Experiment _1 - Introduction to data types and arrays

Part 1:

Aim:

Write the SystemVerilog code to:

i. Declare a 2-state array, my_array that holds four 12-bit values

ii. Initialize my_array so that:

my_array [0] = 12'h012

my_array [1] = 12’h345

my_array [2] = 12'h678

my_array [3] = 12’h9AB

iii. Traverse my_array and print out bits [5:4] of each 12-bit element

a. With a for loop

b. With a foreach loop

Code:

```
module array_methods();
  bit [11:0] my_array[4];

  initial begin
    my_array = '{12'h12,12'h345,12'h678,12'h9AB}; // array initialization

    $display("\nUsing for loop");
    for(int i=0; i<$size(my_array); i++)
      $displayh("my_array(%0d)=",i,my_array[i]); // $displayh displays hex values

    $display("\nUsing foreach loop");
    foreach(my_array[i])
      $displayh("my_array(%0d)=",i,my_array[i]);

    $display("\n\nvalues of bits [5:4]");
    $display("\nUsing for loop");
    for(int i=0; i<$size(my_array); i++)
      $display("my_array(%0d)=%b",i,my_array[i][5:4]); // prints the subset

    $display("\nUsing foreach loop");
    foreach(my_array[i])
      $display("my_array(%0d)=%b",i,my_array[i][5:4]);

    $display("\n\nAdding 4 to each value");
    $display("\nUsing for loop");
    for(int i=0; i<$size(my_array); i++) begin
```

```
      my_array[i] += 4; //add 4 to each value
      $displayh("my_array(%0d)=",i,my_array[i]);
    end

    my_array = '{12'h12,12'h345,12'h678,12'h9AB};
    $display("\nUsing foreach loop");
    foreach(my_array[i]) begin
      my_array[i] += 4; //my_array[i] = my_array[i] + 4
      $displayh("my_array(%0d)=",i,my_array[i]);
    end
  end
endmodule
```

Output:

```
# KERNEL: Using for loop
# KERNEL: my_array(0)=012
# KERNEL: my_array(1)=345
# KERNEL: my_array(2)=678
# KERNEL: my_array(3)=9ab
# KERNEL:
# KERNEL: Using foreach loop
# KERNEL: my_array(0)=012
# KERNEL: my_array(1)=345
# KERNEL: my_array(2)=678
# KERNEL: my_array(3)=9ab
# KERNEL:
# KERNEL:
# KERNEL: values of bits [5:4]
# KERNEL:
# KERNEL: Using for loop
# KERNEL: my_array(0)=01
# KERNEL: my_array(1)=00
# KERNEL: my_array(2)=11
# KERNEL: my_array(3)=10
```

```
# KERNEL: Using foreach loop
# KERNEL: my_array(0)=01
# KERNEL: my_array(1)=00
# KERNEL: my_array(2)=11
# KERNEL: my_array(3)=10
# KERNEL:
# KERNEL:
# KERNEL: Adding 4 to each value
# KERNEL:
# KERNEL: Using for loop
# KERNEL: my_array(0)=016
# KERNEL: my_array(1)=349
# KERNEL: my_array(2)=67c
# KERNEL: my_array(3)=9af
# KERNEL:
# KERNEL: Using foreach loop
# KERNEL: my_array(0)=016
# KERNEL: my_array(1)=349
# KERNEL: my_array(2)=67c
# KERNEL: my_array(3)=9af
# KERNEL: Simulation has finished.
```

Part 2:

Aim:

Write the SystemVerilog code to:

i. Declare a 2-state two dimensional array, array_2d (4 rows and 3 columns), that holds 12 integer values.

ii. Initialize my_array so that:

array_2d[0][0] = 0
array_2d[0][1] = 1
array_2d[0][2] = 2
array_2d[1][0] = 3
array_2d[1][1] = 4
array_2d[1][2] = 5
array_2d[2][0] = 6
array_2d[2][1] = 7
array_2d[2][2] = 8
array_2d[3][0] = 19
array_2d[3][1] = 20
array_2d[3][2] = 21

iii. Print out the values stored in array_2d and verify them with initialized values.

Code:

```
module array_2dim();
  bit [11:0] array_2d[4][3] = '{'{0,1,2},'{3,4,5},'{6,7,8},'{19,20,21}};
  initial foreach(array_2d[i])
      $display("Row %0d of array is",i,array_2d[i][0],array_2d[i][1],array_2d[i][2]);
endmodule
```

Output:

```
# KERNEL: Row 0 of array is    0   1   2
# KERNEL: Row 1 of array is    3   4   5
# KERNEL: Row 2 of array is    6   7   8
# KERNEL: Row 3 of array is   19  20  21
```

Result:

The given problem statement is executed and verified to be correct.