Task 1

module *task\_1* (

*input*  Clk,

*input*  R,

*input*  S,

*output* Q

);

   wire R\_g, S\_g, Qa, Qb  /\* synthesis keep \*/;

   assign R\_g = R & Clk;

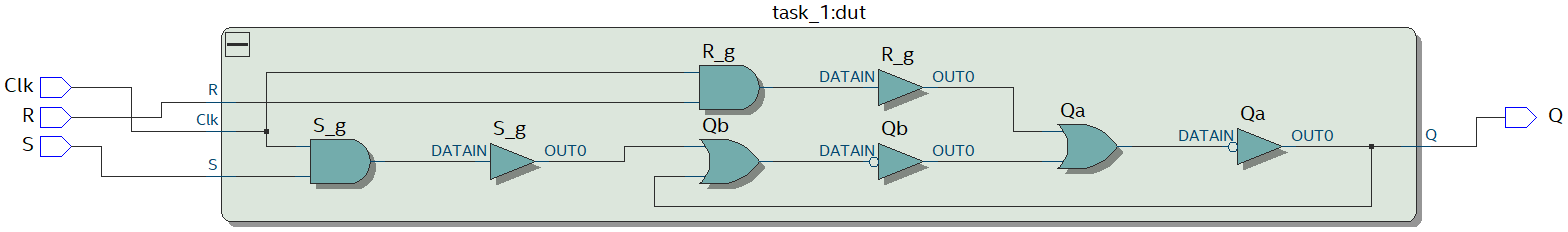
   assign S\_g = S & Clk;

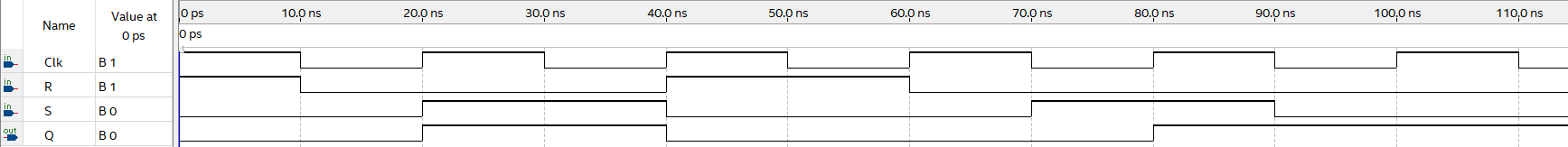
   assign Qa  = ~(R\_g | Qb);

   assign Qb  = ~(S\_g | Qa);

   assign Q   = Qa;

endmodule





Task 2

module *task\_2* (

*input*  Clk,

*input*  D,

*output* Q

);

   wire R, S, R\_g, S\_g, Q\_a, Q\_b  /\* synthesis keep \*/;

   assign R   = ~D;

   assign S   = D;

   assign R\_g = ~(R & Clk);

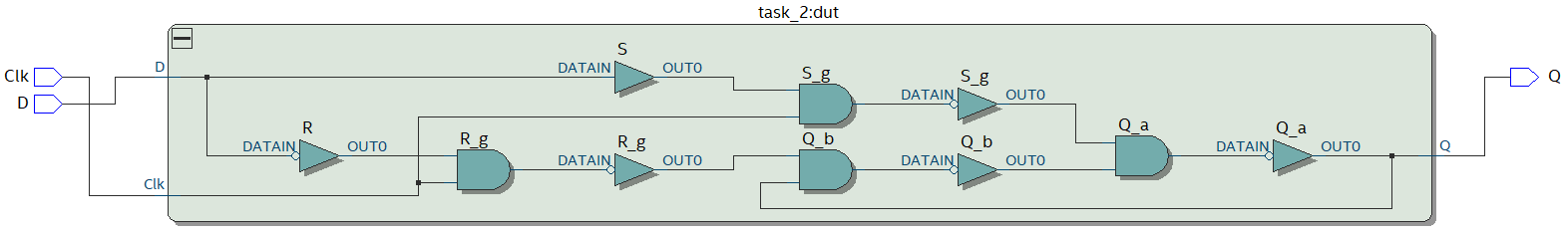
   assign S\_g = ~(S & Clk);

