**Name:** Qasim, Junaid, Adeel, Tahir CH, Farukh

**Roll # :** 161, 144, 146, 164, 165

**D-Morgan Law:**

1. **L.H.S**

module DMorg(Y, A, B);

output Y;

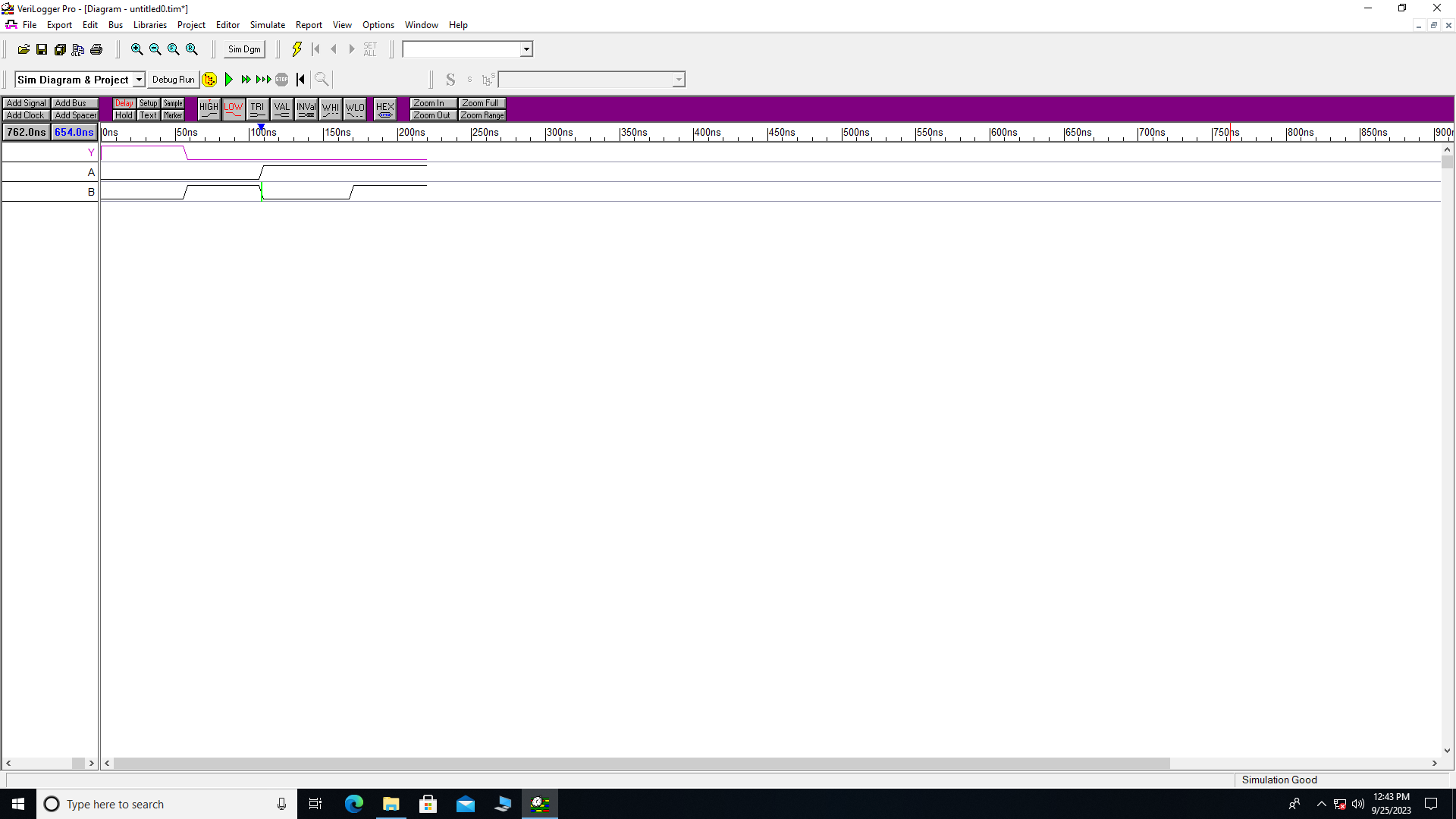
input A, B;

wire W1;

or r (W1, A, B);

not n (Y, W1);

endmodule



1. **R.H.S**

module DMorg(Y, A, B);

output Y;

input A, B;

