Final Review

浙江大学计算机学院 2024 秋冬朋辈辅学：《计算机组成》

刘建翔 & 黄宇凡

2024/12/28

Powered by Slidev

Outline

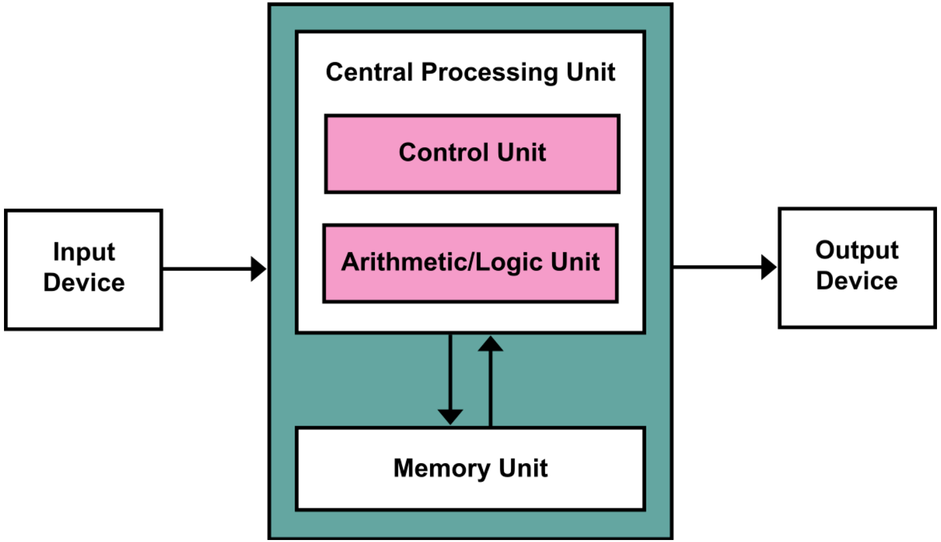
- 知识点回顾与梳理
- 历年卷习题讲解
- 考试技巧与经验分享

知识点回顾与梳理

Review

学了什么?

- Chapter 1: 基本概念与性能评估
- Chapter 3: 算术运算
- Chapter 2: RISC-V 指令集
- Chapter 4: 处理器
- Chapter 5: 层次化存储
- Chapter 6: I/O 系统



Chapter 1: Computer Abstractions and Technology

- 冯·诺依曼结构：控制单元、算术逻辑单元、存储器、输入输出设备
- 八个伟大思想
    - 摩尔定律
    - 抽象简化设计
    - 提速常规情况
    - 并行提高性能
    - 流水线提高性能
    - 预测提高性能
    - 内存层次结构
    - 冗余提高可靠性
- 性能指标：CPU Time、CPI、Amdahl’s Law、MIPS 等
    - 概念与计算方法

第一章易错

- 性能指标的计算
- 八个思想的理解

Chapter 3: Arithmetic for Computers

- 数字的表示
    - 整数：无符号数、有符号数（原码、补码、反码）
    - 浮点数：IEEE 754 标准、subnormal numbers
- 算术运算
    - 加法和减法：进位和溢出
    - 逻辑与移位：逻辑运算、算术移位、逻辑移位
    - 整数乘除法：Multiplier V3、Division
    - 浮点数加减：对阶、规格化、舍入
    - 浮点数乘除：指数运算、尾数运算
- 浮点数精确：舍入模式；Guard、Round、Sticky Bit

第三章易错

- 区分原码、补码、反码
- 逻辑移位与算术移位
- 乘除法器的工作原理
- 浮点数表示：0、NaN、无穷大

Chapter 2: Instructions: Language of the Computer

- RISC-V 指令集
  - 操作数表示：寄存器（别名及其含义）、立即数（范围）、内存（大小端）
  - 指令格式：R、I、S、B、U、J 类型
- 函数（过程）调用
  - 调用规约：传递参数、返回值
  - 状态保存：寄存器、栈
- 寻址方式
  - 寄存器寻址
  - 立即数寻址
  - 基址寻址
  - PC 相对寻址
- 同步：原子操作

第二章易错

- 寄存器的别名及使用规则
- 存储大端和小端的区别
- ld/sd、lb/sb、lh/sh、lw/sw 的区别
- 不同指令格式中的字段含义（尤其是立即数的生成方式）
- 函数调用：压栈、寄存器的保存与恢复
- JAL 与 JALR 的区别
- 书上 IndexOutofBound 的问题

Chapter 4: The Processor

- 单周期CPU
  - 控制单元
    - 哪些信号?
  - 数据通路
    - 哪些元件?
- 简单的流水线CPU
  - Which five stages?
    - 每一级干了啥
  - 如何解决几种冒险?
    - 数据冒险
    - 结构冒险
    - 控制冒险

第四章易错

- 采用 Stall 停几个周期才可以读数据（Double bump）
- Bypass 能解决所有数据冒险吗?
- Bypass 的条件?
- 提前进行 Branch?

