# CMR INSTITUTE OF TECHNOLOGY

## Internal Assesment Test – II

| Sub: | ADVANCED VLSI | | | | | | | Code: | 21EC71 |
|---|---|---|---|---|---|---|---|---|---|
| Date: | 21.11.2024 | Duration: | 90 mins | Max Marks: | 50 | Sem | VII | Branch | ECE-A,B,C,D |

**Answer Any Five Questions**

| | Questions | Marks | OBE | |
|---|---|---|---|---|
| | | | CO | RBT |
| 1. | Explain the primary goals and objectives of floor planning in VLSI design. Describe the methods used to measure delay in floor planning. How do wire length and capacitance affect signal delay? Provide examples. | [10] | CO2 | L2 |
| 2. | Discuss various floor planning tools available for VLSI design. Compare their features and functionalities, and explain how they contribute to the design process. | [10] | CO2 | L3 |
| 3. | Explain the importance of I/O and power planning in floor planning. How do you approach the placement of I/O pads and the design of a power distribution network? | [10] | CO2 | L2 |
| 4. | Write a note of the verification process involving the goals of hardware design and the verification team. Also, explain the Testing at different levels. | [10] | CO3 | L1 |
| 5. | Explain the following methodology basics: a. constrained random stimulus b. randomization, functional coverage, test bench components, layered test bench | [10] | CO3 | L2 |
| 6. | Discuss the various built-in data types in SystemVerilog. How do the different types (e.g., integer, logic, real, string) differ in terms of their storage and usage in a testbench? Provide examples for each type. | [10] | CO3 | L2 |

CI

CCI

HOD

1) The primary goals and objectives of floor planning in VLSI design involves minimizing area and power consumption

The physical team performs floor planning to place the components in such a way that it takes up less space and routing is easily possible. They also check the power consumption and how to effectively reduce it. It is also important to check the number & location of the input-output ports and the power pads. They will figure out how to efficiently get the work done in least space and minimum power according to the protocols.

(i) Placement of the blocks on the chip.

(ii) Assigning location to the input-output (I/O) ports

(iii) Assinging location and finding out the number of power pads to be used.

(iv) Finding out the power efficiency in the chip.

(V) Clock synthesizing and synchronization.

At the floor planning level, only the fanout & and no. of logic gates driven by the circuit is known to the individual.

Delay can be measuring but verifying the provided clock input signal in the rising (or) falling edge and based on it being synchronized or not, the time where the output is obtained provides us with the delay. There are various automated methods to figure out the delay in the system.

more the capacitance in the circuit less the delay, so less capacitors is a minus.

more the length of the wire of the circuit, more the delay and more the complexity, hence more attenuation and more delay.

# Internal Assessment Test - II

(9) Various floor planning tool available for VLSI design.

There are four types of floor planning tool

1) Cadence innovus
2) Synopsis olympay
3) Silvaco Gateway
4) Silicon SUE
5) Multisim

1) Cadence innovus :-

* Cadence innovus is High level floor planning Tool.
* Cadence innovus is Good in routing mechanism the wiring between the blocks are more flexi
* Cadence innovus is time Synthesis tool for the time optimization.
* Cadence have more feature for the anal -ysation of the planning
* It includes many testing for the routing and the blocks spacing.
* High cost compared to other tools
* High performance between the Area and time synthesis.
* It is used for reduction of complexity of the

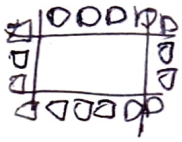
2) Synopsis

* Synopsis is a floor planning tool to design the blocks in the integrated way.
* It is used routing synthesis as a feature of the tool.
* It is more flexible to use all kind of testing in the tool.
* It work on the power consumption of the design for the operation by the floor planning.
* It is similar to the cadence tool
* It is The placing of the components are very much easier.

3) Silvaco Gateway :-

* silvaco Gateway is flexible for the routing of the components
* The Area of the silvaco Gateway is much smaller then the other tool.
* Spply of the power between each and every components can be designed very easily
* The cost is low for the academic compare to other tool.
* not that high performance compare to the other tool.

