走向RISC，指令集并行、CPU多核的

课时韩老师介绍过的，包括Amdahl's

是在作业里频繁登场。

人工智能为计算机架构设计带来的新

件设计也能为硬件架构带来灵感、

可能为架构创新带来机遇。这其实明

摩尔定律将不再适用，但是新兴的

蓬勃活力。领域特定架构、RISC-V.

是相当有潜力的。

的论调是向我们证明，当前景越发

也正是黎明(情)前然来临的契机。研发

的教训中获益，没必要着眼于两大定

不放，而应将架构师从有指令集的链

关注对公众对安全性的需求，轻量级开

有效加速商用。RISC理念经受时间考验

爆发可能在成本、能源、安全、性能的任

目前还不得而知。但只要国内相关行业从

石研究、敏锐捕捉行业发展方向，就有可

领域获得长足进步，就有可能立于潮流

，应对新需求既时是挑战又是机遇。

C2

1.

CISC架构：单个指令任务量大，长度灵活

优势：对编译器和程序存储空间的要求较低

劣势：硬件设计复杂，测试验证难度较高

RISC架构：单个指令任务量小，长度固定

优势：硬件设计较为简单，适合利用流水线提升性能

劣势：对编译器设计的要求较高，程序的代码密度较低

2.

基本指令集：

| | |
|---|---|
| RV32I | 32位整数指令集 |
| RV64I | 64位整数指令集 |
| RV128I | 128位整数指令集 |
| RV32E | 32位嵌入式指令集 |

标准扩展指令集：

| | |
|---|---|
| A | 原子操作指令，比如比较并交换、读改写等指令 |
| B | 位操作指令 |
| D | 双精度浮点指令 |
| F | 单精度浮点指令 |
| M | 包含乘法、除法、取模求余指令 |

4.

(1) RV32I add OP

RV64I addw OP-32

RV32I的add和RV64I的addw指令操作数不同

RV32I的add和RV64I的add指令操作数相同

分析：RV64I中的add指令仅把RV32I地址扩展，addw指令不忽略低位数据，对低32位数据进行结果进行符号扩展至64位并写入目标寄存器

(2) 不需要进行额外的符号扩展

理由：addw和addiw指令写入目标寄存器的结果已进行符号位扩展

5.

HINT的指令(提示指令)通常用于向微架构传达性能提示，除推动PC以及任何可用性能计数器外，并不改变任何体系结构可见的状态。具体实现可以选择忽略这些提示编码

HINT 不会改变体系结构状态的指令(提示)

例：目标寄存器为X0的ADD就是一个HINT

HINT指令空间含有标准HINTs和定制HINTs

预期的HINTs包括：关于内存系统时间和空间局域性的提示、分支预测提示，线程调度提示，安全性标签及用于模拟/仿真的仪表标志

6.

$a_2$ -3

$a_3$ 1

DIV单、REM、DIVW、REMW为~~有符号~~指令执行有符号数运算

DIVU、REMU、DIVUW、REMUW指令执行无符号数运算

Divide除法Div的值进行向量名入

11.

(1) 偏移量寻址

(2) 寄存器间接寻址

(3) 立即数寻址

(4) 寄存器直接寻址

(5) 偏移量寻址