1：什么是同步逻辑和异步逻辑？.....2
2：同步电路和异步电路的区别：.....2
3：时序设计的实质：.....2
4：建立时间与保持时间的概念？.....2
5：为什么触发器要满足建立时间和保持时间？.....2
6：什么是亚稳态？为什么两级触发器可以防止亚稳态传播？.....3
7：系统最高速度计算（最快时钟频率）和流水线设计思想：.....3
8：时序约束的概念和基本策略？.....3
9：附加约束的作用？.....4
10：FPGA 设计工程师努力的方向：.....4
11：对于多位的异步信号如何进行同步？.....4
12：FPGA 和 CPLD 的区别？.....4
13：锁存器（latch）和触发器（flip-flop）区别？.....5
14：FPGA 芯片内有哪两种存储器资源？.....5
15：什么是时钟抖动？.....5
16：FPGA 设计中对时钟的使用？（例如分频等）.....5
17：FPGA 设计中如何实现同步时序电路的延时？.....5
18：FPGA 中可以综合实现为 RAM/ROM/CAM 的三种资源及其注意事项？.....5
19：什么是"线与"逻辑,要实现它,在硬件特性上有什么具体要求?.....6
20：什么是竞争与冒险现象?怎样判断?如何消除?.....6
21.用 Verilog 设计一个 5 分频器。.....6
22.用状态机检测序列为 11001101，输出为 1，否则输出为 0。.....8
23. IC 设计中同步复位与 异步复位的区别.....10
24. MOORE 与 MEELEY 状态机的特征.....10
25. 画状态机，并用 verilog 实现接受 1,2,5 分钱的卖报机，每份报纸 5 分钱。...10
26. FPGA 的基本结构.....12
27. 程序下载到 FPGA 的方式有哪几种，JTAG 有哪几条线。.....13
28. 时钟周期为 T，触发器 D1 的建立时间最大为 T1max，最小为 T1min。组合逻辑电路最大延迟为 T2max，最小为 T2min。问：触发器 D2 的建立时间 T3 和保持时间 T4 应满足什么条件？.....14

1：什么是同步逻辑和异步逻辑？

同步逻辑是时钟之间有固定的因果关系。异步逻辑是各时钟之间没有固定的因果关系。答案应该与上面问题一致

〔补充〕：同步时序逻辑电路的特点：各触发器的时钟端全部连接在一起，并接在系统时钟端，只有当时钟脉冲到来时，电路的状态才能改变。改变后的状态将一直保持到下一个时钟脉冲的到来，此时无论外部输入 x 有无变化，状态表中的每个状态都是稳定的。

异步时序逻辑电路的特点：电路中除可以使用带时钟的触发器外，还可以使用不带时钟的触发器和延迟元件作为存储元件，电路中没有统一的时钟，电路状态的改变由外部输入的变化直接引起。

2：同步电路和异步电路的区别：

同步电路：存储电路中所有触发器的时钟输入端都接同一个时钟脉冲源，因而所有触发器的状态的变化都与所加的时钟脉冲信号同步。

异步电路：电路没有统一的时钟，有些触发器的时钟输入端与时钟脉冲源相连，这有这些触发器的状态变化与时钟脉冲同步，而其他的触发器的状态变化不与时钟脉冲同步。

3：时序设计的实质：

电路设计的难点在时序设计，时序设计的实质就是满足每一个触发器的建立/保持时间的而要求。

4：建立时间与保持时间的概念？

建立时间：触发器在时钟上升沿到来之前，其数据输入端的数据必须保持不变的时间。

保持时间：触发器在时钟上升沿到来之后，其数据输入端的数据必须保持不变的时间。

5：为什么触发器要满足建立时间和保持时间？

因为触发器内部数据的形成是需要一定的时间的，如果不满足建立和保持时间，触发器将进入亚稳态，进入亚稳态后触发器的输出将不稳定，在 0 和 1 之间变化，这时需要经过一个恢复时间，其输出才能稳定，但稳定后的值并不一定是你的输入值。这就是为什么要用两级触发器来同步异步输入信号。这样做可以防止由于异步输入信号对于本级时钟可能不满足建立保持时间而使本级触发器产生的亚稳态传播到后面逻辑中，导致亚稳态的传播。

