

期末考试安排

➤ 期末考试

- 时间：2024年6月27日 10:30-12:30
- 地点：鼓楼校区 - 教220、教222
苏州校区 - 南雍-西108

➤ 题型

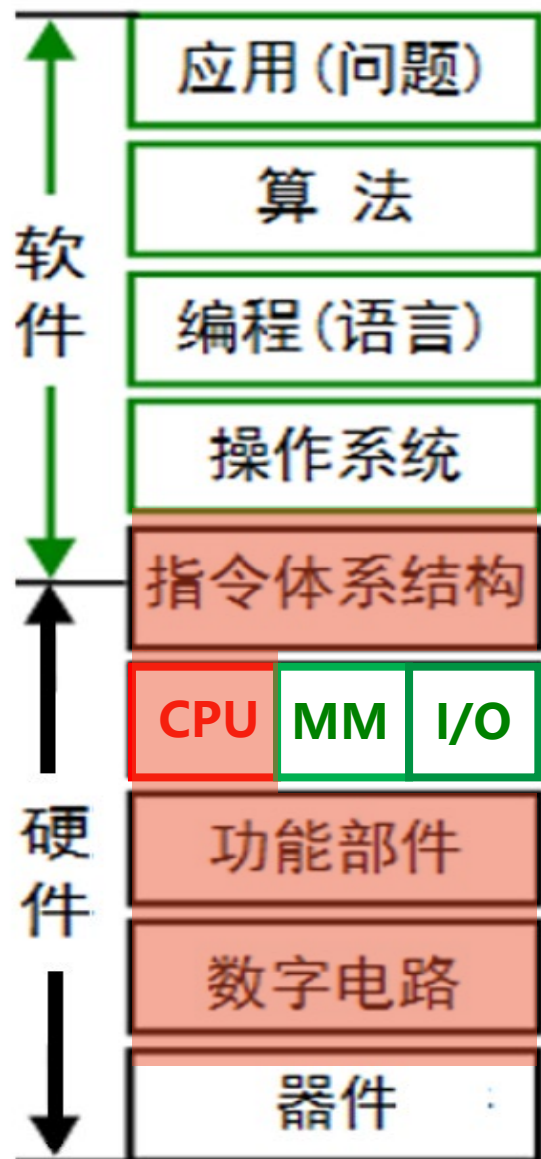
- 选择 (30分)
- 判断 (10分)
- 简答 (20分)
- 分析应用 (40分)

➤ 课程评分

- ❑ 期末考试 60% (KEY)
- ❑ 课后作业 20%
- ❑ 考勤 10%
- ❑ 实验考察 10%



考察范围 – 广度



➤ 课程内容:

- (1) 计算机系统与二进制编码
- (2) 数字逻辑基础
- (3) 组合逻辑电路
- (4) 时序逻辑电路
- (5) 可编程逻辑器件
- (6) 运算方法和运算部件
- (7) 指令系统
- (8) 中央处理器



考察范围 – 深度

知识点考察深度的标注

- 后续ppt中的底色标记:
 - 重要, 易考大题
 - 重要, 需要深入理解
 - 普通知识点
 - 不考
- PPT上带 “*” 的内容不考
- PPT不涉及的内容不考

不需要硬记的知识点

IEEE 754标准 (第一章)

ALU内部电路 (第六章)

乘法器/除法器电路和计算过程 (第六章)

RISC-V指令集卡和指令格式 (第七章)

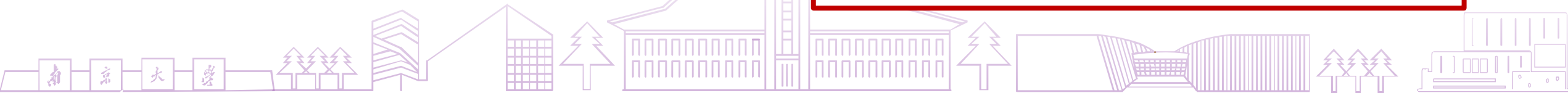
单周期数据通路的电路 (第八章)

多周期数据通路的状态图和电路 (第八章)

.....

不需要记 ≠ 不用掌握!

