



初识玄铁910

软件所智能软件中心PLCT实验室陈影实习生

目录

01 C910简介

02 C910架构特征

03 C910指令集

04 LLVM后端的自学程度

- 搭载AI向量加速引擎的高性能64位RISC-V架构多核处理器
 - 支持16核
 - 单核性能达到7.1 Coremark/MHz
 - 主频达到2.5GHz

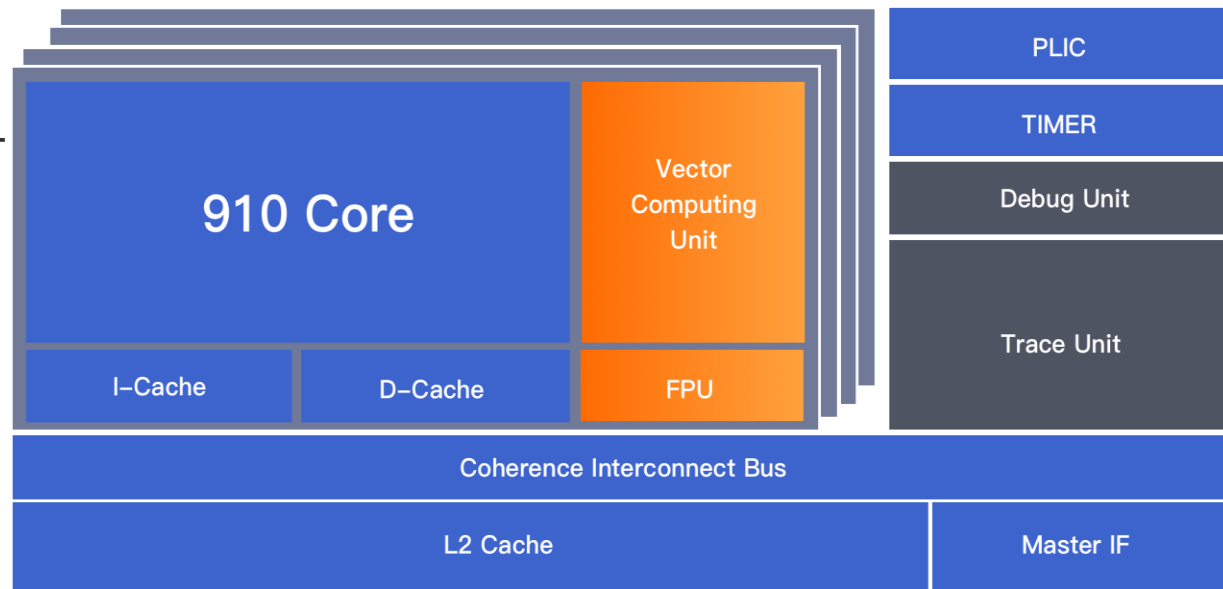
玄铁C910官网介绍: <https://www.t-head.cn/product/c910?spm=a2ouz.12987056.0.0.2cb96245CRx1Tp&lang=zh>

注：什么是CoreMark？

- 评价CPU性能指标的标准主要有三种：Dhrystone、MIPS、CoreMark，其中CoreMark是一种新兴流行的嵌入式系统处理器测试基准，被认为是比Dhrystone和MIPS更具有实际价值的测试基准？。
- CoreMark标准的测试方法：测试在某配置参数组合下单位时间内跑了多少次CoreMark程序，其指标单位为**CoreMark/MHz**。CoreMark数字越高，意味着性能更高。
- CoreMark程序使用C语言写成，包含如下四类算法：矩阵运算（通用矩阵运算）、数组（查找和排序）、状态机（用来确定输入流中是否包含有效数字）、CRC（循环冗余校验），都是在真实的嵌入式应用中很常见的操作，这也是CoreMark比其他测试标准更有实际价值的原因所在。