（比较容易理解的方式）换个方式理解：需要建立时间是因为触发器的 D 段像一个锁存器在接受数据，为了稳定的设置前级门的状态需要一段稳定时间；需要保持时间是因为在时钟沿到来之后，触发器要通过反馈来所存状态，从后级门传到前级门需要时间。

6：什么是亚稳态？为什么两级触发器可以防止亚稳态传播？

这也是一个异步电路同步化的问题，具体的可以参考《EDACN 技术月刊 20050401》。

亚稳态是指触发器无法在某个规定的时间段内到达一个可以确认的状态。使用两级触发器来使异步电路同步化的电路其实叫做“一步同位器”，他只能用来对一位异步信号进行同步。两级触发器可防止亚稳态传播的原理：假设第一级触发器的输入不满足其建立保持时间，它在第一个脉冲沿到来后输出的数据就为亚稳态，那么在下一个脉冲沿到来之前，其输出的亚稳态数据在一段恢复时间后必须稳定下来，而且稳定的数据必须满足第二级触发器的建立时间，如果都满足了，在下一个脉冲沿到来时，第二级触发器将不会出现亚稳态，因为其输入端的数据满足其建立保持时间。**同步器有效的条件：第一级触发器进入亚稳态后的恢复时间 + 第二级触发器的建立时间 < = 时钟周期**。

更确切地说，输入脉冲宽度必须大于同步时钟周期与第一级触发器所需的保持时间之和。最保险的脉冲宽度是两倍同步时钟周期。 所以，这样的同步电路对于从较慢的时钟域来的异步信号进入较快的时钟域比较有效，对于进入一个较慢的时钟域，则没有作用 。

7：系统最高速度计算（最快时钟频率）和流水线设计思想：

同步电路的速度是指同步系统时钟的速度，同步时钟愈快，电路处理数据的时间间隔越短，电路在单位时间内处理的数据量就愈大。假设 Tco 是触发器的输入数据被时钟打入到触发器到数据到达触发器输出端的延时时间；Tdelay 是组合逻辑的延时；Tsetup 是 D 触发器的建立时间。假设数据已被时钟打入 D 触发器，那么数据到达第一个触发器的 Q 输出端需要的延时时间是 Tco，经过组合逻辑的延时时间为 Tdelay，然后到达第二个触发器的 D 端，要希望时钟能在第二个触发器再次被稳定地打入触发器，则时钟的延迟必须大于 Tco＋Tdelay＋Tsetup，也就是说最小的时钟周期 Tmin =Tco＋Tdelay＋Tsetup，即最快的时钟频率 Fmax =1/Tmin。FPGA 开发软件也是通过这种方法来计算系统最高运行速度 Fmax。因为 Tco 和 Tsetup 是由具体的器件工艺决定的，故设计电路时只能改变组合逻辑的延迟时间 Tdelay，所以说缩短触发器间组合逻辑的延时时间是提高同步电路速度的关键所在。由于一般同步电路都大于一级锁存，而要使电路稳定工作，时钟周期必须满足最大延时要求。故只有缩短最长延时路径，才能提高电路的工作频率。可以将较大的组合逻辑分解为较小的 N 块，通过适当的方法平均分配组合逻辑，然后在中间插入触发器，并和原触发器使用相同的时钟，就可以避免在两个触发器之间出现过大的延时，消除速度瓶颈，这样可以提高电路的工作频率。这就是所谓"流水线"技术的基本设计思想，即原设计速度受限部分用一个时钟周期实现，采用流水线技术插入触发器后，可用 N 个时钟周期实现，因此系统的工作速度可以加快，吞吐量加大。注意，流水线设计会在原数据通路上加入延时，另外硬件面积也会稍有增加。