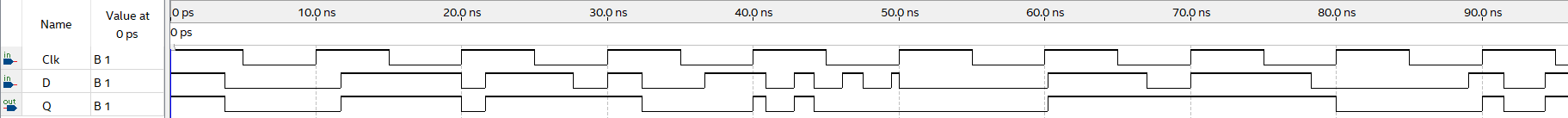
   assign Q\_a = ~(S\_g & Q\_b);

   assign Q\_b = ~(R\_g & Q\_a);

   assign Q   = Q\_a;

endmodule





Task 3

module *task\_3* (

*input*  Clk,

*input*  D,

*output* Q

);

   wire Q\_m, Q\_s, inv\_Clk  /\* synthesis keep \*/;

   assign inv\_Clk = ~Clk;

*task\_2* master\_latch (

       .Clk(inv\_Clk),

       .D  (D),

       .Q  (Q\_m)

   );

*task\_2* slave\_latch (

       .Clk(Clk),

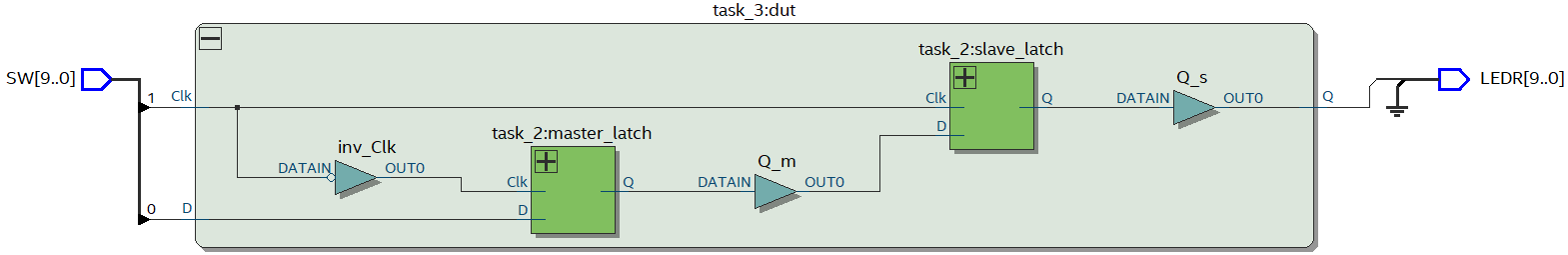
       .D  (Q\_m),

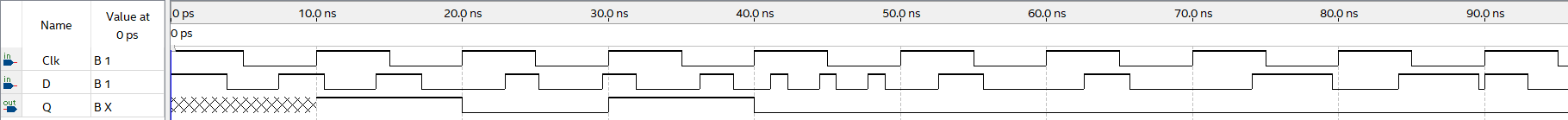
       .Q  (Q\_s)

   );

   assign Q = Q\_s;

endmodule





Task 4