02 C910架构特征

- 指令集: RISC-V RV64GC/RV 64GCV
- 多核: 同构多核, 每个cluster 1-4核心可选配, 1-4 cluster可选配
- 流水线: 12级
- 微架构: 三发射(超标量), 深度乱序
- 高性能混合分支处理
- AI增强的向量计算引擎



<https://www.t-head.cn/product/c910?spm=a2ouz.12987056.0.0.2cb96245CRx1Tp&lang=zh>

注：12级流水线

CPU流水线技术是一种将指令分解为多步，并让不同指令的各步操作重叠，从而实现几条指令并行处理，以加速程序运行过程的技术。

流水线级数越多，工作效率越高。

取指令IF，译码ID，执行EX

超标量深流水乱序流水线

• 前端流水线

单周期获取8条指令
并行译码3条指令
并行发射8条指令

• 后端流水线

双发乱序存储访问单元

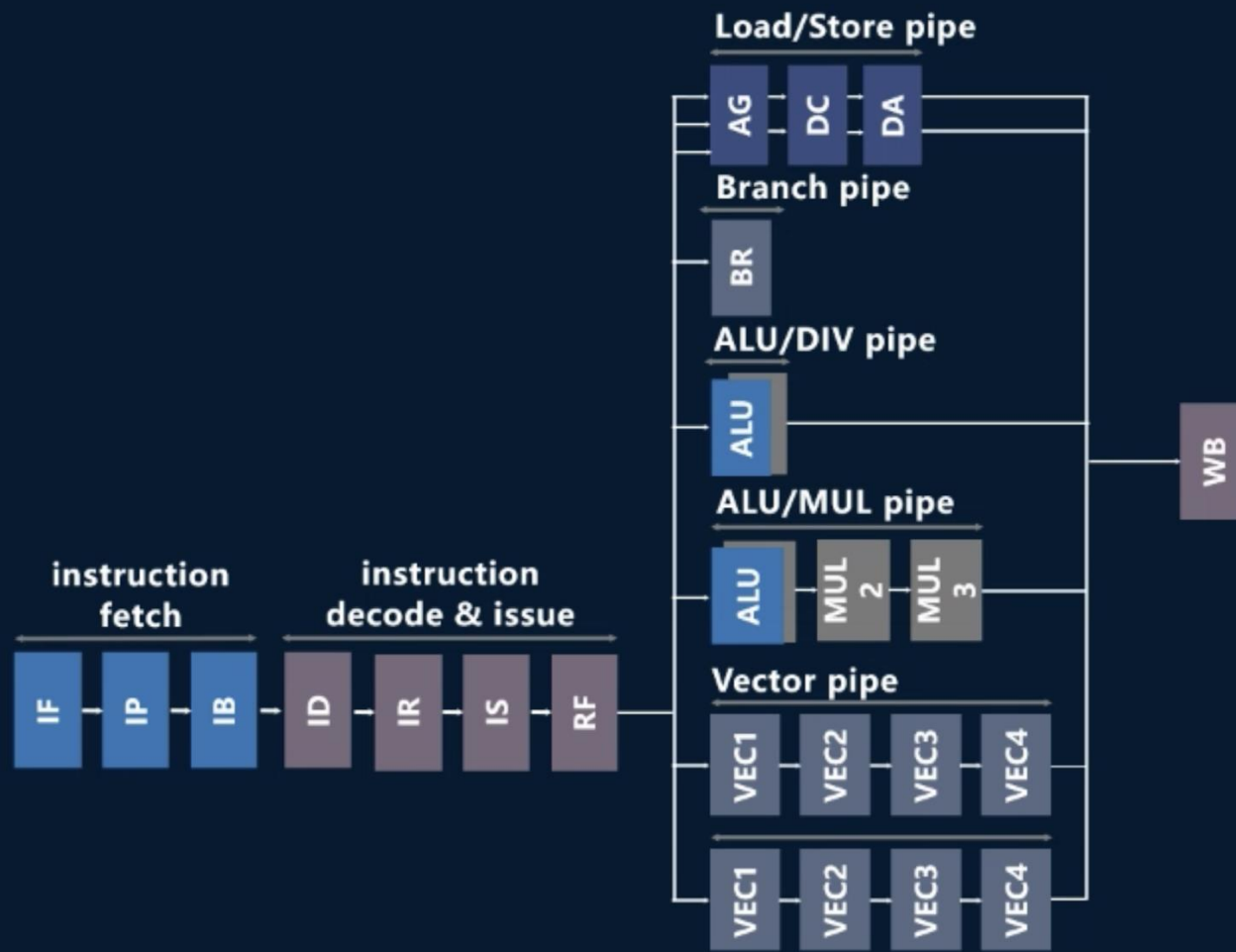
程之光: @孙浩 16,32位指令都有吧

郑重: 可能还要通过译码区分条件分支 无条件分支和函数调用返回

孙浩: @郑重 对的

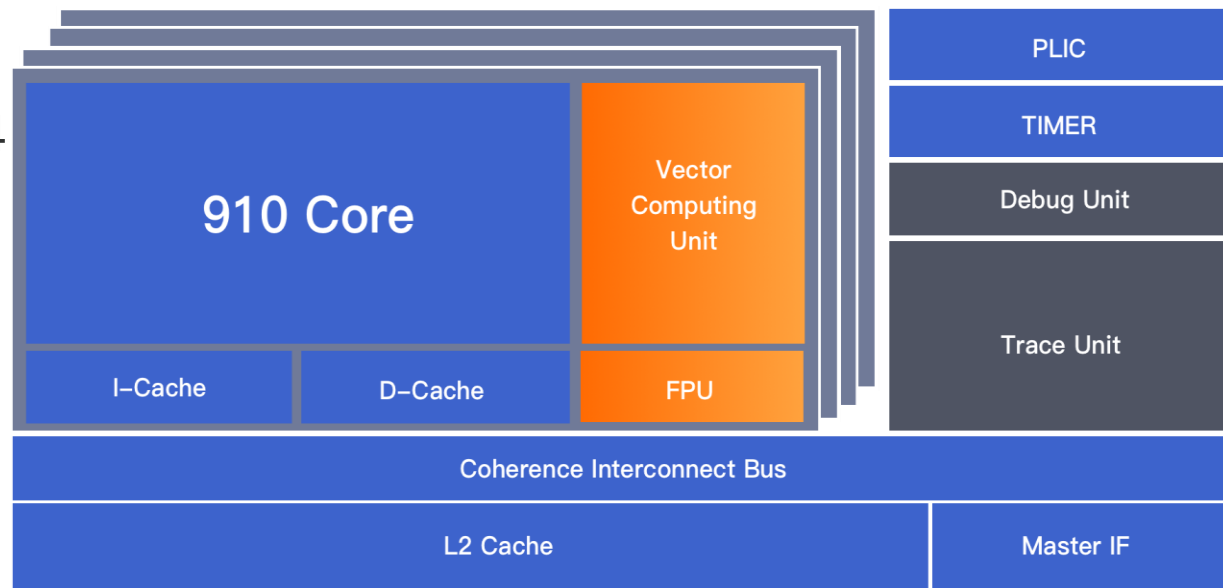
Bin Li: 指令fetch包括分支预测面积占总面积的百分比是多少?

Bin Li: 好像讲的都是CPI 微构型基础的 有



02 C910架构特征

- 指令集: RISC-V RV64GC/RV 64GCV
- 多核: 同构多核, 每个cluster 1-4核心可选配, 1-4 cluster可选配
- 流水线: 12级
- 微架构: 三发射(超标量), 深度乱序
- 高性能混合分支处理
- AI增强的向量计算引擎