8：时序约束的概念和基本策略？

时序约束主要包括周期约束，偏移约束，静态时序路径约束三种。通过附加时序约束可以综合布线工具调整映射和布局布线，是设计达到时序要求。

附加时序约束的一般策略是先附加全局约束，然后对快速和慢速例外路径附加专门约束。附加全局约束时，首先定义设计的所有时钟，对各时钟域内的同步元件进行分组，对分组附加周期约束，然后对 FPGA/CPLD 输入输出 PAD 附加偏移约束、对全组合逻辑的 PAD TO PAD 路径附加约束。附加专门约束时，首先约束分组之间的路径，然后约束快、慢速例外

路径和多周期路径，以及其他特殊路径。

9：附加约束的作用？

作用：1：提高设计的工作频率（减少了逻辑和布线延时）；2：获得正确的时序分析报告；（静态时序分析工具以约束作为判断时序是否满足设计要求的标准，因此要求设计者正确输入约束，以便静态时序分析工具可以正确的输出时序报告）3：指定 FPGA/CPLD 的电气标准和引脚位置。

10：FPGA 设计工程师努力的方向：

SOPC，高速串行 I/O，低功耗，可靠性，可测试性和设计验证流程的优化等方面。随着芯片工艺的提高，芯片容量、集成度都在增加，FPGA 设计也朝着高速、高度集成、低功耗、高可靠性、高可测、可验证性发展。芯片可测、可验证，正在成为复杂设计所必备的条件，尽量在上板之前查出 bug，将发现 bug 的时间提前，这也是一些公司花大力气设计仿真平台的原因。另外随着单板功能的提高、成本的压力，低功耗也逐渐进入 FPGA 设计者的考虑范围，完成相同的功能下，考虑如何能够使芯片的功耗最低，据说 altera、xilinx 都在根据自己的芯片特点整理如何降低功耗的文档。高速串行 IO 的应用，也丰富了 FPGA 的应用范围，象 xilinx 的 v2pro 中的高速链路也逐渐被应用。 总之，学无止境，当掌握一定概念、方法之后，就要开始考虑 FPGA 其它方面的问题了。

11：对于多位的异步信号如何进行同步？

对以一位的异步信号可以使用“一位同步器进行同步”，而对于多位的异步信号，可以采用如下方法：1：可以采用保持寄存器加握手信号的方法（多数据，控制，地址）；2：特殊的具体应用电路结构,根据应用的不同而不同 ；3：异步 FIFO。（最常用的缓存单元是 DPRAM）

12：FPGA 和 CPLD 的区别？

FPGA 是可编程 ASIC。

ASIC:专用集成电路，它是面向专门用途的电路，专门为一个用户设计和制造的。根据一个用户的特定要求，能以低研制成本，短、交货周期供货的全定制，半定制集成电路。与门阵列等其它 ASIC(Application Specific IC)相比，它们又具有设计开发周期短、设计制造成本低、开发工具先进、标准产品无需测试、质量稳定以及可实时在线检验等优点。

	CPLD	FPGA
内部结构	Product－term	Look－up Table
程序存储	内部 EEPROM	SRAM，外挂 EEPROM
资源类型	组合电路资源丰富	触发器资源丰富
集成度	低	高
使用场合	完成控制逻辑	能完成比较复杂的算法
速度	慢	快

其他资源	一	PLL、RAM 和乘法器等
保密性	可加密	一般不能保密

13：锁存器（latch）和触发器（flip-flop）区别？

电平敏感的存储器件称为锁存器。可分为高电平锁存器和低电平锁存器，用于不同时钟之间的信号同步。

有交叉耦合的门构成的双稳态的存储原件称为触发器。分为上升沿触发和下降沿触发。可以认为是两个不同电平敏感的锁存器串连而成。前一个锁存器决定了触发器的建立时间，后一个锁存器则决定了保持时间。

14：FPGA 芯片内有哪两种存储器资源？

FPGA 芯片内有两种存储器资源：一种叫 block ram,另一种是由 LUT 配置成的内部存储器（也就是分布式 ram）。Block ram 由一定数量固定大小的存储块构成的，使用 BLOCK RAM 资源不占用额外的逻辑资源，并且速度快。但是使用的时候消耗的 BLOCK RAM 资源是其块大小的整数倍。

