



ARM, X86和MIPS



清酒与水

2 人赞同了该文章

ARM

ARM架构，过去称作高级精简指令集机器（Advanced RISC Machine，更早称作：Acorn RISC Machine），是一个32位精简指令集reduced instruction set computing（RISC）处理器架构，其广泛地使用在许多嵌入式系统设计。由于节能的特点，ARM处理器非常适用于移动通信领域，符合其主要设计目标为低成本、高性能、低功耗的特性。

ARM is a family of instruction set architectures for computer processors based on a reduced instruction set computing (RISC) architecture developed by British company ARM Holdings.

A RISC-based computer design approach means ARM processors require significantly fewer transistors than typical CISC x86 processors in most personal computers. This approach reduces costs, heat and power use. These are desirable traits for light, portable, battery-powered devices—including smartphones, laptops, tablet and notepad computers, and other embedded systems. A simpler design facilitates more efficient multi-core CPUs and higher core counts at lower cost, providing improved energy efficiency for servers.

ARM 公司本身并不靠自有的设计来制造或出售 CPU，而是将处理器架构授权给有兴趣的厂家。ARM处理器可以在很多消费性电子产品上看到，从可携式设备（PDA、移动电话、多媒体播放器、掌上型电子游戏，和计算机）到计算机外设（硬盘、桌面型路由器），甚至在导弹的弹载计算机等军用设施中都有他的存在。在此还有一些基于ARM设计的派生产品，重要产品还包括Marvell的XScale架构和德州仪器的OMAP系列。

ARM Holdings develops the instruction set and architecture for ARM-based products, but does not manufacture products.

ARM架构包含了以下精简指令集处理器的特性：

- 读取／存储 架构
- 不支持地址不对齐存储器访问（ARMv6内核现已支持）
- 正交指令集（任意访问指令可以任意的寻址方式访问数据Orthogonal instruction set）
- 大量的 16×32 -bit寄存器阵列（register file）
- 固定的32 bits操作码（opcode）长度，降低编码数量所产生的耗费，减轻解码和流水线化的负

担。

- 大多均为一个CPU周期运行。

较新的ARM处理器有一种16-bit指令模式，叫做Thumb，也许跟每个条件式运行指令均耗用4位的情形有关。在Thumb模式下，较小的opcode有更少的功能性。更短的Thumb opcode能更有效地使用有限的存储器带宽，因而提供比32位代码更佳的性能。Thumb-2技术首见于“ARM1156核心”，并于2003年发表。Thumb-2扩充了受限的16位Thumb指令集，以额外的32位指令让指令集的使用更广泛。因此Thumb-2的预期目标是要达到近乎Thumb的编码密度，但能表现出近乎ARM指令集在32位存储器下的性能。

To improve compiled code-density, processors since the ARM7TDMI have featured Thumb instruction set, which have their own state. (The "T" in "TDMI" indicates the Thumb feature.) When in this state, the processor executes the Thumb instruction set, a compact 16-bit encoding for a subset of the ARM instruction set. Most of the Thumb instructions are directly mapped to normal ARM instructions. The space-saving comes from making some of the instruction operands implicit and limiting the number of possibilities compared to the ARM instructions executed in the ARM instruction set state.

Announced in October 2011, ARMv8-A (often called ARMv8 although not all variants are 64-bit such as ARMv8-R) represents a fundamental change to the ARM architecture. It adds a 64-bit architecture, named "AArch64", and a new "A64" instruction set.

架构

处理器家族

ARMv1

ARM1

ARMv2

ARM2、ARM3

ARMv3

ARM6, ARM7

ARMv4

StrongARM、ARM7TDMI、ARM9TDMI

ARMv5

ARM7EJ、ARM9E、ARM10E、XScale

ARMv6

ARM11、ARM Cortex-M

ARMv7

ARM Cortex-A、ARM Cortex-M、ARM Cortex-R

ARMv8

Cortex-A50

X86

x86是一个指令集架构家族，最早由英特尔在“Intel 8086”CPU上开发出来。

x86 is a family of backward compatible instruction set architectures based on the Intel 8086 CPU.

