

目录

1 名词解释	1
1.1 CMOS (Complementary Metal Oxide Semiconductor)	1
1.2 MCU (Micro Controller Unit)	1
1.3 RISC 和 CISC.....	1
1.4 DSP (digital signal processor)	1
1.5 ASIC.....	2
1.6 ECC.....	2
1.7 SDRAM.....	2
1.8 IRQ (Interrupt Request)	2
1.9 USB (Universal Serial BUS)	2
1.10 BIOS (Basic Input Output System)	2
1.11 时钟抖动 (jitter)	3
1.12 时钟歪斜 (skew)	3
1.13 数字集成电路	3
1.14 模拟信号	3
1.15 数字信号	3
1.16 单极型晶体管	3
1.17 双极型晶体管	4
1.18 查找表 LUT.....	4
1.19 VHDL	4
1.20 SDR.....	4
1.21 RAM (RandomAccessMemory)	4
1.22 ROM (Read-OnlyMemory)	4
1.23 SRAM (Static RAM)	5
1.24 DRAM (Dynamic RAM)	5
1.25 SSRAM (Synchronous SRAM)	5
1.26 SDRAM (Synchronous DRAM)	5
1.27 DDR.....	5
1.28 DDR2.....	5
1.29 DDR3	6

1.30 FLASH	6
1.31 EEPROM	6
1.32 摩尔型状态机 (Moore)	6
1.33 米利型状态机 (Mealy)	6
1.34 BCD 码	6
1.35 短沟道效应	6
1.36 闩锁效应	6
1.37 体效应	7
2 简答题	7
2.1 ASIC 设计流程	7
2.2 FPGA 开发基本流程	7
2.3 什么是同步逻辑和异步逻辑	8
2.4 同步电路和异步电路的区别	8
2.5 时序设计的实质	8
2.6 建立时间与保持时间	9
2.7 建立时间和保持时间的作用	9
2.8 亚稳态及抑制方法	9
2.9 系统最快时钟频率和流水线	10
2.10 时序约束的概念和基本策略	10
2.11 附加约束的作用	10
2.12 多位的异步信号同步	11
2.13 FPGA 和 CPLD 的区别	11
2.14 锁存器和触发器	12
2.15 FPGA 芯片内两种存储器资源	12
2.16 门控时钟	12
2.17 FPGA 设计中时钟的使用	13
2.18 FPGA 中实现同步时序电路的延时	13
2.19 FPGA 综合实现为 RAM/ROM/CAM 的注意事项	13
2.20 Xilinx 中与全局时钟资源和 DLL 相关的硬件原语	13
2.21 HDL 语言的层次概念	13
2.22 查找表的原理与结构	14
2.23 IC 设计前端到后端的流程和 EDA 工具	14

2.24 寄生效应在 IC 设计中怎样加以克服和利用	14
2.25 什么是“线与”逻辑，在硬件特性上的具体要求	15
2.26 什么是竞争与冒险现象？怎样判断？如何消除？.....	15
2.27 常用逻辑电平？TTL 与 COMS 电平可以直接互连吗？	15
2.28 Moore 与 Mealy 状态机的特征	16
2.29 多时域设计中，如何处理信号跨时域	16
2.30 静态、动态时序模拟的优缺点	16
2.31 四级的 Mux，第二级信号为关键信号，如何改善 timing.....	17
2.32 关键路径	17
2.33 标准的倒相器中 P 管的宽长比要比 N 管的宽长比大.....	17
2.34 什么是状态图	17
2.35 可编程逻辑器件有哪些	17
2.36 SRAM、FLASH、DRAM、SSRAM 及 SDRAM 的区别	18
2.37 四种复用方式	18
2.38 修正 Setup time violation 和 Hold time violation	18
2.39 如何防止亚稳态	18
2.40 基尔霍夫定理的内容	19
2.41 反馈电路的概念及应用。	19
2.42 有源滤波器和无源滤波器的区别	19
2.43 建立、保持时间	19
2.44 OTP 片 OTP（一次性可编程）、掩膜片，两者的区别何在？	20
2.45 描述你对集成电路设计流程的认识	20
2.46 描述你对集成电路工艺的认识	20
2.47 简述 FPGA 等可编程逻辑器件设计流程	20
2.48 数字 IC 设计流程.....	21
2.49 数字 IC 后端设计流程	21
2.50 自动布局布线？一两种工具软件，需要哪些基本元素？	22
2.51 列举几种集成电路典型工艺。常提到 0.25,0.18 指的是什么？	23
2.52 半导体工艺中，掺杂有哪几种方式	23
2.53 描述 CMOS 电路中闩锁效应产生的过程及最后的结果	23
2.54 解释 latch-up 现象和 Antenna effect 及其预防措施.....	24
2.55 FPGA 公司	24

2.56 可编程单元.....	24
2.57 时序组合逻辑.....	24
2.58 列出你熟悉的前仿真验证工具，并以一种为例简述仿真步骤？	24
2.59 列出仿真过程中常用的 verilog 系统任务并描述作用	25
2.60 PC Chipset.....	25
2.61 短沟道效应.....	25
2.62 串口线传输距离	25
2.63 FIR 滤波器和 IIR 滤波器的区别	26
2.64 序列检测器，用两种方法	26
2.65 IBUFG 的作用。	26
2.66 fifo 深度计算.....	26
2.67 在对信号采样过程之前抗混滤波，其作用是什么	27
2.68 Verilog HDL 中哪些结构是不可综合的	27
2.69 代码覆盖率.....	27
2.70 面积和速度优化	27
2.71 时序检查中对异步复位电路的时序分析叫做什么	28
2.72 FPGA 开发基本流程	28
2.73 数制转换.....	28
2.74 竞争冒险.....	28
2.75 反相器的速度与哪些因素有关	28
2.76 Transition Time（转换时间）	29
2.77 Propagation Delay（传播延时）	29
2.78 什么是高阻态	29
2.79 什么是 open-drain output	29
2.80 相同面积的 cmos 与非门和或非门哪个更快.....	29
2.81 画出 clock gating cell 的原理图。	30
2.82 解释一下 Latch 的建立时间与保持时间。	30
2.83 应避免使用锁存器，尽量使用触发器的原因:.....	30
2.84 多时钟域设计中，如何处理跨时钟域	31
2.85 建立时间，保持时间不满足怎么处理？	31
2.86 微机接口逻辑图	31
2.87 三极管特性曲线	32

2.88 给出单管 DRAM 的原理图.....	32
2.89 IC 设计中同步复位与异步复位的区别	32
2.90 格雷码	33
2.91 化简代码.....	33
2.92 BRAM 和分布式 RAM 的区别	33
2.93 -3dB 增益计算.....	34
2.94 CMOS 和 TTL 电路区别是什么.....	34
2.95 FPGA 型号介绍	34
2.96 FPGA 的面积优化和速度优化有哪几类	35
2.97 FPGA 中属于高速串行接口的是	35
2.98 PCI 总线特点	35
2.99 画出异步 FIFO 架构.....	36
2.100 电路频率计算	36
2.101 RS232c 高电平脉冲对应的 TTL 逻辑是.....	36
2.102 串行通信与同步通信异同	37
2.103 机器周期和时钟周期	37
2.104 施密特电路回差电压	37
2.105 VCO 压控振荡器	37
2.106 什么是状态图	38
2.107 什么耐奎斯特定律	38
2.108 模拟信号转为数字信号	38
2.109 D 触发器和 D 锁存器的区别	38
2.110 拉氏变换和傅立叶变换的表达式及联系	38
2.111 平板电容公式 $C=\epsilon S/4\pi kd$	38
2.112 放大电路的频率补偿的目的是什么，有哪些方法	38
2.113 频率响应	39
2.114 给出一个查分运放，如何相位补偿，并画补偿后的波特图。	39
2.115 基本放大电路种类	39
2.116 差分信号求共模分量和差模分量	40
2.117 画差放的两个输入管	40
2.118 画出由运放构成的基本电路	40
2.119 用运算放大器组成一个 10 倍的放大器	41

2.120 分析积分电路	41
2.121 建立时间保持时间计算	41
2.122 最大时钟频率计算	41
2.123 触发器的种类及优缺点	41
2.124 latch 与 register 的区别	41
2.125 阻塞赋值和非阻塞赋值的区别	42
2.126 产品设计流程	43
2.127 给出单管 DRAM 原理	43
2.128 SRAM 读写原理	43
2.129 FPGA 与 ASIC 设计流程的区别	44
2.130 跨时钟域	44
2.131 CMOS 反相器的功耗包括几部分？分别与哪些因素相关？	45
2.132 建立时间、保持时间、传播延迟、转换时间、恢复时间、移除时间、最小脉冲宽度	45
2.133 ASIC 中低功耗的设计方法和思路（不适用于 FPGA）	46
2.134 最小周期计算	46
2.135 reg 和 wire 的区别	47
2.136 localparam、parameter 和 define 的区别	47
2.137 task 与 function 的区别	47
2.138 谈谈对 Retiming 技术的理解	47
2.139 验证覆盖率	47
2.140 下列功耗措施哪个可以降低峰值功耗（D）	48
2.141 跨时钟域处理	48
2.142 逻辑电路低功耗设计中，无效方法是（A）	48
2.143 噪声容限	48
2.144 FALSE PATH	49
2.145 逻辑分析仪	49
2.146 时序约束	49
2.147 5 级指令流水线	49
2.148 FPGA 时序优化方法	49
2.149 同步复位和异步复位的区别？	50
2.150 编译器编译 C 语言程序的主要过程？	50

2.151 在计算机程序中，堆和栈的区别是什么？	50
2.152 C 语言函数返回值	50
2.153 验证的主要目的是什么？	50
2.154 C 语言中，实型常数是按(A)处理。	50
2.155 FPGA 器件的结温范围	50
2.156 常用的工频陷波电路属于哪种类型的滤波电路(C)。	51
2.156 高速信号	51
2.157 传输延迟和扇出扇入的关系	51
2.158 OOP 的三大特征	51
2.159 NAND 和 NOR flash 的区别	51
2.160 移位计数器	51
2.161 buck 降压电路中，二极管、电容和电感作用	52
2.162 锁存器生成	52
2.163 CMOS 动态功耗计算	52
2.164 OD 门和 OC 门	52
2.165 全加器检测 1 的个数	53
2.166 数字倍频器	53
2.167 一个 16bit 的变量，判断能否被 5 整除	54
2.168 版图设计使用 P 衬底	54

1 名词解释

1.1 CMOS (Complementary Metal Oxide Semiconductor)

互补金属氧化物半导体，电压控制的一种放大器件。是组成 CMOS 数字集成电路的基本单元。

由 PMOS 管和 NMOS 管共同构成，它的特点是低功耗。由于 CMOS 中一对 MOS 组成的门电路在瞬间要么 PMOS 导通、要么 NMOS 导通、要么都截止，比线性的三极管 BJT 效率要高得多，因此功耗很低。

优点 :1.允许的电源电压范围宽 ,方便电源电路的设计 2.逻辑摆幅大 ,使电路抗干扰能力强 3.静态功耗低。 4.隔离栅结构使 CMOS 器件的输入电阻极大 ,从而使 CMOS 器件驱动同类逻辑门的能力比其他系列强得多。

1.2 MCU (Micro Controller Unit)

中文名称为 微控制单元，又称单片微型计算机 SingleChipMicrocomputer 或者单片机，是指随着大规模集成电路的出现及其发展，将计算机的 CPU 、RAM 、ROM 、定时数计 器和多种 I/O 接口集成在一片芯片上，形成芯片级的计算机，为不同的应用场合做不同组合控制。

1.3 RISC 和 CISC

RISC (reduced instruction set computer)

精简指令集计算机，是一种执行较少类型计算机指令的微处理器，起源于 80 年代的 MIPS 主机（即 RISC 机）， RISC 机中采用的微处理器统称 RISC 处理器。这样一来，它能够以更快的速度执行操作（每秒执行更多百万条指令，即 MIPS）。因为计算机执行每个指令类型都需要额外的晶体管和电路元件，计算机指令集越大就会使微处理器更复杂，执行操作也会更慢。CISC 是复杂指令系统计算机（ Complex Instruction Set Computer ）的简称，微处理器是台式计算机系统的基本处理部件，每个微处理器的核心是运行指令的电路。指令由完成任务的多个步骤所组成，把数值传送进寄存器或进行相加运算。

CISC 复杂指令系统计算机

1.4 DSP (digital signal processor)

是一种独特的 微处理器，是以数字信号来处理大量信息的器件。其工作原理是接收模拟信号，转换为 0 或 1 的数字信号。再对数字信号进行修改、删除、强化，并在其他系统芯片中把数字数据解译回模拟数据或实际环境格式。它不仅具有可编程性，而且其实时运行速度可达每秒数以千万条复杂指令程序，远远超过通用微处理器，是数字化电子世界中日益重要的电脑芯片。它的强大数据处理能力和高运行速度，是最值得称道的两大特色。FPGA (Field

Programmable Gate Array),即现场可编程门阵列,它是在 PAL 、GAL 、CPLD 等可编程器件的基础上进一步发展的产物。它是作为 专用集成电路 (ASIC) 领域中的一种半定制电路而出现的, 既解决了定制电路的不足, 又克服了原有可编程器件门电路数有限的缺点。

1.5 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 ASICApplicationSpecificIC 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。PCIPeripheralComponentInterconnect , 中文意思是“外围器件互联”,一种局部并行总线标准 , 外围组件互连, 一种由英特尔 (Intel) 公司 1991 年推出的用于定义局部 总线的标准。

1.6 ECC

是一Error Correcting Code 的简写,中文名称是 一错误检查和纠正 的简写。ECC 是一种能够实现一错误检查和纠正 的技术,ECC 内存就是应用了这种技术的内存,一般多应用在 服务器及图形工作站上,这将使整个电脑系统在工作时更趋于安全稳定。

1.7 SDRAM

DDR=Double Data Rate 双倍速率同步动态随机存储器。严格的说 DDR 应该叫 DDR SDRAM 人们习惯称为 DDR , 其中, SDRAM 是 Synchronous Dynamic Random Access Memory 的缩写,即同步动态 随机存取存储器。

1.8 IRQ (Interrupt Request)

即是 一中断请求 的意思 (以下使用 IRQ 称呼)。IRQ 的作用就是在我们所用的电脑中,执行硬件中断请求的动作,用来停止其相关硬件的工作状态

1.9 USB (Universal Serial BUS)

是英文 Universal Serial BUS (通用串行总线)的缩写,而其中文简称为 一通串线,是一个外部总线标准,用于规范电脑与外部设备的连接和通讯。

1.10 BIOS (Basic Input Output System)

是英文 “Basic Input Output System” 的缩略语,直译过来后中文名称就是 “基本输入输出系统 “。其实,它是一组固化到计算机内主板上一个 ROM 芯片上的程序,它保存着计算机最重要的基本输入输出的程序、系统设置信息、开机后自检程序和系统自启动程序。其主要功能是为计算机提供最底层的、最直接的硬件设置和控制。

1.11 时钟抖动 (jitter)

由于晶振本身稳定性，电源以及温度变化等原因造成了时钟频率的变化，就是 jitter，指的是时钟周期的变化。指两个时钟周期之间存在的差值，这个误差是在时钟发生器内部产生的，和晶振或者 PLL 内部电路有关，布线对其没有影响。由于跟晶振本身的工艺有关，所以在设计中无法避免它能带来的影响，通产只能在设计中留有一定的 margin。

抖动有两种主要类型：确定性抖动和随机性抖动。

确定性抖动：指由可识别的干扰信号造成的，这种抖动通常幅度有限，具备特定的（而非随机的）产生原因，而且不能进行统计分析。

随机抖动：指由较难预测的因素导致的时序变化。例如，能够影响半导体晶体材料迁移率的温度因素，就可能造成载子流的随机变化。另外，半导体加工工艺的变化，例如掺杂密度不均，也可能造成抖动。

1.12 时钟歪斜 (skew)

是指同样的时钟产生的多个子时钟信号之间的延时差异。skew 通常是时钟相位上的不确定。由于时钟源到达不同寄存器所经历路径的驱动和负载的不同，时钟边沿的位置有所差异，因此就带来了 skew。完成布局布线后，物理路径延时是固定的，所以在设计中考虑到时钟偏移，就可以避免偏移带来的影响。

1.13 数字集成电路

是将元器件和连线集成于同一半导体芯片上而制成的数字逻辑电路或系统。

1.14 模拟信号

是指幅度随时间连续变化的信号。例如，人对着话筒讲话，话筒输出的音频电信号就是模拟信号，收音机、收录机、音响设备及电视机中接收、放大的音频信号、电视信号，也是模拟信号。

1.15 数字信号

是指在时间上和幅度上离散取值的信号，例如，电报电码信号，按一下电键，产生一个电信号，而产生的电信号是不连续的。这种不连续的电信号，一般叫做电脉冲或脉冲信号，计算机中运行的信号是脉冲信号，但这些脉冲信号均代表着确切的数字，因而又叫做数字信号。在电子技术中，通常又把模拟信号以外的非连续变化的信号，统称为数字信号。

1.16 单极型晶体管

也称场效应管，简称 FET(Field Effect Transistor)。它是一种电压控制型器件，由输入电压产生的电场效应来控制输出电流的大小。它工作时只有一种载流子（多数