(3) Importance of I/O and power planning in floor planning.



* Importance of I/O

* I/O is the Input output device which is a Hardware peripheral
* It is a kind of channel between the input and output
* The Communication between the chip and the outside device.
* It play a role of passing information from input to output and output to input.

= power planning :-

* power planning is the hardware description where the supply of the power into the Each and Every required area / block.
* before the power planning the Area planning is important. By analyzing how much the power should be given to each area for the Good Performance.
* power planning is a major step in the any Hardware device.
* In the VLSI the power planning plays a very important role for the better performance.

placement of I/O pads

I/O pads are the pin designed inside the device for the communication between the inside device to the outside device.

I/O pads placement Every chips according to the design of the power and Area

Each and Every blocks are interconnected by the routing

The I/O pads are designed surrounding area of the chip.

The placement is where it should be Easier for the communication for input and output device.

design of power distribution network.

The power distribution is limited when we integrate the circuit very small

for Example :- For 3nm chip the power consumption should be very less because if we supply More power then there is chances of blust of the chip. it would damage the blocks functionality

dangerous

CMR

IAT-2

4] The goals of the hardware team include mainly developing devices ( DVDs, sensors) that perform specific tasks. They make sure the component is programmed to perform the exact operation.

The goal of the verification team is to verify if the device is proper functioning according to the required conditions.

The verification process involves a set of activities to make sure the design is working accurately and effectively.

First, the verification needs to map the design to the HDL ( Hardware Description Language ) and check if it is according to whatever ~~the~~ is required.

They understand the code and the implementation of it block wise.

Once this is done a test bench is written where different test cases are provided as input to check if it meets all the required constraints.

They also find bugs and errors in the code as it will cause problem in the normal functioning of the system.

Once all the test cases are passed and the design matches the specifications accurately, the chip is fabricated and packaged.

The testing at different levels involve low level and high level testing.

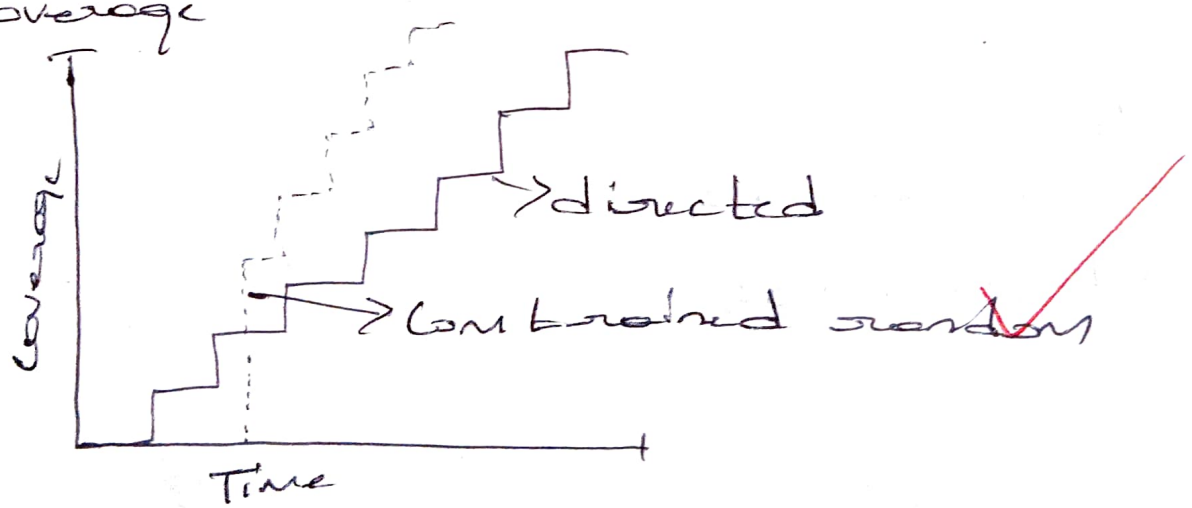
At low level, testing each block and it's functionality is a tedious process. Each block is dependent on the neighbouring block and that makes it a time consuming task to verify and validate. Synthesizing is a big process in low level testing.