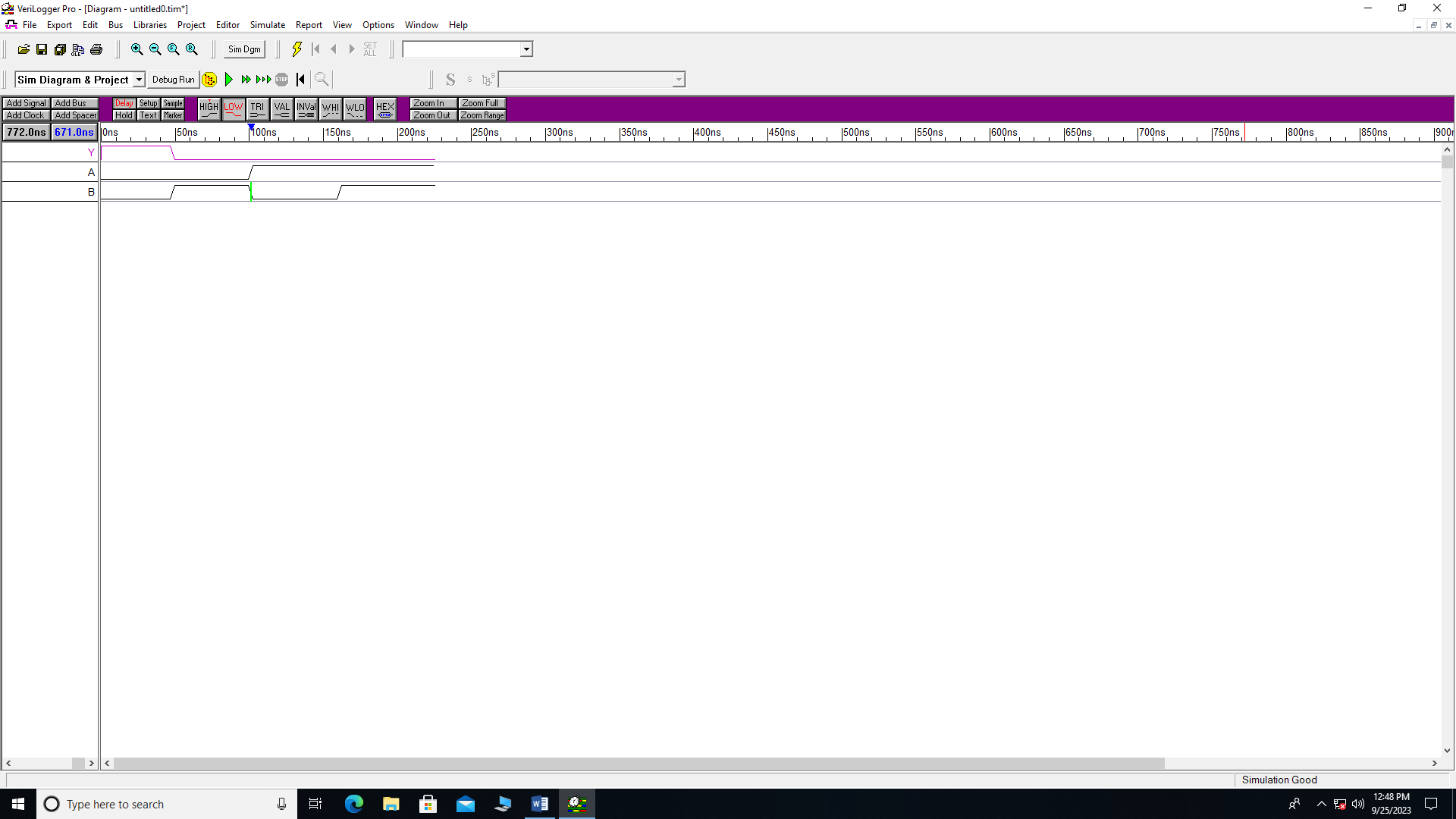
wire W1, W2;