载流子) 参与导电, 故称为单极型晶体管, 分 MOSFET 和 JFET 两种。

1.17 双极型晶体管

也称晶体三极管, 分 NPN 型管和 PNP 型管, 它是一种电流控制型器件, 由输入电流控制输出电流。它工作时有电子和空穴两种载流子参与导电过程, 故称为双极型三极管。同场效应晶体管相比, 双极型晶体管开关速度慢, 输入阻抗小, 功耗大。双极型晶体管体积小、重量轻、耗电少、寿命长、可靠性高。

1.18 查找表 LUT

本质上就是一个 RAM。目前 FPGA 中多使用 4 输入 LUT, 所以每一个 LUT 可以看成一个有 4 位地址线的 16×1 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, FPGA 开发软件会自动计算逻辑电路的所有可能的结果, 并把结果事先写入 RAM, 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。

1.19 VHDL

VHDL 的英文全写是: VHSIC (Very High Speed Integrated Circuit) Hardware Description Language. 翻译成中文就是超高速集成电路硬件描述语言。主要用于描述数字系统的结构、行为、功能和接口。

1.20 SDR

软件无线电, 一种无线电广播通信技术, 它基于软件定义的无线通信协议而非通过硬连线实现。换言之, 频带、空中接口协议和功能可通过软件下载和更新来升级, 而不用完全更换硬件。SDR 针对构建多模式、多频和多功能无线通信设备的问题提供有效而安全的解决方案。

1.21 RAM (RandomAccessMemory)

随机存储器。存储单元的内容可按需随意取出或存入, 且存取的速度与存储单元的位置无关的存储器。这种存储器在断电时将丢失其存储内容, 故主要用于存储短时间使用的程序。按照存储信息的不同, 随机存储器又分为静态随机存储器 (Static RAM,SRAM) 和动态随机存储器 (Dynamic RAM,DRAM)。

1.22 ROM (Read-OnlyMemory)

只读存储器, 是一种只能读出事先所存数据的固态半导体存储器。其特性是一旦储存资料就无法再将之改变或删除。通常用在不需经常变更资料的电子或电脑系统中, 资料并且不会因为电源关闭而消失。

1.23 SRAM (Static RAM)

静态随机存储器 ,不需要刷新电路即能保存它内部存储的数据。

1.24 DRAM (Dynamic RAM)

动态随机存储器 ,每隔一段时间,要刷新充电一次,否则内部的数据即会消失。因此 SRAM 具有较高的性能,但是 SRAM 也有它的缺点,即它的集成度较低,相同容量的 DRAM 内存可以设计为较小的体积,但是 SRAM 却需要很大的体积,且功耗较大。所以在主板上 SRAM 存储器要占用一部分面积。

SRAM 的速率高、性能好,它主要有如下应用:

- 1) CPU 与主存之间的高速缓存。
- 2) CPU 内部的 L1 / L2 或外部的 L2 高速缓存。

1.25 SSRAM (Synchronous SRAM)

即同步静态随机存取存储器。同步是指 Memory 工作需要同步时钟,内部的命令的发送与数据的传输都以它为基准;随机是指数据不是线性依次存储,而是由指定地址进行数据读写。

对于 SSRAM 的所有访问都在时钟的上升 /下降沿启动。地址、数据输入和其它控制信号均于时钟信号相关。这一点与异步 SRAM 不同,异步 SRAM 的访问独立于时钟,数据输入和输出都由地址的变化控制。

1.26 SDRAM (Synchronous DRAM)

即同步动态随机存取存储器。同步是指 Memory 工作需要同步时钟,内部的命令的发送与数据的传输都以它为基准;动态是指存储阵列需要不断的刷新来保证数据不丢失;随机是指数据不是线性依次存储,而是自由指定地址进行数据读写。

1.27 DDR

Double Data Rate 双倍速率同步动态随机存储器。严格的说 DDR 应该叫 DDR SDRAM,SDRAM 在一个时钟周期内只传输一次数据,它是在时钟的上升期进行数据传输;而 DDR 内存则是一个时钟周期内传输两次数据,它能够在时钟的上升期和下降期各传输一次数据,因此称为双倍速率同步动态随机存储器 DDR 内存是在 SDRAM 内存基础上发展而来的,仍然沿用 SDRAM 生产体系,因此对于内存厂商而言,只需对制造普通 SDRAM 的设备稍加改进,即可实现 DDR 内存的生产,可有效的降低成本。

1.28 DDR2

是 DDRSDRAM 内存的第二代产品 ,虽然同是采用了在时钟的上升 / 下降延同时进行数据传输的基本方式,但 DDR2 内存却拥有两倍于上一代 DDR 内存预读取能力(即: 4bit 数据预读取)。换句话说, DDR2 内存每个时钟能够以 4 倍外部总线的速度读 / 写数据,并且能够以内部控制总线 4 倍的速度运行。

1.29 DDR3

的预取为 8bit, 电压从 DDR2 的 1.8V 降低到 1.5V.DDR3L 是 DDR3 的低电压版, 其工作电压为 1.35V, 1.6Gbps。

1.30 FLASH

即闪存。它是一种长寿命的非易失性(在断电情况下仍能保持所存储的数据信息)的存储器, 数据删除不是以单个的字节为单位而是以固定的区块为单位(注意: NOR Flash 为字节存储。), 区块大小一般为 256KB 到 20MB。

1.31 EEPROM

即电子可擦除只读存储器。EEPROM 与闪存不同的是, 它能在字节水平上进行删除和重写而不是整个芯片擦写, 这样闪存就比 EEPROM 的更新速度快。由于其断电时仍能保存数据, 闪存通常被用来保存设置信息, 如在电脑的 BIOS (基本输入输出程序)、PDA (个人数字助理)、数码相机中保存资料等有限状态机 FSM : 时序电路设计中经常采用的一种方法, 尤其适用于设计数字系统的控制模块, 在一些需要控制高速器件的场合, 用状态机设计具有速度快、结构简单、可靠性高等优点。

1.32 摩尔型状态机 (Moore)

的输出只是当前状态的函数(只与当前状态有关)

1.33 米利型状态机 (Mealy)

的输出则是当前状态和当前输入的函数, 其输出在输入变化后立即变化。摩尔型需等待时钟信号到来、状态变化才导致输出变化, 因此比米里型状态机要多等待一个时钟周期。

1.34 BCD 码

Binary-coded-Decimal. 二进制编码的十进制码, 用 4 位二进制表示 1 位十进制中的 0~9.

1.35 短沟道效应

短沟道效应: 缓变沟道的近似不再成立, 这个二维电势分布会导致阈值电压随 L 的缩短而下降, 亚阈值特征的降级以及由于隧穿穿透效应而使电流饱和失效, 在沟道出现二维电势分布以及高电场, 这些不同于长沟道 MOS 场效应晶体管特性的现象, 统称为短沟道效应。

1.36 闩锁效应

闩锁效应是 CMOS 工艺所特有的寄生效应, 严重会导致电路的失效, 甚至烧毁芯

片。闩锁效应是由 NMOS 的有源区、P 衬底、N 阱、PMOS 的有源区构成的 n-p-n-p 结构产生的，当其中一个三极管正偏时，就会构成正反馈形成闩锁。避免闩锁的方法就是要减小衬底和 N 阱的寄生电阻，使寄生的三极管不会处于正偏状态。

1.37 体效应

对于 NMOS 器件，当其源端电位高于 NMOS 管的体（P 衬底或地）电位时，阈值电压会增加。这一变化称为体效应

2 简答题

2.1 ASIC 设计流程

ASIC 开发基本流程

芯片架构，考虑芯片定义、工艺、封装

RTL 设计，使用 Verilog、System Verilog、VHDL 进行描述

功能仿真，理想情况下的仿真

验证，UVM 验证方法学、FPGA 原型验证

综合，逻辑综合，将描述的 RTL 代码映射到基本逻辑单元门、触发器上

DFT 技术，插入扫描链

等价性检查，使用形式验证技术

STA，静态时序分析

布局规划，保证没有太多的内部交互，避免布线上的拥堵和困扰

时钟树综合，均匀地分配时钟，减少设计中不同部分间的时钟偏移

DRC，设计规则检查

LVS，布线图和原理图进行比较

生成 GDSII，这个流程称为 RTL2GDSII，利用 GDSII 来生产芯片的过程称作流片（Tapeout），

以上是一个 Fabless 公司的简易设计流程，最后将 GDSII 送至 Foundry 生产芯片。

2.2 FPGA 开发基本流程

系统规划，系统功能，功能模块划分

RTL 设计，使用 Verilog、System Verilog、VHDL 进行描述

功能仿真，理想情况下的仿真

综合、编译、布局布线，FPGA 厂商自带工具完成

时序仿真，时序分析约束

板级验证

2.3 什么是同步逻辑和异步逻辑

同步逻辑中时钟之间有固定的因果关系。

异步逻辑中各个时钟之间没有固定的因果关系。

同步时序逻辑电路：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：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。

异步电路主要是组合逻辑电路，用于产生地址译码器、FIFO 或 RAM 的读写控制信号脉冲，其逻辑输出与任何时钟信号都没有关系，译码输出产生的毛刺通常是可以监控的。

同步电路是由时序电路 寄存器和各种触发器 和组合逻辑电路构成的电路，其所有操作都是在严格的时钟控制下完成的。这些时序电路共享同一个时钟 CLK，而所有的状态变化都是在时钟的上升沿 或下降沿 完成的。

2.4 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，只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。

异步电路主要是组合逻辑电路，用于产生地址译码器、FIFO 或 RAM 的读写控制信号脉冲，但它同时也用在时序电路中，此时它没有统一的时钟，状态变化的时刻是不稳定的，通常输入信号只在电路处于稳定状态时才发生变化。也就是

说一个时刻允许一个输入发生变化，以避免输入信号之间造成的竞争冒险。电路的稳定需要有可靠的建立时间和持时间。

同步电路是由时序电路 寄存器和各种触发器 和组合逻辑电路构成的电路，其所有操作都是在严格的时钟控制下完成的。这些时序电路共享同一个时钟 CLK，而所有的状态变化都是在时钟的上升沿 或下降沿完成的。比如 D 触发器，当上升沿到来时，寄存器把 D 端的电平传到 Q 输出端。在同步电路设计中一般采用 D 触发器，异步电路设计中一般采用 Latch。

2.5 时序设计的实质

时序设计的实质就是满足每一个触发器的 建立/保持 时间的要求。

2.6 建立时间与保持时间

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

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

2.7 建立时间和保持时间的作用

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。

换个方式理解：需要建立时间是因为触发器的 D 端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 锁存状态，从后级门传到前级门需要时间。

2.8 亚稳态及抑制方法

这也是一个 异步电路同步化的问题。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 →位同步器，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 \leq 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

2.9 系统最快时钟频率和流水线

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 T_{co} 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； T_{delay} 是组合逻辑的延时； T_{setup} 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 T_{co} ，经过组合逻辑的延时时间为 T_{delay} ，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 $T_{co} + T_{delay} + T_{setup}$ ，也就是说最小的时钟周期 $T_{min} = T_{co} + T_{delay} + T_{setup}$ ，即最快的时钟频率 $F_{max} = 1/T_{min}$ 。FPGA 开发软件也是通过这种方法来计算系统最高运行速度 F_{max} 。因为 T_{co} 和 T_{setup} 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 T_{delay} ，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。

2.10 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，使设计达到时序要求。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。

2.11 附加约束的作用

提高设计的工作频率（减少了逻辑和布线延时）；

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）

指定 FPGA/CPLD 的电气标准和引脚位置。

2.12 多位的异步信号同步

对一位的异步信号可以使用一位同步器进行同步（使用两级触发器，慢时钟到快时钟）；

而对于多位的异步信号，可以采用如下方法：

握手信号：可以采用保持寄存器加握手信号的方法（多数据，控制，地址）；

格雷码：特殊的具体应用电路结构，根据应用的不同而不同；

异步 FIFO（最常用的缓存单元是 DPRAM，双口 RAM）

2.13 FPGA 和 CPLD 的区别

CPLD 以 altramax7000 这种 PLD 为例，

可分为三块结构：宏单元（Macrocell），可编程连线（PIA）和 I/O 控制块。宏单元是 PLD 的基本结构，由它来实现基本的逻辑功能。可编程连线负责信号传递，连接所有的宏单元。I/O 控制块负责输入输出的电气特性控制，比如可以设定集电极开路输出，摆率控制，三态输出等。这种基于乘积项（实际就是与或阵列）的 PLD 基本都是由 EEPROM 和 Flash 工艺制造的，一上电就可以工作，无需其他芯片配合。布线方式是全局的，所以延时可预测。CPLD 适合做逻辑设计。

FPGA 基于 LUT，

LUT 本质上就是一个 RAM，每一个 LUT 可以看成是一个有 4 位地址线的 16x1 的 RAM。这也是为什么 FPGA 需要外接一个 rom 来上电配置。以 xilinx 的 Spartan-II 为例，主要包括 CLBs, I/O 块, RAM 块和可编程连线。在 spartan-II 中，一个 CLB 包括 2 个 Slices, 每个 slices 包括两个 LUT，两个触发器和相关逻辑。Slices 可以看成是 SpartanII 实现逻辑的最基本结构。FPGA 的制造工艺确定了 FPGA 芯片中包含的 LUT 和触发器的数量非常多，往往都是几千上万，PLD 一般只能做到 512 个逻辑单元，而且如果用芯片价格除以逻辑单元数量，FPGA 的平均逻辑单元成本大大低于 PLD。所以如果设计中使用到大量触发器，例如设计一个复杂的时序逻辑，那么使用 FPGA 就是一个很好选择。

CPLD 更适合完成各种算法和组合逻辑，FPGA 更适合于完成时序逻辑。换句话说，FPGA 更适合于触发器丰富的结构，而 CPLD 更适合于触发器有限而乘积项丰富的结构。

CPLD 的连续式布线结构决定了它的时序延迟是均匀的和可预测的，而 FPGA 的分段式布线结构决定了其延迟的不可预测性。

在编程上 FPGA 比 CPLD 具有更大的灵活性。CPLD 通过修改具有固定内连电路的逻辑功能来编程，FPGA 主要通过改变内部连线的布线来编程；FPGA 可在逻辑门下编程，而 CPLD 是在逻辑块下编程。

FPGA 的集成度比 CPLD 高，具有更复杂的布线结构和逻辑实现。

CPLD 比 FPGA 使用起来更方便。CPLD 的编程采用 E2PROM 或 FASTFLASH 技术，无需外部存储器芯片，使用简单。而 FPGA 的编程信息需存放在外部存储器上，使用方法复杂。

CPLD 的速度比 FPGA 快，并且具有较大的时间可预测性。这是由于 FPGA 是门

级编程，并且 CLB 之间采用分布式互联，而 CPLD 是逻辑块级编程，并且其逻辑块之间的互联是集总式的。

在编程方式上，CPLD 主要是基于 E2PROM 或 FLASH 存储器编程，编程次数可达 1 万次，优点是系统断电时编程信息也不丢失。CPLD 又可分为在编程器上编程和在系统编程两类。FPGA 大部分是基于 SRAM 编程，编程信息在系统断电时丢失，每次上电时，需从器件外部将编程数据重新写入 SRAM 中。其优点是可以编程任意次，可在工作中快速编程，从而实现板级和系统级的动态配置。CPLD 可加密保密性好，FPGA 保密性差。

一般情况下，CPLD 的功耗要比 FPGA 大，且集成度越高越明显。随著复杂可编程逻辑器件 CPLD 密度的提高，数字器件设计人员在进行大型设计时，既灵活又容易，而且产品可以很快进入市场许多设计人员已经感受到 CPLD 容易使用。时序可预测和速度高等优点，然而，在过去由于受到 CPLD 密度的限制，他们只好转向 FPGA 和 ASIC。现在，设计人员可以体会到密度高达数十万门的 CPLD 所带来的好处。

	CPLD	FPGA
内部结构	Product term 基于乘积项	Look up Table 基于查找表
程序存储	内部 EEPROM/FLASH	SRAM，外挂 EEPROM
资源类型	组合逻辑资源丰富	时序逻辑资源丰富
集成度	低	高
使用场合	完成控制逻辑	实现复杂算法
其他资源	—	PLL、RAM、乘法器

2.14 锁存器和触发器

锁存器（latch）和**触发器**（flip-flop）

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。

2.15 FPGA 芯片内两种存储器资源

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。

2.16 门控时钟

在许多应用中，整个设计项目都采用外部的全局时钟是不可能或不实际的。PLD 具有乘积项逻辑阵列时钟（即时钟是由逻辑产生的），允许任意函数单独地钟控各个触发器。然而，当你用阵列时钟时，应仔细地分析时钟函数，以避免毛刺。通常用阵列时钟构成门控时钟。门控时钟常常同微处理器接口有关，用地址线去

控制写脉冲。然而，每当用组合函数钟控触发器时，通常都存在着门控时钟。如果符合下述条件，门控时钟可以象全局时钟一样可靠地工作：

2.17 FPGA 设计中时钟的使用

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）。

2.18 FPGA 中实现同步时序电路的延时

异步电路：一般是通过加 buffer、两级与非门等来实现延时（不是很清楚），但这是不合同步电路实现延时的。

同步电路：对于比较大的和特殊要求的延时，一般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，可以多打几拍。

2.19 FPGA 综合实现为 RAM/ROM/CAM 的注意事项

三种资源：BLOCK RAM，触发器（FF），查找表（LUT）；注意事项：

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。

弄清 FPGA 的硬件结构，合理使用 BLOCK RAM 资源；

分析 BLOCK RAM 容量，高效使用 BLOCK RAM 资源；

分布式 RAM 资源（DISTRIBUTE RAM）

2.20 Xilinx 中与全局时钟资源和 DLL 相关的硬件原语

常用的与全局时钟资源相关的 Xilinx 器件原语包括：

IBUFG, IBUFGDS, BUFG, BUFGP, BUFGCE, BUFGMUX, BUFGDLL, DCM 等

2.21 HDL 语言的层次概念

HDL 语言是分层次的、类型的，最常用的层次概念有系统与标准级、功能模块级，行为级，寄存器传输级和门级。

系统级，算法级，RTL 级行为级，门级，开关级

2.22 查找表的原理与结构

查找表 (look-up-table) 简称为 LUT , 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 可以看成是一个有 4 位地址线的 16x1 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的结果, 并把结果事先写入 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

2.23 IC 设计前端到后端的流程和 EDA 工具

设计前端也称逻辑设计, 后端设计也称物理设计, 两者并没有严格的界限, 一般涉及到与工艺有关的设计就是后端设计。

规格制定: 客户向芯片设计公司提出设计要求。

详细设计: 芯片设计公司 (Fabless) 根据客户提出的规格要求, 拿出设计解决方案和具体实现架构, 划分模块功能。目前架构的验证一般基于 systemC 语言, 对价后模型的仿真可以使用 systemC 的仿真工具。例如: CoCentric 和 Visual Elite 等。

HDL 编码: 设计输入工具: ultra , visual VHDL 等

仿真实证: modelsim

逻辑综合: synplify

静态时序分析: synopsys 的 Prime Time

形式验证: Synopsys 的 Formality.

