ELEC5566M

FPGA Design for System on Chip

Assignment 1: 2-Bit x 3-Bit Multiplier

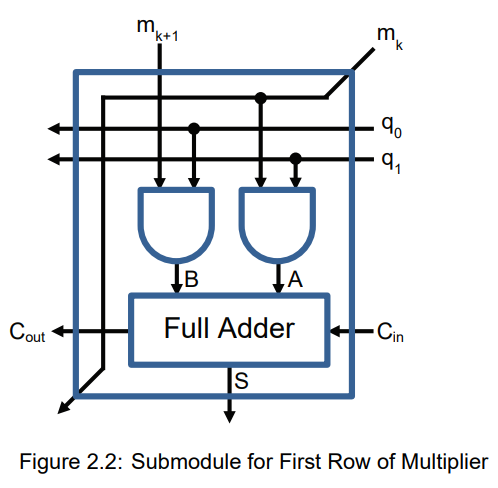
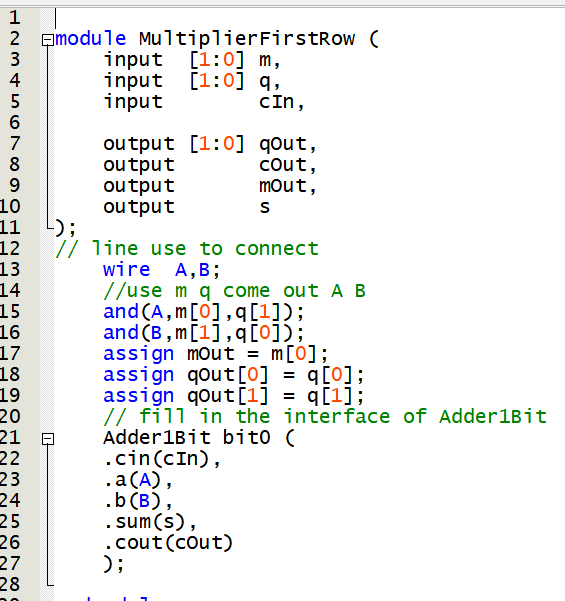
**201703627**

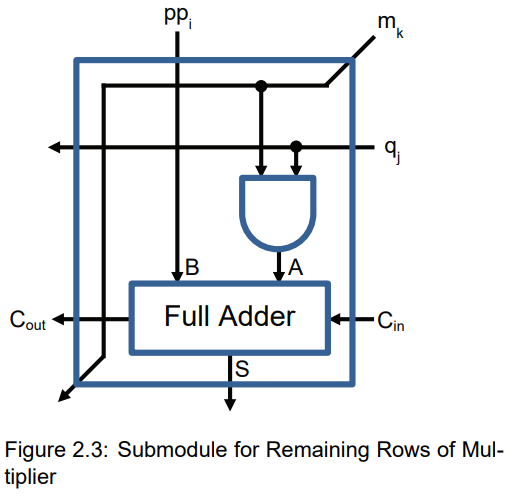
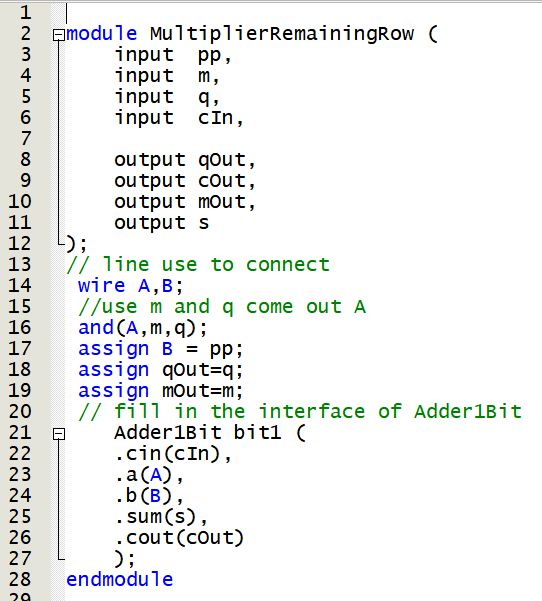
**Zhekun Huang**