如果考试涉及到相关信息, 会在题目中给出



第1章计算机系统与二进制编码

➤ **总体要求：**了解计算机系统工作的基本原理，掌握计算机内部数据表示

➤ **具体要求**

➤ **计算机系统概述**

- ✓ 冯.诺依曼结构计算机
- ✓ 程序的表示和执行过程
- ✓ 计算机系统抽象层

了解计算机系统的基本工作原理和
计算机系统的基本框架

➤ **二进制数的编码表示**

- ✓ 外部信息和内部数据之间的关系
- ✓ 进位计数制（2、8、10、16进制数）

➤ **数值数据的表示**

- ✓ 定点数的二进制编码
- ✓ 无符号整数和带符号整数的表示
- ✓ 浮点数的表示

对具体问题
进行分析

➤ **非数值数据的表示、数据的宽度和存储排列**



第1章 知识点概览

- 冯.诺依曼结构基本思想、存储程序工作方式
- 程序和指令及数据之间的关系
- 进位计数制数之间的转换 (0xfa, FAH, 11111010B,)
- 定点数的编码：原码、补码、移码
- 整数的表示
 - 无符号数：正整数，用来表示地址等；带符号整数：用补码表示
- 浮点数的表示
 - 符号；尾数：定点小数；指数（阶）：定点整数（基不用表示）
- 浮点数的范围
 - 正上溢、正下溢、负上溢、负下溢；与阶码的位数和基的大小有关
- 浮点数的精度：与尾数的位数和是否规格化有关
- 浮点数的表示（IEEE 754标准）：单精度SP（float）和双精度DP（double）
- 十进制数的二进制表示（BCD码）
 - 有权BCD码（8421码）、无权BCD码（余3码、格雷码等）
- 非数值数据的表示（逻辑数据、西文字符、汉字）
- 数据的宽度和排列（位/字节/字；KB/MB/GB；大端/小端方式）



第2章 数字逻辑基础

➤ 总体要求：了解计算机系统最基本的物理基础和数学基础

➤ 具体要求

➤ 逻辑门和数字抽象

✓ 逻辑门和数字抽象

✓ CMOS晶体管和电气特性

了解基本逻辑门的CMOS管实现原理，以及电路的传播延迟

➤ 布尔代数

✓ 逻辑量和电平之间的关系、逻辑表达式

✓ 基本逻辑运算、公理系统和定律

➤ 逻辑关系描述

✓ 逻辑函数、真值表和波形图

✓ 标准范式的表示

➤ 逻辑函数化简

✓ 代数法和卡诺图化简

✓ 等效逻辑符号和逻辑函数变换

属于对后续章节中的具体问题进行分析的基础内容



第2章 知识点概览

- 现实世界的模拟信号需转换为数字信号。数字系统中的所有信号都是二值的，用0和1表示
- 逻辑门是最基础的数字电路，可通过CMOS晶体管实现
- 最基本的逻辑运算有与、或、非三种运算，对应的逻辑门分别为与门、或门和非门
- 布尔代数是数字系统分析和设计的基础理论工具，对应的公理系统和定理可对逻辑表达式进行化简，实现逻辑函数间的相互转换
- 可使用真值表、波形图以及逻辑表达式来描述逻辑变量间的关系
- 可使用代数法、卡诺图等来化简逻辑表达式
- 在实现数字系统时，为了提高速度、降低成本，通常利用与非门和或非门来构建电路。
- 等效电路：与非=非或、或非=非与



