

ARM与MIPS的详细对比

版本：v1.2

Crifan Li

摘要

此文主要介绍ARM和MIPS之间的区别和联系,包括ARM的Thumb,MIPS的MIPS16e ASE



本文提供多种格式供：

在线阅读	HTML ¹	HTMLs ²	PDF ³	CHM ⁴	TXT ⁵	RTF ⁶	WEBHELP ⁷
下载（7zip压缩包）	HTML ⁸	HTMLs ⁹	PDF ¹⁰	CHM ¹¹	TXT ¹²	RTF ¹³	WEBHELP ¹⁴

HTML版本的在线地址为：

http://www.crifan.com/files/doc/docbook/arm_vs_mips/release/html/arm_vs_mips.html

有任何意见，建议，提交bug等，都欢迎去讨论组发帖讨论：

http://www.crifan.com/bbs/categories/arm_vs_mips/

修订历史

修订 1.2	2013-10-04	crl
1. 通过Docbook发布 2. 更新xml:id		
修订 1.0	2011-07-19	crl
1. 详细对比了ARM和MIPS的区别		

¹ http://www.crifan.com/files/doc/docbook/arm_vs_mips/release/html/arm_vs_mips.html

² http://www.crifan.com/files/doc/docbook/arm_vs_mips/release/htmls/index.html

³ http://www.crifan.com/files/doc/docbook/arm_vs_mips/release/pdf/arm_vs_mips.pdf

⁴ http://www.crifan.com/files/doc/docbook/arm_vs_mips/release/chm/arm_vs_mips.chm

⁵ http://www.crifan.com/files/doc/docbook/arm_vs_mips/release/txt/arm_vs_mips.txt

⁶ http://www.crifan.com/files/doc/docbook/arm_vs_mips/release/rtf/arm_vs_mips.rtf

⁷ http://www.crifan.com/files/doc/docbook/arm_vs_mips/release/webhelp/index.html

⁸ http://www.crifan.com/files/doc/docbook/arm_vs_mips/release/html/arm_vs_mips.html.7z

⁹ http://www.crifan.com/files/doc/docbook/arm_vs_mips/release/htmls/index.html.7z

¹⁰ http://www.crifan.com/files/doc/docbook/arm_vs_mips/release/pdf/arm_vs_mips.pdf.7z

¹¹ http://www.crifan.com/files/doc/docbook/arm_vs_mips/release/chm/arm_vs_mips.chm.7z

¹² http://www.crifan.com/files/doc/docbook/arm_vs_mips/release/txt/arm_vs_mips.txt.7z

¹³ http://www.crifan.com/files/doc/docbook/arm_vs_mips/release/rtf/arm_vs_mips.rtf.7z

¹⁴ http://www.crifan.com/files/doc/docbook/arm_vs_mips/release/webhelp/arm_vs_mips.webhelp.7z

ARM与MIPS的详细对比:

Crifan Li

版本 : v1.2

出版日期 2013-10-04

版权 © 2013 Crifan, <http://crifan.com>

本文章遵从 : [署名-非商业性使用 2.5 中国大陆\(CC BY-NC 2.5\)](#)¹⁵

¹⁵ http://www.crifan.com/files/doc/docbook/soft_dev_basic/release/html/soft_dev_basic.html#cc_by_nc

目录

缩略词	1
1. 在精简指令集以减小代码空间占用方面的对比	2
1.1. ARM中的Thumb模式	2
1.1.1. 软件检测CPU处于何种模式	2
1.2. MIPS中的MIPS16e ASE	2
1.2.1. MIPS16e ASE的优势和劣势	2
1.2.2. 软件检测CPU处于何种模式	3
1.2.3. MIPS16e ASE的一些细节特性	3
参考书目	5

插图清单

1.1. Config1(CP0 Register 16, Select 1)寄存器域值说明	3
--	---

缩略词

ASE (ASE)

Application Specific Extension

(根据应用的) 专用扩展

第 1 章 在精简指令集以减小代码空间占用方面的对比

1.1. ARM中的Thumb模式

常见的ARM是32位的，其中有个Thumb模式，即16位模式。

在标准32位模式下，可以切换到Thumb模式下，这样的话，代码大小会压缩不少，以此提高空间利用率。

同样的代码，如果用Thumb模式的话，比普通ARM模式的指令，代码大小最大可以减少大约40%。

程序可以通过对应的指令，在ARM和Thumb模式之间切换。

1.1.1. 软件检测CPU处于何种模式

软件可以通过检测CPSR寄存器中Thumb位，来判断当前是处于ARM模式还是Thumb模式。

1.2. MIPS中的MIPS16e ASE

对应的，MIPS中也有个类似的做法。

普通MIPS是32位的，当然也有MIPS64的。

MIPS中有个MIPS16e ASE，其中ASE是Application Specific Extension，（根据应用的）专用扩展。

关于为何叫做ASE，我的理解是，MIPS16e ASE是针对MIPS32来说的，是对其一种扩展，是在，某些特定的应用（多为嵌入式应用）下减少代码大小，方面的扩展，因此，才叫做Specific Extension专用扩展。