**el23z2h@leeds.ac.uk**

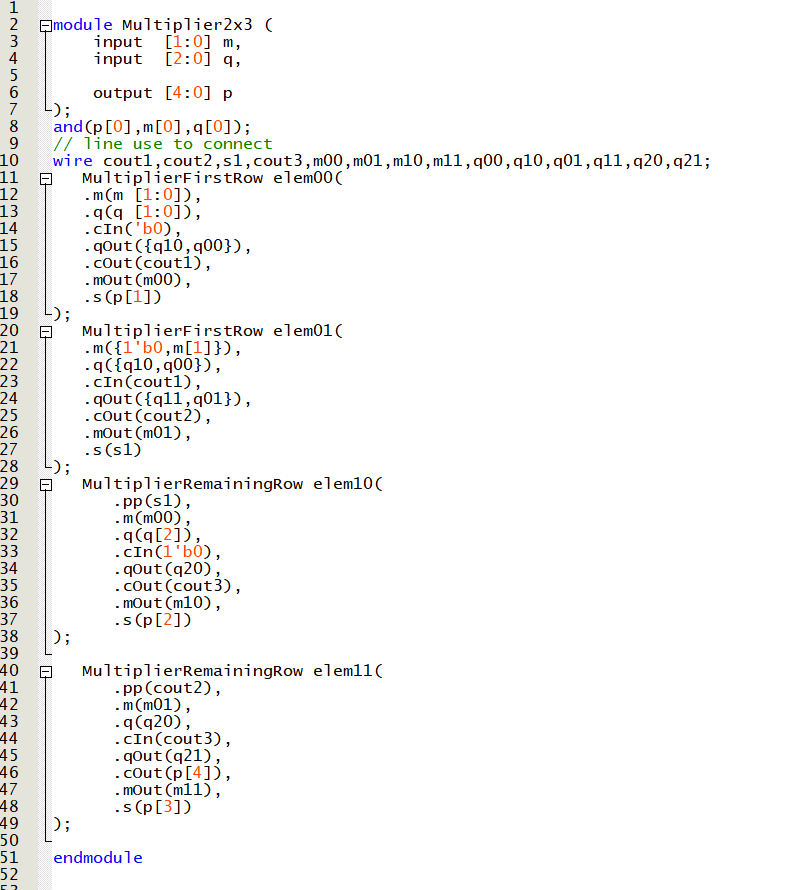
**GitHub username: HammondKun**

Firstly I fellow the assignment Details to finish the design of first row and remaining row