第3章 组合逻辑电路

➤ 总体要求：掌握最基本的组合逻辑电路的功能及其设计方法

➤ 具体要求

➤ 组合逻辑电路设计概述

✓ 组合逻辑电路设计的基本步骤

✓ 无关项、非法值和高阻态

属于后续电路设计的基础内容

➤ 典型组合逻辑部件设计

✓ 译码器和编码器

✓ 多路选择器和多路分配器

✓ 半加器和全加器

理解典型组合逻辑部件的功能，
并能在更复杂的逻辑电路中使用它们实现特定功能

➤ 组合逻辑电路的时序分析

✓ 传播延迟和最小延迟

✓ 竞争冒险

对具体问题进行分析设计

理解电路延迟和竞争冒险的概念，并能进行相关分析



第3章 知识点概览

- 数字逻辑电路由若干**元件**（可以是一个电路）和若干**结点**互连而成
- 组合逻辑电路的**输出**值仅依赖于当前**输入**值
- 组合逻辑电路可以是**两级**电路或**多级**电路，两级电路的传输时间短，但占用集成电路物理空间更多，需进行**时空权衡**
- **组合逻辑电路设计：功能分析-列表-化简-逻辑表达式-画图-评价**
- **无关项**指输出取值可任意的项，真值表中用d表示，可用于化简
- **非法值**指同时被高、低电平驱动的输出结点的值。
- **高阻态**是三态门输出结点的一种非正常逻辑态，相当于“断开”
- **典型组合逻辑部件：译码/编码器、多路选择/分配器、半加/全加器**
- **传输延迟**：关键路径上所有元件的传输延迟之和
- **最小延迟**：最短路径上所有元件的最小延迟之和
- **竞争、冒险(险象)、毛刺**：不同路径延迟作用在同一输出端而引起



第4章 时序逻辑电路

➤ 总体要求：掌握最基本的时序逻辑电路的功能及其设计方法

➤ 具体要求

➤ 时序逻辑电路概述

✓ 时序逻辑与有限状态机

✓ 时序逻辑电路基本结构及其定时

属于后续电路设计的基础内容

➤ 双稳态电路、锁存器和触发器

➤ 同步时序电路设计

✓ 状态图和状态表设计

✓ 状态化简和状态编码

✓ 电路设计和分析

对具体问题进行分析设计

➤ 典型时序逻辑部件设计

✓ 计数器

✓ 寄存器和寄存器堆

✓ 移位寄存器

理解典型时序逻辑部件的功能，
并能在更复杂的逻辑电路中使用它们实现特定功能



第4章 知识点概览

- 时序逻辑电路不仅依赖当前输入，还依赖电路**当前的状态**
- 可用时序逻辑电路实现**有限状态机**。有**Mealy**型和**Mooer**型两类
- 可用**状态图**或**状态表**描述有限状态机，圈表示状态，有向边表示输入/输出
- 锁存器(电平触发): **SR锁存器**(设置标志)、**D锁存器**(锁存数据D)
- 触发器(时钟信号clk边沿触发): **D触发器**(寄存器)、**T触发器**(计数或分频)
- 时序电路设计: **功能分析-状态图-状态化简和编码-逻辑表达式-画图-评价**
- 电路分析: **未用状态分析(挂起/无法自启动)**
- **定时分析(clk-Q时间、时钟周期、setup时间、hold时间)**
- 典型组合逻辑部件: **计数器、寄存器/通用寄存器组、移位寄存器**
- **计数器**: 同步/异步、加1/减1、行波(串行) 进位/并行进位
- **寄存器**: 由n个D触发器构成，同时由时钟信号clk定时
- **通用寄存器组**: 两个读口(组合逻辑); 一个写口(时序逻辑, clk和写使能)
- **移位寄存器(时序逻辑)**: 每次固定左移或右移1位或2位
- **桶型移位器(组合逻辑)**: 移位位数可变, 用大量多路选择器实现



第5章 FPGA设计和硬件描述语言

➤ 总体要求：了解几种可编程逻辑器件(PLD)、存储器阵列、ASIC等电路的特点和用途

➤ 具体要求

➤ 可编程逻辑器件 (PLD)

✓ PLD的基本概念和基本结构

✓ 简单PLD

PROM、PLA、PAL、GAL

✓ 复杂PLD

CPLD、FPGA

➤ 存储器阵列

✓ 静态RAM和动态RAM

✓ ROM

➤ 专用集成电路 (ASIC)

➤ FPGA设计概述和HDL基本概念

了解基本特点
和用途



第6章 运算方法和运算部件

➤ **总体要求：** 掌握基本运算部件和补码加/减运算电路的功能和结构，理解原码/补码乘法运算和原码/补码除法运算的基本原理和基本电路结构。

➤ **具体要求**

➤ **基本运算部件**

✓ 串行进位加法器、带标志加法器

✓ 算术逻辑运算部件 (ALU)