15：什么是时钟抖动？

时钟抖动是指芯片的某一个给定点上时钟周期发生暂时性变化，也就是说时钟周期在不同的周期上可能加长或缩短。它是一个平均值为 0 的平均变量。

16：FPGA 设计中对时钟的使用？（例如分频等）

FPGA 芯片有固定的时钟路由，这些路由能有减少时钟抖动和偏差。需要对时钟进行相位移动或变频的时候，一般不允许对时钟进行逻辑操作，这样不仅会增加时钟的偏差和抖动，还会使时钟带上毛刺。一般的处理方法是采用 FPGA 芯片自带的时钟管理器如 PLL,DLL 或 DCM，或者把逻辑转换到触发器的 D 输入（这些也是对时钟逻辑操作的替代方案）。

17：FPGA 设计中如何实现同步时序电路的延时？

首先说说异步电路的延时实现：异步电路一半是通过加 buffer、两级与非门等（我还没用过所以也不是很清楚），但这是不适合同步电路实现延时的。在同步电路中，对于比较大的和特殊要求的延时，一半通过高速时钟产生计数器，通过计数器来控制延时；对于比较小的延时，可以通过触发器打一拍，不过这样只能延迟一个时钟周期。

18：FPGA 中可以综合实现为 RAM/ROM/CAM 的三种资源及其注意事项？

三种资源：block ram;触发器（FF），查找表（LUT）；

注意事项：1：在生成 RAM 等存储单元时，应该首选 block ram 资源；其原因有二：第一：使用 block ram 等资源，可以节约更多的 FF 和 4-LUT 等底层可编程单元。使用 block ram 可以说是“不用白不用”，是最大程度发挥器件效能，节约成本的一种体现；第二：block ram 是一种可以配置的硬件结构，其可靠性和速度与用 LUT 和 register 构建的存储器更有优势。2：弄清 FPGA 的硬件结构，合理使用 block ram 资源；3：分析 block ram 容量，高效使用 block ram 资源；4：分布式 ram 资源（distribute ram）

19：什么是"线与"逻辑,要实现它,在硬件特性上有什么具体要求?

线与逻辑是两个输出信号相连可以实现与的功能。在硬件上,要用 oc 门来实现,由于不用 oc 门可能使灌电流过大,而烧坏逻辑门. 同时在输出端口应加一个上拉电阻。Oc 门就是集电极开路门。

20：什么是竞争与冒险现象?怎样判断?如何消除?

在组合电路中，某一输入变量经过不同途径传输后，到达电路中某一汇合点的时间有先有后，这种现象称竞争；由于竞争而使电路输出发生瞬时错误的现象叫做冒险。（也就是由于竞争产生的毛刺叫做冒险）。判断方法：代数法（如果布尔式中有相反的信号则可能产生竞争和冒险现象）；卡诺图：有两个相切的卡诺圈并且相切处没有被其他卡诺圈包围，就有可能出现竞争冒险；实验法：示波器观测；

解决方法：1：加滤波电路，消除毛刺的影响；2：加选通信号，避开毛刺；3：增加冗余项消除逻辑冒险。

21.用 Verilog 设计一个 5 分频器。

5 分频，奇数分频都可以类似这么做,只需要改 div1 和 div2 的参数。div1 为奇数分频除 2 的余数。采用上升延和下降延分别触发不同波形，最后叠加的方式产生奇数分频。

```
module divfreq(clk, clk1x, rst, clk1xpose, clk1xnege, coutpose, coutnege);
input clk;
input rst;
output clk1x;
output clk1xpose;
output clk1xnege;
output[2:0] coutpose;
output[2:0] coutnege;

reg clk1xpose;
reg clk1xnege;
reg[2:0] coutpose;
reg[2:0] coutnege;
```

```
parameter div1 = 2 , div2 = 4;              // div1  =  5 / 2, div2 = 5 - 1
assign clk1x = clk1xpose | clk1xnege;
always@(posedge clk or negedge rst)
begin
   if(!rst)
      clk1xpose = 0;
   else if(coutpose == div1)
      clk1xpose = ~clk1xpose;
   else if(coutpose == div2)
      clk1xpose = ~clk1xpose;
   else
      clk1xpose = clk1xpose;
end

always@(negedge clk or negedge rst)
begin
   if(!rst)
      clk1xnege = 0;
   else if(coutnege == div1)
      clk1xnege = ~clk1xnege;
   else if(coutnege == div2)
      clk1xnege = ~clk1xnege;
   else
      clk1xnege = clk1xnege;
end

always@(posedge clk or negedge rst)
begin
   if(!rst)
      coutpose = 0;
   else if(coutpose == div2)
      coutpose = 0;
   else
      coutpose = coutpose + 1;
end

always@(negedge clk or negedge rst)
begin
   if(!rst)
      coutnege = 0;
   else if(coutnege == div2)
      coutnege = 0;
   else
```

```
        coutnege = coutnege + 1;
end
endmodule
```

