FreeBSD kernel i386 PCI device code Reference Manual

Generated by Doxygen 1.4.7

Sat Feb 24 11:11:36 2007

Contents

Chapter 1

FreeBSD kernel i386 PCI device code Main Page

IMPORTANT: This API documentation may contain both functions which are public and functions that are for internal use only. Since we have not reviewed every part of the documentation yet, *some internal functions are not marked as such*. Until we finish reviewing the API documentation and add appropriate comments to functions which are only for internal use, you should take this into account. In case you want to use a function of this kernel subsystem in another kernel subsystem you should search for precedence of use outside this subsystem. If the function is not used outside this subsystem you should ask on the mailinglists about it, else you risk breaking something.

Chapter 2

FreeBSD kernel i386 PCI device code Directory Hierarchy

2.1 FreeBSD kernel i386 PCI device code Directories

This directory hierarchy is sorted roughly, but not completely, alphabetically:

Chapter 3

FreeBSD kernel i386 PCI device code Data Structure Index

3.1 FreeBSD kernel i386 PCI device code Data Structures

Here are the data structures with brief descriptions:

Chapter 4

FreeBSD kernel i386 PCI device code File Index

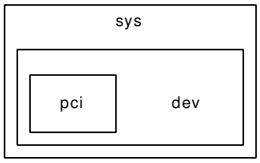
4.1 FreeBSD kernel i386 PCI device code File List

Here is a list of all files with brief descriptions:

Chapter 5

FreeBSD kernel i386 PCI device code Directory Documentation

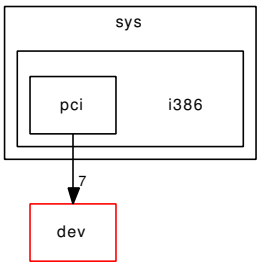
5.1 /usr/src/sys/dev/ Directory Reference



Directories

- directory pci

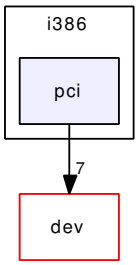
5.2 /usr/src/sys/i386/ Directory Reference



Directories

- directory pci

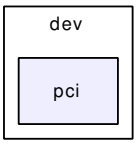
5.3 /usr/src/sys/i386/pci/ Directory Reference



Files

- file pci_bus.c
- file pci_cfgreg.c
- file pci_pir.c

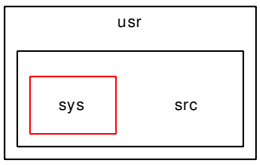
5.4 /usr/src/sys/dev/pci/ Directory Reference



Files

- file eisa_pci.c
- file fixup_pci.c
- file hostb_pci.c
- file ignore_pci.c
- file isa_pci.c
- file pci.c
- file pci_if.m
- file pci_pci.c
- file pci_private.h
- file pci_user.c
- file pcib_if.m
- file pcib_private.h
- file pcireg.h
- file pcivar.h
- file vga_pci.c

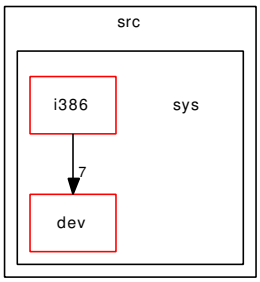
5.5 /usr/src/ Directory Reference



Directories

- directory sys

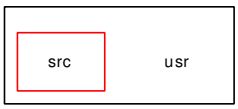
5.6 /usr/src/sys/ Directory Reference



Directories

- directory dev
- directory i386

5.7 /usr/ Directory Reference



Directories

- directory src

Chapter 6

FreeBSD kernel i386 PCI device code Data Structure Documentation

6.1 pci_dev_lookup Struct Reference

Data Fields

- uint8_t link
- int bus
- int device
- int pin

6.1.1 Detailed Description

Definition at line 72 of file pci_pir.c.

6.1.2 Field Documentation

6.1.2.1 int pci_dev_lookup::bus

Definition at line 74 of file pci_pir.c.

Referenced by pir_resume(), and pir_resume_find_device().

6.1.2.2 int pci_dev_lookup::device

Definition at line 75 of file pci_pir.c.

Referenced by pir_resume(), and pir_resume_find_device().

6.1.2.3 uint8_t pci_dev_lookup::link

Definition at line 73 of file pci_pir.c.

Referenced by pir_resume(), and pir_resume_find_device().

6.1.2.4 int pci_dev_lookup::pin

Definition at line 76 of file pci_pir.c.

Referenced by pir_resume(), and pir_resume_find_device().

The documentation for this struct was generated from the following file:

- /usr/src/sys/i386/pci/pci_pir.c

6.2 pci_link Struct Reference

6.2.1 Detailed Description

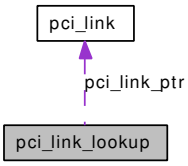
Definition at line 56 of file pci_pir.c.

The documentation for this struct was generated from the following file:

- /usr/src/sys/i386/pci/pci_pir.c

6.3 pci_link_lookup Struct Reference

Collaboration diagram for pci_link_lookup:



Data Fields

- pci_link ** pci_link_ptr
- int bus
- int device
- int pin

6.3.1 Detailed Description

Definition at line 65 of file pci_pir.c.

6.3.2 Field Documentation

6.3.2.1 int pci_link_lookup::bus

Definition at line 67 of file pci_pir.c.

Referenced by pci_pir_find_link_handler(), and pci_pir_route_interrupt().

6.3.2.2 int pci_link_lookup::device

Definition at line 68 of file pci_pir.c.

Referenced by pci_pir_find_link_handler(), and pci_pir_route_interrupt().

6.3.2.3 struct pci_link** pci_link_lookup::pci_link_ptr

Definition at line 66 of file pci_pir.c.

Referenced by pci_pir_find_link_handler(), and pci_pir_route_interrupt().

6.3.2.4 int pci_link_lookup::pin

Definition at line 69 of file pci_pir.c.

Referenced by pci_pir_find_link_handler(), and pci_pir_route_interrupt().

The documentation for this struct was generated from the following file:

- /usr/src/sys/i386/pci/pci_pir.c

6.4 pci_quirk Struct Reference

Data Fields

- uint32_t devid
- int type
- int arg1
- int arg2

6.4.1 Detailed Description

Definition at line 168 of file pci.c.

6.4.2 Field Documentation

6.4.2.1 int pci_quirk::arg1

Definition at line 173 of file pci.c.

Referenced by pci_add_resources().

6.4.2.2 int pci_quirk::arg2

Definition at line 174 of file pci.c.

6.4.2.3 uint32_t pci_quirk::devid

Definition at line 169 of file pci.c.

Referenced by pci_add_resources(), and pci_msi_device_blacklisted().

6.4.2.4 int pci_quirk::type

Definition at line 170 of file pci.c.

Referenced by pci_add_resources(), and pci_msi_device_blacklisted().

The documentation for this struct was generated from the following file:

- /usr/src/sys/dev/pci/pci.c

6.5 pcib_softc Struct Reference

`#include <pcib_private.h>`

Data Fields

- device_t dev
- uint32_t flags
- uint16_t command
- uint8_t secbus
- uint8_t subbus
- pci_addr_t pmembase
- pci_addr_t pmemlimit
- pci_addr_t membase
- pci_addr_t memlimit
- uint32_t iobase
- uint32_t iolimit
- uint16_t secstat
- uint16_t bridgectl
- uint8_t seclat

6.5.1 Detailed Description

Definition at line 44 of file pcib_private.h.

6.5.2 Field Documentation

6.5.2.1 uint16_t pcib_softc::bridgectl

Definition at line 60 of file pcib_private.h.

Referenced by pcib_alloc_resource(), and pcib_attach_common().

6.5.2.2 uint16_t pcib_softc::command

Definition at line 50 of file pcib_private.h.

Referenced by pcib_attach_common().

6.5.2.3 device_t pcib_softc::dev

Definition at line 46 of file pcib_private.h.

Referenced by pcib_attach_common().

6.5.2.4 uint32_t pcib_softc::flags

Definition at line 47 of file pcib_private.h.

Referenced by pcib_alloc_msi(), pcib_alloc_msix(), pcib_alloc_resource(), and pcib_attach_common().

6.5.2.5 uint32_t pcib_softc::iobase

Definition at line 57 of file pcib_private.h.

Referenced by pcib_alloc_resource(), pcib_attach_common(), and pcib_is_io_open().

6.5.2.6 uint32_t pcib_softc::iolimit

Definition at line 58 of file pcib_private.h.

Referenced by pcib_alloc_resource(), pcib_attach_common(), and pcib_is_io_open().

6.5.2.7 pci_addr_t pcib_softc::membase

Definition at line 55 of file pcib_private.h.

Referenced by pcib_alloc_resource(), pcib_attach_common(), and pcib_is_nonprefetch_open().

6.5.2.8 pci_addr_t pcib_softc::memlimit

Definition at line 56 of file pcib_private.h.

Referenced by pcib_alloc_resource(), pcib_attach_common(), and pcib_is_nonprefetch_open().

6.5.2.9 pci_addr_t pcib_softc::pmembase

Definition at line 53 of file pcib_private.h.

Referenced by pcib_alloc_resource(), pcib_attach_common(), and pcib_is_prefetch_open().

6.5.2.10 pci_addr_t pcib_softc::pmemlimit

Definition at line 54 of file pcib_private.h.

Referenced by pcib_alloc_resource(), pcib_attach_common(), and pcib_is_prefetch_open().

6.5.2.11 uint8_t pcib_softc::secbus

Definition at line 51 of file pcib_private.h.

Referenced by pcib_attach(), pcib_attach_common(), pcib_read_ivar(), and pcib_write_ivar().

6.5.2.12 uint8_t pcib_softc::seclat

Definition at line 61 of file pcib_private.h.

Referenced by pcib_attach_common().

6.5.2.13 uint16_t pcib_softc::secstat

Definition at line 59 of file pcib_private.h.

Referenced by pcib_attach_common().

6.5.2.14 uint8_t pcib_softc::subbus

Definition at line 52 of file pcib_private.h.

Referenced by pcib_attach_common().

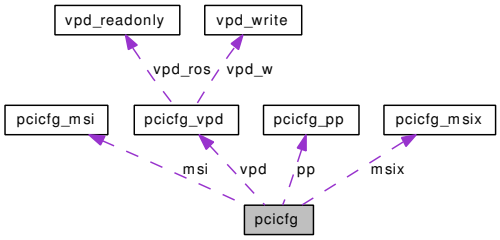
The documentation for this struct was generated from the following file:

- /usr/src/sys/dev/pci/pcib_private.h

6.6 pcicfg Struct Reference

`#include <pcivar.h>`

Collaboration diagram for pcicfg:



Data Fields

- device ∗ dev
- uint32_t bar [PCI_MAXMAPS_0]
- uint32_t bios
- uint16_t subvendor
- uint16_t subdevice
- uint16_t vendor
- uint16_t device
- uint16_t cmdreg
- uint16_t statreg
- uint8_t baseclass
- uint8_t subclass
- uint8_t progif
- uint8_t revid
- uint8_t hdrtype
- uint8_t cachelnsz
- uint8_t intpin
- uint8_t intline
- uint8_t mingnt
- uint8_t maxlat
- uint8_t lattimer
- uint8_t mfdev
- uint8_t nummaps
- uint8_t bus
- uint8_t slot
- uint8_t func
- pcicfg_pp pp
- pcicfg_vpd vpd
- pcicfg_msi msi
- pcicfg_msix msix

6.6.1 Detailed Description

Definition at line 102 of file pcivar.h.

6.6.2 Field Documentation

6.6.2.1 uint32_t pcicfg::bar[PCI_MAXMAPS_0]

Definition at line 105 of file pcivar.h.

6.6.2.2 uint8_t pcicfg::baseclass

Definition at line 116 of file pcivar.h.

Referenced by pci_child_pnpinfo_str_method(), pci_fixancient(), pci_print_verbose(), pci_read_device(), and pci_read_ivar().

6.6.2.3 uint32_t pcicfg::bios

Definition at line 106 of file pcivar.h.

6.6.2.4 uint8_t pcicfg::bus

Definition at line 133 of file pcivar.h.

Referenced by pci_add_resources(), pci_assign_interrupt(), pci_print_verbose(), pci_read_config_method(), pci_read_device(), pci_read_extcap(), pci_read_ivar(), pci_read_vpd(), and pci_write_config_method().

6.6.2.5 uint8_t pcicfg::cachelnsz

Definition at line 122 of file pcivar.h.

Referenced by pci_print_verbose(), pci_read_device(), and pci_read_ivar().

6.6.2.6 uint16_t pcicfg::cmdreg

Definition at line 113 of file pcivar.h.

Referenced by pci_print_verbose(), pci_read_device(), and pci_read_ivar().

6.6.2.7 struct device* pcicfg::dev

Definition at line 103 of file pcivar.h.

Referenced by pci_print_verbose().

6.6.2.8 uint16_t pcicfg::device

Definition at line 111 of file pcivar.h.

Referenced by pci_add_resources(), pci_child_pnpinfo_str_method(), pci_print_verbose(), pci_read_device(), and pci_read_ivar().

6.6.2.9 uint8_t pcicfg::func

Definition at line 135 of file pcivar.h.

Referenced by pci_add_resources(), pci_print_verbose(), pci_read_config_method(), pci_read_device(), pci_read_extcap(), pci_read_ivar(), pci_read_vpd(), and pci_write_config_method().

6.6.2.10 uint8_t pcicfg::hdrtype

Definition at line 121 of file pcivar.h.

Referenced by pci_find_extcap_method(), pci_fixancient(), pci_hdrtypedata(), pci_print_verbose(), pci_read_device(), and pci_read_extcap().

6.6.2.11 uint8_t pcicfg::intline

Definition at line 124 of file pcivar.h.

Referenced by pci_add_resources(), pci_alloc_resource(), pci_assign_interrupt(), pci_print_verbose(), pci_read_device(), and pci_read_ivar().

6.6.2.12 uint8_t pcicfg::intpin

Definition at line 123 of file pcivar.h.

Referenced by pci_add_resources(), pci_alloc_resource(), pci_assign_interrupt(), pci_assign_interrupt_method(), pci_print_verbose(), pci_read_device(), and pci_read_ivar().

6.6.2.13 uint8_t pcicfg::lattimer

Definition at line 128 of file pcivar.h.

Referenced by pci_print_verbose(), pci_read_device(), and pci_read_ivar().

6.6.2.14 uint8_t pcicfg::maxlat

Definition at line 127 of file pcivar.h.

Referenced by pci_print_verbose(), pci_read_device(), and pci_read_ivar().

6.6.2.15 uint8_t pcicfg::mfdev

Definition at line 130 of file pcivar.h.

Referenced by pci_print_verbose(), and pci_read_device().

6.6.2.16 uint8_t pcicfg::mingnt

Definition at line 126 of file pcivar.h.

Referenced by pci_print_verbose(), pci_read_device(), and pci_read_ivar().

6.6.2.17 struct pcicfg_msi pcicfg::msi

Definition at line 139 of file pcivar.h.

Referenced by pci_alloc_msi_method(), pci_alloc_msix_method(), pci_alloc_resource(), pci_enable_msi(), pci_msi_count_method(), pci_print_verbose(), pci_read_extcap(), pci_release_msi_method(), and pci_resume_msi().

6.6.2.18 struct pcicfg_msix pcicfg::msix

Definition at line 140 of file pcivar.h.

Referenced by pci_alloc_msi_method(), pci_alloc_msix_method(), pci_alloc_resource(), pci_enable_msix(), pci_mask_msix(), pci_msix_count_method(), pci_pending_msix(), pci_print_verbose(), pci_read_extcap(), pci_release_msix(), pci_remap_msix_method(), and pci_unmask_msix().

6.6.2.19 uint8_t pcicfg::nummaps

Definition at line 131 of file pcivar.h.

Referenced by pci_add_resources(), pci_alloc_resource(), and pci_hdrtypedata().

6.6.2.20 struct pcicfg_pp pcicfg::pp

Definition at line 137 of file pcivar.h.

Referenced by pci_get_powerstate_method(), pci_print_verbose(), pci_read_extcap(), and pci_set_powerstate_method().

6.6.2.21 uint8_t pcicfg::progif

Definition at line 118 of file pcivar.h.

Referenced by pci_child_pnpinfo_str_method(), pci_print_verbose(), pci_read_device(), and pci_read_ivar().

6.6.2.22 uint8_t pcicfg::revid

Definition at line 119 of file pcivar.h.

Referenced by pci_print_verbose(), pci_read_device(), and pci_read_ivar().

6.6.2.23 uint8_t pcicfg::slot

Definition at line 134 of file pcivar.h.

Referenced by pci_add_resources(), pci_assign_interrupt(), pci_print_verbose(), pci_read_config_method(), pci_read_device(), pci_read_extcap(), pci_read_ivar(), pci_read_vpd(), and pci_write_config_method().

6.6.2.24 uint16_t pcicfg::statreg

Definition at line 114 of file pcivar.h.

Referenced by pci_print_verbose(), and pci_read_device().

6.6.2.25 uint8_t pcicfg::subclass

Definition at line 117 of file pcivar.h.

Referenced by pci_child_pnpinfo_str_method(), pci_fixancient(), pci_print_verbose(), pci_read_device(), and pci_read_ivar().

6.6.2.26 uint16_t pcicfg::subdevice

Definition at line 109 of file pcivar.h.

Referenced by pci_child_pnpinfo_str_method(), pci_hdrtypedata(), pci_read_device(), pci_read_extcap(), and pci_read_ivar().

6.6.2.27 uint16_t pcicfg::subvendor

Definition at line 108 of file pcivar.h.

Referenced by pci_child_pnpinfo_str_method(), pci_hdrtypedata(), pci_read_device(), pci_read_extcap(), and pci_read_ivar().

6.6.2.28 uint16_t pcicfg::vendor

Definition at line 110 of file pcivar.h.

Referenced by pci_add_resources(), pci_child_pnpinfo_str_method(), pci_print_verbose(), pci_read_device(), and pci_read_ivar().

6.6.2.29 struct pcicfg_vpd pcicfg::vpd

Definition at line 138 of file pcivar.h.

Referenced by pci_get_vpd_ident_method(), pci_get_vpd_readonly_method(), pci_print_verbose(), pci_read_extcap(), pci_read_vpd(), and pci_read_vpd_reg().

The documentation for this struct was generated from the following file:

- /usr/src/sys/dev/pci/pcivar.h

6.7 pcicfg_msi Struct Reference

`#include <pcivar.h>`

Data Fields

- uint16_t msi_ctrl
- uint8_t msi_location
- uint8_t msi_msgnum
- int msi_alloc
- uint64_t msi_addr
- uint16_t msi_data

6.7.1 Detailed Description

Definition at line 78 of file pcivar.h.

6.7.2 Field Documentation

6.7.2.1 uint64_t pcicfg_msi::msi_addr

Definition at line 83 of file pcivar.h.

Referenced by pci_enable_msi(), and pci_resume_msi().

6.7.2.2 int pcicfg_msi::msi_alloc

Definition at line 82 of file pcivar.h.

Referenced by pci_alloc_msi_method(), pci_alloc_msix_method(), pci_alloc_resource(), and pci_release_msi_method().

6.7.2.3 uint16_t pcicfg_msi::msi_ctrl

Definition at line 79 of file pcivar.h.

Referenced by pci_alloc_msi_method(), pci_enable_msi(), pci_print_verbose(), pci_read_extcap(), pci_release_msi_method(), and pci_resume_msi().

6.7.2.4 uint16_t pcicfg_msi::msi_data

Definition at line 84 of file pcivar.h.

Referenced by pci_enable_msi(), and pci_resume_msi().

6.7.2.5 uint8_t pcicfg_msi::msi_location

Definition at line 80 of file pcivar.h.

Referenced by pci_alloc_msi_method(), pci_enable_msi(), pci_msi_count_method(), pci_print_verbose(), pci_read_extcap(), pci_release_msi_method(), and pci_resume_msi().

6.7.2.6 uint8_t pcicfg_msi::msi_msgnum

Definition at line 81 of file pcivar.h.

Referenced by pci_alloc_msi_method(), pci_msi_count_method(), pci_print_verbose(), and pci_read_extcap().

The documentation for this struct was generated from the following file:

- /usr/src/sys/dev/pci/pcivar.h

6.8 pcicfg_msix Struct Reference

`#include <pcivar.h>`

Data Fields

- uint16_t msix_ctrl
- uint8_t msix_location
- uint16_t msix_msgnum
- int msix_alloc
- uint8_t msix_table_bar
- uint8_t msix_pba_bar
- uint32_t msix_table_offset
- uint32_t msix_pba_offset
- resource * msix_table_res
- resource * msix_pba_res

6.8.1 Detailed Description

Definition at line 88 of file pcivar.h.

6.8.2 Field Documentation

6.8.2.1 int pcicfg_msix::msix_alloc

Definition at line 92 of file pcivar.h.

Referenced by pci_alloc_msi_method(), pci_alloc_msix_method(), pci_alloc_resource(), pci_enable_msix(), pci_pending_msix(), pci_release_msix(), pci_remap_msix_method(), and pci_unmask_msix().

6.8.2.2 uint16_t pcicfg_msix::msix_ctrl

Definition at line 89 of file pcivar.h.

Referenced by pci_read_extcap(), and pci_release_msix().

6.8.2.3 uint8_t pcicfg_msix::msix_location

Definition at line 90 of file pcivar.h.

Referenced by pci_alloc_msix_method(), pci_msix_count_method(), pci_print_verbose(), pci_read_extcap(), and pci_release_msix().

6.8.2.4 uint16_t pcicfg_msix::msix_msgnum

Definition at line 91 of file pcivar.h.

Referenced by pci_alloc_msix_method(), pci_mask_msix(), pci_msix_count_method(), pci_print_verbose(), pci_read_extcap(), and pci_remap_msix_method().

6.8.2.5 uint8_t pcicfg_msix::msix_pba_bar

Definition at line 94 of file pcivar.h.

Referenced by pci_alloc_msix_method(), pci_print_verbose(), and pci_read_extcap().

6.8.2.6 uint32_t pcicfg_msix::msix_pba_offset

Definition at line 96 of file pcivar.h.

Referenced by pci_pending_msix(), and pci_read_extcap().

6.8.2.7 struct resource* pcicfg_msix::msix_pba_res

Definition at line 98 of file pcivar.h.

Referenced by pci_alloc_msix_method(), and pci_pending_msix().

6.8.2.8 uint8_t pcicfg_msix::msix_table_bar

Definition at line 93 of file pcivar.h.

Referenced by pci_alloc_msix_method(), pci_print_verbose(), and pci_read_extcap().

6.8.2.9 uint32_t pcicfg_msix::msix_table_offset

Definition at line 95 of file pcivar.h.

Referenced by pci_enable_msix(), pci_mask_msix(), pci_read_extcap(), and pci_unmask_msix().

6.8.2.10 struct resource* pcicfg_msix::msix_table_res

Definition at line 97 of file pcivar.h.

Referenced by pci_alloc_msix_method(), pci_enable_msix(), pci_mask_msix(), and pci_unmask_msix().

The documentation for this struct was generated from the following file:

- /usr/src/sys/dev/pci/pcivar.h

6.9 pcicfg_pp Struct Reference

`#include <pcivar.h>`

Data Fields

- uint16_t pp_cap
- uint8_t pp_status
- uint8_t pp_pmcsr
- uint8_t pp_data

6.9.1 Detailed Description

Definition at line 49 of file pcivar.h.

6.9.2 Field Documentation

6.9.2.1 uint16_t pcicfg_pp::pp_cap

Definition at line 50 of file pcivar.h.

Referenced by pci_get_powerstate_method(), pci_print_verbose(), pci_read_extcap(), and pci_set_powerstate_method().

6.9.2.2 uint8_t pcicfg_pp::pp_data

Definition at line 53 of file pcivar.h.

Referenced by pci_read_extcap().

6.9.2.3 uint8_t pcicfg_pp::pp_pmcsr

Definition at line 52 of file pcivar.h.

Referenced by pci_read_extcap().

6.9.2.4 uint8_t pcicfg_pp::pp_status

Definition at line 51 of file pcivar.h.

Referenced by pci_get_powerstate_method(), pci_print_verbose(), pci_read_extcap(), and pci_set_powerstate_method().

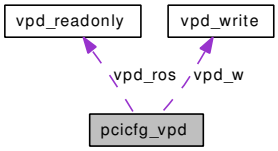
The documentation for this struct was generated from the following file:

- /usr/src/sys/dev/pci/pcivar.h

6.10 pcicfg_vpd Struct Reference

`#include <pcivar.h>`

Collaboration diagram for pcicfg_vpd:



Data Fields

- uint8_t vpd_reg
- char * vpd_ident
- int vpd_rocnt
- vpd_readonly * vpd_ros
- int vpd_wcnt
- vpd_write * vpd_w

6.10.1 Detailed Description

Definition at line 68 of file pcivar.h.

6.10.2 Field Documentation

6.10.2.1 char* pcicfg_vpd::vpd_ident

Definition at line 70 of file pcivar.h.

Referenced by pci_get_vpd_ident_method(), pci_print_verbose(), and pci_read_vpd().

6.10.2.2 uint8_t pcicfg_vpd::vpd_reg

Definition at line 69 of file pcivar.h.

Referenced by pci_print_verbose(), pci_read_extcap(), and pci_read_vpd_reg().

6.10.2.3 int pcicfg_vpd::vpd_rocnt

Definition at line 71 of file pcivar.h.

Referenced by pci_get_vpd_readonly_method(), pci_print_verbose(), and pci_read_vpd().

6.10.2.4 struct vpd_readonly* pcicfg_vpd::vpd_ros

Definition at line 72 of file pcivar.h.

Referenced by pci_get_vpd_readonly_method(), pci_print_verbose(), and pci_read_vpd().

6.10.2.5 struct vpd_write* pcicfg_vpd::vpd_w

Definition at line 74 of file pcivar.h.

Referenced by pci_print_verbose(), and pci_read_vpd().

6.10.2.6 int pcicfg_vpd::vpd_wcnt

Definition at line 73 of file pcivar.h.

Referenced by pci_print_verbose(), and pci_read_vpd().

The documentation for this struct was generated from the following file:

- /usr/src/sys/dev/pci/pcivar.h

6.11 pcie_cfg_elem Struct Reference

6.11.1 Detailed Description

Definition at line 64 of file pci_cfgreg.c.

The documentation for this struct was generated from the following file:

- /usr/src/sys/i386/pci/pci_cfgreg.c

6.12 pcih1cfgregs Struct Reference

`#include <pcivar.h>`

Data Fields

- pci_addr_t pmembase
- pci_addr_t pmemlimit
- uint32_t membase
- uint32_t memlimit
- uint32_t iobase
- uint32_t iolimit
- uint16_t secstat
- uint16_t bridgectl
- uint8_t seclat

6.12.1 Detailed Description

Definition at line 150 of file pcivar.h.

6.12.2 Field Documentation

6.12.2.1 uint16_t pcih1cfgregs::bridgectl

Definition at line 158 of file pcivar.h.

6.12.2.2 uint32_t pcih1cfgregs::iobase

Definition at line 155 of file pcivar.h.

6.12.2.3 uint32_t pcih1cfgregs::iolimit

Definition at line 156 of file pcivar.h.

6.12.2.4 uint32_t pcih1cfgregs::membase

Definition at line 153 of file pcivar.h.

6.12.2.5 uint32_t pcih1cfgregs::memlimit

Definition at line 154 of file pcivar.h.

6.12.2.6 pci_addr_t pcih1cfgregs::pmembase

Definition at line 151 of file pcivar.h.

6.12.2.7 pci_addr_t pcih1cfgregs::pmemlimit

Definition at line 152 of file pcivar.h.

6.12.2.8 uint8_t pcih1cfgregs::seclat

Definition at line 159 of file pcivar.h.

6.12.2.9 uint16_t pcih1cfgregs::secstat

Definition at line 157 of file pcivar.h.

The documentation for this struct was generated from the following file:

- /usr/src/sys/dev/pci/pcivar.h

6.13 pcih2cfgregs Struct Reference

`#include <pcivar.h>`

Data Fields

- uint32_t membase0
- uint32_t memlimit0
- uint32_t membase1
- uint32_t memlimit1
- uint32_t iobase0
- uint32_t iolimit0
- uint32_t iobase1
- uint32_t iolimit1
- uint32_t pccardif
- uint16_t secstat
- uint16_t bridgectl
- uint8_t seclat

6.13.1 Detailed Description

Definition at line 164 of file pcivar.h.

6.13.2 Field Documentation

6.13.2.1 uint16_t pcih2cfgregs::bridgectl

Definition at line 175 of file pcivar.h.

6.13.2.2 uint32_t pcih2cfgregs::iobase0

Definition at line 169 of file pcivar.h.

6.13.2.3 uint32_t pcih2cfgregs::iobase1

Definition at line 171 of file pcivar.h.

6.13.2.4 uint32_t pcih2cfgregs::iolimit0

Definition at line 170 of file pcivar.h.

6.13.2.5 uint32_t pcih2cfgregs::iolimit1

Definition at line 172 of file pcivar.h.

6.13.2.6 uint32_t pcih2cfgregs::membase0

Definition at line 165 of file pcivar.h.

6.13.2.7 uint32_t pcih2cfgregs::membase1

Definition at line 167 of file pcivar.h.

6.13.2.8 uint32_t pcih2cfgregs::memlimit0

Definition at line 166 of file pcivar.h.

6.13.2.9 uint32_t pcih2cfgregs::memlimit1

Definition at line 168 of file pcivar.h.

6.13.2.10 uint32_t pcih2cfgregs::pccardif

Definition at line 173 of file pcivar.h.

6.13.2.11 uint8_t pcih2cfgregs::seclat

Definition at line 176 of file pcivar.h.

6.13.2.12 uint16_t pcih2cfgregs::secstat

Definition at line 174 of file pcivar.h.

The documentation for this struct was generated from the following file:

- /usr/src/sys/dev/pci/pcivar.h

6.14 vpd_readonly Struct Reference

`#include <pcivar.h>`

Data Fields

- char keyword [2]
- char * value

6.14.1 Detailed Description

Definition at line 56 of file pcivar.h.

6.14.2 Field Documentation

6.14.2.1 char vpd_readonly::keyword[2]

Definition at line 57 of file pcivar.h.

Referenced by pci_get_vpd_readonly_method(), and pci_read_vpd().

6.14.2.2 char* vpd_readonly::value

Definition at line 58 of file pcivar.h.

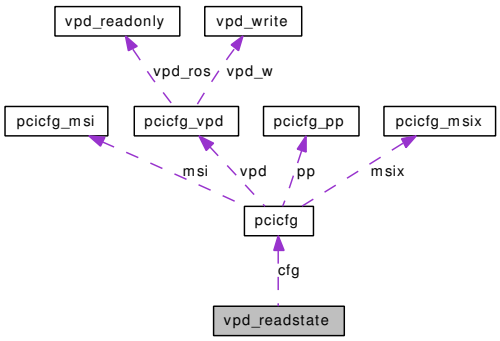
Referenced by pci_get_vpd_readonly_method(), and pci_read_vpd().

The documentation for this struct was generated from the following file:

- /usr/src/sys/dev/pci/pcivar.h

6.15 vpd_readstate Struct Reference

Collaboration diagram for vpd_readstate:



Data Fields

- device_t pcib
- pcicfgregs * cfg
- uint32_t val
- int bytesinval
- int off
- uint8_t cksum

6.15.1 Detailed Description

Definition at line 649 of file pci.c.

