

# FPGA/集成电路笔试面试题

## 明德扬科技教育有限公司

官 网: [www.mdyl-edu.com](http://www.mdyl-edu.com)

淘 宝: [mdyl-edu.taobao.com](http://mdyl-edu.taobao.com)

QQ 群: 97925396

1. Write a sequence of 3-bit grey code. Can you derive a general equation to convert binary to grey code? [AMD 2008]

[答案: 08310001] **注:“08310001”对应答案视频序号**

2. 怎样将一个 single-bit 信号从快时钟域送到慢时钟域, 或慢送到快? Multi-bit 信号呢?  
[AMD 2008]

[答案: 08310002]

3. 设计一个计算连续 Leading Zeros 个数的电路。输入 8-bit, 输出 4-bit。[AMD 2008]

00001000      0100

00100010      0010

10001000      0000

可以 parameterize 你的设计吗? 其 hardware 是什么样子的?

[答案: 09090001]

4. 出下面两个状态机的逻辑综合图, 并说明两种写法的优缺点! [凹凸 2008]

```
always @(posedge clk or negedge rst)
    if(!rst)begin
        state<=0;
        out<=4'b0000;
    end
    else
        case(state)
            0:begin
                state<=1;
                out<=4'b0000;
            end
            1:begin
                state<=0;
                out<=4'b0001;
            end
        endcase
```

```
always @(posedge clk or negedge rst)
    if(!rst)
        state<=0;
    else
```

```
case(state)
    0:state<=1;
    1:state<=0;
endcase

always@(state)
if(!state)
    out=4'b0000;
else
    out=4'b0001;
```

前者 state 和 out[0]分别综合成两个触发器,其中 state 触发器的输入为它本身的反向。out[0]的触发器为 state 的当前输入。后者只有 state 一个触发器,输出直接赋值给 out[0],光靠这两段看不出它到底要考啥,估计可能是想考状态机 1 段, 2 段, 3 段式的优缺点。

第一个 out 要综合成四位的寄存器, 占用资源;  
第二个则比较节约了 呵呵

2 段和 1 段相比, 没有节省资源吧, 还多用了寄存器了。

不过肯定是提高了程序代码的可读性和维护性我觉得这两段的输出加一步寄存会更好一些, 可以减少逻辑电路带来的毛刺 吴继华, 王城编的《verilog 设计和验证》有一章对这个讲得很详细你可以到论坛里面去找找, 好像是有电子版的

第一段代码速度上比较快, 但所占资源多; 第二段正好相反, 只用了一个寄存器, 但 Timing 会差点。所谓可读性, 我觉得在这里没什么区别, 关键还是硬件实现上的区别。

我的观点是: 二段式的状态机将组合逻辑和时序逻辑分离开, 便于综合工具进行分析。ls 各位说的, 只是基于这两个例子但是题目是关于状态机的写法的优缺点这是我的观点

[答案: 09090002]

## 5. 设计地址生成器。[nVidia 2008]

要求依次输出以下序列:

0,8,2,10,4,12,6,14,1,9,3,11,5,13,7,15,  
16,24,18,26,.....,31,  
32,40,34,42,.....,47,  
48,56,50,58,.....,63,  
64,72,66,76,.....,79

[答案: 09090003]

6. 假设存在 positive clock skew 为 10ns, 问最高电路频率。[SIRF 2008]