该系列较早期的处理器名称是以数字来表示80x86。由于以“86”作为结尾，包括Intel 8086、80186、80286、80386以及80486，因此其架构被称为“x86”。由于数字并不能作为注册商标，因此Intel及其竞争者均在新一代处理器使用可注册的名称，如Pentium。现时英特尔把x86-32称为IA-32，全名为“Intel Architecture, 32-bit”；不过，由于x86包括16位处理器，这样的命名也出现麻烦。

8086是16位处理器；直到1985年32位的80386的开发，这个架构都维持是16位。接着一系列的处理器表示了32位架构的细微改进，推出了数种的扩充，直到2003年AMD对于这个架构发展了64位的扩充，并命名为AMD64。后来英特尔也推出了与之兼容的处理器，并命名为Intel 64。两者一般被称为x86-64或x64，开创了x86的64位时代。

英特尔早在1990年代就与惠普合作提出了一种用在安腾系列处理器中的独立的64位架构，这种架构被称为IA-64。IA-64是一种崭新的系统，和x86架构完全没有相似性；不应该把它与x86-64或x64弄混。

x86架构是重要地可变指令长度的CISC（复杂指令集计算机，Complex Instruction Set Computer）。字组（word, 4位组）长度的存储器访问允许不对齐存储器地址，字组是以低位字节在前的顺序存储在存储器中。向后兼容性一直都是在x86架构的发展背后一股驱动力量（设计的需要决定了这项因素而常常导致批评，尤其是来自对手处理器的拥护者和理论界，他们对于一个被广泛认为是落后设计的架构的持续成功感到不解）。但在较新的微架构中，x86处理器会把x86指令转换为更像RISC的微指令再予执行，从而获得可与RISC比拟的超标量性能，而仍然保持向前兼容。x86架构的处理器一共有四种执行模式，分别是真实模式，保护模式，系统管理模式以及虚拟V86模式。

操作码长度为1、2或3字节，此外ModR/M中还可能还有3位。对于双字节指令码或三字节指令码，其中的第1个字节为0FH，用于与指令前缀区分。

在1999年Intel推出SSE指令集，增加了八个新的128-bit暂存器（不跟其他的暂存器重叠使用）。这些指令类似于AMD的3DNow!，主要是增加浮点数运算的SIMD指令。

MIPS

MIPS架构（英语：MIPS architecture，为Microprocessor without Interlocked Pipeline Stages的缩写，亦为Millions of Instructions Per Second的相关语），是一种采取精简指令集（RISC）的处理器架构，1981年出现，由MIPS科技公司开发并授权，广泛被使用在许多电子产品、网络设备、个人娱乐装置与商业装置上。最早的MIPS架构是32位元，最新的版本已经变成64位元。

MIPS cores have been commercially successful, now being used in many consumer and industrial applications. MIPS cores can be found in newer Cisco, Linksys and Mikrotik's routerboard routers, cable modems and ADSL modems, smartcards, laser printer engines,

set-top boxes, robots, handheld computers, Nintendo 64, Sony PlayStation 2 and Sony PlayStation Portable. In cellphone/PDA applications, MIPS has been largely unable to displace the incumbent, competing ARM architecture.

ARM与MIPS平台优劣对比分析

摘自: cnblogs.com/xmphoenix/a...

1.流水线结构 pipeline

– MIPS 是最简单的体系结构之一, 所以使大学喜欢选择 MIPS 体系结构来介绍计算体系结构课程。

– ARM has barrel shifter

shifter是两面性的, 一方面它可以提高数学逻辑运算速度, 另一方面它也增加了硬件的复杂性。所以和可以完成同样功能的adder/shift register相比, 效率更高, 但是也占用更多的芯片面积。

– MIPS have “branch delay slot” and “load delay slot”

MIPS使用编译器来解决上面的两个问题。因为MIPS最初的设计思想就是使用简单的RISC硬件, 然后靠编译器及其他软件技术, 来达成RISC的完整概念。

2.指令结构 instruction

– MIPS have 32bit and 64bit architecture, but ARM only have 32bit architecture

ARM11 局部64位

– MIPS是开放式的架构, 用户可以在开发的内核中加入自己的指令,

– ARM has 4-bit condition code in every instruction

ARM在这一点很像x86。MIPS在MIPS IV也加入” conditional move” 指令, 来提高pipeline的效率。

– ARM has pre- and post-increment addressing modes

auto-increment/decrement on load/store instructions

– 在节省代码空间方面, MIPS16 很类似ARM Thumb

3.寄存器 register

-由于MIPS内核中有32个寄存器(Register), 而ARM只有16个, 这种结构设计上的先天优势, 决定了在同等性能表现下, MIPS的芯片面积和功耗会更小。

– ARM 有一组特殊用途寄存器cp0-cp15, 可以使用MCR, MRC等指令控制; 相对应的, MIPS也有cp0-0-30, 使用mfc0, mtc0 指令控制。

– Register banking in ARM. r8-r12 FIQ mode; r13:SP r14:LR

感觉不出banked register有什么好处。

– MIPS has a hard-wired-to-zero register, but ARM not

MIPS use register \$0 for Zero

4.地址空间 address space

– MIPS 起始地址是0xbfc00000, 会有4Mbyte的大小限制, 但一般MIPS芯片都会采取一些方法解决这个问题。

ARM没有这种问题。

MIPS24K 起始地址改到了0xbf000000, 现在有16Mbyte的空间了。

– MIPS don’ t have to turn paging on to enable the cache.

