1. **a.** With reference to the Verilog Hardware Description Language, describe with the aid of a timing diagram, what you understand by:

i. Inertial Delay

ii. Transport Delay

(8)

b. Draw the circuit diagram for the linear feedback shift register described by the Verilog code shown (^ represents the exclusive-or function in Verilog).

```
module lfsr (
  input clk,
  input rst_n,

  output reg [4:0] data
);

reg [4:0] data_next;

always @* begin
  data_next[4] = data[4]^data[1];
  data_next[3] = data[3]^data[0];
  data_next[2] = data[2]^data_next[4];
  data_next[1] = data[1]^data_next[3];
  data_next[0] = data[0]^data_next[2];
end

always @(posedge clk or negedge rst_n)
  if(!rst_n)
    data <= 5'h1f;
  else
    data <= data_next;

endmodule
```

The circuit is simulated with the following testbench values. Draw a timing diagram clearly showing clk, rst_n and the data values which are output.

```
      clk = 0;
      rst = 1;
      #100 rst = 0;
      #100 rst = 1;
      #100 clk = 1;
      #100 clk = 0;
      #100 clk = 1;
      #100 clk = 0;
      #100 clk = 1;
      #100 clk = 0;
```

(8)

c. The following Verilog code has been used to model a multiplexer. What is the problem with this model and what would be the unintended result when the code is synthesized?

```
module
Mux4(I3, I2, I1, I0, S1, S0, D);
input
I3, I2, I1, I0;
input S1, S0;
output D;
reg D;

always @(S1, S0)
begin
  if
    (S1==0 && S0==0)
     D <= I0;
  else if
    (S1==0 && S0==1)
     D <= I1;
  else if
    (S1==1 && S0==0)
     D <= I2;
  else
     D <= I3;
  end
endmodule
```

(4)