6.15.2 Field Documentation

6.15.2.1 int vpd_readstate::bytesinval

Definition at line 653 of file pci.c.

Referenced by pci_read_vpd(), and vpd_nextbyte().

6.15.2.2 pcicfgregs* vpd_readstate::cfg

Definition at line 651 of file pci.c.

Referenced by pci_read_vpd(), and vpd_nextbyte().

6.15.2.3 uint8_t vpd_readstate::cksum

Definition at line 655 of file pci.c.

Referenced by pci_read_vpd(), and vpd_nextbyte().

6.15.2.4 int vpd_readstate::off

Definition at line 654 of file pci.c.

Referenced by pci_read_vpd(), and vpd_nextbyte().

6.15.2.5 device_t vpd_readstate::pcib

Definition at line 650 of file pci.c.

Referenced by pci_read_vpd(), and vpd_nextbyte().

6.15.2.6 uint32_t vpd_readstate::val

Definition at line 652 of file pci.c.

Referenced by pci_read_vpd(), and vpd_nextbyte().

The documentation for this struct was generated from the following file:

- /usr/src/sys/dev/pci/pci.c

6.16 vpd_write Struct Reference

`#include <pcivar.h>`

Data Fields

- char keyword [2]
- char * value
- int start
- int len

6.16.1 Detailed Description

Definition at line 61 of file pcivar.h.

6.16.2 Field Documentation

6.16.2.1 char vpd_write::keyword[2]

Definition at line 62 of file pcivar.h.

Referenced by pci_read_vpd().

6.16.2.2 int vpd_write::len

Definition at line 65 of file pcivar.h.

Referenced by pci_read_vpd().

6.16.2.3 int vpd_write::start

Definition at line 64 of file pcivar.h.

Referenced by pci_read_vpd().

6.16.2.4 char* vpd_write::value

Definition at line 63 of file pcivar.h.

Referenced by pci_read_vpd().

The documentation for this struct was generated from the following file:

- /usr/src/sys/dev/pci/pcivar.h

Chapter 7

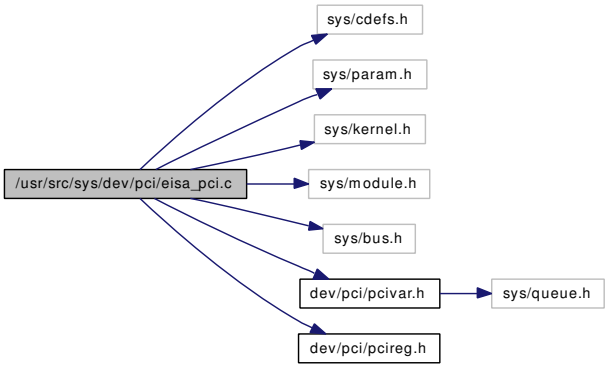
FreeBSD kernel i386 PCI device code File Documentation

7.1 notreviewed.dox File Reference

7.2 /usr/src/sys/dev/pci/eisa_pci.c File Reference

```
#include <sys/cdefs.h>
#include <sys/param.h>
#include <sys/kernel.h>
#include <sys/module.h>
#include <sys/bus.h>
#include <dev/pci/pcivar.h>
#include <dev/pci/pcireg.h>
```

Include dependency graph for eisa_pci.c:



Functions

- __FBSDID ("$FreeBSD: src/sys/dev/pci/eisa_pci.c,v 1.6 2004/05/30 20:08:37 phk Exp $")
- static int eisab_probe (device_t dev)
- static int eisab_attach (device_t dev)
- DRIVER_MODULE (eisab, pci, eisab_driver, eisab_devclass, 0, 0)

Variables

- static device_method_t eisab_methods [ ]
- static driver_t eisab_driver
- static devclass_t eisab_devclass

7.2.1 Function Documentation

7.2.1.1 __FBSDID ("$FreeBSD: src/sys/dev/pci/eisa_pci. *c*, v 1.6 2004/05/30 20:08:37 phk Exp $")

7.2.1.2 DRIVER_MODULE (eisab, pci, eisab_driver, eisab_devclass, 0, 0)

7.2.1.3 static int eisab_attach (device_t *dev*) [static]

Definition at line 110 of file eisa_pci.c.

7.2.1.4 static int eisab_probe (device_t *dev*) [static]

Definition at line 80 of file eisa_pci.c.

References PCIC_BRIDGE, and PCIS_BRIDGE_EISA.

7.2.2 Variable Documentation

7.2.2.1 devclass_t eisab_devclass [static]

Definition at line 75 of file eisa_pci.c.

7.2.2.2 driver_t eisab_driver [static]

Initial value:

```
{
    "eisab",
    eisab_methods,
    0,
}
```

Definition at line 69 of file eisa_pci.c.

7.2.2.3 device_method_t eisab_methods[ ] [static]

Initial value:

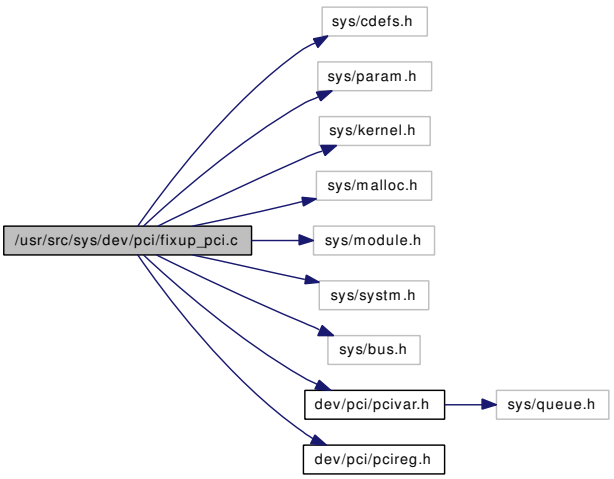
```
{

    DEVMETHOD(device_probe,             eisab_probe),
    DEVMETHOD(device_attach,            eisab_attach),
    DEVMETHOD(device_shutdown,          bus_generic_shutdown),
    DEVMETHOD(device_suspend,           bus_generic_suspend),
    DEVMETHOD(device_resume,            bus_generic_resume),


    DEVMETHOD(bus_print_child,          bus_generic_print_child),
    DEVMETHOD(bus_alloc_resource,       bus_generic_alloc_resource),
    DEVMETHOD(bus_release_resource,     bus_generic_release_resource),
    DEVMETHOD(bus_activate_resource,    bus_generic_activate_resource),
    DEVMETHOD(bus_deactivate_resource,  bus_generic_deactivate_resource),
    DEVMETHOD(bus_setup_intr,           bus_generic_setup_intr),
    DEVMETHOD(bus_teardown_intr,        bus_generic_teardown_intr),

    { 0, 0 }
}
```

Definition at line 49 of file eisa_pci.c.

7.3 /usr/src/sys/dev/pci/fixup_pci.c File Reference

```
#include <sys/cdefs.h>
#include <sys/param.h>
#include <sys/kernel.h>
#include <sys/malloc.h>
#include <sys/module.h>
#include <sys/systm.h>
#include <sys/bus.h>
#include <dev/pci/pcivar.h>
#include <dev/pci/pcireg.h>
```

Include dependency graph for fixup_pci.c:



Functions

- __FBSDID ("$FreeBSD: src/sys/dev/pci/fixup_pci.c,v 1.7 2006/05/24 14:08:31 jhb Exp $")
- static int fixup_pci_probe (device_t dev)
- static void fixwsc_natoma (device_t dev)
- static void fixc1_nforce2 (device_t dev)
- DRIVER_MODULE (fixup_pci, pci, fixup_pci_driver, fixup_pci_devclass, 0, 0)

Variables

- static device_method_t fixup_pci_methods [ ]
- static driver_t fixup_pci_driver
- static devclass_t fixup_pci_devclass

7.3.1 Function Documentation

7.3.1.1 __FBSDID ("$FreeBSD: src/sys/dev/pci/fixup_pci. *c*, v 1.7 2006/05/24 14:08:31 jhb Exp $")

7.3.1.2 DRIVER_MODULE (fixup_pci, pci, fixup_pci_driver, fixup_pci_devclass, 0, 0)

7.3.1.3 static void fixc1_nforce2 (device_t *dev*) [static]

Definition at line 127 of file fixup_pci.c.

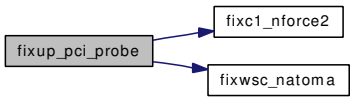
Referenced by fixup_pci_probe().

7.3.1.4 static int fixup_pci_probe (device_t *dev*) [static]

Definition at line 74 of file fixup_pci.c.

References fixc1_nforce2(), and fixwsc_natoma().

Here is the call graph for this function:



7.3.1.5 static void fixwsc_natoma (device_t *dev*) [static]

Definition at line 88 of file fixup_pci.c.

Referenced by fixup_pci_probe().

7.3.2 Variable Documentation

7.3.2.1 devclass_t fixup_pci_devclass [static]

Definition at line 69 of file fixup_pci.c.

7.3.2.2 driver_t fixup_pci_driver [static]

Initial value:

```
{
    "fixup_pci",
    fixup_pci_methods,
    0,
}
```

Definition at line 63 of file fixup_pci.c.

7.3.2.3 device_method_t fixup_pci_methods[ ] [static]

Initial value:

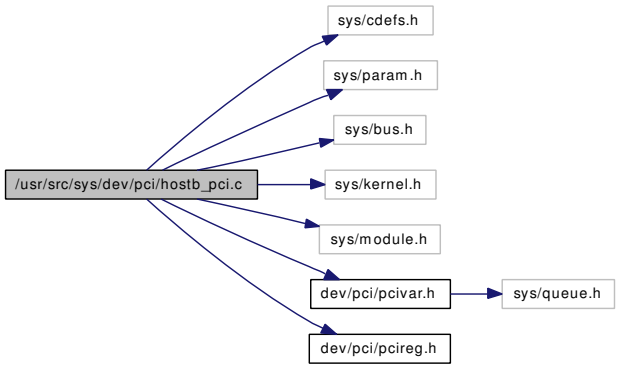
```
{

    DEVMETHOD(device_probe,             fixup_pci_probe),
    DEVMETHOD(device_attach,            bus_generic_attach),
    { 0, 0 }
}
```

Definition at line 56 of file fixup_pci.c.

7.4 /usr/src/sys/dev/pci/hostb_pci.c File Reference

```
#include <sys/cdefs.h>
#include <sys/param.h>
#include <sys/bus.h>
#include <sys/kernel.h>
#include <sys/module.h>
#include <dev/pci/pcivar.h>
#include <dev/pci/pcireg.h>
```

Include dependency graph for hostb_pci.c:



Functions

- __FBSDID ("$FreeBSD: src/sys/dev/pci/hostb_pci.c,v 1.1 2005/12/20 21:09:44 jhb Exp $")
- static int pci_hostb_probe (device_t dev)
- static int pci_hostb_attach (device_t dev)
- static int pci_hostb_read_ivar (device_t dev, device_t child, int which, uintptr_t *result)
- static int pci_hostb_write_ivar (device_t dev, device_t child, int which, uintptr_t value)
- static struct resource * pci_hostb_alloc_resource (device_t dev, device_t child, int type, int *rid, u_long start, u_long end, u_long count, u_int flags)
- static int pci_hostb_release_resource (device_t dev, device_t child, int type, int rid, struct resource *r)
- static uint32_t pci_hostb_read_config (device_t dev, device_t child, int reg, int width)
- static void pci_hostb_write_config (device_t dev, device_t child, int reg, uint32_t val, int width)
- static int pci_hostb_enable_busmaster (device_t dev, device_t child)
- static int pci_hostb_disable_busmaster (device_t dev, device_t child)
- static int pci_hostb_enable_io (device_t dev, device_t child, int space)
- static int pci_hostb_disable_io (device_t dev, device_t child, int space)
- static int pci_hostb_set_powerstate (device_t dev, device_t child, int state)
- static int pci_hostb_get_powerstate (device_t dev, device_t child)
- static int pci_hostb_assign_interrupt (device_t dev, device_t child)
- static int pci_hostb_find_extcap (device_t dev, device_t child, int capability, int *capreg)
- DRIVER_MODULE (hostb, pci, pci_hostb_driver, pci_hostb_devclass, 0, 0)

Variables

- static device_method_t pci_hostb_methods [ ]
- static driver_t pci_hostb_driver
- static devclass_t pci_hostb_devclass

7.4.1 Function Documentation

7.4.1.1 __FBSDID ("$FreeBSD: src/sys/dev/pci/hostb_pci. *c*, v 1.1 2005/12/20 21:09:44 jhb Exp $")

7.4.1.2 DRIVER_MODULE (hostb, pci, pci_hostb_driver, pci_hostb_devclass, 0, 0)

7.4.1.3 static struct resource* pci_hostb_alloc_resource (device_t *dev*, device_t *child*, int *type*, int * *rid*, u_long *start*, u_long *end*, u_long *count*, u_int *flags*) [static]

Definition at line 104 of file hostb_pci.c.

7.4.1.4 static int pci_hostb_assign_interrupt (device_t *dev*, device_t *child*) [static]

Definition at line 191 of file hostb_pci.c.

7.4.1.5 static int pci_hostb_attach (device_t *dev*) [static]

Definition at line 72 of file hostb_pci.c.

References PCIY_AGP.

7.4.1.6 static int pci_hostb_disable_busmaster (device_t *dev*, device_t *child*) [static]

Definition at line 146 of file hostb_pci.c.

7.4.1.7 static int pci_hostb_disable_io (device_t *dev*, device_t *child*, int *space*) [static]

Definition at line 164 of file hostb_pci.c.

7.4.1.8 static int pci_hostb_enable_busmaster (device_t *dev*, device_t *child*) [static]

Definition at line 137 of file hostb_pci.c.

7.4.1.9 static int pci_hostb_enable_io (device_t *dev*, device_t *child*, int *space*) [static]

Definition at line 155 of file hostb_pci.c.

7.4.1.10 static int pci_hostb_find_extcap (device_t *dev*, device_t *child*, int *capability*, int * *capreg*) [static]

Definition at line 200 of file hostb_pci.c.

7.4.1.11 static int pci_hostb_get_powerstate (device_t *dev*, device_t *child*) [static]

Definition at line 182 of file hostb_pci.c.

7.4.1.12 static int pci_hostb_probe (device_t *dev*) [static]

Definition at line 46 of file hostb_pci.c.

References PCIC_BRIDGE, and PCIS_BRIDGE_HOST.

7.4.1.13 static uint32_t pci_hostb_read_config (device_t *dev*, device_t *child*, int *reg*, int *width*) [static]

Definition at line 122 of file hostb_pci.c.

7.4.1.14 static int pci_hostb_read_ivar (device_t *dev*, device_t *child*, int *which*, uintptr_t * *result*) [static]

Definition at line 90 of file hostb_pci.c.

7.4.1.15 static int pci_hostb_release_resource (device_t *dev*, device_t *child*, int *type*, int *rid*, struct resource * *r*) [static]

Definition at line 112 of file hostb_pci.c.

7.4.1.16 static int pci_hostb_set_powerstate (device_t *dev*, device_t *child*, int *state*) [static]

Definition at line 173 of file hostb_pci.c.

7.4.1.17 static void pci_hostb_write_config (device_t *dev*, device_t *child*, int *reg*, uint32_t *val*, int *width*) [static]

Definition at line 129 of file hostb_pci.c.

7.4.1.18 static int pci_hostb_write_ivar (device_t *dev*, device_t *child*, int *which*, uintptr_t *value*) [static]

Definition at line 97 of file hostb_pci.c.

7.4.2 Variable Documentation

7.4.2.1 devclass_t pci_hostb_devclass [static]

Definition at line 247 of file hostb_pci.c.

7.4.2.2 driver_t pci_hostb_driver [static]

Initial value:

```
{
        "hostb",
        pci_hostb_methods,
        1,
}
```

Definition at line 241 of file hostb_pci.c.

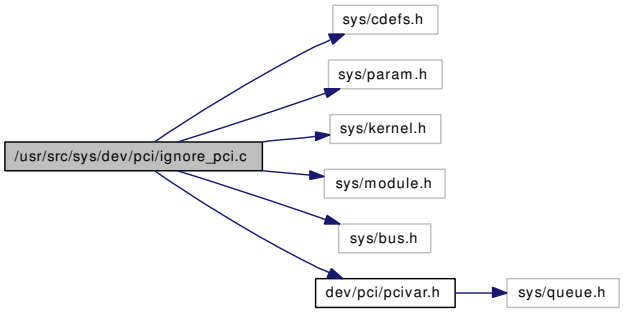
7.4.2.3 device_method_t pci_hostb_methods[ ] [static]

Definition at line 207 of file hostb_pci.c.

7.5 /usr/src/sys/dev/pci/ignore_pci.c File Reference

```
#include <sys/cdefs.h>
#include <sys/param.h>
#include <sys/kernel.h>
#include <sys/module.h>
#include <sys/bus.h>
#include <dev/pci/pcivar.h>
```

Include dependency graph for ignore_pci.c:



Functions

- __FBSDID ("$FreeBSD: src/sys/dev/pci/ignore_pci.c,v 1.4 2004/05/30 17:57:41 phk Exp $")
- static int ignore_pci_probe (device_t dev)
- DRIVER_MODULE (ignore_pci, pci, ignore_pci_driver, ignore_pci_devclass, 0, 0)

Variables

- static device_method_t ignore_pci_methods [ ]
- static driver_t ignore_pci_driver
- static devclass_t ignore_pci_devclass

7.5.1 Function Documentation

7.5.1.1 __FBSDID ("$FreeBSD: src/sys/dev/pci/ignore_pci. *c*, v 1.4 2004/05/30 17:57:41 phk Exp $")

7.5.1.2 DRIVER_MODULE (ignore_pci, pci, ignore_pci_driver, ignore_pci_devclass, 0, 0)

7.5.1.3 static int ignore_pci_probe (device_t *dev*) [static]

Definition at line 63 of file ignore_pci.c.

7.5.2 Variable Documentation

7.5.2.1 devclass_t ignore_pci_devclass [static]

Definition at line 58 of file ignore_pci.c.

7.5.2.2 driver_t ignore_pci_driver [static]

Initial value:

```
{
    "ignore_pci",
    ignore_pci_methods,
    0,
}
```

Definition at line 52 of file ignore_pci.c.

7.5.2.3 device_method_t ignore_pci_methods[ ] [static]

Initial value:

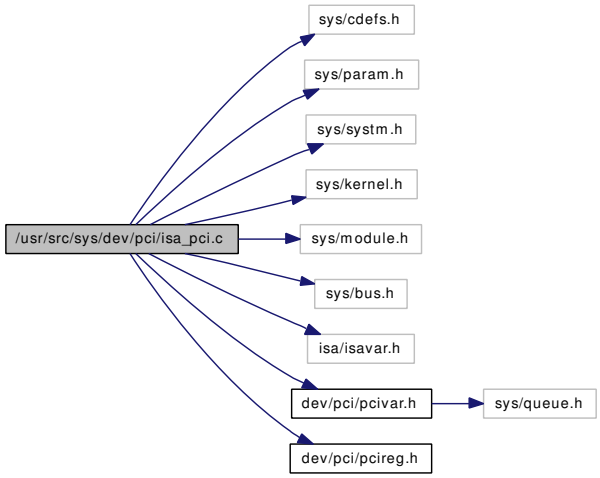
```
{

    DEVMETHOD(device_probe,             ignore_pci_probe),
    DEVMETHOD(device_attach,            bus_generic_attach),
    { 0, 0 }
}
```

Definition at line 45 of file ignore_pci.c.

7.6 /usr/src/sys/dev/pci/isa_pci.c File Reference

```
#include <sys/cdefs.h>
#include <sys/param.h>
#include <sys/systm.h>
#include <sys/kernel.h>
#include <sys/module.h>
#include <sys/bus.h>
#include <isa/isavar.h>
#include <dev/pci/pcivar.h>
#include <dev/pci/pcireg.h>
```

Include dependency graph for isa_pci.c:



Functions

- __FBSDID ("$FreeBSD: src/sys/dev/pci/isa_pci.c,v 1.13 2005/09/29 15:00:09 jhb Exp $")
- static int isab_probe (device_t dev)
- DRIVER_MODULE (isab, pci, isab_driver, isab_devclass, 0, 0)

Variables

- static device_method_t isab_methods [ ]
- static driver_t isab_driver

7.6.1 Function Documentation

7.6.1.1 __FBSDID ("$FreeBSD: src/sys/dev/pci/isa_pci. *c*, v 1.13 2005/09/29 15:00:09 jhb Exp $")

7.6.1.2 DRIVER_MODULE (isab, pci, isab_driver, isab_devclass, 0, 0)

7.6.1.3 static int isab_probe (device_t *dev*) [static]

Definition at line 84 of file isa_pci.c.

References PCIC_BRIDGE, and PCIS_BRIDGE_ISA.

7.6.2 Variable Documentation

7.6.2.1 driver_t isab_driver [static]

Initial value:

```
{
    "isab",
    isab_methods,
    0,
}
```

Definition at line 71 of file isa_pci.c.

7.6.2.2 device_method_t isab_methods[ ] [static]

Initial value:

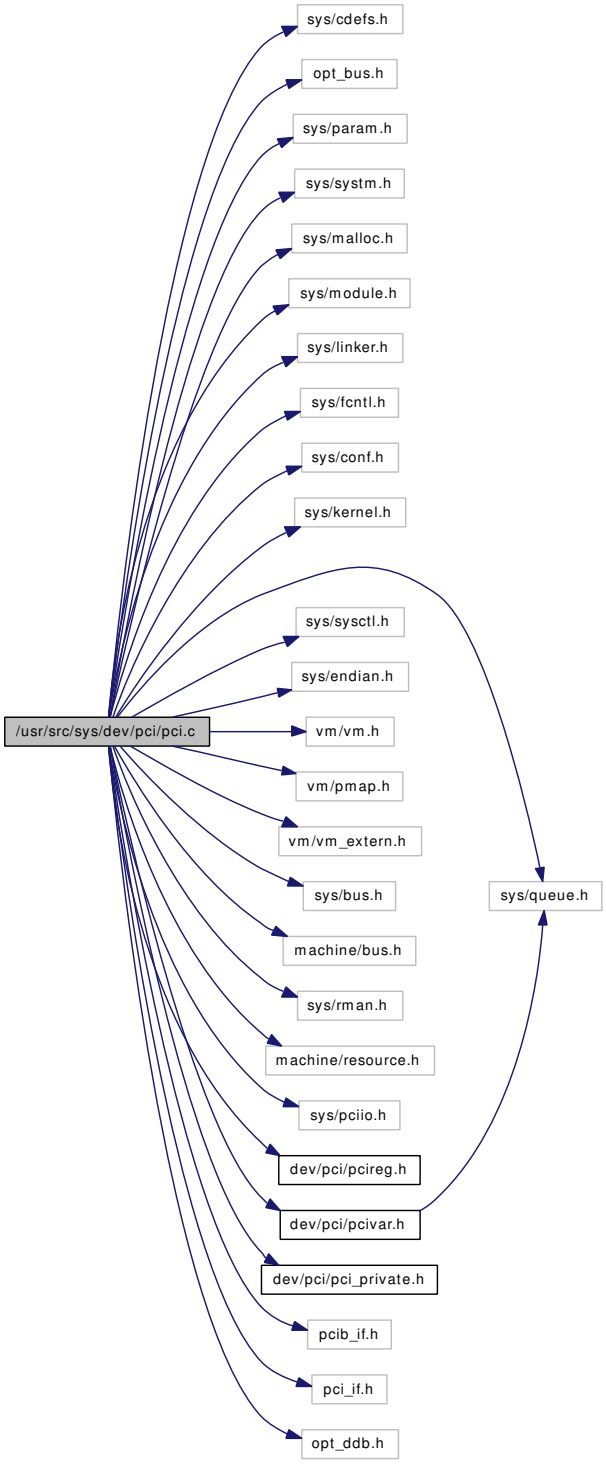
```
{

    DEVMETHOD(device_probe,             isab_probe),
    DEVMETHOD(device_attach,            isab_attach),
    DEVMETHOD(device_detach,            bus_generic_detach),
    DEVMETHOD(device_shutdown,          bus_generic_shutdown),
    DEVMETHOD(device_suspend,           bus_generic_suspend),
    DEVMETHOD(device_resume,            bus_generic_resume),


    DEVMETHOD(bus_print_child,          bus_generic_print_child),
    DEVMETHOD(bus_alloc_resource,       bus_generic_alloc_resource),
    DEVMETHOD(bus_release_resource,     bus_generic_release_resource),
    DEVMETHOD(bus_activate_resource,    bus_generic_activate_resource),
    DEVMETHOD(bus_deactivate_resource,  bus_generic_deactivate_resource),
    DEVMETHOD(bus_setup_intr,           bus_generic_setup_intr),
    DEVMETHOD(bus_teardown_intr,        bus_generic_teardown_intr),

    { 0, 0 }
}
```

Definition at line 50 of file isa_pci.c.

7.7 /usr/src/sys/dev/pci/pci.c File Reference

```
#include <sys/cdefs.h>
#include "opt_bus.h"
#include <sys/param.h>
#include <sys/systm.h>
#include <sys/malloc.h>
#include <sys/module.h>
#include <sys/linker.h>
#include <sys/fcntl.h>
#include <sys/conf.h>
#include <sys/kernel.h>
#include <sys/queue.h>
#include <sys/sysctl.h>
#include <sys/endian.h>
#include <vm/vm.h>
#include <vm/pmap.h>
#include <vm/vm_extern.h>
#include <sys/bus.h>
#include <machine/bus.h>
#include <sys/rman.h>
#include <machine/resource.h>
#include <sys/pciio.h>
#include <dev/pci/pcireg.h>
#include <dev/pci/pcivar.h>
#include <dev/pci/pci_private.h>
#include "pcib_if.h"
#include "pci_if.h"
#include "opt_ddb.h"
```

Include dependency graph for pci.c:



Data Structures

- struct pci_quirk
- struct vpd_readstate

Defines

- #define ACPI_PWR_FOR_SLEEP(x, y, z)
- #define PCI_QUIRK_MAP_REG 1
- #define PCI_QUIRK_DISABLE_MSI 2
- #define PCI_MAPMEM 0x01
- #define PCI_MAPMEMP 0x02
- #define PCI_MAPPORT 0x04
- #define REG(n, w) PCIB_READ_CONFIG(pcib, b, s, f, n, w)
- #define REG(n, w) PCIB_READ_CONFIG(pcib, b, s, f, n, w)
- #define REG(n, w) PCIB_READ_CONFIG(pcib, cfg → bus, cfg → slot, cfg → func, n, w)
- #define WREG(n, v, w) PCIB_WRITE_CONFIG(pcib, cfg → bus, cfg → slot, cfg → func, n, v, w)
- #define REG(n, w) PCIB_READ_CONFIG(pcib, busno, s, f, n, w)

Functions

- __FBSDID ("$FreeBSD: src/sys/dev/pci/pci.c,v 1.343 2007/02/17 16:56:39 sos Exp $")
- static uint32_t pci_mapbase (unsigned mapreg)
- static int pci_maptype (unsigned mapreg)
- static int pci_mapsize (unsigned testval)
- static int pci_maprange (unsigned mapreg)
- static void pci_fixancient (pcicfgregs *cfg)
- static int pci_porten (device_t pcib, int b, int s, int f)
- static int pci_memen (device_t pcib, int b, int s, int f)
- static void pci_assign_interrupt (device_t bus, device_t dev, int force_route)
- static int pci_add_map (device_t pcib, device_t bus, device_t dev, int b, int s, int f, int reg, struct resource_list *rl, int force, int prefetch)
- static int pci_probe (device_t dev)
- static int pci_attach (device_t dev)
- static void pci_load_vendor_data (void)
- static int pci_describe_parse_line (char **ptr, int *vendor, int *device, char **desc)
- static char * pci_describe_device (device_t dev)
- static int pci_modevent (module_t mod, int what, void *arg)
- static void pci_hdrtypedata (device_t pcib, int b, int s, int f, pcicfgregs *cfg)
- static void pci_read_extcap (device_t pcib, pcicfgregs *cfg)
- static uint32_t pci_read_vpd_reg (device_t pcib, pcicfgregs *cfg, int reg)
- static void pci_read_vpd (device_t pcib, pcicfgregs *cfg)
- static int pci_msi_blacklisted (void)
- DEFINE_CLASS_0 (pci, pci_driver, pci_methods, 0)
- DRIVER_MODULE (pci, pcib, pci_driver, pci_devclass, pci_modevent, 0)
- MODULE_VERSION (pci, 1)
- SYSCTL_NODE (_hw, OID_AUTO, pci, CTLFLAG_RD, 0,"PCI bus tuning parameters")
- TUNABLE_INT ("hw.pci.enable_io_modes",&pci_enable_io_modes)
- SYSCTL_INT (_hw_pci, OID_AUTO, enable_io_modes, CTLFLAG_RW,&pci_enable_io_modes, 1,"Enable I/O and memory bits in the config register. Some BIOSes do not\n\ enable these bits correctly. We'd like to do this all the time, but there\n\ are some peripherals that this causes problems with.")
- TUNABLE_INT ("hw.pci.do_power_nodriver",&pci_do_power_nodriver)

- SYSCTL_INT (_hw_pci, OID_AUTO, do_power_nodriver, CTLFLAG_RW,&pci_do_power_-nodriver, 0,"Place a function into D3 state when no driver attaches to it. 0 means\n\ disable. 1 means conservatively place devices into D3 state. 2 means\n\ agressively place devices into D3 state. 3 means put absolutely everything\n\ in D3 state.")
- TUNABLE_INT ("hw.pci.do_power_resume",&pci_do_power_resume)
- SYSCTL_INT (_hw_pci, OID_AUTO, do_power_resume, CTLFLAG_RW,&pci_do_power_-resume, 1,"Transition from D3 → D0 on resume.")
- TUNABLE_INT ("hw.pci.enable_vpd",&pci_do_vpd)
- SYSCTL_INT (_hw_pci, OID_AUTO, enable_vpd, CTLFLAG_RW,&pci_do_vpd, 1,"Enable support for VPD.")
- TUNABLE_INT ("hw.pci.enable_msi",&pci_do_msi)
- SYSCTL_INT (_hw_pci, OID_AUTO, enable_msi, CTLFLAG_RW,&pci_do_msi, 1,"Enable support for MSI interrupts")
- TUNABLE_INT ("hw.pci.enable_msix",&pci_do_msix)
- SYSCTL_INT (_hw_pci, OID_AUTO, enable_msix, CTLFLAG_RW,&pci_do_msix, 1,"Enable support for MSI-X interrupts")
- TUNABLE_INT ("hw.pci.honor_msi_blacklist",&pci_honor_msi_blacklist)
- SYSCTL_INT (_hw_pci, OID_AUTO, honor_msi_blacklist, CTLFLAG_RD,&pci_honor_msi_-blacklist, 1,"Honor chipset blacklist for MSI")
- device_t pci_find_bsf (uint8_t bus, uint8_t slot, uint8_t func)
- device_t pci_find_device (uint16_t vendor, uint16_t device)
- static uint32_t pci_mapbase (uint32_t mapreg)
- static int pci_mapsize (uint32_t testval)
- pci_devinfo ∗ pci_read_device (device_t pcib, int b, int s, int f, size_t size)
- static uint8_t vpd_nextbyte (struct vpd_readstate ∗vrs)
- int pci_get_vpd_ident_method (device_t dev, device_t child, const char ∗∗identptr)
- int pci_get_vpd_readonly_method (device_t dev, device_t child, const char ∗kw, const char ∗∗vptr)
- int pci_find_extcap_method (device_t dev, device_t child, int capability, int ∗capreg)
- void pci_enable_msix (device_t dev, u_int index, uint64_t address, uint32_t data)
- void pci_mask_msix (device_t dev, u_int index)
- void pci_unmask_msix (device_t dev, u_int index)
- int pci_pending_msix (device_t dev, u_int index)
- int pci_alloc_msix_method (device_t dev, device_t child, int ∗count)
- int pci_remap_msix_method (device_t dev, device_t child, u_int ∗indices)
- static int pci_release_msix (device_t dev, device_t child)
- int pci_msix_count_method (device_t dev, device_t child)
- void pci_enable_msi (device_t dev, uint64_t address, uint16_t data)
- static void pci_resume_msi (device_t dev)
- int pci_msi_device_blacklisted (device_t dev)
- int pci_alloc_msi_method (device_t dev, device_t child, int ∗count)
- int pci_release_msi_method (device_t dev, device_t child)
- int pci_msi_count_method (device_t dev, device_t child)
- int pci_freecfg (struct pci_devinfo ∗dinfo)
- int pci_set_powerstate_method (device_t dev, device_t child, int state)
- int pci_get_powerstate_method (device_t dev, device_t child)
- static __inline void pci_set_command_bit (device_t dev, device_t child, uint16_t bit)
- static __inline void pci_clear_command_bit (device_t dev, device_t child, uint16_t bit)
- int pci_enable_busmaster_method (device_t dev, device_t child)
- int pci_disable_busmaster_method (device_t dev, device_t child)
- int pci_enable_io_method (device_t dev, device_t child, int space)
- int pci_disable_io_method (device_t dev, device_t child, int space)

- void pci_print_verbose (struct pci_devinfo *dinfo)
- static void pci_ata_maps (device_t pcib, device_t bus, device_t dev, int b, int s, int f, struct resource_-list *rl, int force, uint32_t prefetchmask)
- void pci_add_resources (device_t bus, device_t dev, int force, uint32_t prefetchmask)
- void pci_add_children (device_t dev, int busno, size_t dinfo_size)
- void pci_add_child (device_t bus, struct pci_devinfo *dinfo)
- int pci_suspend (device_t dev)
- int pci_resume (device_t dev)
- void pci_driver_added (device_t dev, driver_t *driver)
- int pci_print_child (device_t dev, device_t child)
- void pci_probe_nomatch (device_t dev, device_t child)
- int pci_read_ivar (device_t dev, device_t child, int which, uintptr_t *result)
- int pci_write_ivar (device_t dev, device_t child, int which, uintptr_t value)
- static struct resource * pci_alloc_map (device_t dev, device_t child, int type, int *rid, u_long start, u_long end, u_long count, u_int flags)
- resource * pci_alloc_resource (device_t dev, device_t child, int type, int *rid, u_long start, u_long end, u_long count, u_int flags)
- void pci_delete_resource (device_t dev, device_t child, int type, int rid)
- resource_list * pci_get_resource_list (device_t dev, device_t child)
- uint32_t pci_read_config_method (device_t dev, device_t child, int reg, int width)
- void pci_write_config_method (device_t dev, device_t child, int reg, uint32_t val, int width)
- int pci_child_location_str_method (device_t dev, device_t child, char *buf, size_t buflen)
- int pci_child_pnpinfo_str_method (device_t dev, device_t child, char *buf, size_t buflen)
- int pci_assign_interrupt_method (device_t dev, device_t child)
- void pci_cfg_restore (device_t dev, struct pci_devinfo *dinfo)
- void pci_cfg_save (device_t dev, struct pci_devinfo *dinfo, int setstate)

Variables

- static device_method_t pci_methods [ ]
- static devclass_t pci_devclass
- static char * pci_vendordata
- static size_t pci_vendordata_size
- pci_quirk pci_quirks [ ]
- devlist pci_devq
- uint32_t pci_generation
- uint32_t pci_numdevs = 0
- static int pcie_chipset
- static int pcix_chipset
- static int pci_enable_io_modes = 1
- static int pci_do_power_nodriver = 0
- static int pci_do_power_resume = 1
- static int pci_do_vpd = 1
- static int pci_do_msi = 1
- static int pci_do_msix = 1
- static int pci_honor_msi_blacklist = 1
- struct {
  int class
  int subclass
  char * desc
  } pci_nomatch_tab [ ]

7.7.1 Define Documentation

7.7.1.1 #define ACPI_PWR_FOR_SLEEP(x, y, z)

Definition at line 71 of file pci.c.