not n1 (W1, A);

not n2 (W2, B);

and a(Y, W1, W2);

endmodule



**2. L.H.S**

module DMorg(Y, A, B);

output Y;

input A, B;

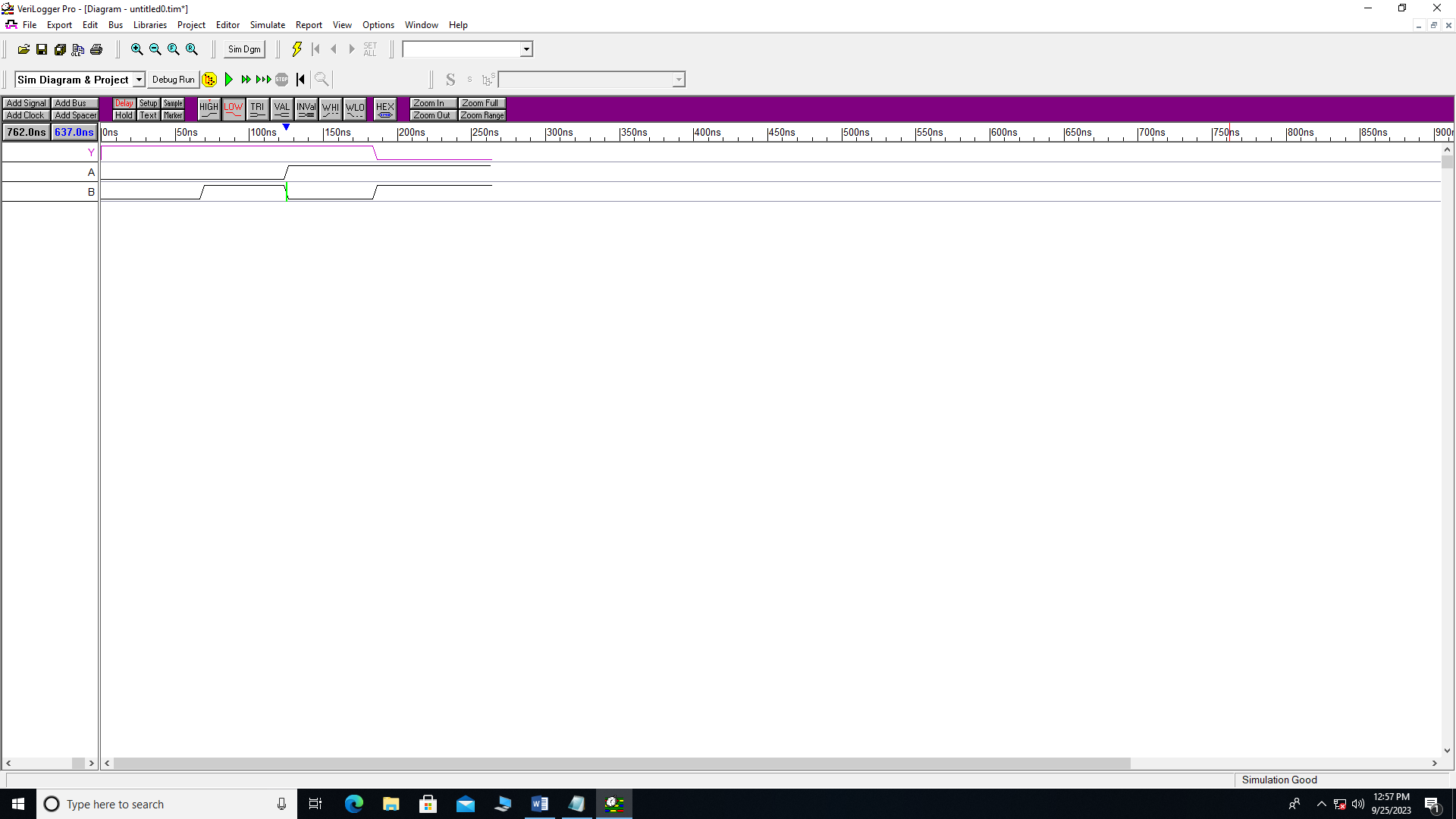
wire W1, W2;

