Solutions to Quiz 2

Question 1

There were two versions this question. The questions and answers for each version are given below.

Version1

- A signal (variable) has the value 0 in a Verilog statement. Would it be considered True or False?

  In Verilog a zero is considered False.

- You need to set a one-bit output named **first** to a true value. Would you assign it a value of 0 or 1 in Verilog?

  The name implies the output is active-high. To set an output true it must be set high. To set a output high in Verilog we must write a 1.

- An input indicates that something is ready when it is at a low logic level. Would this signal be named **ready** or $\overline{\textbf{ready}}$?

  If a signal is true when it is low it is active-low so it would be named $\overline{\textbf{ready}}$.

- An input named **full** indicates if something is full. If it is full, would the value read by a Verilog module have the value 0 or 1?

  The input is active-high and is true so it must be high. When input by Verilog it must be 1.

- The value read from an input port named $\overline{\textbf{dark}}$ is used in a Verilog expression and has a true value. Was that port at a high or low logic level?

  If a single-bit value is used in an expression and has a true value it must be 1. A value read from an input port is 1 if high and 0 if low. So the level must be high (the name is not relevant in this case).

-

Version2

- A signal (variable) has the value 1 in a Verilog statement. Would it be considered True or False?

  In Verilog a non-zero value is considered True.

- You need to set a one-bit output named $\overline{\textbf{first}}$ to a true value. Would you assign it a value of 0 or 1 in Verilog?

  The name implies the output is active-low. To set an output true it must be set low. To set a output low in Verilog we must write a 0.

- An input indicates that something is ready when it is at a high logic level. Would this signal be named **ready** or $\overline{\textbf{ready}}$?

  If a signal is true when it is high it is active-high so it would be named **ready**.

- An input named $\overline{\textbf{full}}$ indicates if something is full. If it is full, would the value read by a Verilog module have the value 0 or 1?

  The input is active-low and is true so it must be low. When input by Verilog it must be 0.

- The value read from an input port named **dark** is used in a Verilog expression and has a true value. Was that port at a high or low logic level?

  If a single-bit value is used in an expression and has a true value it must be 1. A value read from an input port is 1 if high and 0 if low. So the level must be high (the name is not relevant in this case).

-

Question 2

Label the underlined blanks in the diagram below and connect the module ports so the diagram matches the module **M**. There are two (2) missing module:instance names, ten (10) missing port names and five (5) missing connections. One module:instance name, two port names and two connections are shown as examples.

```
module N ( input logic x, y, output logic z ) ;
   // ...
endmodule
module O ( input logic a, b, output logic c ) ;
```

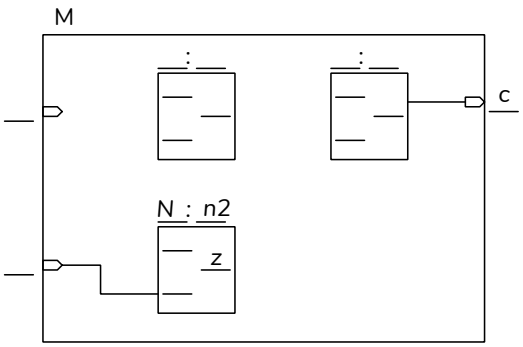
```
   // ...
endmodule

module M ( output logic c, input logic p1, p2 ) ;

   logic a, b ;

   N n1 (.z(a), .y(p1), .x(p2)) ;
   N n2 (p1, p2, b) ;
   O o1 (.*) ;

endmodule
```



Answer

To solve this problem we can rewrite the component instantiations with explicit .port(signal) mappings. Arranging them in the same order as in the module declarations helps avoid errors. The result is:

```
module M ( output logic c, input logic p1, p2 ) ;

   logic a, b ;

   N n1 (.x(p2), .y(p1), .z(a)) ;
   N n2 (.x(p1), .y(p2), .z(b)) ;
   O o1 (.p1(p1), .p2(p2), .c(c)) ;

endmodule
```

From which we can fill in the diagram. The RTL netlist generated by Quartus when it synthesized the code given in the question is: