自主决定命运，创新成就未来
www.loongson.cn

LoongArch架构的ACPI支持

吕建民/20211024

LoongArch架构介绍

- **龙芯中科基于二十年的CPU研制和生态建设积累推出了龙芯架构（Loongson Architecture，以下简称龙芯架构或LoongArch），包括基础架构部分和向量指令、虚拟化、二进制翻译等扩展部分，近2000条指令。**

龙芯中科

LOONGSON

龙芯架构参考手册

卷一：基础架构

V1.0

龙芯中科

LOONGSON

龙芯架构参考手册

卷二-a：向量指令扩展

V1.0

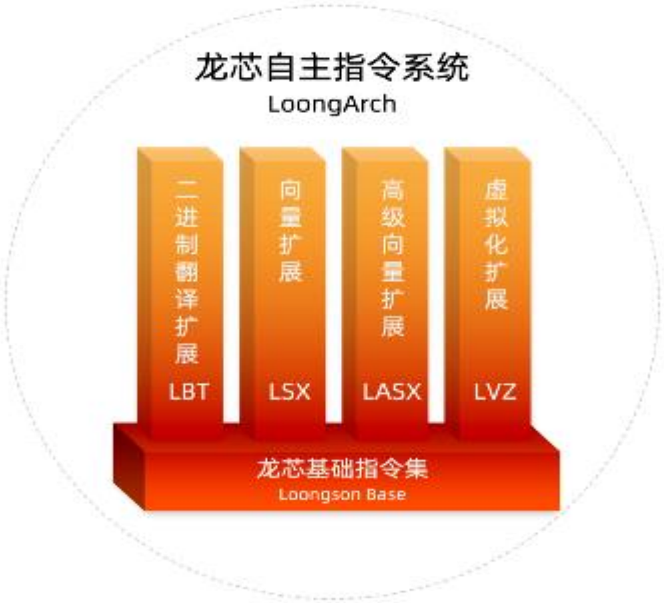
龙芯中科

LOONGSON

龙芯架构参考手册

卷三：虚拟化及二进制翻译扩展

V1.0



- **基础架构通过国内第三方知名知识产权评估机构的评估，并在2021年信息技术应用创新论坛主论坛上正式对外发布。**

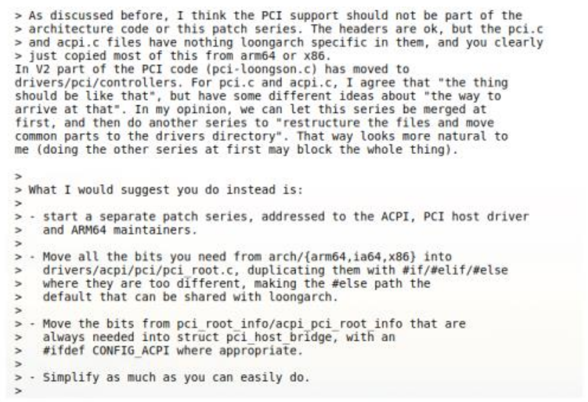
LoongArch架构介绍

- **目前，支持龙芯架构的龙芯3A5000处理器芯片已经流片成功，基于新架构的完整操作系统已经在3A5000计算机上稳定运行。从其它主流指令系统到LoongArch的二进制翻译系统已经可以在3A5000计算机上演示运行基于其它主流指令系统的复杂应用程序。**

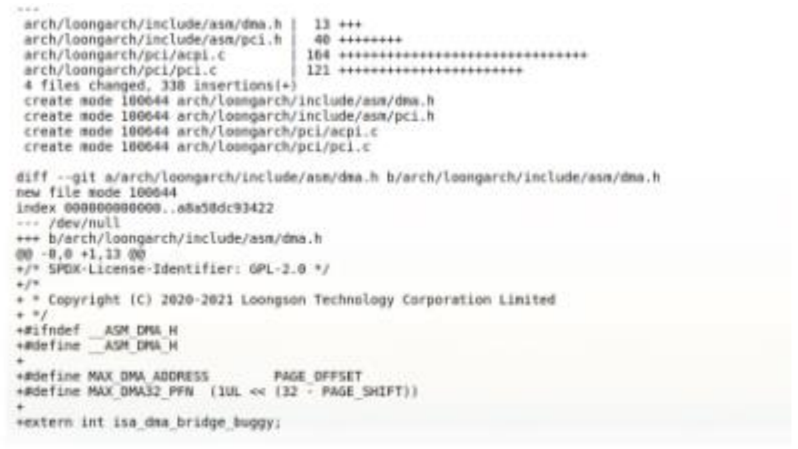
- **龙芯中科从2020年起新研的CPU均支持LoongArch架构。**

LoongArch架构介绍

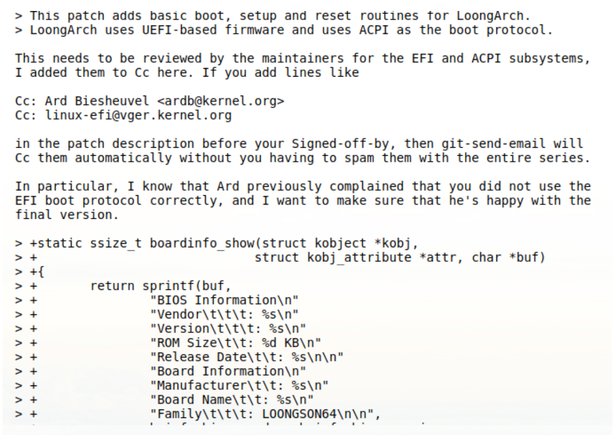
- **龙芯内核团队已经向Linux内核社区提交了LoongArch支持代码，正在积极与内核社区维护者沟通相关问题。**



```
> As discussed before, I think the PCI support should not be part of the
> architecture code or this patch series. The headers are ok, but the pci.c
> and acpi.c files have nothing loongarch specific in them, and you clearly
> just copied most of this from arm64 or x86.
In V2 part of the PCI code (pci-loongson.c) has moved to
drivers/pci/controllers. For pci.c and acpi.c, I agree that "the thing
should be like that", but have some different ideas about "the way to
arrive at that". In my opinion, we can let this series be merged at
first, and then do another series to "restructure the files and move
common parts to the drivers directory". That way looks more natural to
me (doing the other series at first may block the whole thing).

>
> What I would suggest you do instead is:
>
> - start a separate patch series, addressed to the ACPI, PCI host driver
>   and ARM64 maintainers.
>
> - Move all the bits you need from arch/{arm64,ia64,x86} into
>   drivers/acpi/pci/pci_root.c, duplicating them with #if/#elif/#else
>   where they are too different, making the #else path the
>   default that can be shared with loongarch.
>
> - Move the bits from pci_root_info/acpi_pci_root_info that are
>   always needed into struct pci_host_bridge, with an
>   #ifdef CONFIG_ACPI where appropriate.
>
> - Simplify as much as you can easily do.
>
```



```
---
 arch/loongarch/include/asm/dma.h |  13 +++
 arch/loongarch/include/asm/pci.h |  40 ++++++++
 arch/loongarch/pci/acpi.c        | 104 ++++++++++++++++++++++++++++++++
 arch/loongarch/pci/pci.c         | 121 +++++++++++++++++++++++++
 4 files changed, 338 insertions(+)
 create mode 100644 arch/loongarch/include/asm/dma.h
 create mode 100644 arch/loongarch/include/asm/pci.h
 create mode 100644 arch/loongarch/pci/acpi.c
 create mode 100644 arch/loongarch/pci/pci.c

diff --git a/arch/loongarch/include/asm/dma.h b/arch/loongarch/include/asm/dma.h
new file mode 100644
index 000000000000..a8a58dc93422
--- /dev/null
+++ b/arch/loongarch/include/asm/dma.h
@@ -0,0 +1,13 @@
+/* SPDX-License-Identifier: GPL-2.0 */
+/*
+ * Copyright (C) 2020-2021 Loongson Technology Corporation Limited
+ */
+#ifndef __ASM_DMA_H
+#define __ASM_DMA_H
+
+#define MAX_DMA_ADDRESS        PAGE_OFFSET
+#define MAX_DMA32_PFN  (1UL << (32 - PAGE_SHIFT))
+
+extern int isa_dma_bridge_buggy;
```



```
> This patch adds basic boot, setup and reset routines for LoongArch.
> LoongArch uses UEFI-based firmware and uses ACPI as the boot protocol.

This needs to be reviewed by the maintainers for the EFI and ACPI subsystems,
I added them to Cc here. If you add lines like

Cc: Ard Biesheuvel <ardb@kernel.org>
Cc: linux-efi@vger.kernel.org

in the patch description before your Signed-off-by, then git-send-email will
Cc them automatically without you having to spam them with the entire series.

In particular, I know that Ard previously complained that you did not use the
EFI boot protocol correctly, and I want to make sure that he's happy with the
final version.

> +static ssize_t boardinfo_show(struct kobject *kobj,
> +                              struct kobj_attribute *attr, char *buf)
> +{
> +       return sprintf(buf,
> +               "BIOS Information\n"
> +               "Vendor\t\t\t: %s\n"
> +               "Version\t\t\t: %s\n"
> +               "ROM Size\t\t\t: %d KB\n"
> +               "Release Date\t\t: %s\n\n"
> +               "Board Information\n"
> +               "Manufacturer\t\t: %s\n"
> +               "Board Name\t\t: %s\n"
> +               "Family\t\t\t: LOONGSON64\n\n",
```

LoongArch架构介绍

ACPI规范支持

内核适配

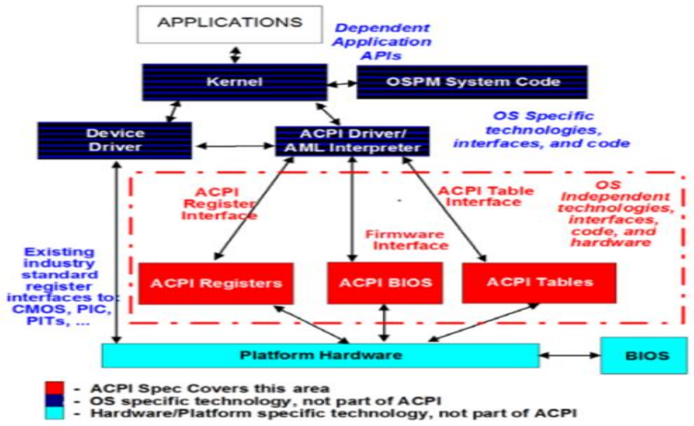
后续工作

问题？

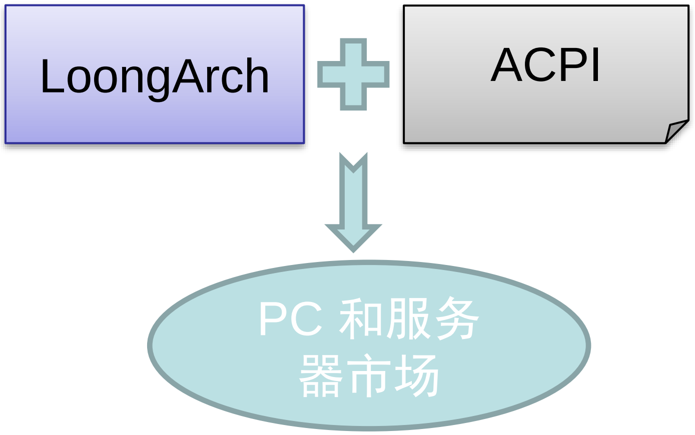
ACPI规范支持

LoongArch支持ACPI规范的必要性

•ACPI作为内核与固件之间的配置规范，在PC和服务器产品上已经成为业界公认的标准，x86、ARM64平台均有成熟的支持。支持ACPI规范，可以使用此规范中完善的机制支持固件与内核的接口，同时保持灵活的可扩展性。

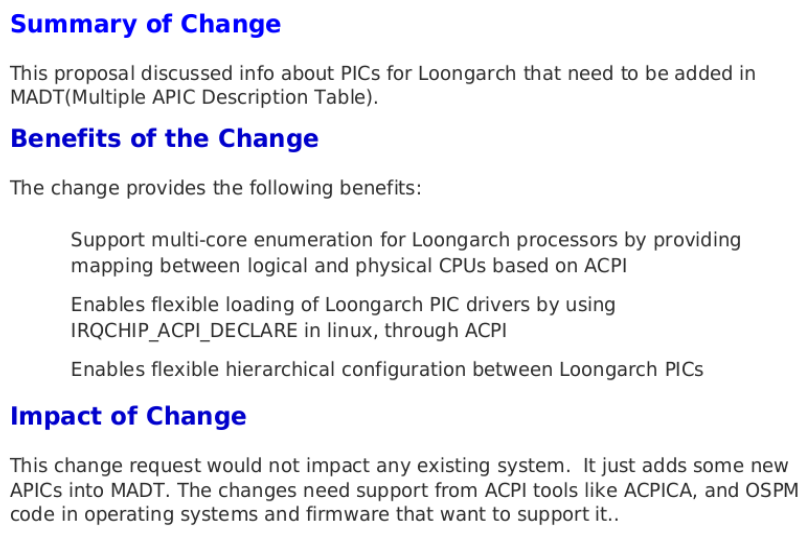

APPLICATIONS
Dependent Application APIs
Kernel
OSPM System Code
Device Driver
ACPI Driver/ AML Interpreter
OS Specific technologies, interfaces, and code
ACPI Register Interface
ACPI Table Interface
OS Independent technologies, interfaces, code, and hardware
Firmware Interface
Existing industry standard register interfaces to: CMOS, PIC, PITs, ...
ACPI Registers
ACPI BIOS
ACPI Tables
Platform Hardware
BIOS
- ACPI Spec Covers this area
- OS specific technology, not part of ACPI
- Hardware/Platform specific technology, not part of ACPI

•龙芯中科作为国产自主处理器的领军企业，充分考虑行业需求，对标成熟的架构，在PC及服务器领域，全面支持ACPI，让熟悉ACPI配置的操作系统厂商、主板设计厂商，高效、快速实现基于龙芯产品的开发与生产，缩短产品开发周期，降低产品维护难度。



ACPI规范支持

- **2021年2月16日，龙芯向ASWG(ACPI Specification Work Group)提交了ECR文件，申请将龙芯中断模型加入ACPI规范的MADT。**



Summary of Change

This proposal discussed info about PICs for Loongarch that need to be added in MADT(Multiple APIC Description Table).

Benefits of the Change

The change provides the following benefits:

Support multi-core enumeration for Loongarch processors by providing mapping between logical and physical CPUs based on ACPI

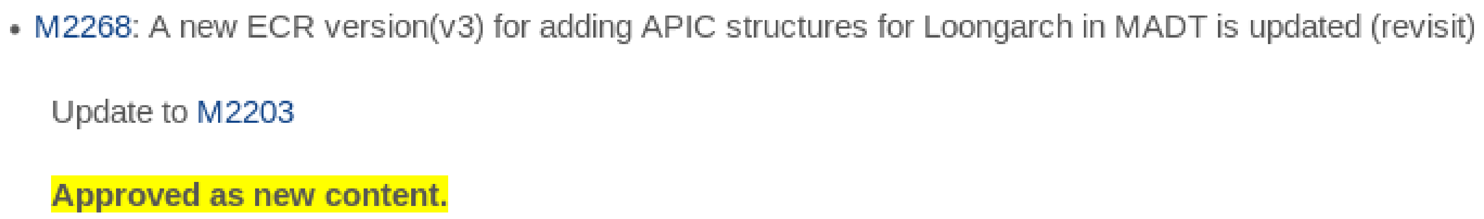
Enables flexible loading of Loongarch PIC drivers by using IRQCHIP_ACPI_DECLARE in linux, through ACPI

Enables flexible hierarchical configuration between Loongarch PICs

Impact of Change

This change request would not impact any existing system. It just adds some new APICs into MADT. The changes need support from ACPI tools like ACPICA, and OSPM code in operating systems and firmware that want to support it..

- **2021年4月1日，龙芯中断模型被正式批准写入ACPI规范，将从下一版开始支持龙芯中断模型。**



- M2268: A new ECR version(v3) for adding APIC structures for Loongarch in MADT is updated (revisit)

Update to M2203

Approved as new content.

ACPI规范支持

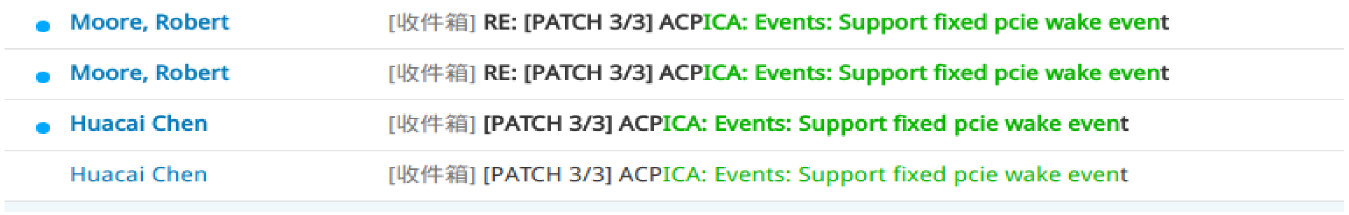
龙芯中科 LOONGSON TECHNOLOGY

龙芯ACPI中断模型结构列表

Value	Description	_MAT forProcess orobject (a)	_MAT for anI/O APIC object (b)	Reference
0x11	Core Programmable Interrupt Controller (CORE PIC)	no	no	5.2.12.20
0x12	Legacy I/O Interrupt Programmable Controller (LIO PIC)	no	no	5.2.12.21
0x13	HyperTransport Programmable Interrupt Controller (HT PIC)	no	no	5.2.12.22
0x14	Extend I/O Programmable Interrupt Controller (EIO PIC)	no	no	5.2.12.23
0x15	MSI Programmable Interrupt Controller (MSI PIC)	no	no	5.2.12.24
0x16	Bridge I/O Programmable Interrupt Controller (BIO PIC)	no	no	5.2.12.25
0x17	Low Pin Count Programmable Interrupt Controller (LPC PIC)	no	no	5.2.12.26

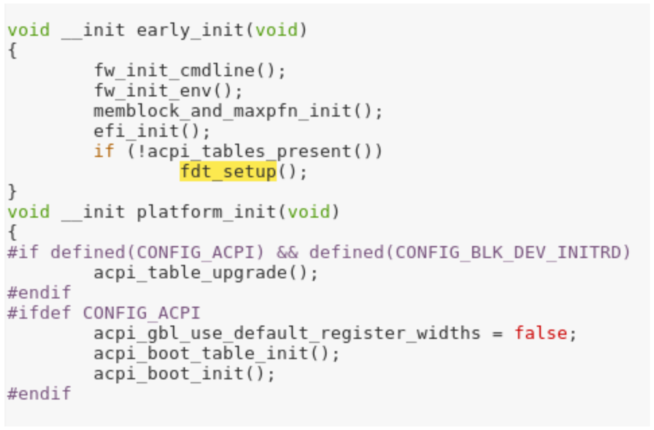
内核适配

- 适配的内核版本
  已经在4.19、5.4、5.10完成了适配
  正在申请提交到上游内核社区



● Moore, Robert	[收件箱] RE: [PATCH 3/3] ACPICA: Events: Support fixed pcie wake event
● Moore, Robert	[收件箱] RE: [PATCH 3/3] ACPICA: Events: Support fixed pcie wake event
● Huacai Chen	[收件箱] [PATCH 3/3] ACPICA: Events: Support fixed pcie wake event
Huacai Chen	[收件箱] [PATCH 3/3] ACPICA: Events: Support fixed pcie wake event

- LoongArch内核兼容FDT与下一版ACPI规范，FDT：应用于嵌入式处理器ACPI：应用于通用PC处理器



```
void __init early_init(void)
{
        fw_init_cmdline();
        fw_init_env();
        memblock_and_maxpfn_init();
        efi_init();
        if (!acpi_tables_present())
                fdt_setup();
}
void __init platform_init(void)
{
#if defined(CONFIG_ACPI) && defined(CONFIG_BLK_DEV_INITRD)
        acpi_table_upgrade();
#endif
#ifdef CONFIG_ACPI
        acpi_gbl_use_default_register_widths = false;
        acpi_boot_table_init();
        acpi_boot_init();
#endif
```

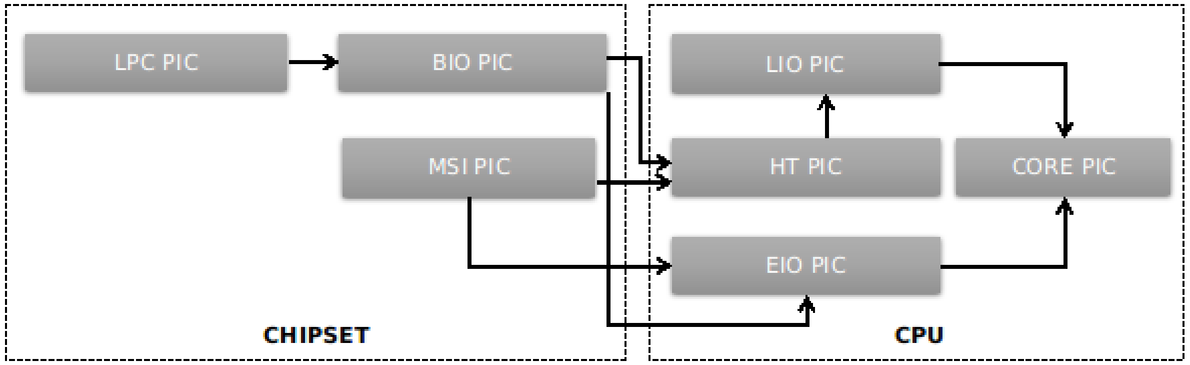
内核适配

- **LoongArch已经适配的ACPI表项**

表	描述	强制
RSDP	Root System Description Pointer	●
XSDT	Extended System Description Table	●
MADT	Multiple APIC Description Table	●
SRAT	System Resource Affinity Table	●
FADT	Fixed ACPI Description Table	●
DSDT	Differentiated System Description Table	●
FACS	Firmware ACPI Control Structure	●
MCFG	PCI Express Memory-mapped Configuration Space base address description table	●
SLIT	System Locality Distance Information Table	
SPCR	Serial Port Console Redirection Table	●

内核适配

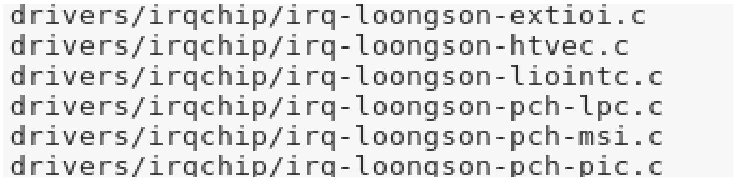
- **LoongArch ACPI中断模型**



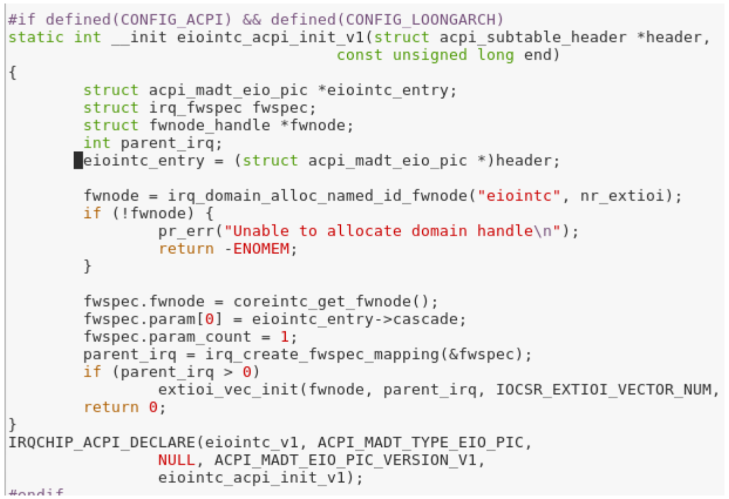
Core Programmable Interrupt Controller (CORE PIC)
Legacy I/O Interrupt Programmable Controller (LIO PIC)
HyperTransport Programmable Interrupt Controller (HT PIC)
Extend I/O Programmable Interrupt Controller (EIO PIC)
MSI Programmable Interrupt Controller (MSI PIC)
Bridge I/O Programmable Interrupt Controller (BIO PIC)
Low Pin Count Programmable Interrupt Controller (LPC PIC)

内核适配

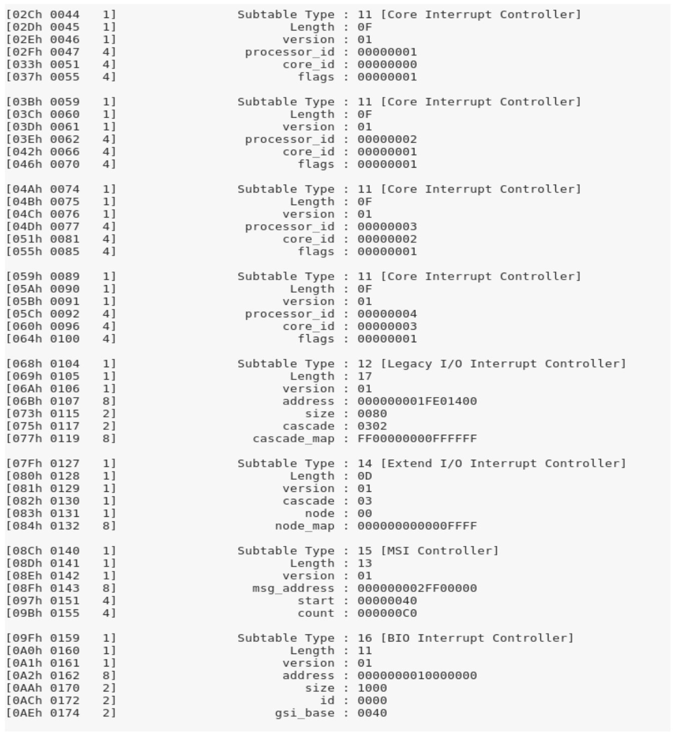
- **LoongArch ACPI中断模型**



```
drivers/irqchip/irq-loongson-extioi.c
drivers/irqchip/irq-loongson-htvec.c
drivers/irqchip/irq-loongson-liointc.c
drivers/irqchip/irq-loongson-pch-lpc.c
drivers/irqchip/irq-loongson-pch-msi.c
drivers/irqchip/irq-loongson-pch-pic.c
```



```c
#if defined(CONFIG_ACPI) && defined(CONFIG_LOONGARCH)
static int __init eiointc_acpi_init_v1(struct acpi_subtable_header *header,
                                   const unsigned long end)
{
        struct acpi_madt_eio_pic *eiointc_entry;
        struct irq_fwspec fwspec;
        struct fwnode_handle *fwnode;
        int parent_irq;
        eiointc_entry = (struct acpi_madt_eio_pic *)header;

        fwnode = irq_domain_alloc_named_id_fwnode("eiointc", nr_extioi);
        if (!fwnode) {
                pr_err("Unable to allocate domain handle\n");
                return -ENOMEM;
        }

        fwspec.fwnode = coreintc_get_fwnode();
        fwspec.param[0] = eiointc_entry->cascade;
        fwspec.param_count = 1;
        parent_irq = irq_create_fwspec_mapping(&fwspec);
        if (parent_irq > 0)
                extioi_vec_init(fwnode, parent_irq, IOCSR_EXTIOI_VECTOR_NUM,
        return 0;
}
IRQCHIP_ACPI_DECLARE(eiointc_v1, ACPI_MADT_TYPE_EIO_PIC,
                NULL, ACPI_MADT_EIO_PIC_VERSION_V1,
                eiointc_acpi_init_v1);
#endif
```

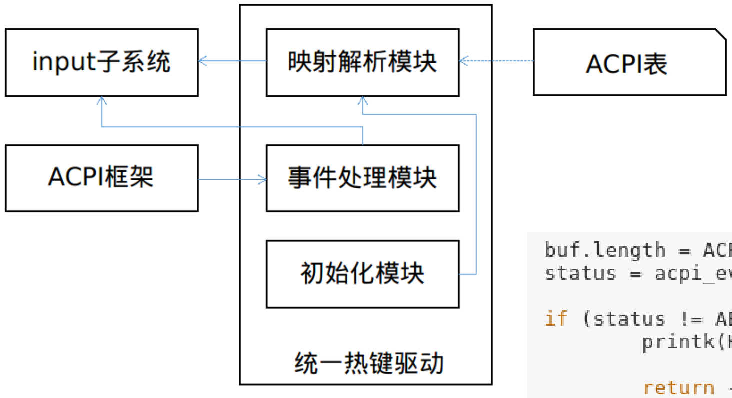


```
[02Ch 0044   1]               Subtable Type : 11 [Core Interrupt Controller]
[02Dh 0045   1]                      Length : 0F
[02Eh 0046   1]                     version : 01
[02Fh 0047   4]                processor_id : 00000001
[033h 0051   4]                     core_id : 00000000
[037h 0055   4]                       flags : 00000001

[03Bh 0059   1]               Subtable Type : 11 [Core Interrupt Controller]
[03Ch 0060   1]                      Length : 0F
[03Dh 0061   1]                     version : 01
[03Eh 0062   4]                processor_id : 00000002
[042h 0066   4]                     core_id : 00000001
[046h 0070   4]                       flags : 00000001

[04Ah 0074   1]               Subtable Type : 11 [Core Interrupt Controller]
[04Bh 0075   1]                      Length : 0F
[04Ch 0076   1]                     version : 01
[04Dh 0077   4]                processor_id : 00000003
[051h 0081   4]                     core_id : 00000002
[055h 0085   4]                       flags : 00000001

[059h 0089   1]               Subtable Type : 11 [Core Interrupt Controller]
[05Ah 0090   1]                      Length : 0F
[05Bh 0091   1]                     version : 01
[05Ch 0092   4]                processor_id : 00000004
[060h 0096   4]                     core_id : 00000003
[064h 0100   4]                       flags : 00000001

[068h 0104   1]               Subtable Type : 12 [Legacy I/O Interrupt Controller]
[069h 0105   1]                      Length : 17
[06Ah 0106   1]                     version : 01
[06Bh 0107   8]                     address : 000000001FE01400
[073h 0115   2]                        size : 0080
[075h 0117   2]                     cascade : 0302
[077h 0119   8]                 cascade_map : FF00000000FFFFFF

[07Fh 0127   1]               Subtable Type : 14 [Extend I/O Interrupt Controller]
[080h 0128   1]                      Length : 0D
[081h 0129   1]                     version : 01
[082h 0130   1]                     cascade : 03
[083h 0131   1]                        node : 00
[084h 0132   8]                    node_map : 000000000000FFFF

[08Ch 0140   1]               Subtable Type : 15 [MSI Controller]
[08Dh 0141   1]                      Length : 13
[08Eh 0142   1]                     version : 01
[08Fh 0143   8]                 msg_address : 000000002FF00000
[097h 0151   4]                       start : 00000040
[09Bh 0155   4]                       count : 000000C0

[09Fh 0159   1]               Subtable Type : 16 [BIO Interrupt Controller]
[0A0h 0160   1]                      Length : 11
[0A1h 0161   1]                     version : 01
[0A2h 0162   8]                     address : 0000000010000000
[0AAh 0170   2]                        size : 1000
[0ACh 0172   2]                          id : 0000
[0AEh 0174   2]                    gsi_base : 0040
```