not n1 (W1, A);

not n2 (W2, B);

or a(Y, W1, W2);

endmodule



**2. R.H.S**

module DMorg(Y, A, B);

output Y;

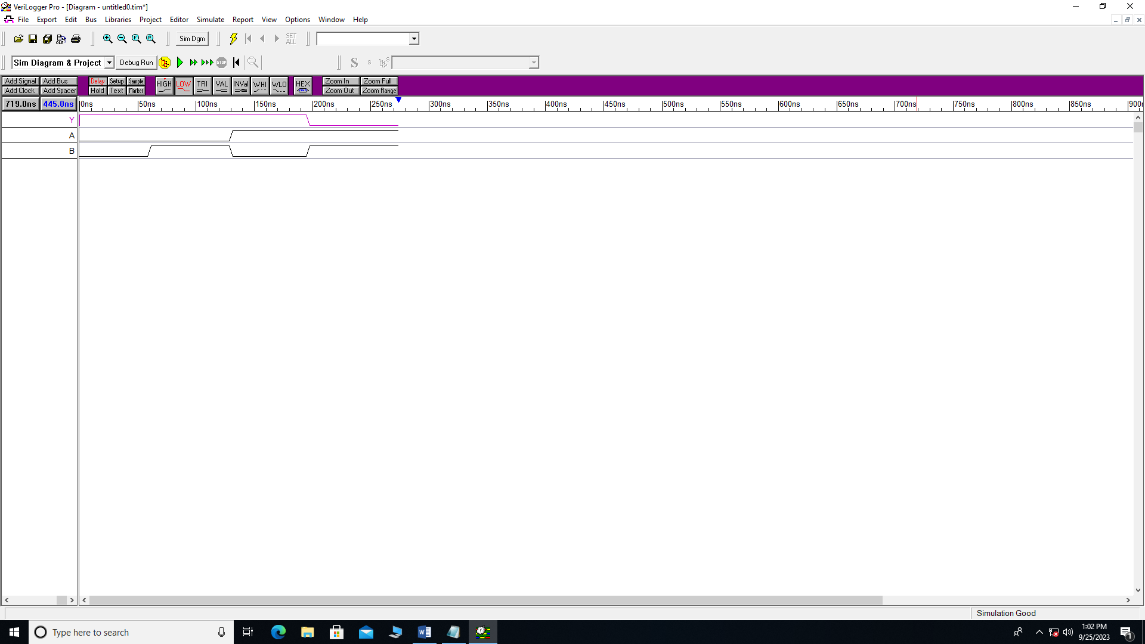
input A, B;

wire W1;

and a(W1, A, B);

not n1 (Y, W1);

endmodule



**Gates:**

module gates(Y, X, Z, L, A, B, C, D, E, M, N);

output Y, X, Z, L;

input A, B, C, D, E, M, N;