On the other hand, high level testing involves testing the entire DUT in one go, so synthesizing becomes an easier task.

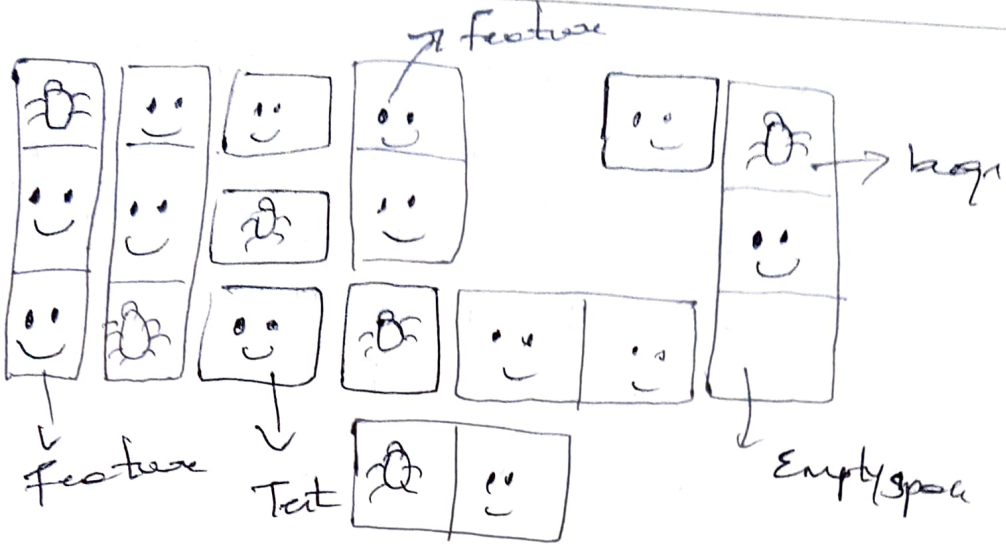
It involves checking the RTL logic implemented by the hardware design team, which may or may not exactly work according to the required logic.

(5) a) Constrained random stimulus

* Constrained random stimulus in the test verification.
* It takes random input for the verification
* Compare to the directed test Constrained random stimulus is easier and fast.
* Constrained random stimulus in not Complex
* It uses many test case to solve the bug present in the design,
* It identify a bug by using the test case to the design.
* It takes random test case for the functional coverage




It do not requires more Time for the verification of the Each test.




b. Randomization :-

* Randomization in which takes the input/functio randomly for the verification of the design.
* Randomization do not take much time for the coverage and verification of the design.

Functional coverage :-

* Functional coverage is the testing of the blocks by using functions or a input

Test bench components :-

* Test bench components uses the test case for the solving of the bugs.
* Uses functionalities like string, integers etc.

## Layered test bench :-

+ There are many layers in the test bench it includes power, Area and time
+ To minimise the bugs in the Each and every blocks we use layered test bench
+ This requires more time for the verification

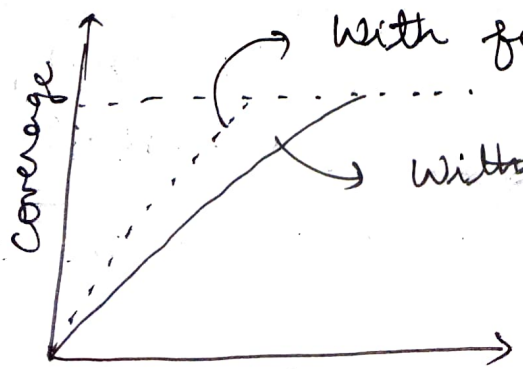
5]

(c) Functional Coverage

In this the functioning of the ASIC based on the code written and the test cases passed.