2.24 寄生效应在 IC 设计中怎样加以克服和利用

所谓寄生效应就是那些在你的 PCB 并在电路中大施破坏、令人头痛、原因不明的小故障。它们就是 渗入高速电路中隐藏的寄生电容和寄生电感。其中包括由封装引脚和印制线过长形成的寄生电感; 焊盘到地、焊盘到电源平面和焊盘到印制线之间形成的寄生电容; 通孔之间的相互影响, 以及许多其它可能的寄生效应。

理想状态下, 导线是没有电阻, 电容和电感的。而在实际中, 导线用到了金属铜, 它有一定的电阻率, 如果导线足够长, 积累的电阻也相当可观。两条平行的导线, 如果互相之间有电压差异, 就相当于形成了一个平行板电容器 (你想象一下)。通电的导线周围会形成磁场 (特别是电流变化时), 磁场会产生感生电场, 会对电子的移动产生影响, 可以说每条实际的导线包括元器件的管脚都会产生感生电动势, 这也就是寄生电感。在直流或者低频情况下, 这种寄生效应看不太出来。而在交流特别是高频交流条件下, 影响就非常巨大了。根据复阻抗公式, 电容、电感会在交流情况下会对电流的移动产生巨大阻碍, 也就可以折算成阻抗。这种寄生效应很难克服, 也难摸到。只能通过优化线路, 尽量使用管脚短的 SMT 元器件来减少其影响, 要完全消除是不可能的。

2.25 什么是“线与”逻辑，在硬件特性上的具体要求

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。od 门是漏极开路门。

2.26 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：加滤波电容，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。

门电路两个输入信号同时向相反的逻辑电平跳变称为竞争；由于竞争而在电路的输出端可能产生尖峰脉冲的现象称为竞争冒险。如果逻辑函数在一定条件下可以化简成 $Y=A+A'$ 或 $Y=AA'$ 则可以判断存在竞争冒险现象（只是一个变量变化的情况）。消除方法，接入滤波电容，引入选通脉冲，增加冗余逻辑

2.27 常用逻辑电平？TTL 与 COMS 电平可以直接互连吗？

常用逻辑电平：TTL、CMOS、LVTTTL、LVCMOS、ECL(Emitter Coupled Logic)、PECL(Pseudo/Positive Emitter Coupled Logic)、LVDS(Low Voltage Differential Signaling)、GTL(Gunning Transceiver Logic)、BTL(Backplane Transceiver Logic)、ETL(enhanced transceiver logic)、GTLP(Gunning Transceiver Logic Plus)；RS232、RS422、RS485(12V, 5V, 3.3V)；

也有一种答案是：常用逻辑电平：12V，5V，3.3V。

TTL 和 CMOS 不可以直接互连，由于 TTL 是在 0.3-3.6V 之间，而 CMOS 则是有在 12V 的有在 5V 的。

CMOS 接 TTL：是可以直接互连。

TTL 接 CMOS：需要在输出端口加一上拉电阻接到 5V 或者 12V。用 CMOS 可

直接驱动 TTL；加上拉电阻后，TTL 可驱动 CMOS。

上拉电阻用途：

当 TTL 电路驱动 COMS 电路时，如果 TTL 电路输出的高电平低于 COMS 电路的最低高电平（一般为 3.5V），这时就需要在 TTL 的输出端接上拉电阻，以提高输出高电平的值。

OC 门电路必须加上拉电阻，以提高输出的高电平值。

为加大输出引脚的驱动能力，有的单片机管脚上也常使用上拉电阻。

在 COMS 芯片上，为了防止静造成损坏，不用的管脚不能悬空，一般接上拉电阻产生降低输入阻抗，提供泄荷通路。

芯片的管脚加上拉电阻来提高输出电平，从而提高芯片输入信号的噪声容限增强抗干扰能力。

提高总线的抗电磁干扰能力。管脚悬空就比较容易接受外界的电磁干扰。

长线传输中电阻不匹配容易引起反射波干扰，加上下拉电阻是电阻匹配，有效的抑制反射波干扰。

上拉电阻阻值的选择原则包括：

从节约功耗及芯片的灌电流能力考虑应当足够大；电阻大，电流小。

从确保足够的驱动电流考虑应当足够小；电阻小，电流大。

对于高速电路，过大的上拉电阻可能边沿变平缓。综合考虑以上三点，通常在 1k 到 10k 之间选取。对下拉电阻也有类似道理。OC 门电路必须加上拉电阻，以提高输出的高电平值。OC 门电路要输出 -1 时才需要加上拉电阻不加根本就没有高电平在有时我们用 OC 门作驱动（例如控制一个 LED）灌电流工作时就可以不加上拉电阻总之加上拉电阻能够提高驱动能力。

2.28 Moore 与 Mealy 状态机的特征

Moore 状态机：输出仅与当前状态值有关，且只在时钟边沿到来时才会有状态变化。

Mealy 状态机：输出不仅与当前状态值有关，而且与当前输入值有关。

2.29 多时域设计中，如何处理信号跨时域

不同的时钟域之间信号通信时需要进行同步处理，这样可以防止新时钟域中第一级触发器的亚稳态信号对下级逻辑造成影响。

单个信号跨时钟域时：可以采用两级触发器来同步；

数据或地址总线跨时钟域：采用异步 FIFO 来实现时钟同步；格雷码；第三种方法就是采用握手信号。

2.30 静态、动态时序模拟的优缺点

静态时序分析：是采用穷尽分析方法来提取出整个电路存在的所有时序路径，计

算信号在这些路径上的传播延时，检查信号的建立和保持时间是否满足时序要求，通过对最大路径延时和最小路径延时的分析，找出违背时序约束的错误。它不需要输入向量就能穷尽所有的路径，且运行速度很快、占用内存较少，不仅可以对芯片设计进行全面的时序功能检查，而且还可利用时序分析的结果来优化设计，因此静态时序分析已经越来越多地被用到数字集成电路设计的验证中。

动态时序模拟：就是通常的仿真，因为不可能产生完备的测试向量，覆盖门级网表中的每一条路径。因此在动态时序分析中，无法暴露一些路径上可能存在的时序问题；

2.31 四级的 Mux，第二级信号为关键信号，如何改善 timing

关键：将第二级信号放到最后输出一级输出，同时注意修改片选信号，保证其优先级未被修改。

2.32 关键路径

关键路径就是输入到输出延时最大的路径，找到了关键路径便能求得最大时钟频率。

2.33 标准的倒相器中 P 管的宽长比要比 N 管的宽长比大

和载流子有关，P 管是空穴导电，N 管是电子导电，电子的迁移率大于空穴，同样的电场下，N 管的电流大于 P 管，因此要增大 P 管的宽长比，使之对称，这样才能使得两者上升时间下降时间相等、高低电平的噪声容限一样、充电放电的时间相等。

2.34 什么是状态图

状态图是以几何图形的方式来描述时序逻辑电路的状态转移规律以及输出与输入的关系

可以分为摩尔型和米利型，注意区分

2.35 可编程逻辑器件有哪些

PAL，PLA，GAL，CPLD，FPGA

PAL（可编程阵列逻辑），

PLD（可编程逻辑器件），

CPLD（复杂可编程逻辑器件、Complex Programmable Logic Device），

FPGA（现场可编程门阵列、Field-Programmable Gate Array）。

ROM 只读存储器、PLA 可编程逻辑阵列、FPLA 现场可编程逻辑阵列、PAL 可编程阵列逻辑 GAL 通用阵列逻辑、EPLD 可擦除的可编程逻辑器件、FPGA 现场可编程门阵列、CPLD 复杂可编程逻辑器件等，其中 ROM、FPLA、PAL、GAL、EPLD

是出现较早的可编程逻辑器件，而 FPGA 和 CPLD 是当今最流行的两类可编程逻辑器件。
FPGA 是基于查找表结构的，而 CPLD 是基于乘积项结构的。

2.36 SRAM、FLASH、DRAM、SSRAM 及 SDRAM 的区别

SRAM：静态随机存储器，存取速度快，但容量小，掉电后数据会丢失，不像 DRAM 需要不停的 REFRESH，制造成本较高，通常用来作为快取 CACHE 记忆体使用。

FLASH：闪存，存取速度慢，容量大，掉电后数据不会丢失

DRAM：动态随机存储器，必须不断的重新加强 REFRESHED 电位差量，否则电位差将降低至无法有足够的能量表现每一个记忆单位处于何种状态。价格比 SRAM 便宜，但访问速度较慢，耗电量较大，常用作计算机的内存使用。

SSRAM：即同步静态随机存取存储器。对于 SSRAM 的所有访问都在时钟的上升/下降沿启动。地址、数据输入和其它控制信号均于时钟信号相关。

SDRAM：即同步动态随机存取存储器。

2.37 四种复用方式

四种复用方式：频分多路复用（FDMA），时分多路复用（TDMA），码分多路复用（CDMA），波分多路复用（WDMA）。

2.38 修正 Setup time violation 和 Hold time violation

ASIC 设计流程中什么时候修正 Setup time violation 和 Hold time violation?

如何修正？解释 setup 和 hold time violation，画图说明，并说明解决办法。

Setup/hold time 是测试芯片对输入信号和时钟信号之间的时间要求。建立时间是指触发器的时钟信号上升沿到来以前，数据稳定不变的时间。输入信号应提前时钟上升沿如上升沿有效 T 时间到达芯片，这个 T 就是建立时间 -Setup time。如不满足 setup time，这个数据就不能被这一时钟打入触发器，只有在下一个时钟上升沿，数据才能被打入触发器。保持时间是指触发器的时钟信号上升沿到来以后，数据稳定不变的时间。如果 hold time 不够，数据同样不能被打入触发器。建立时间 SetupTime 和保持时间 Holdtime。建立时间是指在时钟边沿前，数据信号需要保持不变的时间。保持时间是指时钟跳变边沿后数据信号需要保持不变的时间。如果不满足建立和保持时间的话，那么 DFF 将不能正确地采样到数据，将会出现 metastability 的情况。如果数据信号在时钟沿触发前后持续的时间均超过建立和保持时间，那么超过量就分别被称为建立时间裕量和保持时间裕量。

2.39 如何防止亚稳态

亚稳态是指触发器无法在某个规定时间段内达到一个可确认的状态。当一个触发器进入亚稳态时，既无法预测该单元的输出电平，也无法预测何时输出才能稳定在某个正确的电平上。在这个稳定期间，触发器输出一些中间级电平，或者

可能处于振荡状态，并且这种无用的输出电平可以沿信号通道上的各个触发器级联式传播下去。

解决方法：

- 1、降低系统时钟频率
- 2、用反应更快的 FF
- 3、引入同步机制，防止亚稳态传播（可以采用前面说的加两级触发器）。
- 4、改善时钟质量，用边沿变化快速的时钟信号

2.40 基尔霍夫定理的内容

基尔霍夫定律包括电流定律和电压定律：

电流定律： 在集成电路中，在任一瞬时，流向某一结点的电流之和恒等于由该结点流出的电流之和。

电压定律： 在集成电路中，在任一瞬间，沿电路中的任一回路绕行一周，在该回路上电动势之和恒等于各电阻上的电压降之和。

2.41 反馈电路的概念及应用。

反馈，就是在电路系统中，把输出回路中的电量（电压或电流）输入到输入回路中去。

反馈的类型有：电压串联负反馈、电流串联负反馈、电压并联负反馈、电流并联负反馈。

负反馈的优点：降低放大器的增益灵敏度，改变输入电阻和输出电阻，改善放大器的线性和非线性失真，有效地扩展放大器的通频带，自动调节作用。

电压负反馈的特点：电路的输出电压趋向于维持恒定。

电流负反馈的特点：电路的输出电流趋向于维持恒定。

2.42 有源滤波器和无源滤波器的区别

无源滤波器：这种电路主要有无源元件 R、L 和 C 组成

有源滤波器：集成运放和 R、C 组成，具有不用电感、体积小、重量轻等优点。集成运放的开环电压增益和输入阻抗均很高，输出电阻小，构成有源滤波电路后还具有一定的电压放大和缓冲作用。但集成运放带宽有限，所以目前的有源滤波电路的工作频率难以做得很高。

2.43 建立、保持时间

$$T_{\text{delay}} < T_{\text{period}} - T_{\text{setup}} - T_{\text{co}}$$

$$T_{\text{period}} > T_{\text{setup}} + T_{\text{co}} + T_{\text{delay}} \quad (\text{用来计算最高时钟频率})$$

T_{co} 即触发器的传输延时

时钟周期为 T ，触发器 D1 的寄存器到输出时间（触发器延时 T_{co} ）最大为 $T_{1\text{max}}$ ，最小为 $T_{1\text{min}}$ 。组合逻辑电路最大延迟为 $T_{2\text{max}}$ ，最小为 $T_{2\text{min}}$ 。问，触发器 D2 的建立时间 T_3 和保持时间应满足什么条件。

$T_{3\text{setup}} > T + T_{2\text{max}}$ 时钟沿到来之前数据稳定的时间（越大越好），一个时钟周期

T 加上最大的逻辑延时。

$T_{3hold} > T_{1min} + T_{2min}$ 时钟沿到来之后数据保持的最短时间，一定要大于最小的延时也就是 $T_{1min} + T_{2min}$

给出某个一般时序电路的图，有 T_{setup} ， T_{delay} ， $T_{ck \rightarrow q}(T_{co})$ ，还有 clock 的 delay，写出决定最大时钟的因素，同时给出表达式。

$T + T_{clkdealy} > T_{setup} + T_{co} + T_{delay}$;

$T_{hold} > T_{clkdelay} + T_{co} + T_{delay}$; 保持时间与时钟周期无关

2.44 OTP 片 OTP（一次性可编程）、掩膜片，两者的区别何在？

OTP 与掩膜 OTP 是一次性写入的单片。过去认为一个单片机产品的成熟是以投产掩膜型单片机为标志的。由于掩膜需要一定的生产周期，而 OTP 型单片机价格不断下降，使得近年来直接使用 OTP 完成最终产品制造更为流行。它较之掩膜具有生产周期短、风险小的特点。近年来，OTP 型单片机需量大幅度上扬，为适应这种需求许多单片机都采用了在系

统编程技术 InSystemProgramming。未编程的 OTP 芯片可采用裸片 Bonding 技术或表面贴技术，先焊在印刷板上，然后通过单片机上引出的编程线、串行数据、时钟线等对单片机编程。解决了批量写 OTP 芯片时容易出现的芯片与写入器接触不好的问题。使 OTP 的裸片得以广泛使用，降低了产品的成本。编程线与 I/O 线共用，不增加单片机的额外引脚。而一些生产厂商推出的单片机不再有掩膜型，全部为有 ISP 功能的 OTP。

2.45 描述你对集成电路设计流程的认识

制定规格书 - 任务划分 - 设计输入 - 功能仿真 - 综合 - 优化 - 布局布线 - 时序仿真时序分析 - 芯片流片 - 芯片测试验证

2.46 描述你对集成电路工艺的认识

工艺分类：TTL，CMOS 两种比较流行，TTL 速度快功耗高，CMOS 速度慢功耗低。集成电路的工艺主要是指 CMOS 电路的制造工艺，主要分为以下几个步骤：衬底准备 - 氧化、光刻 - 扩散和离子注入 - 淀积 - 刻蚀 - 平面化。

