46th day

In SystemVerilog, write a constraint for a 12-bit array named rst such that:

- A random variable size determines how many bits are active (with a maximum value of 12).
- If size is **less than 12**, then:
  - The bits from index 0 to size-1 must **toggle alternately** (i.e., each bit should be the complement of the previous bit).
  - The bits from index size to 11 must be **forced to 0**.
- If size == 12, all bits from index 0 to 11 must follow the toggling pattern.

Code

```
class msd;

 rand bit rst[12];              // Fixed 12-bit pattern
 rand int unsigned size;     // How many bits to toggle

 constraint c_size {
  size inside {[2:12]};   // Limit size from 2 to 12
 }

 constraint c_toggle {
  foreach (rst[i]) {
    if (i > 0 && i < size)
      rst[i] ^ rst[i-1] == 1; // Toggle
    else if (i >= size)
```

```
      rst[i] == 0;          // Fill remaining with 0
    }
  }

  function void post_randomize();
    $display("\nRandomization Successfull");

    $display("Size = %0d", size);
    foreach (rst[i])
      $display("rst[%0d] = %b", i, rst[i]);
  endfunction

endclass

module tb;
  initial
    begin
    msd m = new();

     repeat (3)
      begin
      m.randomize();

      end

     end
endmodule
```

Simulation

```
Randomization Successfull
Size = 4
rst[0] = 1
rst[1] = 0
rst[2] = 1
rst[3] = 0
rst[4] = 0
rst[5] = 0
rst[6] = 0
rst[7] = 0
rst[8] = 0
rst[9] = 0
rst[10] = 0
rst[11] = 0

Randomization Successfull
Size = 9
rst[0] = 0
rst[1] = 1
rst[2] = 0
rst[3] = 1
rst[4] = 0
rst[5] = 1
rst[6] = 0
rst[7] = 1
rst[8] = 0
rst[9] = 0
rst[10] = 0
rst[11] = 0

Randomization Successfull
Size = 5
rst[0] = 0
rst[1] = 1
rst[2] = 0
rst[3] = 1
rst[4] = 0
rst[5] = 0
rst[6] = 0
rst[7] = 0
rst[8] = 0
rst[9] = 0
rst[10] = 0
rst[11] = 0
         V C S   S i m u l a t i o n   R e p o r t
Time: 0 ns
CPU Time:      0.480 seconds;       Data structure size:   0.0Mb
```