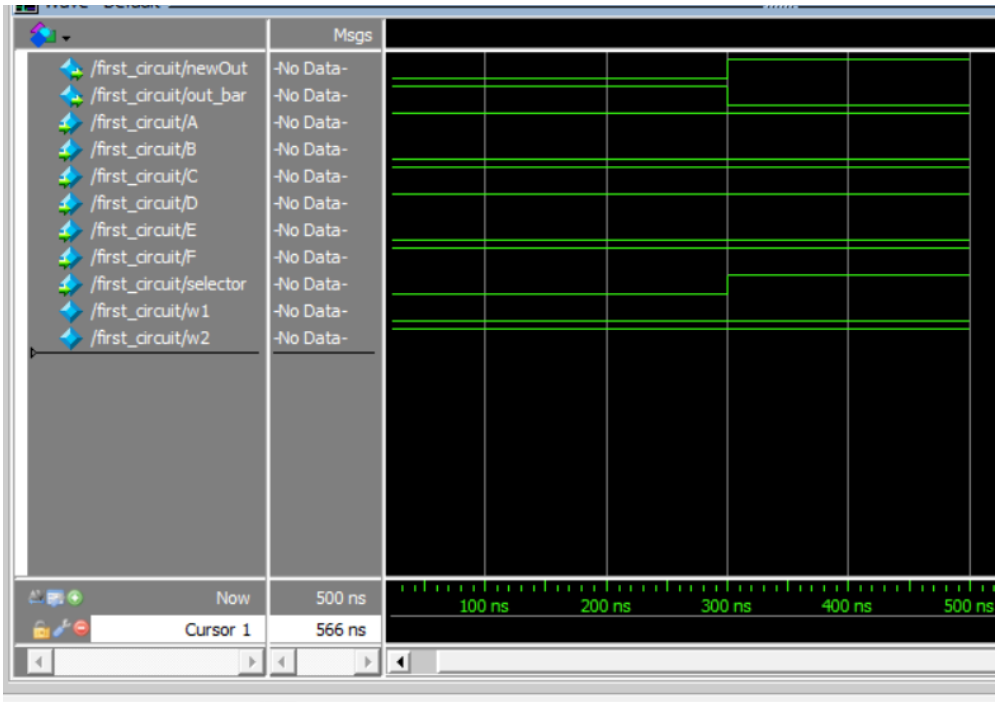
By: Abdelrahman Ahmed Sayed

# First circuit

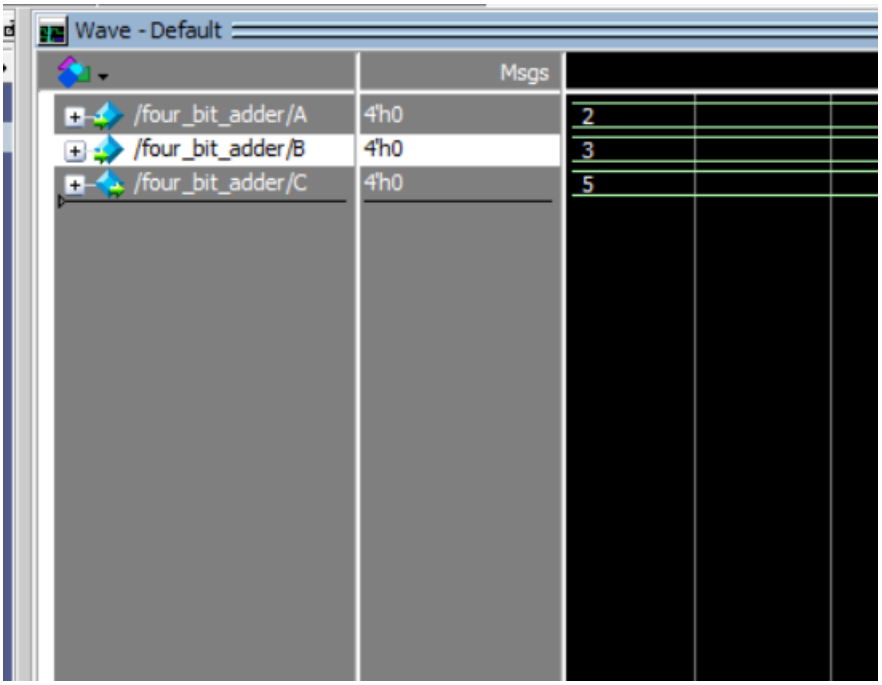
```v
module first_circuit (output newOut, out_bar, input A, B,C,D,E,F, selector);

wire w1, w2 ;

assign w1 = A && B && C;
assign w2 = D ~^ E ~^ F;

mux2 m(.out(newOut), .in0(w1), .in1(w2), .sel(selector));

assign out_bar = !newOut;

endmodule
```

```v
module mux2(in0, in1, sel, out);
input in0, in1, sel;
output out;
assign out = (sel == 1)? in1: in0;

endmodule
```