22.用状态机检测序列为 11001101，输出为 1，否则输出为 0。

```
module machine_test2(clk,rst,din,dout);
input clk,rst;
input din;
output dout;

//11001101
parameter idle=0,
      st0=1,
      st1=2,
      st2=3,
      st3=4,
      st4=5,
      st5=6,
      st6=7,
      st7=8;

reg [3:0] current_state,next_state;
reg dout;

always @ (posedge clk or negedge rst)
begin
  if(!rst)
    current_state<=idle;
  else
    current_state<=next_state;
end

always @ (next_state or current_state or din or dout)
begin
  case(current_state)
  idle:
        if(din)
              next_state=st0;
        else
              next_state=idle;
  st0:
        if(din)
              next_state=st1;
```

```
        else
                next_state=idle;
    st1:
        if(!din)
                next_state=st2;
        else
                next_state=st1;
    st2:
        if(!din)
                next_state=st3;
        else
                next_state=st0;
    st3:
        if(din)
                next_state=st4;
        else
                next_state=idle;
    st4:
        if(din)
                next_state=st5;
        else
                next_state=idle;
    st5:
        if(!din)
                next_state=st6;
        else
                next_state=st1;
    st6:
        if(din)
                next_state=st7;
        else
                next_state=st3;
    st7:
        if(din)
                next_state=st1;
        else
                next_state=idle;
    default:
                next_state=idle;
    endcase
end

always @ (posedge clk)
begin
```

```
if(next_state==st7)
    dout<=1'd1;
else
    dout<=1'd0;
end
endmodule
```

23. IC 设计中同步复位与 异步复位的区别

异步复位是不受时钟影响的，需要一个全局的信号来对整个芯片进行整体的复位，到一个初始的确定状态。而同步复位需要在时钟沿来临的时候才会对整个系统进行复位。使用 FPGA 设计时芯片的异步复位必须要走全局网络。

异步复位最大的优点是, 数据通路就可以不依赖于时钟而确保清洁可控。然而, 异步复位 STA (静态时序分析) 的复杂性要高于同步复位设计; 如果异步复位信号在触发器时钟有效沿附近“释放”(复位信号从有效变为无效) 的话, 可能会导致触发器输出的亚稳态。

24. MOORE 与 MEELEY 状态机的特征

Moore 有限状态机输出只与当前状态有关，与输入信号的当前值无关。在时钟脉冲的有效边沿作用后的有限个门延后，输出达到稳定值。从时序上看，Moore 状态机属于同步输出状态机。Moore 有限状态机最重要的特点就是将输入与输出信号隔离开来。

