Sheet 9

1. What is the difference between ignore bins and illegal bins?

Ignore_bins :Used to list the values i don't care about it in my testing coverage "not to add it in my report"
illegal_bins: If occured through an error, (used in very complex constraint, some inputs may make the system made unexpected behaviour, so i want to ensure that my test values, doesn't contain it)

2. What are the two different types of constructs used for functional coverage implementation in System Verilog?

1-coveragegroup covgroup;
endgroup
2-cover property begin end
else begin end
--> it's same as assert property, but the main difference assert if fail --> error (Interested in Fail), while cover property count the succes cases (Interested in success)

3. How can we write a coverpoint to look for transition coverage on an expression?

```
5
6 covergroup covport;
7   C1: coverpoint tr.x ;
8   C2: coverpoint tr.y {//Options}
9 endgroup
```

4. What are the transitions covered by the following coverpoint?

```
coverpoint my_variable {
  bins trans_bin[] = ( a,b,c => x, y);
}
```

trans_bin[0] = (a =>x)
trans_bin[1] = (a =>y)
trans_bin[2] = (b =>x)
trans_bin[3] = (b =>y)
trans_bin[4] = (c =>x)
trans_bin[5] = (c =>y)

if

```
5
6 bins trans_bin = (a,b,c => x,y);
```

any of mentioned there Sucess ✔

5. What does following bin try to cover?

```
covergroup test_cg @(posedge clk);
  coverpoint var_a {
    bin hit_bin = { 3[*4]};
  }
endgroup
```

bins hit_bin = (3=>3=>3=>3);

--> Occurence of the 3 , 4 times

6. What are wildcard bins?

```
5
6 wildcard bins t1 = 4'bxy?1 ;
```

--> means it is interested in only the bit zero to be 1, if the first 3 bits x or y or anything, it is not interested in them

7. What is cross coverage? When is cross coverage useful in Verification?

it's used to find the coverage of cross product of 2 coverpoints, for ex:
--> a = 0,1 & b = 0,1 --> cross a,b check a=0, b=0 & a=0,b=1,.....

8. What are the different ways in which a covergroup can be sampled?

Explicitly using --> xyz.sample()
at the declaration -->

covergroup covport @(ifc.clk);
endgroup

9. What is the difference between coverage per instance and per type? How do we control the same using coverage options?

```
covergroup covport;
  options.instance = 1;
  options.instance = 0; //default value
  C1: coverpoint tr.p
  {
    bins port[] = {[0:$]};
  }
  C2: coverpoint tr.k
  {
    bins low = {[0:7]};
    bins high = {[8:15]};
  }
  C3: cross C1, C2
  {
  }
endgroup
```

```
5
6 covport x1;
7 covport x2;
8 x1 = new();
9 x2 = new();
10
```

--> Since they are 2 instances of same Coveragegroup--> default value (instance Per type) if it has coverpoint a; --> a[0] , a[1],
in Coverage per type, if once at x1 a[0] occurs & once at x2 a[1] occurs, it will result in 100% Coverage

while Coverage per instance it will result in 50% coverage in x1 & 50% in x2