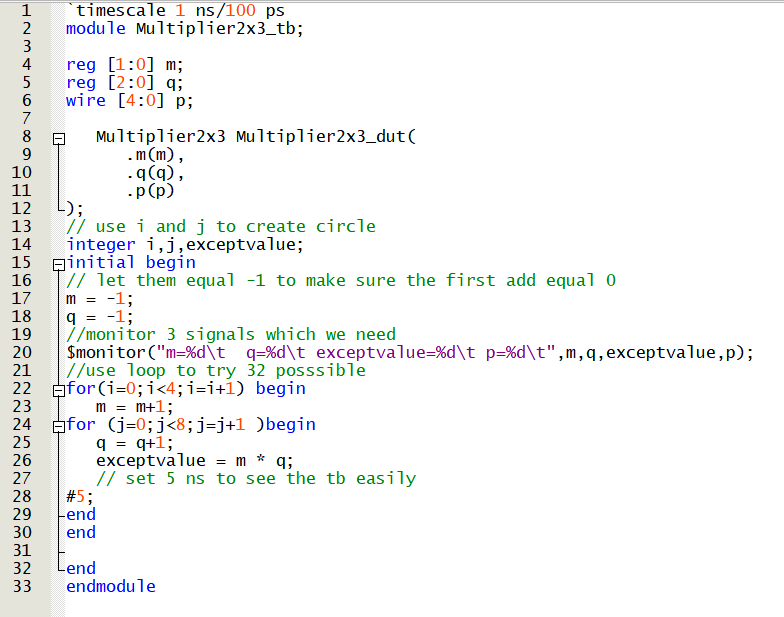




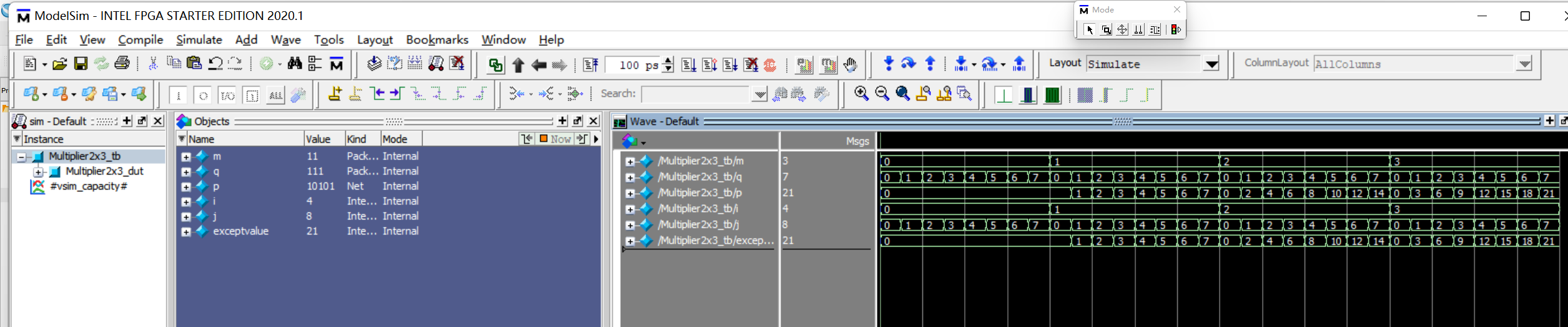
Then I fellow schematic diagram and mixed them up.



During the test bench, I use the for the achieve the loop



The result works well



**Appendix:**

**MultiplierFirstRow.v:**

module MultiplierFirstRow (

input [1:0] m,

input [1:0] q,

input cIn,

output [1:0] qOut,

output cOut,

output mOut,

output s

);

// line use to connect

wire A,B;

//use m q come out A B

and(A,m[0],q[1]);

and(B,m[1],q[0]);

assign mOut = m[0];

assign qOut[0] = q[0];

assign qOut[1] = q[1];

// fill in the interface of Adder1Bit

Adder1Bit bit0 (

.cin(cIn),

.a(A),

.b(B),

.sum(s),

.cout(cOut)

);

endmodule

**MultiplierRemainingRow.v:**

module MultiplierRemainingRow (

input pp,

input m,

input q,

input cIn,

output qOut,

output cOut,

output mOut,

output s

);

// line use to connect

wire A,B;

//use m and q come out A

and(A,m,q);

assign B = pp;

assign qOut=q;

assign mOut=m;

// fill in the interface of Adder1Bit

Adder1Bit bit1 (

.cin(cIn),

.a(A),

.b(B),

.sum(s),

.cout(cOut)

);

endmodule

**Multiplier2x3.v:**

module Multiplier2x3 (

input [1:0] m,

input [2:0] q,

output [4:0] p

);

and(p[0],m[0],q[0]);

// line use to connect

wire cout1,cout2,s1,cout3,m00,m01,m10,m11,q00,q10,q01,q11,q20,q21;

MultiplierFirstRow elem00(

.m(m [1:0]),

.q(q [1:0]),

.cIn('b0),

.qOut({q10,q00}),

.cOut(cout1),

.mOut(m00),

.s(p[1])

);

MultiplierFirstRow elem01(

.m({1'b0,m[1]}),

.q({q10,q00}),

.cIn(cout1),

.qOut({q11,q01}),

.cOut(cout2),

.mOut(m01),

.s(s1)

);

MultiplierRemainingRow elem10(

.pp(s1),

.m(m00),

.q(q[2]),

.cIn(1'b0),

.qOut(q20),

.cOut(cout3),

.mOut(m10),

.s(p[2])

);

MultiplierRemainingRow elem11(

.pp(cout2),

.m(m01),

.q(q20),

.cIn(cout3),

.qOut(q21),

.cOut(p[4]),

.mOut(m11),

.s(p[3])

);

endmodule

**Multiplier2x3\_tb.v:**

`timescale 1 ns/100 ps

module Multiplier2x3\_tb;

reg [1:0] m;

reg [2:0] q;

wire [4:0] p;

Multiplier2x3 Multiplier2x3\_dut(

.m(m),

.q(q),

.p(p)

);

// use i and j to create circle

integer i,j,exceptvalue;

initial begin

// let them equal -1 to make sure the first add equal 0

m = -1;

q = -1;

//monitor 3 signals which we need

$monitor("m=%d\t q=%d\t exceptvalue=%d\t p=%d\t",m,q,exceptvalue,p);

//use loop to try 32 posssible

for(i=0;i<4;i=i+1) begin

m = m+1;

for (j=0;j<8;j=j+1 )begin

q = q+1;

exceptvalue = m \* q;

// set 5 ns to see the tb easily

#5;

end

end

end

endmodule