

Week 19 Assignment

1. Write a program in the system Verilog to store elements in an array and print it.

```
1 typedef class Array;
2
3 module store_elements_array;
4     Array arr_obj;
5
6     initial begin
7         //Using class object
8         arr_obj = new();
9         arr_obj.setArrayElements();
10        arr_obj.displayArrayElements();
11
12        //Using Scope resolution operator
13        /*Array::setArrayElements();
14        Array::displayArrayElements();*/
15    end
16 endmodule
17
18 class Array;
19
20     int array[10];
21
22     function void setArrayElements();
23         foreach (array[i]) begin
24             if(i == 0)
25                 array[i] = 1;
26             else
27                 array[i] = i;
28         end
29     endfunction
30
31     function void displayArrayElements();
32         $display("\n");
33         $write("Elements in array are : ");
34         foreach (array[i])
35             $write("%0d ",array[i]);
36         $display("\n");
37     endfunction
38
39 endclass
```

Output:

```
Elements in array are : 1 1 2 3 4 5 6 7 8 9
```

2. Write a program in the system Verilog to read n number of values in an array and display it in reverse order.

```
1 typedef class Array;
2 //typedef int integer_array[];
3
4 module read_reverse_array;
5     Array #(5) arr;
6     initial begin
7         arr = new();
8         arr.setArrayValues();
9         arr.displayArrayValues("Array values before reversing : ");
10        arr.reverseArray();
11        arr.displayArrayValues("Array values after reversing : ");
12    end
13 endmodule
14
15 class Array #(parameter size_of_array=10);
16
17     int array[];
18     //int reversed_array [];
19
20     function void setArrayValues();
21         array = new [size_of_array];
22         foreach (array[i])
23             array[i] = $urandom_range(1,10);
24     endfunction
25
26     function void reverseArray();
27         /*reversed_array = new[array.size()];
28         foreach (array[i])
29             reversed_array[array.size() - 1 - i] = array[i];*/
30         array.reverse();
31     endfunction
32
33     function void displayArrayValues(string str);
34         $display();
35         $write("%s",str);
36         foreach (array[i])
37             $write("%0d ",array[i]);
38         $display();
39     endfunction
40
41 endclass
```

Output:

```
Array values before reversing : 9 1 4 1 8
Array values after reversing : 8 1 4 1 9
```

3. Write a program in system Verilog to find the sum of all elements of the array.

```
1 typedef class Array;
2
3 module sum_of_elements_array;
4     Array #(5) arr;
5     initial begin
6         arr = new();
7         arr.setArrayValues();
8         arr.displayArrayValues("Elements of array are : ");
9         $display("The sum of elements of array is %0d",arr.calculate_sum());
10    end
11 endmodule
12
13 class Array #(parameter size_of_array=10);
14     int array[];
15     int sum;
16
17     function void setArrayValues();
18         array = new [size_of_array];
19         foreach (array[i])
20             array[i] = $urandom_range(1,10);
21     endfunction
22
23     function int calculate_sum();
24         foreach (array[i])
25             sum = sum + array[i];
26         return sum;
27     endfunction
28
29     function void displayArrayValues(string str);
30         $display();
31         $write("%s",str);
32         foreach (array[i])
33             $write("%0d ",array[i]);
34         $display();
35     endfunction
36
37
38 endclass
```

Output:

```
Elements of array are : 9 1 4 1 8
The sum of elements of array is 23
```

4. Write a program in the system Verilog to copy the elements of one array into another array.

```
1 typedef class Array;
2
3 module copy_elements;
4     int array_1[];
5     int array_2[];
6
7     Array arr_obj;
8
9     initial begin
10         array_1 = new[5];
11         array_2 = new[5];
12         arr_obj = new();
13
14         array_1 = '{65,87,53,24,99};
15         arr_obj.copy_array_values(array_1,array_2);
16         arr_obj.display(array_1,"Array - 1");
17         arr_obj.display(array_2,"Array - 2");
18     end
19 endmodule
20
21
22 class Array;
23
24     task copy_array_values(input int arr_1[], output int arr_2[]);
25         //$display("-----%p",arr_1);
26         arr_2 = new[arr_1.size()];
27         foreach (arr_1[i])
28             arr_2[i] = arr_1[i];
29         //$display("-----%p",arr_2);
30     endtask
31
32     function void display(int arr[], string str);
33         $write("Elements of %s are : ",str);
34         foreach (arr[i])
35             $write("%0d ",arr[i]);
36         $display();
37     endfunction
38
39 endclass
```

Output:

```
Elements of Array - 1 are : 65 87 53 24 99
Elements of Array - 2 are : 65 87 53 24 99
```

