Source code for vedic multiplier using Carry Save Adder(CSA)

----------------Design code-------------

```
module vedic_multiplier_4x4(a,b,pp);
   input [3:0] a,b;
   output [7:0]pp;
   wire [3:0]p,q,r,s;
   wire [3:0]s1,c3;
   wire c1,c2,c4;
   vedic_multiplier_2x2 vm0(a[1:0],b[1:0],p);
   vedic_multiplier_2x2 vm1(a[1:0],b[3:2],q);
   vedic_multiplier_2x2 vm2(a[3:2],b[1:0],r);
   vedic_multiplier_2x2 vm3(a[3:2],b[3:2],s);
   csa1 ca1(r,q,p[3:2],s1,c1);
   csa2 ca2(s,s1[3:2],c1,c3);
   assign pp={c3,s1[1:0],p[1:0]};
endmodule


module vedic_multiplier_2x2(a,b,p);
   input [1:0] a, b;
   output [3:0] p;
   wire a0b0, a0b1, a1b0, a1b1, sum1, carry1, sum2, carry2;
   assign a0b0 = a[0] & b[0];
   assign a0b1 = a[0] & b[1];
   assign a1b0 = a[1] & b[0];
   assign a1b1 = a[1] & b[1];
   half_add ha1 (a0b1, a1b0, sum1, carry1);
   half_add ha2 (sum1, 0, sum2, carry2);
   assign p[0] = a0b0;
   assign p[1] = sum1;
   assign p[2] = a1b1 ^ carry1;
   assign p[3] = a1b1 & carry1;
endmodule
```

```
module csa1(a,b,cin,s,cout); //CSA1
   input [3:0] a,b;
   input [1:0]cin;
   output [3:0] s;
   output cout;
   wire [3:0]s1,c1,c2;
   full_add fa11(a[0],b[0],cin[0],s1[0],c1[0]);
   full_add fa12(a[1],b[1],cin[1],s1[1],c1[1]);
   half_add ha11(a[2],b[2],s1[2],c1[2]);
   half_add ha12(a[3],b[3],s1[3],c1[3]);
   assign s[0]=s1[0];
   half_add ha21(s1[1],c1[0],s[1],c2[0]);
   full_add fa21(s1[2],c1[1],c2[0],s[2],c2[1]);
   full_add fa22(s1[3],c1[2],c2[1],s[3],c2[2]);
   half_add ha22(c1[3],c2[2],cout,c2[3]);
endmodule
module csa2(a,b,c,s); //CSA2
     input [3:0] a;
     input [1:0] b;
     output c;
     output [3:0] s;
     wire [2:0] s1,c1;
     wire [2:0] c2;
     half_add ha11(a[0],b[0],s1[0],c1[0]);
     half_add ha12(a[1],b[1],s1[1],c1[1]);
     half_add ha13(a[2],c,s1[2],c1[2]);
     assign s[0]=s1[0];
     half_add ha21(c1[0],s1[1],s[1],c2[0]);
     full_add fa21(c1[1],s1[2],c2[0],s[2],c2[1]);
     full_add fa22(c1[2],a[3],c2[1],s[3],c2[2]);
endmodule
```

```
module half_add(a,b,sum,carry);
   input a,b;
   output sum,carry;
   assign sum = a ^ b;
   assign carry = a & b;
endmodule
module full_add(a,b,cin,sum,carry);
   input a,b,cin;
   output sum,carry;
   assign sum=a^b^cin;
   assign carry =(a&b)|(a&cin)|(b&cin);
endmodule
```

--------testbench--------

```
module vedic4x4_tb;
   reg [3:0]a,b;
   wire [7:0]pp;
vedic_multiplier_4x4 DUT(a,b,pp);
initial
begin
a=4'd0;b=4'd0;
#10 a=4'd6;
#20 a=4'd15;b=4'd15;
#200 $finish;
end
always #10 b=b+4'd1;
endmodule
```