2.47 简述 FPGA 等可编程逻辑器件设计流程

通常可将 FPGA/CPLD 设计流程归纳为以下 7 个步骤，这与 ASIC 设计有相似之处。

1、**设计输入**。Verilog 或 VHDL 编写代码。

2、**前仿真**（功能仿真）。设计的电路必须在布局布线前验证电路功能是否有效。（ASIC 设计中，这一步骤称为第一次 Sign-off）PLD 设计中，有时跳过这一步。

3、**设计编译（综合）**。设计输入之后就有一个从高层次系统行为设计向门级逻辑电路设转化翻译过程，即把设计输入的某种或某几种数据格式网表转化为软件可识别的某种数据格式网表。

4、**优化**。对于上述综合生成的网表，根据布尔方程功能等效的原则，用更小更快的综合结果代替一些复杂的单元，并与指定的库映射生成新的网表，这是减小电路规模的一条必由之路。

5、**布局布线**。

6、**后仿真（时序仿真）** 需要利用在布局布线中获得的精确参数再次验证电路的时序。（ASCI 设计中，这一步骤称为第二次 Sign-off）。

7、**生产**。布线和后仿真完成之后，就可以开始 ASCII 或 PLD 芯片的投产

2.48 数字 IC 设计流程

1. 需求分析 制定规格书。分析用户或市场的需求，并将其翻译成对芯片产品的技术需求。
2. 算法设计。设计和优化芯片中所使用的算法。这一阶段一般使用高级编程语言（如 C/C++），利用算法级建模和仿真工具（如 MATLAB，SPW）进行浮点和定点的仿真，进而对算法进行评估和优化。
3. 构架设计。根据设计的功能需求和算法分析的结果，设计芯片的构架，并对不同的方案进行比较，选择性能价格最优的方案。这一阶段可以使用 SystemC 语言对芯片构架进行模拟和分析。
4. RTL 设计（代码输入）。使用 HDL 语言完成对设计实体的 RTL 级描述。这一阶段使用 VHDL 和 Verilog HDL 语言的输入工具编写代码。
5. RTL 验证（功能仿真）。使用仿真工具或其他 RTL 代码分析工具，验证 RTL 代码的质量和性能。
6. 综合。从 RTL 代码生成描述实际电路的门级网表文件。
7. 门级验证（综合后仿真）。对综合产生的门级网表进行验证。这一阶段通常会使用仿真、静态时序分析和形式验证等工具。
8. 布局布线。后端设计对综合产生的门级网表进行布局规划（Floorplanning）、布局（Placement）、布线（Routing），生成生产用的版图。
9. 电路参数提取确定芯片中互连线的寄生参数，从而获得门级的延时信息。
10. 版图后验证。根据后端设计后取得的新的延时信息，再次验证设计是否能够实现所有的功能和性能指标。
11. 芯片生产。生产在特定的芯片工艺线上制造出芯片。
12. 芯片测试。对制造好的芯片进行测试，检测生产中产生的缺陷和问题。

2.49 数字 IC 后端设计流程

1. 数据准备。对于 Cadance 的 SE 而言后端设计所需的数据主要有 Foundry 厂提供的标准单元、宏单元和 I/O Pad 的库文件，它包括物理库、时序库及网表库，分别以 .lef、.tlf 和 .v 的形式给出。前端的芯片设计经过综合后生成的门级网表，具有时序约束和时钟定义的脚本文件和由此产生的 .gcf 约束文件以及定义电源 Pad 的 DEF（Design Exchange Format）文件。对 synopsys 的 Astro 而言，经过综合后生成的门级网表，时序约束文件 SDC 是一样的，Pad 的定义文件 - tdf，.tf 文件 - technologyfile，Foundry 厂提供的标准单元、宏单元和 I/O Pad 的库文件就与 FRAM, CELLview, LMview 形式给出（Milkway 参考库 andDB, LIBfile

2. 布局规划。主要是标准单元、I/O Pad 和宏单元的布局。I/O Pad 预

先给出了位置，而宏单元则根据时序要求进行摆放，标准单元则是给出了一定的区域由工具自动摆放。布局规划后，芯片的大小，Core 的面积，Row 的形式、电源及地线的 Ring 和 Strip 都确定下来了。如果必要在自动放置标准单元和宏单元之后，你可以先做一次 PNA(power network analysis) - IR drop and EM。

3.Placement -自动放置标准单元。布局规划后，宏单元、I/O Pad 的位置和放置标准单元的区域都已确定，这些信息 SE(Silicon Ensemble)会通过 DEF 文件传递给 PCPhysicalCompiler,PC 根据由综合给出的 .DB 文件获得网表和时序约束信息进行自动放置标准单元，同时进行时序检查和单元放置优化。如果你用的是 PC +Astro 那你可用 write_milkway, read_milkway 传递数据。

4.时钟树生成 CTSClocktreesynthesis。芯片中的时钟网络要驱动电路中所有的时序单元，所以时钟源端门单元带载很多，其负载延时很大并且不平衡，需要插入缓冲器减小负载和平衡延时。时钟网络及其上的缓冲器构成了时钟树。一般要反复几次才可以做出一个比较理想的时钟树。

5.STA 静态时序分析和后仿真。时钟树插入后，每个单元的位置都确定下来了，工具可以提出 Global Route 形式的连线寄生参数，此时对延时参数的提取就比较准确了。SE 把 .V 和 .SDF 文件传递给 PrimeTime 做静态时序分析。确认没有时序违规后，将这来两个文件传递给前端人员做后仿真。对 Astro 而言，在 detail routing 之后，用 starRC XT 参数提取，生成的 E.V 和 .SDF 文件传递给 PrimeTime 做静态时序分析，那将会更准确。

6.ECOEngineeringChangeOrder。针对静态时序分析和后仿真中出现的问题，对电路和单元布局进行小范围的改动。

7.filler 的插入 padfliier,cellfiller。Filler 指的是标准单元库和 I/O Pad 库中定义的与逻辑无关的填充物，用来填充标准单元和标准单元之间，I/O Pad 和 I/O Pad 之间的间隙，它主要是把扩散层连接起来，满足 DRC 规则和设计需要。

8.布线 Routing。Global route - Track assign -Detail routing - Routing optimization 布线是指在满足工艺规则和布线层数限制、线宽、线间距限制和各线网可靠绝缘的电性能约束的条件下，根据电路的连接关系将各单元和 I/O Pad 用互连线连接起来，这些是在时序驱动 Timingdriven 的条件下进行的，保证关键时序路径上的连线长度能够最小。 -Timing report clear

9.Dummy Metal 的增加。Foundry 厂都有对金属密度的规定，使其金属密度不要低于一定的值，以防在芯片制造过程中的刻蚀阶段对连线的金属层过度刻蚀从而降低电路的性能。加入 Dummy Metal 是为了增加金属的密度。

10.DRC 和 LVS。DRC 是对芯片版图中的各层物理图形进行设计规则检查 spacing,width,它也包括天线效应的检查，以确保芯片正常流片。LVS 主要是将版图和电路网表进行比较，来保证流片出来的版图电路和实际需要的电路一致。DRC 和 LVS 的检查 -EDA 工具 Synopsys hercules/mentor calibre/CDN Dracula 进行的 .Astro also include LVS/DRC check commands.

11.Tape out。在所有检查和验证都正确无误的情况下把最后的版图 GDS II 文件传递给 Foundry 厂进行掩膜制造

2.50 自动布局布线？一两种工具软件，需要哪些基本元素？

自动布局布线 其基本流程如下：

- 1、读入网表,跟 foundry 提供的标准单元库和 Pad 库以及宏模块库进行映射;
- 2、整体布局,规定了芯片的大致面积和管脚位置以及宏单元位置等粗略的信息;
- 3、读入时序约束文件, 设置好 timing setup 菜单, 为后面进行时序驱动的布局布线做准备;
- 4、详细布局, 力求使后面布线能顺利满足布线布通率 100% 的要求和时序的要求;
- 5、时钟树综合, 为了降低 clock skew 而产生由许多 buffer 单元组成的 一时钟树 II;
- 6、布线, 先对电源线和时钟信号布线, 然后对信号线布线, 目标是最大程度地满足时序;
- 7、为满足 design rule 从而 foundry 能成功制造出该芯片而做的修补工作, 如填充一些 dummy 等。

常用的工具有 Synopsys 的 ASTRO , Cadence 的 SE, ISE, Quartus II 也可实现布局布线

2.51 列举几种集成电路典型工艺。常提到 0.25, 0.18 指的是什么?

典型工艺: 氧化, 离子注入, 光刻, 刻蚀, 扩散, 淀积。 /0.13, 90, 65
制造工艺: 我们经常说的 0.18 微米、0.13 微米制程, 就是指制造工艺了。制造工艺直接关系到 cpu 的电气性能。而 0.18 微米、0.13 微米这个尺度就是指的是 cpu 核心中 线路的宽度。线宽越小, cpu 的功耗和发热量就越低, 并可以工作在更高的频率上了。所以以前 0.18 微米的 cpu 最高的频率比较低, 用 0.13 微米制造工艺的 cpu 会比 0.18 微米的制造工艺的发热量低都是这个道理了。

2.52 半导体工艺中, 掺杂有哪几种方式

根据掺入的杂质不同, 杂质半导体可以分为 N 型和 P 型两大类。N 型半导体中掺入的杂质为磷等五价元素, 磷原子在取代原晶体结构中的原子并构成共价键时, 多余的第五个价电子很容易摆脱磷原子核的束缚而成为自由电子, 于是半导体中的自由电子数目大量增加, 自由电子成为多数载流子, 空穴则成为少数载流子。P 型半导体中掺入的杂质为硼或其他三价元素, 硼原子在取代原晶体结构中的原子并构成共价键时, 将因缺少一个价电子而形成空穴, 于是半导体中的空穴数目大量增加, 空穴成为多数载流子, 而自由电子则成为少数载流子。

热扩散, 离子注入

2.53 描述 CMOS 电路中闩锁效应产生的过程及最后的结果

闩锁效应是 CMOS 工艺所特有的寄生效应, 严重会导致电路的失效, 甚至烧毁芯片。闩锁效应是由 NMOS 的有源区、P 衬底、N 阱、PMOS 的有源区构成的 n-p-n-p 结构产生的, 当其中一个三极管正偏时, 就会构成正反馈形成闩锁。

避免闩锁的方法就是要减小衬底和 N 阱的寄生电阻，使寄生的三极管不会处于正偏状态。静电是一种看不见的破坏力，会对电子元器件产生影响。ESD 和相关的电压瞬变都会引起闩锁效应（latch-up）是半导体器件失效的主要原因之一。如果有一个强电场施加在器件结构中的氧化物薄膜上，则该氧化物薄膜就会因介质击穿而损坏。很细的金属化迹线会由于大电流而损坏，并会由于浪涌电流造成的过热而形成开路。这就是所谓的“闩锁效应”。在闩锁情况下，器件在电源与地之间形成短路，造成大电流、EOS（电过载）和器件损坏。

2.54 解释 latch-up 现象和 Antenna effect 及其预防措施

在芯片生产过程中，暴露的金属线或者多晶硅 polysilicon 等导体，就象是一根根天线，会收集电荷（如等离子刻蚀产生的带电粒子）导致电位升高。天线越长，收集的电荷也就越多，电压就越高。若这片导体碰巧只接了 MOS 的栅，那么高电压就可能把薄栅氧化层击穿，使电路失效，这种现象我们称之为“天线效应”。随着工艺技术的发展，栅的尺寸越来越小，金属的层数越来越多，发生天线效应的可能性就越大

2.55 FPGA 公司

目前世界上有十几家生产 CPLD/FPGA 的公司，最大的两家是
答案：Xilinx、Altera

2.56 可编程单元

FPGA 的基本结构由 3 种可编程单元和一个用于存放编程数据的静态存储器组成。这 3 种可编程的单元分别是（）、（）和（）。

答案：IOB——输入输出模块、CLB——可编程逻辑模块、IR——互联资源或可编程内部连线

2.57 时序组合逻辑

Verilog 语言信号赋值包括非阻塞赋值和阻塞赋值，一般非阻塞赋值用在描述中，阻塞赋值用在描述中；

答案：时序电路、组合逻辑

2.58 列出你熟悉的前仿真验证工具，并以一种为例简述仿真步骤？

答案：modelsim、vcs、NC 等；步骤：

- 1) 创建工程 newproject
- 2) 导入设计文件（add file）
- 3) 编译文件（compile）
- 4) 添加仿真库（library）

- 5) 加载设计 loaddesign
- 6) 运行仿真器 (run)

2.59 列出仿真过程中常用的 verilog 系统任务并描述作用

- 1). 显示任务: display、write

主要作用是在仿真过程中将仿真数据打印输出到屏幕;

- 2). 监控任务: monitor 主要用于监控仿真过程中指定的参数, 并将参数值输出打印至屏幕; 3) 探测任务: strobe 主要用于在指定的时间之后显示仿真数据;

- 4) 文件输入和输出任务: fopen、fdisplay、fclose 用于打开硬盘上的某个文件。主要用于将仿真信息写入到某个文件, 形成仿真日志, 待仿真完成后分析; fread、readmemb 和 readmemh 用于从文件中读出数据; 可以将仿真读取的数据加载到存储器中, 待需要使用的时候取出数据;

- 5) 模拟控制任务

stop: 该任务可以使仿真器被挂起。finish : 该任务使仿真器退出;

- 6) 模拟时间任务

time: 用于返回 64 位整型模拟时间; stime : 返回 32 位时间 realtime: 返回实型模拟时间;

- 7) random: 用于产生 32 位有符号整数;

2.60 PC Chipset

芯片组 (Chipset) 是主板的核心组成部分, 按照在主板上的排列位置的不同, 一般分为北桥芯片和南桥芯片。北桥芯片提供对 CPU 的类型和主频、内存的类型和最大容量、ISA/PCI/AGP 插槽、ECC 纠错等支持。南桥芯片则提供对 KBC (键盘控制器)、RTC (实时时钟控制器)、USB (通用串行总线)、Ultra DMA/3366EIDE 数据传输方式和 ACPI (高级能源管理) 等的支持。其中北桥芯片起着主导性的作用, 也称为主桥 (Host Bridge)。除了最通用的南北桥结构外, 当前芯片组正向更高级的加速集线架构发展, Intel 的 8xx 系列芯片组就是这类芯片组的代表, 它将一些子系统如 IDE 接口、音效、MODEM 和 USB 直接接入主芯片, 能够提供比 PCI 总线宽一倍的带宽, 达到资料仅供参考了 266MB/s。

2.61 短沟道效应

短沟道效应: 缓变沟道的近似不再成立, 这个二维电势分布会导致阈值电压随 L 的缩短而下降, 亚阈值特征的降级以及由于隧穿穿透效应而使电流饱和失效, 在沟道出现二维电势分布以及高电场, 这些不同于长沟道 MOS 场效应晶体管特性的现象, 统称为短沟道效应。

2.62 串口线传输距离

USB 最大传输长度为 5 米。传输速度 12Mbps, 最新 USB2.0 可达 480Mbps;

RS232 最大传输距离为 15 米; 最高速率为 20kb/s

RS485、RS422 最大传输距离为 1200 米；

2.63 FIR 滤波器和 IIR 滤波器的区别

1. 两种滤波器都是数字滤波器。根据冲激响应的不同，将数字滤波器分为有限冲激响应（FIR）滤波器和无限冲激响应（IIR）滤波器。对于 FIR 滤波器，冲激响应在有限时间内衰减为零，其输出仅取决于当前和过去的输入信号值。对于 IIR 滤波器，冲激响应理论上应会无限持续，其输出不仅取决于当前和过去的输入信号值，也取决于过去的信号输出值。

2. FIR：有限脉冲响应滤波器。有限说明其脉冲响应是有限的。与 IIR 相比，它具有线性相位、容易设计的优点。这也就说明，IIR 滤波器具有相位非线性，不容易设计的缺点。而另一方面，IIR 却拥有 FIR 所不具有的缺点，那就是设计同样参数的滤波器，FIR 比 IIR 需要更多的参数。这也就说明，要增加 DSP 的计算量。DSP 需要更多的计算时间，对 DSP 的实时性有影响。

2.64 序列检测器，用两种方法

状态机

移位寄存器

2.65 IBUFG 的作用。

使得时钟延时抖动最小

IBUFG 即输入全局缓冲，是与专用全局时钟输入管脚相连接的首级全局缓冲。所有从全局时钟管脚输入的信号必须经过 IBUF 元，否则在布局布线时会报错。IBUFG 支持 AGP、CTT、GTL、GTLP、HSTL、LVCMOS、LVDCI、LVDS、LVPECL、LVTTL、PCI、PCIX 和 SSTL 等多种格式的 IO 标准。G 单 BUFG 是全局缓冲，它的输入是 IBUFG 的输出，BUFG 的输出到达 FPGA 内部的 IOB、CLB、选择性块 RAM 的时钟延迟和抖动最小。