对具体问题
进行分析

➤ 带标志补码加减运算电路

➤ 乘、除运算和移位运算之间的关系

➤ 定点乘法运算及其电路

✓ 无符号整数乘法运算

✓ 原码和补码的乘法运算

理解乘、除运算的基本原理和
乘法器和除法器的基本结构

➤ 定点除法运算及其电路

✓ 无符号整数除法运算

✓ 原码和补码的除法运算



第6章 知识点概览

➤ 移位运算

- 逻辑移位：对无符号数进行，左（右）边补0，低（高）位移出
- 算术移位：对带符号整数进行，移位前后符号位不变，编码不同，方式不同。
- 循环移位：最左（右）边位移到最低（高）位，其他位左（右）移一位。

➤ 扩展运算

- 零扩展：对无符号整数进行高位补0。
- 符号扩展：对补码整数在高位直接补符。

➤ 加减运算

- 补码加/减运算：用于整数加/减运算。符号位和数值位一起运算，减法用加法实现。同号相加时，若结果的符号不同于加数的符号，则会发生溢出。
- 原码加/减运算：用于浮点数尾数加/减运算。符号位和数值位分开运算，同号相加，异号相减；加法直接加；减法用加负数补码实现。

➤ 乘法运算：用加法和右移实现。

- 补码乘法：用于整数乘法运算。符号位和数值位一起运算。采用Booth算法。
- 原码乘法：用于浮点数尾数乘法运算。符号位和数值位分开运算。数值部分用无符号数乘法实现。

➤ 除法运算：用加/减法和左移实现。

- 补码除法：用于整数除法运算。符号位和数值位一起运算。
- 原码除法：用于浮点数尾数除法运算。符号位和数值位分开运算。数值部分用无符号数除法实现。

乘/除运算器：由ALU + (移位)寄存器 + 计数器 + 控制逻辑实现。



第7章 指令系统

➤ 总体要求

掌握指令系统涉及的指令格式、指令类型、操作码编码、操作数寻址、标志信息生成与使用、指令设计风格等内容。

➤ 具体要求

➤ 指令系统设计

✓ 定长指令格式/变长指令格式

✓ 寻址方式

✓ 操作码的编码：定长 / 变长

✓ 标志信息的生成和使用

✓ 指令类型和指令设计风格

寄存器之间传送、算术/逻辑运算、

条件/无条件转移、调用/返回、

访存（装入/存储）……

➤ 实例：RISC-V指令系统

基于对RISC-V指令系统的理解，能对具体问题进行分析



第7章 知识点概览

- 指令格式：定长指令字、变长指令字
- 操作类型：数据传送、操作运算、逻辑运算、程序流控制（跳转）、系统控制
- 操作数类型：序数或指针、整数、实数、十进制数、字符串
- 操作数宽度：有多种，如字节、16位、32位、64位等
- 寻址方式
 - 立即、直接、间接、寄存器(直接)、寄存器间接、栈
 - 偏移寻址（用基地址+形式地址）：变址寻址、相对寻址、基址寻址
- 指令系统风格：决定了处理器的设计
 - 按地址码指定风格来分：累加器型、栈型、通用寄存器型、装入/存储型（运算只用寄存器）
 - 按指令系统的复杂度来分：CISC：复杂指令系统计算机；RISC：精简指令系统计算机
- RISC-V指令系统（有6种指令格式）
 - R-型为寄存器操作、I-型为短立即数或装入（Load）、S-型为存储（Store）、
 - B-型为条件跳转、U-型为长立即数操作、J-型为无条件跳转



第8章 中央处理器

- **总体要求：**理解指令系统与处理器的关系，理解单周期处理器、多周期处理器、流水线处理器设计的基本原理、电路结构和各控制信号含义。

- **具体要求**

- **中央处理器概述**

基于对RISC-V流水线CPU的理解，能具体分析

- ✓ 指令执行过程、CPU的基本组成、数据通路与时序控制、计算机性能与CPU时间

- **单周期数据通路设计**

- ✓ **RV32I单周期数据通路设计(考分析)：**扩展器、ALU、取指令电路、完整数据通路

- **单周期控制器设计**

- ✓ **RV32I指令与控制信号的关系、控制器电路结构（PLA阵列）**

- **多周期处理器设计**

- ✓ **与单周期处理器的差别、多周期数据通路设计基本思想、硬连线路控制器和微程序控制器、带异常处理的处理器设计**

- **流水线处理器设计**

- ✓ **流水线处理器设计基本思想**

了解多周期处理器的基本设计原理及基本结构



第8章 知识点概览 (1)