Referenced by pci_resume(), and pci_suspend().

7.7.1.2 #define PCI_MAPMEM 0x01

Definition at line 213 of file pci.c.

Referenced by pci_add_map(), pci_alloc_map(), and pci_maptype().

7.7.1.3 #define PCI_MAPMEMP 0x02

Definition at line 214 of file pci.c.

Referenced by pci_maptype().

7.7.1.4 #define PCI_MAPPORT 0x04

Definition at line 215 of file pci.c.

Referenced by pci_maptype().

7.7.1.5 #define PCI_QUIRK_DISABLE_MSI 2

Definition at line 172 of file pci.c.

Referenced by pci_msi_device_blacklisted().

7.7.1.6 #define PCI_QUIRK_MAP_REG 1

Definition at line 171 of file pci.c.

Referenced by pci_add_resources().

7.7.1.7 #define REG(n, w) PCIB_READ_CONFIG(pcib, busno, s, f, n, w)

7.7.1.8 #define REG(n, w) PCIB_READ_CONFIG(pcib, cfg → bus, cfg → slot, cfg → func, n, w)

7.7.1.9 #define REG(n, w) PCIB_READ_CONFIG(pcib, b, s, f, n, w)

7.7.1.10 #define REG(n, w) PCIB_READ_CONFIG(pcib, b, s, f, n, w)

Referenced by pci_add_children(), pci_hdrtypedata(), pci_read_device(), pci_read_extcap(), and pci_-read_vpd_reg().

7.7.1.11 #define WREG(n, v, w) PCIB_WRITE_CONFIG(pcib, cfg → bus, cfg → slot, cfg → func, n, v, w)

Referenced by pci_read_extcap(), and pci_read_vpd_reg().

7.7.2 Function Documentation

7.7.2.1 __FBSDID ("$FreeBSD: src/sys/dev/pci/pci. *c*, v 1.343 2007/02/17 16:56:39 sos Exp $")

7.7.2.2 DEFINE_CLASS_0 (pci, pci_driver, pci_methods, 0)

7.7.2.3 DRIVER_MODULE (pci, pcib, pci_driver, pci_devclass, pci_modevent, 0)

7.7.2.4 MODULE_VERSION (pci, 1)

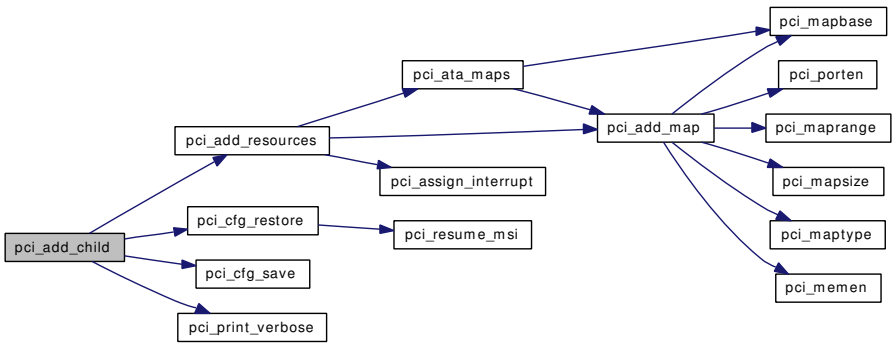
7.7.2.5 void pci_add_child (device_t *bus*, struct pci_devinfo ∗ *dinfo*)

Definition at line 2301 of file pci.c.

References pci_add_resources(), pci_cfg_restore(), pci_cfg_save(), and pci_print_verbose().

Referenced by pci_add_children().

Here is the call graph for this function:



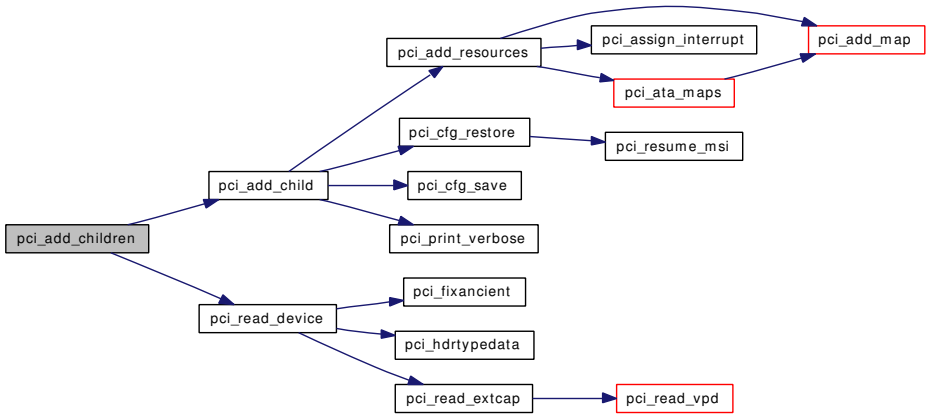
7.7.2.6 void pci_add_children (device_t *dev*, int *busno*, size_t *dinfo_size*)

Definition at line 2269 of file pci.c.

References pci_add_child(), PCI_FUNCMAX, PCI_MAXHDRTYPE, pci_read_device(), PCIM_-HDRTYPE, PCIM_MFDEV, PCIR_HDRTYPE, and REG.

Referenced by pci_attach().

Here is the call graph for this function:



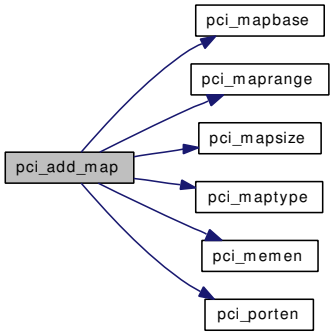
7.7.2.7 static int pci_add_map (device_t *pcib*, device_t *bus*, device_t *dev*, int *b*, int *s*, int *f*, int *reg*, struct resource_list * *rl*, int *force*, int *prefetch*) [static]

Definition at line 1972 of file pci.c.

References pci_enable_io_modes, pci_mapbase(), PCI_MAPMEM, pci_maprange(), pci_mapsize(), pci_maptype(), pci_memen(), pci_porten(), PCIM_CMD_MEMEN, PCIM_CMD_PORTEN, and PCIR_COMMAND.

Referenced by pci_add_resources(), and pci_ata_maps().

Here is the call graph for this function:



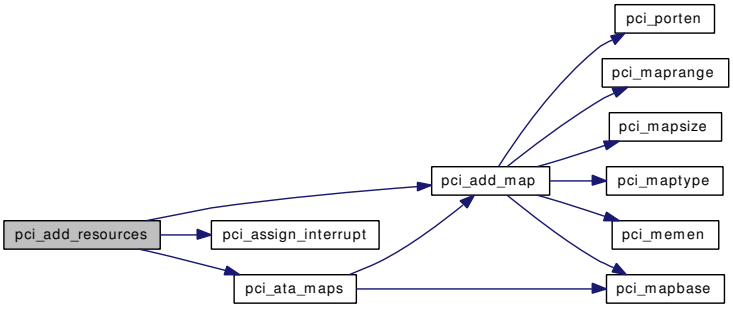
7.7.2.8 void pci_add_resources (device_t *bus*, device_t *dev*, int *force*, uint32_t *prefetchmask*)

Definition at line 2216 of file pci.c.

References pci_quirk::arg1, pcicfg::bus, pcicfg::device, pci_quirk::devid, pcicfg::func, pcicfg::intline, pcicfg::intpin, pcicfg::nummaps, pci_add_map(), pci_assign_interrupt(), pci_ata_maps(), PCI_QUIRK_MAP_REG, pci_quirks, PCIC_STORAGE, PCIP_STORAGE_IDE_MASTERDEV, PCIR_BAR, PCIS_STORAGE_IDE, pcicfg::slot, pci_quirk::type, and pcicfg::vendor.

Referenced by pci_add_child().

Here is the call graph for this function:



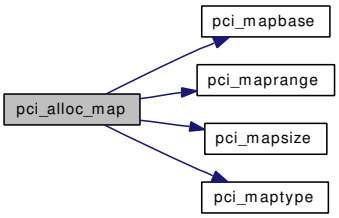
7.7.2.9 static struct resource* pci_alloc_map (device_t *dev*, device_t *child*, int *type*, int * *rid*, u_long *start*, u_long *end*, u_long *count*, u_int *flags*) [static]

Definition at line 2915 of file pci.c.

References pci_mapbase(), PCI_MAPMEM, pci_maprange(), pci_mapsize(), and pci_maptype().

Referenced by pci_alloc_resource().

Here is the call graph for this function:



7.7.2.10 int pci_alloc_msi_method (device_t *dev*, device_t *child*, int * *count*)

Definition at line 1445 of file pci.c.

References pcicfg::msi, pcicfg_msi::msi_alloc, pcicfg_msi::msi_ctrl, pcicfg_msi::msi_location, pcicfg_msi::msi_msgnum, pcicfg::msix, pcicfg_msix::msix_alloc, pci_do_msi, pci_msi_blacklisted(), PCIM_MSICTRL_MME_MASK, and PCIR_MSI_CTRL.

Here is the call graph for this function:

7.7.2.11 int pci_alloc_msix_method (device_t *dev*, device_t *child*, int * *count*)

Definition at line 1067 of file pci.c.

References pcicfg::msi, pcicfg_msi::msi_alloc, pcicfg::msix, pcicfg_msix::msix_alloc, pcicfg_-msix::msix_location, pcicfg_msix::msix_msgnum, pcicfg_msix::msix_pba_bar, pcicfg_msix::msix_pba_-res, pcicfg_msix::msix_table_bar, pcicfg_msix::msix_table_res, pci_do_msix, and pci_msi_blacklisted().

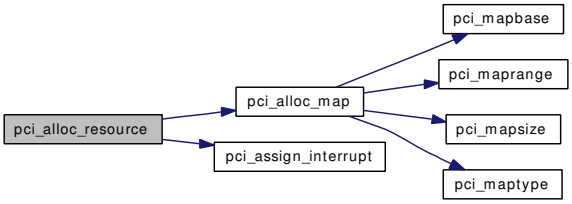
Here is the call graph for this function:

7.7.2.12 struct resource* pci_alloc_resource (device_t *dev*, device_t *child*, int *type*, int * *rid*, u_long *start*, u_long *end*, u_long *count*, u_int *flags*)

Definition at line 3005 of file pci.c.

References pcicfg::intline, pcicfg::intpin, pcicfg::msi, pcicfg_msi::msi_alloc, pcicfg::msix, pcicfg_-msix::msix_alloc, pcicfg::nummaps, pci_alloc_map(), pci_assign_interrupt(), and PCIR_BAR.

Here is the call graph for this function:



7.7.2.13 static void pci_assign_interrupt (device_t *bus*, device_t *dev*, int *force_route*) [static]

Definition at line 2169 of file pci.c.

References pcicfg::bus, pcicfg::intline, pcicfg::intpin, PCIR_INTLINE, and pcicfg::slot.

Referenced by pci_add_resources(), and pci_alloc_resource().

7.7.2.14 int pci_assign_interrupt_method (device_t *dev*, device_t *child*)

Definition at line 3173 of file pci.c.

References pcicfg::intpin.

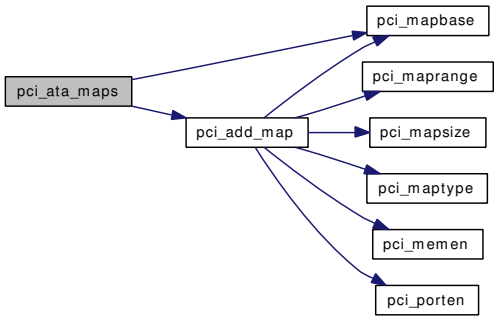
7.7.2.15 static void pci_ata_maps (device_t *pcib*, device_t *bus*, device_t *dev*, int *b*, int *s*, int *f*, struct resource_list * *rl*, int *force*, uint32_t *prefetchmask*) [static]

Definition at line 2115 of file pci.c.

References pci_add_map(), pci_mapbase(), PCIP_STORAGE_IDE_MODEPRIM, PCIP_STORAGE_-IDE_MODESEC, PCIR_BAR, and PCIR_PROGIF.

Referenced by pci_add_resources().

Here is the call graph for this function:

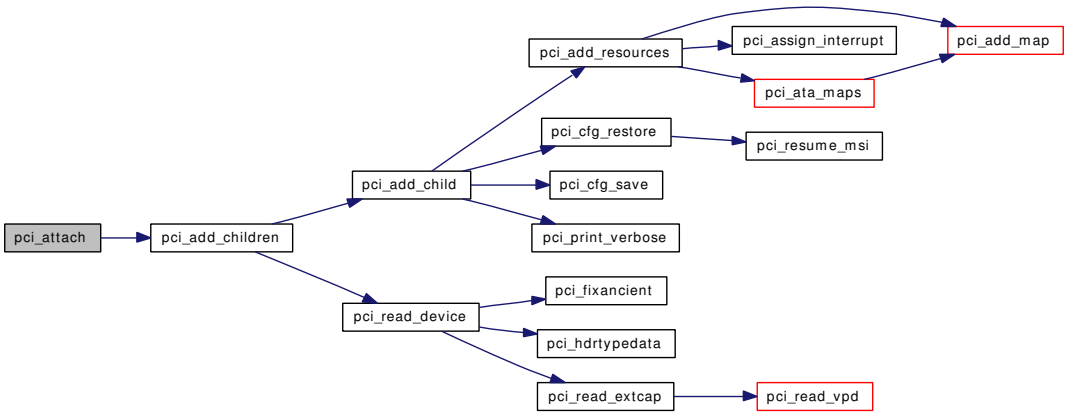


7.7.2.16 static int pci_attach (device_t *dev*) [static]

Definition at line 2323 of file pci.c.

References pci_add_children().

Here is the call graph for this function:



7.7.2.17 void pci_cfg_restore (device_t *dev*, struct pci_devinfo * *dinfo*)

Definition at line 3205 of file pci.c.

References pci_resume_msi(), PCIR_BAR, PCIR_BIOS, PCIR_CACHELNSZ, PCIR_COMMAND, PCIR_INTLINE, PCIR_INTPIN, PCIR_LATTIMER, PCIR_MAXLAT, PCIR_MINGNT, PCIR_PROGIF, and PCIR_REVID.

Referenced by pci_add_child(), pci_driver_added(), and pci_resume().

Here is the call graph for this function:

7.7.2.18 void pci_cfg_save (device_t *dev*, struct pci_devinfo * *dinfo*, int *setstate*)

Definition at line 3252 of file pci.c.

References pci_do_power_nodriver, PCIC_BASEPERIPH, PCIC_DISPLAY, PCIC_MEMORY, PCIC_STORAGE, PCIR_BAR, PCIR_BIOS, PCIR_CACHELNSZ, PCIR_CLASS, PCIR_COMMAND, PCIR_DEVICE, PCIR_INTLINE, PCIR_INTPIN, PCIR_LATTIMER, PCIR_MAXLAT, PCIR_MINGNT, PCIR_PROGIF, PCIR_REVID, PCIR_SUBCLASS, PCIR_SUBDEV_0, PCIR_SUBVEND_0, and PCIR_VENDOR.

Referenced by pci_add_child(), pci_driver_added(), pci_probe_nomatch(), and pci_suspend().

7.7.2.19 int pci_child_location_str_method (device_t *dev*, device_t *child*, char * *buf*, size_t *buflen*)

Definition at line 3147 of file pci.c.

7.7.2.20 int pci_child_pnpinfo_str_method (device_t *dev*, device_t *child*, char * *buf*, size_t *buflen*)

Definition at line 3157 of file pci.c.

References pcicfg::baseclass, pcicfg::device, pcicfg::progif, pcicfg::subclass, pcicfg::subdevice, pcicfg::subvendor, and pcicfg::vendor.

7.7.2.21 static __inline void pci_clear_command_bit (device_t *dev*, device_t *child*, uint16_t *bit*) [static]

Definition at line 1780 of file pci.c.

References PCIR_COMMAND.

Referenced by pci_disable_busmaster_method(), and pci_disable_io_method().

7.7.2.22 void pci_delete_resource (device_t *dev*, device_t *child*, int *type*, int *rid*)

Definition at line 3082 of file pci.c.

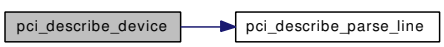
7.7.2.23 static char * pci_describe_device (device_t *dev*) [static]

Definition at line 2699 of file pci.c.

References desc, pci_describe_parse_line(), and pci_vendordata.

Referenced by pci_probe_nomatch().

Here is the call graph for this function:



7.7.2.24 static int pci_describe_parse_line (char ** *ptr*, int * *vendor*, int * *device*, char ** *desc*) [static]

Definition at line 2653 of file pci.c.

References pci_vendordata, and pci_vendordata_size.

Referenced by pci_describe_device().

7.7.2.25 int pci_disable_busmaster_method (device_t *dev*, device_t *child*)

Definition at line 1797 of file pci.c.

References pci_clear_command_bit(), and PCIM_CMD_BUSMASTEREN.

Here is the call graph for this function:

7.7.2.26 int pci_disable_io_method (device_t *dev*, device_t *child*, int *space*)

Definition at line 1835 of file pci.c.

References pci_clear_command_bit(), PCIM_CMD_MEMEN, PCIM_CMD_PORTEN, and PCIR_-COMMAND.

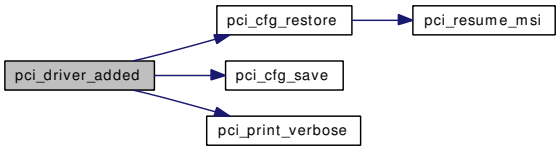
Here is the call graph for this function:

7.7.2.27 void pci_driver_added (device_t *dev*, driver_t * *driver*)

Definition at line 2441 of file pci.c.

References pci_cfg_restore(), pci_cfg_save(), and pci_print_verbose().

Here is the call graph for this function:



7.7.2.28 int pci_enable_busmaster_method (device_t *dev*, device_t *child*)

Definition at line 1790 of file pci.c.

References pci_set_command_bit(), and PCIM_CMD_BUSMASTEREN.

Here is the call graph for this function:

7.7.2.29 int pci_enable_io_method (device_t *dev*, device_t *child*, int *space*)

Definition at line 1804 of file pci.c.

References pci_set_command_bit(), PCIM_CMD_MEMEN, PCIM_CMD_PORTEN, and PCIR_-COMMAND.

Here is the call graph for this function:

7.7.2.30 void pci_enable_msi (device_t *dev*, uint64_t *address*, uint16_t *data*)

Definition at line 1338 of file pci.c.

References pcicfg::msi, pcicfg_msi::msi_addr, pcicfg_msi::msi_ctrl, pcicfg_msi::msi_data, pcicfg_-msi::msi_location, PCIM_MSICTRL_64BIT, PCIM_MSICTRL_MSI_ENABLE, PCIR_MSI_ADDR, PCIR_MSI_ADDR_HIGH, PCIR_MSI_CTRL, PCIR_MSI_DATA, and PCIR_MSI_DATA_64BIT.

7.7.2.31 void pci_enable_msix (device_t *dev*, u_int *index*, uint64_t *address*, uint32_t *data*)

Definition at line 1003 of file pci.c.

References pcicfg::msix, pcicfg_msix::msix_alloc, pcicfg_msix::msix_table_offset, and pcicfg_-msix::msix_table_res.

7.7.2.32 device_t pci_find_bsf (uint8_t *bus*, uint8_t *slot*, uint8_t *func*)

Definition at line 271 of file pci.c.

References pci_devq.

Referenced by pci_ioctl(), and pci_msi_blacklisted().

7.7.2.33 device_t pci_find_device (uint16_t *vendor*, uint16_t *device*)

Definition at line 289 of file pci.c.

References pci_devq.

7.7.2.34 int pci_find_extcap_method (device_t *dev*, device_t *child*, int *capability*, int * *capreg*)

Definition at line 952 of file pci.c.

References pcicfg::hdrtype, PCICAP_ID, PCICAP_NEXTPTR, PCIM_HDRTYPE, PCIM_STATUS_-CAPPRESENT, PCIR_CAP_PTR, PCIR_CAP_PTR_2, and PCIR_STATUS.

7.7.2.35 static void pci_fixancient (pcicfgregs * *cfg*) [static]

Definition at line 377 of file pci.c.

References pcicfg::baseclass, pcicfg::hdrtype, PCIC_BRIDGE, PCIS_BRIDGE_PCI, and pcicfg::subclass.

Referenced by pci_read_device().

7.7.2.36 int pci_freecfg (struct pci_devinfo * *dinfo*)

Definition at line 1626 of file pci.c.

References pci_devq, pci_generation, and pci_numdevs.

7.7.2.37 int pci_get_powerstate_method (device_t *dev*, device_t *child*)

Definition at line 1732 of file pci.c.

References PCIM_PSTAT_D0, PCIM_PSTAT_D1, PCIM_PSTAT_D2, PCIM_PSTAT_D3, PCIM_PSTAT_DMASK, pcicfg::pp, pcicfg_pp::pp_cap, and pcicfg_pp::pp_status.

7.7.2.38 struct resource_list* pci_get_resource_list (device_t *dev*, device_t *child*)

Definition at line 3118 of file pci.c.

7.7.2.39 int pci_get_vpd_ident_method (device_t *dev*, device_t *child*, const char ** *identptr*)

Definition at line 913 of file pci.c.

References pcicfg::vpd, and pcicfg_vpd::vpd_ident.

7.7.2.40 int pci_get_vpd_readonly_method (device_t *dev*, device_t *child*, const char * *kw*, const char ** *vptr*)

Definition at line 927 of file pci.c.

References vpd_readonly::keyword, vpd_readonly::value, pcicfg::vpd, pcicfg_vpd::vpd_rocnt, and pcicfg_vpd::vpd_ros.

7.7.2.41 static void pci_hdrtypedata (device_t *pcib*, int *b*, int *s*, int *f*, pcicfgregs * *cfg*) [static]

Definition at line 390 of file pci.c.

References pcicfg::hdrtype, pcicfg::nummaps, PCI_MAXMAPS_0, PCI_MAXMAPS_1, PCI_MAXMAPS_2, PCIR_SUBDEV_0, PCIR_SUBDEV_2, PCIR_SUBVEND_0, PCIR_SUBVEND_2, REG, pcicfg::subdevice, and pcicfg::subvendor.

Referenced by pci_read_device().

7.7.2.42 static void pci_load_vendor_data (void) [static]

Definition at line 2426 of file pci.c.

References pci_vendordata, and pci_vendordata_size.

Referenced by pci_modevent().

7.7.2.43 static uint32_t pci_mapbase (uint32_t *mapreg*) [static]

Definition at line 306 of file pci.c.

7.7.2.44 static uint32_t pci_mapbase (unsigned *mapreg*) [static]

Referenced by pci_add_map(), pci_alloc_map(), pci_ata_maps(), and pci_mapsize().

7.7.2.45 static int pci_maprange (unsigned *mapreg*) [static]

Definition at line 355 of file pci.c.

Referenced by pci_add_map(), and pci_alloc_map().

7.7.2.46 static int pci_mapsize (uint32_t *testval*) [static]

Definition at line 336 of file pci.c.

References pci_mapbase().

Here is the call graph for this function:

7.7.2.47 static int pci_mapsize (unsigned *testval*) [static]

Referenced by pci_add_map(), and pci_alloc_map().

7.7.2.48 static int pci_maptype (unsigned *mapreg*) [static]

Definition at line 317 of file pci.c.

References PCI_MAPMEM, PCI_MAPMEMP, and PCI_MAPPORT.

Referenced by pci_add_map(), and pci_alloc_map().

7.7.2.49 void pci_mask_msix (device_t *dev*, u_int *index*)

Definition at line 1017 of file pci.c.

References pcicfg::msix, pcicfg_msix::msix_msgnum, pcicfg_msix::msix_table_offset, pcicfg_msix::msix_table_res, and PCIM_MSIX_VCTRL_MASK.

7.7.2.50 static int pci_memen (device_t *pcib*, int *b*, int *s*, int *f*) [static]

Definition at line 1961 of file pci.c.

References PCIM_CMD_MEMEN, and PCIR_COMMAND.

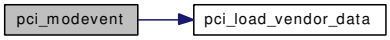
Referenced by pci_add_map().

7.7.2.51 static int pci_modevent (module_t *mod*, int *what*, void * *arg*) [static]

Definition at line 3183 of file pci.c.

References pci_devq, pci_generation, pci_load_vendor_data(), and pcicdev.

Here is the call graph for this function:



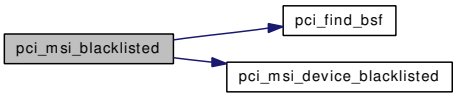
7.7.2.52 static int pci_msi_blacklisted (void) [static]

Definition at line 1422 of file pci.c.

References pci_find_bsf(), pci_honor_msi_blacklist, pci_msi_device_blacklisted(), pcie_chipset, and pcix_chipset.

Referenced by pci_alloc_msi_method(), and pci_alloc_msix_method().

Here is the call graph for this function:



7.7.2.53 int pci_msi_count_method (device_t *dev*, device_t *child*)

Definition at line 1613 of file pci.c.

References pcicfg::msi, pcicfg_msi::msi_location, pcicfg_msi::msi_msgnum, and pci_do_msi.

7.7.2.54 int pci_msi_device_blacklisted (device_t *dev*)

Definition at line 1399 of file pci.c.

References pci_quirk::devid, pci_honor_msi_blacklist, PCI_QUIRK_DISABLE_MSI, pci_quirks, and pci_quirk::type.

Referenced by pci_msi_blacklisted(), and pcib_attach_common().

7.7.2.55 int pci_msix_count_method (device_t *dev*, device_t *child*)

Definition at line 1324 of file pci.c.

References pcicfg::msix, pcicfg_msix::msix_location, pcicfg_msix::msix_msgnum, and pci_do_msix.

7.7.2.56 int pci_pending_msix (device_t *dev*, u_int *index*)

Definition at line 1049 of file pci.c.

References pcicfg::msix, pcicfg_msix::msix_alloc, pcicfg_msix::msix_pba_offset, and pcicfg_-msix::msix_pba_res.

7.7.2.57 static int pci_porten (device_t *pcib*, int *b*, int *s*, int *f*) [static]

Definition at line 1954 of file pci.c.

References PCIM_CMD_PORTEN, and PCIR_COMMAND.

Referenced by pci_add_map().

7.7.2.58 int pci_print_child (device_t *dev*, device_t *child*)

Definition at line 2470 of file pci.c.

7.7.2.59 void pci_print_verbose (struct pci_devinfo * *dinfo*)

Definition at line 1871 of file pci.c.