2.66 fifo 深度计算

对于同步 fifo，每 100 个 cycle 可以写入 80 个数据，每 10 个 cycle 可以读出 8 个数据，fifo 的深度至少为？**

写时钟频率 w_clk ,

读时钟频率 r_clk ,

写时钟周期里，每 B 个时钟周期会有 A 个数据写入 FIFO

读时钟周期里，每 Y 个时钟周期会有 X 个数据读出 FIFO

则，FIFO 的最小深度是？

计算公式如下：

$$fifo_depth = burst_length - burst_length * X/Y * r_clk/w_clk$$

例举说明：

如果 100 个写时钟周期可以写入 80 个数据，10 个读时钟可以读出 8 个数据。令 $wclk=rclk$ ，考虑背靠背（20 个 clk 不发数据+80 clk 发数据+80 clk 发数据+20 个 clk 不发数据的 200 个 clk ）代入公式可计算 FIFO 的深度

$$fifo_depth = 160 - 160 \times (80\%) = 160 - 128 = 32$$

拓展：

如果令 $wclk=200\text{mhz}$ ，改为 100 个 $wclk$ 里写入 40 个， $rclk=100\text{mhz}$ ，10 个 $rclk$ 里读出 8 个。那么 $fifo$ 深度为 48

$$\text{计算如下 } fifo_depth = 80 - 80 \times (80\%) \times 100 / 200 = 80 - 32 = 48$$

2.67 在对信号采样过程之前抗混滤波，其作用是什么

它选用何种滤波器？其截止频率如何确定？

根据“奈奎斯特采样定律”：在对模拟信号进行离散化时，采样频率 f_2 至少应 2 倍于被分析的信号的最高频率 f_1 ，即： $f_2 \geq 2 f_1$ ；否则可能出现因采样频率不够高，模拟信号中的高频信号折叠到低频段，出现虚假频率成分的现象，但工程测量中采样频率不可能无限高也不需要无限高，因为我们一般只关心一定频率范围内的信号成份。

为解决频率混叠，在对模拟信号进行离散化采集前，采用低通滤波器滤除高于 $1/2$ 采样频率的频率成份。实际仪器设计中，这个低通滤波器的截止频率 f_c 为：
截止频率 (f_c) = 采样频率 (f_z) / 2.56

2.68 Verilog HDL 中哪些结构是不可综合的

(1) 所有综合工具都支持的结构: always, assign, begin, end, case, wire, tri, supply0, supply1, reg, integer, default, for, function, and, nand, or, nor, xor, xnor, buf, not, bufif0, bufif1, notif0, notif1, if, inout, input, instantiation, module, negedge, posedge, operators, output, parameter。

(2) 所有综合工具都不支持的结构: time, defparam, \$finish, fork, join, initial, delays, UDP, wait。

(3) 有些工具支持有些工具不支持的结构: casex, casez, wand, triand, wor, trior, real, disable, forever, arrays, memories, repeat, task, while。

2.69 代码覆盖率

包括语句覆盖，判定覆盖，条件覆盖，路径覆盖；

2.70 面积和速度优化

面积优化：资源共享，逻辑复制，串行化；

速度优化：流水线，关键路径法，寄存器配平；

2.71 时序检查中对异步复位电路的时序分析叫做什么

这个经指点，应该是恢复时间检查和移除时间检查。

2.72 FPGA 开发基本流程

系统规划，系统功能，功能模块划分

RTL 设计，使用 Verilog、System Verilog、VHDL 进行描述

功能仿真，理想情况下的仿真

综合、编译、布局布线，FPGA 厂商自带工具完成

时序仿真，时序分析约束

板级验证

2.73 数制转换

R 进制数转换为十进制数：按权展开，相加

十进制数转化为 R 进制数：整数部分，除 R 取余法，除到商为 0 为止。小数部分，乘 R 取整法，乘到积为 0 为止。

二进制数转化八进制数：三位一组，整数部分左边补 0，小数部分右边补 0。反之亦然。

二进制数转化十六进制数：四位一组，整数部分左边补 0，小数部分右边补 0。反之亦然。

2.74 竞争冒险

实际上如果算上逻辑门的延迟的话，那么 F 最后就会产生毛刺。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。

2.75 反相器的速度与哪些因素有关

电容（负载电容、自载电容、连线电容）较小，漏端扩散区的面积应尽可能小。

输入电容要考虑：（1） C_{gs} 随栅压而变化（2）密勒效应（3）自举效应

加大晶体管的尺寸（驱动能力），使晶体管的等效导通电阻（输出电阻）较小。

但这同时加大自载电容和负载电容（下一级晶体管的输入电容）。

提高电源电压，提高电源电压可以降低延时，即用功耗换取性能但超过一定程度后改善有限。电压过高会引起可靠性问题（氧化层击穿、热电子等）。

2.76 Transition Time (转换时间)

上升时间：从 $10\%V_{dd}$ 上升到 $90\%V_{dd}$ 的时间，下降时间 t_f 从 $90\%V_{dd}$ 下降到 $10\%V_{dd}$ 的时间。上升时间和下降时间统称为 Transition Time，也有定义为 20%到 80%。

2.77 Propagation Delay (传播延时)

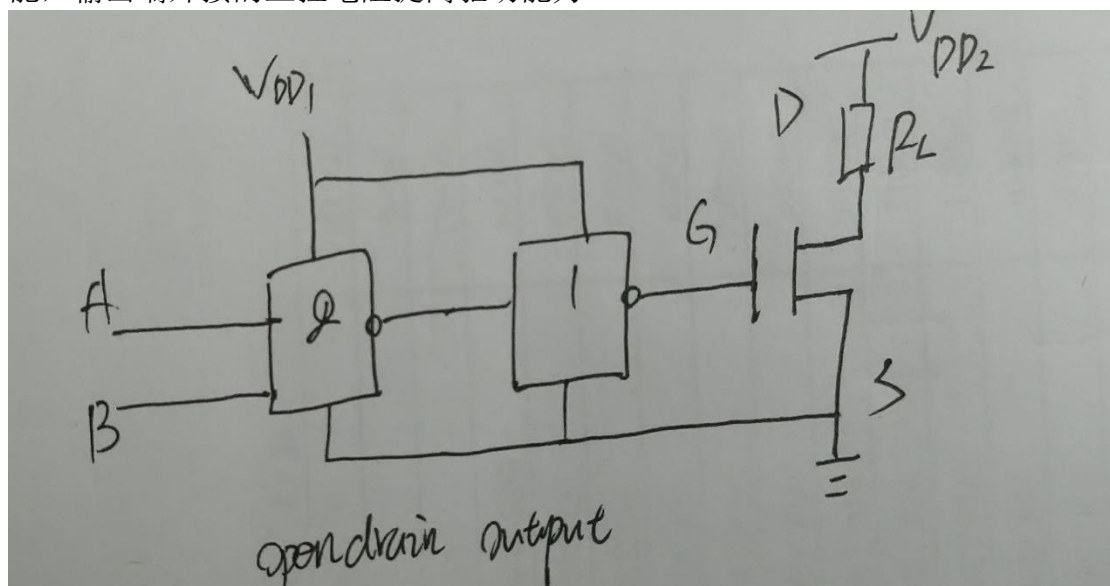
在输入信号变化到 $50\%V_{dd}$ 到输出信号变化到 $50\%V_{dd}$ 之间的时间。

2.78 什么是高阻态

高阻态：电路的一种输出状态，既不是高电平也不是低电平，如果高阻态再输入下一级电路的话，对下级电路无任何影响，可以理解为断路，不被任何东西所驱动，也不驱动任何东西

2.79 什么是 open-drain output

Open-Drain Output 漏极开路输出，称为 OD 门，两个 OD 门并联可以实现线与功能，输出端外接的上拉电阻提高驱动能力



2.80 相同面积的 cmos 与非门和或非门哪个更快

电子迁移率是空穴的 2.5 倍（在硅基 CMOS 工艺中），运算就是用这些大大小小的 MOS 管驱动后一级的负载电容，翻转速度和负载大小一级前级驱动能力相关。为了上升延迟和下降延迟相同，PMOS 需要做成 NMOS 两倍多大小。

载流子的迁移率，对 PMOS 而言，载流子是空穴；对 NMOS 而言，载流子是电子。

PMOS 采用空穴导电，NMOS 采用电子导电，由于 PMOS 的载流子的迁移率比 NMOS 的迁移率小，所以，同样尺寸条件下，PMOS 的充电时间要大于 NMOS 的充电时间长，在互补 CMOS 电路中，与非门是 PMOS 管并联，NMOS 管串联，而或非门正好

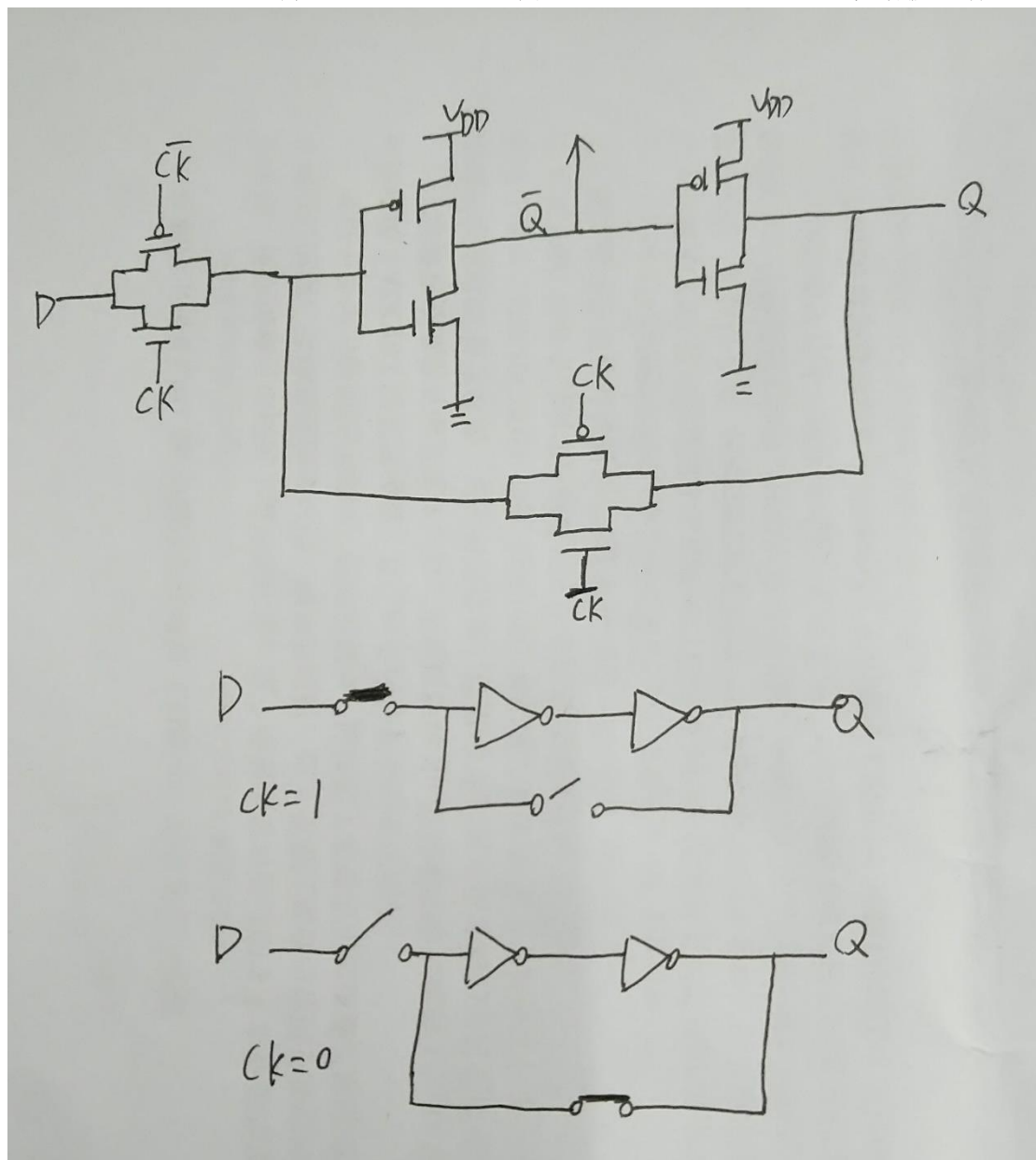
相反，所以，同样尺寸条件下，与非门的速度快，所以，在互补 CMOS 电路中，优先选择与非门。

2.81 画出 clock gating cell 的原理图。

ICG 和 CG

2.82 解释一下 Latch 的建立时间与保持时间。

在 Latch 的输入开关断开和环路开关闭合时，有效的 D 输入信号在跳变前（建立时间）和跳变后（保持时间）的短时间内不能发生变化。以保证数据被锁存。



2.83 应避免使用锁存器，尽量使用触发器的原因：

1、latch 容易产生毛刺，对输入电平敏感，受布线延迟影响大

2、在 FPGA 的资源中没有标准 latch，但有 FF，所以需要用一个逻辑门和 ff 来组成锁存器，这样就浪费了资源。

3、latch 会让时序十分复杂。没有时钟端

2.84 多时钟域设计中，如何处理跨时钟域

1、单 bit：两级触发器同步（适用于慢到快）

2、多 bit：采用异步 FIFO，异步双口 RAM

3、加握手信号

4、格雷码转换

2.85 建立时间，保持时间不满足怎么处理？

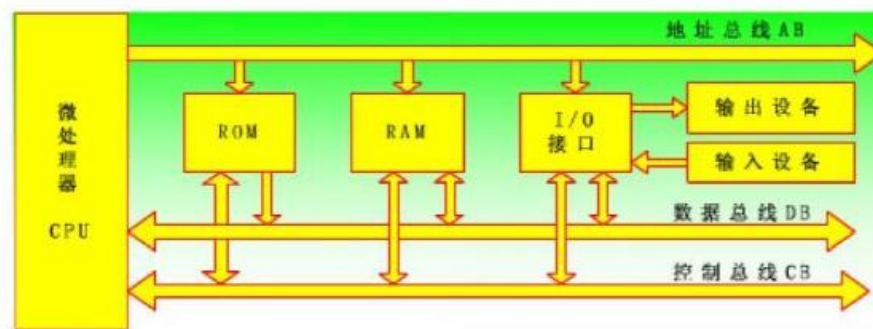
1，降低时钟频率； 2，更换更好器件

Setup time 不满足：减少组合逻辑时间，或分开做成流水线

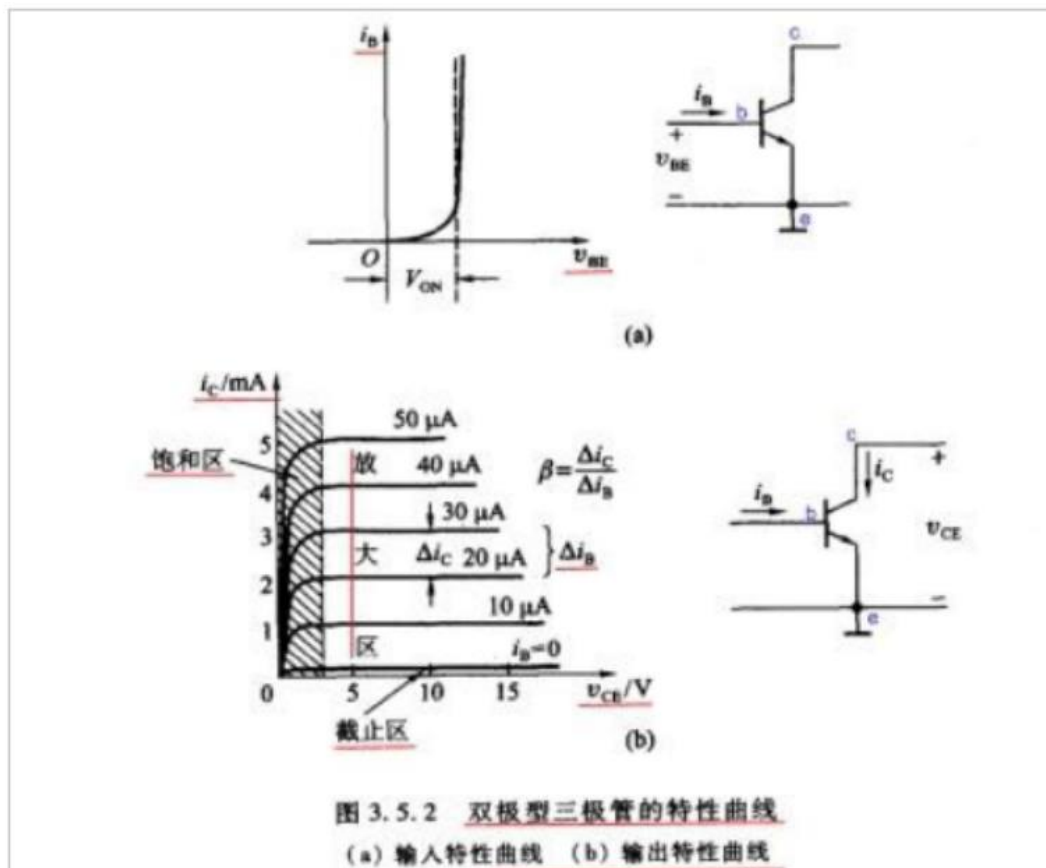
Hold time 不满足：增大组合逻辑延迟

2.86 微机接口逻辑图

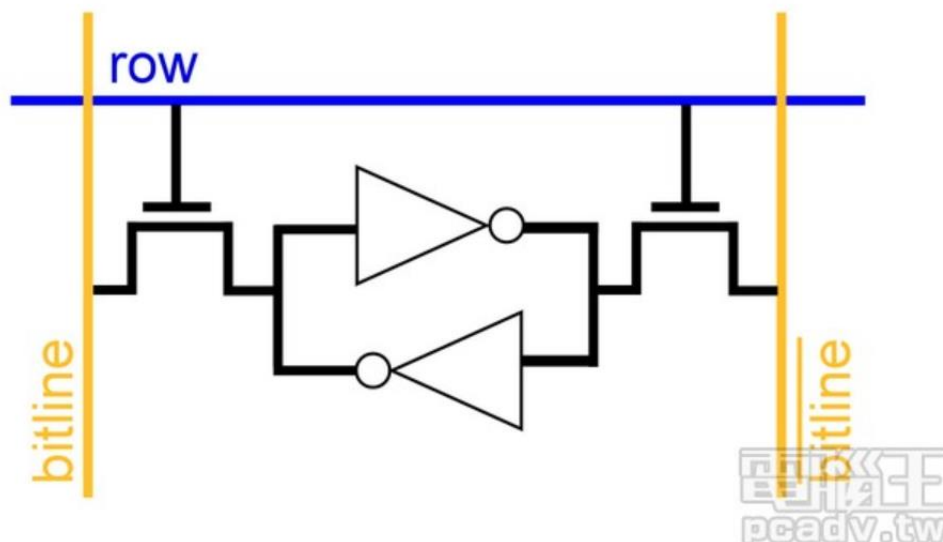
典型的输入设备与微机接口逻辑示意图(数据接口、控制接口、寄存器 / 缓冲器)。



2.87 三极管特性曲线



2.88 给出单管 DRAM 的原理图



2.89 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。

2.90 格雷码

连续的格雷码之间只有单比特信号变化，多用在异步时钟域处理上，如异步 FIFO 中地址指针的索引就采用格雷码编码。

分析：只有单比特信号跨时钟域时，我们能通过双触发器构成的同步器在另一个时钟域内得到有效脉冲，但多比特信号若采用同样的方法，会出现各个比特更新时刻不一致导致数据错误的情况。而在异步 FIFO 设计中，两侧信号属于不同时钟域，FIFO 必须通过对比来自不同时钟域的读写地址指针数值给出空满指示信号，地址指针需要多个比特信号才能代表 FIFO 深度。格雷码的单比特变化特性正好适用于这一场合，使用单比特信号同步策略完全适用于格雷码。