Mealy 状态机的输出是现态和所有输入的函数，随输入变化而随时发生变化。从时序上看，Mealy 状态机属于异步输出状态机，它不依赖于时钟。

25. 画状态机，并用 verilog 实现接受 1,2,5 分钱的卖报机，每份报纸 5 分钱。

```
module machine_125
    (
        input    clk            ,
        input    rst            ,
        input    [2:0]din   ,    //din[0]:1 分钱，din[1]:2 分钱，din[2]:5 分钱。
        output   [1:0]dout //dout[0]:输出报纸 1 份；dout[1]：找零 1 分钱。
    );

    parameter    idle=0,
                 st1=1,
                 st2=2,
                 st3=3,
                 st4=4,
                 st5=5,
                 st6=6;

    reg [2:0] current_state,next_state;
```

```
reg dout;

always @ (posedge clk or negedge rst)
begin
 if(!rst)
   current_state<=idle;
 else
   current_state<=next_state;
end

always @ (next_state or current_state or din or dout)
begin
 case(current_state)
 idle:
      if(din==3'b100)
            next_state=st1;
      else if(din==3'b010)
            next_state=st2;
      else if(din==3'b001)
            next_state=st5;
      else if(din==3'b000)
            next_state=idle;
 st1:
      if(din==3'b100)
            next_state=st2;
      else if(din==3'b010)
            next_state=st3;
      else if(din==3'b000)
            next_state=st1;
 st2:
      if(din==3'b100)
            next_state=st3;
      else if(din==3'b010)
            next_state=st4;
      else if(din==3'b000)
            next_state=st2;
 st3:
      if(din==3'b100)
            next_state=st4;
      else if(din==3'b010)
            next_state=st5;
      else if(din==3'b000)
            next_state=st3;
 st4:
```

```
            if(din==3'b100)
                next_state=st5;
            else if(din==3'b010)
                next_state=st6;
            else if(din==3'b000)
                next_state=st4;
        st5:
            if(din==3'b100)
                next_state=st1;
            else if(din==3'b010)
                next_state=st2;
            else if(din==3'b000)
                next_state=idle;
        st6:
            if(din==3'b100)
                next_state=st2;
            else if(din==3'b010)
                next_state=st3;
            else if(din==3'b000)
                next_state=st1;
        default:
                next_state=idle;
        endcase
    end

    always @ (posedge clk)
    begin
    if(next_state==st5)
        dout<=2'b10;
    else if(next_state==st6)
        dout<=2'b11;
    else
        dout<=2'b00;
    end
endmodule
```

26. FPGA 的基本结构

目前主流 FPGA 都采用了基于 SRAM 工艺的查找表结构，LUT 本质上就是一个 RAM。目前 FPGA 中多使用 4 输入的 LUT，所以每一个 LUT 可以看成一个有 4 位地址线的 的 RAM。当用户通过原理图或 HDL 语言描述了一个逻辑电路以后，PLD/FPGA 开发软件会自动计算逻辑电路的所有可能结果，并把真值表（即结果）事先写入 RAM，这样，每输入一个信号进行逻辑运算就等于输入一个地址进行查表，找出地址对应的内容，然后输出即可。

其组成部分主要有可编程输入/输出单元、基本可编程逻辑单元、嵌入式块 RAM、丰富

的布线资源、底层嵌入功能单元、内嵌专用硬核等。

1． 可编程输入输出单元（IOB）

可编程输入/输出单元简称 I/O 单元，是芯片与外界电路的接口部分，完成不同电气特性下对输入/输出信号的驱动与匹配要求。FPGA 内的 I/O 按组分类，每组都能够独立地支持不同的 I/O 标准。通过软件的灵活配置，可适配不同的电气标准与 I/O 物理特性，可以调整驱动电流的大小，可以改变上、下拉电阻。目前，I/O 口的频率也越来越高，一些高端的 FPGA 通过 DDR 寄存器技术可以支持高达 2Gbps 的数据速率。

为了便于管理和适应多种电器标准，FPGA 的 IOB 被划分为若干个组（bank），每个 bank 的接口标准由其接口电压 VCCO 决定，一个 bank 只能有一种 VCCO，但不同 bank 的 VCCO 可以不同。只有相同电气标准的端口才能连接在一起，VCCO 电压相同是接口标准的基本条件。

2． 可配置逻辑块（CLB）

FPGA 的基本可编程逻辑单元是由查找表和寄存器组成的，查找表完成纯组合逻辑功能。FPGA 内部寄存器可配置成触发器或锁存器。Altera 基本可编程单元 LE 配置为 1 寄存器加一个查找表。

