**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

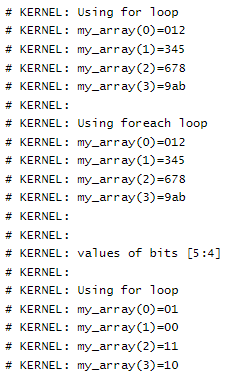
      $displayh("my\_array(%0d)=",i,my\_array[i]);

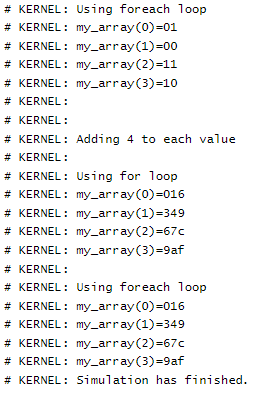
    end

  end

endmodule

**Output:**

****

****

**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();

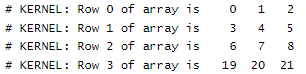
  bit [11:0] array\_2d[4][3] = '{'{0,1,2},'{3,4,5},'{6,7,8},'{19,20,21}};

  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:**

****

**Result:**

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