5. Write a program in system Verilog to count a total number of duplicate elements in an array.

```
1 typedef class Array;
2
3 module duplicate_elements;
4
5     int array[] = '{1,1,1,2,2,3,4,5,5,5,5};
6     Array arr_obj;
7
8     initial begin
9         arr_obj = new();
10        $display("\nArray elements are : %p",array);
11        $display("Count of duplicate elements of array is : %0d\n",arr_obj.findDuplicateElementsCount(array));
12    end
13
14 endmodule
15
16 class Array;
17     function int findDuplicateElementsCount(int arr[]);
18         int unique_arr_size;
19         int original_size;
20         int unique_elements[$];
21         int checker[$];
22
23         original_size = arr.size();
24         //unique_elements = arr.unique();
25         foreach (arr[i]) begin
26             checker = unique_elements.find_first with (item == arr[i]);
27             if(checker.size()==0)
28                 unique_elements.push_front(arr[i]);
29         end
30         $display("Unique elements ----- %p",unique_elements);
31         unique_arr_size = unique_elements.size();
32         return (original_size - unique_arr_size);
33     endfunction
34 endclass
```

Output:

```
Array elements are : '{1, 1, 1, 2, 2, 3, 4, 5, 5, 5, 5}
Unique elements ----- '{5, 4, 3, 2, 1}
Count of duplicate elements of array is : 6
```

6. Write a program in system Verilog to print all unique elements in an array.

```
1 typedef class Array;
2
3 module unique_elements;
4     Array arr_obj;
5     int arr[9] = '{1,1,2,2,3,4,5,6,6};
6     int arr_unique[$];
7     initial begin
8         arr_obj = new();
9         arr_obj.display(arr);
10
11         arr_unique = arr_obj.unique_elements(arr);
12         $display("The unique elements found in the array are : ");
13         arr_obj.display(arr_unique);
14     end
15 endmodule
16
17 class Array;
18     typedef int que[$];
19
20     function void display(int arr[]);
21         foreach (arr[i])
22             $display("arr[%0d] = %0d",i,arr[i]);
23         $display();
24     endfunction
25
26     function que unique_elements(int arr[]);
27         int arr_unique[$];
28         int arr_partially_unique[$];
29         int arr_element_frequency[int];
30         int count = 0;
31         arr_partially_unique = arr.unique();
32
33         foreach (arr_partially_unique[i]) begin
34             count = 0;
35             foreach (arr[j]) begin
36                 if(arr[j] == arr_partially_unique[i])
37                     count++;
38             end
39             arr_element_frequency[arr_partially_unique[i]] = count;
40         end
41
42         foreach (arr_element_frequency[i]) begin
43             $display("-----%d %d",i,arr_element_frequency[i]);
44             arr_unique = arr_element_frequency.find_index with (item == 1);
45         end
46         return arr_unique;
47     endfunction
48 endclass
```

Output:

```
arr[0] = 1
arr[1] = 1
arr[2] = 2
arr[3] = 2
arr[4] = 3
arr[5] = 4
arr[6] = 5
arr[7] = 6
arr[8] = 6

-----      1      2
-----      2      2
-----      3      1
-----      4      1
-----      5      1
-----      6      2
The unique elements found in the array are :
arr[0] = 3
arr[1] = 4
arr[2] = 5
```

7. Write a program in system Verilog to merge two arrays of same size sorted in descending order.

```
1 typedef class Array;
2
3 module merge_descending_array;
4
5     Array obj = new();
6     int array_1[5] = '{1,2,3,8,9};
7     int array_2[4] = '{5,1,4,2};
8
9     int arr_out[$];
10
11     initial begin
12         $display("-----Array 1-----");
13         obj.dispaly(array_1);
14         $display("-----Array 2-----");
15         obj.dispaly(array_2);
16
17         arr_out = obj.merge_and_sort(array_1,array_2);
18
19         $display("-----Merged Array in Descending order-----");
20         obj.dispaly(arr_out);
21     end
22 endmodule
23
24 class Array;
25     typedef int que[$];
26
27     function void dispaly(int arr[]);
28         foreach (arr[i])
29             $display("arr[%0d] = %0d",i,arr[i]);
30     endfunction
31
32     function que merge_and_sort(int arr1[], arr2[]);
33         int out[$];
34
35         out = {arr1,arr2};
36         out.rsort();
37
38         return out;
39     endfunction
40 endclass
```

