Ain Shams University
Faculty of Engineering
Computer and Systems Department
Spring 2023

CSE 313s
Selected Topics in Computer Engineering
Sheet 2

1. Static Analysis involves simulating a model.
   a. True [✓ b. False]

2. Which of the following is a technique covered in Static Analysis?
   a. Formal Verification
   b. Model checking
   c. Equivalence checking
   [✓ d. All of the above]

3. Select the disadvantage of using Model Checking
   a. Concurrent systems cannot be analyzed using this method.
   b. Producing a mathematical specification requires a detailed analysis of the requirements.
   [✓ c. They require the use of specialized notations that can only be understood by domain experts.]
   d. All of the above

4. Which of the following is incorrect with respect to Model Checking?
   a. Model checking is particularly valuable for verifying concurrent systems
   [✓ b. Model checking is computationally very expensive]
   c. The model checker explores all possible paths through the model
   d. All of the above

5. What is a BDD?
   a. Boolean Decision Diagram
   [✓ b. Binary Decision Diagram]
   c. Binary Decision Device
   d. Binary Device Diagram

6. What are the two inputs to the model checker?
   a. Implementation and specification
   b. BDD and FSM
   [✓ c. FSM of the design and properties]
   d. Verification and Validation

7. What is meant by the "counter example" generated by the Model Checker?

   sequence of i/p to reach the error state (property failure stimulus)

8. An escaped bug is a design bug that is not detected during pre-silicon verification, and it is only caught during post-silicon verification.
   a. True
   [✓ b. False]

   post-silicon = TC

9. Formal verification encompasses all techniques that leverage mathematical reasoning and proofs to determine the correctness of a silicon design.
   [✓ a. True]
   b. False

10. Name three of the coverage metrics that we discussed in lectures.

    code coverage: statement, branch, condition, expression, toggle, FSM
    functional coverage

11. Which of the following techniques can be used to prove a general SAFETY PROPERTY?
    a. Equivalence checking
    b. Simulation
    [✓ c. Model Checking]
    d. Emulation

Consider the function: F = c.b + b.(a + d) + a'.d' For the questions below, use variable order (from top to bottom): a,b,c,d.

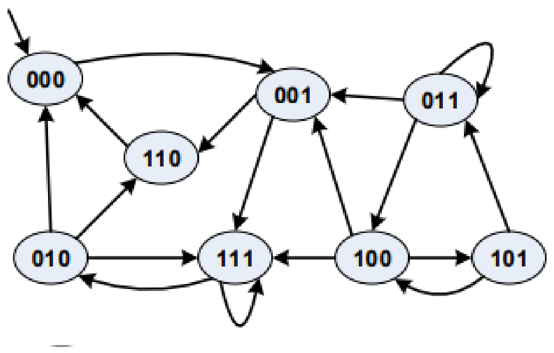
12. What are the cofactors of F w.r.t. a (i.e $F_{a=0}$, $F_{a=1}$)? Provide a simplified expression with the minimum number of literals

    f0= bc +bd +d' f1=b

13. Draw the BDD for the function F. Remember to mark the edge with '0' or '1' to indicate 0 and 1 cofactors. Please draw the '0' edges on the left and the '1' edges on the right.

14. Consider the following function: F (a,b,c,d) = Σ(2,3,12,14). Draw the BDD to represent this function. Remember to mark the edge with '0' or '1' to indicate 0 and 1 cofactors. Please draw the '0' edges on the left and the '1' edges on the right. F=a'b'c + abd'

15. For the following figure, perform a reachability analysis. Are there unreachable states?



16. For the previous problem, is the following property true?
"State 101 can eventually be reached from the initial state"
No

17. What are the two types of assertions, and what are the differences between them?
immediate: evaluated once executed.
concurrent: evaluated using a trigger.

18. What is the type of the following assertion. Explain its meaning.

```
property hash_delay_prop;
  @(posedge prop_clk) req ##5 gnt;
endproperty

hash_delay_check: assert property (hash_delay_prop);
```

concurrent

at each posedge of prop_clk, req must be =1 and after 5 clks gnt =1

19. What is the type of the following assertion. Explain its meaning.

```
assert (grant && request) begin
   $display ("Seems to be working as expected");
 end
else begin
   current_time = $time;
   #1  $error("assert failed at time %0t", current_time);
 end
```

immediate

assert is equivalent to IF

20. Assume in a coverage report, you found the functional coverage = 60% while the code coverage = 100%. What could be the reasons for that? In another experiment you found functional coverage = 100% while the code coverage = 60%. What could be the reasons in that case?

---

first case: executing every statement in code doesn't imply testing all features (for example, some features (scenarios) may only be tested by executing a certain loop for certain number of iterations, however executing the loop for one iteration will achieve statement coverage)
second case: either we have useless code that is not used in any feature (very rare to happen) or it indicates that the functional coverage model is missing some key features of the design(most probably the case).