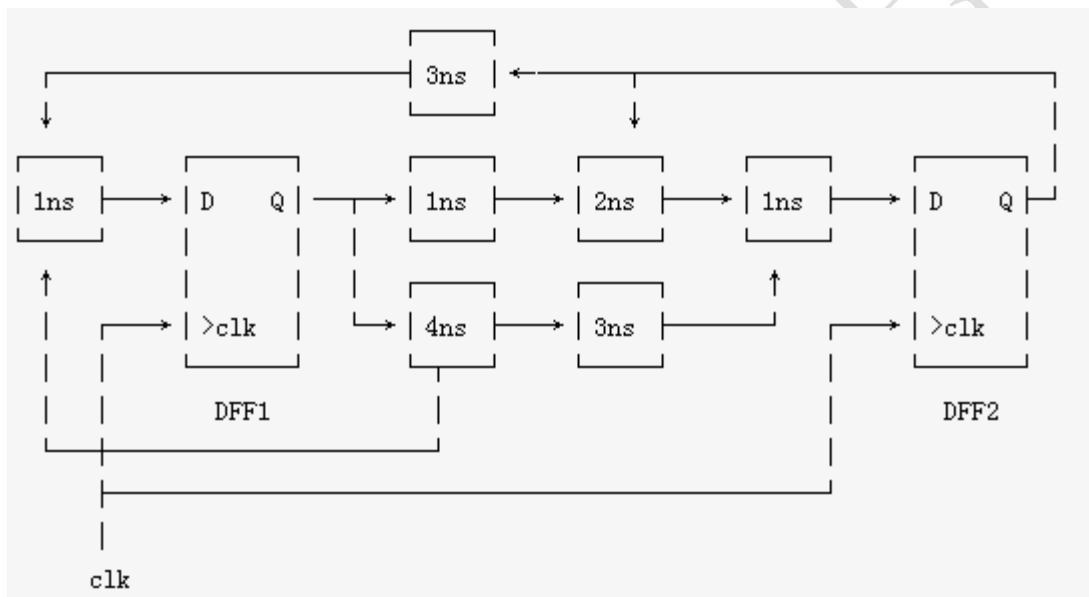
能容忍的最大 positive clock skew

能容忍的最大 negative clock skew

positive clock skew: DFF2 的 clock 比 DFF1 的来的晚

negative clock skew: DFF2 的 clock 比 DFF1 的来的早

$T_{\text{setup}}=1\text{ns}$      $T_{\text{hold}}=1\text{ns}$      $T_{\text{clk} \rightarrow \text{q}}=1\text{ns}$



[答案: 09090004]

7. 阻塞赋值和非阻塞赋值的区别[Trident]

`always@(posedge clk)`

`begin`

`b=a;`

`c=b;`

`end`

`always@(posedge clk)`

`begin`

`b<=a;`

`c<=b;`

`end`

上面两段代码的硬件结构是怎么样的?

[答案: 09090005]

8. 化简代码使硬件尽可能少[Trident]

```
always@ (sel or a or b or c)
```

```
if(sel)
```

```
    y = a + b;
```

```
else
```

```
    y = a + c;
```

[答案: 09090006]

9. 2 进制的 1101.101 变成十进制是多少? [Trident]

[答案: 09090007]

10. 下面哪种写法会产生 latch? 为什么? [SIRF 2008]

<pre>always  @(*)begin     if(d)         a = b; end</pre>	<pre>always  @(*)begin     if(d)         a = b;     else         a = a; end</pre>	<pre>always @ (b or d) case(d)     2'b00:  a=b&gt;&gt;1;     2'b11:  c=b&gt;&gt;1; default:     begin         a=b;         c=b;     end endcase</pre>
<pre>always @(b or d)begin     a=b;     c=b;     case(d)         2'b00:  a=b&gt;&gt;1;         2'b11:  c=b&gt;&gt;1;     endcase end</pre>	<pre>always@(b or d)begin     case(d) //synopsys full_case         2'b00:  a=b&gt;&gt;1;         2'b11:  c=b&gt;&gt;1;     endcase end</pre>	

[答案: 09090008]

11. 从仿真的角度设计测试 32(bit)\*32(bit)的乘法器能否正常工作的过程?

[答案: 09110001]

12. 从仿真的角度设计测试 1024-depth 的 SRAM 能否正常工作的步骤或过程，功能：有 10 位的读写指针，并且读操作与写操作可以同时进行，负责读和写的部分由一个控制器控制。

[答案：09110002]

13. 报文替换 ID 的功能

已知某传输流的包长为 188 字节，其数据格式如下：  
0x47 (包头)，0x00, 0x00, ID, 数据, ...。  
其中，0x47 为包头信号；ID 为 1 字节的数据，取值范围为 0x00~0xFF；数据为有效负载。  
负载中不会出现 0x47。  
需要把流中的 ID 映射成新的值，映射表由外部上位机实时设置 RAM 实现。模块接口如下：

```

module id_map(
    input          rst_i          , // 复位信号
    input          clk_i          , // 数据时钟
    input [7:0]    dat_i          , // 输入数据信号
    input          vld_i          , // 输入数据有效信号，高有效
    //映射表
    input          cpu_clk_i      , // 主控时钟
    input [7:0]    cpu_dat_i      , // 主控的 RAM 写数据
    input [7:0]    cpu_addr_i     , // 主控的 RAM 写地址
    input          cpu_wr_i       , // 主控的 RAM 写使能
    //输出
    output         dat_o          , // 映射后的数据信号
    output         hdr_o          , // 映射后的数据包头信号
    output         vld_o          , // 映射后的有效信号
);

```

RAM 的读数延时为 1 个时钟周期，其接口信号如下：

```

module ram (
    input          wrclock        , // RAM 写时钟
    input [7:0]    wraddress      , // RAM 写地址
    input          wren           , // RAM 写使能
    input [7:0]    data          , // RAM 写数据
    input          rdclock        , // RAM 读时钟
    input [7:0]    rdaddress      , // RAM 读地址
    input          rden           , // RAM 读使能
    output [7:0]   q              , // RAM 读数据
);

