Verilog HDL
设计举例

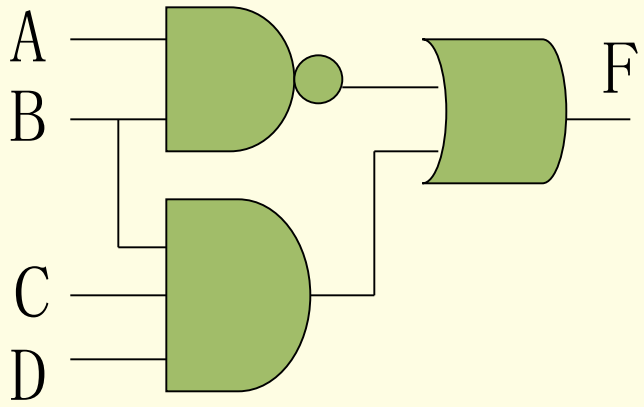
常用组合电路模块的设计

【例1】基本门电路的几种描述方法

F=(A•B)’+B • C • D

(1)调用门原语

```
module gate1(F,A,B,C,D);
output F;
input A,B,C,D;
wire F1,F2;
nand(F1,A,B);
and(F2,B,C,D);
or(F,F1,F2);
endmodule
```



基本门电路的几种描述方法

(2)用assign连续赋值语句描述

```
module gate2(F,A,B,C,D);
output F;
input A,B,C,D;
assign F=~(A&B)|(B&C&D);
endmodule
```

基本门电路的几种描述方法

(3)用过程赋值语句

```
module gate3(F,A,B,C,D);
output F;
input A,B,C,D;
reg F;
always @(A or B or C or D)
    begin
        F=~(A&B)|(B&C&D);
    end
endmodule
```

4位全加器

【例2】 4位全加器

```
module  add4 ( cout,sum,a,b,cin );
output  cout;
output [3:0] sum;
input [3:0]  a,b;
input   cin;
   assign {cout,sum}=a+b+cin;
endmodule
```

3-8译码器

【例3】3-8译码器

```
module decoder_38(out,in,en);
 output [7:0]out;
 input [2:0]in;
 input en;
 reg [7:0]out;
 always @(in or en)
   begin
     if (en)
       out=8'b11111111;
     else
       case(in)
         3'b000: out=8'b11111110;
         3'b001: out=8'b11111101;
         3'b010: out=8'b11111011;
         3'b011: out=8'b11110111;
         3'b100: out=8'b11101111;
         3'b101: out=8'b11011111;
         3'b110: out=8'b10111111;
         3'b111: out=8'b01111111;
       endcase
   end
endmodule
```

数值比较器

【例4】4位数值比较器

```
module   compare4 ( equal,a,b );
output  equal;
input [3:0] a,b;
   assign  equal=(a==b)? 1:0;
/*如果两个输入信号相等, 输出为1,
否则为0*/
endmodule
```

多路选择器1

【例5】 2选1 MUX

```
module  mux2_1 ( out,a,b, sel );
output  out;
input  a,b,sel;
assign  out=sel ? a : b;
endmodule
```

多路选择器2

【例6】 4选1 MUX

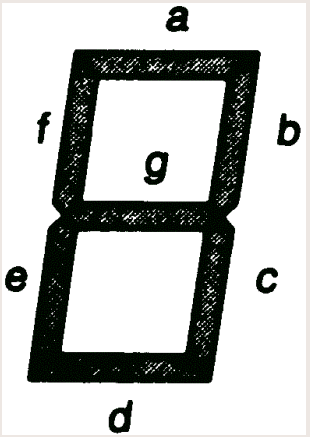
```
 module  mux4_1 ( out,in0,in1,in2,in3,sel );
output  out;
input  in0,in1,in2,in3;
input [1:0]sel;
reg out;
always @(in0 or in1 or in2 or in3 or sel)
  begin
     case(sel)
        2’b00: out=in0;
        2’b01: out=in1;
        2’b10: out=in2;
        default: out=in3;
     endcase
  end
endmodule
```

奇偶校验产生器

【例7】奇偶校验产生器

```
module  parity(even,odd, data );
output  even,odd;
input[7:0] data;
assign  odd=^data;    //产生奇校验位
assign  even=~odd;    //产生偶校验位
endmodule
```

7段LED数码管译码器



【例8】 7段LED数码管译码器

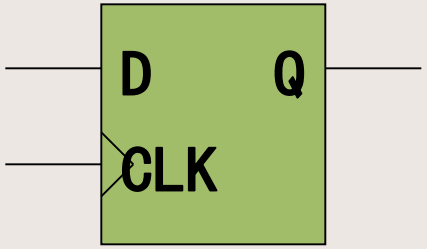
```
module decode4_7(a,b,c,d,e,f,g,din);
output a,b,c,d,e,f,g;
input [3:0]din;  //输入的4位BCD码
reg a,b,c,d,e,f,g;
always @(din)
   begin
      case(din)
         4'b0000:{a,b,c,d,e,f,g}=7'b1111110;
         4'b0001:{a,b,c,d,e,f,g}=7'b0110000;
```