Here, the code is fed to the ASIC and it is programmed. Then, various unique test cases are created and the logic is checked according to the results obtained each time. These results are noted down and the most accurate one of them is documented. Covering all the test cases and then documenting them is called functional coverage.

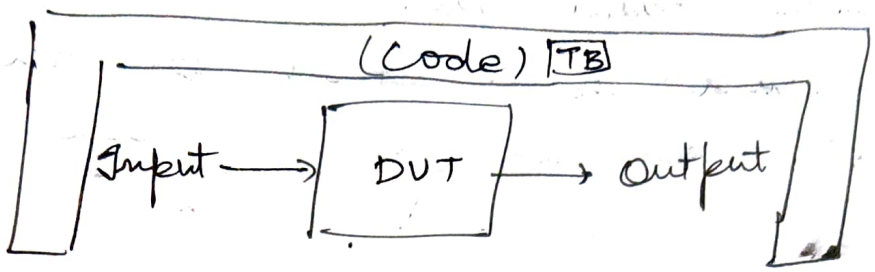
Here, feedback is also added to the testing, so we can narrow down the results sooner. With feedback, we cover everything very quickly as compared to without feedback.



(b) Randomization: This involves giving different inputs in a random manner to check all the results the design is able to provide. A lot of things are randomized when it isn't an ASIC and it's a general purpose chip to check what are the possibilities the chip can provide as results.

(c) Layered Test bench: There are types of test bench of both low level and high level. The test bench involves checking the HDL coded with the logic and obtaining the results. The low level testing involves a lot of work as synthesizing each block's dependency on the neighbouring blocks causes a lot of time consumption. Meanwhile, the high level test bench synchronization is easier and less time consuming.

(d) Test bench components:




The components are inputs; outputs, a code and the DUT.

① The inputs are the unique set of conditions provided to the design with expected outputs. With feedback, this narrows down and a solution will be obtained sooner.

② Outputs are the results provided to the obtained inputs. These output are verified with the provided outputs to check if the functionality of the system is accurate as required.

③ Code (Test bench): This is the entire test of the DUT consisting of inputs outputs, conditions, clock and time intervals. Only after the test bench is implemented and verified, the chip is considered working properly

④ DUT (Design Under Test): This is the design that was created and is being put under test. It is provided with input and the outputs are verified.

6] There are various built in data types in System Verilog.

→ Int → This is for integer values. There are different categories in int

int – 32 bit unsigned.
integer – 32 bit signed.
shortint – 16 bit unsigned.
longint – 64 bit unsigned.

Takes up less space compared to the rest

→ Logic → This is for float values. Including decimal points. It takes up more space than the int.

These are used in test benches while providing inputs and outputs to the DUT.

→ String → This takes a series of numbers / integers

6 * Built-in data types in System verilog is that the data which is present inside the System without any external input

* Implicient is we a Built-in data type in System verilog

* Verilog system use verilog language for the system design.

* There are many built-in data types in the System verilog like module, reg, read, Posedge, negedge, End and many other.

= There are different types differ in term of their storage and usage in a test ben

1) Integer
2) logic
3) real
4) string

All this are different types of storag and usage in a test bench.

short int long ?

1) Integer :-

Integer in which use the digits for the representation of input and output value Integer can be any number for 0 to 9

Example :-

Int_Value = 49;

2) Logic :-

In the logic test bench the operation uses logical function to use for the mathematical functioning. Such as addition, multiplication, division etc.

Example :- operators (+, -, *, %, $, =)

* logic test bench use the digit 0, 1 which the system can understand

```
int_value = 1;
```

(or)

```
int_value = a+b;
```

3) real :-

real test bench uses the floating point operation which uses the decimal point in the operation.

Integer uses floating point to create a real test bench

Example :-

```
float_value = 0.4
Flot_pie = 0.34
```

exmp pro1

) String :-

String in the test bench which uses the alphabets for the representation of the values.

Example :-

```
Std_input = "welcome to VLSI"
```