```
// dadda multiplier
// A - 8 bits , B - 8bits, y(output) - 16bits

module dadda_8(A,B,y);


    input [7:0] A;
    input [7:0] B;
    output wire [15:0] y;
    wire  gen_pp [0:7][7:0];
// stage-1 sum and carry
    wire [0:5]s1,c1;
// stage-2 sum and carry
    wire [0:13]s2,c2;
// stage-3 sum and carry
    wire [0:9]s3,c3;
// stage-4 sum and carry
    wire [0:11]s4,c4;
// stage-5 sum and carry
    wire [0:13]s5,c5;




// generating partial products
genvar i;
genvar j;

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

   for(j = 0; j<8;j = j+1)begin
      assign gen_pp[i][j] = A[j]*B[i];
```

```
end

end

//Reduction by stages.

// di_values = 2,3,4,6,8,13...

//Stage 1 - reducing fom 8 to 6

   HA h1(.a(gen_pp[6][0]),.b(gen_pp[5][1]),.Sum(s1[0]),.Cout(c1[0]));

   HA h2(.a(gen_pp[4][3]),.b(gen_pp[3][4]),.Sum(s1[2]),.Cout(c1[2]));

   HA h3(.a(gen_pp[4][4]),.b(gen_pp[3][5]),.Sum(s1[4]),.Cout(c1[4]));

   csa_dadda c11(.A(gen_pp[7][0]),.B(gen_pp[6][1]),.Cin(gen_pp[5][2]),.Y(s1[1]),.Cout(c1[1]));

   csa_dadda c12(.A(gen_pp[7][1]),.B(gen_pp[6][2]),.Cin(gen_pp[5][3]),.Y(s1[3]),.Cout(c1[3]));

   csa_dadda c13(.A(gen_pp[7][2]),.B(gen_pp[6][3]),.Cin(gen_pp[5][4]),.Y(s1[5]),.Cout(c1[5]));

//Stage 2 - reducing fom 6 to 4

   HA h4(.a(gen_pp[4][0]),.b(gen_pp[3][1]),.Sum(s2[0]),.Cout(c2[0]));

   HA h5(.a(gen_pp[2][3]),.b(gen_pp[1][4]),.Sum(s2[2]),.Cout(c2[2]));

   csa_dadda c21(.A(gen_pp[5][0]),.B(gen_pp[4][1]),.Cin(gen_pp[3][2]),.Y(s2[1]),.Cout(c2[1]));

   csa_dadda c22(.A(s1[0]),.B(gen_pp[4][2]),.Cin(gen_pp[3][3]),.Y(s2[3]),.Cout(c2[3]));

   csa_dadda c23(.A(gen_pp[2][4]),.B(gen_pp[1][5]),.Cin(gen_pp[0][6]),.Y(s2[4]),.Cout(c2[4]));

   csa_dadda c24(.A(s1[1]),.B(s1[2]),.Cin(c1[0]),.Y(s2[5]),.Cout(c2[5]));

   csa_dadda c25(.A(gen_pp[2][5]),.B(gen_pp[1][6]),.Cin(gen_pp[0][7]),.Y(s2[6]),.Cout(c2[6]));
```

```
    csa_dadda c26(.A(s1[3]),.B(s1[4]),.Cin(c1[1]),.Y(s2[7]),.Cout(c2[7]));
    csa_dadda c27(.A(c1[2]),.B(gen_pp[2][6]),.Cin(gen_pp[1][7]),.Y(s2[8]),.Cout(c2[8]));
    csa_dadda c28(.A(s1[5]),.B(c1[3]),.Cin(c1[4]),.Y(s2[9]),.Cout(c2[9]));
    csa_dadda c29(.A(gen_pp[4][5]),.B(gen_pp[3][6]),.Cin(gen_pp[2][7]),.Y(s2[10]),.Cout(c2[10]));
    csa_dadda c210(.A(gen_pp[7][3]),.B(c1[5]),.Cin(gen_pp[6][4]),.Y(s2[11]),.Cout(c2[11]));
    csa_dadda c211(.A(gen_pp[5][5]),.B(gen_pp[4][6]),.Cin(gen_pp[3][7]),.Y(s2[12]),.Cout(c2[12]));
    csa_dadda c212(.A(gen_pp[7][4]),.B(gen_pp[6][5]),.Cin(gen_pp[5][6]),.Y(s2[13]),.Cout(c2[13]));

//Stage 3 - reducing fom 4 to 3

    HA h6(.a(gen_pp[3][0]),.b(gen_pp[2][1]),.Sum(s3[0]),.Cout(c3[0]));

    csa_dadda c31(.A(s2[0]),.B(gen_pp[2][2]),.Cin(gen_pp[1][3]),.Y(s3[1]),.Cout(c3[1]));
    csa_dadda c32(.A(s2[1]),.B(s2[2]),.Cin(c2[0]),.Y(s3[2]),.Cout(c3[2]));
    csa_dadda c33(.A(c2[1]),.B(c2[2]),.Cin(s2[3]),.Y(s3[3]),.Cout(c3[3]));
    csa_dadda c34(.A(c2[3]),.B(c2[4]),.Cin(s2[5]),.Y(s3[4]),.Cout(c3[4]));
    csa_dadda c35(.A(c2[5]),.B(c2[6]),.Cin(s2[7]),.Y(s3[5]),.Cout(c3[5]));
    csa_dadda c36(.A(c2[7]),.B(c2[8]),.Cin(s2[9]),.Y(s3[6]),.Cout(c3[6]));
    csa_dadda c37(.A(c2[9]),.B(c2[10]),.Cin(s2[11]),.Y(s3[7]),.Cout(c3[7]));
