fault model used in industry.
- It models manufacturing defects which occurs when a circuit node is shorted to VDD (stuck-at-1 fault) or GND (stuck-at-0 fault) permanently.

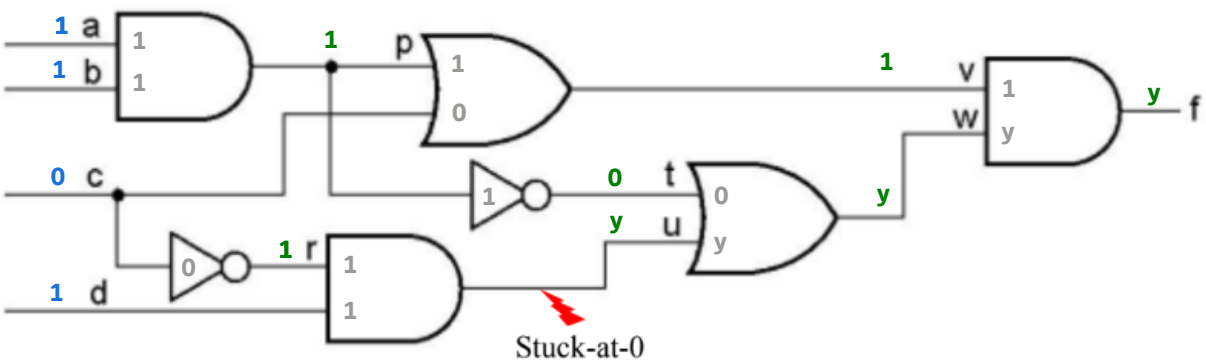
لكل line, فيه اتنين faults يأما الstuck-at-1 يأما الstuck-at-0

---

12. The quality of the test set is measured by
a. fault margin b. fault detection c. fault correction d. fault coverage

---

13. The following circuit has a stuck-at-0 fault as shown. Find a testing pattern for ABCD such that we can observe the fault at the output F.



CSE 313s: Digital Design Verification
ASSIGNMENT 1

المطلوب هو أننا نعمل formal-equivalence checking using a miter circuit, عشان نشوف هل الtwo circuits دول equivalent لبعض ولا لا لأني كان غرضي الoptimization مع الحفاظ على الbehavior الأساسي من غير ما اخربه