3． 嵌入式块 RAM（BRAM）

块 RAM 可被配置为单端口 RAM、双端口 RAM、内容地址存储器（CAM）以及 FIFO 等常用存储结构。CAM 存储器在其内部的每个存储单元中都有一个比较逻辑，写入 CAM 中的数据会和内部的每一个数据进行比较，并返回与端口数据相同的所有数据的地址，因而在路由的地址交换器中有广泛的应用。除了块 RAM，还可以将 FPGA 中的 LUT 灵活地配置成 RAM、ROM 和 FIFO 等结构。

4． 丰富的布线资源

布线资源连通 FPGA 内部的所有单元，而连线的长度和工艺决定着信号在连线上的驱动能力和传输速度。FPGA 芯片内部有着丰富的布线资源，根据工艺、长度、宽度和分布位置的不同而划分为 4 类不同的类别。第一类是全局布线资源，用于芯片内部全局时钟和全局复位/置位的布线；第二类是长线资源，用以完成芯片 Bank 间的高速信号和第二全局时钟信号的布线；第三类是短线资源，用于完成基本逻辑单元之间的逻辑互连和布线；第四类是分布式的布线资源，用于专有时钟、复位等控制信号线。

5． 底层内嵌功能单元

内嵌功能模块主要指 DLL（Delay Locked Loop）、PLL（Phase Locked Loop）、DSP 和 CPU 等软处理核（Soft Core）。现在越来越丰富的内嵌功能单元，使得单片 FPGA 成为了系统级的设计工具，使其具备了软硬件联合设计的能力，逐步向 SOC 平台过渡。

6. 内嵌专用硬核

内嵌专用硬核是相对底层嵌入的软核而言的，指 FPGA 处理能力强大的硬核（Hard Core），等效于 ASIC 电路。为了提高 FPGA 性能，芯片生产商在芯片内部集成了一些专用的硬核。例如：为了提高 FPGA 的乘法速度，主流的 FPGA 中都集成了专用乘法器；为了适用通信总线与接口标准，很多高端的 FPGA 内部都集成了串并收发器（SERDES），可以达到数十 Gbps 的收发速度。

