DAY-57
#100DAYSRTL

"System Verilog: Static & Dynamic Arrays"

Introduction:-

An array is several memory addresses grouped. Thus, before entering the data into the array, the size of the array needs to be defined so that the compiler knows how much memory it needs to allocate. But in the case of the dynamic array the size of the array is known only during the run time. This makes it very hard for the simulator to assign continuous memory addresses.

Static Arrays:-

In static arrays, the size of the array is known in the compilation time. This makes it possible for the simulator to ensure that a continuous chunk of memory is allocated for the arrays. Also, as the size is defined in the compile-time, it is not possible to change the size of the array later on during run-time. Thus these are known as static arrays.

Syntax:-

- `<datatype> [<packed_arr_size>] <identifier> [<unpacked_arr_sizes>]`
- `<datatype> <identifier> [<unpacked_arr_sizes>]`

Code Practising:-

```
module tb;
  bit [7:0] arr1[2];
  bit [7:0] arr2[2];
  initial begin
    arr1='{8'b10,8'b11};
    arr2[0]=8'b10;
    $display("arr1=%p",arr1);
    $display("arr2=%p",arr2);
  end
endmodule
```

Result:-

```
arr1='{2, 3}
arr2='{2, 0}
```

Dynamic Arrays:-

In dynamic arrays, the size of the array is known only in the run-time, which makes it difficult for the simulator to ensure that a continuous chunk of memory is allocated to the array. The size of the array can be easily changed during the run time thus giving it the name dynamic arrays

Syntax:-

- `<datatype> <identifier>[] = '{<elements of array>};`
- `<datatype> <identifier>[];`
- `<identifier>[] = new [<size of array>];`

Note :-

Packed array can not be a dynamic array. As discussed earlier packed arrays must have a continuous memory chunk which is not ensured in dynamic arrays. Also if a mixed array is declared only the unpacked part of the array can be dynamic.

- bit [] arr ; // **illegal.**
- bit [7:0] arr[]; //**Legal**

Code Practising:-

```
module tb;
  int arr1[];
  int  arr2[];
  initial begin
    arr1='{1,2,3};
    $display("arr1=%p",arr1);
    $display("Size of arr1=%d",arr1.size());//Size method
    arr2=arr1; //Copy method
    $display("arr2=%p",arr2);
    $display("Size of arr2=%d",arr2.size());
    arr2=new[6](arr1);
    $display("arr2=%p",arr2);
    $display("Size of arr2=%d",arr2.size());
    arr2=new[4];
    $display("arr2=%p",arr2);
    $display("Size of arr2=%d",arr2.size());
    arr2.delete();
    $display("arr2=%p",arr2);
    $display("Size of arr2=%d",arr2.size());
  end
endmodule
```

Result:-

```
arr1='{1, 2, 3}
Size of arr1=           3
arr2='{1, 2, 3}
Size of arr2=           3
arr2='{1, 2, 3, 0, 0, 0}
Size of arr2=           6
arr2='{0, 0, 0, 0}
Size of arr2=           4
arr2='{}
Size of arr2=           0
```

Static Vs Dynamic Array:-

Static	Dynamic
1. In static arrays the size of the array is fixed during the entire simulation.	1. The Size of the dynamic array can be changed during the simulation.
2. There is lot of wastage of memory as the size of array is considered keeping in mind the maximum possible elements that can be stored. It is not necessary that we use all the location in simulation.	2. As the size can be changed dynamically thus there is no wastage of memory.
3. It is faster as the memory allocation is done during compile time.	3. It may not be efficient as memory allocation is done during the run-time and the size can change during simulation. Also it is not guaranteed that continous memory block will be allocated slowing the access to array.
4. Can be used to declare both packed and unpacked arrays.	4. Can be used to declare only unpacked arrays.