References pcicfg::baseclass, pcicfg::bus, pcicfg::cachelnsz, pcicfg::cmdreg, pcicfg::dev, pcicfg::device, pcicfg::func, pcicfg::hdrtype, pcicfg::intline, pcicfg::intpin, pcicfg::lattimer, pcicfg::maxlat, pcicfg::mfdev, pcicfg::mingnt, pcicfg::msi, pcicfg_msi::msi_ctrl, pcicfg_msi::msi_location, pcicfg_msi::msi_msgnum, pcicfg::msix, pcicfg_msix::msix_location, pcicfg_msix::msix_msgnum, pcicfg_msix::msix_pba_bar, pcicfg_msix::msix_table_bar, PCIM_MSICTRL_64BIT, PCIM_MSICTRL_VECTOR, PCIM_PCAP_D1SUPP, PCIM_PCAP_D2SUPP, PCIM_PCAP_SPEC, PCIM_PSTAT_DMASK, pcicfg::pp, pcicfg_pp::pp_cap, pcicfg_pp::pp_status, pcicfg::progif, pcicfg::revid, pcicfg::slot, pcicfg::statreg, pcicfg::subclass, pcicfg::vendor, pcicfg::vpd, pcicfg_vpd::vpd_ident, pcicfg_vpd::vpd_reg, pcicfg_vpd::vpd_rocnt, pcicfg_vpd::vpd_ros, pcicfg_vpd::vpd_w, and pcicfg_vpd::vpd_wcnt.

Referenced by pci_add_child(), and pci_driver_added().

7.7.2.60 static int pci_probe (device_t *dev*) [static]

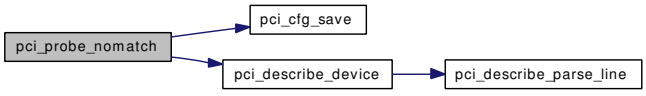
Definition at line 2313 of file pci.c.

7.7.2.61 void pci_probe_nomatch (device_t *dev*, device_t *child*)

Definition at line 2583 of file pci.c.

References desc, pci_cfg_save(), pci_describe_device(), pci_do_power_nodriver, pci_nomatch_tab, and subclass.

Here is the call graph for this function:



7.7.2.62 uint32_t pci_read_config_method (device_t *dev*, device_t *child*, int *reg*, int *width*)

Definition at line 3126 of file pci.c.

References pcicfg::bus, pcicfg::func, and pcicfg::slot.

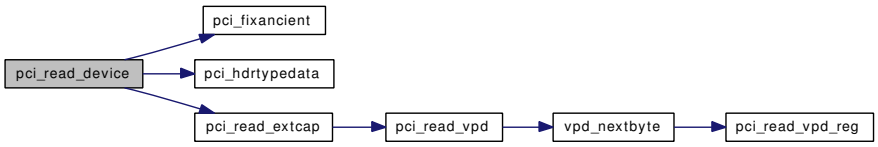
7.7.2.63 struct pci_devinfo* pci_read_device (device_t *pcib*, int *b*, int *s*, int *f*, size_t *size*)

Definition at line 413 of file pci.c.

References pcicfg::baseclass, pcicfg::bus, pcicfg::cachelnsz, pcicfg::cmdreg, pcicfg::device, pcicfg::func, pcicfg::hdrtype, pcicfg::intline, pcicfg::intpin, pcicfg::lattimer, pcicfg::maxlat, pcicfg::mfdev, pcicfg::mingnt, pci_devq, pci_fixancient(), pci_generation, pci_hdrtypedata(), pci_numdevs, pci_read_extcap(), PCIM_MFDEV, PCIM_STATUS_CAPPRESENT, PCIR_CACHELNSZ, PCIR_CLASS, PCIR_COMMAND, PCIR_DEVICE, PCIR_DEVVENDOR, PCIR_HDRTYPE, PCIR_INTLINE, PCIR_INTPIN, PCIR_LATTIMER, PCIR_MAXLAT, PCIR_MINGNT, PCIR_PROGIF, PCIR_REVID, PCIR_STATUS, PCIR_SUBCLASS, PCIR_VENDOR, pcicfg::progif, REG, pcicfg::revid, pcicfg::slot, pcicfg::statreg, pcicfg::subclass, pcicfg::subdevice, pcicfg::subvendor, and pcicfg::vendor.

Referenced by pci_add_children().

Here is the call graph for this function:



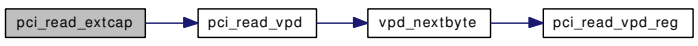
7.7.2.64 static void pci_read_extcap (device_t *pcib*, pcicfgregs * *cfg*) [static]

Definition at line 485 of file pci.c.

References pcicfg::bus, pcicfg::func, pcicfg::hdrtype, pcicfg::msi, pcicfg_msi::msi_ctrl, pcicfg_msi::msi_location, pcicfg_msi::msi_msgnum, pcicfg::msix, pcicfg_msix::msix_ctrl, pcicfg_msix::msix_location, pcicfg_msix::msix_msgnum, pcicfg_msix::msix_pba_bar, pcicfg_msix::msix_pba_offset, pcicfg_msix::msix_table_bar, pcicfg_msix::msix_table_offset, pci_do_vpd, pci_read_vpd(), PCICAP_ID, PCICAP_NEXTPTR, pcie_chipset, PCIM_EXP_FLAGS_TYPE, PCIM_EXP_TYPE_ROOT_PORT, PCIM_HDRTYPE, PCIM_HTCAP_MSI_MAPPING, PCIM_HTCMD_CAP_MASK, PCIM_HTCMD_MSI_ENABLE, PCIM_MSICTRL_MMC_MASK, PCIM_MSIX_BIR_MASK, PCIM_MSIXCTRL_TABLE_SIZE, PCIR_BAR, PCIR_CAP_PTR, PCIR_CAP_PTR_2, PCIR_EXPRESS_FLAGS, PCIR_HT_COMMAND, PCIR_HTMSI_ADDRESS_HI, PCIR_HTMSI_ADDRESS_LO, PCIR_MSI_CTRL, PCIR_MSIX_CTRL, PCIR_MSIX_PBA, PCIR_MSIX_TABLE, PCIR_POWER_CAP, PCIR_POWER_DATA, PCIR_POWER_PMCSR, PCIR_POWER_STATUS, PCIR_SUBVENDCAP_ID, pcix_chipset, PCIY_EXPRESS, PCIY_HT, PCIY_MSI, PCIY_MSIX, PCIY_PCIX, PCIY_PMG, PCIY_SUBVENDOR, PCIY_VPD, pcicfg::pp, pcicfg_pp::pp_cap, pcicfg_pp::pp_data, pcicfg_pp::pp_pmcsr, pcicfg_pp::pp_status, REG, pcicfg::slot, pcicfg::subdevice, pcicfg::subvendor, pcicfg::vpd, pcicfg_vpd::vpd_reg, and WREG.

Referenced by pci_read_device().

Here is the call graph for this function:



7.7.2.65 int pci_read_ivar (device_t *dev*, device_t *child*, int *which*, uintptr_t * *result*)

Definition at line 2752 of file pci.c.

References pcicfg::baseclass, pcicfg::bus, pcicfg::cachelnsz, pcicfg::cmdreg, pcicfg::device, pcicfg::func, pcicfg::intline, pcicfg::intpin, pcicfg::lattimer, pcicfg::maxlat, pcicfg::mingnt, pcicfg::progif, pcicfg::revid, pcicfg::slot, pcicfg::subclass, pcicfg::subdevice, pcicfg::subvendor, and pcicfg::vendor.

7.7.2.66 static void pci_read_vpd (device_t *pcib*, pcicfgregs * *cfg*) [static]

Definition at line 680 of file pci.c.

References pcicfg::bus, vpd_readstate::bytesinval, vpd_readstate::cfg, vpd_readstate::cksum, pcicfg::func, vpd_write::keyword, vpd_readonly::keyword, vpd_write::len, vpd_readstate::off, vpd_readstate::pcib, pcicfg::slot, vpd_write::start, vpd_readstate::val, vpd_write::value, vpd_readonly::value, pcicfg::vpd, pcicfg_vpd::vpd_ident, vpd_nextbyte(), pcicfg_vpd::vpd_rocnt, pcicfg_vpd::vpd_ros, pcicfg_vpd::vpd_w, and pcicfg_vpd::vpd_wcnt.

Referenced by pci_read_extcap().

Here is the call graph for this function:

7.7.2.67 static uint32_t pci_read_vpd_reg (device_t *pcib*, pcicfgregs * *cfg*, int *reg*) [static]

Definition at line 622 of file pci.c.

References REG, pcicfg::vpd, pcicfg_vpd::vpd_reg, and WREG.

Referenced by vpd_nextbyte().

7.7.2.68 int pci_release_msi_method (device_t *dev*, device_t *child*)

Definition at line 1564 of file pci.c.

References pcicfg::msi, pcicfg_msi::msi_alloc, pcicfg_msi::msi_ctrl, pcicfg_msi::msi_location, pci_release_msix(), PCIM_MSICTRL_MME_MASK, PCIM_MSICTRL_MSI_ENABLE, and PCIR_MSI_CTRL.

Here is the call graph for this function:

7.7.2.69 static int pci_release_msix (device_t *dev*, device_t *child*) [static]

Definition at line 1275 of file pci.c.

References pcicfg::msix, pcicfg_msix::msix_alloc, pcicfg_msix::msix_ctrl, pcicfg_msix::msix_location, PCIM_MSIXCTRL_MSIX_ENABLE, and PCIR_MSIX_CTRL.

Referenced by pci_release_msi_method().

7.7.2.70 int pci_remap_msix_method (device_t *dev*, device_t *child*, u_int * *indices*)

Definition at line 1203 of file pci.c.

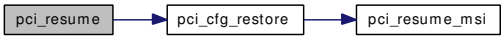
References pcicfg::msix, pcicfg_msix::msix_alloc, and pcicfg_msix::msix_msgnum.

7.7.2.71 int pci_resume (device_t *dev*)

Definition at line 2391 of file pci.c.

References ACPI_PWR_FOR_SLEEP, pci_cfg_restore(), and pci_do_power_resume.

Here is the call graph for this function:



7.7.2.72 static void pci_resume_msi (device_t *dev*) [static]

Definition at line 1369 of file pci.c.

References pcicfg::msi, pcicfg_msi::msi_addr, pcicfg_msi::msi_ctrl, pcicfg_msi::msi_data, pcicfg_msi::msi_location, PCIM_MSICTRL_64BIT, PCIM_MSICTRL_MSI_ENABLE, PCIR_MSI_ADDR, PCIR_MSI_ADDR_HIGH, PCIR_MSI_CTRL, PCIR_MSI_DATA, and PCIR_MSI_DATA_64BIT.

Referenced by pci_cfg_restore().

7.7.2.73 static __inline void pci_set_command_bit (device_t *dev*, device_t *child*, uint16_t *bit*) [static]

Definition at line 1770 of file pci.c.

References PCIR_COMMAND.

Referenced by pci_enable_busmaster_method(), and pci_enable_io_method().

7.7.2.74 int pci_set_powerstate_method (device_t *dev*, device_t *child*, int *state*)

Definition at line 1657 of file pci.c.

References PCIM_PCAP_D1SUPP, PCIM_PCAP_D2SUPP, PCIM_PSTAT_D0, PCIM_PSTAT_D1, PCIM_PSTAT_D2, PCIM_PSTAT_D3, PCIM_PSTAT_DMASK, pcicfg::pp, pcicfg_pp::pp_cap, and pcicfg_pp::pp_status.

7.7.2.75 int pci_suspend (device_t *dev*)

Definition at line 2343 of file pci.c.

References ACPI_PWR_FOR_SLEEP, pci_cfg_save(), and pci_do_power_resume.

Here is the call graph for this function:

7.7.2.76 void pci_unmask_msix (device_t *dev*, u_int *index*)

Definition at line 1033 of file pci.c.

References pcicfg::msix, pcicfg_msix::msix_alloc, pcicfg_msix::msix_table_offset, pcicfg_msix::msix_table_res, and PCIM_MSIX_VCTRL_MASK.

7.7.2.77 void pci_write_config_method (device_t *dev*, device_t *child*, int *reg*, uint32_t *val*, int *width*)

Definition at line 3136 of file pci.c.

References pcicfg::bus, pcicfg::func, and pcicfg::slot.

7.7.2.78 int pci_write_ivar (device_t *dev*, device_t *child*, int *which*, uintptr_t *value*)

Definition at line 2832 of file pci.c.

7.7.2.79 SYSCTL_INT (_hw_pci, OID_AUTO, honor_msi_blacklist, CTLFLAG_RD, & *pci_honor_msi_blacklist*, 1, "Honor chipset blacklist for MSI")

7.7.2.80 SYSCTL_INT (_hw_pci, OID_AUTO, enable_msix, CTLFLAG_RW, & *pci_do_msix*, 1, "Enable support for MSI-X interrupts")

7.7.2.81 SYSCTL_INT (_hw_pci, OID_AUTO, enable_msi, CTLFLAG_RW, & *pci_do_msi*, 1, "Enable support for MSI interrupts")

7.7.2.82 SYSCTL_INT (_hw_pci, OID_AUTO, enable_vpd, CTLFLAG_RW, & *pci_do_vpd*, 1, "Enable support for VPD.")

7.7.2.83 SYSCTL_INT (_hw_pci, OID_AUTO, do_power_resume, CTLFLAG_RW, & *pci_do_power_resume*, 1, "Transition from D3 → D0 on resume.")

7.7.2.84 SYSCTL_INT (_hw_pci, OID_AUTO, do_power_nodriver, CTLFLAG_RW, & *pci_do_power_nodriver*, 0, "Place a function into D3 state when no driver attaches to it. 0 means\n\disable. 1 means conservatively place devices into D3 state. 2 means\n\agressively place devices into D3 state. 3 means put absolutely everything\n\in D3 state.")

7.7.2.85 SYSCTL_INT (_hw_pci, OID_AUTO, enable_io_modes, CTLFLAG_RW, & *pci_enable_io_modes*, 1, "Enable I/O and memory bits in the config register. Some BIOSes do not\n\enable these bits correctly. We'd like to do this all the *time*, but there\n\are some peripherals that this causes problems with.")

7.7.2.86 SYSCTL_NODE (_hw, OID_AUTO, pci, CTLFLAG_RD, 0, "PCI bus tuning parameters")

7.7.2.87 TUNABLE_INT ("hw.pci.honor_msi_blacklist", & *pci_honor_msi_blacklist*)

7.7.2.88 TUNABLE_INT ("hw.pci.enable_msix", & *pci_do_msix*)

7.7.2.89 TUNABLE_INT ("hw.pci.enable_msi", & *pci_do_msi*)

7.7.2.90 TUNABLE_INT ("hw.pci.enable_vpd", & *pci_do_vpd*)

7.7.2.91 TUNABLE_INT ("hw.pci.do_power_resume", & *pci_do_power_resume*)

7.7.2.92 TUNABLE_INT ("hw.pci.do_power_nodriver", & *pci_do_power_nodriver*)

7.7.2.93 TUNABLE_INT ("hw.pci.enable_io_modes", & *pci_enable_io_modes*)

7.7.2.94 static uint8_t vpd_nextbyte (struct vpd_readstate * *vrs*) `[static]`

Definition at line 659 of file pci.c.

References vpd_readstate::bytesinval, vpd_readstate::cfg, vpd_readstate::cksum, vpd_readstate::off, pci_-read_vpd_reg(), vpd_readstate::pcib, and vpd_readstate::val.

Referenced by pci_read_vpd().

Here is the call graph for this function:

7.7.3 Variable Documentation

7.7.3.1 int class

Definition at line 2497 of file pci.c.

7.7.3.2 char* desc

Definition at line 2499 of file pci.c.

Referenced by pci_describe_device(), and pci_probe_nomatch().

7.7.3.3 devclass_t pci_devclass [static]

Definition at line 160 of file pci.c.

Referenced by legacy_pcib_identify().

7.7.3.4 struct devlist pci_devq

Definition at line 217 of file pci.c.

Referenced by pci_find_bsf(), pci_find_device(), pci_freecfg(), pci_ioctl(), pci_modevent(), and pci_read_device().

7.7.3.5 int pci_do_msi = 1 [static]

Definition at line 253 of file pci.c.

Referenced by pci_alloc_msi_method(), and pci_msi_count_method().

7.7.3.6 int pci_do_msix = 1 [static]

Definition at line 258 of file pci.c.

Referenced by pci_alloc_msix_method(), and pci_msix_count_method().

7.7.3.7 int pci_do_power_nodriver = 0 [static]

Definition at line 233 of file pci.c.

Referenced by pci_cfg_save(), and pci_probe_nomatch().

7.7.3.8 int pci_do_power_resume = 1 [static]

Definition at line 242 of file pci.c.

Referenced by pci_resume(), and pci_suspend().

7.7.3.9 int pci_do_vpd = 1 [static]

Definition at line 248 of file pci.c.

Referenced by pci_read_extcap().

7.7.3.10 int pci_enable_io_modes = 1 [static]

Definition at line 225 of file pci.c.

Referenced by pci_add_map().

7.7.3.11 uint32_t pci_generation

Definition at line 218 of file pci.c.

Referenced by pci_freecfg(), pci_ioctl(), pci_modevent(), and pci_read_device().

7.7.3.12 int pci_honor_msi_blacklist = 1 [static]

Definition at line 263 of file pci.c.

Referenced by pci_msi_blacklisted(), and pci_msi_device_blacklisted().

7.7.3.13 device_method_t pci_methods[ ] [static]

Definition at line 106 of file pci.c.

7.7.3.14 struct { ... } pci_nomatch_tab[ ] [static]

Referenced by pci_probe_nomatch().

7.7.3.15 uint32_t pci_numdevs = 0

Definition at line 219 of file pci.c.

Referenced by pci_freecfg(), pci_ioctl(), and pci_read_device().

7.7.3.16 struct pci_quirk pci_quirks[ ]

Initial value:

```
{

        { 0x71138086, PCI_QUIRK_MAP_REG,        0x90,     0 },
        { 0x719b8086, PCI_QUIRK_MAP_REG,        0x90,     0 },

        { 0x02001166, PCI_QUIRK_MAP_REG,        0x90,     0 },


        { 0x00141166, PCI_QUIRK_DISABLE_MSI,    0,        0 },
        { 0x00171166, PCI_QUIRK_DISABLE_MSI,    0,        0 },
```

```
        { 0x25408086, PCI_QUIRK_DISABLE_MSI,    0,      0 },
        { 0x254c8086, PCI_QUIRK_DISABLE_MSI,    0,      0 },
        { 0x25508086, PCI_QUIRK_DISABLE_MSI,    0,      0 },
        { 0x25608086, PCI_QUIRK_DISABLE_MSI,    0,      0 },
        { 0x25708086, PCI_QUIRK_DISABLE_MSI,    0,      0 },
        { 0x25788086, PCI_QUIRK_DISABLE_MSI,    0,      0 },
        { 0x35808086, PCI_QUIRK_DISABLE_MSI,    0,      0 },


        { 0x74501022, PCI_QUIRK_DISABLE_MSI,    0,      0 },

        { 0 }
}
```

Definition at line 177 of file pci.c.

Referenced by pci_add_resources(), and pci_msi_device_blacklisted().

7.7.3.17 char* pci_vendordata [static]

Definition at line 164 of file pci.c.

Referenced by pci_describe_device(), pci_describe_parse_line(), and pci_load_vendor_data().

7.7.3.18 size_t pci_vendordata_size [static]

Definition at line 165 of file pci.c.

Referenced by pci_describe_parse_line(), and pci_load_vendor_data().

7.7.3.19 int pcie_chipset [static]

Definition at line 220 of file pci.c.

Referenced by pci_msi_blacklisted(), and pci_read_extcap().

7.7.3.20 int pcix_chipset [static]

Definition at line 220 of file pci.c.

Referenced by pci_msi_blacklisted(), and pci_read_extcap().

7.7.3.21 int subclass

Definition at line 2498 of file pci.c.

Referenced by legacy_pcib_identify(), and pci_probe_nomatch().

7.8 /usr/src/sys/dev/pci/pci_if.m File Reference

```
#include <sys/bus.h>
```

Include dependency graph for pci_if.m:

Variables

- INTERFACE pci

7.8.1 Variable Documentation

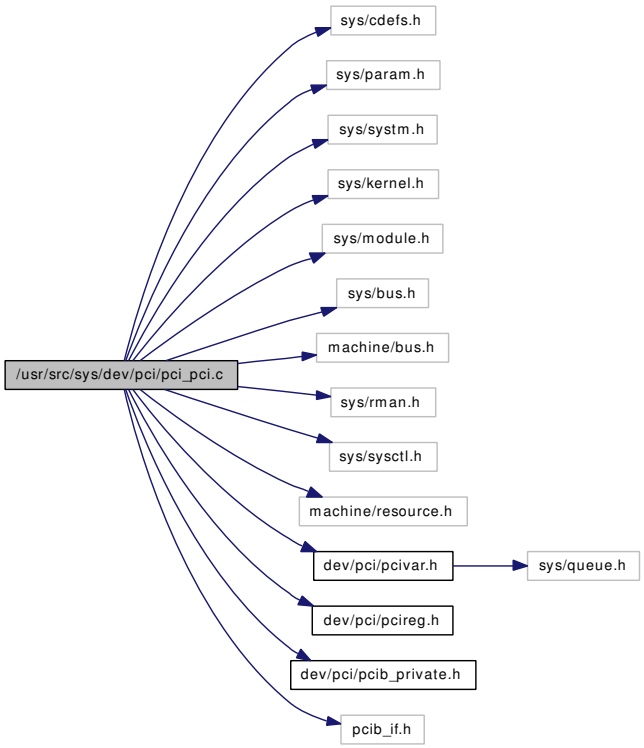
7.8.1.1 INTERFACE pci

Definition at line 31 of file pci_if.m.

7.9 /usr/src/sys/dev/pci/pci_pci.c File Reference

```
#include <sys/cdefs.h>
#include <sys/param.h>
#include <sys/systm.h>
#include <sys/kernel.h>
#include <sys/module.h>
#include <sys/bus.h>
#include <machine/bus.h>
#include <sys/rman.h>
#include <sys/sysctl.h>
#include <machine/resource.h>
#include <dev/pci/pcivar.h>
#include <dev/pci/pcireg.h>
#include <dev/pci/pcib_private.h>
#include "pcib_if.h"
```

Include dependency graph for pci_pci.c:



Functions

- __FBSDID ("$FreeBSD: src/sys/dev/pci/pci_pci.c,v 1.46 2007/01/22 21:48:43 jhb Exp $")

- static int pcib_probe (device_t dev)
- DEFINE_CLASS_0 (pcib, pcib_driver, pcib_methods, sizeof(struct pcib_softc))
- DRIVER_MODULE (pcib, pci, pcib_driver, pcib_devclass, 0, 0)
- static int pcib_is_prefetch_open (struct pcib_softc *sc)
- static int pcib_is_nonprefetch_open (struct pcib_softc *sc)
- static int pcib_is_io_open (struct pcib_softc *sc)
- void pcib_attach_common (device_t dev)
- int pcib_attach (device_t dev)
- int pcib_read_ivar (device_t dev, device_t child, int which, uintptr_t *result)
- int pcib_write_ivar (device_t dev, device_t child, int which, uintptr_t value)
- resource * pcib_alloc_resource (device_t dev, device_t child, int type, int *rid, u_long start, u_long end, u_long count, u_int flags)
- int pcib_maxslots (device_t dev)
- uint32_t pcib_read_config (device_t dev, int b, int s, int f, int reg, int width)
- void pcib_write_config (device_t dev, int b, int s, int f, int reg, uint32_t val, int width)
- int pcib_route_interrupt (device_t pcib, device_t dev, int pin)
- int pcib_alloc_msi (device_t pcib, device_t dev, int count, int maxcount, int *irqs)
- int pcib_release_msi (device_t pcib, device_t dev, int count, int *irqs)
- int pcib_alloc_msix (device_t pcib, device_t dev, int index, int *irq)
- int pcib_remap_msix (device_t pcib, device_t dev, int index, int irq)
- int pcib_release_msix (device_t pcib, device_t dev, int irq)
- int host_pcib_get_busno (pci_read_config_fn read_config, int bus, int slot, int func, uint8_t *busnum)

Variables

- static device_method_t pcib_methods [ ]
- static devclass_t pcib_devclass

7.9.1 Function Documentation

7.9.1.1 __FBSDID ("$FreeBSD: src/sys/dev/pci/pci_pci. *c*, v 1.46 2007/01/22 21:48:43 jhb Exp $")

7.9.1.2 DEFINE_CLASS_0 (pcib, pcib_driver, pcib_methods, sizeof(struct pcib_softc))

7.9.1.3 DRIVER_MODULE (pcib, pci, pcib_driver, pcib_devclass, 0, 0)

7.9.1.4 int host_pcib_get_busno (pci_read_config_fn *read_config*, int *bus*, int *slot*, int *func*, uint8_t * *busnum*)

Definition at line 612 of file pci_pci.c.

References PCIR_DEVVENDOR.

7.9.1.5 int pcib_alloc_msi (device_t *pcib*, device_t *dev*, int *count*, int *maxcount*, int * *irqs*)

Definition at line 552 of file pci_pci.c.

References pcib_softc::flags, and PCIB_DISABLE_MSI.

7.9.1.6 int pcib_alloc_msix (device_t *pcib*, device_t *dev*, int *index*, int * *irq*)

Definition at line 576 of file pci_pci.c.

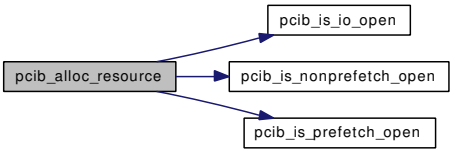
References pcib_softc::flags, and PCIB_DISABLE_MSI.

7.9.1.7 struct resource* pcib_alloc_resource (device_t *dev*, device_t *child*, int *type*, int * *rid*, u_long *start*, u_long *end*, u_long *count*, u_int *flags*)

Definition at line 337 of file pci_pci.c.

References pcib_softc::bridgectl, pcib_softc::flags, pcib_softc::iobase, pcib_softc::iolimit, pcib_softc::membase, pcib_softc::memlimit, PCIB_BCR_VGA_ENABLE, pcib_is_io_open(), pcib_is_nonprefetch_open(), pcib_is_prefetch_open(), PCIB_SUBTRACTIVE, pcib_softc::pmembase, and pcib_softc::pmemlimit.

Here is the call graph for this function:

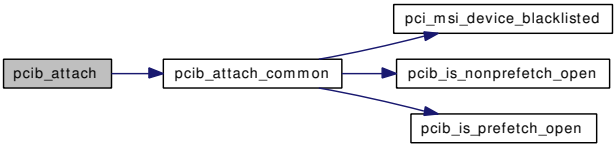


7.9.1.8 int pcib_attach (device_t *dev*)

Definition at line 289 of file pci_pci.c.

References pcib_attach_common(), and pcib_softc::secbus.

Here is the call graph for this function:



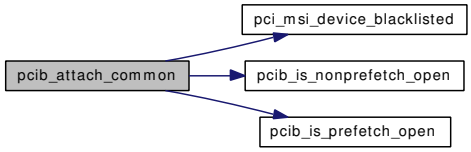
7.9.1.9 void pcib_attach_common (device_t *dev*)

Definition at line 138 of file pci_pci.c.

References pcib_softc::bridgectl, pcib_softc::command, pcib_softc::dev, pcib_softc::flags, pcib_softc::iobase, pcib_softc::iolimit, pcib_softc::membase, pcib_softc::memlimit, pci_msi_device_blacklisted(), PCI_PPBIOBASE, PCI_PPBIOLIMIT, PCI_PPBMEMBASE, PCI_PPBMEMLIMIT, PCIB_DISABLE_MSI, pcib_is_nonprefetch_open(), pcib_is_prefetch_open(), PCIB_SUBTRACTIVE, PCIM_BRIO_32, PCIM_BRIO_MASK, PCIM_CMD_MEMEN, PCIM_CMD_PORTEN, PCIR_BRIDGECTL_1, PCIR_COMMAND, PCIR_IOBASEH_1, PCIR_IOBASEL_1, PCIR_IOLIMITH_1, PCIR_IOLIMITL_1, PCIR_MEMBASE_1, PCIR_MEMLIMIT_1, PCIR_PMBASEH_1, PCIR_PMBASEL_1, PCIR_PMLIMITH_1, PCIR_PMLIMITL_1, PCIR_PROGIF, PCIR_SECBUS_1, PCIR_SECLAT_1, PCIR_SECSTAT_1, PCIR_SUBBUS_1, pcib_softc::pmembase, pcib_softc::pmemlimit, pcib_softc::secbus, pcib_softc::seclat, pcib_softc::secstat, and pcib_softc::subbus.

Referenced by pcib_attach().

Here is the call graph for this function:



7.9.1.10 static int pcib_is_io_open (struct pcib_softc * *sc*) [static]

Definition at line 118 of file pci_pci.c.

References pcib_softc::iobase, and pcib_softc::iolimit.

Referenced by pcib_alloc_resource().

7.9.1.11 static int pcib_is_nonprefetch_open (struct pcib_softc * *sc*) [static]

Definition at line 109 of file pci_pci.c.

References pcib_softc::membase, and pcib_softc::memlimit.

Referenced by pcib_alloc_resource(), and pcib_attach_common().

7.9.1.12 static int pcib_is_prefetch_open (struct pcib_softc * *sc*) [static]

Definition at line 100 of file pci_pci.c.

References pcib_softc::pmembase, and pcib_softc::pmemlimit.

Referenced by pcib_alloc_resource(), and pcib_attach_common().

7.9.1.13 int pcib_maxslots (device_t *dev*)

Definition at line 494 of file pci_pci.c.

References PCI_SLOTMAX.

7.9.1.14 static int pcib_probe (device_t *dev*) [static]

Definition at line 127 of file pci_pci.c.

References PCIC_BRIDGE, and PCIS_BRIDGE_PCI.

7.9.1.15 uint32_t pcib_read_config (device_t *dev*, int *b*, int *s*, int *f*, int *reg*, int *width*)

Definition at line 503 of file pci_pci.c.

7.9.1.16 int pcib_read_ivar (device_t *dev*, device_t *child*, int *which*, uintptr_t * *result*)

Definition at line 307 of file pci_pci.c.

References pcib_softc::secbus.

7.9.1.17 int pcib_release_msi (device_t *pcib*, device_t *dev*, int *count*, int * *irqs*)

Definition at line 566 of file pci_pci.c.

7.9.1.18 int pcib_release_msix (device_t *pcib*, device_t *dev*, int *irq*)

Definition at line 599 of file pci_pci.c.

7.9.1.19 int pcib_remap_msix (device_t *pcib*, device_t *dev*, int *index*, int *irq*)

Definition at line 589 of file pci_pci.c.

7.9.1.20 int pcib_route_interrupt (device_t *pcib*, device_t *dev*, int *pin*)

Definition at line 518 of file pci_pci.c.

7.9.1.21 void pcib_write_config (device_t *dev*, int *b*, int *s*, int *f*, int *reg*, uint32_t *val*, int *width*)

Definition at line 509 of file pci_pci.c.

7.9.1.22 int pcib_write_ivar (device_t *dev*, device_t *child*, int *which*, uintptr_t *value*)

Definition at line 320 of file pci_pci.c.

References pcib_softc::secbus.

7.9.2 Variable Documentation

7.9.2.1 devclass_t pcib_devclass [static]

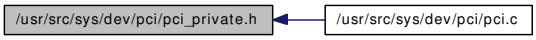
Definition at line 91 of file pci_pci.c.

7.9.2.2 device_method_t pcib_methods[ ] [static]

Definition at line 57 of file pci_pci.c.

