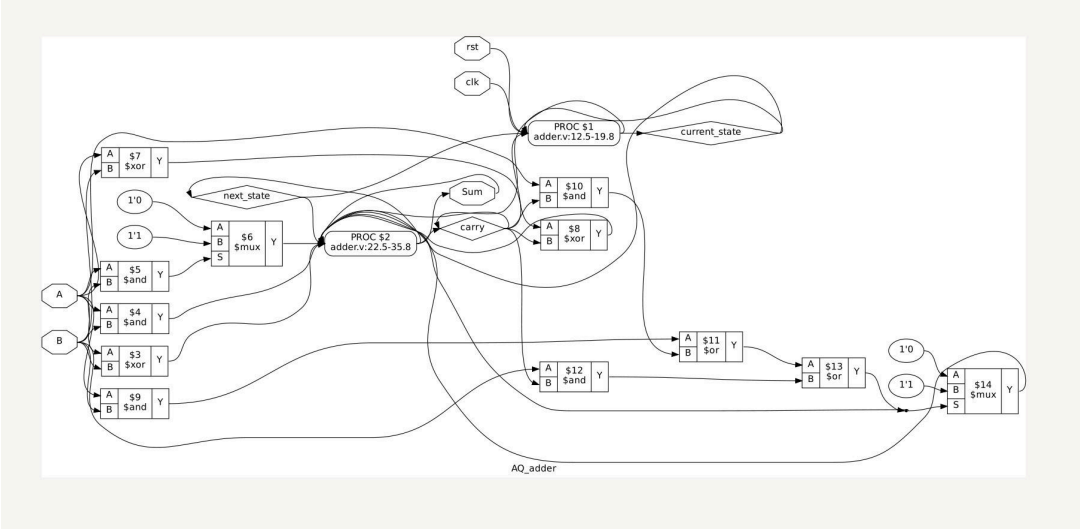
Name: Abdul Qadeer
Roll No: EL-02/2025
Advanced Digital System Design Mid 1 Paper

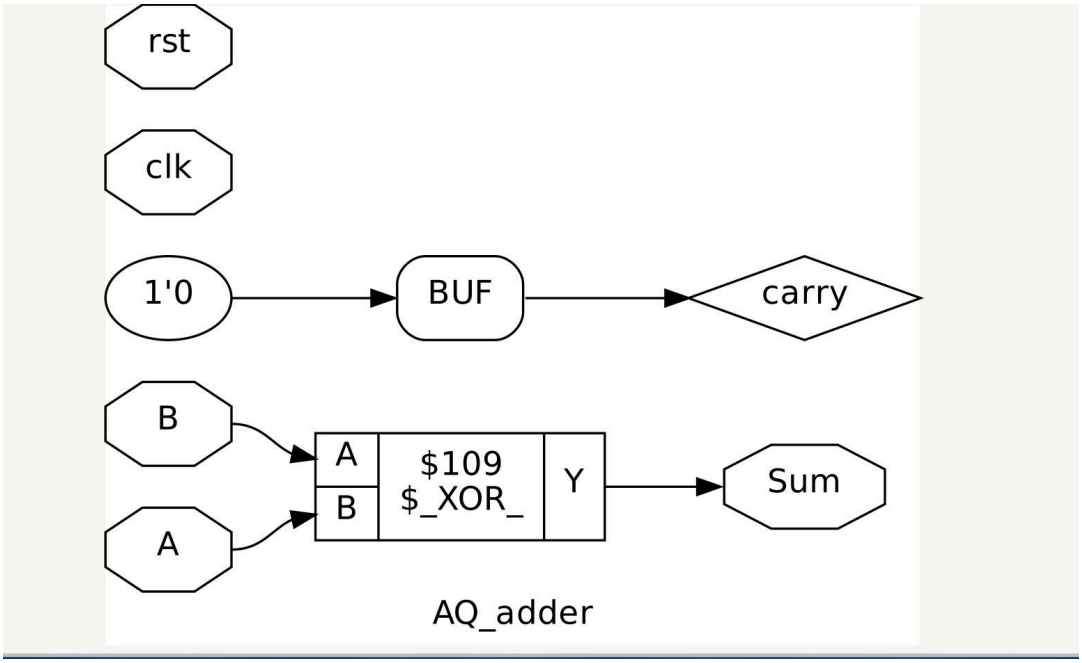
Output Files

Yosys Synthesis Output

Before Optimization



After Optimization



EDA Design Code

```
module binary_stream_adder (
   clk, rst, A, B, Sum
);
   input clk, rst, A, B;   // Declare inputs separately
   output reg Sum;        // Declare output as `reg` for sequential logic
   reg carry;             // Declare carry as `reg`

   // Define State Encoding
   typedef enum reg {S0, S1} state_t;
   state_t current_state, next_state;

   // Next state and output logic (Mealy Machine)
   always @(posedge clk or posedge rst) begin
      if (rst) begin
         current_state <= S0;
         carry <= 0;
      end else begin
         current_state <= next_state;
      end
   end

   // State transition logic
   always @(*) begin
      case (current_state)
         S0: begin
            Sum = A ^ B;      // Sum output = A XOR B
            carry = A & B;    // Carry = A AND B
            next_state = (A & B) ? S1 : S0;
         end
         S1: begin
            Sum = A ^ B ^ carry; // Sum with carry
            carry = (A & B) | (A & carry) | (B & carry); // Updated Carry
            next_state = (carry) ? S1 : S0;
         end
      endcase
   end

Endmodule
```

EDA Test Bench Code

```
timescale 1ns / 1ps

module adder_tb;
    // Inputs
    reg A, B, Cs;

    // Outputs
    wire S, C_next;

    // Instantiate the adder module
    adder uut (
        .A(A),
        .B(B),
        .Cs(Cs),
        .S(S),
        .C_next(C_next)
    );

    // Test procedure
    initial begin
        // Monitor values
        $monitor("Time = %0t | A = %b | B = %b | Cs = %b || S = %b | C_next = %b",
                 $time, A, B, Cs, S, C_next);

        // Apply test cases
        A = 0; B = 0; Cs = 0; #10;
        A = 0; B = 0; Cs = 1; #10;
        A = 0; B = 1; Cs = 0; #10;
        A = 0; B = 1; Cs = 1; #10;
        A = 1; B = 0; Cs = 0; #10;
        A = 1; B = 0; Cs = 1; #10;
        A = 1; B = 1; Cs = 0; #10;
        A = 1; B = 1; Cs = 1; #10;

        // End simulation
        $finish;
    end
endmodule
```

EDA Output Wave