内核适配

- **统一热键模型**

龙芯平台针对热键布局差异化问题，设计统一的热键驱动
通过ACPI配置实现不同厂商热键的差异化设计

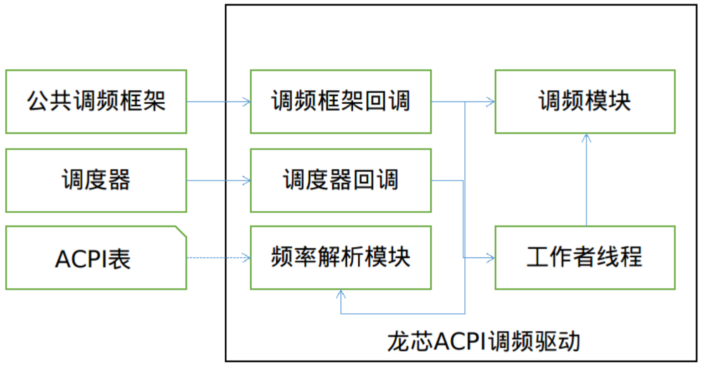


```
buf.length = ACPI_ALLOCATE_BUFFER;
status = acpi_evaluate_object_typed(hkey_handle,
                METHOD_NAME__KMAP, NULL, &buf, ACPI_TYPE_PACKAGE);
if (status != AE_OK) {
        printk(KERN_ERR ": ACPI exception: %s\n",
                        acpi_format_exception(status));
        return -1;
}
pack = buf.pointer;
for (index = 0; index < pack->package.count; index++) {
        union acpi_object *sub_pack = &pack->package.elements[index];
        union acpi_object *element = &sub_pack->package.elements[0];

        hotkey_keycode_map[index].type = element->integer.value;
        element = &sub_pack->package.elements[1];
        hotkey_keycode_map[index].code = element->integer.value;
        element = &sub_pack->package.elements[2];
        hotkey_keycode_map[index].keycode = element->integer.value;
}
return 0;
```

内核适配

- **CPU功耗管理**

通过_PSS通知内核频率信息

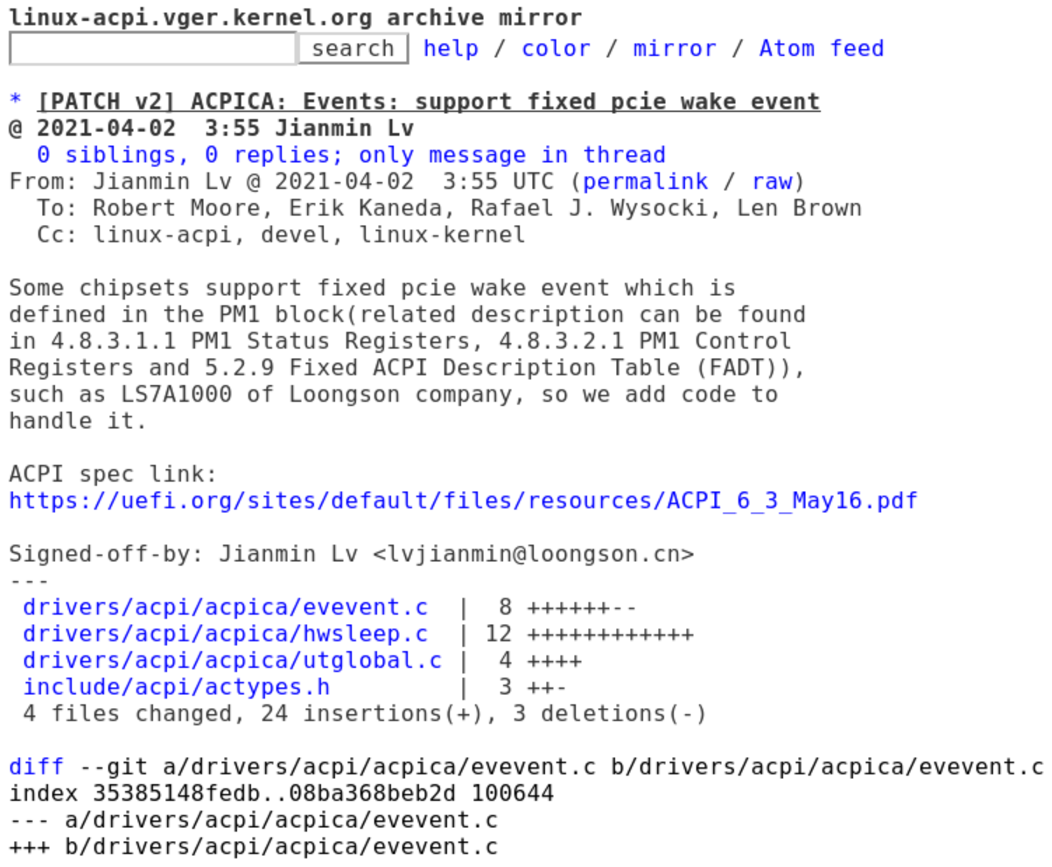


```
/* table init */
for (i = 0; i < perf->state_count; i++) {
        freq_table[i].driver_data = (perf->states[i].control & LOONGSON_BOOST_FREQ_MASK) >> 8;
        if (freq_table[i].driver_data)
                freq_table[i].flags |= CPUFREQ_BOOST_FREQ;
        freq_table[i].frequency =
                perf->states[i].core_frequency * 1000;
}
freq_table[i].frequency = CPUFREQ_TABLE_END;
policy->freq_table = freq_table;
perf->state = 0;
```