7.10 /usr/src/sys/dev/pci/pci_private.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- DECLARE_CLASS (pci_driver)
- void pci_add_children (device_t dev, int busno, size_t dinfo_size)
- void pci_add_child (device_t bus, struct pci_devinfo *dinfo)
- void pci_add_resources (device_t bus, device_t dev, int force, uint32_t prefetchmask)
- void pci_driver_added (device_t dev, driver_t *driver)
- int pci_print_child (device_t dev, device_t child)
- void pci_probe_nomatch (device_t dev, device_t child)
- int pci_read_ivar (device_t dev, device_t child, int which, uintptr_t *result)
- int pci_write_ivar (device_t dev, device_t child, int which, uintptr_t value)
- int pci_get_vpd_ident_method (device_t dev, device_t child, const char **identptr)
- int pci_get_vpd_readonly_method (device_t dev, device_t child, const char *kw, const char **vptr)
- int pci_set_powerstate_method (device_t dev, device_t child, int state)
- int pci_get_powerstate_method (device_t dev, device_t child)
- uint32_t pci_read_config_method (device_t dev, device_t child, int reg, int width)
- void pci_write_config_method (device_t dev, device_t child, int reg, uint32_t val, int width)
- int pci_enable_busmaster_method (device_t dev, device_t child)
- int pci_disable_busmaster_method (device_t dev, device_t child)
- int pci_enable_io_method (device_t dev, device_t child, int space)
- int pci_disable_io_method (device_t dev, device_t child, int space)
- int pci_find_extcap_method (device_t dev, device_t child, int capability, int *capreg)
- int pci_alloc_msi_method (device_t dev, device_t child, int *count)
- int pci_alloc_msix_method (device_t dev, device_t child, int *count)
- int pci_remap_msix_method (device_t dev, device_t child, u_int *indices)
- int pci_release_msi_method (device_t dev, device_t child)
- int pci_msi_count_method (device_t dev, device_t child)
- int pci_msix_count_method (device_t dev, device_t child)
- resource * pci_alloc_resource (device_t dev, device_t child, int type, int *rid, u_long start, u_long end, u_long count, u_int flags)
- void pci_delete_resource (device_t dev, device_t child, int type, int rid)
- resource_list * pci_get_resource_list (device_t dev, device_t child)
- pci_devinfo * pci_read_device (device_t pcib, int b, int s, int f, size_t size)
- void pci_print_verbose (struct pci_devinfo *dinfo)
- int pci_freecfg (struct pci_devinfo *dinfo)
- int pci_child_location_str_method (device_t cbdev, device_t child, char *buf, size_t buflen)
- int pci_child_pnpinfo_str_method (device_t cbdev, device_t child, char *buf, size_t buflen)
- int pci_assign_interrupt_method (device_t dev, device_t child)
- int pci_resume (device_t dev)
- int pci_suspend (device_t dev)
- void pci_cfg_restore (device_t, struct pci_devinfo *)
- void pci_cfg_save (device_t, struct pci_devinfo *, int)

7.10.1 Function Documentation

7.10.1.1 DECLARE_CLASS (pci_driver)

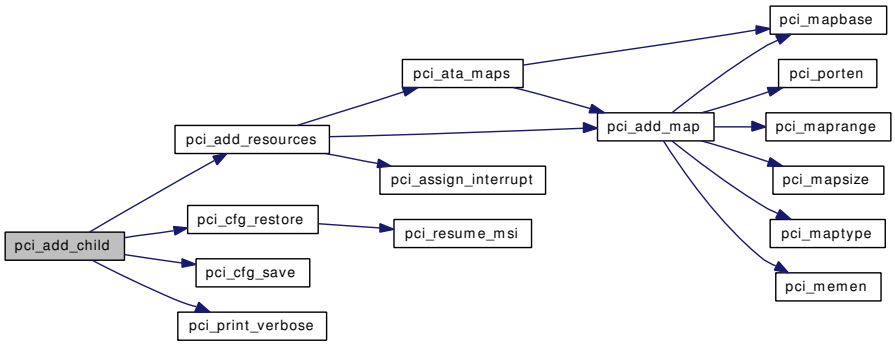
7.10.1.2 void pci_add_child (device_t *bus*, struct pci_devinfo * *dinfo*)

Definition at line 2301 of file pci.c.

References pci_add_resources(), pci_cfg_restore(), pci_cfg_save(), and pci_print_verbose().

Referenced by pci_add_children().

Here is the call graph for this function:



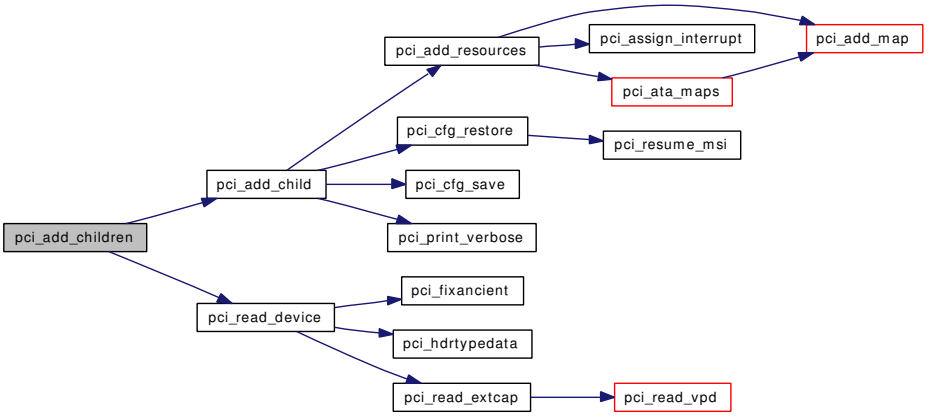
7.10.1.3 void pci_add_children (device_t *dev*, int *busno*, size_t *dinfo_size*)

Definition at line 2269 of file pci.c.

References pci_add_child(), PCI_FUNCMAX, PCI_MAXHDRTYPE, pci_read_device(), PCIM_-HDRTYPE, PCIM_MFDEV, PCIR_HDRTYPE, and REG.

Referenced by pci_attach().

Here is the call graph for this function:



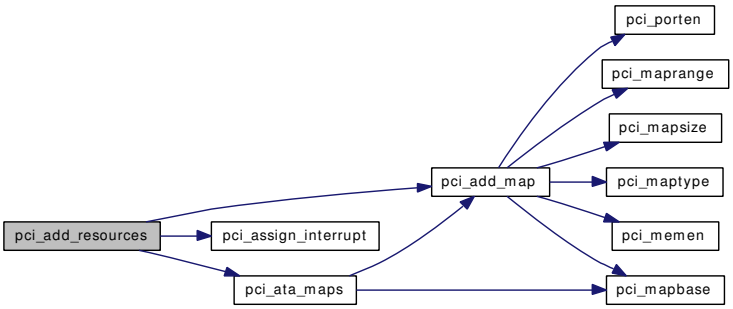
7.10.1.4 void pci_add_resources (device_t *bus*, device_t *dev*, int *force*, uint32_t *prefetchmask*)

Definition at line 2216 of file pci.c.

References pci_quirk::arg1, pcicfg::bus, pcicfg::device, pci_quirk::devid, pcicfg::func, pcicfg::intline, pcicfg::intpin, pcicfg::nummaps, pci_add_map(), pci_assign_interrupt(), pci_ata_maps(), PCI_QUIRK_MAP_REG, pci_quirks, PCIC_STORAGE, PCIP_STORAGE_IDE_MASTERDEV, PCIR_BAR, PCIS_STORAGE_IDE, pcicfg::slot, pci_quirk::type, and pcicfg::vendor.

Referenced by pci_add_child().

Here is the call graph for this function:



7.10.1.5 int pci_alloc_msi_method (device_t *dev*, device_t *child*, int * *count*)

Definition at line 1445 of file pci.c.

References pcicfg::msi, pcicfg_msi::msi_alloc, pcicfg_msi::msi_ctrl, pcicfg_msi::msi_location, pcicfg_msi::msi_msgnum, pcicfg::msix, pcicfg_msix::msix_alloc, pci_do_msi, pci_msi_blacklisted(), PCIM_MSICTRL_MME_MASK, and PCIR_MSI_CTRL.

Here is the call graph for this function:

7.10.1.6 int pci_alloc_msix_method (device_t *dev*, device_t *child*, int * *count*)

Definition at line 1067 of file pci.c.

References pcicfg::msi, pcicfg_msi::msi_alloc, pcicfg::msix, pcicfg_msix::msix_alloc, pcicfg_msix::msix_location, pcicfg_msix::msix_msgnum, pcicfg_msix::msix_pba_bar, pcicfg_msix::msix_pba_res, pcicfg_msix::msix_table_bar, pcicfg_msix::msix_table_res, pci_do_msix, and pci_msi_blacklisted().

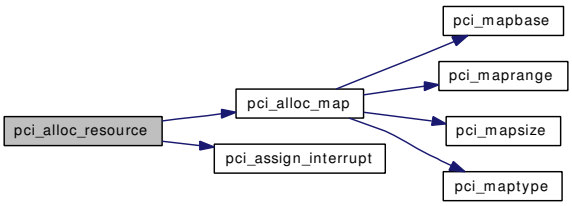
Here is the call graph for this function:

7.10.1.7 struct resource* pci_alloc_resource (device_t *dev*, device_t *child*, int *type*, int * *rid*, u_long *start*, u_long *end*, u_long *count*, u_int *flags*)

Definition at line 3005 of file pci.c.

References pcicfg::intline, pcicfg::intpin, pcicfg::msi, pcicfg_msi::msi_alloc, pcicfg::msix, pcicfg_msix::msix_alloc, pcicfg::nummaps, pci_alloc_map(), pci_assign_interrupt(), and PCIR_BAR.

Here is the call graph for this function:



7.10.1.8 int pci_assign_interrupt_method (device_t *dev*, device_t *child*)

Definition at line 3173 of file pci.c.

References pcicfg::intpin.

7.10.1.9 void pci_cfg_restore (device_t, struct pci_devinfo *)

Definition at line 3205 of file pci.c.

References pci_resume_msi(), PCIR_BAR, PCIR_BIOS, PCIR_CACHELNSZ, PCIR_COMMAND, PCIR_INTLINE, PCIR_INTPIN, PCIR_LATTIMER, PCIR_MAXLAT, PCIR_MINGNT, PCIR_PROGIF, and PCIR_REVID.

Referenced by pci_add_child(), pci_driver_added(), and pci_resume().

Here is the call graph for this function:

7.10.1.10 void pci_cfg_save (device_t, struct pci_devinfo *, int)

Definition at line 3252 of file pci.c.

References pci_do_power_nodriver, PCIC_BASEPERIPH, PCIC_DISPLAY, PCIC_MEMORY, PCIC_STORAGE, PCIR_BAR, PCIR_BIOS, PCIR_CACHELNSZ, PCIR_CLASS, PCIR_COMMAND, PCIR_DEVICE, PCIR_INTLINE, PCIR_INTPIN, PCIR_LATTIMER, PCIR_MAXLAT, PCIR_MINGNT, PCIR_PROGIF, PCIR_REVID, PCIR_SUBCLASS, PCIR_SUBDEV_0, PCIR_SUBVEND_0, and PCIR_VENDOR.

Referenced by pci_add_child(), pci_driver_added(), pci_probe_nomatch(), and pci_suspend().

7.10.1.11 int pci_child_location_str_method (device_t *cbdev*, device_t *child*, char * *buf*, size_t *buflen*)

Definition at line 3147 of file pci.c.

7.10.1.12 int pci_child_pnpinfo_str_method (device_t *cbdev*, device_t *child*, char * *buf*, size_t *buflen*)

Definition at line 3157 of file pci.c.

References pcicfg::baseclass, pcicfg::device, pcicfg::progif, pcicfg::subclass, pcicfg::subdevice, pcicfg::subvendor, and pcicfg::vendor.

7.10.1.13 void pci_delete_resource (device_t *dev*, device_t *child*, int *type*, int *rid*)

Definition at line 3082 of file pci.c.

7.10.1.14 int pci_disable_busmaster_method (device_t *dev*, device_t *child*)

Definition at line 1797 of file pci.c.

References pci_clear_command_bit(), and PCIM_CMD_BUSMASTEREN.

Here is the call graph for this function:

7.10.1.15 int pci_disable_io_method (device_t *dev*, device_t *child*, int *space*)

Definition at line 1835 of file pci.c.

References pci_clear_command_bit(), PCIM_CMD_MEMEN, PCIM_CMD_PORTEN, and PCIR_COMMAND.

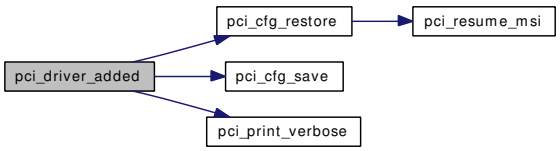
Here is the call graph for this function:

7.10.1.16 void pci_driver_added (device_t *dev*, driver_t * *driver*)

Definition at line 2441 of file pci.c.

References pci_cfg_restore(), pci_cfg_save(), and pci_print_verbose().

Here is the call graph for this function:



7.10.1.17 int pci_enable_busmaster_method (device_t *dev*, device_t *child*)

Definition at line 1790 of file pci.c.

References pci_set_command_bit(), and PCIM_CMD_BUSMASTEREN.

Here is the call graph for this function:

7.10.1.18 int pci_enable_io_method (device_t *dev*, device_t *child*, int *space*)

Definition at line 1804 of file pci.c.

References pci_set_command_bit(), PCIM_CMD_MEMEN, PCIM_CMD_PORTEN, and PCIR_-COMMAND.

Here is the call graph for this function:

7.10.1.19 int pci_find_extcap_method (device_t *dev*, device_t *child*, int *capability*, int * *capreg*)

Definition at line 952 of file pci.c.

References pcicfg::hdrtype, PCICAP_ID, PCICAP_NEXTPTR, PCIM_HDRTYPE, PCIM_STATUS_-CAPPRESENT, PCIR_CAP_PTR, PCIR_CAP_PTR_2, and PCIR_STATUS.

7.10.1.20 int pci_freecfg (struct pci_devinfo * *dinfo*)

Definition at line 1626 of file pci.c.

References pci_devq, pci_generation, and pci_numdevs.

7.10.1.21 int pci_get_powerstate_method (device_t *dev*, device_t *child*)

Definition at line 1732 of file pci.c.

References PCIM_PSTAT_D0, PCIM_PSTAT_D1, PCIM_PSTAT_D2, PCIM_PSTAT_D3, PCIM_-PSTAT_DMASK, pcicfg::pp, pcicfg_pp::pp_cap, and pcicfg_pp::pp_status.

7.10.1.22 struct resource_list* pci_get_resource_list (device_t *dev*, device_t *child*)

Definition at line 3118 of file pci.c.

7.10.1.23 int pci_get_vpd_ident_method (device_t *dev*, device_t *child*, const char ** *identptr*)

Definition at line 913 of file pci.c.

References pcicfg::vpd, and pcicfg_vpd::vpd_ident.

7.10.1.24 int pci_get_vpd_readonly_method (device_t *dev*, device_t *child*, const char * *kw*, const char ** *vptr*)

Definition at line 927 of file pci.c.

References vpd_readonly::keyword, vpd_readonly::value, pcicfg::vpd, pcicfg_vpd::vpd_rocnt, and pcicfg_vpd::vpd_ros.

7.10.1.25 int pci_msi_count_method (device_t *dev*, device_t *child*)

Definition at line 1613 of file pci.c.

References pcicfg::msi, pcicfg_msi::msi_location, pcicfg_msi::msi_msgnum, and pci_do_msi.

7.10.1.26 int pci_msix_count_method (device_t *dev*, device_t *child*)

Definition at line 1324 of file pci.c.

References pcicfg::msix, pcicfg_msix::msix_location, pcicfg_msix::msix_msgnum, and pci_do_msix.

7.10.1.27 int pci_print_child (device_t *dev*, device_t *child*)

Definition at line 2470 of file pci.c.

7.10.1.28 void pci_print_verbose (struct pci_devinfo * *dinfo*)

Definition at line 1871 of file pci.c.

References pcicfg::baseclass, pcicfg::bus, pcicfg::cachelnsz, pcicfg::cmdreg, pcicfg::dev, pcicfg::device, pcicfg::func, pcicfg::hdrtype, pcicfg::intline, pcicfg::intpin, pcicfg::lattimer, pcicfg::maxlat, pcicfg::mfdev, pcicfg::mingnt, pcicfg::msi, pcicfg_msi::msi_ctrl, pcicfg_msi::msi_location, pcicfg_msi::msi_msgnum, pcicfg::msix, pcicfg_msix::msix_location, pcicfg_msix::msix_msgnum, pcicfg_msix::msix_pba_bar, pcicfg_msix::msix_table_bar, PCIM_MSICTRL_64BIT, PCIM_MSICTRL_VECTOR, PCIM_PCAP_D1SUPP, PCIM_PCAP_D2SUPP, PCIM_PCAP_SPEC, PCIM_PSTAT_DMASK, pcicfg::pp, pcicfg_pp::pp_cap, pcicfg_pp::pp_status, pcicfg::progif, pcicfg::revid, pcicfg::slot, pcicfg::statreg, pcicfg::subclass, pcicfg::vendor, pcicfg::vpd, pcicfg_vpd::vpd_ident, pcicfg_vpd::vpd_reg, pcicfg_vpd::vpd_rocnt, pcicfg_vpd::vpd_ros, pcicfg_vpd::vpd_w, and pcicfg_vpd::vpd_wcnt.

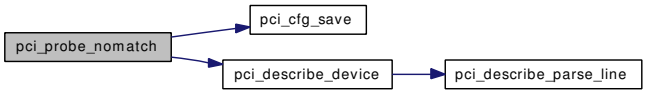
Referenced by pci_add_child(), and pci_driver_added().

7.10.1.29 void pci_probe_nomatch (device_t *dev*, device_t *child*)

Definition at line 2583 of file pci.c.

References desc, pci_cfg_save(), pci_describe_device(), pci_do_power_nodriver, pci_nomatch_tab, and subclass.

Here is the call graph for this function:



7.10.1.30 uint32_t pci_read_config_method (device_t *dev*, device_t *child*, int *reg*, int *width*)

Definition at line 3126 of file pci.c.

References pcicfg::bus, pcicfg::func, and pcicfg::slot.

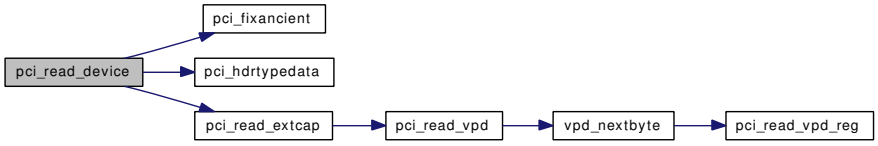
7.10.1.31 struct pci_devinfo* pci_read_device (device_t *pcib*, int *b*, int *s*, int *f*, size_t *size*)

Definition at line 413 of file pci.c.

References pcicfg::baseclass, pcicfg::bus, pcicfg::cachelnsz, pcicfg::cmdreg, pcicfg::device, pcicfg::func, pcicfg::hdrtype, pcicfg::intline, pcicfg::intpin, pcicfg::lattimer, pcicfg::maxlat, pcicfg::mfdev, pcicfg::mingnt, pci_devq, pci_fixancient(), pci_generation, pci_hdrtypedata(), pci_numdevs, pci_read_extcap(), PCIM_MFDEV, PCIM_STATUS_CAPPRESENT, PCIR_CACHELNSZ, PCIR_CLASS, PCIR_COMMAND, PCIR_DEVICE, PCIR_DEVVENDOR, PCIR_HDRTYPE, PCIR_INTLINE, PCIR_INTPIN, PCIR_LATTIMER, PCIR_MAXLAT, PCIR_MINGNT, PCIR_PROGIF, PCIR_REVID, PCIR_STATUS, PCIR_SUBCLASS, PCIR_VENDOR, pcicfg::progif, REG, pcicfg::revid, pcicfg::slot, pcicfg::statreg, pcicfg::subclass, pcicfg::subdevice, pcicfg::subvendor, and pcicfg::vendor.

Referenced by pci_add_children().

Here is the call graph for this function:



7.10.1.32 int pci_read_ivar (device_t *dev*, device_t *child*, int *which*, uintptr_t * *result*)

Definition at line 2752 of file pci.c.

References pcicfg::baseclass, pcicfg::bus, pcicfg::cachelnsz, pcicfg::cmdreg, pcicfg::device, pcicfg::func, pcicfg::intline, pcicfg::intpin, pcicfg::lattimer, pcicfg::maxlat, pcicfg::mingnt, pcicfg::progif, pcicfg::revid, pcicfg::slot, pcicfg::subclass, pcicfg::subdevice, pcicfg::subvendor, and pcicfg::vendor.

7.10.1.33 int pci_release_msi_method (device_t *dev*, device_t *child*)

Definition at line 1564 of file pci.c.

References pcicfg::msi, pcicfg_msi::msi_alloc, pcicfg_msi::msi_ctrl, pcicfg_msi::msi_location, pci_release_msix(), PCIM_MSICTRL_MME_MASK, PCIM_MSICTRL_MSI_ENABLE, and PCIR_MSI_CTRL.

Here is the call graph for this function:

7.10.1.34 int pci_remap_msix_method (device_t *dev*, device_t *child*, u_int * *indices*)

Definition at line 1203 of file pci.c.

References pcicfg::msix, pcicfg_msix::msix_alloc, and pcicfg_msix::msix_msgnum.

7.10.1.35 int pci_resume (device_t *dev*)

Definition at line 2391 of file pci.c.

References ACPI_PWR_FOR_SLEEP, pci_cfg_restore(), and pci_do_power_resume.

Here is the call graph for this function:

7.10.1.36 int pci_set_powerstate_method (device_t *dev*, device_t *child*, int *state*)

Definition at line 1657 of file pci.c.

References PCIM_PCAP_D1SUPP, PCIM_PCAP_D2SUPP, PCIM_PSTAT_D0, PCIM_PSTAT_D1, PCIM_PSTAT_D2, PCIM_PSTAT_D3, PCIM_PSTAT_DMASK, pcicfg::pp, pcicfg_pp::pp_cap, and pcicfg_pp::pp_status.

7.10.1.37 int pci_suspend (device_t *dev*)

Definition at line 2343 of file pci.c.

References ACPI_PWR_FOR_SLEEP, pci_cfg_save(), and pci_do_power_resume.

Here is the call graph for this function:

7.10.1.38 void pci_write_config_method (device_t *dev*, device_t *child*, int *reg*, uint32_t *val*, int *width*)

Definition at line 3136 of file pci.c.

References pcicfg::bus, pcicfg::func, and pcicfg::slot.

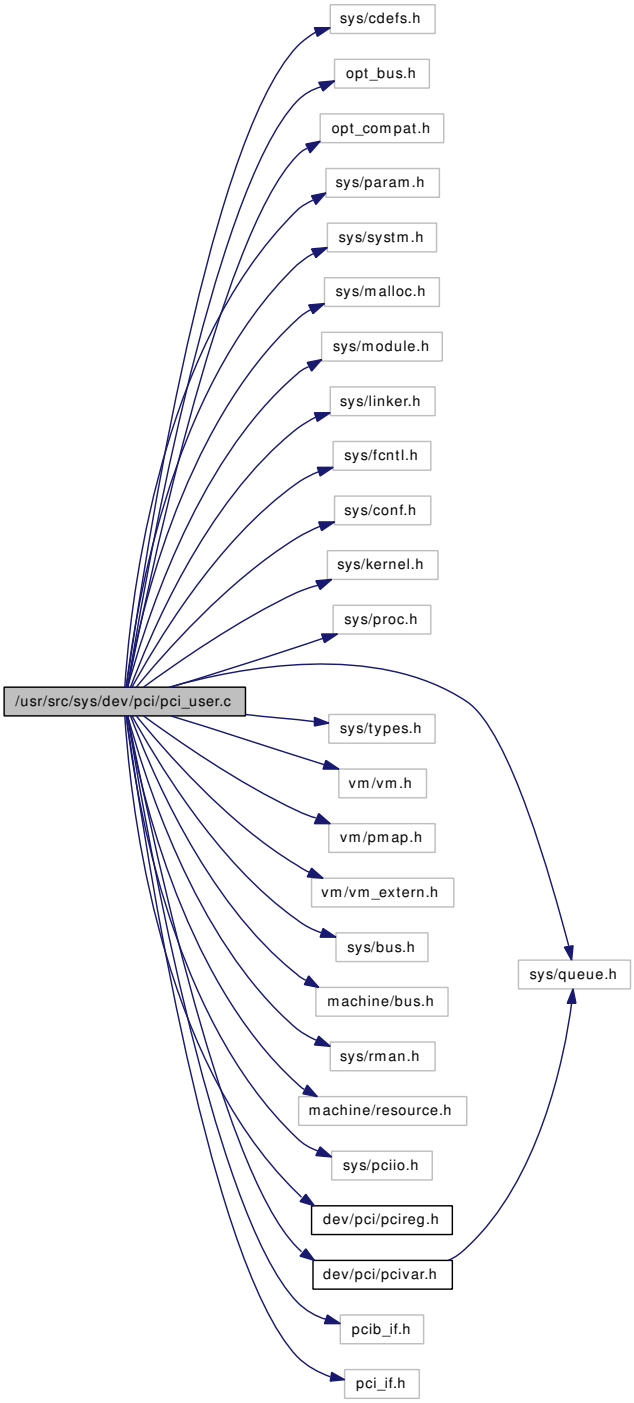
7.10.1.39 int pci_write_ivar (device_t *dev*, device_t *child*, int *which*, uintptr_t *value*)

Definition at line 2832 of file pci.c.

7.11 /usr/src/sys/dev/pci/pci_user.c File Reference

```
#include <sys/cdefs.h>
#include "opt_bus.h"
#include "opt_compat.h"
#include <sys/param.h>
#include <sys/systm.h>
#include <sys/malloc.h>
#include <sys/module.h>
#include <sys/linker.h>
#include <sys/fcntl.h>
#include <sys/conf.h>
#include <sys/kernel.h>
#include <sys/proc.h>
#include <sys/queue.h>
#include <sys/types.h>
#include <vm/vm.h>
#include <vm/pmap.h>
#include <vm/vm_extern.h>
#include <sys/bus.h>
#include <machine/bus.h>
#include <sys/rman.h>
#include <machine/resource.h>
#include <sys/pciio.h>
#include <dev/pci/pcireg.h>
#include <dev/pci/pcivar.h>
#include "pcib_if.h"
#include "pci_if.h"
```

Include dependency graph for pci_user.c:



Functions

- __FBSDID ("$FreeBSD: src/sys/dev/pci/pci_user.c,v 1.21 2006/10/06 14:31:32 ru Exp $")
- static int pci_conf_match (struct pci_match_conf *matches, int num_matches, struct pci_conf *match_buf)
- static int pci_open (struct cdev *dev, int oflags, int devtype, struct thread *td)
- static int pci_close (struct cdev *dev, int flag, int devtype, struct thread *td)

- static int pci_ioctl (struct cdev *dev, u_long cmd, caddr_t data, int flag, struct thread *td)

Variables

- static d_open_t pci_open
- static d_close_t pci_close
- static d_ioctl_t pci_ioctl
- cdevsw pcicdev

7.11.1 Function Documentation

7.11.1.1 __FBSDID ("$FreeBSD: src/sys/dev/pci/pci_user. *c*, v 1.21 2006/10/06 14:31:32 ru Exp $")

7.11.1.2 static int pci_close (struct cdev * *dev*, int *flag*, int *devtype*, struct thread * *td*) [static]

Definition at line 95 of file pci_user.c.

7.11.1.3 static int pci_conf_match (struct pci_match_conf * *matches*, int *num_matches*, struct pci_conf * *match_buf*) [static]

Definition at line 108 of file pci_user.c.

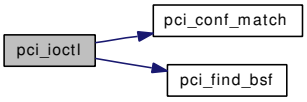
Referenced by pci_ioctl().

7.11.1.4 static int pci_ioctl (struct cdev * *dev*, u_long *cmd*, caddr_t *data*, int *flag*, struct thread * *td*) [static]

Definition at line 168 of file pci_user.c.

References pci_conf_match(), pci_devq, pci_find_bsf(), pci_generation, pci_numdevs, and PCI_-REGMAX.

Here is the call graph for this function:



7.11.1.5 static int pci_open (struct cdev * *dev*, int *oflags*, int *devtype*, struct thread * *td*) [static]

Definition at line 81 of file pci_user.c.

7.11.2 Variable Documentation

7.11.2.1 d_close_t pci_close [static]

Definition at line 66 of file pci_user.c.

7.11.2.2 d_ioctl_t pci_ioctl [static]

Definition at line 69 of file pci_user.c.

7.11.2.3 d_open_t pci_open [static]

Definition at line 65 of file pci_user.c.

7.11.2.4 struct cdevsw pcicdev

Initial value:

```
{
        .d_version =    D_VERSION,
        .d_flags =      D_NEEDGIANT,
        .d_open =       pci_open,
        .d_close =      pci_close,
        .d_ioctl =      pci_ioctl,
        .d_name =       "pci",
}
```

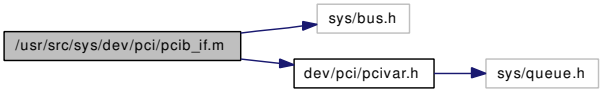
Definition at line 71 of file pci_user.c.

Referenced by pci_modevent().

7.12 /usr/src/sys/dev/pci/pcib_if.m File Reference

```
#include <sys/bus.h>
#include <dev/pci/pcivar.h>
```

Include dependency graph for pcib_if.m:



Variables

- INTERFACE pcib

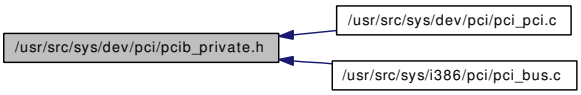
7.12.1 Variable Documentation

7.12.1.1 INTERFACE pcib

Definition at line 32 of file pcib_if.m.

7.13 /usr/src/sys/dev/pci/pcib_private.h File Reference

This graph shows which files directly or indirectly include this file:



Data Structures

- struct pcib_softc

Defines

- #define PCIB_SUBTRACTIVE 0x1
- #define PCIB_DISABLE_MSI 0x2

Typedefs

- typedef uint32_t pci_read_config_fn (int b, int s, int f, int reg, int width)

Functions

- int host_pcib_get_busno (pci_read_config_fn read_config, int bus, int slot, int func, uint8_t *busnum)
- int pcib_attach (device_t dev)
- void pcib_attach_common (device_t dev)
- int pcib_read_ivar (device_t dev, device_t child, int which, uintptr_t *result)
- int pcib_write_ivar (device_t dev, device_t child, int which, uintptr_t value)
- resource * pcib_alloc_resource (device_t dev, device_t child, int type, int *rid, u_long start, u_long end, u_long count, u_int flags)
- int pcib_maxslots (device_t dev)
- uint32_t pcib_read_config (device_t dev, int b, int s, int f, int reg, int width)
- void pcib_write_config (device_t dev, int b, int s, int f, int reg, uint32_t val, int width)
- int pcib_route_interrupt (device_t pcib, device_t dev, int pin)
- int pcib_alloc_msi (device_t pcib, device_t dev, int count, int maxcount, int *irqs)
- int pcib_release_msi (device_t pcib, device_t dev, int count, int *irqs)
- int pcib_alloc_msix (device_t pcib, device_t dev, int index, int *irq)
- int pcib_remap_msix (device_t pcib, device_t dev, int index, int irq)
- int pcib_release_msix (device_t pcib, device_t dev, int irq)

7.13.1 Define Documentation

7.13.1.1 #define PCIB_DISABLE_MSI 0x2

Definition at line 49 of file pcib_private.h.

