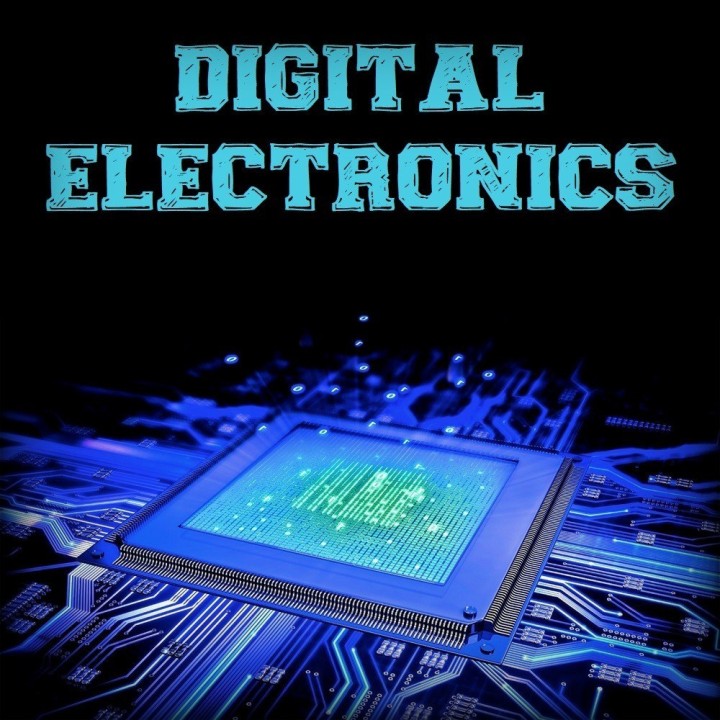
***Abanob Evram***

***Assignmen1[Extra]***



A white background with black text

Description automatically generated**[Q1]**

**The code:**

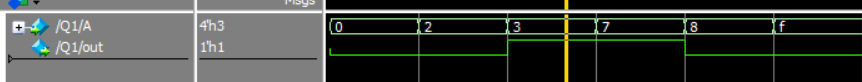
module Q1(A,out);

input [3:0] A;

output out ;

assign out = (2<A&&A<8)?1:0;

endmodule



A diagram of a waveform

Description automatically generated**[Q2** **]**

A paper with writing on it

Description automatically generated

A diagram of a wire

Description automatically generated**[Q3** **]**

**The code:**

module Q3(A,B,C,F);

input A,B,C;

output F;

wire z0,z1;

assign z0 = (A^B) ;

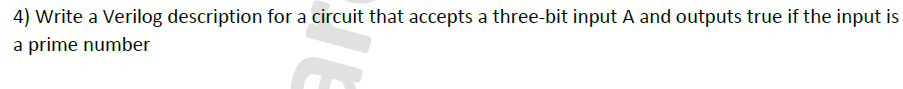
assign z1 = ~(B^C);

assign F=z0&z1&C;

endmodule

**A black and green rectangular object with white lines

Description automatically generated with medium confidenceNote:** the only case that F=1 it is [A=0,B=1,C=1]

**[Q4** **]**

**The code:**

module prime(A,Out);

input [2:0] A ;

output reg Out;

always @(\*) begin

case (A)

2 :Out =1;

3 :Out =1;

5 :Out =1;

7 :Out =1;

default: Out=0;

endcase

end

endmodule

**Another code:**

module prime (A,Out);

input [2:0] A;

output Out;

assign out=(A==2 ||A==3 ||A==5 || A==7)? 1:0;

endmodule



**[Q5** **]**

A screenshot of a computer

Description automatically generated

**The code:**

module Mux2(In0\_Mux2,In1\_Mux2,Sel\_Mux2,Out\_Mux2);

input In0\_Mux2,In1\_Mux2,Sel\_Mux2;

output Out\_Mux2;

assign Out\_Mux2=(Sel\_Mux2==1)?In1\_Mux2:In0\_Mux2;

endmodule

module Mux3(In0\_Mux3,In1\_Mux3,In2\_Mux3,Sel\_Mux3,Out\_Mux3);

input In0\_Mux3,In1\_Mux3,In2\_Mux3;

input [1:0] Sel\_Mux3;

output Out\_Mux3;

assign Out\_Mux3=(Sel\_Mux3==0)?In0\_Mux3:(Sel\_Mux3==1)?In1\_Mux3:In2\_Mux3;

endmodule

module Adder2(In0\_Adder,In1\_Adder,Carryout\_Adder,Carryin\_Adder,Out\_Adder);

input In0\_Adder,In1\_Adder,Carryin\_Adder;

output Out\_Adder,Carryout\_Adder;

assign Out\_Adder =In0\_Adder+In1\_Adder+Carryin\_Adder ;

assign Carryout\_Adder=(In0\_Adder&In1\_Adder)|(Carryin\_Adder&In1\_Adder)|(Carryin\_Adder&In0\_Adder);

endmodule

module Alu(A,B,Ainvert,Binvert,Carryin,Operation,Carryout,Result);

input A,B,Ainvert,Binvert,Carryin;

input [1:0] Operation;

output Carryout,Result;

wire Z0,Z1,Z2;

Mux2 m1(.Sel\_Mux2(Ainvert),.In0\_Mux2(A),.In1\_Mux2(~A),.Out\_Mux2(Z0));

Mux2 m2(.Sel\_Mux2(Binvert),.In0\_Mux2(B),.In1\_Mux2(~B),.Out\_Mux2(Z1));

Adder2 a1(.In0\_Adder(Z0),.In1\_Adder(Z1),.Carryin\_Adder(Carryin),.Carryout\_Adder(Carryout),.Out\_Adder(Z2));

Mux3 m3(.Sel\_Mux3(Operation),.In0\_Mux3(Z0&Z1),.In1\_Mux3(Z0|Z1),.In2\_Mux3(Z2),.Out\_Mux3(Result));

endmodule

A screenshot of a computer

Description automatically generated

A paper with writing on it

Description automatically generated