内核适配

- **事件编程模型**

ACPI_EVENT_PCIE_WAKE支持

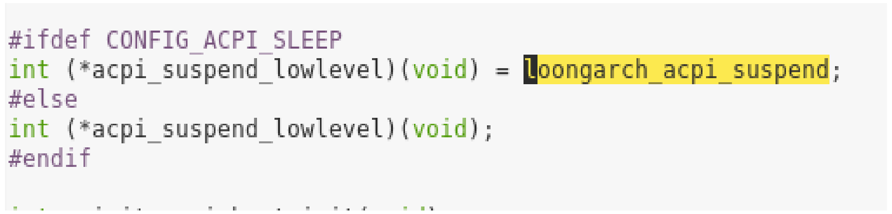


```
linux-acpi.vger.kernel.org archive mirror
                     search  help / color / mirror / Atom feed

* [PATCH v2] ACPICA: Events: support fixed pcie wake event
@ 2021-04-02  3:55 Jianmin Lv
  0 siblings, 0 replies; only message in thread
From: Jianmin Lv @ 2021-04-02  3:55 UTC (permalink / raw)
  To: Robert Moore, Erik Kaneda, Rafael J. Wysocki, Len Brown
  Cc: linux-acpi, devel, linux-kernel

Some chipsets support fixed pcie wake event which is
defined in the PM1 block(related description can be found
in 4.8.3.1.1 PM1 Status Registers, 4.8.3.2.1 PM1 Control
Registers and 5.2.9 Fixed ACPI Description Table (FADT)),
such as LS7A1000 of Loongson company, so we add code to
handle it.

ACPI spec link:
https://uefi.org/sites/default/files/resources/ACPI_6_3_May16.pdf

Signed-off-by: Jianmin Lv <lvjianmin@loongson.cn>
---
 drivers/acpi/acpica/evevent.c   |  8 ++++++--
 drivers/acpi/acpica/hwsleep.c   | 12 ++++++++++++
 drivers/acpi/acpica/utglobal.c  |  4 ++++
 include/acpi/actypes.h          |  3 ++-
 4 files changed, 24 insertions(+), 3 deletions(-)

diff --git a/drivers/acpi/acpica/evevent.c b/drivers/acpi/acpica/evevent.c
index 35385148fedb..08ba368beb2d 100644
--- a/drivers/acpi/acpica/evevent.c
+++ b/drivers/acpi/acpica/evevent.c
```

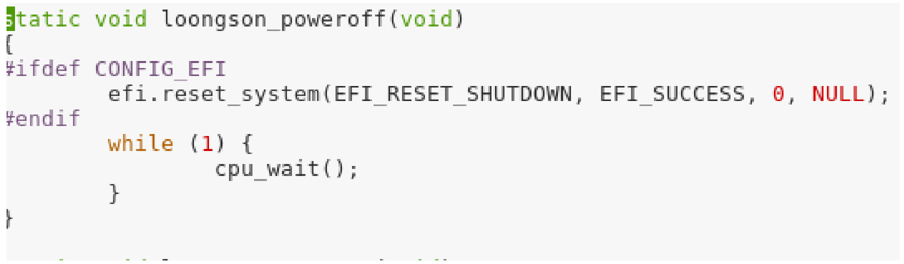
内核适配

- **系统电源管理**

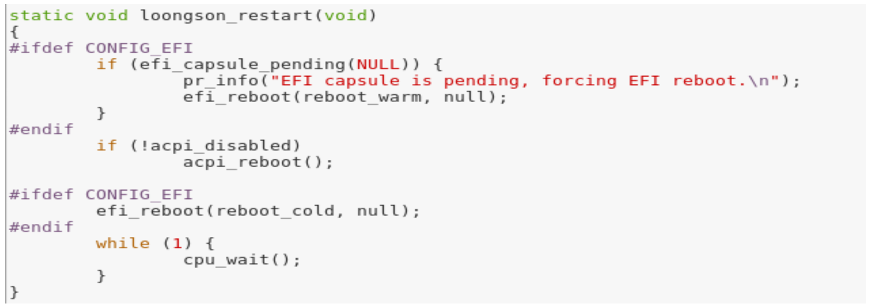
S3/S4/S5



```
#ifdef CONFIG_ACPI_SLEEP
int (*acpi_suspend_lowlevel)(void) = loongarch_acpi_suspend;
#else
int (*acpi_suspend_lowlevel)(void);
#endif
```



```
static void loongson_poweroff(void)
{
#ifdef CONFIG_EFI
	efi.reset_system(EFI_RESET_SHUTDOWN, EFI_SUCCESS, 0, NULL);
#endif
	while (1) {
		cpu_wait();
	}
}
```



```
static void loongson_restart(void)
{
#ifdef CONFIG_EFI
	if (efi_capsule_pending(NULL)) {
		pr_info("EFI capsule is pending, forcing EFI reboot.\n");
		efi_reboot(reboot_warm, null);
	}
#endif
	if (!acpi_disabled)
		acpi_reboot();

#ifdef CONFIG_EFI
	efi_reboot(reboot_cold, null);
#endif
	while (1) {
		cpu_wait();
	}
}
```

后续工作

- PPTT（Processor Properties Topology Table）适配
- 完善对新一代龙芯7A芯片组ACPI功能支持
- 优化处理器功耗调节方式
- 完善统一热键模型
- 完善龙芯平台EC规范

谢谢！

Thanks!