Referenced by pcib_alloc_msi(), pcib_alloc_msix(), and pcib_attach_common().

7.13.1.2 #define PCIB_SUBTRACTIVE 0x1

Definition at line 48 of file pcib_private.h.

Referenced by pcib_alloc_resource(), and pcib_attach_common().

7.13.2 Typedef Documentation

7.13.2.1 typedef uint32_t pci_read_config_fn(int b, int s, int f, int reg, int width)

Definition at line 64 of file pcib_private.h.

7.13.3 Function Documentation

7.13.3.1 int host_pcib_get_busno (pci_read_config_fn *read_config*, int *bus*, int *slot*, int *func*, uint8_t * *busnum*)

Definition at line 612 of file pci_pci.c.

References PCIR_DEVVENDOR.

7.13.3.2 int pcib_alloc_msi (device_t *pcib*, device_t *dev*, int *count*, int *maxcount*, int * *irqs*)

Definition at line 552 of file pci_pci.c.

References pcib_softc::flags, and PCIB_DISABLE_MSI.

7.13.3.3 int pcib_alloc_msix (device_t *pcib*, device_t *dev*, int *index*, int * *irq*)

Definition at line 576 of file pci_pci.c.

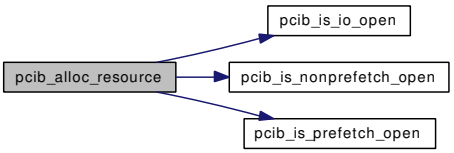
References pcib_softc::flags, and PCIB_DISABLE_MSI.

7.13.3.4 struct resource* pcib_alloc_resource (device_t *dev*, device_t *child*, int *type*, int * *rid*, u_long *start*, u_long *end*, u_long *count*, u_int *flags*)

Definition at line 337 of file pci_pci.c.

References pcib_softc::bridgectl, pcib_softc::flags, pcib_softc::iobase, pcib_softc::iolimit, pcib_softc::membase, pcib_softc::memlimit, PCIB_BCR_VGA_ENABLE, pcib_is_io_open(), pcib_is_nonprefetch_open(), pcib_is_prefetch_open(), PCIB_SUBTRACTIVE, pcib_softc::pmembase, and pcib_softc::pmemlimit.

Here is the call graph for this function:

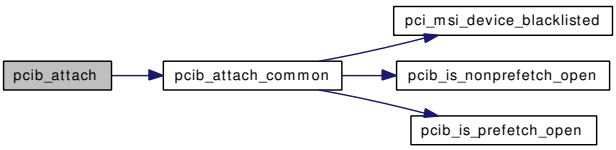


7.13.3.5 int pcib_attach (device_t *dev*)

Definition at line 289 of file pci_pci.c.

References pcib_attach_common(), and pcib_softc::secbus.

Here is the call graph for this function:



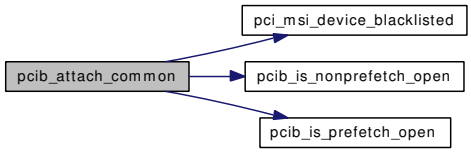
7.13.3.6 void pcib_attach_common (device_t *dev*)

Definition at line 138 of file pci_pci.c.

References pcib_softc::bridgectl, pcib_softc::command, pcib_softc::dev, pcib_softc::flags, pcib_softc::iobase, pcib_softc::iolimit, pcib_softc::membase, pcib_softc::memlimit, pci_msi_device_blacklisted(), PCI_PPBIOBASE, PCI_PPBIOLIMIT, PCI_PPBMEMBASE, PCI_PPBMEMLIMIT, PCIB_DISABLE_MSI, pcib_is_nonprefetch_open(), pcib_is_prefetch_open(), PCIB_SUBTRACTIVE, PCIM_BRIO_32, PCIM_BRIO_MASK, PCIM_CMD_MEMEN, PCIM_CMD_PORTEN, PCIR_BRIDGECTL_1, PCIR_COMMAND, PCIR_IOBASEH_1, PCIR_IOBASEL_1, PCIR_IOLIMITH_1, PCIR_IOLIMITL_1, PCIR_MEMBASE_1, PCIR_MEMLIMIT_1, PCIR_PMBASEH_1, PCIR_PMBASEL_1, PCIR_PMLIMITH_1, PCIR_PMLIMITL_1, PCIR_PROGIF, PCIR_SECBUS_1, PCIR_SECLAT_1, PCIR_SECSTAT_1, PCIR_SUBBUS_1, pcib_softc::pmembase, pcib_softc::pmemlimit, pcib_softc::secbus, pcib_softc::seclat, pcib_softc::secstat, and pcib_softc::subbus.

Referenced by pcib_attach().

Here is the call graph for this function:



7.13.3.7 int pcib_maxslots (device_t *dev*)

Definition at line 494 of file pci_pci.c.

References PCI_SLOTMAX.

7.13.3.8 uint32_t pcib_read_config (device_t *dev*, int *b*, int *s*, int *f*, int *reg*, int *width*)

Definition at line 503 of file pci_pci.c.

7.13.3.9 int pcib_read_ivar (device_t *dev*, device_t *child*, int *which*, uintptr_t * *result*)

Definition at line 307 of file pci_pci.c.

References pcib_softc::secbus.

7.13.3.10 int pcib_release_msi (device_t *pcib*, device_t *dev*, int *count*, int * *irqs*)

Definition at line 566 of file pci_pci.c.

7.13.3.11 int pcib_release_msix (device_t *pcib*, device_t *dev*, int *irq*)

Definition at line 599 of file pci_pci.c.

7.13.3.12 int pcib_remap_msix (device_t *pcib*, device_t *dev*, int *index*, int *irq*)

Definition at line 589 of file pci_pci.c.

7.13.3.13 int pcib_route_interrupt (device_t *pcib*, device_t *dev*, int *pin*)

Definition at line 518 of file pci_pci.c.

7.13.3.14 void pcib_write_config (device_t *dev*, int *b*, int *s*, int *f*, int *reg*, uint32_t *val*, int *width*)

Definition at line 509 of file pci_pci.c.

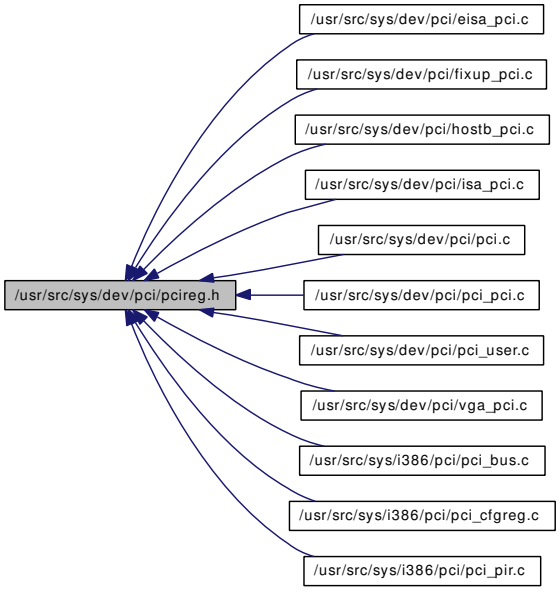
7.13.3.15 int pcib_write_ivar (device_t *dev*, device_t *child*, int *which*, uintptr_t *value*)

Definition at line 320 of file pci_pci.c.

References pcib_softc::secbus.

7.14 /usr/src/sys/dev/pci/pcireg.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- #define PCI_BUSMAX 255
- #define PCI_SLOTMAX 31
- #define PCI_FUNCMAX 7
- #define PCI_REGMAX 255
- #define PCI_MAXHDRTYPE 2
- #define PCIR_DEVVENDOR 0x00
- #define PCIR_VENDOR 0x00
- #define PCIR_DEVICE 0x02
- #define PCIR_COMMAND 0x04
- #define PCIM_CMD_PORTEN 0x0001
- #define PCIM_CMD_MEMEN 0x0002
- #define PCIM_CMD_BUSMASTEREN 0x0004
- #define PCIM_CMD_SPECIALEN 0x0008
- #define PCIM_CMD_MWRICEN 0x0010
- #define PCIM_CMD_PERRESPEN 0x0040
- #define PCIM_CMD_SERRESPEN 0x0100
- #define PCIM_CMD_BACKTOBACK 0x0200
- #define PCIR_STATUS 0x06
- #define PCIM_STATUS_CAPPRESENT 0x0010
- #define PCIM_STATUS_66CAPABLE 0x0020
- #define PCIM_STATUS_BACKTOBACK 0x0080
- #define PCIM_STATUS_PERRREPORT 0x0100
- #define PCIM_STATUS_SEL_FAST 0x0000
- #define PCIM_STATUS_SEL_MEDIMUM 0x0200

- #define PCIM_STATUS_SEL_SLOW 0x0400
- #define PCIM_STATUS_SEL_MASK 0x0600
- #define PCIM_STATUS_STABORT 0x0800
- #define PCIM_STATUS_RTABORT 0x1000
- #define PCIM_STATUS_RMABORT 0x2000
- #define PCIM_STATUS_SERR 0x4000
- #define PCIM_STATUS_PERR 0x8000
- #define PCIR_REVID 0x08
- #define PCIR_PROGIF 0x09
- #define PCIR_SUBCLASS 0x0a
- #define PCIR_CLASS 0x0b
- #define PCIR_CACHELNSZ 0x0c
- #define PCIR_LATTIMER 0x0d
- #define PCIR_HDRTYPE 0x0e
- #define PCIM_HDRTYPE 0x7f
- #define PCIM_HDRTYPE_NORMAL 0x00
- #define PCIM_HDRTYPE_BRIDGE 0x01
- #define PCIM_HDRTYPE_CARDBUS 0x02
- #define PCIM_MFDEV 0x80
- #define PCIR_BIST 0x0f
- #define PCICAP_ID 0x0
- #define PCICAP_NEXTPTR 0x1
- #define PCIY_PMG 0x01
- #define PCIY_AGP 0x02
- #define PCIY_VPD 0x03
- #define PCIY_SLOTID 0x04
- #define PCIY_MSI 0x05
- #define PCIY_CHSWP 0x06
- #define PCIY_PCIX 0x07
- #define PCIY_HT 0x08
- #define PCIY_VENDOR 0x09
- #define PCIY_DEBUG 0x0a
- #define PCIY_CRES 0x0b
- #define PCIY_HOTPLUG 0x0c
- #define PCIY_SUBVENDOR 0x0d
- #define PCIY_AGP8X 0x0e
- #define PCIY_SECDEV 0x0f
- #define PCIY_EXPRESS 0x10
- #define PCIY_MSIX 0x11
- #define PCIR_BARS 0x10
- #define PCIR_BAR(x) (PCIR_BARS + (x) ∗ 4)
- #define PCI_RID2BAR(rid) (((rid)-PCIR_BARS)/4)
- #define PCIR_CIS 0x28
- #define PCIM_CIS_ASI_MASK 0x7
- #define PCIM_CIS_ASI_TUPLE 0
- #define PCIM_CIS_ASI_BAR0 1
- #define PCIM_CIS_ASI_BAR1 2
- #define PCIM_CIS_ASI_BAR2 3
- #define PCIM_CIS_ASI_BAR3 4
- #define PCIM_CIS_ASI_BAR4 5

- #define PCIM_CIS_ASI_BAR5 6
- #define PCIM_CIS_ASI_ROM 7
- #define PCIM_CIS_ADDR_MASK 0x0ffffff8
- #define PCIM_CIS_ROM_MASK 0xf0000000
- #define PCIR_SUBVEND_0 0x2c
- #define PCIR_SUBDEV_0 0x2e
- #define PCIR_BIOS 0x30
- #define PCIM_BIOS_ENABLE 0x01
- #define PCIM_BIOS_ADDR_MASK 0xfffff800
- #define PCIR_CAP_PTR 0x34
- #define PCIR_INTLINE 0x3c
- #define PCIR_INTPIN 0x3d
- #define PCIR_MINGNT 0x3e
- #define PCIR_MAXLAT 0x3f
- #define PCIR_SECSTAT_1 0x1e
- #define PCIR_PRIBUS_1 0x18
- #define PCIR_SECBUS_1 0x19
- #define PCIR_SUBBUS_1 0x1a
- #define PCIR_SECLAT_1 0x1b
- #define PCIR_IOBASEL_1 0x1c
- #define PCIR_IOLIMITL_1 0x1d
- #define PCIR_IOBASEH_1 0x30
- #define PCIR_IOLIMITH_1 0x32
- #define PCIM_BRIO_16 0x0
- #define PCIM_BRIO_32 0x1
- #define PCIM_BRIO_MASK 0xf
- #define PCIR_MEMBASE_1 0x20
- #define PCIR_MEMLIMIT_1 0x22
- #define PCIR_PMBASEL_1 0x24
- #define PCIR_PMLIMITL_1 0x26
- #define PCIR_PMBASEH_1 0x28
- #define PCIR_PMLIMITH_1 0x2c
- #define PCIR_BRIDGECTL_1 0x3e
- #define PCIR_CAP_PTR_2 0x14
- #define PCIR_SECSTAT_2 0x16
- #define PCIR_PRIBUS_2 0x18
- #define PCIR_SECBUS_2 0x19
- #define PCIR_SUBBUS_2 0x1a
- #define PCIR_SECLAT_2 0x1b
- #define PCIR_MEMBASE0_2 0x1c
- #define PCIR_MEMLIMIT0_2 0x20
- #define PCIR_MEMBASE1_2 0x24
- #define PCIR_MEMLIMIT1_2 0x28
- #define PCIR_IOBASE0_2 0x2c
- #define PCIR_IOLIMIT0_2 0x30
- #define PCIR_IOBASE1_2 0x34
- #define PCIR_IOLIMIT1_2 0x38
- #define PCIR_BRIDGECTL_2 0x3e
- #define PCIR_SUBVEND_2 0x40
- #define PCIR_SUBDEV_2 0x42

- #define PCIR_PCCARDIF_2 0x44
- #define PCIC_OLD 0x00
- #define PCIS_OLD_NONVGA 0x00
- #define PCIS_OLD_VGA 0x01
- #define PCIC_STORAGE 0x01
- #define PCIS_STORAGE_SCSI 0x00
- #define PCIS_STORAGE_IDE 0x01
- #define PCIP_STORAGE_IDE_MODEPRIM 0x01
- #define PCIP_STORAGE_IDE_PROGINDPRIM 0x02
- #define PCIP_STORAGE_IDE_MODESEC 0x04
- #define PCIP_STORAGE_IDE_PROGINDSEC 0x08
- #define PCIP_STORAGE_IDE_MASTERDEV 0x80
- #define PCIS_STORAGE_FLOPPY 0x02
- #define PCIS_STORAGE_IPI 0x03
- #define PCIS_STORAGE_RAID 0x04
- #define PCIS_STORAGE_OTHER 0x80
- #define PCIC_NETWORK 0x02
- #define PCIS_NETWORK_ETHERNET 0x00
- #define PCIS_NETWORK_TOKENRING 0x01
- #define PCIS_NETWORK_FDDI 0x02
- #define PCIS_NETWORK_ATM 0x03
- #define PCIS_NETWORK_ISDN 0x04
- #define PCIS_NETWORK_OTHER 0x80
- #define PCIC_DISPLAY 0x03
- #define PCIS_DISPLAY_VGA 0x00
- #define PCIS_DISPLAY_XGA 0x01
- #define PCIS_DISPLAY_3D 0x02
- #define PCIS_DISPLAY_OTHER 0x80
- #define PCIC_MULTIMEDIA 0x04
- #define PCIS_MULTIMEDIA_VIDEO 0x00
- #define PCIS_MULTIMEDIA_AUDIO 0x01
- #define PCIS_MULTIMEDIA_TELE 0x02
- #define PCIS_MULTIMEDIA_OTHER 0x80
- #define PCIC_MEMORY 0x05
- #define PCIS_MEMORY_RAM 0x00
- #define PCIS_MEMORY_FLASH 0x01
- #define PCIS_MEMORY_OTHER 0x80
- #define PCIC_BRIDGE 0x06
- #define PCIS_BRIDGE_HOST 0x00
- #define PCIS_BRIDGE_ISA 0x01
- #define PCIS_BRIDGE_EISA 0x02
- #define PCIS_BRIDGE_MCA 0x03
- #define PCIS_BRIDGE_PCI 0x04
- #define PCIS_BRIDGE_PCMCIA 0x05
- #define PCIS_BRIDGE_NUBUS 0x06
- #define PCIS_BRIDGE_CARDBUS 0x07
- #define PCIS_BRIDGE_RACEWAY 0x08
- #define PCIS_BRIDGE_OTHER 0x80
- #define PCIC_SIMPLECOMM 0x07
- #define PCIS_SIMPLECOMM_UART 0x00

- #define PCIP_SIMPLECOMM_UART_16550A 0x02
- #define PCIS_SIMPLECOMM_PAR 0x01
- #define PCIS_SIMPLECOMM_MULSER 0x02
- #define PCIS_SIMPLECOMM_MODEM 0x03
- #define PCIS_SIMPLECOMM_OTHER 0x80
- #define PCIC_BASEPERIPH 0x08
- #define PCIS_BASEPERIPH_PIC 0x00
- #define PCIS_BASEPERIPH_DMA 0x01
- #define PCIS_BASEPERIPH_TIMER 0x02
- #define PCIS_BASEPERIPH_RTC 0x03
- #define PCIS_BASEPERIPH_PCIHOT 0x04
- #define PCIS_BASEPERIPH_SDHC 0x05
- #define PCIS_BASEPERIPH_OTHER 0x80
- #define PCIC_INPUTDEV 0x09
- #define PCIS_INPUTDEV_KEYBOARD 0x00
- #define PCIS_INPUTDEV_DIGITIZER 0x01
- #define PCIS_INPUTDEV_MOUSE 0x02
- #define PCIS_INPUTDEV_SCANNER 0x03
- #define PCIS_INPUTDEV_GAMEPORT 0x04
- #define PCIS_INPUTDEV_OTHER 0x80
- #define PCIC_DOCKING 0x0a
- #define PCIS_DOCKING_GENERIC 0x00
- #define PCIS_DOCKING_OTHER 0x80
- #define PCIC_PROCESSOR 0x0b
- #define PCIS_PROCESSOR_386 0x00
- #define PCIS_PROCESSOR_486 0x01
- #define PCIS_PROCESSOR_PENTIUM 0x02
- #define PCIS_PROCESSOR_ALPHA 0x10
- #define PCIS_PROCESSOR_POWERPC 0x20
- #define PCIS_PROCESSOR_MIPS 0x30
- #define PCIS_PROCESSOR_COPROC 0x40
- #define PCIC_SERIALBUS 0x0c
- #define PCIS_SERIALBUS_FW 0x00
- #define PCIS_SERIALBUS_ACCESS 0x01
- #define PCIS_SERIALBUS_SSA 0x02
- #define PCIS_SERIALBUS_USB 0x03
- #define PCIP_SERIALBUS_USB_UHCI 0x00
- #define PCIP_SERIALBUS_USB_OHCI 0x10
- #define PCIP_SERIALBUS_USB_EHCI 0x20
- #define PCIS_SERIALBUS_FC 0x04
- #define PCIS_SERIALBUS_SMBUS 0x05
- #define PCIC_WIRELESS 0x0d
- #define PCIS_WIRELESS_IRDA 0x00
- #define PCIS_WIRELESS_IR 0x01
- #define PCIS_WIRELESS_RF 0x10
- #define PCIS_WIRELESS_OTHER 0x80
- #define PCIC_INTELLIIO 0x0e
- #define PCIS_INTELLIIO_I2O 0x00
- #define PCIC_SATCOM 0x0f
- #define PCIS_SATCOM_TV 0x01

- #define PCIS_SATCOM_AUDIO 0x02
- #define PCIS_SATCOM_VOICE 0x03
- #define PCIS_SATCOM_DATA 0x04
- #define PCIC_CRYPTO 0x10
- #define PCIS_CRYPTO_NETCOMP 0x00
- #define PCIS_CRYPTO_ENTERTAIN 0x10
- #define PCIS_CRYPTO_OTHER 0x80
- #define PCIC_DASP 0x11
- #define PCIS_DASP_DPIO 0x00
- #define PCIS_DASP_OTHER 0x80
- #define PCIC_OTHER 0xff
- #define PCIB_BCR_PERR_ENABLE 0x0001
- #define PCIB_BCR_SERR_ENABLE 0x0002
- #define PCIB_BCR_ISA_ENABLE 0x0004
- #define PCIB_BCR_VGA_ENABLE 0x0008
- #define PCIB_BCR_MASTER_ABORT_MODE 0x0020
- #define PCIB_BCR_SECBUS_RESET 0x0040
- #define PCIB_BCR_SECBUS_BACKTOBACK 0x0080
- #define PCIB_BCR_PRI_DISCARD_TIMEOUT 0x0100
- #define PCIB_BCR_SEC_DISCARD_TIMEOUT 0x0200
- #define PCIB_BCR_DISCARD_TIMER_STATUS 0x0400
- #define PCIB_BCR_DISCARD_TIMER_SERREN 0x0800
- #define PCIR_POWER_CAP 0x2
- #define PCIM_PCAP_SPEC 0x0007
- #define PCIM_PCAP_PMEREQCLK 0x0008
- #define PCIM_PCAP_PMEREQPWR 0x0010
- #define PCIM_PCAP_DEVSPECINIT 0x0020
- #define PCIM_PCAP_DYNCLOCK 0x0040
- #define PCIM_PCAP_SECCLOCK 0x00c0
- #define PCIM_PCAP_CLOCKMASK 0x00c0
- #define PCIM_PCAP_REQFULLCLOCK 0x0100
- #define PCIM_PCAP_D1SUPP 0x0200
- #define PCIM_PCAP_D2SUPP 0x0400
- #define PCIM_PCAP_D0PME 0x1000
- #define PCIM_PCAP_D1PME 0x2000
- #define PCIM_PCAP_D2PME 0x4000
- #define PCIR_POWER_STATUS 0x4
- #define PCIM_PSTAT_D0 0x0000
- #define PCIM_PSTAT_D1 0x0001
- #define PCIM_PSTAT_D2 0x0002
- #define PCIM_PSTAT_D3 0x0003
- #define PCIM_PSTAT_DMASK 0x0003
- #define PCIM_PSTAT_REPENABLE 0x0010
- #define PCIM_PSTAT_PMEENABLE 0x0100
- #define PCIM_PSTAT_D0POWER 0x0000
- #define PCIM_PSTAT_D1POWER 0x0200
- #define PCIM_PSTAT_D2POWER 0x0400
- #define PCIM_PSTAT_D3POWER 0x0600
- #define PCIM_PSTAT_D0HEAT 0x0800
- #define PCIM_PSTAT_D1HEAT 0x1000

- #define PCIM_PSTAT_D2HEAT 0x1200
- #define PCIM_PSTAT_D3HEAT 0x1400
- #define PCIM_PSTAT_DATAUNKN 0x0000
- #define PCIM_PSTAT_DATADIV10 0x2000
- #define PCIM_PSTAT_DATADIV100 0x4000
- #define PCIM_PSTAT_DATADIV1000 0x6000
- #define PCIM_PSTAT_DATADIVMASK 0x6000
- #define PCIM_PSTAT_PME 0x8000
- #define PCIR_POWER_PMCSR 0x6
- #define PCIM_PMCSR_DCLOCK 0x10
- #define PCIM_PMCSR_B2SUPP 0x20
- #define PCIM_BMCSR_B3SUPP 0x40
- #define PCIM_BMCSR_BPCE 0x80
- #define PCIR_POWER_DATA 0x7
- #define PCIR_MSI_CTRL 0x2
- #define PCIM_MSICTRL_VECTOR 0x0100
- #define PCIM_MSICTRL_64BIT 0x0080
- #define PCIM_MSICTRL_MME_MASK 0x0070
- #define PCIM_MSICTRL_MME_1 0x0000
- #define PCIM_MSICTRL_MME_2 0x0010
- #define PCIM_MSICTRL_MME_4 0x0020
- #define PCIM_MSICTRL_MME_8 0x0030
- #define PCIM_MSICTRL_MME_16 0x0040
- #define PCIM_MSICTRL_MME_32 0x0050
- #define PCIM_MSICTRL_MMC_MASK 0x000E
- #define PCIM_MSICTRL_MMC_1 0x0000
- #define PCIM_MSICTRL_MMC_2 0x0002
- #define PCIM_MSICTRL_MMC_4 0x0004
- #define PCIM_MSICTRL_MMC_8 0x0006
- #define PCIM_MSICTRL_MMC_16 0x0008
- #define PCIM_MSICTRL_MMC_32 0x000A
- #define PCIM_MSICTRL_MSI_ENABLE 0x0001
- #define PCIR_MSI_ADDR 0x4
- #define PCIR_MSI_ADDR_HIGH 0x8
- #define PCIR_MSI_DATA 0x8
- #define PCIR_MSI_DATA_64BIT 0xc
- #define PCIR_MSI_MASK 0x10
- #define PCIR_MSI_PENDING 0x14
- #define PCIXR_COMMAND 0x2
- #define PCIXM_COMMAND_DPERR_E 0x0001
- #define PCIXM_COMMAND_ERO 0x0002
- #define PCIXM_COMMAND_MAX_READ 0x000c
- #define PCIXM_COMMAND_MAX_READ_512 0x0000
- #define PCIXM_COMMAND_MAX_READ_1024 0x0004
- #define PCIXM_COMMAND_MAX_READ_2048 0x0008
- #define PCIXM_COMMAND_MAX_READ_4096 0x000c
- #define PCIXM_COMMAND_MAX_SPLITS 0x0070
- #define PCIXM_COMMAND_MAX_SPLITS_1 0x0000
- #define PCIXM_COMMAND_MAX_SPLITS_2 0x0010
- #define PCIXM_COMMAND_MAX_SPLITS_3 0x0020

- #define PCIXM_COMMAND_MAX_SPLITS_4 0x0030
- #define PCIXM_COMMAND_MAX_SPLITS_8 0x0040
- #define PCIXM_COMMAND_MAX_SPLITS_12 0x0050
- #define PCIXM_COMMAND_MAX_SPLITS_16 0x0060
- #define PCIXM_COMMAND_MAX_SPLITS_32 0x0070
- #define PCIXM_COMMAND_VERSION 0x3000
- #define PCIXR_STATUS 0x4
- #define PCIXM_STATUS_DEVFN 0x000000FF
- #define PCIXM_STATUS_BUS 0x0000FF00
- #define PCIXM_STATUS_64BIT 0x00010000
- #define PCIXM_STATUS_133CAP 0x00020000
- #define PCIXM_STATUS_SC_DISCARDED 0x00040000
- #define PCIXM_STATUS_UNEXP_SC 0x00080000
- #define PCIXM_STATUS_COMPLEX_DEV 0x00100000
- #define PCIXM_STATUS_MAX_READ 0x00600000
- #define PCIXM_STATUS_MAX_READ_512 0x00000000
- #define PCIXM_STATUS_MAX_READ_1024 0x00200000
- #define PCIXM_STATUS_MAX_READ_2048 0x00400000
- #define PCIXM_STATUS_MAX_READ_4096 0x00600000
- #define PCIXM_STATUS_MAX_SPLITS 0x03800000
- #define PCIXM_STATUS_MAX_SPLITS_1 0x00000000
- #define PCIXM_STATUS_MAX_SPLITS_2 0x00800000
- #define PCIXM_STATUS_MAX_SPLITS_3 0x01000000
- #define PCIXM_STATUS_MAX_SPLITS_4 0x01800000
- #define PCIXM_STATUS_MAX_SPLITS_8 0x02000000
- #define PCIXM_STATUS_MAX_SPLITS_12 0x02800000
- #define PCIXM_STATUS_MAX_SPLITS_16 0x03000000
- #define PCIXM_STATUS_MAX_SPLITS_32 0x03800000
- #define PCIXM_STATUS_MAX_CUM_READ 0x1C000000
- #define PCIXM_STATUS_RCVD_SC_ERR 0x20000000
- #define PCIXM_STATUS_266CAP 0x40000000
- #define PCIXM_STATUS_533CAP 0x80000000
- #define PCIXR_SEC_STATUS 0x2
- #define PCIXM_SEC_STATUS_64BIT 0x0001
- #define PCIXM_SEC_STATUS_133CAP 0x0002
- #define PCIXM_SEC_STATUS_SC_DISC 0x0004
- #define PCIXM_SEC_STATUS_UNEXP_SC 0x0008
- #define PCIXM_SEC_STATUS_SC_OVERRUN 0x0010
- #define PCIXM_SEC_STATUS_SR_DELAYED 0x0020
- #define PCIXM_SEC_STATUS_BUS_MODE 0x03c0
- #define PCIXM_SEC_STATUS_VERSION 0x3000
- #define PCIXM_SEC_STATUS_266CAP 0x4000
- #define PCIXM_SEC_STATUS_533CAP 0x8000
- #define PCIXR_BRIDGE_STATUS 0x4
- #define PCIXM_BRIDGE_STATUS_DEVFN 0x000000FF
- #define PCIXM_BRIDGE_STATUS_BUS 0x0000FF00
- #define PCIXM_BRIDGE_STATUS_64BIT 0x00010000
- #define PCIXM_BRIDGE_STATUS_133CAP 0x00020000
- #define PCIXM_BRIDGE_STATUS_SC_DISCARDED 0x00040000
- #define PCIXM_BRIDGE_STATUS_UNEXP_SC 0x00080000

- #define PCIXM_BRIDGE_STATUS_SC_OVERRUN 0x00100000
- #define PCIXM_BRIDGE_STATUS_SR_DELAYED 0x00200000
- #define PCIXM_BRIDGE_STATUS_DEVID_MSGCAP 0x20000000
- #define PCIXM_BRIDGE_STATUS_266CAP 0x40000000
- #define PCIXM_BRIDGE_STATUS_533CAP 0x80000000
- #define PCIR_HT_COMMAND 0x2
- #define PCIM_HTCMD_CAP_MASK 0xf800
- #define PCIM_HTCAP_SLAVE 0x0000
- #define PCIM_HTCAP_HOST 0x2000
- #define PCIM_HTCAP_SWITCH 0x4000
- #define PCIM_HTCAP_INTERRUPT 0x8000
- #define PCIM_HTCAP_REVISION_ID 0x8800
- #define PCIM_HTCAP_UNITID_CLUMPING 0x9000
- #define PCIM_HTCAP_EXT_CONFIG_SPACE 0x9800
- #define PCIM_HTCAP_ADDRESS_MAPPING 0xa000
- #define PCIM_HTCAP_MSI_MAPPING 0xa800
- #define PCIM_HTCAP_DIRECT_ROUTE 0xb000
- #define PCIM_HTCAP_VCSET 0xb800
- #define PCIM_HTCAP_RETRY_MODE 0xc000
- #define PCIM_HTCMD_MSI_ENABLE 0x0001
- #define PCIR_HTMSI_ADDRESS_LO 0x4
- #define PCIR_HTMSI_ADDRESS_HI 0x8
- #define PCIR_VENDOR_LENGTH 0x2
- #define PCIR_VENDOR_DATA 0x3
- #define PCIR_DEBUG_PORT 0x2
- #define PCIM_DEBUG_PORT_OFFSET 0x1FFF
- #define PCIM_DEBUG_PORT_BAR 0xe000
- #define PCIR_SUBVENDCAP_ID 0x4
- #define PCIR_EXPRESS_FLAGS 0x2
- #define PCIM_EXP_FLAGS_VERSION 0x000F
- #define PCIM_EXP_FLAGS_TYPE 0x00F0
- #define PCIM_EXP_TYPE_ENDPOINT 0x0000
- #define PCIM_EXP_TYPE_LEGACY_ENDPOINT 0x0010
- #define PCIM_EXP_TYPE_ROOT_PORT 0x0040
- #define PCIM_EXP_TYPE_UPSTREAM_PORT 0x0050
- #define PCIM_EXP_TYPE_DOWNSTREAM_PORT 0x0060
- #define PCIM_EXP_TYPE_PCI_BRIDGE 0x0070
- #define PCIM_EXP_FLAGS_SLOT 0x0100
- #define PCIM_EXP_FLAGS_IRQ 0x3e00
- #define PCIR_MSIX_CTRL 0x2
- #define PCIM_MSIXCTRL_MSIX_ENABLE 0x8000
- #define PCIM_MSIXCTRL_FUNCTION_MASK 0x4000
- #define PCIM_MSIXCTRL_TABLE_SIZE 0x07FF
- #define PCIR_MSIX_TABLE 0x4
- #define PCIR_MSIX_PBA 0x8
- #define PCIM_MSIX_BIR_MASK 0x7
- #define PCIM_MSIX_BIR_BAR_10 0
- #define PCIM_MSIX_BIR_BAR_14 1
- #define PCIM_MSIX_BIR_BAR_18 2
- #define PCIM_MSIX_BIR_BAR_1C 3
- #define PCIM_MSIX_BIR_BAR_20 4
- #define PCIM_MSIX_BIR_BAR_24 5
- #define PCIM_MSIX_VCTRL_MASK 0x1

7.14.1 Define Documentation

7.14.1.1 #define PCI_BUSMAX 255

Definition at line 43 of file pcireg.h.