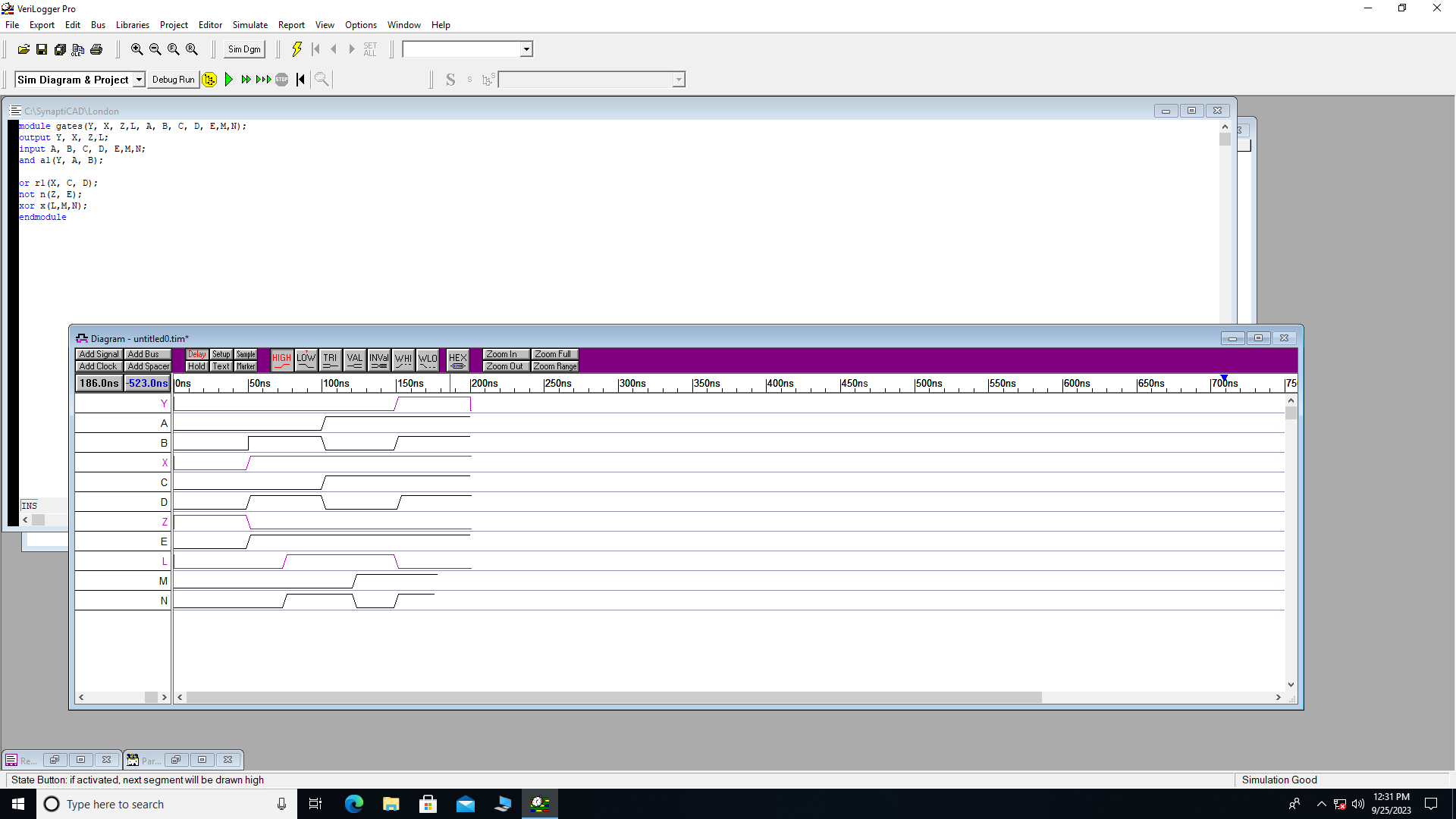
and a1 (Y, A, B);

or r (X, C, D);

not n (Z, E);

xor x1 (L, M, N);

endmodule



**Distributive Law:**

1. **L.H.S**

module Dist(Y, A, B, C);

output Y;

