SystemVerilog Basics

Reference

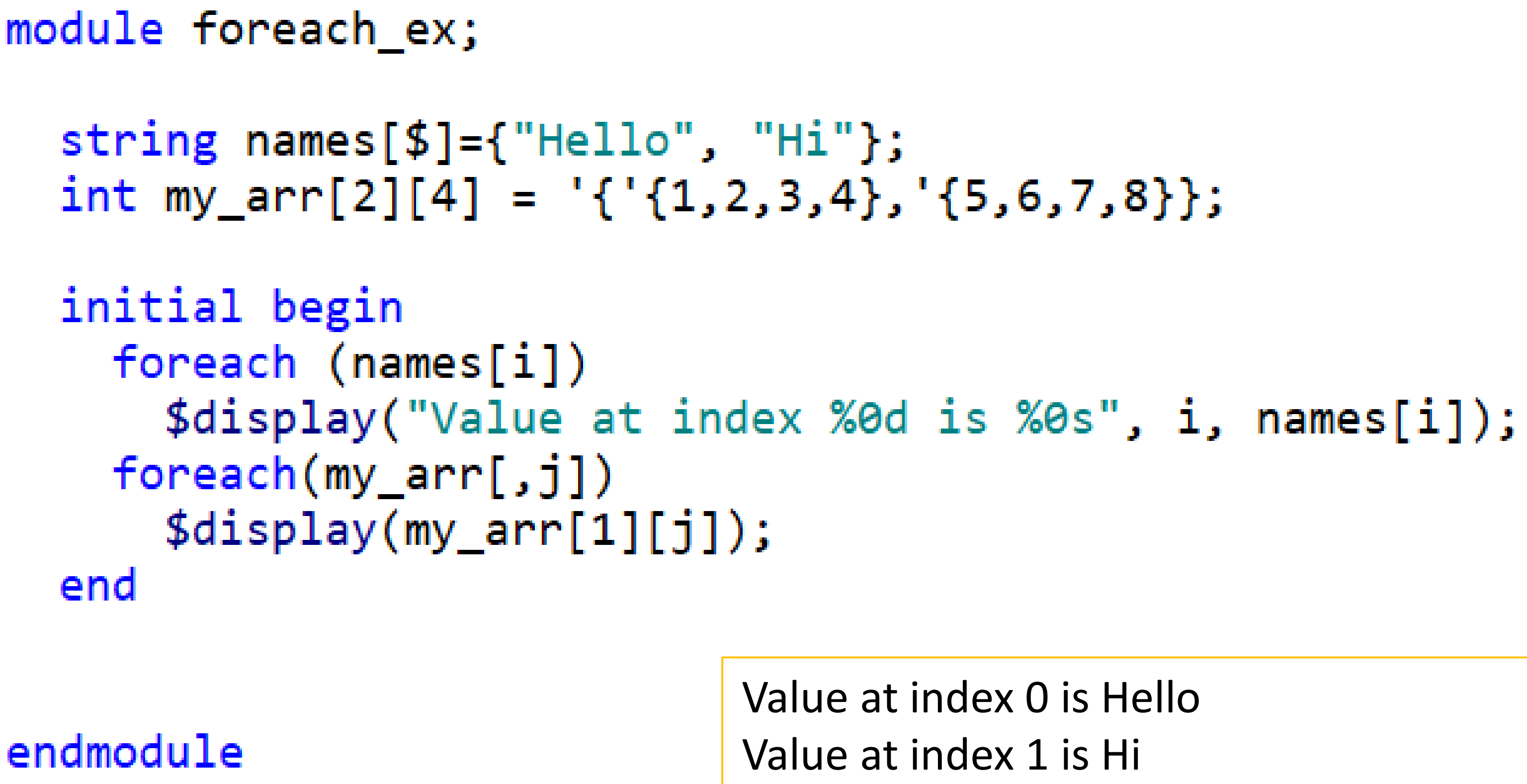
SystemVerilog Language Reference Manual

Contents

- Additional Constructs in SV

foreach

- foreach construct specifies iteration over the elements of an single dimensional fixed-size arrays, dynamic arrays

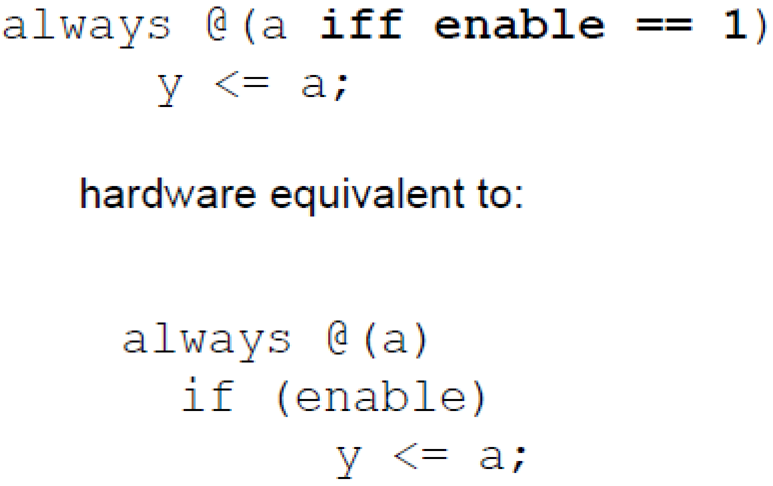


```
module foreach_ex;

  string names[$]={"Hello", "Hi"};
  int my_arr[2][4] = '{'{1,2,3,4},'{5,6,7,8}};

  initial begin
    foreach (names[i])
      $display("Value at index %0d is %0s", i, names[i]);
    foreach(my_arr[,j])
      $display(my_arr[1][j]);
  end


endmodule
```

```
Value at index 0 is Hello
Value at index 1 is Hi
5
6
7
8
```

