

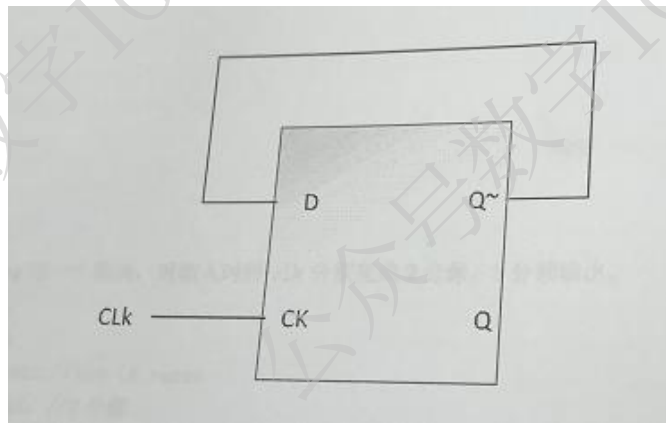
简述 ASIC 设计流程的主要步骤？

芯片架构（需求分析）、RTL 设计、功能仿真、FPGA 验证、综合、sta、P&R、流片

求最大时钟频率？

$T_{su} = 2ns$, $T_{hold} = 1ns$, $T_{cq} = 2ns$;

T_{cq} is the delay from Dff 's CK pin to Q` pin.



$T_{min} = T_{cq} + T_{su} = 4ns$;

$f_{max} = 1/T_{min} = 250MHz$;

同步复位和异步复位的区别？

异步复位：无论有时钟沿，只要复位信号有效，系统就进行复位。

同步复位：复位信号只有在时钟上升沿到来，才有效。

什么是竞争和冒险？怎么判断？如何消除？

信号由于经由不同路径传输达到某一汇合点的时间有

先有后的现象，就称之为竞争，由于竞争现象所引起的电路输出发生瞬间错误的现象，

就称之为冒险，FPGA 设计中

最简单的避免方法是尽量使用时序逻辑同步输入输出。

加滤波电容，消除毛刺的影响

加选通信号，避开毛刺

增加冗余项，消除逻辑冒险。

什么是亚稳态，如何防止亚稳态？

亚稳态：就是触发器输出处于不稳定状态，输出可能为 1 或 0；

亚稳态就是由于时序不满足而导致的触发器的输出处于不稳定的状态，这个状态可能

是 0 或 1，也有可能是非 0 非 1 的中间状态，以及处于跳变的状态等。

解决办法：

同步时序逻辑中，改善时序，例如可以对设计进行适当的约束等；

在跨时钟传输过程中，可以使用单比特同步器（打两拍），异步 FIFO；

单比特，使用同步器

多比特，异步 FIFO，握手信号

用 Verilog 写一个模块，堆输入时钟 clk 分别完成 2 分频，3 分频输出。

```
module fre_div(
```

```
input clk,
```

```
input resetn,
```

```
output clk2,
```

```
output clk3
```

```
);
```

```
//your code starts here:
```

```
endmodule
```

```
/* module fre3(
```

```
input clk,
```

```
input reset,
```

```
output clk_div2,
```

```
output clk_div3
```

```
);
```

```
reg q0;
```

```
reg qq0, qq1;
```

```
wire in1, in2;
```

```
//二分频
```

```
always@(posedge clk or posedge reset) begin
```

```
    if(reset) q0 <= 0;
```

```
    else q0 <= in1;
```

```
end
```

```
assign in1 = ~q0;
```

```
assign clk_div2 = q0;

always@(posedge clk or posedge reset) begin

    if(reset) begin

        qq0 <= 0;

        qq1 <= 0;

    end

    else begin

        qq0 <= in2;

        qq1 <= qq0;

    end

end

assign in2 = ~qq0 & ~qq1;

assign clk_div3 = qq0;

endmodule */
```

```
module fre3(

    input clk,

    input reset,
```

```

output clk_div2,
output clk_div3
);

//二分频

always@(posedge clk or posedge reset) begin

    if(reset) q0 <= 0;

    else q0 <= ~q0;

end

assign clk_div2 = q0;

//三分频

//-----占空比为 1:3 的 3 分频开始-----

//计数器

reg [1:0] cnt;

always@(posedge clk or posedge reset) begin

    if(reset) cnt <= 0;

    else if(cnt <= 2) cnt <= cnt + 1;

    else cnt <= 0;

end

reg clk_mid_fre3;

always@(posedge clk or posedge reset) begin

    if(reset) clk_mid_fre3 <= 1;