二进制码→格雷码（编码）：

此方法从对应的 n 位二进制码字中直接得到 n 位格雷码码字，步骤如下：

对 n 位二进制的码字，从右到左，以 0 到 n-1 编号

如果二进制码字的第 i 位和 i+1 位相同，则对应的格雷码的第 i 位为 0，否则为 1（当 i+1=n 时，二进制码字的第 n 位被认为是 0，即第 n-1 位不变）

格雷码**→二进制码（解码）：

从左边第二位起，将每位与左边一位解码后的值异或，作为该位解码后的值（最左边一位依然不变）。依次异或，直到最低位。依次异或转换后的值（二进制数）就是格雷码转换后二进制码的值。

2.91 化简代码

```
verilog
always@ (sel or a or b or c)begin
    if(sel)
        y = a + b;
    else
        y = a + c;
end

reg data_r;
always@(*)begin
    if(sel)
        data_r = b;
    else
        data_r = c;
end
assign y = a + data_r;
```

2.92 BRAM 和分布式 RAM 的区别

CLB 单元生成的 distribute ram，CLB 是 FPGA 的基本单元，blockram 也是基本

单元，但分布 ram 要由 CLB 单元生成。

分布式 ram，消耗逻辑资源

blockram 则固定在 fpga 内，无论使用与否都存在于那里。

当 block ram 不够用时可以用逻辑资源生成分布式 ram

2.93 -3dB 增益计算

dB 的两个定义方式

1. 电压（电流）放大倍数分贝数定义： $K=20\lg V_o/V_i$ ，其中 K 为放大倍数的分贝数， V_o 为放大信号输出， V_i 为信号输入；

2. 功率放大倍数分贝数定义： $K=10\lg P_o/P_i$ ，其中 K 为放大倍数的分贝数， P_o 为放大信号输出， P_i 为信号输入；

$K>0$ 说明信号被放大， $K=0$ 信号直通， $K<0$ 说明信号被衰减；

$$10\lg 2 = 3$$

关于 -3dB 带宽

-3dB 也叫半功率点或截止频率点。这时功率是正常时的一半，电压或电流是正常时的 0.707。在电声系统中， $\pm 3\text{dB}$ 的差别被认为不会影响总特性。所以各种设备指标，如频率范围，输出电平等，不加说明的话都可能有 $\pm 3\text{dB}$ 的出入。

随着输入频率上升，放大电路的电压放大倍数将下降，当电压幅度降至最大值的 0.707 倍时的位置，为截止频率。这时功率值恰好是最大功率的一半所以又称为半功率点。用分贝表示正好下降了 3dB（根据电压幅度计算： $20\lg 0.707 = -3\text{dB}$ ，根据功率计算： $10\lg (0.5) = -3\text{dB}$ ），对应的频率称为上截止频率，又常称为 -3dB 带宽。

2.94 CMOS 和 TTL 电路区别是什么

两者区别主要体现在三个方面：

a. 结构：CMOS 电路由场效应管构成，TTL 由双极性晶体管构成。

b. 电平范围：CMOS 逻辑电平范围大（5~15V），TTL 只工作在 5V 以下，因此 CMOS 噪声容限比 TTL 大，抗干扰能力强。

c. 功耗与速率：CMOS 的功耗比 TTL 小，但工作频率低于 TTL。

2.95 FPGA 型号介绍

举例：EP1C6Q240C8

EP 表示 ALTERA

1c 是 cyclone 1, 是一个系列

6 代表 6000 个逻辑单元（K LES）

q 是 QFP 封装 还有 F

240 是引脚

c 为商业级 I 工业级
8 是速度等级

XC7VX485T-2FFG1761C

XC 表示 XILLINX 公司的产品

7VX 表示 V7 系列的芯片，

485T 表示其有 48.5 万个逻辑单元。

-2 表示速度等级，对于 Xilinx FPGA 来说，一般有-1，-2，-3 三个等级，值越大，速度越高。

FFG 表示封装方式

1761 表示引脚数

C 代表的是温度等级 Temperature grade，这里是商用 Commercial，如果是 I 就是工业用。

2.96 FPGA 的面积优化和速度优化有哪几类

面积优化：资源共享，逻辑复制，串行化

速度优化：流水线，关键路径法，寄存器配平

2.97 FPGA 中属于高速串行接口的是

PCIe、USB、SPI、Rapid IO

2.98 PCI 总线特点

1 传输速率高 最大数据传输率为 132MB/s，当数据宽度升级到 64 位，数据传输率可达 264MB/s。这是

其他总线难以比拟的。它大大缓解了数据 I/O 瓶颈，使高性能 CPU 的功能得以充分发挥，适应高速设备数

据传输的需要。

2 多总线共存采用 PCI 总线可在一个系统中让多种总线共存，容纳不同速度的设备一起工作。通过

HOST-PCI 桥接组件芯片，使 CPU 总线和 PCI 总线桥接；通过 PCI-ISA/EISA 桥接组件芯片，将 PCI 总线

与 ISA/EISA 总线桥接，构成一个分层次的多总线系统。高速设备从 ISA/EISA 总线卸下来，移到 PCI 总线

上，低速设备仍可挂在 ISA/EISA 总线上，继承原有资源，扩大了系统的兼容性。

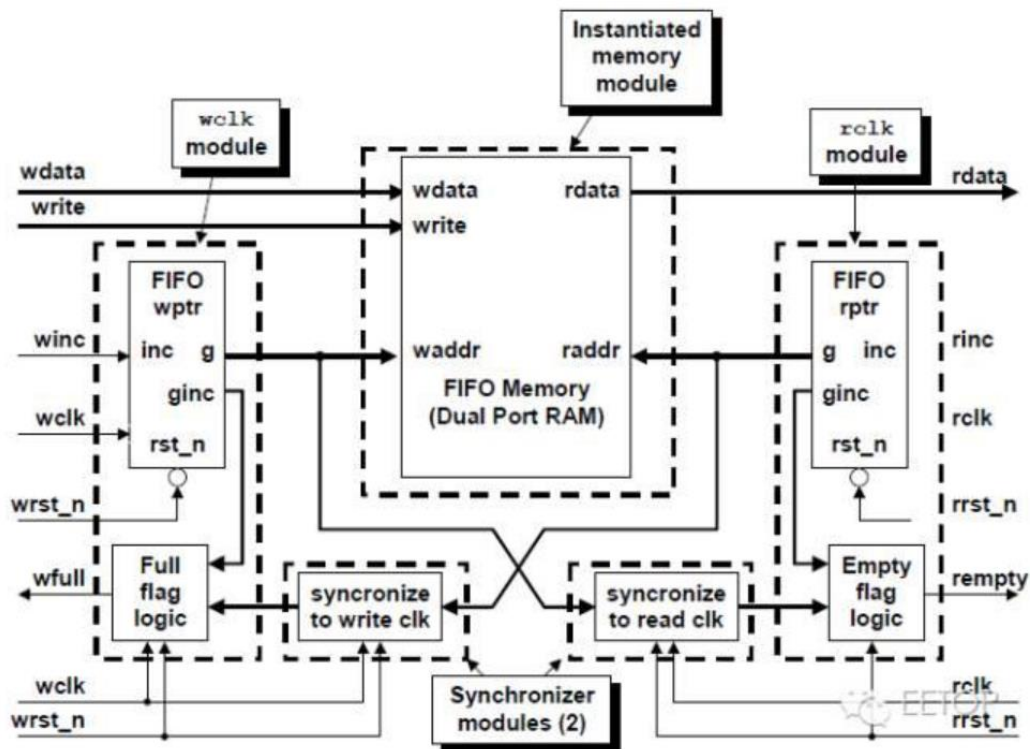
3 独立于 CPU PCI 总线不依附于某一具体处理器，即 PCI 总线支持多种处理器及将来发展的新处理器，

在更改处理器品种时，更换相应的桥接组件即可。

4 自动识别与配置外设，用户使用方便。

5 并行操作能力。

2.99 画出异步 FIFO 架构



2.100 电路频率计算

假设存在 positive clock skew 为 10ns, 问最高电路频率。

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

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

$$clk_time = T_{co} + T_{delay} + T_{setup} + T_{skew_nege}$$

2.101 RS232c 高电平脉冲对应的 TTL 逻辑是

负逻辑

首先解释一下什么是正逻辑和负逻辑。正逻辑：用高电平表示逻辑 1，用低电平表示逻辑 0。负逻辑：用低电平表示逻辑 1，用高电平表示逻辑 0。在数字系统的逻辑设计中，若采用 NPN 晶体管和 NMOS 管，电源电压是正值，一般采用正逻辑。若采用的是 PNP 管和 PMOS 管，电源电压为负值，则采用负逻辑比较方便。除非特别说明，一般电路都是采用正逻辑

对于 RS232C 的数据线，逻辑 1 (MARK) = -3V ~ -15V；逻辑 0 (SPACE) = +3 ~ +15V，因此对应的 TTL 逻辑为负逻辑

2.102 串行通信与同步通信异同

串行通信是指 使用一条数据线，将数据一位一位地依次传输，每一位数据占据一个固定的时间长度。其只需要少数几条线就可以在系统间交换信息，特别适用于计算机与计算机、计算机与外设之间的远距离通信。在计算机和终端之间的数据传输通常是靠电缆或信道上的电流或电压变化实现的。

并行通信，一组数据的各数据位在多条线上同时被传输。

同步通信与异步通信区别：

1. 同步通信要求接收端的时钟频率和发送端时钟频率一致，发送端发送连续的比特流；异步通信时不要求接收端时钟和发送端时钟同步，发送端发送完一个字节后，可经过任意长的时间间隔再发送下一个字节。
2. 同步通信效率高；异步通信效率较低。
3. 同步通信较复杂，双方时钟的允许误差较小；异步通信简单，双方时钟可允许一定误差。
4. 同步通信可用于点对多点；异步通信只适用于点对点

2.103 机器周期和时钟周期

8051 系列单片机的一个机器周期由 6 个 S 周期（状态周期）组成。也就是说一个机器周期 = 6 个状态周期 = 12 个振荡周期（即时钟周期）

2.104 施密特电路回差电压

回差电压属于施密特触发器的参数。

当输入信号电压上升到 U_K 时，施密特触发器状态会从第 1 种状态 A 转变为第 2 种状态 B；当输入信号电压下降到 U_T 时，它又会从第 2 种状态 B 翻转到第 1 种状态 A。输入信号两次触发电压是存在差距的，这种情况称为回差现象，这两个电压的差值称为回差电压 ΔU 。

image-20200517130638377

image-20200517130638377

通常将 U_K 称为施密特触发器的上门限电压， U_T 则称为下门限电压。

详解待整理

2.105 VCO 压控振荡器

压控振荡器指输出频率与输入控制电压有对应关系的振荡电路 VCO，频率是输入信号电压的函数的振荡器 VCO，振荡器的工作状态或振荡回路的元件参数受输入控制电压的控制，就可构成一个压控振荡器。

参数：频率调谐范围，输出功率，长期及短期频率稳定度，相位噪声，频谱纯度，

电调速度，推频系数，频率牵引等。

2.106 什么是状态图

状态转换表、状态转换图和时序图是时序电路特有的描述方法，描述出电路在 CP 作用下，状态转换全部过程。

2.107 什么奈奎斯特定律

奈奎斯特证明，对于一个带宽为 W 赫兹的理想信道，其最大码元（信号）速率为 $2W$ 波特。这一限制是由于存在码间干扰。如果被传输的信号包含了 M 个状态值（信号的状态数是 M ），那么 W 赫兹信道所能承载的最大数据传输速率（信道容量）是：

$$C = 2 \times W \times \log_2 M \text{ (bps)}$$

奈奎斯特采样定理

奈奎斯特采样定理解释了采样率和所测信号频率之间的关系。

阐述了采样率 f_s 必须大于被测信号感兴趣最高频率分量的两倍。该频率通常被称为奈奎斯特频率 f_N 。

2.108 模拟信号转为数字信号

采样、量化、编码

2.109 D 触发器和 D 锁存器的区别

触发器对时钟脉冲边沿（上升或下降）敏感，在边沿来临时变化状态；锁存器对时钟脉冲电平（持续时间）敏感，在一持续电平期间都运作。

2.110 拉氏变换和傅立叶变换的表达式及联系

拉普拉斯变换是傅里叶变换的推广，是一种更普遍的表达形式。在进行信号与系统的分析过程中，可以先得到拉普拉斯变换度这种更普遍的结果，然后再得到傅里叶变换这种特殊的结果。

2.111 平板电容公式 $C = \epsilon S / 4 \pi k d$

2.112 放大电路的频率补偿的目的是什么，有哪些方法

一是改善放大电路的高频特性，二是克服由于引入负反馈而可能出现自激振荡现象，使放大器能够稳定工作。在放大电路中，由于晶体管结电容的存在常常会使放大电路频率响应的高频段不理想，为了解决这一问题，常用的方法就是在电路中引入负反馈。然后，负反馈的引入又引入了新的问题，那就是负反馈电路会出现自激振荡现象，所以为了使放大电路能够正常稳定工作，必须对放大电路进行频率补偿。

频率补偿的方法可以分为超前补偿和滞后补偿，

主要是通过接入一些阻容元件来改变放大电路的开环增益在高频段的相频特性，目前使用最多的就是锁相环。

2.113 频率响应

如：怎么才算是稳定的，如何改变频响曲线的几个方法。

幅度特性是常数，相位特性是过原点的直线。好像是这样

频率响应通常亦称频率特性，频率响应或频率特性是衡量放大电路对不同频率输入信号适应能力的一项技术指标。实质上，频率响应就是指放大器的增益与频率的关系。通常讲一个好的放大器，不但要有足够的放大倍数，而且要有良好的保真性能，即：放大器的非线性失真要小，放大器的频率响应要好。“好”：指放大器对不同频率的信号要有同等的放大。之所以放大器具有频率响应问题，原因有二：一是实际放大的信号频率不是单一的；二是放大器具有电抗元件和电抗因素。由于放大电路中存在电抗元件（如管子的极间电容，电路的负载电容、分布电容、耦合电容、射极旁路电容等），使得放大器可能对不同频率信号分量的放大倍数和相移不同。如放大电路对不同频率信号的幅值放大不同，就会引起幅度失真；如放大电路对不同频率信号产生的相移不同就会引起相位失真。幅度失真和相位失真总称为频率失真，由于此失真是由电路的线性电抗元件（电阻、电容、电感等）引起的，故不称为线性失真。为实现信号不失真放大所以要需研究放大器的频率响应。

答案

由于放大器件本身具有极间电容，以及放大电路中有时存在电抗性元件，所以，当输入不同频率信号时，电路的放大倍数将成 为频率的函数，这个特性就是频率特性或者频率响应。分为幅频特性和相频特性。