```

要求：

- 1) 描述清楚模块的设计思路。说明上位机设置 RAM 表的数据格式。
- 2) 完成 Verilog HDL 或者 VHDL 代码。
- 3) 简述该模块由上位机实时设置映射 RAM 表可能带来的问题，如何处理。

[答案：09110003]

14. flip-flop 和 latch 的区别, rtl 中 latch 是如何产生的[SIRF 2008]

[答案: 09120001]

15. 多时钟域设计中, 如何处理跨时钟域信号? [SIRF 2008]

[答案: 09120002]

16. 锁存器比寄存器省面积, 但为什么在 IC 设计中通常使用寄存器? [SIRF 2008]

[答案: 091200013]

17. 用 verilog/vhdl 写一个 fifo 控制器(包括空, 满, 半满信号)。(飞利浦一大唐笔试)

reg[N-1:0] memory[0:M-1]; 定义 FIFO 为 N 位字长容量 M

[答案: 10190001]

18. FPGA 的片上 RAM 资源, 可以在设计中如下哪些应用?

- a、Shift Register      b、ROM
- c、RAM                  d、FIFO

[答案: 09250002]

19. 下列哪些属于时钟约束?

- a、set\_false\_path      b、set\_input\_path
- c、set\_max\_delay      d、set\_multicycle path

[答案: 09250002]

20. FPGA 可以有哪些工艺?

- a、SDRAM      b、SRAM      c、EEPOM      b、DDR      e、FLASH

[答案: 09250002]

21. 下列哪些是 FPGA 片内资源？

- a、RAM      b、LUT      c、DSP      d、SDRAM

[答案：09250002]

22. 下列哪些选项是 FPGA 设计中必须的设计约束？

- a、管脚约束      b、跨时钟域约束  
c、时钟周期约束      d、片上 RAM 位置约束

[答案：09250002]

23. 判断：FPGA 中，需要一个 1MByte 的存储空间，用片上 RAM 实现即可。

[答案：09250002]

24. 判断：Latch 和 Register 的结构是不同的，Latch 是电位控制器件，Register 是时序控制器件。

[答案：09250002]

25. 判断：FPGA 设计中，访问 FLASH 的速度比 DDR 快。

[答案：09250002]

26. 阐述以下数字电路中时钟属性：(1) Jitter ：时钟抖动 (2) clock\_skew ：时钟偏移。问题：这两个不同吗？

[答案：09250002]



## 27. 分析时序报告

335. 下面是一条 timing report, 请填空会带相关问题。

Startpoint: sx0\_0/SYS/Dinput\_mtp (rising edge-triggered flip-flop clock by vp1l1),  
Endpoint: sx0\_0/SYS/Doutput\_mtp (rising edge-triggered flip-flop clock by vp1l1),  
Path group: vp1l1,

Point	fanout	cap	trans	incr	Path
Clock vp1l1(rising edge)				0.000	0.000
Clock network delay(propagated)				0.080	0.080
sx0_0/SYS/Dinput_mtp/cp(DFCSND1)			0.006	0.000	0.080
sx0_0/SYS/Dinput_mtp/Q(DFCSND1)			0.005	0.091	0.171
sx0_0/SYS/n1(net)	1	0.001			
sx0_0/SYS/fb_control/io(MUX2D1LVT)			0.005	0.000	0.171
sx0_0/SYS/fb_control/Z(MUX2D1LVT)			0.006	0.022	0.193
sx0_0/SYS/n2(net)	1	0.001			
sx0_0/SYS/fb_mtp/io(MUX2D1LVT)			0.009	0.002	0.195
sx0_0/SYS/fb_mtp/Z(MUX2D1LVT)			0.020	0.043	0.238
sx0_0/SYS/n3(net)	1	0.001			
sx0_0/SYS/Doutput_mtp/D(DFCSND1)			0.020	0.001	0.239
data arrival time					0.239
Clock vp1l1(rising edge)				0.000	0.000

92.

Clock network delay(propagated)	0.104	0.104
Clock reconvergence pessimism	-0.104	0.090
Clock uncertainty	0.050	0.140
sx0_0/SYS/Doutput_mtp/CP(DFCSND1)		0.140
library hold time	0.005	0.145
data required time		0.145

data required time	( )
data arrival time	( )
slack(MET? ___ or VIOLATED? ___)(mark ✓)	( )

这条 timing report 是 setup path 还是 hold path, 为什么? 是否有 timing violation, 如果有, 可以通过哪些方式来改进?