```

```

        else if(cnt == 0) clk_mid_fre3 <= ~clk_mid_fre3;
        else if(cnt == 2) clk_mid_fre3 <= ~clk_mid_fre3;
        else clk_mid_fre3 <= clk_mid_fre3;
    end

    //-----占空比为 1:3 的 3 分频结束-----

    //下降沿采样

    reg clk_mid_fre3n;

    always@(negedge clk or posedge reset) begin
        if(reset) clk_mid_fre3n <= 1;
        else clk_mid_fre3n <= clk_mid_fre3;
    end

    assign clk_div3 = clk_mid_fre3 | clk_mid_fre3n;

endmodule

```

脉冲同步器的基本功能是从某个时钟域取出一个单时钟宽度脉冲，然后在新的时钟域中建立另一个单时钟宽度脉冲，请按照下面的模块接口信号设计一个脉冲检测器，并说明你设计的脉冲同步器的使用限制：

```

module sync_pulse(
    input in_rst_n,
    input in_clk,
    input in_pulse,

```

```
input out_rst_n,  
input out_clk,  
output out_pulse  
);  
  
//.....  
  
endmodule
```

```
module Sync_Pulse(  
input    clka,  
input    clkb,  
input    rst_n,  
input    pulse_ina,  
output   pulse_outb  
);
```

```
//-----  
reg    signal_a;  
reg    signal_b;  
reg    signal_b_r;  
reg    signal_b_rr;  
reg    signal_a_r;  
reg    signal_a_rr;  
  
//-----
```

//在 clka 下, 生成展宽信号 signal_a

```
always @(posedge clka or negedge rst_n)begin
```

```
    if(rst_n == 1'b0)begin
```

```
        signal_a <= 1'b0;
```

```
    end
```

```
    else if(pulse_ina == 1'b1)begin
```

```
        signal_a <= 1'b1;
```

```
    end
```

```
    else if(signal_a_rr == 1'b1)
```

```
        signal_a <= 1'b0;
```

```
    else
```

```
        signal_a <= signal_a;
```

```
end
```

```
//-----
```

//在 clkb 下同步 signal_a

```
always @(posedge clkb or negedge rst_n)begin
```

```
    if(rst_n == 1'b0)begin
```

```
        signal_b <= 1'b0;
```

```
    end
```

```
    else begin
```

```
        signal_b <= signal_a;
```

```
    end
```



```

end

//-----

//在 clkb 下生成脉冲信号和输出信号

always @(posedge clkb or negedge rst_n)begin

    if(rst_n == 1'b0)begin

        signal_b_r <= 'b0;

        signal_b_rr <= 'b0;

    end

    else begin

        signal_b_rr <= signal_b_r;

        signal_b_r <= signal_b;

    end

end

end

assign pulse_outb = ~signal_b_rr & signal_b_r;

//-----

//在 clka 下采集 signal_b_rr, 生成 signal_a_rr 用于反馈拉低 signal_a

always @(posedge clka or negedge rst_n)begin

    if(rst_n == 1'b0)begin

        signal_a_r <= 'b0;

        signal_a_rr <= 'b0;

    end

    else begin

```

```
    signal_a_rr <= signal_a_r;  
    signal_a_r <= signal_b_rr;  
end  
end  
endmodule
```

Regular expression (RE) in perl/python is used to find/match character strings.

Fill out the responding special words in RE:

Pattern to match

- 1) any character except new line
- 2) start of a string
- 3) empty string
- 4) any decimal digits
- 5) whitespace
- 6) to match 0 or more repetitions of the preceding RE

RE special characters

- 1)
- 2)
- 3)
- 4)

5)

6)

编译器编译 C 语言程序的主要过程？

主要包括预处理，编译，汇编，链接过程

在计算机程序中，堆和栈的区别是什么？

栈由系统分配，堆由用户申请分配

在 C 语言中，函数通常只能输出一个值，如果希望有多个值输出，你能想到哪些方法呢？

(1) 将多个输出值封装在结构体中，将结构体输出；(2) 通过函数参数输出

请解释程序中递归的概念？（验证岗位必答）

递归是程序调用自身的一种编程技巧，它体现了算法设计中分而治之的思想，将原问题转化为与原问题相类似的较小问题，大大减少了代码量

验证的目的主要是为了保证什么？（至少说出三种，验证岗位必答）

保证设计的正确性、证明没有 bug、验证功能点完全、代码覆盖率完全、性能达标

你在以前的学习或工作中遇到的与验证相关的问题有哪些？说说你对验证的理解，需要掌握什么技能，搭建验证环境需要注意什么？（验证岗位必答）

三、加分题（选答）

众所周知，海光被美国列入实体清单，如果你是海光的战略顾问，根据当前的国际局势以及公司现状，你会给海光高层哪些建议以辅助进行公司决策？至少列出 5 条建议。

公司应该考虑；

- 1) 可持续发展，加大对应届生的招聘与培养；
- 2) 鼓励各种创新方式，继续加大创新投资；
- 3) 加强与国内企业的合作，强化自身的生存能力；
- 4) ...