改变频响曲线就是改变其幅度和相位响应，可以通过外加 RC，LC 网络来改变其幅频特性和相频特性

2.114 给出一个查分运放，如何相位补偿，并画补偿后的波特图。

随着工作频率的升高，放大器会产生附加相移，可能使负反馈变成正反馈而引起自激。进行相位补偿可以消除高频自激。相位补偿的原理是：在具有高放大倍数的中间级，利用一小电容 C （几十~几百微微法）构成电压并联负反馈 电路。可以使用电容校正、RC 校正分别对相频特性和幅频特性进行修改。

波特图就是在画放大电路的频率特性曲线时使用对数坐标。波特图由对数幅 频特性和对数相频特性两部分组成，它们的横轴采用对数刻度 $\lg f$ ，幅频特性的纵轴采用 $\lg |A_u|$ 表示，单位为 dB；相频特性的纵轴仍用 ϕ 表示。

2.115 基本放大电路种类

（电压放大器，电流放大器，互导放大器和互阻放大器），优缺点，特别是广泛采用差分结构的原因。

优缺点，特别是广泛采用差分结构的原因。答：放大电路的作用：放大电路是电子技术中广泛使用的电路之一，其作用是将微弱的输入信号（电压、电流、功率）不失真地放大到负载所需要的数值。放大电路种类：

（1）电压放大器：输入信号很小，要求获得不失真的较大的输出压，也称小信号放大器；

（2）功率放大器：输入信号较大，要求放大器输出足够的功率，也称大信号放大器。

差分电路是具有这样一种功能的电路。该电路的输入端是两个信号的输入，这两个信号的差值，为电路有效输入信号，电路的输出是对这两个输入信号之差的放大。设想这样一种情景，如果存在共模干扰信号，会对两个输入信号产生相同的干扰，通过二者之差，干扰信号的有效输入为零，这就达到了抗共模干扰的目的

2.116 差分信号求共模分量和差模分量

给出一差分电路，告诉其输出电压 V_+ 和 V_- ，求共模分量和差模分量。（未知）

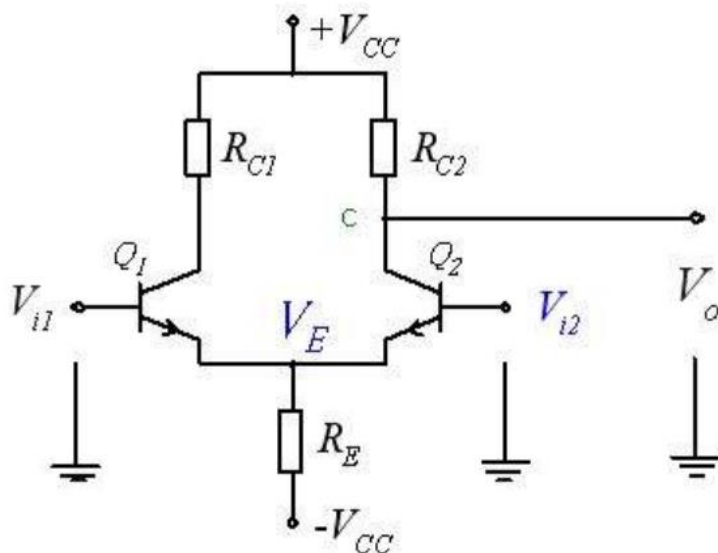
共模信号： $V_{com} = (V_1 + V_2) / 2$

差模信号：

对于 V_1 ： $V_1 - (V_1 + V_2) / 2 = (v_1 - v_2) / 2$

对于 V_2 ： $V_2 - (v_1 + v_2) / 2 = - (v_1 - v_2) / 2$

2.117 画差放的两个输入管



2.118 画出由运放构成的基本电路

构成加法、减法、微分、积分运算的电路原理图。并画出一个晶体管级的运放电路。（仕兰微电子）

注：应该是负反馈电路

加法:

减法:

微分:

积分:

2.119 用运算放大器组成一个 10 倍的放大器

放大器

2.120 分析积分电路

给出一个简单电路，让你分析输出电压的特性（就是个积分电路），并求输出端某点的 rise/fall 时间。 Infineon 笔试试题

118、高通滤波器与低通滤波器

电阻 R 和电容 C 串联，输入电压为 R 和 C 之间的电压，输出电压分别为 C 上电压和 R 上电压，要求制这两种电路输入电压的频谱，判断这两种电路何为高通滤波器，何为低通滤波器。当 $RC \ll \text{period} - \text{setup} - \text{hold}$

2.121 建立时间保持时间计算

时钟周期为 T ，触发器 $D1$ 的建立时间最大为 $T1_{\max}$ ，最小为 $T1_{\min}$ 。组合逻辑电路最大延迟为 $T2_{\max}$ ，最小为 $T2_{\min}$ 。问，触发器 $D2$ 的建立时间 $T3$ 和保持时间应满足什么条件。（华为）

2.122 最大时钟频率计算

给出某个一般时序电路的图，有 T_{setup} , T_{delay} , $T_{\text{ck} \rightarrow \text{q}}$ ，还有 clock 的 delay，写出决定最大时钟的因素，同时给出表达式。（威盛 VIA 2003.11.06 上海笔试试题）

2.123 触发器的种类及优缺点

有几种（区别，优点）

D 触发器

JK 触发器

T 触发器

2.124 latch 与 register 的区别

为什么现在多用 register. 行为级描述中 latch 如何产生的

<https://www.cnblogs.com/xuzhi-fpga/p/3522454.html> 参考

为什么现在多用 register. 行为级描述中 latch 如何产生的行为级描述中 latch

一般是由于 if 或 case 逻辑表述不完全产生的。

异同：

1、 latch 由电平触发，非同步控制。在使能信号有效时 latch 相当于通路，在使能信号无效时 latch 保持输出状态。DFF 由时钟沿触发，同步控制。

2、 latch 容易产生毛刺 (glitch)，DFF 则不易产生毛刺。

3、 如果使用门电路来搭建 latch 和 DFF，则 latch 消耗的门资源比 DFF 要少，这是 latch 比 DFF 优越的地方。所以，在 ASIC 中使用 latch 的集成度比 DFF 高，但在 FPGA 中正好相反，因为 FPGA 中没有标准的 latch 单元，但有 DFF 单元，一个 LATCH 需要多个 LE 才能实现。

4、 latch 将静态时序分析变得极为复杂。

一般的设计规则是：在绝大多数设计中避免产生 latch。它会让您设计的时序完蛋，并且它的隐蔽性很强，非老手不能查出。latch 最大的危害在于不能过滤毛刺。这对于下一级电路是极其危险的。所以，只要能用 D 触发器的地方，就不用 latch。有些地方没有时钟，也只能用 latch 了。比如现在用一个 clk 接到 latch 的使能端假设是高电平使能，这样需要的 setup 时间，就是数据在时钟的下降沿之前需要的时间，但是如果是一个 DFF，那么 setup 时间就是在时钟的上升沿需要的时间。这就说明如果数据晚于控制信号的情况下，只能用 latch，这种情况就是，前面所提到的 latch timing borrow。基本上相当于借了一个高电平时间。也就是说，latch 借的时间也是有限的。

2.125 阻塞赋值和非阻塞赋值的区别

阻塞赋值：前面语句执行完，才可执行下一条语句；即：前面语句的执行 (b=a) 阻塞了后面语句的执行 (c=b)。即：always 块内，2 条语句顺序执行。

```
always @(posedge i_clk)
```

```
begin
```

```
    b = a;
```

```
    c = b;
```

```
end
```

注意：a 的值赋给 b，b 的值赋给 c，是在同一个时钟上升沿完成，如下图仿真结果

非阻塞赋值：always 块内，2 条语句同时执行。即：前面语句的执行 (b=a) 不会阻塞后面语句的执行 (c=b)。

```
always @(posedge i_clk)
```

```
begin
```

```
    b <= a;
```

```
    c <= b;
```

```
end
```

注意：第 1 个 clk 上升沿 a 的值赋给 b，此时 b 的值还没有更新；第 2 个 clk 上升沿，b 的值赋给 c，此时 c 才能获得 b 更新后的值。所以 c 获得 a 的值，需要 2 个 clk 完成。如下图仿真结果。

2.126 产品设计流程

现有一用户需要一种集成电路产品，要求该产品能够实现如下功能： $y=\ln x$ ，其中， x 为

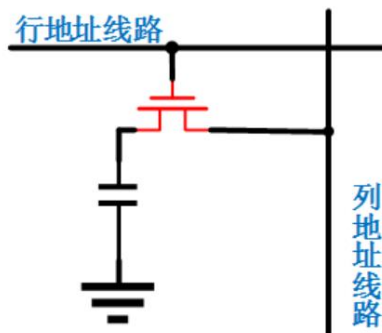
4 位二进制整数输入信号。 y 为二进制小数输出，要求保留两位小数。

电源电压为 3~5v 假设公司接到该项目后，交由你来负责该产品的设计，试讨论该产品的设计全程。（仕兰微电子）

2.127 给出单管 DRAM 原理

问你有什么办法提高 refresh time，总共有 5 个问题，记不起来了。（降低温度，增大电容存储容量，

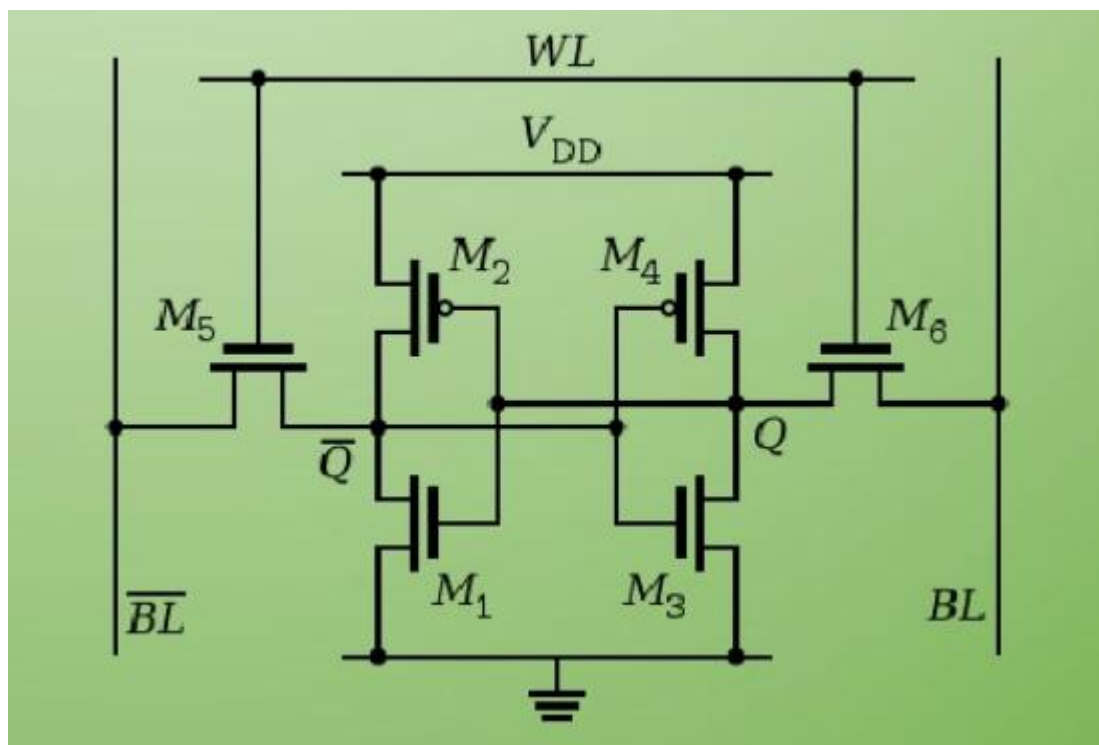
<https://blog.csdn.net/chenming092100/article/details/78562730>



2.128 SRAM 读写原理

Please draw schematic of a common SRAM cell with 6 transistors, point out which nodes can store data and which node is word line control?

详情查看: <https://wenku.baidu.com/view/4938078b16fc700abb68fc89.html>



共需要 6 个 mos 管，其中两个反相器（4 个 mos 管），两个选通 mos 管，用来读取写入数据；
通过选通 mos 管，来对数据进行改写，有比逻辑，通过选通管上拉或者下拉数据

2.129 FPGA 与 ASIC 设计流程的区别

首先我们要清楚 FPGA 的基本单元是 LUT，而 ASIC 的基本单元是寄存器；

对于 FPGA：首先要进行模块划分，然后编写 RTL 代码，进行 RTL 仿真，综合后将生成的网表文件在 FPGA 上实现，也就是布局布线，该步骤完成后进行 STA，没有问题的话就生成 bit 流文件，将 bit 流文件烧录到 FPGA 开发板上进行板级调试；

对于 ASIC：模块划分、RTL 设计、验证、综合、STA、形式验证、布局规划、DRC 设计规则检查、时钟树综合、生成 GDSII 文件；

版权声明：本文为 CSDN 博主「bleaucht」的原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接及本声明。

原文链接：<https://blog.csdn.net/bleaucht/article/details/98314001>

2.130 跨时钟域

image-20200530170907791

image-20200530170907791

2.131 CMOS 反相器的功耗包括几部分？分别与哪些因素相关？

$P_{dynamic}$ 是电路翻转产生的动态功耗

P_{short} 是 P 管和 N 管同时导通时产生的短路功耗

$P_{leakage}$ 是由扩散区和衬底之间的反向偏置漏电流引起的静态功耗

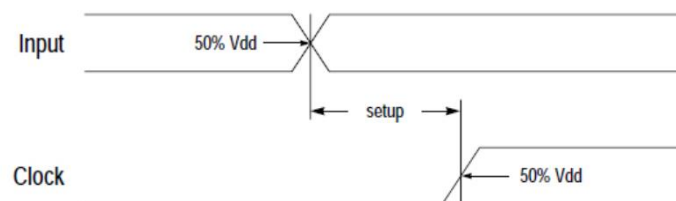
静态功耗：CMOS 反相器在静态时，P、N 管只有一个导通。由于没有 Vdd 到 GND 的直流通路，所以 CMOS 的静态功耗应该等于零。但实际上，由于扩散区和衬底的 PN 结上存在反向漏电流，所以会产生静态功耗。

短路功耗：CMOS 电路在“0”和“1”的转换过程中，P、N 管会同时导通，产生一个由 Vdd 到 VSS 窄脉冲电流，由此引起功耗

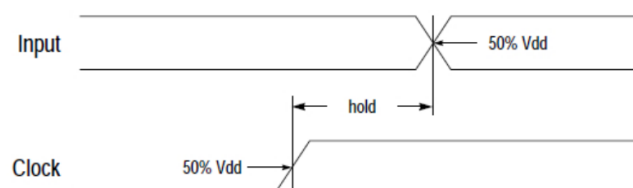
动态功耗： C_L 这个 CMOS 反相器的输出负载电容，由 NMOS 和 PMOS 晶体管的漏扩散电容、连线电容和扇出门的输入电容组成。

2.132 建立时间、保持时间、传播延迟、转换时间、恢复时间、移除时间、最小脉冲宽度

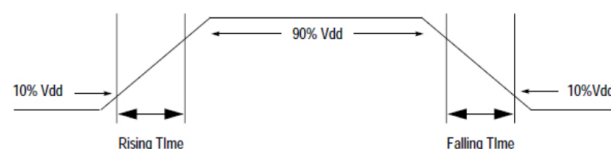
建立时间：在时钟沿来临前，输入信号的变化超过 50%Vdd 的时间到时钟变化超过 50%Vdd 的时间中，输入信号保持稳定的最小时间。



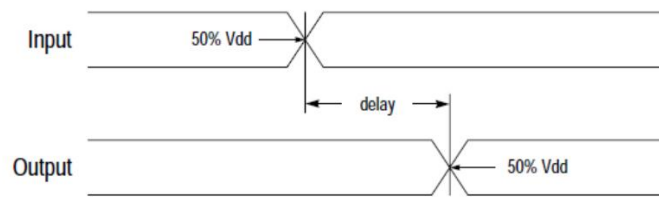
保持时间：在时钟沿来临后，输入信号的变化超过 50%Vdd 的时间到时钟变化超过 50%Vdd 的时间中，输入信号保持稳定的最小时间。



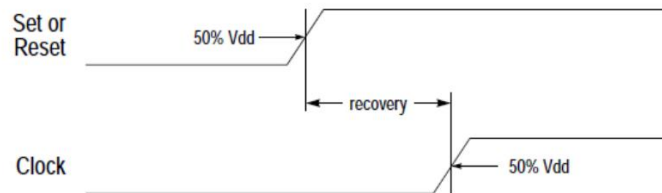
转换时间：输入和输出信号，上升时间：从 10%Vdd 上升到 90%Vdd 的时间，下降时间 t_f 从 90%Vdd 下降到 10%Vdd 的时间。上升时间和下降时间统称为 Transition Time。



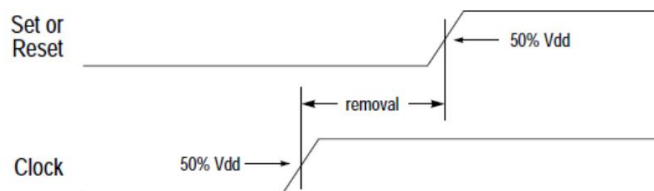
传输延迟：在输入信号变化到超过 50%Vdd 到输出信号变化到超过 50%Vdd 之间的时间。



