يوسف أحمد محمد ابراهيم

Assignment 1

Extra

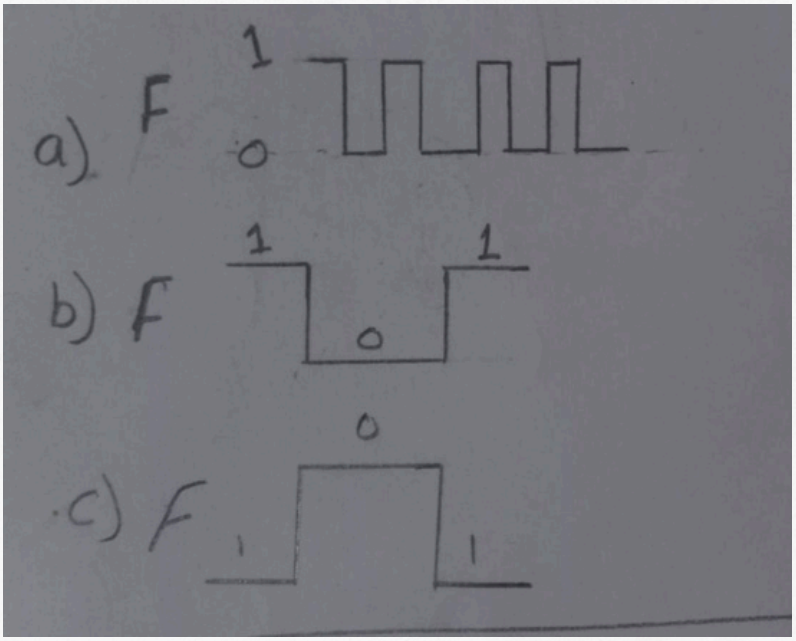
Circuit_1 :

- **module:**

```
1   module circuit_1(input [3:0]a, output out);
2       assign out = (a < 4'b1000 && a > 4'b0010) ? (1'b1) : (0);
3   endmodule
```

- **Output:**

2.

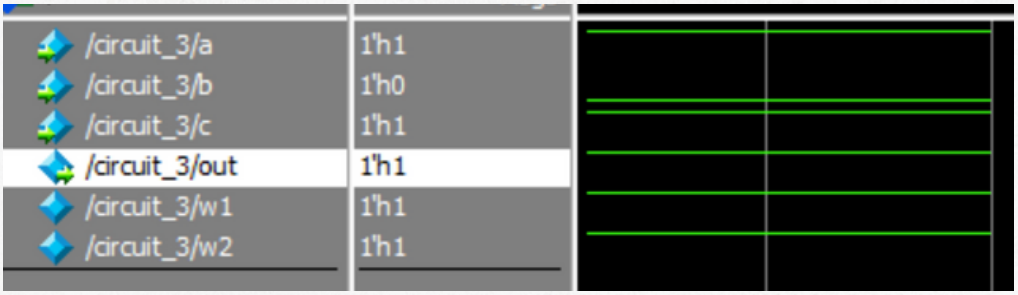


Circuit_3:

- module

```
1   module circuit_3(input a, b, c, output out);
2       xor (w1, a, b);
3       xnor (w2, c, b);
4       and (out, c, w1, w2);
5   endmodule
```

- waveform:

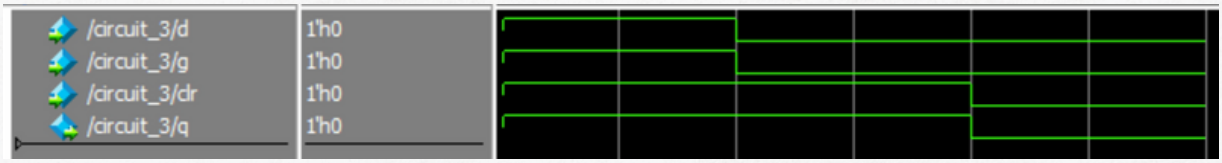


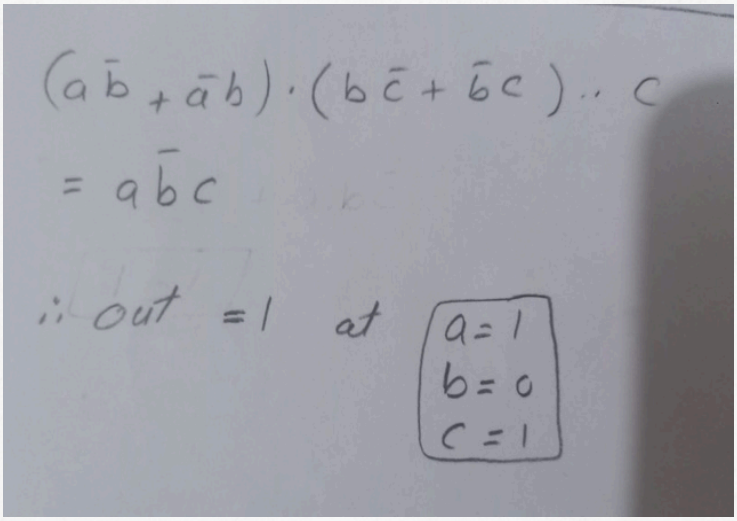
Circuit_3:

- code

```
1  module circuit_3(input d, g, clr, output q);
2      always @(*) begin
3          if(~clr)
4              q = 0;
5          else if(g)
6              q = d;
7      end
8  endmodule
```

- output:





4. Using nand gate

Circuit_5:

- **code**

```
module circuit_5(input a, b, ainv, binv, [1:0]op, carryin, output reg result);

    reg wirea, wireb;


    always @(*) begin
        if(ainv)
            wirea = ~a;
        else
            wire a = a;
        if(binv)
            wireb = ~b;
        else
            wire b = b;

        case (op)
            2'b0 : result = wirea & wireb;
            2'b1 : result = wirea | wireb;
            default : result = wirea + wireb + carryin;
        endcase
    end
endmodule
```

- **output:**