Chapter 5: Large and Fast: Exploiting Memory Hierarchy

- 为啥需要缓存
- 哪几种缓存
  - N路组关联
  - 全关联
  - 直接映射
- 缓存策略
  - write back/write through
  - write allocate/no write allocate
  - LRU/FIFO/Random
- 3C
  - Capacity, Compulsory, Conflict

- 虚拟内存
  - 如何翻译?
  - TLB 是啥
  - 缺页会怎样 内存访问顺序?

第五章易错

- Write stall 是什么?
- 区分 3C

Chapter 6: I/O Systems

- Bus类型
  - 同步
  - 异步
- 总线仲裁
- IO 设备如何和CPU通信?
  - Memory mapped IO
  - 中断
  - 轮询
  - DMA

历年卷习题讲解

Exercises

Chapter 1

> **CPI 计算**
>
> **1. (23-24 Final)** A testbench contains 10% instructions of CPI 1, 20% instructions of CPI 2, 30% instructions of CPI 3, and 40% instructions of CPI 4. Calculate the average CPI of the testbench.

Answer: $CPI = 1 * 0.1 + 2 * 0.2 + 3 * 0.3 + 4 * 0.4 = 3.0$

> **CPI 计算**
>
> **2. (22-23 Final)** Processor A has average CPI of 1.5 with a clock rate of 3.0 GHz. Processor B has average CPI of 1.0 with a clock rate of 2.5 GHz. Processor C has average CPI of 2.2 with a clock rate of 4.0 GHz. Which processor is the fastest?

Answer:

Processor A: 1.5 / 3.0 = 0.5
Processor B: 1.0 / 2.5 = 0.4
Processor C: 2.2 / 4.0 = 0.55
Processor B is the fastest.

Chapter 3

整数加减法溢出与进位

1. (22-23 Final) Among all the addition operations of 2′s complement numbers, which one will cause overflow(OF) but no carry(CF)?

A. 0x12 + 0x34

B. 0x12 + 0xEF

C. 0x80 + 0x80

D. 0x12 + 0x78

Answer:

A: 0x12 + 0x34 = 0x46, OF=0, CF=0
B: 0x12 + 0xEF = 0x01, OF=0, CF=0
C: 0x80 + 0x80 = 0x00, OF=1, CF=0
D: 0x12 + 0x78 = 0x8A, OF=1, CF=1

Chapter 3

数据表示

2. (21-22 Final) Sort the following six numbers:

- 0xF0000000 in signed magnitude, 1’s complement, 2’s complement and IEEE 754
- 0xFFFFFFFF in 1’s complement and 2’s complement

Answer:

Sorted list (from smallest to largest):
0xF0000000 in IEEE 754: Approximately -1.5845633e29
0xF0000000 in signed magnitude: -1879048192
0xF0000000 in 2’s complement: −268435456
0xF0000000 in 1’s complement: −268435455
0xFFFFFFFF in 2’s complement: −1
0xFFFFFFFF in 1’s complement: -0

Chapter 3

浮点数运算

3. (23-24 Final) Among all the results of the following operations, which one is **NOT** a NaN?

A. 0 * inf

B. +inf - (-inf)

C. 8 / 0

D. Any arithmetic operation with NaN

Answer: C

Chapter 3

> **浮点数表示**
>
> **4. (23-24 Final)** IEEE 754-2008 defined a new floating-point format with 1 bit for sign, 5 bits for exponent, and 10 bits for fraction. What is the range of the floating-point number in this format?

Answer:

bias = 2^(5-1) - 1 = 15

smallest number: ± 0 00001 0000000000 = 2^(-14)

largest number: ± 0 11110 1111111111 = 2^15 * 1.1111111111

Chapter 2

> **机器码转汇编**
>
> **1. (23-24 Final)** What is the RISC-V assembly code of 0xE2952023?

Answer:

sw x9, -480(x10)，注意谁是 r1 谁是 r2

> **指令格式**
>
> **2. (22-23《计算机组成与设计》Final)** PC is currently 0x30000000, what is the range of PC with JAL instruction?

Answer:

JAL imm has range of -2^20 to 2^20-1
target = PC + imm * 2
min target = 0x30000000 + 2 * -2^20 = 0x2FE00000
max target = 0x30000000 + 2 ∗ 2^20-1 = 0x301FFFFF

Chapter 2

汇编编程

3. (21-22 Final) Write a RISC-V assembly code to exchange the values of two registers `x10` and `x11` **without using any additional registers**.

