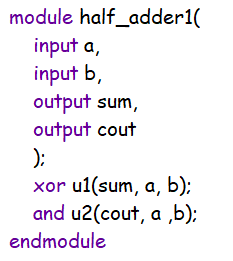
**半加器：**

**结构描述：**



**行为描述：**

module half\_adder2(

input a,

input b,

output sum,

output cout

);

reg sum,cout;

always @(a or b)

begin

case ({a, b})

2'b00:begin

sum = 0;

cout = 0;

end

2'b01:begin

sum = 1;

cout = 0;

end

2'b10:begin

sum = 1;

cout = 0;

end

2'b11:begin

sum = 0;

cout = 1;

end

endcase

end

endmodule

**数据流描述：**

module half\_adder3(

input a,

input b,

output sum,

output cout

);

assign sum = a ^ b;

assign cout = a & b;

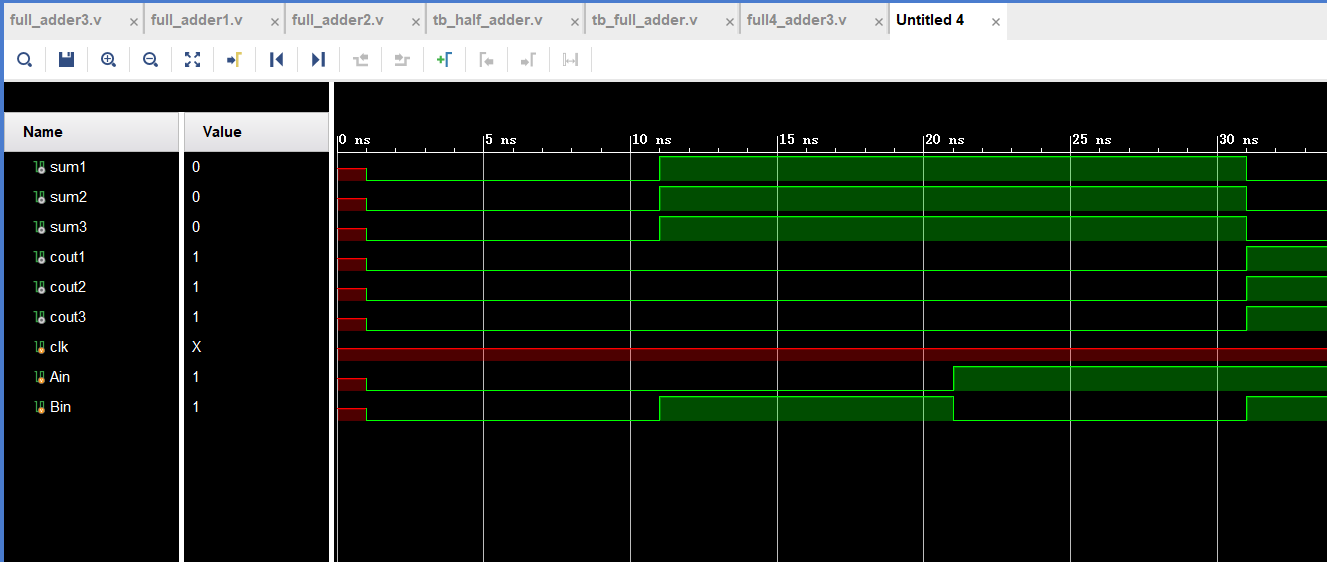
endmodule

半加器的和与进位表达式为：

Sum = a ^ b;

Cout = a & b;

仿真结果：



**全加器：**

**结构描述**：

module full\_adder1(

input a,

input b,

input cin,

output sum,

output cout

);

wire s1,m1,m2,m3;

xor u1(s1, a, b);

xor u2(sum, s1, cin);

and (m1, a, b);

and (m2, a, cin);

and (m3, b ,cin);

or (cout, m1, m2, m3);

endmodule

**行为描述**：

module full\_adder2(

input a,

input b,

input cin,

output sum,

output cout

);

reg sum, cout;

always @(a or b or cin)

case ({a, b, cin})

3'b000:begin

sum = 0;

cout = 0;

end

3'b001:begin

sum = 1;

cout = 0;

end

3'b010:begin

sum = 1;

cout = 0;

end

3'b011:begin

sum = 0;

cout = 1;

end

3'b100:begin

sum = 1;

cout = 0;

end

3'b101:begin

sum = 0;

cout = 1;

end

3'b110:begin

sum = 0;

cout = 1;

end

3'b111:begin

sum = 1;

cout = 1;

end

endcase

endmodule

数据流描述：

module full\_adder3(

input a,

input b,

input cin,

output sum,

output cout

);

assign sum = a ^ b^ cin;

assign cout = (a & b) | (cin & (a ^ b));

endmodule

仿真结果：



串行四位全加器

代码：

module full4\_adder\_serial(

input [3:0] a,b,

input cin,

output [3:0] sum,

output cout

);

wire cin1, cin2, cin3;

full\_adder3 fadder3\_1(a[0], b[0], cin, sum[0], cin1);

full\_adder3 fadder3\_2(a[1], b[1], cin1, sum[1], cin2);

full\_adder3 fadder3\_3(a[2], b[2], cin2, sum[2], cin3);

full\_adder3 fadder3\_4(a[3], b[3], cin3, sum[3], cout);

endmodule

测试代码：

module tb\_full4\_adder();

wire [3:0] Sum;

wire cout;

reg [3:0] Ain;

reg [3:0] Bin;

reg cin;

initial

begin

//0000 0001 1

Ain = 4'b0000;

Bin = 4'b0001;

cin = 1;

//0001 0011 0

#10

Ain = 4'b0001;

Bin = 4'b0011;

cin = 0;

//0111 0010 1

#10

Ain = 4'b0111;

Bin = 4'b0010;

cin = 1;

//1111 1111 0

#10

Ain = 4'b1111;

Bin = 4'b1111;

cin = 0;

#10 $stop;

end

// full4\_adder3 f4adder3(Ain, Bin, cin, Sum, cout);

full4\_adder\_serial fadder\_serial(Ain, Bin, cin, Sum, cout);

endmodule

仿真结果：