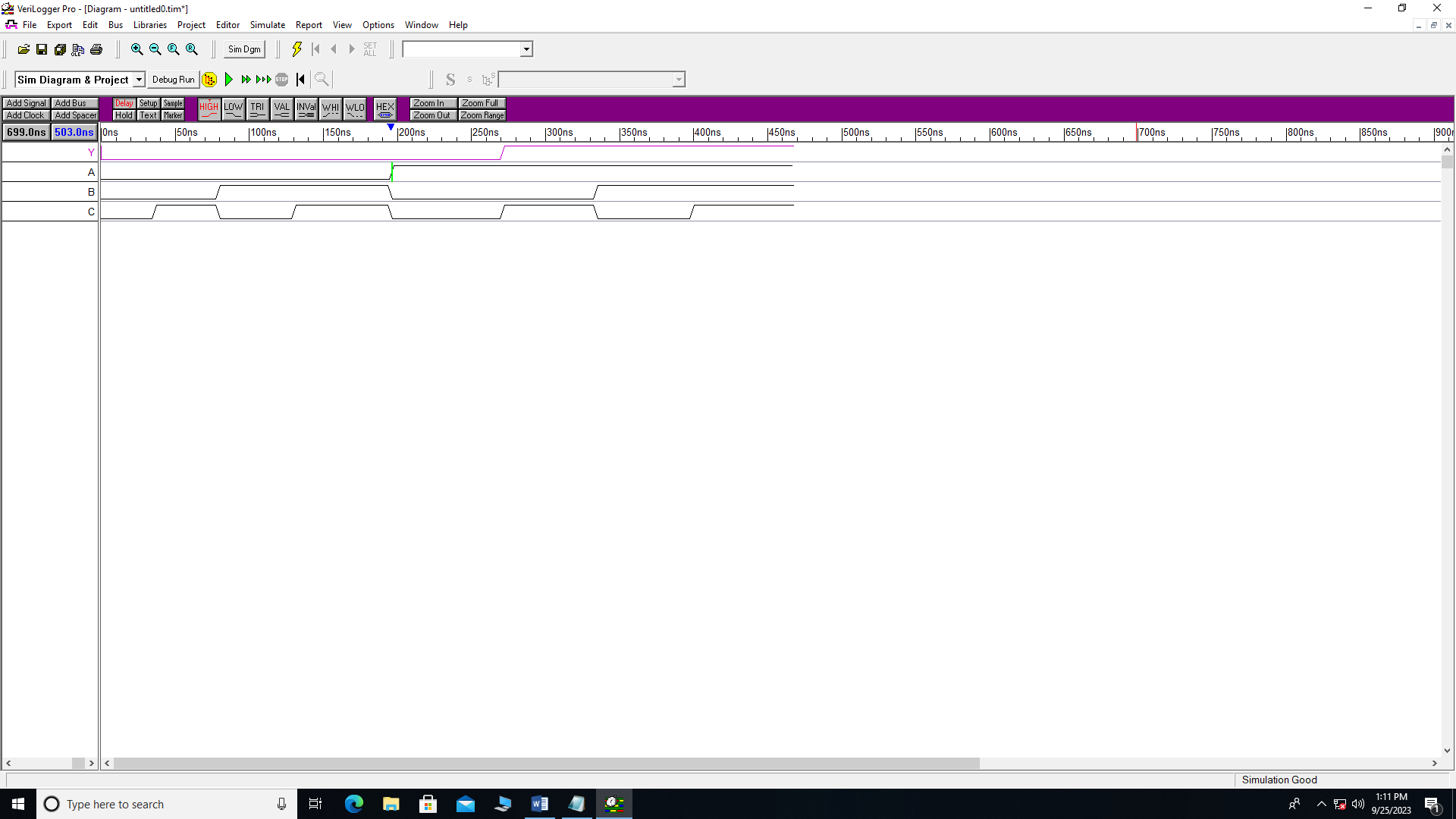
input A, B, C;

wire W1;

or r(W1, B, C);

and a(Y, W1, A);

endmodule

****

1. **R.H.S**

module Dist(Y, A, B, C);

output Y;

input A, B, C;

wire W1, W2;

and a1(W1, A, B);

and a(W2, A, C);

or r (Y, W1, W2);

endmodule