MIPS16e ASE也常被简称为MIPS16e。

在MIPS16e ASE之前，有个MIPS16™，提供了一些额外的指令，用于提高代码的兼容性。而MIPS16e ASE，可以看作是MIPS16™的增强版。

相比MIPS32，利用MIPS16e编译的应用程序平均减小30%，最多可减少40%。

MIPS16e的指令集，兼容MIPS32 和 MIPS64的指令集。

程序也可以在MIPS16e 模式和MIPS 32位模式之间切换。

1.2.1. MIPS16e ASE的优势和劣势

用了MIPS16e ASE后，代码量减少，也相应的带来一个好处，还能提高指令缓存命中率(cache hit rate)，即降低了cache miss rate。

而由于缓存是个耗电大户，因此，相应地减少了功耗，实现省电，环保（有利于生态系统Ecosystem）的功效。

与优点相对应的缺点是，MIPS16e ASE比普通的MIPS32，增加了解码的复杂度，即解码时间相对增加了。

不过，拿MIPS16e ASE的劣质和优势相比，总体优势还是很明显的。

不过，貌似实际应用中用到MIPS16e ASE的不多。

1.2.2. 软件检测CPU处于何种模式

软件中可以通过检测MIPS16e ASE中的Config1 CP0寄存器的CA bit位来判断是否是处在MIPS16e ASE模式。

图 1.1. Config1(CP0 Register 16, Select 1)寄存器域值说明

Table 5-22 Config1 Register Field Descriptions — Select 1 (Continued)

Fields		Description	Read/ Write	Reset State
Name	Bit(s)			
DL	12:10	This field contains the data cache line size. If a data cache is present, it must contain a line size of 16 bytes. 0x0: No Dcache present 0x3: 16 bytes 0x1, 0x2, 0x4 - 0x7: Reserved	R	Preset
DA	9:7	This field contains the type of set associativity for the data cache: 0x0: Direct mapped 0x1: 2-way 0x2: 3-way 0x3: 4-way 0x4 - 0x7: Reserved	R	Preset
0	6:5	Must be written as zero; returns zero on read.	0	0
PC	4	Performance Counter registers implemented. Always a 0 since the cores do not implement any.	R	0
WR	3	Watch registers implemented. This bit always reads as 1 since the cores each contain one pair of Watch registers.	R	1
CA	2	Code compression (MIPS16™) implemented. This bit always reads as 0 because MIPS16 is not supported.	R	0
EP	1	EJTAG present. This bit is always set to indicate that the core implements EJTAG.	R	1
FP	0	FPU implemented. This bit is always zero since the core does not contain a floating-point unit.	R	0

1.2.3. MIPS16e ASE的一些细节特性

1. MIPS16e指令为16位指令
2. 32个通用寄存器中有8个可用于MIPS16e模式；其中包括常用于变元传递(argument passing)和返回值的寄存器
3. 与MIPS32一起使用时，支持8位、16位和32位数据类型；与MIPS64一起使用时，支持8位、16位、32位和64位数据类型
4. MIPS16e 和 MIPS32/64之间的模式切换允许特权架构/浮点/ASE支持；通过一条特殊的跳转指令来实现模式切换的软件控制
5. EXTEND指令可让16位指令有效创建16位立即值(immediate value)
6. SAVE/RESTORE指令可提高栈帧处理(stack frame handling)的代码密度
7. PC相关寻址可提高创建导入/存储地址的代码密度

8. 某些指令采用隐型寄存器描述符(implied register specifier)来提高代码密度--用于导入/存储的堆栈指针或程序计数器；分配一个GPR(通用寄存器)来比较分支结果
9. 仍支持专用宏指令序列

参考书目

- [1] [MIPS16e架构](#)¹
- [2] [MIPS16e[®]](#)²
- [3] [MIPS32™ Architecture for Programmers, Volume IV-a: The MIPS16e™ Application-Specific Extension to the MIPS32™ Architecture](#)³
- [4] [关于MIPS16e指令集](#)⁴
- [5] [Wiki: MIPS](#)⁵

¹ <http://www.mips.com.cn/products/architectures/MIPS16e/>

² <http://www.mips.com/products/architectures/mips16e/>

³ <http://www.weblearn.hs-bremen.de/risse/RST/docs/MIPS/MD00076-2B-MIPS1632-AFP-00.96.pdf>

⁴ <http://www.linuxforum.net/forum/showflat.php?>

<http://www.linuxforum.net/forum/showflat.php?Cat=&Board=cpu&Number=351019&page=&view=&sb=&o=&vc=1>

⁵ http://en.wikipedia.org/wiki/MIPS_architecture