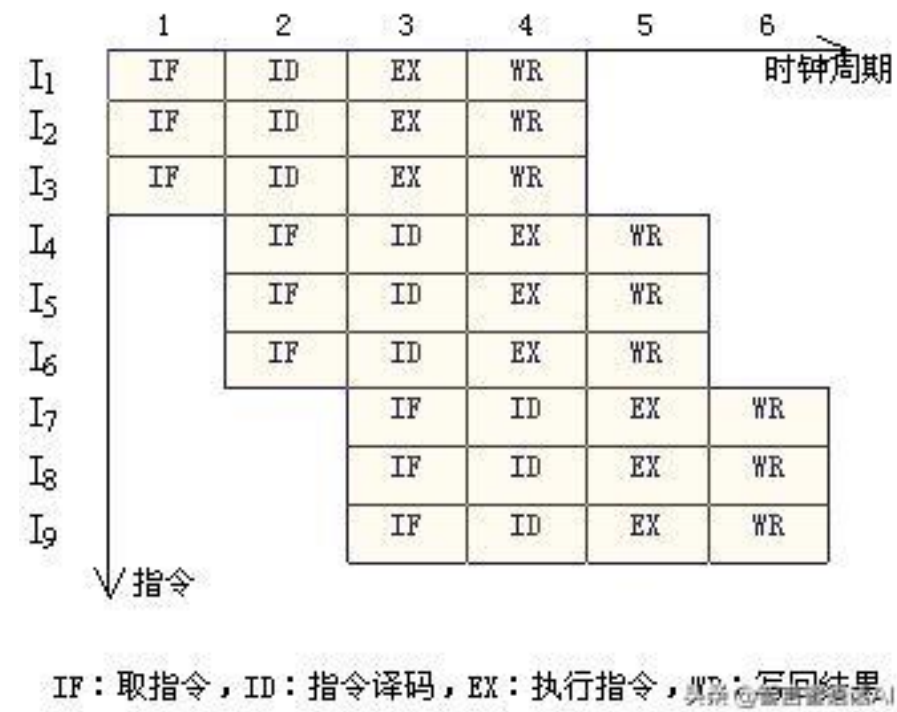


<https://www.t-head.cn/product/c910?spm=a2ouz.12987056.0.0.2cb96245CRx1Tp&lang=zh>

注：三发射

单发射是指指令在一个时钟周期内只从存储器中取出一条指令，并且只对一条指令进行译码，只执行一条指令，只写一个运算结果。

多发射是指在一个基本时钟周期内同时从指令Cache中读出多条指令，同时对多条指令进行译码。为了实现在一个时钟周期同时发射多条指令，通常需要有多个取指令部件，多个指令译码部件和多个写结果部件。