Answer:

```
xor x10, x10, x11
xor x11, x10, x11
xor x10, x10, x11
```

Chapter 2

大数字生成

4. (23-24 Final) Write a RISC-V assembly code to generate the number 0x12345678ABCDEF in register `x10`.

Answer:

```
lui x10, 0x12345         # Load upper 20 bits (0x12345) into x10, shifting it left by 12 bits
addi x10, x10, 0x678     # Add lower 12 bits (0x678) to x10
slli x10, x10, 24        # Shift left by 24 bits
lui t0, 0xABCDE          # Load upper 20 bits (0xABCDE) into t0
srli t0, t0, 8           # Shift right by 8 bits
addi t0, t0, 0xF         # Add lower 12 bits (0xF) to t0
or x10, x10, t0          # Combine the upper and lower parts into x10
```

Chapter 2

C 转汇编

5. (22-23《计算机组成与设计》Final) What is the RISC-V assembly code of the following C code `B[k]=A[i-j]` with `A` and `B` in `x10` and `x11` respectively, `i` in `x12`, `j` in `x13`, and `k` in `x14`?(Assume `A` and `B` are arrays of 64-bit integers)

Answer:

```
sub  x15, x12, x13          # x15 = i - j (calculate the index for A)
slli x15, x15, 3            # x15 = (i - j) * 8
ld   x16, 0(x15)            # x16 = A[i - j]
slli x14, x14, 3            # x14 = k * 8
sd   x16, 0(x14)            # Store the value in B[k]
```

Chapter 2

函数调用

6. (23-24 Final) Convert the following C code to RISC-V assembly code:

```c
int sum ( int num ) {
    if ( num < 10 )
        return num ;
    else return num%10 + sum( num/10 ) ;
}
```

where `int` is 64-bit integer and the input `num` is in register `x10`, and the output is in register `x11`.

Chapter 2

Answer:

```
sum:
    addi x5, x0, 10
    bgeu x10, x5, else   # 如果 num >= 10，跳转到 else
    addi x11, x10, 0
    jalr x0, 0(x1)
else:
    sd x1, -8(x2)        # 保存 ra
    addi x2, x2, -8      # 栈指针减 4
    rem x6, x10, x5      # x6 = num % 10
    div x10, x10, x5     # x10 = num / 10
    jal x1, sum          # 递归调用 sum
    add x11, x11, x6     # x11 = x11 + x6
    ld x1, -8(x2)        # 恢复 ra
    addi x2, x2, 8       # 栈指针加 4
    jalr x0, 0(x1)
```

Chapter 4

> **单周期CPU**
>
> 单周期 CPU 中，以下哪些操作是不能在一个时钟周期内完成的
>
> A. 从内存里读，并写数据
>
> B. ALU 计算，并写数据到内存
>
> C. 更新 PC, 并写数据到内存
>
> D. 从寄存器堆读值，进行 ALU 计算，并写数据到内存。

Answer: A

Chapter 5

Cache

1. 提高 cache 的组相联度可以优化

A. capacity miss

B. hit time

C. conflict miss

D. compulsory miss

Answer: C

Chapter 5

> **Cache**
>
> **2.** 在一个虚拟地址 36 位的地址空间中，每个页的大小是 8KB，页表项大小为 4Bytes。则页表的大小是：
>
> A. 8MB
>
> B. 4MB
>
> C. 32MB
>
> D. 512MB

Answer: C

考试技巧与经验分享

Tips

考试形式（请以考试通知为准）

- 半开卷：一张 A4 纸笔记
- 包含客观题和主观题
- 22 级题量较大，注意合理分配时间
- 诚信考试，不得抄袭

关于 cheating paper

- 对考试而言没啥用
- 但是可以边抄边复习
- 一些比较边缘的东西，比如 IO，中断那块可以抄上去以备万一
- 建议画个1-16的十进制、二进制、十六进制表，考试的时候可以加速转换

复习参考资料

- 课本：《计算机组成与设计》（第五版）
- 课件：PPT、朋辈辅学课件
- 实验：尤其是单周期 CPU 和流水线 CPU 的实验
- 历年卷：
  - https://www.cc98.org/topic/5922030 其中包含了 23-24 期末卷、22-23 期末卷等，还包含笔记；
  - https://www.cc98.org/topic/5640722 《计算机组成与设计》22-23 期末卷（23-24 学年的《计算机组成与设计》和《计算机组成》用的是同一份卷子）；

感谢聆听🙏

计算机学院-朋辈辅学 2024/12/28

祝大家考试顺利！🎉