7段LED数码管译码器

```
    4’b0010:{a,b,c,d,e,f,g}=7’b1101101;
     4’b0011:{a,b,c,d,e,f,g}=7’b1111001;
     4’b0100:{a,b,c,d,e,f,g}=7’b0110011;
     4’b0101:{a,b,c,d,e,f,g}=7’b1011011;
     4’b0110:{a,b,c,d,e,f,g}=7’b1011111;
     4’b0111:{a,b,c,d,e,f,g}=7’b1110000;
     4’b1000:{a,b,c,d,e,f,g}=7’b1111111;
     4’b1001:{a,b,c,d,e,f,g}=7’b1111011;
     default:{a,b,c,d,e,f,g}=7’b0000000;
   endcase
  end
endmodule
```

常用时序电路模块的设计

【例9】基本D触发器




```
module DFF(Q,D,CLK);
output Q;
input D,CLK;
reg Q;
always @(posedge CLK)
  begin
     Q=D;
   end
endmodule
```

带同步清0、置1端的D触发器

【例10】带同步清0、置1端(高电平有效)的D触发器

```
module DFF2(Q,QN,D,CLK,SET,RESET);
output  Q,QN;
input  D,CLK,SET,RESET;
reg  Q,QN;
always @(posedge CLK)
  begin
    if (RESET)     begin   Q=0; QN=1;    end
    else if(SET)   begin   Q=1; QN=0;    end
    else           begin   Q=D;QN=~D; end
   end
endmodule
```

带异步清0、置1端的D触发器

【例10】带异步清0、置1端(高电平有效)的D触发器

```
module DFF2(Q,QN,D,CLK,SET,RESET);
output  Q,QN;
input  D,CLK,SET,RESET;
reg  Q,QN;
always @(SET or RESET)
  begin
    if (RESET)      begin    Q=0; QN=1;    end
    else if(SET)    begin    Q=1; QN=0;    end
  end
always @(posedge CLK)
  begin
   Q=D;QN=~D;
   end
endmodule
```

数据锁存器

【例11】电平敏感的一位数据锁存器

```
module latch_1(Q,D,clk);
output Q;
input D,clk;
assign Q=clk ? D : Q;
/*
  在时钟信号为高电平时，将输入端数据锁存；
  否则，在时钟信号为低电平时，输出端保持不变。
*/
endmodule
```

数据锁存器

【例12】8位数据锁存器

```
module latch_8(Q,D,clk);
output  [7:0]Q;
input  [7:0]D;
input  clk;
 reg  [7:0]Q;
 always @(clk or D)
    begin
      if (clk)
        Q=D;
    end
endmodule
```

数据寄存器

【例13】8位数据寄存器（异步清0）

```
module  reg8(dout,din,clk,clr);
output [7:0]dout;
input [7:0]din;
input clk,clr;
reg [7:0]dout;
always @(posedge clk or posedge clr)
  begin
    if (clr)   dout=0;
    else       dout=din;
  end
endmodule
```

移位寄存器

【例14】8位移位寄存器（同步清0）

```
module  shifter8(dout,din,clk,clr);
output [7:0]dout;
input din,clk,clr;
reg [7:0]dout;
always @(posedge clk)
  begin
     if (clr)   dout=8'b00000000;
     else       begin
                   dout=dout<<1;
                   dout[0]=din;
                end
  end
endmodule
```

计数器

【例15】4位同步清0、同步预置二进制计数器

```
module   counter4(out,cout,data,load,clr,clk);
output[3:0]  out;
output  cout;
input[3:0]   data;
input  load, clr,clk;
reg[3:0]  out;
always @(posedge clk)
   begin
      if  (!clr)  out=0;
      else if  (load)  out=data;
      else  out =out+1;
   end
assign  cout=&out;
endmodule
```

计数器

【例16】4位同步清0、同步预置十进制计数器

```
module   counter10(out,cout,data,load,clr,clk);
output[3:0]  out;
output  cout;
input[3:0]   data;
input  load, clr,clk;
reg[3:0]  out;
always @(posedge clk)
   begin
      if  (!clr)  out=0;
      else if  (load)  out=data;
      else if  (out==9) out=0;
      else  out =out+1;
   end
assign  cout=(out==9)?1:0;
endmodule
```

小结

1. 用Verilog HDL设计组合电路模块。
2. 用Verilog HDL设计常用的时序电路模块。