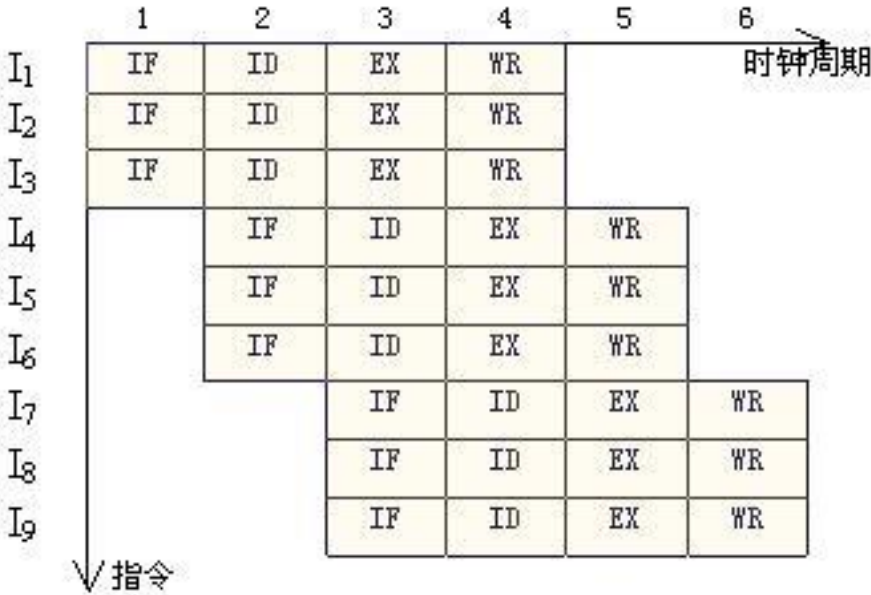


一个三发射的指令时序图

图源：https://www.toutiao.com/i6719679359768068616/?group_id=6719679359768068616

注：超标量

超标量是指在CPU中有一条以上的流水线，并且每时钟周期内可以完成一条以上的指令，这种设计就叫超标量技术。其实质是以空间换取时间。



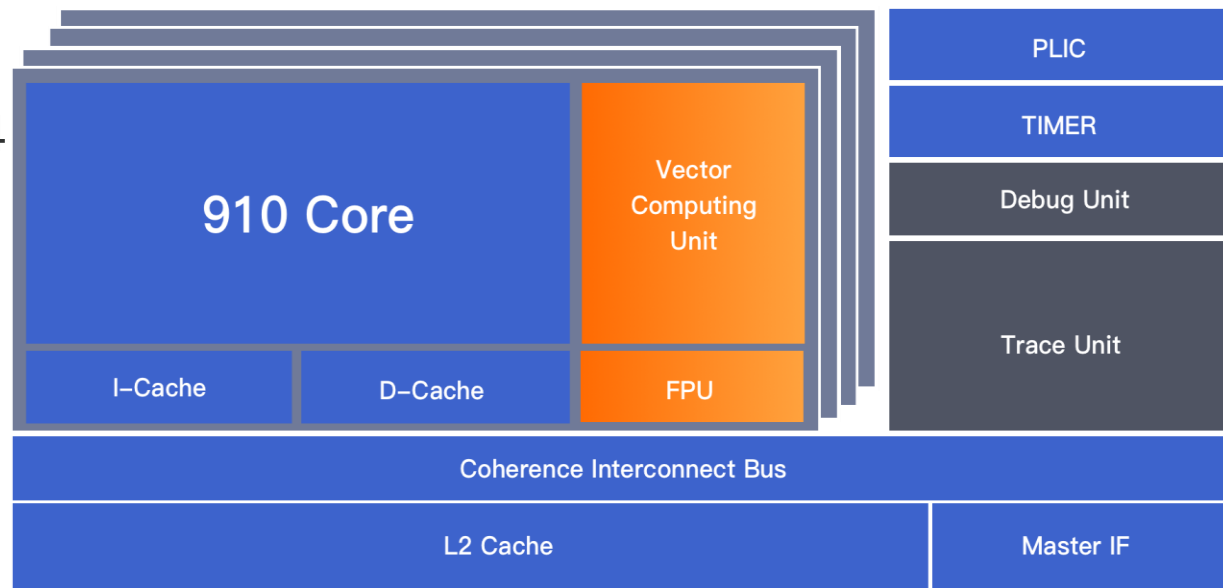
IF：取指令，ID：指令译码，EX：执行指令，WR：写回结果

一个三发射的指令时序图

图源：https://www.toutiao.com/i6719679359768068616/?group_id=6719679359768068616

02 C910架构特征

- 指令集: RISC-V RV64GC/RV 64GCV
- 多核: 同构多核, 每个cluster 1-4核心可选配, 1-4 cluster可选配
- 流水线: 12级
- 微架构: 三发射(超标量), 深度乱序
- 高性能混合分支处理
- AI增强的向量计算引擎



<https://www.t-head.cn/product/c910?spm=a2ouz.12987056.0.0.2cb96245CRx1Tp&lang=zh>

