```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:    00:54:18 09/18/2015
// Design Name:
// Module Name:    BCDadder
// Project Name:
// Target Devices:
// Tool versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
module BCDadder(A,B,Cin,sM,CoutM,CinM,Couts);


	input	[3:0] A;			//BCDadder total input
	input	[3:0] B;
	input	Cin,CinM;
	output [3:0] sM;			 //BCDadder total output
	output	CoutM,Couts;

	 wire [3:0] s;
	 wire [3:0] P ,PM;			//The connection between Logic gate inside a
Fulladder block
	 wire [3:0] G, GM;			 //The connection between Logic gate inside a
Fulladder block
	 wire [3:0] K, KM;			//The connection between Logic gate inside a
Fulladder block
	 wire	Acout1;
```

```
          wire    Acout2;
          wire    Cout,Couts;
            wire    c0,c1,c2,c0M,c1M,c2M;                           //the connection
between Fulladder blocks
            //cin[0]=0;
            //cinM[0]=0;


          xor E10(P[0],A[0],B[0]);              //the Logic circuit of Fulladder 0
          and A10(G[0],A[0],B[0]);
            xor E20(s[0],P[0],0);
            and A20(K[0],P[0],0);
            or    O10(c0,K[0],G[0]);
            //cin[1]=c0;
            //cinM[1]=c0M;


            xor E11(P[1],A[1],B[1]);                //the Logic circuit of Fulladder 1
          and A11(G[1],A[1],B[1]);
            xor E21(s[1],P[1],c0);
            and A21(K[1],P[1],c0);
            or    O11(c1,K[1],G[1]);
            //cin[2]=c1;
            //cinM[2]=c1M;


            xor E12(P[2],A[2],B[2]);                //the Logic circuit of Fulladder 2
          and A12(G[2],A[2],B[2]);
            xor E22(s[2],P[2],c1);
            and A22(K[2],P[2],c1);
            or    O12(c2,K[2],G[2]);
            //cin[3]=c2;
            //cinM[3]=c2M;


            xor E13(P[3],A[3],B[3]);                //the Logic circuit of Fulladder 3
          and A13(G[3],A[3],B[3]);
            xor E23(s[3],P[3],c2);
```

```
        and A23(K[3],P[3],c2);
        or   O13(Cout,K[3],G[3]);




        and A1(Acout1,s[3],s[2]);           //correction circuit
        and A2(Acout2,s[3],s[1]);
        or   Os(Couts,Acout1,Acout2,Cout);

        // AM[0]=s[0];
        // AM[1]=s[1];
        // AM[2]=s[2];
        // AM[3]=s[3];
        // BM[0]=0;
        // BM[1]=1;
        // BM[2]=1;
        // BM[3]=0;



        //CinM[0]=0;


        xor ME10(PM[0],s[0],0);             //the Logic circuit of Fulladder 0   for
correction
       and MA10(GM[0],s[0],0);
        xor ME20(sM[0],PM[0],CinM);
        and MA20(KM[0],PM[0],CinM);
        or   MO10(c0M,KM[0],GM[0]);
        //CinM[1]=c0M;


        xor ME11(PM[1],s[1],Couts);               //the Logic circuit of Fulladder 1 for
correction
       and MA11(GM[1],s[1],Couts);
        xor ME21(sM[1],PM[1],c0M);
```

```
            and MA21(KM[1],PM[1],c0M);
            or    MO11(c1M,KM[1],GM[1]);
            //CinM[2]=c1M;

            xor ME12(PM[2],s[2],Couts);                  //the Logic circuit of Fulladder 2 for
correction
          and MA12(GM[2],s[2],Couts);
            xor ME22(sM[2],PM[2],c1M);
            and MA22(KM[2],PM[2],c1M);
            or    MO12(c2M,KM[2],GM[2]);
            //CinM[3]=c2M;

            xor ME13(PM[3],s[3],0);                  //the Logic circuit of Fulladder 3 for
correction
          and MA13(GM[3],s[3],0);
            xor ME23(sM[3],PM[3],c2M);
            and MA23(KM[3],PM[3],c2M);
            or    MO13(CoutM,KM[3],GM[3]);

            //CoutM=x;

endmodule
```