MIPS have the address space for both cache and un-cache
but ARM need enable/disable cache

5.功能 function

– Float point: MIPS64 has.

ARM' s support for FP is limited, and usually not included, and it is a 32 bit architecture
– ARM use JTAG,MIPS use EJTAG。Debug工具一般两种都支持。使用起来感觉差不多。

6.性能 performance

– 具体性能比较,因为差异性太大,所以很难分出谁好谁坏。从个人经验来讲 MIPS4k和ARM9基本上是同一个级别的,但ARM9性能似乎要比MIPS4K好。

同样是32bit的MIPS24K性能上比MIPS4K有很大提升,也应该比ARM9要好些。

因为没有用过ARM11和MIPS34K的芯片,没法比较,但感觉这两个似乎是一个级别的。

Cortex-A8和 MIPS 74K都是最新的设计,应该性能也差不多。

7.应用

– 在1000MHz以上的应用,很难找到采用ARM架构的产品。

MIPS架构用在200MHz或者是266MHz以下的应用比较少,而这恰恰是ARM的主攻市场。

– ARM 由于功耗小,普遍用在在手机/PDA等便携式消费电子领域; MIPS 在住宅网关、线缆调制解调器、线缆机顶盒等,由于MIPS 多核的发展,现在大型网关设备也多用它。

– ARM 采用硬核授权;MIPS 采用软核授权,用户可以自己配置,做自己的产品。

8.未来发展

– ARM的下一代走向多内核结构,而MIPS公司的下一代核心则转向硬件多线程功能(multithreading)

MIPS的multi-threading 很类似Intel 的 HyperThreading技术。从现在的发展来看,多内核占上风。

2008.12.21: 其实今天看来,也不好妄自评论孰好孰坏,虽说,HT技术在intel上发展并不好(已经被dual core)代替,但是并不能推断说MT在MIPS上就发展不好,毕竟mips的应用场合多是嵌入式领域,而MT的功耗小芯片面积小的优势正好能发挥作用。

2008.12.29: MIPS也并不是只有multithreading, 其实现在在很多的MIPS cpu的厂商都有multi-core 的mips cpu在出,比如cavium, broadcom, infineon,国内的龙芯也有多核的产品。

2009.6.11: MIPS的多核发展显然比ARM要好。从 Cavium,RMI,公司的产品大量的应用便可见一斑。

9.总结

自己感觉ARM和MIPS在一开始的RISC的设计上有很多不同,但随着技术的发展,各自扬长避短,好的技术大家都会使用。比如ARM11和MIPS R1000就使用了很多一样的技术。感觉RISC做到了极至就都一样了。

Linux-C语言学习交流群【721709245】在学的进群一起交流,资料自己群文件下载

相关资料:

嵌入式开发直播课 - 走进嵌入式

www.makeru.com.cn/live/1392_303.html?...



嵌入式开发直播课 - ARM体系结构

www.makeru.com.cn/live/1392_587.html?...



发布于 2020-07-10 13:51

精简指令集 (RISC)

中央处理器 (CPU)

ARM

写下你的评论...



还没有评论，发表第一个评论吧

推荐阅读

ARM vs X86 之争尘埃落定

虽然很久以前我就遇见到了 ARM 平台的统治力（为何 iPad Pro 和 Surface 是不同的生产力工具？／用 iPad Pro 工作，以及再见 x86）。但是，当时的评论里面有有力的证明：在办公软件和游戏两…

李楠

ARM 总线介绍

1. AMBA 介绍AMBA: Advanced Microcontroller Bus Architecture 主要用于SOC内部单元的管理和互联 发展历程： 2. APB 总线介绍 APB: Advanced Peripheral Bus 2.1 APB2 Interface Signal 3. …

麦墨

关于ARM和x86的能效

曾经回答过一个问题。夏晶晶：请问 X86 与 ARM 的功耗控制有什么区别？这事其实很多人是不信的，还有说我是AMD高端黑_(:3」_/_ 我换一个视角给一组数字供参考。天底下各家foundry目前都…

夏晶晶



arm的那些

AI电堂