iff event control

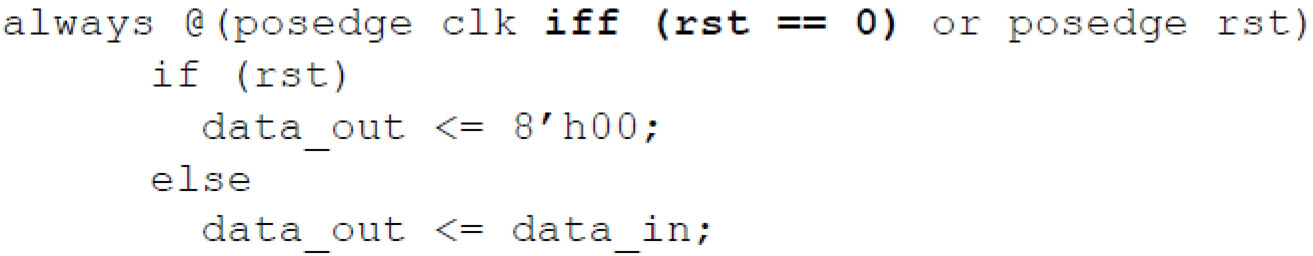
- A procedural event control
- Event expression triggers only if the iff condition is true



```
always @(a iff enable == 1)
     y <= a;
```

hardware equivalent to:

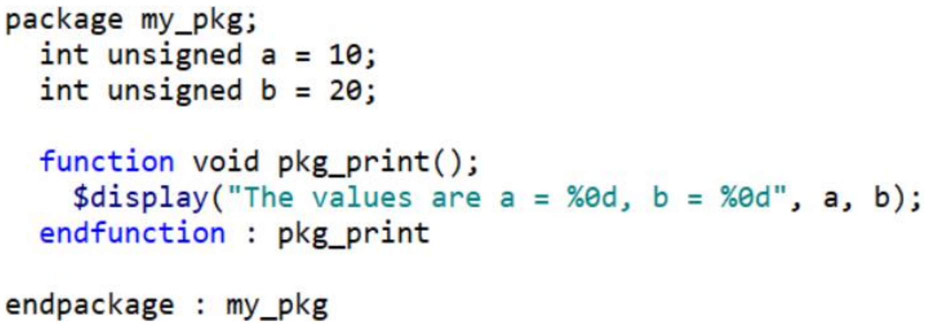
```
always @(a)
  if (enable)
       y <= a;
```



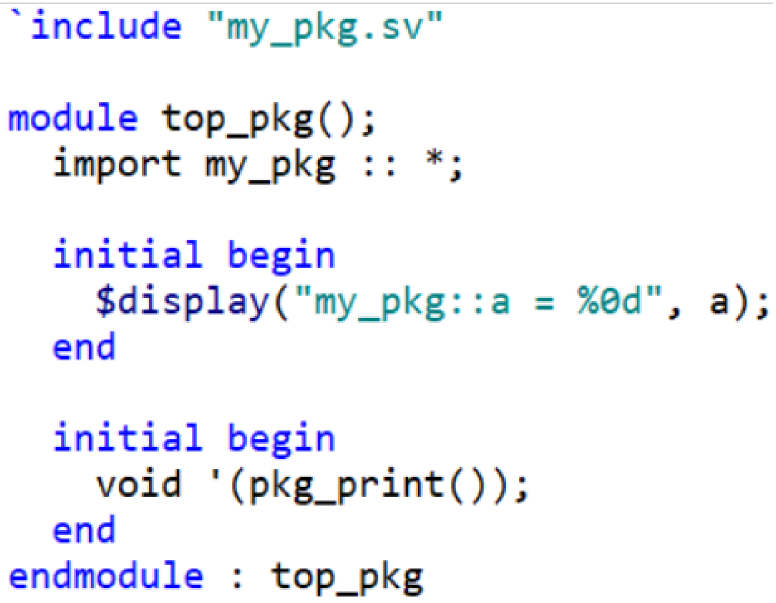
```
always @(posedge clk iff (rst == 0) or posedge rst)
     if (rst)
       data_out <= 8’h00;
     else
       data_out <= data_in;
```

Packages

- Packages provide a mechanism for storing and sharing data, methods, property, parameters that can be re-used in multiple other modules, interfaces or programs.
- They have explicitly named scopes that exist at the same level as the top-level module.
- So, all parameters and enumerations can be referenced via this scope.



```
package my_pkg;
  int unsigned a = 10;
  int unsigned b = 20;

  function void pkg_print();
    $display("The values are a = %0d, b = %0d", a, b);
  endfunction : pkg_print

endpackage : my_pkg
```



```
`include "my_pkg.sv"

module top_pkg();
  import my_pkg :: *;

  initial begin
    $display("my_pkg::a = %0d", a);
  end

  initial begin
    void '(pkg_print());
  end
endmodule : top_pkg
```