    csa_dadda c38(.A(c2[11]),.B(c2[12]),.Cin(s2[13]),.Y(s3[8]),.Cout(c3[8]));
    csa_dadda c39(.A(gen_pp[7][5]),.B(gen_pp[6][6]),.Cin(gen_pp[5][7]),.Y(s3[9]),.Cout(c3[9]));

//Stage 4 - reducing fom 3 to 2

    HA h7(.a(gen_pp[2][0]),.b(gen_pp[1][1]),.Sum(s4[0]),.Cout(c4[0]));


    csa_dadda c41(.A(s3[0]),.B(gen_pp[1][2]),.Cin(gen_pp[0][3]),.Y(s4[1]),.Cout(c4[1]));
    csa_dadda c42(.A(c3[0]),.B(s3[1]),.Cin(gen_pp[0][4]),.Y(s4[2]),.Cout(c4[2]));
    csa_dadda c43(.A(c3[1]),.B(s3[2]),.Cin(gen_pp[0][5]),.Y(s4[3]),.Cout(c4[3]));
    csa_dadda c44(.A(c3[2]),.B(s3[3]),.Cin(s2[4]),.Y(s4[4]),.Cout(c4[4]));
    csa_dadda c45(.A(c3[3]),.B(s3[4]),.Cin(s2[6]),.Y(s4[5]),.Cout(c4[5]));

    csa_dadda c46(.A(c3[4]),.B(s3[5]),.Cin(s2[8]),.Y(s4[6]),.Cout(c4[6]));

    csa_dadda c47(.A(c3[5]),.B(s3[6]),.Cin(s2[10]),.Y(s4[7]),.Cout(c4[7]));

    csa_dadda c48(.A(c3[6]),.B(s3[7]),.Cin(s2[12]),.Y(s4[8]),.Cout(c4[8]));

    csa_dadda c49(.A(c3[7]),.B(s3[8]),.Cin(gen_pp[4][7]),.Y(s4[9]),.Cout(c4[9]));

    csa_dadda c410(.A(c3[8]),.B(s3[9]),.Cin(c2[13]),.Y(s4[10]),.Cout(c4[10]));

    csa_dadda c411(.A(c3[9]),.B(gen_pp[7][6]),.Cin(gen_pp[6][7]),.Y(s4[11]),.Cout(c4[11]));


//Stage 5 - reducing fom 2 to 1

    // adding total sum and carry to get final output


    HA h8(.a(gen_pp[1][0]),.b(gen_pp[0][1]),.Sum(y[1]),.Cout(c5[0]));




    csa_dadda c51(.A(s4[0]),.B(gen_pp[0][2]),.Cin(c5[0]),.Y(y[2]),.Cout(c5[1]));

    csa_dadda c52(.A(c4[0]),.B(s4[1]),.Cin(c5[1]),.Y(y[3]),.Cout(c5[2]));

    csa_dadda c54(.A(c4[1]),.B(s4[2]),.Cin(c5[2]),.Y(y[4]),.Cout(c5[3]));

    csa_dadda c55(.A(c4[2]),.B(s4[3]),.Cin(c5[3]),.Y(y[5]),.Cout(c5[4]));

    csa_dadda c56(.A(c4[3]),.B(s4[4]),.Cin(c5[4]),.Y(y[6]),.Cout(c5[5]));

    csa_dadda c57(.A(c4[4]),.B(s4[5]),.Cin(c5[5]),.Y(y[7]),.Cout(c5[6]));

    csa_dadda c58(.A(c4[5]),.B(s4[6]),.Cin(c5[6]),.Y(y[8]),.Cout(c5[7]));

    csa_dadda c59(.A(c4[6]),.B(s4[7]),.Cin(c5[7]),.Y(y[9]),.Cout(c5[8]));

    csa_dadda c510(.A(c4[7]),.B(s4[8]),.Cin(c5[8]),.Y(y[10]),.Cout(c5[9]));

    csa_dadda c511(.A(c4[8]),.B(s4[9]),.Cin(c5[9]),.Y(y[11]),.Cout(c5[10]));

    csa_dadda c512(.A(c4[9]),.B(s4[10]),.Cin(c5[10]),.Y(y[12]),.Cout(c5[11]));

    csa_dadda c513(.A(c4[10]),.B(s4[11]),.Cin(c5[11]),.Y(y[13]),.Cout(c5[12]));

    csa_dadda c514(.A(c4[11]),.B(gen_pp[7][7]),.Cin(c5[12]),.Y(y[14]),.Cout(c5[13]));

    assign y[0] =  gen_pp[0][0];

    assign y[15] = c5[13]

endmodule
```

```
//self checking test bench for 8*8 dadda

module dadda_8TB();

parameter M=8,N=8;


reg [N-1:0]A;

reg [M-1:0]B;

wire [N+M-1:0]y;


//---- Instantiation of main test module----

//Array_MUL_USign #(64,64) UUT(A,B,Y); //M=4,N=6

 dadda_8 UUT(.A(A),.B(B),.y(Y));




// initializing the inputs to the test module

// initial block executes only once

initial

repeat(15)

begin

        #10 A = $random; B = $random;

        #100//give required simulation time to complete the operation one by one.

        #100

        #10

        //-----VERIFICATION OF THE OBTAINED RESULT WITH EXISTING RESULT------

        $display(" A=%d,B=%d,AxB=%d",(A),(B),(y));


        if( (A)*(B) != (Y))

                $display(" *ERROR* ");


end


endmodule
```