Referenced by pci_cfgenable().

7.14.1.2 #define PCI_FUNCMAX 7

Definition at line 45 of file pcireg.h.

Referenced by legacy_pcib_identify(), pci_add_children(), pci_cfgenable(), and pci_pir_search_irq().

7.14.1.3 #define PCI_MAXHDRTYPE 2

Definition at line 47 of file pcireg.h.

Referenced by legacy_pcib_identify(), pci_add_children(), and pci_pir_search_irq().

7.14.1.4 #define PCI_REGMAX 255

Definition at line 46 of file pcireg.h.

Referenced by pci_cfgenable(), and pci_ioctl().

7.14.1.5 #define PCI_RID2BAR(rid) (((rid)-PCIR_BARS)/4)

Definition at line 120 of file pcireg.h.

7.14.1.6 #define PCI_SLOTMAX 31

Definition at line 44 of file pcireg.h.

Referenced by legacy_pcib_identify(), and pcib_maxslots().

7.14.1.7 #define PCIB_BCR_DISCARD_TIMER_SERREN 0x0800

Definition at line 339 of file pcireg.h.

7.14.1.8 #define PCIB_BCR_DISCARD_TIMER_STATUS 0x0400

Definition at line 338 of file pcireg.h.

7.14.1.9 #define PCIB_BCR_ISA_ENABLE 0x0004

Definition at line 331 of file pcireg.h.

7.14.1.10 #define PCIB_BCR_MASTER_ABORT_MODE 0x0020

Definition at line 333 of file pcireg.h.

7.14.1.11 #define PCIB_BCR_PERR_ENABLE 0x0001

Definition at line 329 of file pcireg.h.

7.14.1.12 #define PCIB_BCR_PRI_DISCARD_TIMEOUT 0x0100

Definition at line 336 of file pcireg.h.

7.14.1.13 #define PCIB_BCR_SEC_DISCARD_TIMEOUT 0x0200

Definition at line 337 of file pcireg.h.

7.14.1.14 #define PCIB_BCR_SECBUS_BACKTOBACK 0x0080

Definition at line 335 of file pcireg.h.

7.14.1.15 #define PCIB_BCR_SECBUS_RESET 0x0040

Definition at line 334 of file pcireg.h.

7.14.1.16 #define PCIB_BCR_SERR_ENABLE 0x0002

Definition at line 330 of file pcireg.h.

7.14.1.17 #define PCIB_BCR_VGA_ENABLE 0x0008

Definition at line 332 of file pcireg.h.

Referenced by pcib_alloc_resource().

7.14.1.18 #define PCIC_BASEPERIPH 0x08

Definition at line 261 of file pcireg.h.

Referenced by pci_cfg_save().

7.14.1.19 #define PCIC_BRIDGE 0x06

Definition at line 241 of file pcireg.h.

Referenced by eisab_probe(), isab_probe(), legacy_pcib_is_host_bridge(), pci_fixancient(), pci_hostb_-probe(), pcib_probe(), and pcibios_pcib_probe().

7.14.1.20 #define PCIC_CRYPTO 0x10

Definition at line 317 of file pcireg.h.

7.14.1.21 #define PCIC_DASP 0x11

Definition at line 322 of file pcireg.h.

7.14.1.22 #define PCIC_DISPLAY 0x03

Definition at line 224 of file pcireg.h.

Referenced by pci_cfg_save(), and vga_pci_probe().

7.14.1.23 #define PCIC_DOCKING 0x0a

Definition at line 278 of file pcireg.h.

7.14.1.24 #define PCIC_INPUTDEV 0x09

Definition at line 270 of file pcireg.h.

7.14.1.25 #define PCIC_INTELLIIO 0x0e

Definition at line 308 of file pcireg.h.

7.14.1.26 #define PCIC_MEMORY 0x05

Definition at line 236 of file pcireg.h.

Referenced by pci_cfg_save().

7.14.1.27 #define PCIC_MULTIMEDIA 0x04

Definition at line 230 of file pcireg.h.

7.14.1.28 #define PCIC_NETWORK 0x02

Definition at line 216 of file pcireg.h.

7.14.1.29 #define PCIC_OLD 0x00

Definition at line 199 of file pcireg.h.

Referenced by vga_pci_probe().

7.14.1.30 #define PCIC_OTHER 0xff

Definition at line 326 of file pcireg.h.

7.14.1.31 #define PCIC_PROCESSOR 0x0b

Definition at line 282 of file pcireg.h.

7.14.1.32 #define PCIC_SATCOM 0x0f

Definition at line 311 of file pcireg.h.

7.14.1.33 #define PCIC_SERIALBUS 0x0c

Definition at line 291 of file pcireg.h.

7.14.1.34 #define PCIC_SIMPLECOMM 0x07

Definition at line 253 of file pcireg.h.

7.14.1.35 #define PCIC_STORAGE 0x01

Definition at line 203 of file pcireg.h.
Referenced by pci_add_resources(), and pci_cfg_save().

7.14.1.36 #define PCIC_WIRELESS 0x0d

Definition at line 302 of file pcireg.h.

7.14.1.37 #define PCICAP_ID 0x0

Definition at line 93 of file pcireg.h.
Referenced by pci_find_extcap_method(), and pci_read_extcap().

7.14.1.38 #define PCICAP_NEXTPTR 0x1

Definition at line 94 of file pcireg.h.
Referenced by pci_find_extcap_method(), and pci_read_extcap().

7.14.1.39 #define PCIM_BIOS_ADDR_MASK 0xfffff800

Definition at line 137 of file pcireg.h.

7.14.1.40 #define PCIM_BIOS_ENABLE 0x01

Definition at line 136 of file pcireg.h.

7.14.1.41 #define PCIM_BMCSR_B3SUPP 0x40

Definition at line 384 of file pcireg.h.

7.14.1.42 #define PCIM_BMCSR_BPCE 0x80

Definition at line 385 of file pcireg.h.

7.14.1.43 #define PCIM_BRIO_16 0x0

Definition at line 157 of file pcireg.h.

7.14.1.44 #define PCIM_BRIO_32 0x1

Definition at line 158 of file pcireg.h.
Referenced by pcib_attach_common().

7.14.1.45 #define PCIM_BRIO_MASK 0xf

Definition at line 159 of file pcireg.h.
Referenced by pcib_attach_common().

7.14.1.46 #define PCIM_CIS_ADDR_MASK 0x0ffffff8

Definition at line 131 of file pcireg.h.

7.14.1.47 #define PCIM_CIS_ASI_BAR0 1

Definition at line 124 of file pcireg.h.

7.14.1.48 #define PCIM_CIS_ASI_BAR1 2

Definition at line 125 of file pcireg.h.

7.14.1.49 #define PCIM_CIS_ASI_BAR2 3

Definition at line 126 of file pcireg.h.

7.14.1.50 #define PCIM_CIS_ASI_BAR3 4

Definition at line 127 of file pcireg.h.

7.14.1.51 #define PCIM_CIS_ASI_BAR4 5

Definition at line 128 of file pcireg.h.

7.14.1.52 #define PCIM_CIS_ASI_BAR5 6

Definition at line 129 of file pcireg.h.

7.14.1.53 #define PCIM_CIS_ASI_MASK 0x7

Definition at line 122 of file pcireg.h.

7.14.1.54 #define PCIM_CIS_ASI_ROM 7

Definition at line 130 of file pcireg.h.

7.14.1.55 #define PCIM_CIS_ASI_TUPLE 0

Definition at line 123 of file pcireg.h.

7.14.1.56 #define PCIM_CIS_ROM_MASK 0xf0000000

Definition at line 132 of file pcireg.h.

7.14.1.57 #define PCIM_CMD_BACKTOBACK 0x0200

Definition at line 62 of file pcireg.h.

7.14.1.58 #define PCIM_CMD_BUSMASTEREN 0x0004

Definition at line 57 of file pcireg.h.

Referenced by pci_disable_busmaster_method(), and pci_enable_busmaster_method().

7.14.1.59 #define PCIM_CMD_MEMEN 0x0002

Definition at line 56 of file pcireg.h.

Referenced by pci_add_map(), pci_disable_io_method(), pci_enable_io_method(), pci_memen(), and pcib_attach_common().

7.14.1.60 #define PCIM_CMD_MWRICEN 0x0010

Definition at line 59 of file pcireg.h.

7.14.1.61 #define PCIM_CMD_PERRESPEN 0x0040

Definition at line 60 of file pcireg.h.

7.14.1.62 #define PCIM_CMD_PORTEN 0x0001

Definition at line 55 of file pcireg.h.

Referenced by pci_add_map(), pci_disable_io_method(), pci_enable_io_method(), pci_porten(), and pcib_attach_common().

7.14.1.63 #define PCIM_CMD_SERRESPEN 0x0100

Definition at line 61 of file pcireg.h.

7.14.1.64 #define PCIM_CMD_SPECIALEN 0x0008

Definition at line 58 of file pcireg.h.

7.14.1.65 #define PCIM_DEBUG_PORT_BAR 0xe000

Definition at line 516 of file pcireg.h.

7.14.1.66 #define PCIM_DEBUG_PORT_OFFSET 0x1FFF

Definition at line 515 of file pcireg.h.

7.14.1.67 #define PCIM_EXP_FLAGS_IRQ 0x3e00

Definition at line 532 of file pcireg.h.

7.14.1.68 #define PCIM_EXP_FLAGS_SLOT 0x0100

Definition at line 531 of file pcireg.h.

7.14.1.69 #define PCIM_EXP_FLAGS_TYPE 0x00F0

Definition at line 524 of file pcireg.h.

Referenced by pci_read_extcap().

7.14.1.70 #define PCIM_EXP_FLAGS_VERSION 0x000F

Definition at line 523 of file pcireg.h.

7.14.1.71 #define PCIM_EXP_TYPE_DOWNSTREAM_PORT 0x0060

Definition at line 529 of file pcireg.h.

7.14.1.72 #define PCIM_EXP_TYPE_ENDPOINT 0x0000

Definition at line 525 of file pcireg.h.

7.14.1.73 #define PCIM_EXP_TYPE_LEGACY_ENDPOINT 0x0010

Definition at line 526 of file pcireg.h.

7.14.1.74 #define PCIM_EXP_TYPE_PCI_BRIDGE 0x0070

Definition at line 530 of file pcireg.h.

7.14.1.75 #define PCIM_EXP_TYPE_ROOT_PORT 0x0040

Definition at line 527 of file pcireg.h.

Referenced by pci_read_extcap().

7.14.1.76 #define PCIM_EXP_TYPE_UPSTREAM_PORT 0x0050

Definition at line 528 of file pcireg.h.

7.14.1.77 #define PCIM_HDRTYPE 0x7f

Definition at line 84 of file pcireg.h.

Referenced by legacy_pcib_identify(), pci_add_children(), pci_find_extcap_method(), pci_pir_search_irq(), and pci_read_extcap().

7.14.1.78 #define PCIM_HDRTYPE_BRIDGE 0x01

Definition at line 86 of file pcireg.h.

7.14.1.79 #define PCIM_HDRTYPE_CARDBUS 0x02

Definition at line 87 of file pcireg.h.

7.14.1.80 #define PCIM_HDRTYPE_NORMAL 0x00

Definition at line 85 of file pcireg.h.

7.14.1.81 #define PCIM_HTCAP_ADDRESS_MAPPING 0xa000

Definition at line 498 of file pcireg.h.

7.14.1.82 #define PCIM_HTCAP_DIRECT_ROUTE 0xb000

Definition at line 500 of file pcireg.h.

7.14.1.83 #define PCIM_HTCAP_EXT_CONFIG_SPACE 0x9800

Definition at line 497 of file pcireg.h.

7.14.1.84 #define PCIM_HTCAP_HOST 0x2000

Definition at line 492 of file pcireg.h.

7.14.1.85 #define PCIM_HTCAP_INTERRUPT 0x8000

Definition at line 494 of file pcireg.h.

7.14.1.86 #define PCIM_HTCAP_MSI_MAPPING 0xa800

Definition at line 499 of file pcireg.h.

Referenced by pci_read_extcap().

7.14.1.87 #define PCIM_HTCAP_RETRY_MODE 0xc000

Definition at line 502 of file pcireg.h.

7.14.1.88 #define PCIM_HTCAP_REVISION_ID 0x8800

Definition at line 495 of file pcireg.h.

7.14.1.89 #define PCIM_HTCAP_SLAVE 0x0000

Definition at line 491 of file pcireg.h.

7.14.1.90 #define PCIM_HTCAP_SWITCH 0x4000

Definition at line 493 of file pcireg.h.

7.14.1.91 #define PCIM_HTCAP_UNITID_CLUMPING 0x9000

Definition at line 496 of file pcireg.h.

7.14.1.92 #define PCIM_HTCAP_VCSET 0xb800

Definition at line 501 of file pcireg.h.

7.14.1.93 #define PCIM_HTCMD_CAP_MASK 0xf800

Definition at line 490 of file pcireg.h.

Referenced by pci_read_extcap().

7.14.1.94 #define PCIM_HTCMD_MSI_ENABLE 0x0001

Definition at line 505 of file pcireg.h.

Referenced by pci_read_extcap().

7.14.1.95 #define PCIM_MFDEV 0x80

Definition at line 88 of file pcireg.h.

Referenced by legacy_pcib_identify(), pci_add_children(), pci_pir_search_irq(), and pci_read_device().

7.14.1.96 #define PCIM_MSICTRL_64BIT 0x0080

Definition at line 392 of file pcireg.h.

Referenced by pci_enable_msi(), pci_print_verbose(), and pci_resume_msi().

7.14.1.97 #define PCIM_MSICTRL_MMC_1 0x0000

Definition at line 401 of file pcireg.h.

7.14.1.98 #define PCIM_MSICTRL_MMC_16 0x0008

Definition at line 405 of file pcireg.h.

7.14.1.99 #define PCIM_MSICTRL_MMC_2 0x0002

Definition at line 402 of file pcireg.h.

7.14.1.100 #define PCIM_MSICTRL_MMC_32 0x000A

Definition at line 406 of file pcireg.h.

7.14.1.101 #define PCIM_MSICTRL_MMC_4 0x0004

Definition at line 403 of file pcireg.h.

7.14.1.102 #define PCIM_MSICTRL_MMC_8 0x0006

Definition at line 404 of file pcireg.h.

7.14.1.103 #define PCIM_MSICTRL_MMC_MASK 0x000E

Definition at line 400 of file pcireg.h.

Referenced by pci_read_extcap().

7.14.1.104 #define PCIM_MSICTRL_MME_1 0x0000

Definition at line 394 of file pcireg.h.

7.14.1.105 #define PCIM_MSICTRL_MME_16 0x0040

Definition at line 398 of file pcireg.h.

7.14.1.106 #define PCIM_MSICTRL_MME_2 0x0010

Definition at line 395 of file pcireg.h.

7.14.1.107 #define PCIM_MSICTRL_MME_32 0x0050

Definition at line 399 of file pcireg.h.

7.14.1.108 #define PCIM_MSICTRL_MME_4 0x0020

Definition at line 396 of file pcireg.h.

7.14.1.109 #define PCIM_MSICTRL_MME_8 0x0030

Definition at line 397 of file pcireg.h.

7.14.1.110 #define PCIM_MSICTRL_MME_MASK 0x0070

Definition at line 393 of file pcireg.h.
Referenced by pci_alloc_msi_method(), and pci_release_msi_method().

7.14.1.111 #define PCIM_MSICTRL_MSI_ENABLE 0x0001

Definition at line 407 of file pcireg.h.
Referenced by pci_enable_msi(), pci_release_msi_method(), and pci_resume_msi().

7.14.1.112 #define PCIM_MSICTRL_VECTOR 0x0100

Definition at line 391 of file pcireg.h.
Referenced by pci_print_verbose().

7.14.1.113 #define PCIM_MSIX_BIR_BAR_10 0

Definition at line 542 of file pcireg.h.

7.14.1.114 #define PCIM_MSIX_BIR_BAR_14 1

Definition at line 543 of file pcireg.h.

7.14.1.115 #define PCIM_MSIX_BIR_BAR_18 2

Definition at line 544 of file pcireg.h.

7.14.1.116 #define PCIM_MSIX_BIR_BAR_1C 3

Definition at line 545 of file pcireg.h.

7.14.1.117 #define PCIM_MSIX_BIR_BAR_20 4

Definition at line 546 of file pcireg.h.

7.14.1.118 #define PCIM_MSIX_BIR_BAR_24 5

Definition at line 547 of file pcireg.h.

7.14.1.119 #define PCIM_MSIX_BIR_MASK 0x7

Definition at line 541 of file pcireg.h.
Referenced by pci_read_extcap().

7.14.1.120 #define PCIM_MSIX_VCTRL_MASK 0x1

Definition at line 548 of file pcireg.h.
Referenced by pci_mask_msix(), and pci_unmask_msix().

7.14.1.121 #define PCIM_MSIXCTRL_FUNCTION_MASK 0x4000

Definition at line 537 of file pcireg.h.

7.14.1.122 #define PCIM_MSIXCTRL_MSIX_ENABLE 0x8000

Definition at line 536 of file pcireg.h.
Referenced by pci_release_msix().

7.14.1.123 #define PCIM_MSIXCTRL_TABLE_SIZE 0x07FF

Definition at line 538 of file pcireg.h.
Referenced by pci_read_extcap().

7.14.1.124 #define PCIM_PCAP_CLOCKMASK 0x00c0

Definition at line 350 of file pcireg.h.

7.14.1.125 #define PCIM_PCAP_D0PME 0x1000

Definition at line 354 of file pcireg.h.

7.14.1.126 #define PCIM_PCAP_D1PME 0x2000

Definition at line 355 of file pcireg.h.

7.14.1.127 #define PCIM_PCAP_D1SUPP 0x0200

Definition at line 352 of file pcireg.h.
Referenced by pci_print_verbose(), and pci_set_powerstate_method().

7.14.1.128 #define PCIM_PCAP_D2PME 0x4000

Definition at line 356 of file pcireg.h.

7.14.1.129 #define PCIM_PCAP_D2SUPP 0x0400

Definition at line 353 of file pcireg.h.
Referenced by pci_print_verbose(), and pci_set_powerstate_method().

7.14.1.130 #define PCIM_PCAP_DEVSPECINIT 0x0020

Definition at line 347 of file pcireg.h.

7.14.1.131 #define PCIM_PCAP_DYNCLOCK 0x0040

Definition at line 348 of file pcireg.h.

7.14.1.132 #define PCIM_PCAP_PMEREQCLK 0x0008

Definition at line 345 of file pcireg.h.

7.14.1.133 #define PCIM_PCAP_PMEREQPWR 0x0010

Definition at line 346 of file pcireg.h.

7.14.1.134 #define PCIM_PCAP_REQFULLCLOCK 0x0100

Definition at line 351 of file pcireg.h.

7.14.1.135 #define PCIM_PCAP_SECCLOCK 0x00c0

Definition at line 349 of file pcireg.h.

7.14.1.136 #define PCIM_PCAP_SPEC 0x0007

Definition at line 344 of file pcireg.h.
Referenced by pci_print_verbose().

7.14.1.137 #define PCIM_PMCSR_B2SUPP 0x20

Definition at line 383 of file pcireg.h.

7.14.1.138 #define PCIM_PMCSR_DCLOCK 0x10

Definition at line 382 of file pcireg.h.

7.14.1.139 #define PCIM_PSTAT_D0 0x0000

Definition at line 359 of file pcireg.h.
Referenced by pci_get_powerstate_method(), and pci_set_powerstate_method().

7.14.1.140 #define PCIM_PSTAT_D0HEAT 0x0800

Definition at line 370 of file pcireg.h.

7.14.1.141 #define PCIM_PSTAT_D0POWER 0x0000

Definition at line 366 of file pcireg.h.

7.14.1.142 #define PCIM_PSTAT_D1 0x0001

Definition at line 360 of file pcireg.h.
Referenced by pci_get_powerstate_method(), and pci_set_powerstate_method().

7.14.1.143 #define PCIM_PSTAT_D1HEAT 0x1000

Definition at line 371 of file pcireg.h.

7.14.1.144 #define PCIM_PSTAT_D1POWER 0x0200

Definition at line 367 of file pcireg.h.

7.14.1.145 #define PCIM_PSTAT_D2 0x0002

Definition at line 361 of file pcireg.h.

Referenced by pci_get_powerstate_method(), and pci_set_powerstate_method().

7.14.1.146 #define PCIM_PSTAT_D2HEAT 0x1200

Definition at line 372 of file pcireg.h.

7.14.1.147 #define PCIM_PSTAT_D2POWER 0x0400

Definition at line 368 of file pcireg.h.

7.14.1.148 #define PCIM_PSTAT_D3 0x0003

Definition at line 362 of file pcireg.h.

Referenced by pci_get_powerstate_method(), and pci_set_powerstate_method().

7.14.1.149 #define PCIM_PSTAT_D3HEAT 0x1400

Definition at line 373 of file pcireg.h.

7.14.1.150 #define PCIM_PSTAT_D3POWER 0x0600

Definition at line 369 of file pcireg.h.

7.14.1.151 #define PCIM_PSTAT_DATADIV10 0x2000

Definition at line 375 of file pcireg.h.

7.14.1.152 #define PCIM_PSTAT_DATADIV100 0x4000

Definition at line 376 of file pcireg.h.

7.14.1.153 #define PCIM_PSTAT_DATADIV1000 0x6000

Definition at line 377 of file pcireg.h.

7.14.1.154 #define PCIM_PSTAT_DATADIVMASK 0x6000

Definition at line 378 of file pcireg.h.

7.14.1.155 #define PCIM_PSTAT_DATAUNKN 0x0000

Definition at line 374 of file pcireg.h.

7.14.1.156 #define PCIM_PSTAT_DMASK 0x0003

Definition at line 363 of file pcireg.h.

Referenced by pci_get_powerstate_method(), pci_print_verbose(), and pci_set_powerstate_method().

7.14.1.157 #define PCIM_PSTAT_PME 0x8000

Definition at line 379 of file pcireg.h.

7.14.1.158 #define PCIM_PSTAT_PMEENABLE 0x0100

Definition at line 365 of file pcireg.h.

7.14.1.159 #define PCIM_PSTAT_REPENABLE 0x0010

Definition at line 364 of file pcireg.h.

7.14.1.160 #define PCIM_STATUS_66CAPABLE 0x0020

Definition at line 65 of file pcireg.h.

7.14.1.161 #define PCIM_STATUS_BACKTOBACK 0x0080

Definition at line 66 of file pcireg.h.

7.14.1.162 #define PCIM_STATUS_CAPPRESENT 0x0010

Definition at line 64 of file pcireg.h.

Referenced by pci_find_extcap_method(), and pci_read_device().

7.14.1.163 #define PCIM_STATUS_PERR 0x8000

Definition at line 76 of file pcireg.h.

7.14.1.164 #define PCIM_STATUS_PERRREPORT 0x0100

Definition at line 67 of file pcireg.h.

7.14.1.165 #define PCIM_STATUS_RMABORT 0x2000

Definition at line 74 of file pcireg.h.

7.14.1.166 #define PCIM_STATUS_RTABORT 0x1000

Definition at line 73 of file pcireg.h.

7.14.1.167 #define PCIM_STATUS_SEL_FAST 0x0000

Definition at line 68 of file pcireg.h.

7.14.1.168 #define PCIM_STATUS_SEL_MASK 0x0600

Definition at line 71 of file pcireg.h.

7.14.1.169 #define PCIM_STATUS_SEL_MEDIMUM 0x0200

Definition at line 69 of file pcireg.h.

7.14.1.170 #define PCIM_STATUS_SEL_SLOW 0x0400

Definition at line 70 of file pcireg.h.

7.14.1.171 #define PCIM_STATUS_SERR 0x4000

Definition at line 75 of file pcireg.h.

7.14.1.172 #define PCIM_STATUS_STABORT 0x0800

Definition at line 72 of file pcireg.h.

7.14.1.173 #define PCIP_SERIALBUS_USB_EHCI 0x20

Definition at line 298 of file pcireg.h.

7.14.1.174 #define PCIP_SERIALBUS_USB_OHCI 0x10

Definition at line 297 of file pcireg.h.

7.14.1.175 #define PCIP_SERIALBUS_USB_UHCI 0x00

Definition at line 296 of file pcireg.h.

7.14.1.176 #define PCIP_SIMPLECOMM_UART_16550A 0x02

Definition at line 255 of file pcireg.h.

7.14.1.177 #define PCIP_STORAGE_IDE_MASTERDEV 0x80

Definition at line 210 of file pcireg.h.

Referenced by pci_add_resources().

7.14.1.178 #define PCIP_STORAGE_IDE_MODEPRIM 0x01

Definition at line 206 of file pcireg.h.

Referenced by pci_ata_maps().

7.14.1.179 #define PCIP_STORAGE_IDE_MODESEC 0x04

Definition at line 208 of file pcireg.h.

Referenced by pci_ata_maps().

7.14.1.180 #define PCIP_STORAGE_IDE_PROGINDPRIM 0x02

Definition at line 207 of file pcireg.h.

7.14.1.181 #define PCIP_STORAGE_IDE_PROGINDSEC 0x08

Definition at line 209 of file pcireg.h.

7.14.1.182 #define PCIR_BAR(x) (PCIR_BARS + (x) ∗ 4)

Definition at line 119 of file pcireg.h.

Referenced by pci_add_resources(), pci_alloc_resource(), pci_ata_maps(), pci_cfg_restore(), pci_cfg_-save(), and pci_read_extcap().

7.14.1.183 #define PCIR_BARS 0x10

Definition at line 118 of file pcireg.h.

7.14.1.184 #define PCIR_BIOS 0x30

Definition at line 135 of file pcireg.h.

Referenced by pci_cfg_restore(), and pci_cfg_save().

7.14.1.185 #define PCIR_BIST 0x0f

Definition at line 89 of file pcireg.h.

7.14.1.186 #define PCIR_BRIDGECTL_1 0x3e

Definition at line 169 of file pcireg.h.

Referenced by pcib_attach_common().

7.14.1.187 #define PCIR_BRIDGECTL_2 0x3e

Definition at line 190 of file pcireg.h.

7.14.1.188 #define PCIR_CACHELNSZ 0x0c

Definition at line 81 of file pcireg.h.

Referenced by pci_cfg_restore(), pci_cfg_save(), and pci_read_device().

7.14.1.189 #define PCIR_CAP_PTR 0x34

Definition at line 138 of file pcireg.h.

Referenced by pci_find_extcap_method(), and pci_read_extcap().

7.14.1.190 #define PCIR_CAP_PTR_2 0x14

Definition at line 173 of file pcireg.h.

Referenced by pci_find_extcap_method(), and pci_read_extcap().

7.14.1.191 #define PCIR_CIS 0x28

Definition at line 121 of file pcireg.h.

7.14.1.192 #define PCIR_CLASS 0x0b

Definition at line 80 of file pcireg.h.

Referenced by pci_cfg_save(), and pci_read_device().

7.14.1.193 #define PCIR_COMMAND 0x04

Definition at line 54 of file pcireg.h.

Referenced by pci_add_map(), pci_cfg_restore(), pci_cfg_save(), pci_clear_command_bit(), pci_disable_io_method(), pci_enable_io_method(), pci_memen(), pci_porten(), pci_read_device(), pci_set_command_bit(), and pcib_attach_common().

7.14.1.194 #define PCIR_DEBUG_PORT 0x2

Definition at line 514 of file pcireg.h.

7.14.1.195 #define PCIR_DEVICE 0x02

Definition at line 53 of file pcireg.h.

Referenced by pci_cfg_save(), pci_cfgregopen(), and pci_read_device().

7.14.1.196 #define PCIR_DEVVENDOR 0x00

Definition at line 51 of file pcireg.h.

Referenced by host_pcib_get_busno(), legacy_pcib_identify(), pci_pir_search_irq(), and pci_read_device().

7.14.1.197 #define PCIR_EXPRESS_FLAGS 0x2

Definition at line 522 of file pcireg.h.

Referenced by pci_read_extcap().

7.14.1.198 #define PCIR_HDRTYPE 0x0e

Definition at line 83 of file pcireg.h.

Referenced by legacy_pcib_identify(), pci_add_children(), pci_pir_search_irq(), and pci_read_device().

7.14.1.199 #define PCIR_HT_COMMAND 0x2

Definition at line 489 of file pcireg.h.

Referenced by pci_read_extcap().

7.14.1.200 #define PCIR_HTMSI_ADDRESS_HI 0x8

Definition at line 507 of file pcireg.h.

Referenced by pci_read_extcap().

7.14.1.201 #define PCIR_HTMSI_ADDRESS_LO 0x4

Definition at line 506 of file pcireg.h.

Referenced by pci_read_extcap().

7.14.1.202 #define PCIR_INTLINE 0x3c

Definition at line 139 of file pcireg.h.

Referenced by pci_assign_interrupt(), pci_cfg_restore(), pci_cfg_save(), pci_cfgregread(), pci_pir_-search_irq(), and pci_read_device().

7.14.1.203 #define PCIR_INTPIN 0x3d

Definition at line 140 of file pcireg.h.

Referenced by pci_cfg_restore(), pci_cfg_save(), pci_pir_search_irq(), and pci_read_device().

7.14.1.204 #define PCIR_IOBASE0_2 0x2c

Definition at line 185 of file pcireg.h.

7.14.1.205 #define PCIR_IOBASE1_2 0x34

Definition at line 187 of file pcireg.h.

7.14.1.206 #define PCIR_IOBASEH_1 0x30

Definition at line 155 of file pcireg.h.

Referenced by pcib_attach_common().

7.14.1.207 #define PCIR_IOBASEL_1 0x1c

Definition at line 153 of file pcireg.h.

Referenced by pcib_attach_common().

