Week 5: Assignment 5

The due date for submitting this assignment is 11:59 PM.

Due on 2021-09-01, 23:59 IST.

1) Consider the following Verilog code using datamodules:

```verilog
module (unknown [public]);
output p;
input ABC;
wire Q1;
assign p = (ABC[2] AND ABC[1]) || 
         (ABC[1] AND ABC[0]) || 
         (ABC[0] AND ABC[2]) || 
         ((ABC[0] || ABC[1]) AND NOT ABC[2]);
endmodule
```

---

2) For the code given in Q1, if C = 1, the Boolean expression for p would be

Accepted Answers:

```
C = 1, Boolean expression for p would be:
```

---

3) The code in 0.4 implements the "or" output of an 8-bit adder with inputs a and b if

Accepted Answers:

```
a = a [7:0] || b [7:0];
```

---

4) Consider the following Verilog code:

```verilog
module (unknown [public]);
output p;
input ABC;
wire Q1;
assign p = (ABC[2] AND ABC[1]) || 
         (ABC[1] AND ABC[0]) || 
         (ABC[0] AND ABC[2]) || 
         ((ABC[0] || ABC[1]) AND NOT ABC[2]);
endmodule
```

---

5) The code in 0.4 implements the "or" output of an 8-bit adder with inputs a and b if

Accepted Answers:

```
a = a [7:0] || b [7:0];
```

---

6) Consider the following Verilog code:

```verilog
module (unknown [public]);
output p;
input ABC;
wire Q1;
assign p = (ABC[2] AND ABC[1]) || 
         (ABC[1] AND ABC[0]) || 
         (ABC[0] AND ABC[2]) || 
         ((ABC[0] || ABC[1]) AND NOT ABC[2]);
endmodule
```

---

7) The code in 0.4 implements the "or" output of an 8-bit adder with inputs a and b if

Accepted Answers:

```
a = a [7:0] || b [7:0];
```

---

8) Consider the following Verilog code:

```verilog
module (unknown [public]);
output p;
input ABC;
wire Q1;
assign p = (ABC[2] AND ABC[1]) || 
         (ABC[1] AND ABC[0]) || 
         (ABC[0] AND ABC[2]) || 
         ((ABC[0] || ABC[1]) AND NOT ABC[2]);
endmodule
```

---

9) Consider the following Verilog code:

```verilog
module (unknown [public]);
output p;
input ABC;
wire Q1;
assign p = (ABC[2] AND ABC[1]) || 
         (ABC[1] AND ABC[0]) || 
         (ABC[0] AND ABC[2]) || 
         ((ABC[0] || ABC[1]) AND NOT ABC[2]);
endmodule
```

---

10) Consider the following Verilog code:

```verilog
module (unknown [public]);
output p;
input ABC;
wire Q1;
assign p = (ABC[2] AND ABC[1]) || 
         (ABC[1] AND ABC[0]) || 
         (ABC[0] AND ABC[2]) || 
         ((ABC[0] || ABC[1]) AND NOT ABC[2]);
endmodule
```