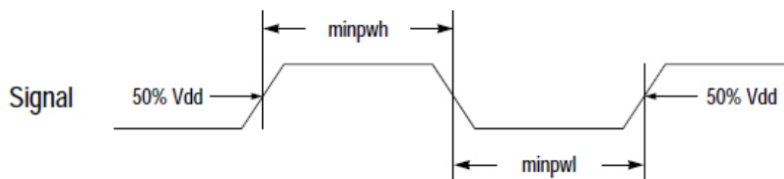
恢复时间：复位或者置位信号变化超过 50%Vdd 的时间到时钟变化超过 50%Vdd 的时间中，时钟沿来临的前最小时间，保证复位或置位完成。



移除时间：复位或者置位信号变化超过 50%Vdd 的时间到时钟变化超过 50%Vdd 的时间中，时钟沿来临的后最小时间，保证置位或复位完成。



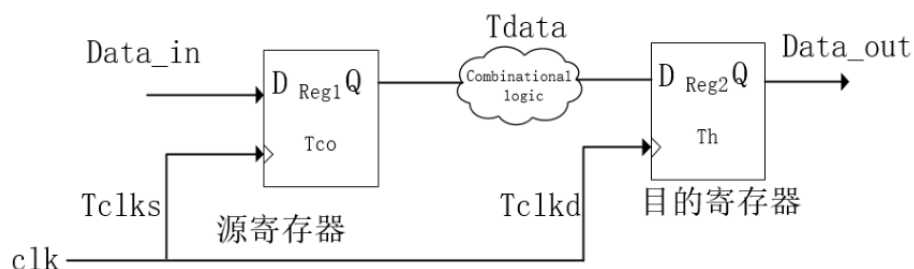
最小脉冲宽度：最小脉冲宽度就是信号上升沿变化超过 50%Vdd 到下降沿变化低于 50%Vdd 时，测量高电平的最小脉冲宽度，低电平最小宽度同理。



2. 133 ASIC 中低功耗的设计方法和思路（不适用于 FPGA）

- 1、合理规划芯片的工作模式，通过功耗管理模块控制芯片各模块的 Clock，Reset 起到控制功耗的目的。
- 2、门控时钟（Clockgating）：有效降低动态功耗
- 3、多电压供电：通过控制模块的电压来降低功耗
- 4、多阈值电压

2. 134 最小周期计算



Tco: 寄存器更新延迟。clock output delay, 时钟触发到数据输出的最大延迟时间

最小时钟周期: $T_{min} = T_{co} + T_{data} + T_{su} - T_{skew}$ 。最快频率 $F_{max} = 1/T_{min}$

$T_{skew} = T_{clkd} - T_{clks}$ 。

2.135 reg 和 wire 的区别

reg 是寄存器类型可以存储数据, wire 是线网型

reg 型在 always 块和 initial 块中赋值, wire 型用 assign 赋值

reg 型可用于时序逻辑和组合逻辑赋值, wire 型只能用于组合逻辑赋值

wire 表示直通, 即只要输入有变化, 输出马上出现结果, reg 表示一定要有触发, 输出才会反映输入

2.136 localparam、parameter 和 define 的区别

声明:

localparam xx = yy;

parameter xx = yy;

`define XX YY 使用: xx `XX

localparam 只能在当前 Verilog 文件中使用

parameter 与 define 都可以用来定义常量

parameter 写在模块中, 可以被上一层模块调用时, 进行参数传递。

parameter 作用于声明的那个文件; define 从编译器读到这条指令开始到编译结束都有效, 或者遇到

`undef 命令使之失效。

2.137 task 与 function 的区别

<https://blog.csdn.net/kobesdu/article/details/39080571>

2.138 谈谈对 Retiming 技术的理解

Retiming 就是重新调整时序, 例如电路中遇到复杂的组合逻辑, 延迟过大, 电路时序不满足, 这个

时候采用流水线技术, 在组合逻辑中插入寄存器加流水线, 进行操作, 面积换速度思想。

2.139 验证覆盖率

目标上可以分为: 代码覆盖率和功能覆盖率

代码覆盖率可以分为: 行覆盖率、分支覆盖率、路径覆盖率、条件覆盖率、翻转覆盖率、状态机覆盖率

功能覆盖率可分为: 基于控制的覆盖率、基于数据的覆盖率

2.140 下列功耗措施哪个可以降低峰值功耗（D）

- A 静态模块级 Clock Gating
- B Memory Shut Down
- C Power Gating
- D 大幅度提高 HVT 比例

解析：首先峰值功耗是属于动态功耗中的短路功耗，即 NMOS 和 PMOS 同时导通所引起的峰值电流，最终带来的功耗。这个功耗和电源电压，时钟翻转率，以及峰值电流有关。

A 选项静态门控时钟，所以 A 选项不正确。

B 选项存储关闭。即不被访问的时候，关闭存储器，因而也是降低静态功耗。

C 选项电源门控技术，即模块不工作的时候，关闭电源，模块睡眠，工作时候再启动电源，是降低静态功耗。

D 选项即采用高阈值电压的晶体管，阈值电压增加的效果在于降低亚阈值漏电流，并且短路功耗公式为： $P_{short} = \tau A \beta (V_{dd} - V_{th})^3$ ，只跟 V_{th} 有关，而 D 选项中大幅提高 HVT（High Voltage Value）带入短路功耗公式中会使短路功耗变小，从而降低动态功耗中的峰值功耗。所以 D 选项正确。

2.141 跨时钟域处理

假设一个 3bit 计数器（计数范围 0-6），工作在 38M 时钟域下，要把此计数器的值传递到另一个异步 100M 时钟域，以下不正确的是

- A 使用异步 FIFO
- B 锁存+握手信号
- C 使用格雷码
- D 使用 DMUX 电路

觉得光有格雷码不行，需要打拍，所以选 C；至于 DMUX 有下图解

2.142 逻辑电路低功耗设计中，无效方法是（A）

- A 采用慢速设计
- B 减少信号翻转
- C 采用较慢速的时钟
- D 提高阈值电压

2.143 噪声容限

高电平噪声容限 = 最小输出高电平电压-最小输入高电平电压

低电平噪声容限 = 最大输入低电平电压-最大输出低电平电压

噪声容限 = min{高电平噪声容限，低电平噪声容限}

2.144 FALSE PATH

FALSE PATH 就是我们在进行时序分析时，不希望工具进行分析的那些路径。一般不需要工具时序分析的路径指的是异步的路径，异步路径就是指的不同时钟域的路径。

在 QuartusII 的一个培训文档里面解释了什么时候要用到 FALSE PATH:

1. 从逻辑上考虑，与电路正常工作不相关的那些路径，比如测试逻辑，静态或准静态逻辑。
2. 从时序上考虑，我们在综合时不需要分析的那些路径，比如跨越异步时钟域的路径。

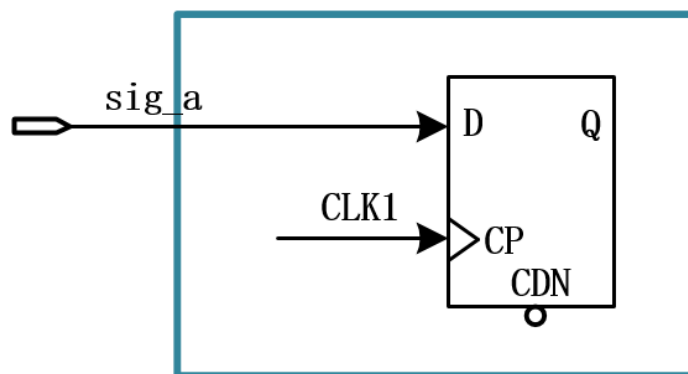
2.145 逻辑分析仪

14、以下哪种异常可以用逻辑分析仪测试(C)

- A 信号间串扰过大 B 6 个信号的异常组合
C 信号占空比超标 D 信号上升沿缓慢

2.146 时序约束

如下图所示，当 CLK1 为 100MHz 时，约束 `set_input_delay -clocks CLK1 -max 5 sig_a`，如果 CLK1 变成 50MHz，需要保证的约束效果不变，此时 `input_delay` 应该设置为多少？(A)



- A 15ns B 20ns C 5ns D 10ns

2.147 5 级指令流水线

5 级流水线操作分别是取指、译码、执指、缓冲和写回。

2.148 FPGA 时序优化方法

1. 插寄存器，缩短组合逻辑的延时
2. 寄存器复制
3. 加两级寄存器，消除亚稳态，异步信号的同步处理

- 4.进行时序约束
- 5.进行 logic lock 物理约束
- 6.采用 multicycle 设计
- 7.FIFO

并行化设计
均衡设计
优化路径

2. 149 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。

2. 150 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

2. 151 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

2. 152 C 语言函数返回值

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；

2. 153 验证的主要目的是什么？

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

2. 154 C 语言中，实型常数是按(A)处理。

A double B long double C 其他都不对 D float

2. 155 FPGA 器件的结温范围

商用级的芯片可以正常工作的结温范围为 0~85 摄氏度，
而工业级芯片的范围是-40~100 摄氏度

2. 156 常用的工频陷波电路属于哪种类型的滤波电路(C)。

A HPF B BPF C BEF D LPF

HPF: 高通滤波器

LPF: 低通滤波器

BPF: 带通滤波器

BEF: 带阻滤波器

2. 156 高速信号

高速信号与高频信号没有关系

信号是否为高速信号，取决于两个因素：信号频率和信号传输路径长度（信号线长度）。

信号频率，指的并非是信号的周期频率，而是信号的有效频率，所谓信号的有效频率(Fr)，指的是由信号的上升下降时间所决定的频率，若假设某信号上升下降时间相同，则该信号的有效频率为

$$Fr = 0.5 / Tr(10\%-90\%)$$

其中，Tr(10%-90%)为由信号的 10%到 90%所需时间。

通常认为，信号线长度小于信号有效波长的 1/6 为低速信号，大于信号有效波长的 1/6 为高速信号，而信号有效波长为

$$Lb = c / Fr$$

其中，c 为常数，可认为 c=光速；Fr 为信号有效频率。

因此，区分某信号为高速信号还是低速信号的步骤是：

- (1) 获取信号有效频率 Fr
- (2) 计算信号有效波长 Lb
- (3) 获取实际信号线长度 L
- (4) 若 $L < 1/6 Lb$ ，为低速信号，若 $L > 1/6 Lb$ ，为高速信号

2. 157 传输延迟和扇出扇入的关系

传输延迟将随扇出的增大而增大，随扇入的增大而增大

2. 158 OOP 的三大特征

面向对象编程的三大特性：封装性、继承性、多态性

2. 159 NAND 和 NOR flash 的区别

- 1: NOR 读取速度快、NAND 写入速度快
- 2: NAND 容量大，尺寸小
- 3: NAND flash 通常存在坏块

2. 160 移位计数器

题目：移位寄存器由 8 级触发器构成，则构成的扭环计数器有多少个有效状态？环形计数器？线性反馈移位寄存器？

扭环计数器的实现是设置一个初始状态，将最高位取反，作为最低位的输入，通过移位即可得到。也是基于移位寄存器的计数器，是对环形计数器的改进，对于 n 个移位寄存器构成的计数器，有 $2n$ 个有效状态
 环形也是基于移位寄存器的计数器，对于 n 个移位寄存器构成的计数器，只有 n 个有效状态。

设置一个初始状态，通过移位即可得到。

线性反馈移位寄存器有 $2^n - 1$

https://blog.csdn.net/qq_44113393/article/details/89852994

2. 161 buck 降压电路中，二极管、电容和电感作用

电感是储能、降压的作用；

电容是平稳电压。

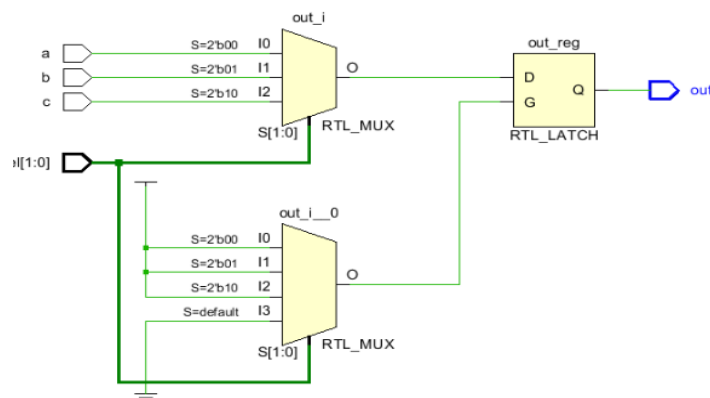
二极管是保护器件不被感应电压烧坏，以及环路导通

三极管提供控制作用

简单的说，开关管导通，电感充电，电容提供电压，开关管截止，电感放电，电容稳定电压。

2. 162 锁存器生成

case 条件没列全生成锁存器的电路如何画？



先确定必定产生锁存器

然后还是case

那就用四选—多路选择器

列了两种情况，那就肯定有两种情况到不了，那就保持上一个原值呗，如何保持呢？再多用一个多路选择器

2. 163 CMOS 动态功耗计算

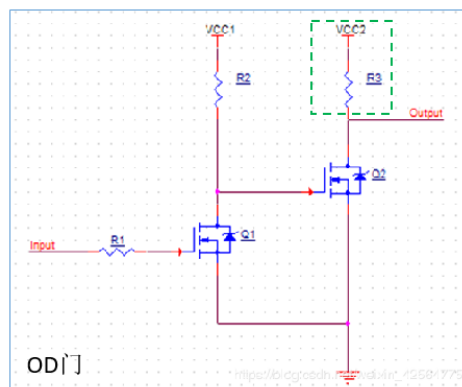
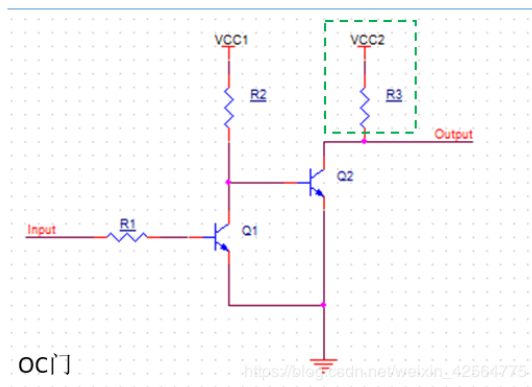
$$P = (CL * VDD^2) / 2$$

2. 164 OD 门和 OC 门

OC (Open Collector) 门，又称集电极开路，一般为 TTL 电路。

OD (Open Drain) 门，又称漏极开路，一般为 MOS 电路。

线与逻辑：即两个输出端（包括两个以上）直接互连就可以实现“AND”的逻辑功能。



注虚线框中的电阻是额外添加的，不在OC/OD门当中，输出要么为0，要么为高阻态

线与逻辑指多个输出连接到一起，实现与的功能，实现方法为将 OC/OD 门的输出连接在一起，并添加一个上拉电阻；即可

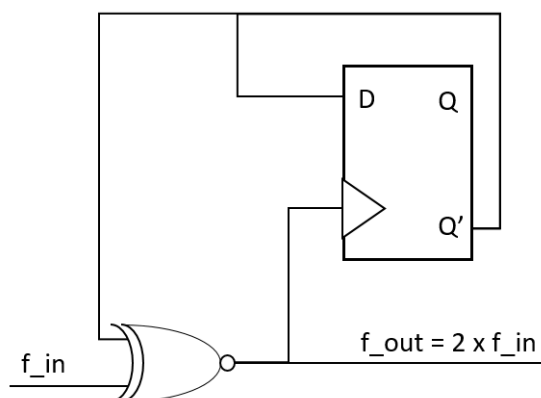
2.165 全加器检测 1 的个数

输入一个 8bit 数，输出其中 1 的个数。如果只能使用 1bit 全加器，最少需要几个？
需要 7 个全加器

有 8 个数：a7 a6 a5 a4 a3 a2 a1 a0，检测 1 的个数

a0	a3	a6	s0
a1	a4	a7	s1
a2	a5	0	s2
C0 S0	C1 S1	C2 S2	C3 S3
C0	C3	C5	
C1	S4	C4	
C2	0	0	
C4 S4	C5 S5	C6 S6	

2.166 数字倍频器



由于 Q' 到同或门的输入存在延时，造成出现倍频的情况

2.167 一个 16bit 的变量，判断能否被 5 整除

假设一个 16bit 的向量，用 16 进制表示为 ABCD，则值为

$$F = A * 16^3 + B * 16^2 + C * 16^1 + D * 16^0$$

$$F = A * (15 + 1)^3 + B * (15 + 1)^2 + C * (15 + 1)^1 + D$$

根据多项式展开的特点，会发现结果都是关于 15 的倍数与 A+B+C+D 的和，因此判断能被 5 整除，即判断 A+B+C+D 的值能否被 5 整除即可；

2.168 版图设计使用 P 衬底

p 衬底上做的是 nmos 器件，而由于电子迁移率要大于空穴，因此 nmos 器件比 pmos 器件更快，所以集成电路中使用 nmos 器件较多，所以多用 p 衬底！
不过现在用双井工艺的地方也挺多啊！