DAY-58
#100DAYSRTL

"System Verilog: Associative Arrays"

Introduction:-

Associative arrays are an extension of dynamic arrays which provide more flexibility in the way in which data can be stored. In associative arrays, the index can be of any data type including strings which makes it very beneficial for certain scenarios.

Associative Arrays:-

- Associative arrays are a special type of dynamic arrays in which the memory is not allocated immediately when the array is declared. Instead, the memory is allocated as and when data is stored in the array. As the memory is not allocated immediately the allocated memory is not continuous, which makes it slower than dynamic arrays.
- This can be compared to key: value pairs in another language. The index can be used as a key to get the value stored. The index can be referred to as a key to get the value stored.
- Also, the key data type should not necessarily be an int, but it can be any data type including string or real

Syntax:-

- `[value datatype] [arr_name] [index datatype];`

Code Practising:-

```
module tb;
  int VLSI [string];
  initial begin
    VLSI='{"RTL":1,"DV":2};
    $display("VLSI:-%p",VLSI);
    $display("VLSI_RTL:-%d",VLSI["RTL"]);
  end
endmodule
```

Result:-

```
VLSI:-'{"DV":2, "RTL":1}
VLSI_RTL:-          1
```

Important Note:-

Associative arrays are very useful when we want to model memories on the test bench.

This is because memories are arrays having a large number of elements. If the dynamic array is used, then the memory will be allocated immediately after the array is defined which can lead to system failures. In an associative array, the memory will be allocated only when some value is stored in a particular key. Thus the memory requirement in the associative array will be on a need basis

Methods:-

1. **int num()** - returns the number of values or entries stored in the associative array.
2. **int size ()** - Also returns the number of values stored in the associative array. Returns 0 for empty array
3. **int delete([key])** - If the key is passed to this function then the value associated with the key is deleted. If no key is provided, then it deletes the entire array.
4. **int exists(key)** - checks whether a value is associated with the given key or not. If no value is present, it returns 0.
5. **int first (ref key)** - assigns the key variable passed with the first key of the associative array. If the array is empty 0 is returned
6. **int last(ref key)** - assigns the key variable passed with the last key of the associative array. If the array is empty 0 is returned.

7. **int next(ref key)** - assigns the key variable passed with the key which comes just after the specified key. If the array is empty or if the specified key is last, 0 is returned.
8. **int prev(ref key)** - assigns the key variable passed with the key which comes just before the specified key. If the array is empty or if the specified key is first, 0 is returned.

Code Practising:-

```
module associative_array();
    int num_cars[string];
    string brand;
    initial begin
        num_cars = '{"hyundai":2, "suzuki":1, "tata": 3};
        $display("Number of cars = %p", num_cars);
        $display("Number of cars for Suzuki is = %0d", num_cars["suzuki"]);
        // num and size
        $display("Number of brands stored = %0d", num_cars.num());
        $display("Size of the array = %0d\n", num_cars.size());
        // get first brand stored in array
        if(num_cars.first(brand))
            $display("First brand found is = %s", brand);
        // get last brand stored in array
        if(num_cars.last(brand))
            $display("Last brand found is = %s\n", brand);
        // get brand stored before last one
        if(num_cars.prev(brand))
            $display("Brand previous to last brand found is = %s", brand);
        // get brand stored before last one
        if(num_cars.next(brand))
            $display("Brand next to last brand found is = %s\n", brand);
        // get brand stored before last one
        if(num_cars.next(brand))
            $display("Brand next to last brand found is = %s\n", brand);
        else
            $display("Either no brand found or this is the last brand\n");
        // check whether a particular brand found or not
        brand = "suzuki";
        if(num_cars.exists(brand))
            $display("Enteries found for %s", brand);
        else
            $display("No enteries found for %s", brand);
        // check whether a particular brand found or not
        brand = "ferrari";
        if(num_cars.exists(brand))
            $display("Enteries found for %s\n", brand);
        else
            $display("No enteries found for %s\n", brand);
        // delete the array
        num_cars.delete();
        $display("Number of brands after deleting = %0d", num_cars.num());
    end
endmodule
```

Result:-

```
Number of cars = '{"hyundai":2, "suzuki":1, "tata":3 }
Number of cars for Suzuki is = 1
Number of brands stored = 3
Size of the array = 3

First brand found is = hyundai
Last brand found is = tata

Brand previous to last brand found is = suzuki
Brand next to last brand found is = tata

Either no brand found or this is the last brand

Enteries found for suzuki
No enteries found for ferrari

Number of brands after deleting = 0
```

Dynamic Array inside Associative Array :-

Code Practising:-

```
module associative_dyn_array();
    int num_cars[string];
    string car_name[string][];
    string brand;
    initial begin
      num_cars = '{"Tesla":2, "Ferari":1, "Hundai": 3};
        foreach (num_cars[brand]) begin
            car_name[brand] = new [num_cars[brand]];
            for (int i=0; i<car_name[brand].size(); ++i) begin
                car_name[brand][i] = $sformatf("%s i%0d", brand, (i+1)*10);
            end
            $display("Cars for %s brand are: %p", brand, car_name[brand]);
        end
      $display("car_name=%p",car_name);
    end
endmodule
```

Result:-

```
Cars for Ferari brand are: '{"Ferari i10"}
Cars for Hundai brand are: '{"Hundai i10", "Hundai i20", "Hundai i30"}
Cars for Tesla brand are: '{"Tesla i10", "Tesla i20"}
car_name='{"Ferari":'{"Ferari i10"}, "Hundai":'{"Hundai i10", "Hundai i20", "Hundai i30"}, "Tesla":'{"Tesla i10", "Tesla i20"}}
```

Associative Array inside Dynamic Array :-

Code Practising:-

```
module dyn_associative_array();
    int marks [][string];
    string sub;
    initial begin
        marks = new [2];
        marks[0] = '{"physics":45, "maths": 65, "emt": 56};
        marks[1] = '{"chemistry":67, "maths": 78, "optical communication": 89};
        foreach (marks[i]) begin
            $display("Marks of student %0d is %p", i+1, marks[i]);
        end
    end
endmodule
```

Result:-

```
Marks of student 1 is '{"emt":56, "maths":65, "physics":45}
Marks of student 2 is '{"chemistry":67, "maths":78, "optical communication":89}
```