# second circuit

```
mux2.v   first_question.v   second_question.v ×
second_question.v > four_bit_adder
1  module four_bit_adder (input [3:0] A, B, output [3:0] C);
2
3  assign C = A + B;
4
5  endmodule
```

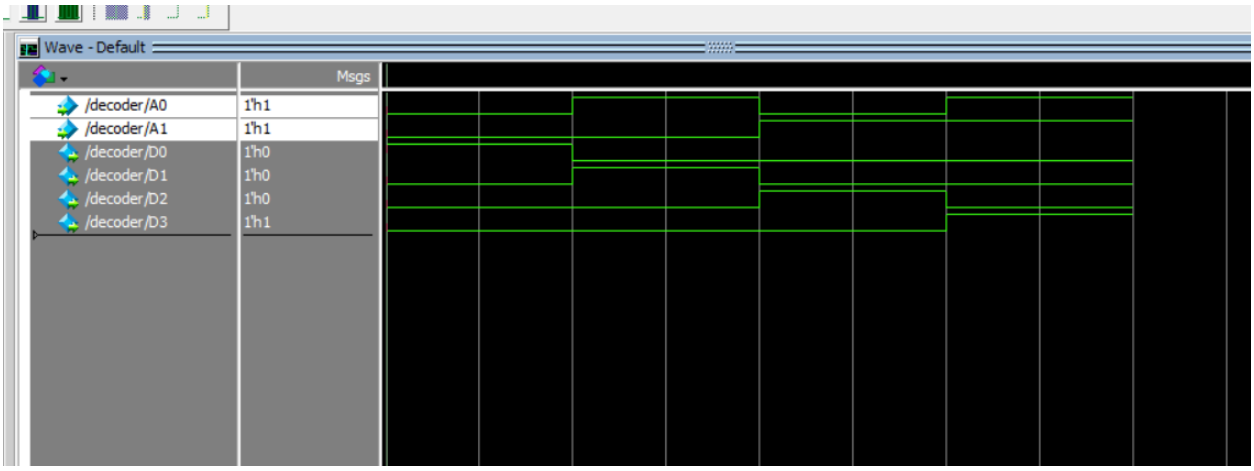


# Third circuit

mux2.v | lab_1.v | lab_3.v | lab_2.v

Assignment_1 > lab_3.v > ...

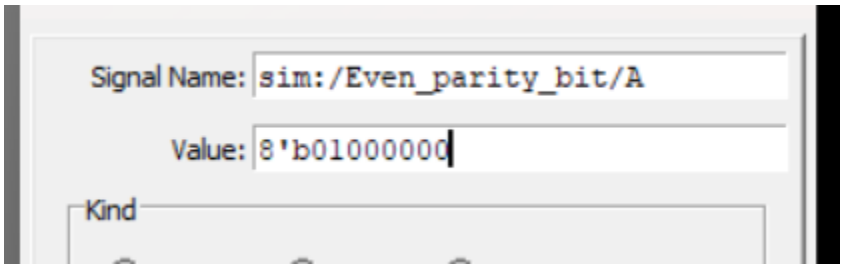
```
1  module decoder(input A0, A1, output D0, D1, D2, D3);
2
3  assign D0 = (A1 == 0 && A0 == 0) ? 1 : 0;
4  assign D1 = (A1 == 0 && A0 == 1) ? 1 : 0;
5  assign D2 = (A1 == 1 && A0 == 0) ? 1 : 0;
6  assign D3 = (A1 == 1 && A0 == 1) ? 1 : 0;
7
8  endmodule
9
```



# fourth circuit

```
mux2.v   lab_1.v   lab_3.v   lab_4.v   lab_2.v
Assignment_1 > lab_4.v > Even_parity_bit
1  module Even_parity_bit(input [7:0] A, output [8:0] out);
2
3  wire parity_bit;
4  assign parity_bit = ^A;
5  assign out = {A, parity_bit};
6
7  endmodule
```

first wave




# 5 circuit

Assignment_1 > lab_5.v > ...

```
1  module Comparator (
2  input [3:0] A, B, output A_greaterthan_B, output A_equals_B, output A_lessthan_B);
3
4  assign A_greaterthan_B = (A > B) ? 1'b1 : 1'b0;
5  assign A_equals_B = (A == B) ? 1'b1 : 1'b0;
6  assign A_lessthan_B = (A < B) ? 1'b1 : 1'b0;
7
8  endmodule
```