- CPU的主要功能：周而复始执行指令、处理异常中断
- CPU的内部结构：由数据通路(Datapath)和控制单元(Control unit)组成
 - 数据通路中包含组合逻辑单元（操作元件）和存储信息的状态单元
 - 操作元件：用于数据处理，如：加法器、运算器ALU、扩展器（0扩展或符号扩展）、多路选择器、以及状态单元的读操作线路等。
 - 状态单元包括触发器、寄存器、寄存器堆、数据/指令存储器等，用于对指令执行的中间状态或最终结果进行保存。
 - 控制单元对指令进行译码，与指令执行得到的条件码或当前机器的状态、时序信号（时钟）等组合，生成对数据通路进行控制的控制信号
- 计算机性能评价： $\text{CPU执行时间} = \text{程序指令条数} \times \text{CPI} \times \text{时钟周期}$
- CPU中的寄存器：
 - 用户可见寄存器（用户程序可使用）：通用寄存器、程序计数器PC
 - 控制和状态寄存器（内核程序可使用）：程序状态字寄存器PSWR（标志寄存器）
 - 内部寄存器（程序不可见）：IR、MAR、MDR.....



第8章 知识点概览 (2)

- 指令执行过程：取指、译码、取数、运算、存结果、查中断
- 数据通路的定时方式
 - 现代计算机都采用时钟信号进行定时，一旦时钟有效信号到来，状态单元可以开始写入信息
- 数据通路中信息的流动过程
 - 每条指令在取指令阶段和指令译码阶段都一样，但在数据通路中所经过的部件和路径可能不同
 - 数据在数据通路中的流动过程由控制信号确定，控制信号由控制器根据指令代码来生成
- 单周期处理器的设计
 - 每条指令都在一个时钟周期内完成；时钟周期以最长的Load指令所花时间为准
 - 无需加临时寄存器存放指令执行的中间结果；同一个功能部件不能重复使用
 - 控制信号在整个指令执行过程中不变，所以控制器设计简单，只要写出指令和控制信号之间的真值表，就可以设计出控制器
- 多周期处理器的设计
 - 每条指令分成多个阶段，每个阶段在一个时钟内完成；不同指令包含的时钟个数不同
 - 阶段的划分要均衡，每个阶段只能完成一个功能，如：一次ALU操作、存储器访问、寄存器存取
 - 需加内部寄存器（IR、MAR、MDR等）存放指令执行的中间结果；同一个功能部件能在不同的时钟中被重复使用；可用有限状态机来表示指令执行流程，并以此设计控制器



第8章 知识点概览 (3)

➤ 控制单元实现方式

— 有限状态机描述方式 (硬布线控制器)

- 每个时钟周期包含的控制信号的值的组合看成一个状态，每来一个时钟，控制信号会有一组新的取值，也就是一个新的状态；
- 所有指令的执行过程可用一个有限状态转换图来描述用一个组合逻辑电路（一般为PLA电路）来生成控制信号，用一个状态寄存器实现状态之间的转换；

— 微程序描述方式 (微程序控制器)

- 每个时钟周期所包含的控制信号的值的组合看成是一个0/1序列，每个控制信号对应一个微命令，控制信号取不同的值，就发出不同的微命令
- 多个微命令组合成一个微指令，多个微指令组成一个微程序，一个时钟执行一条微指令
- 每条指令对应一个微程序，执行时，先找到对应的第一条微指令，然后按照特定的顺序取出后续的微指令执行；所有微程序存放在控制存储器（CS）中，CS用ROM实现

➤ 指令流水线的设计

- 将每条指令的执行规整化为若干个同样的流水阶段
- 每个流水阶段的执行时间一样，都等于一个时钟
- 理想情况下，每个时钟有一条指令进入流水线，也有一条指令执行结束



NJU-3DV Lab

生成 ← 三维数字人 → 驱动



单幅图像生成三维数字人

"This middle-aged man is a westerner. He has big and black eyes with the double eyelid. He has a medium-sized nose with a high nose bridge. His face is square and medium. He has a dense and black beard. He looks like Tony Stark."