注：深度乱序

乱序执行是指先执行后面不依赖该数据的指令。

指令的乱序执行，包含有：

1) 指令相关，如当2条指令共用寄存器时，他们就有可能相关。

2) 控制相关，如前一条指令是跳转指令，而后一条指令的执行需要跳转指令的结果。

例如：

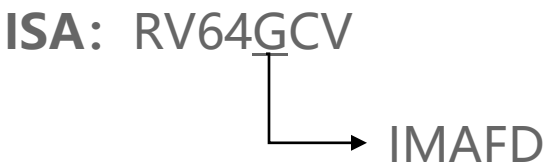
1. $b = a * 5$

2. $v = \underline{*}b$

3. $c = a + 3$

3	指令集.....	43
3.1	RV64GCV 指令.....	44
3.1.1	RV64I 整型指令集.....	44
3.1.2	RV64M 乘除法指令集.....	48
3.1.3	RV64A 原子指令集.....	49
3.1.4	RV64F 单精度浮点指令集.....	49
3.1.5	RV64D 双精度浮点指令集.....	52
3.1.6	RVC 压缩指令集.....	54
3.1.7	RVV 矢量指令集.....	56
3.2	扩展指令集.....	73
3.2.1	Cache 指令子集.....	73
3.2.2	多核同步指令集.....	75
3.2.3	算术运算指令集.....	76
3.2.4	位操作指令子集.....	77
3.2.5	存储指令子集.....	78

完全兼容RISC-V规范



Vector: RISC-V Vector Extension, FP16/32/64,
INT8/16/32/64

总结一下

C910采用12级超标量流水线，兼容RISC-V架构且针对算术运算、内存访问以及多核同步等方面进行了增强，同时标配内存管理单元，可运行Linux等操作系统。

采用3发射、8执行的深度乱序执行架构，配有单/双精度浮点引擎，可进一步选配面向AI加速的向量计算引擎，适用于5G、人工智能等对性能要求很高的应用领域。

C910的模拟器

发布版目录:

<https://github.com/c-sky/buildroot/releases>

包含标准指令集, THead扩展指令集两种版本, 操作步骤可以按照readme

1. 910 compact linux-5.1 glibc: 标准指令集

<https://c-sky.gitlab.io/-/buildroot/-/jobs/400737386/artifacts/output/images/readme.t>

对应的conf为: thead_910_compat_5.1_glibc_br_defconfig

2. 910 enhanced linux-5.1 glibc: 支持THead 扩展指令

<https://c-sky.gitlab.io/-/buildroot/-/jobs/400825255/artifacts/output/images/readme.txt>

对应的conf为: thead_910_enhanced_5.1_glibc_br_defconfig

https://occ.t-head.cn/community/download_detail?id=575996958993285120

04 关于LLVM后端的自学程度

```
test.ll (~/.llvmtoy) - gedit
打开(O) 保存(S)

; ModuleID = 'test.bc'
source_filename = "test.c"
target_datalayout = "e-m:e-i64:64-f80:128-n8:16:32:64-S128"
target_triple = "x86_64-unknown-linux-gnu"

@.str = private unnamed_addr constant [2 x i8] c"\0A\00", align 1

; Function Attrs: noinline nounwind optnone uwtable
define dso_local void @keep() #0 {
    %1 = call i32 @i8*, ... @printf(i8* getelementptr inbounds ([2 x i8], [2 x i8]*
@.str, i64 0, i64 0))
    ret void
}

declare dso_local i32 @printf(i8*, ...) #1

; Function Attrs: noinline nounwind optnone uwtable
define dso_local i32 @add(i32, i32) #0 {
    %3 = alloca i32, align 4
    %4 = alloca i32, align 4
    store i32 %0, i32* %3, align 4
    store i32 %1, i32* %4, align 4
    %5 = load i32, i32* %3, align 4
    %6 = load i32, i32* %4, align 4
    %7 = add nsw i32 %5, %6
    ret i32 %7
}
```

LLVM IR 制表符宽度: 8 行 1, 列 1 插入

clang -emit-llvm test.c -c -o test.bc

clang -emit-llvm test.c -S -c -o test.ll

llvm-as test.ll -o test.bc

llvm-dis test.bc -o test.ll

谢谢

欢迎交流合作

2020/02/05