27. 程序下载到 FPGA 的方式有哪几种，JTAG 有哪几条线。

JTAG：下载 sof 文件，掉电丢失。可以将 sof 转换为 jic 文件，用 EPCSx 配置，掉电不丢失。

AS：下载 pof 文件，配置 EPCSx，掉电不丢失。

PS：由外部计算机或控制器控制配置过程。

JTAG 有以下几条线：

TCK：测试时钟输入，用于移位控制，上升沿将测试指令、测试数据和控制输入信号移入芯片；下降沿时将数据从芯片移出。

TMS：测试模式选择，串行输入端，用于控制芯片内部的 JTAG 状态机。

TDI：测试数据输入，串行输入端，用于指令和编程数据的输入，在时钟上升沿，数据被捕获。

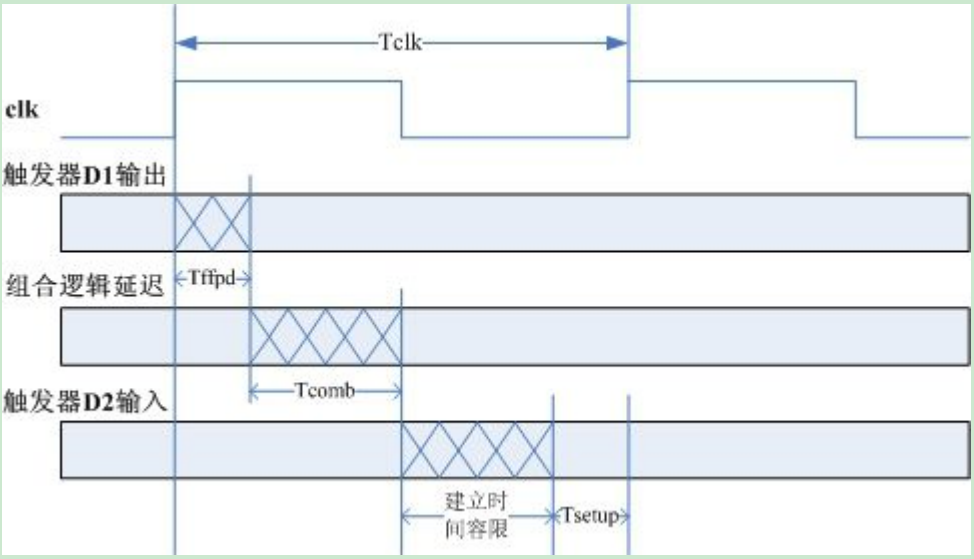
TDO：测试数据输出，串行输出端，时钟下降沿，数据被驱动输出。

TRST：测试复位输入（仅用于扩展 JTAG），异步、低电平有效，用于 JTAG 初始化时。

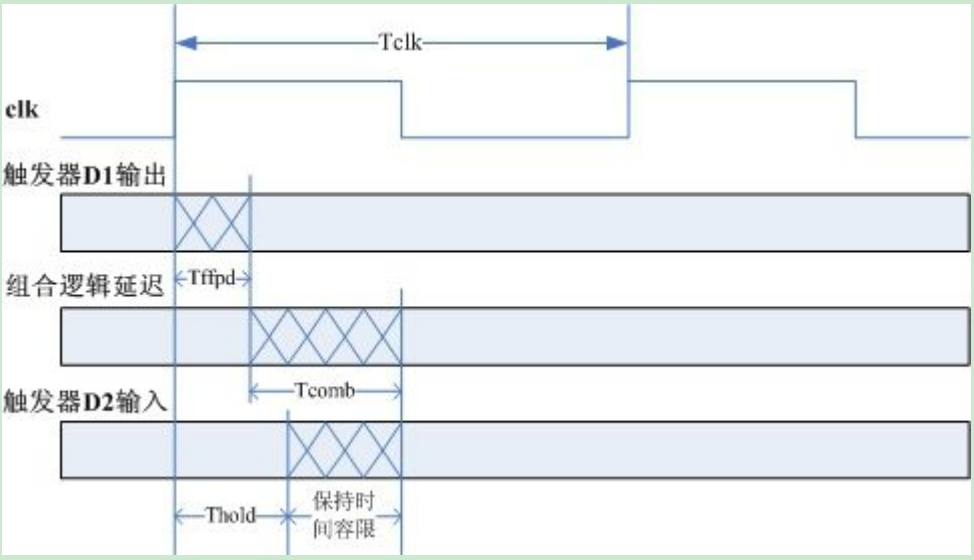
28. 时钟周期为 T，触发器 D1 的建立时间最大为 T1max，最小为 T1min。组合逻辑电路最大延迟为 T2max，最小为 T2min。问：触发器 D2 的建立时间 T3 和保持时间 T4 应满足什么条件？

建立时间容限：相当于保护时间，这里要求建立时间容限大于等于 0。

保持时间容限：保持时间容限也要求大于等于 0。



由上图可知，建立时间容限＝Tclk-Tffpd(max)-Tcomb(max)-Tsetup，根据建立时间容限≥0，也就是 Tclk-Tffpd(max)-Tcomb(max)-Tsetup≥0，可以得到触发器 D2 的 Tsetup≤Tclk-Tffpd(max)-Tcomb(max)，由于题目没有考虑 Tffpd，所以我们认为 Tffpd＝0，于是得到 Tsetup≤T-T2max。



由上图可知，保持时间容限+Thold＝Tffpd(min)+Tcomb(min)，所以保持时间容限＝Tffpd(min)+Tcomb(min)-Thold，根据保持时间容限≥0，也就是 Tffpd(min)+Tcomb(min)-Thold≥0,得到触发器 D2 的 Thold≤Tffpd(min)+Tcomb(min)，由于题目没有考虑 Tffpd，所以我们认为 Tffpd＝0，于是得到 Thold≤T2min。关于保持时间的理解就是，在触发器 D2 的输入信号还处在保持时间的时候，如果触发器 D1 的输出已经通过组合逻辑到达 D2 的输入端的话，将会破坏 D2 本来应该保持的数据。