文本生成三维数字人



语音驱动的数字人动画（视频）



casual personal photos of [W] clothes



[V] in [W] clothes



casual personal photos of character [V]



[V] in armor

“文本+图像”生成全身三维数字人



三维姿态驱动的数字人动画（视频）

NJU-3DV Lab



南京大学三维视觉实验室招生



南京大学智能科学与技术学院三维视觉实验室（NJU-3DV）招募2025级保研生、直博生及申请考核制博士生。实验室聚焦三维计算机视觉与智能科学方向研究，包括三维重建、数字人建模、可微渲染、三维生成等课题。欢迎自驱力强、对三维视觉感兴趣、有志于在相关领域做出影响力工作的优秀同学联系！

导师简介

姚遥，南京大学智能科学与技术学院副教授，国家级人才计划青年项目入选者。前苹果公司高级研究员，Altizure创始团队核心成员（被苹果收购）。主要研究方向为三维计算机视觉，包括三维重建、可微渲染及三维内容生成。代表工作包括MVSNet系列工作、BlendedMVS数据集及NeILF系列工作，文章总引用数超3000，曾获2020年国际模式识别大会最佳学生论文奖。个人主页：<https://yoyo000.github.io>

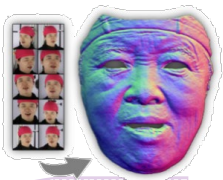
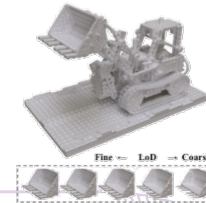
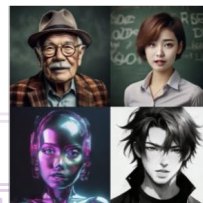
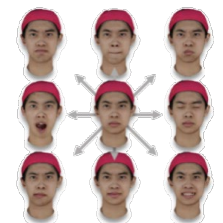
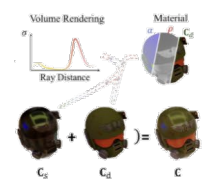
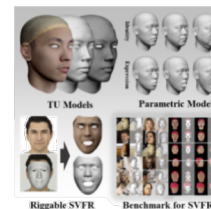
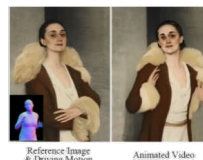
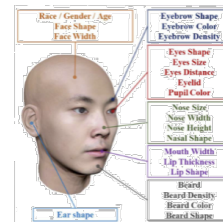
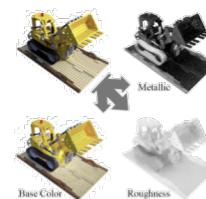
朱昊，南京大学智能科学与技术学院助理教授，姑苏青年教授，博导。聚焦三维数字人建模，在TPAMI、CVPR、AAAI等期刊会议上发表学术论文20余篇。代表工作为FaceScape系列，包括大规模高质量三维人脸数据集和数字人重建、生成、驱动模型。入选中国科协青年人才托举工程、腾讯犀牛鸟专项计划，获得IEEE杰出青年作者奖、南京大学博士生校长特别奖等荣誉。个人主页：<http://zhuhao.cc/home>

研究资源

实验室计算资源充裕，近两年新增获批经费超1200万元，硬件设施包括20余台高性能显卡服务器、50余台单反相机以及30余台4K高速同步摄像机。同时，实验室现与多个海内外学术界及工业界团队建立了良好的合作关系，已主动推荐同学至苹果、腾讯、阿里、蔚来及领域内创业公司实习。实验室将积极寻找途径让同学有机会进行百卡及更大规模的模型训练，近期工作包括 [Direct2.5](#)、[Champ](#)等。



个人主页: zhuhao.cc



祝大家取得好成绩！

Thanks!