Output:

```
-----Array 1-----
arr[0] = 1
arr[1] = 2
arr[2] = 3
arr[3] = 8
arr[4] = 9
-----Array 2-----
arr[0] = 5
arr[1] = 1
arr[2] = 4
arr[3] = 2
-----Merged Array in Descending order-----
arr[0] = 9
arr[1] = 8
arr[2] = 5
arr[3] = 4
arr[4] = 3
arr[5] = 2
arr[6] = 2
arr[7] = 1
arr[8] = 1
```

8. Write a program in system Verilog to count the frequency of each element of an array.

```
1 typedef class Array;
2 typedef int que[int];
3
4 module test;
5     Array obj = new();
6     int array[10] = '{1,2,3,4,15,3,1,9,4,5};
7     que out;
8     initial begin
9         $display("Original Array");
10        obj.display(array);
11        out = obj.find_frequency(array);
12        $display("\nArray Element Frequency");
13        obj.display_out(out);
14    end
15 endmodule
16
17 class Array;
18     function void display(int arr[]);
19         foreach (arr[i])
20             $display("arr[%0d] = %0d",i,arr[i]);
21     endfunction
22
23     function void display_out(int arr[int]);
24         foreach (arr[i])
25             $display("arr[%0d] = %0d",i,arr[i]);
26     endfunction
27
28     function que find_frequency(int arr[]);
29         que out;
30         int unique_elements[$];
31         int count;
32         unique_elements = arr.unique();
33
34         foreach (unique_elements[i]) begin
35             count = 0;
36             foreach (arr[j])
37                 if(arr[j] == unique_elements[i])
38                     count++;
39             out[unique_elements[i]] = count;
40         end
41         return out;
42     endfunction
43 endclass
```

Output:

```
Original Array
arr[0] = 1
arr[1] = 2
arr[2] = 3
arr[3] = 4
arr[4] = 15
arr[5] = 3
arr[6] = 1
arr[7] = 9
arr[8] = 4
arr[9] = 5

Array Element Frequency
arr[1] = 2
arr[2] = 1
arr[3] = 2
arr[4] = 2
arr[5] = 1
arr[9] = 1
arr[15] = 1
```


9. Write a program in system Verilog to find the maximum and minimum element in an array.

```
1 typedef class Array;
2
3 module test;
4     int arr[10] = '{9,12,70,23,78,56,25,80,87,100};
5     Array obj = new();
6
7     initial begin
8         obj.display(arr);
9         $display("The minimum element in the array is %0d",obj.minimum(arr));
10        $display("The maximum element in the array is %0d",obj.maximum(arr));
11    end
12 endmodule
13
14 class Array;
15     function void display(int a[]);
16         foreach (a[i])
17             $display("array[%0d] = %0d",i,a[i]);
18     endfunction
19
20     function int minimum(int a[]);
21         int q[$] = a.min();
22         int min = q.pop_back();
23         return min;
24     endfunction
25
26     function int maximum(int a[]);
27         int q[$] = a.max();
28         int max = q.pop_front();
29         return max;
30     endfunction
31
32 endclass
```

Output:

```
array[0] = 9
array[1] = 12
array[2] = 70
array[3] = 23
array[4] = 78
array[5] = 56
array[6] = 25
array[7] = 80
array[8] = 87
array[9] = 100
The minimum element in the array is 9
The maximum element in the array is 100
```

10. Write a program in system Verilog to separate odd and even integers in separate arrays.

```
1 typedef class Array;
2 typedef int que[$];
3
4 module test;
5     int arr[10] = '{9,12,70,23,78,56,25,80,87,100};
6     Array obj = new();
7     que even_elements, odd_elements;
8
9     initial begin
10         obj.display(arr);
11         even_elements = obj.find_even_elements(arr);
12         odd_elements = obj.find_odd_elements(arr);
13         $display("The even elements in the array are : %0p",even_elements);
14         $display("The odd elements in the array are : %0p",odd_elements);
15     end
16 endmodule
17
18 class Array;
19     function void display(int a[]);
20         foreach (a[i])
21             $display("array[%0d] = %0d",i,a[i]);
22     endfunction
23
24     function que find_even_elements(int a[]);
25         que out = a.find with (item%2 == 0);
26         return out;
27     endfunction
28
29     function que find_odd_elements(int a[]);
30         que out = a.find(x) with (x%2 == 1);
31         return out;
32     endfunction
33
34 endclass
```

Output:

```
array[0] = 9
array[1] = 12
array[2] = 70
array[3] = 23
array[4] = 78
array[5] = 56
array[6] = 25
array[7] = 80
array[8] = 87
array[9] = 100
The even elements in the array are : '{12, 70, 78, 56, 80, 100}
The odd elements in the array are : '{9, 23, 25, 87}
```