7.14.1.208 #define PCIR_IOLIMIT0_2 0x30

Definition at line 186 of file pcireg.h.

7.14.1.209 #define PCIR_IOLIMIT1_2 0x38

Definition at line 188 of file pcireg.h.

7.14.1.210 #define PCIR_IOLIMITH_1 0x32

Definition at line 156 of file pcireg.h.

Referenced by pcib_attach_common().

7.14.1.211 #define PCIR_IOLIMITL_1 0x1d

Definition at line 154 of file pcireg.h.

Referenced by pcib_attach_common().

7.14.1.212 #define PCIR_LATTIMER 0x0d

Definition at line 82 of file pcireg.h.

Referenced by pci_cfg_restore(), pci_cfg_save(), and pci_read_device().

7.14.1.213 #define PCIR_MAXLAT 0x3f

Definition at line 142 of file pcireg.h.

Referenced by pci_cfg_restore(), pci_cfg_save(), and pci_read_device().

7.14.1.214 #define PCIR_MEMBASE0_2 0x1c

Definition at line 181 of file pcireg.h.

7.14.1.215 #define PCIR_MEMBASE1_2 0x24

Definition at line 183 of file pcireg.h.

7.14.1.216 #define PCIR_MEMBASE_1 0x20

Definition at line 161 of file pcireg.h.

Referenced by pcib_attach_common().

7.14.1.217 #define PCIR_MEMLIMIT0_2 0x20

Definition at line 182 of file pcireg.h.

7.14.1.218 #define PCIR_MEMLIMIT1_2 0x28

Definition at line 184 of file pcireg.h.

7.14.1.219 #define PCIR_MEMLIMIT_1 0x22

Definition at line 162 of file pcireg.h.

Referenced by pcib_attach_common().

7.14.1.220 #define PCIR_MINGNT 0x3e

Definition at line 141 of file pcireg.h.

Referenced by pci_cfg_restore(), pci_cfg_save(), and pci_read_device().

7.14.1.221 #define PCIR_MSI_ADDR 0x4

Definition at line 408 of file pcireg.h.

Referenced by pci_enable_msi(), and pci_resume_msi().

7.14.1.222 #define PCIR_MSI_ADDR_HIGH 0x8

Definition at line 409 of file pcireg.h.

Referenced by pci_enable_msi(), and pci_resume_msi().

7.14.1.223 #define PCIR_MSI_CTRL 0x2

Definition at line 390 of file pcireg.h.

Referenced by pci_alloc_msi_method(), pci_enable_msi(), pci_read_extcap(), pci_release_msi_method(), and pci_resume_msi().

7.14.1.224 #define PCIR_MSI_DATA 0x8

Definition at line 410 of file pcireg.h.

Referenced by pci_enable_msi(), and pci_resume_msi().

7.14.1.225 #define PCIR_MSI_DATA_64BIT 0xc

Definition at line 411 of file pcireg.h.

Referenced by pci_enable_msi(), and pci_resume_msi().

7.14.1.226 #define PCIR_MSI_MASK 0x10

Definition at line 412 of file pcireg.h.

7.14.1.227 #define PCIR_MSI_PENDING 0x14

Definition at line 413 of file pcireg.h.

7.14.1.228 #define PCIR_MSIX_CTRL 0x2

Definition at line 535 of file pcireg.h.

Referenced by pci_read_extcap(), and pci_release_msix().

7.14.1.229 #define PCIR_MSIX_PBA 0x8

Definition at line 540 of file pcireg.h.

Referenced by pci_read_extcap().

7.14.1.230 #define PCIR_MSIX_TABLE 0x4

Definition at line 539 of file pcireg.h.

Referenced by pci_read_extcap().

7.14.1.231 #define PCIR_PCCARDIF_2 0x44

Definition at line 195 of file pcireg.h.

7.14.1.232 #define PCIR_PMBASEH_1 0x28

Definition at line 166 of file pcireg.h.

Referenced by pcib_attach_common().

7.14.1.233 #define PCIR_PMBASEL_1 0x24

Definition at line 164 of file pcireg.h.

Referenced by pcib_attach_common().

7.14.1.234 #define PCIR_PMLIMITH_1 0x2c

Definition at line 167 of file pcireg.h.

Referenced by pcib_attach_common().

7.14.1.235 #define PCIR_PMLIMITL_1 0x26

Definition at line 165 of file pcireg.h.

Referenced by pcib_attach_common().

7.14.1.236 #define PCIR_POWER_CAP 0x2

Definition at line 343 of file pcireg.h.

Referenced by pci_read_extcap().

7.14.1.237 #define PCIR_POWER_DATA 0x7

Definition at line 387 of file pcireg.h.

Referenced by pci_read_extcap().

7.14.1.238 #define PCIR_POWER_PMCSR 0x6

Definition at line 381 of file pcireg.h.

Referenced by pci_read_extcap().

7.14.1.239 #define PCIR_POWER_STATUS 0x4

Definition at line 358 of file pcireg.h.

Referenced by pci_read_extcap().

7.14.1.240 #define PCIR_PRIBUS_1 0x18

Definition at line 148 of file pcireg.h.

7.14.1.241 #define PCIR_PRIBUS_2 0x18

Definition at line 176 of file pcireg.h.

7.14.1.242 #define PCIR_PROGIF 0x09

Definition at line 78 of file pcireg.h.

Referenced by pci_ata_maps(), pci_cfg_restore(), pci_cfg_save(), pci_read_device(), and pcib_attach_-common().

7.14.1.243 #define PCIR_REVID 0x08

Definition at line 77 of file pcireg.h.

Referenced by pci_cfg_restore(), pci_cfg_save(), and pci_read_device().

7.14.1.244 #define PCIR_SECBUS_1 0x19

Definition at line 149 of file pcireg.h.

Referenced by pcib_attach_common(), and pcibios_pcib_probe().

7.14.1.245 #define PCIR_SECBUS_2 0x19

Definition at line 177 of file pcireg.h.

7.14.1.246 #define PCIR_SECLAT_1 0x1b

Definition at line 151 of file pcireg.h.

Referenced by pcib_attach_common().

7.14.1.247 #define PCIR_SECLAT_2 0x1b

Definition at line 179 of file pcireg.h.

7.14.1.248 #define PCIR_SECSTAT_1 0x1e

Definition at line 146 of file pcireg.h.

Referenced by pcib_attach_common().

7.14.1.249 #define PCIR_SECSTAT_2 0x16

Definition at line 174 of file pcireg.h.

7.14.1.250 #define PCIR_STATUS 0x06

Definition at line 63 of file pcireg.h.

Referenced by pci_find_extcap_method(), and pci_read_device().

7.14.1.251 #define PCIR_SUBBUS_1 0x1a

Definition at line 150 of file pcireg.h.

Referenced by pcib_attach_common().

7.14.1.252 #define PCIR_SUBBUS_2 0x1a

Definition at line 178 of file pcireg.h.

7.14.1.253 #define PCIR_SUBCLASS 0x0a

Definition at line 79 of file pcireg.h.

Referenced by legacy_pcib_identify(), pci_cfg_save(), and pci_read_device().

7.14.1.254 #define PCIR_SUBDEV_0 0x2e

Definition at line 134 of file pcireg.h.

Referenced by pci_cfg_save(), and pci_hdrtypedata().

7.14.1.255 #define PCIR_SUBDEV_2 0x42

Definition at line 193 of file pcireg.h.

Referenced by pci_hdrtypedata().

7.14.1.256 #define PCIR_SUBVEND_0 0x2c

Definition at line 133 of file pcireg.h.

Referenced by pci_cfg_save(), and pci_hdrtypedata().

7.14.1.257 #define PCIR_SUBVEND_2 0x40

Definition at line 192 of file pcireg.h.

Referenced by pci_hdrtypedata().

7.14.1.258 #define PCIR_SUBVENDCAP_ID 0x4

Definition at line 519 of file pcireg.h.

Referenced by pci_read_extcap().

7.14.1.259 #define PCIR_VENDOR 0x00

Definition at line 52 of file pcireg.h.

Referenced by pci_cfg_save(), pci_cfgregopen(), and pci_read_device().

7.14.1.260 #define PCIR_VENDOR_DATA 0x3

Definition at line 511 of file pcireg.h.

7.14.1.261 #define PCIR_VENDOR_LENGTH 0x2

Definition at line 510 of file pcireg.h.

7.14.1.262 #define PCIS_BASEPERIPH_DMA 0x01

Definition at line 263 of file pcireg.h.

7.14.1.263 #define PCIS_BASEPERIPH_OTHER 0x80

Definition at line 268 of file pcireg.h.

7.14.1.264 #define PCIS_BASEPERIPH_PCIHOT 0x04

Definition at line 266 of file pcireg.h.

7.14.1.265 #define PCIS_BASEPERIPH_PIC 0x00

Definition at line 262 of file pcireg.h.

7.14.1.266 #define PCIS_BASEPERIPH_RTC 0x03

Definition at line 265 of file pcireg.h.

7.14.1.267 #define PCIS_BASEPERIPH_SDHC 0x05

Definition at line 267 of file pcireg.h.

7.14.1.268 #define PCIS_BASEPERIPH_TIMER 0x02

Definition at line 264 of file pcireg.h.

7.14.1.269 #define PCIS_BRIDGE_CARDBUS 0x07

Definition at line 249 of file pcireg.h.

7.14.1.270 #define PCIS_BRIDGE_EISA 0x02

Definition at line 244 of file pcireg.h.

Referenced by eisab_probe().

7.14.1.271 #define PCIS_BRIDGE_HOST 0x00

Definition at line 242 of file pcireg.h.

Referenced by legacy_pcib_is_host_bridge(), and pci_hostb_probe().

7.14.1.272 #define PCIS_BRIDGE_ISA 0x01

Definition at line 243 of file pcireg.h.

Referenced by isab_probe().

7.14.1.273 #define PCIS_BRIDGE_MCA 0x03

Definition at line 245 of file pcireg.h.

7.14.1.274 #define PCIS_BRIDGE_NUBUS 0x06

Definition at line 248 of file pcireg.h.

7.14.1.275 #define PCIS_BRIDGE_OTHER 0x80

Definition at line 251 of file pcireg.h.

7.14.1.276 #define PCIS_BRIDGE_PCI 0x04

Definition at line 246 of file pcireg.h.

Referenced by pci_fixancient(), pcib_probe(), and pcibios_pcib_probe().

7.14.1.277 #define PCIS_BRIDGE_PCMCIA 0x05

Definition at line 247 of file pcireg.h.

7.14.1.278 #define PCIS_BRIDGE_RACEWAY 0x08

Definition at line 250 of file pcireg.h.

7.14.1.279 #define PCIS_CRYPTO_ENTERTAIN 0x10

Definition at line 319 of file pcireg.h.

7.14.1.280 #define PCIS_CRYPTO_NETCOMP 0x00

Definition at line 318 of file pcireg.h.

7.14.1.281 #define PCIS_CRYPTO_OTHER 0x80

Definition at line 320 of file pcireg.h.

7.14.1.282 #define PCIS_DASP_DPIO 0x00

Definition at line 323 of file pcireg.h.

7.14.1.283 #define PCIS_DASP_OTHER 0x80

Definition at line 324 of file pcireg.h.

7.14.1.284 #define PCIS_DISPLAY_3D 0x02

Definition at line 227 of file pcireg.h.

7.14.1.285 #define PCIS_DISPLAY_OTHER 0x80

Definition at line 228 of file pcireg.h.

7.14.1.286 #define PCIS_DISPLAY_VGA 0x00

Definition at line 225 of file pcireg.h.

7.14.1.287 #define PCIS_DISPLAY_XGA 0x01

Definition at line 226 of file pcireg.h.

7.14.1.288 #define PCIS_DOCKING_GENERIC 0x00

Definition at line 279 of file pcireg.h.

7.14.1.289 #define PCIS_DOCKING_OTHER 0x80

Definition at line 280 of file pcireg.h.

7.14.1.290 #define PCIS_INPUTDEV_DIGITIZER 0x01

Definition at line 272 of file pcireg.h.

7.14.1.291 #define PCIS_INPUTDEV_GAMEPORT 0x04

Definition at line 275 of file pcireg.h.

7.14.1.292 #define PCIS_INPUTDEV_KEYBOARD 0x00

Definition at line 271 of file pcireg.h.

7.14.1.293 #define PCIS_INPUTDEV_MOUSE 0x02

Definition at line 273 of file pcireg.h.

7.14.1.294 #define PCIS_INPUTDEV_OTHER 0x80

Definition at line 276 of file pcireg.h.

7.14.1.295 #define PCIS_INPUTDEV_SCANNER 0x03

Definition at line 274 of file pcireg.h.

7.14.1.296 #define PCIS_INTELLIIO_I2O 0x00

Definition at line 309 of file pcireg.h.

7.14.1.297 #define PCIS_MEMORY_FLASH 0x01

Definition at line 238 of file pcireg.h.

7.14.1.298 #define PCIS_MEMORY_OTHER 0x80

Definition at line 239 of file pcireg.h.

7.14.1.299 #define PCIS_MEMORY_RAM 0x00

Definition at line 237 of file pcireg.h.

7.14.1.300 #define PCIS_MULTIMEDIA_AUDIO 0x01

Definition at line 232 of file pcireg.h.

7.14.1.301 #define PCIS_MULTIMEDIA_OTHER 0x80

Definition at line 234 of file pcireg.h.

7.14.1.302 #define PCIS_MULTIMEDIA_TELE 0x02

Definition at line 233 of file pcireg.h.

7.14.1.303 #define PCIS_MULTIMEDIA_VIDEO 0x00

Definition at line 231 of file pcireg.h.

7.14.1.304 #define PCIS_NETWORK_ATM 0x03

Definition at line 220 of file pcireg.h.

7.14.1.305 #define PCIS_NETWORK_ETHERNET 0x00

Definition at line 217 of file pcireg.h.

7.14.1.306 #define PCIS_NETWORK_FDDI 0x02

Definition at line 219 of file pcireg.h.

7.14.1.307 #define PCIS_NETWORK_ISDN 0x04

Definition at line 221 of file pcireg.h.

7.14.1.308 #define PCIS_NETWORK_OTHER 0x80

Definition at line 222 of file pcireg.h.

7.14.1.309 #define PCIS_NETWORK_TOKENRING 0x01

Definition at line 218 of file pcireg.h.

7.14.1.310 #define PCIS_OLD_NONVGA 0x00

Definition at line 200 of file pcireg.h.

7.14.1.311 #define PCIS_OLD_VGA 0x01

Definition at line 201 of file pcireg.h.
Referenced by vga_pci_probe().

7.14.1.312 #define PCIS_PROCESSOR_386 0x00

Definition at line 283 of file pcireg.h.

7.14.1.313 #define PCIS_PROCESSOR_486 0x01

Definition at line 284 of file pcireg.h.

7.14.1.314 #define PCIS_PROCESSOR_ALPHA 0x10

Definition at line 286 of file pcireg.h.

7.14.1.315 #define PCIS_PROCESSOR_COPROC 0x40

Definition at line 289 of file pcireg.h.

7.14.1.316 #define PCIS_PROCESSOR_MIPS 0x30

Definition at line 288 of file pcireg.h.

7.14.1.317 #define PCIS_PROCESSOR_PENTIUM 0x02

Definition at line 285 of file pcireg.h.

7.14.1.318 #define PCIS_PROCESSOR_POWERPC 0x20

Definition at line 287 of file pcireg.h.

7.14.1.319 #define PCIS_SATCOM_AUDIO 0x02

Definition at line 313 of file pcireg.h.

7.14.1.320 #define PCIS_SATCOM_DATA 0x04

Definition at line 315 of file pcireg.h.

7.14.1.321 #define PCIS_SATCOM_TV 0x01

Definition at line 312 of file pcireg.h.

7.14.1.322 #define PCIS_SATCOM_VOICE 0x03

Definition at line 314 of file pcireg.h.

7.14.1.323 #define PCIS_SERIALBUS_ACCESS 0x01

Definition at line 293 of file pcireg.h.

7.14.1.324 #define PCIS_SERIALBUS_FC 0x04

Definition at line 299 of file pcireg.h.

7.14.1.325 #define PCIS_SERIALBUS_FW 0x00

Definition at line 292 of file pcireg.h.

7.14.1.326 #define PCIS_SERIALBUS_SMBUS 0x05

Definition at line 300 of file pcireg.h.

7.14.1.327 #define PCIS_SERIALBUS_SSA 0x02

Definition at line 294 of file pcireg.h.

7.14.1.328 #define PCIS_SERIALBUS_USB 0x03

Definition at line 295 of file pcireg.h.

7.14.1.329 #define PCIS_SIMPLECOMM_MODEM 0x03

Definition at line 258 of file pcireg.h.

7.14.1.330 #define PCIS_SIMPLECOMM_MULSER 0x02

Definition at line 257 of file pcireg.h.

7.14.1.331 #define PCIS_SIMPLECOMM_OTHER 0x80

Definition at line 259 of file pcireg.h.

7.14.1.332 #define PCIS_SIMPLECOMM_PAR 0x01

Definition at line 256 of file pcireg.h.

7.14.1.333 #define PCIS_SIMPLECOMM_UART 0x00

Definition at line 254 of file pcireg.h.

7.14.1.334 #define PCIS_STORAGE_FLOPPY 0x02

Definition at line 211 of file pcireg.h.

7.14.1.335 #define PCIS_STORAGE_IDE 0x01

Definition at line 205 of file pcireg.h.

Referenced by pci_add_resources().

7.14.1.336 #define PCIS_STORAGE_IPI 0x03

Definition at line 212 of file pcireg.h.

7.14.1.337 #define PCIS_STORAGE_OTHER 0x80

Definition at line 214 of file pcireg.h.

7.14.1.338 #define PCIS_STORAGE_RAID 0x04

Definition at line 213 of file pcireg.h.

7.14.1.339 #define PCIS_STORAGE_SCSI 0x00

Definition at line 204 of file pcireg.h.

7.14.1.340 #define PCIS_WIRELESS_IR 0x01

Definition at line 304 of file pcireg.h.

7.14.1.341 #define PCIS_WIRELESS_IRDA 0x00

Definition at line 303 of file pcireg.h.

7.14.1.342 #define PCIS_WIRELESS_OTHER 0x80

Definition at line 306 of file pcireg.h.

7.14.1.343 #define PCIS_WIRELESS_RF 0x10

Definition at line 305 of file pcireg.h.

7.14.1.344 #define PCIXM_BRIDGE_STATUS_133CAP 0x00020000

Definition at line 479 of file pcireg.h.

7.14.1.345 #define PCIXM_BRIDGE_STATUS_266CAP 0x40000000

Definition at line 485 of file pcireg.h.

7.14.1.346 #define PCIXM_BRIDGE_STATUS_533CAP 0x80000000

Definition at line 486 of file pcireg.h.

7.14.1.347 #define PCIXM_BRIDGE_STATUS_64BIT 0x00010000

Definition at line 478 of file pcireg.h.

7.14.1.348 #define PCIXM_BRIDGE_STATUS_BUS 0x0000FF00

Definition at line 477 of file pcireg.h.

7.14.1.349 #define PCIXM_BRIDGE_STATUS_DEVFN 0x000000FF

Definition at line 476 of file pcireg.h.

7.14.1.350 #define PCIXM_BRIDGE_STATUS_DEVID_MSGCAP 0x20000000

Definition at line 484 of file pcireg.h.

7.14.1.351 #define PCIXM_BRIDGE_STATUS_SC_DISCARDED 0x00040000

Definition at line 480 of file pcireg.h.

7.14.1.352 #define PCIXM_BRIDGE_STATUS_SC_OVERRUN 0x00100000

Definition at line 482 of file pcireg.h.

7.14.1.353 #define PCIXM_BRIDGE_STATUS_SR_DELAYED 0x00200000

Definition at line 483 of file pcireg.h.

7.14.1.354 #define PCIXM_BRIDGE_STATUS_UNEXP_SC 0x00080000

Definition at line 481 of file pcireg.h.

7.14.1.355 #define PCIXM_COMMAND_DPERR_E 0x0001

Definition at line 419 of file pcireg.h.

7.14.1.356 #define PCIXM_COMMAND_ERO 0x0002

Definition at line 420 of file pcireg.h.

7.14.1.357 #define PCIXM_COMMAND_MAX_READ 0x000c

Definition at line 421 of file pcireg.h.

7.14.1.358 #define PCIXM_COMMAND_MAX_READ_1024 0x0004

Definition at line 423 of file pcireg.h.

7.14.1.359 #define PCIXM_COMMAND_MAX_READ_2048 0x0008

Definition at line 424 of file pcireg.h.

7.14.1.360 #define PCIXM_COMMAND_MAX_READ_4096 0x000c

Definition at line 425 of file pcireg.h.

7.14.1.361 #define PCIXM_COMMAND_MAX_READ_512 0x0000

Definition at line 422 of file pcireg.h.

7.14.1.362 #define PCIXM_COMMAND_MAX_SPLITS 0x0070

Definition at line 426 of file pcireg.h.

7.14.1.363 #define PCIXM_COMMAND_MAX_SPLITS_1 0x0000

Definition at line 427 of file pcireg.h.

7.14.1.364 #define PCIXM_COMMAND_MAX_SPLITS_12 0x0050

Definition at line 432 of file pcireg.h.

7.14.1.365 #define PCIXM_COMMAND_MAX_SPLITS_16 0x0060

Definition at line 433 of file pcireg.h.

7.14.1.366 #define PCIXM_COMMAND_MAX_SPLITS_2 0x0010

Definition at line 428 of file pcireg.h.

7.14.1.367 #define PCIXM_COMMAND_MAX_SPLITS_3 0x0020

Definition at line 429 of file pcireg.h.

7.14.1.368 #define PCIXM_COMMAND_MAX_SPLITS_32 0x0070

Definition at line 434 of file pcireg.h.

7.14.1.369 #define PCIXM_COMMAND_MAX_SPLITS_4 0x0030

Definition at line 430 of file pcireg.h.

7.14.1.370 #define PCIXM_COMMAND_MAX_SPLITS_8 0x0040

Definition at line 431 of file pcireg.h.

7.14.1.371 #define PCIXM_COMMAND_VERSION 0x3000

Definition at line 435 of file pcireg.h.

7.14.1.372 #define PCIXM_SEC_STATUS_133CAP 0x0002

Definition at line 466 of file pcireg.h.

7.14.1.373 #define PCIXM_SEC_STATUS_266CAP 0x4000

Definition at line 473 of file pcireg.h.

7.14.1.374 #define PCIXM_SEC_STATUS_533CAP 0x8000

Definition at line 474 of file pcireg.h.

7.14.1.375 #define PCIXM_SEC_STATUS_64BIT 0x0001

Definition at line 465 of file pcireg.h.

7.14.1.376 #define PCIXM_SEC_STATUS_BUS_MODE 0x03c0

Definition at line 471 of file pcireg.h.

7.14.1.377 #define PCIXM_SEC_STATUS_SC_DISC 0x0004

Definition at line 467 of file pcireg.h.

7.14.1.378 #define PCIXM_SEC_STATUS_SC_OVERRUN 0x0010

Definition at line 469 of file pcireg.h.

7.14.1.379 #define PCIXM_SEC_STATUS_SR_DELAYED 0x0020

Definition at line 470 of file pcireg.h.

7.14.1.380 #define PCIXM_SEC_STATUS_UNEXP_SC 0x0008

Definition at line 468 of file pcireg.h.

7.14.1.381 #define PCIXM_SEC_STATUS_VERSION 0x3000

Definition at line 472 of file pcireg.h.

7.14.1.382 #define PCIXM_STATUS_133CAP 0x00020000

Definition at line 440 of file pcireg.h.

7.14.1.383 #define PCIXM_STATUS_266CAP 0x40000000

Definition at line 460 of file pcireg.h.

7.14.1.384 #define PCIXM_STATUS_533CAP 0x80000000

Definition at line 461 of file pcireg.h.

7.14.1.385 #define PCIXM_STATUS_64BIT 0x00010000

Definition at line 439 of file pcireg.h.

7.14.1.386 #define PCIXM_STATUS_BUS 0x0000FF00

Definition at line 438 of file pcireg.h.

7.14.1.387 #define PCIXM_STATUS_COMPLEX_DEV 0x00100000

Definition at line 443 of file pcireg.h.

7.14.1.388 #define PCIXM_STATUS_DEVFN 0x000000FF

Definition at line 437 of file pcireg.h.

7.14.1.389 #define PCIXM_STATUS_MAX_CUM_READ 0x1C000000

Definition at line 458 of file pcireg.h.

7.14.1.390 #define PCIXM_STATUS_MAX_READ 0x00600000

Definition at line 444 of file pcireg.h.

7.14.1.391 #define PCIXM_STATUS_MAX_READ_1024 0x00200000

Definition at line 446 of file pcireg.h.

7.14.1.392 #define PCIXM_STATUS_MAX_READ_2048 0x00400000

Definition at line 447 of file pcireg.h.

7.14.1.393 #define PCIXM_STATUS_MAX_READ_4096 0x00600000

Definition at line 448 of file pcireg.h.

7.14.1.394 #define PCIXM_STATUS_MAX_READ_512 0x00000000

Definition at line 445 of file pcireg.h.

7.14.1.395 #define PCIXM_STATUS_MAX_SPLITS 0x03800000

Definition at line 449 of file pcireg.h.

7.14.1.396 #define PCIXM_STATUS_MAX_SPLITS_1 0x00000000

Definition at line 450 of file pcireg.h.

7.14.1.397 #define PCIXM_STATUS_MAX_SPLITS_12 0x02800000

Definition at line 455 of file pcireg.h.

7.14.1.398 #define PCIXM_STATUS_MAX_SPLITS_16 0x03000000

Definition at line 456 of file pcireg.h.

7.14.1.399 #define PCIXM_STATUS_MAX_SPLITS_2 0x00800000

Definition at line 451 of file pcireg.h.

7.14.1.400 #define PCIXM_STATUS_MAX_SPLITS_3 0x01000000

Definition at line 452 of file pcireg.h.

7.14.1.401 #define PCIXM_STATUS_MAX_SPLITS_32 0x03800000

Definition at line 457 of file pcireg.h.

7.14.1.402 #define PCIXM_STATUS_MAX_SPLITS_4 0x01800000

Definition at line 453 of file pcireg.h.

7.14.1.403 #define PCIXM_STATUS_MAX_SPLITS_8 0x02000000

Definition at line 454 of file pcireg.h.

7.14.1.404 #define PCIXM_STATUS_RCVD_SC_ERR 0x20000000

Definition at line 459 of file pcireg.h.

7.14.1.405 #define PCIXM_STATUS_SC_DISCARDED 0x00040000

Definition at line 441 of file pcireg.h.

7.14.1.406 #define PCIXM_STATUS_UNEXP_SC 0x00080000

Definition at line 442 of file pcireg.h.

7.14.1.407 #define PCIXR_BRIDGE_STATUS 0x4

Definition at line 475 of file pcireg.h.

7.14.1.408 #define PCIXR_COMMAND 0x2

Definition at line 418 of file pcireg.h.

7.14.1.409 #define PCIXR_SEC_STATUS 0x2

Definition at line 464 of file pcireg.h.

7.14.1.410 #define PCIXR_STATUS 0x4

Definition at line 436 of file pcireg.h.

7.14.1.411 #define PCIY_AGP 0x02

Definition at line 99 of file pcireg.h.
Referenced by pci_hostb_attach().

7.14.1.412 #define PCIY_AGP8X 0x0e

Definition at line 111 of file pcireg.h.

7.14.1.413 #define PCIY_CHSWP 0x06

Definition at line 103 of file pcireg.h.

7.14.1.414 #define PCIY_CRES 0x0b

Definition at line 108 of file pcireg.h.

7.14.1.415 #define PCIY_DEBUG 0x0a

Definition at line 107 of file pcireg.h.

7.14.1.416 #define PCIY_EXPRESS 0x10

Definition at line 113 of file pcireg.h.

Referenced by pci_read_extcap().

7.14.1.417 #define PCIY_HOTPLUG 0x0c

Definition at line 109 of file pcireg.h.

7.14.1.418 #define PCIY_HT 0x08

Definition at line 105 of file pcireg.h.

Referenced by pci_read_extcap().

7.14.1.419 #define PCIY_MSI 0x05

Definition at line 102 of file pcireg.h.

Referenced by pci_read_extcap().

7.14.1.420 #define PCIY_MSIX 0x11

Definition at line 114 of file pcireg.h.

Referenced by pci_read_extcap().

7.14.1.421 #define PCIY_PCIX 0x07

Definition at line 104 of file pcireg.h.

Referenced by pci_read_extcap().

7.14.1.422 #define PCIY_PMG 0x01

Definition at line 98 of file pcireg.h.

Referenced by pci_read_extcap().

7.14.1.423 #define PCIY_SECDEV 0x0f

Definition at line 112 of file pcireg.h.

7.14.1.424 #define PCIY_SLOTID 0x04

Definition at line 101 of file pcireg.h.

7.14.1.425 #define PCIY_SUBVENDOR 0x0d

Definition at line 110 of file pcireg.h.

Referenced by pci_read_extcap().

7.14.1.426 #define PCIY_VENDOR 0x09

Definition at line 106 of file pcireg.h.

7.14.1.427 #define PCIY_VPD 0x03

Definition at line 100 of file pcireg.h.

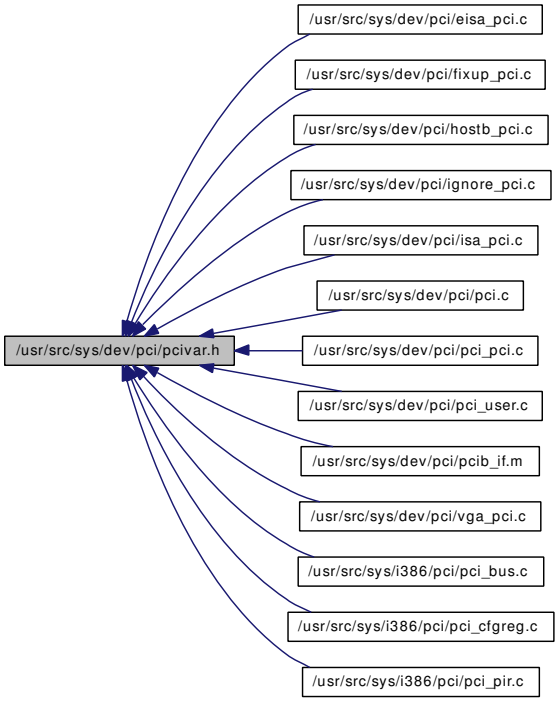
Referenced by pci_read_extcap().

7.15 /usr/src/sys/dev/pci/pcivar.h File Reference

```
#include <sys/queue.h>
```

Include dependency graph for pcivar.h:

This graph shows which files directly or indirectly include this file:



Data Structures

- struct pcicfg_pp
- struct vpd_readonly
- struct vpd_write
- struct pcicfg_vpd
- struct pcicfg_msi
- struct pcicfg_msix
- struct pcicfg
- struct pcih1cfgregs
- struct pcih2cfgregs

Defines

- #define PCI_BUSMAX 255
- #define PCI_SLOTMAX 31

- #define PCI_FUNCMAX 7
- #define PCI_REGMAX 255
- #define PCI_MAXMAPS_0 6
- #define PCI_MAXMAPS_1 2
- #define PCI_MAXMAPS_2 1
- #define PCI_PPBMEMBASE(h, l) ((((pci_addr_t)(h) << 32) + ((l)<<16)) & ~0xfffff)
- #define PCI_PPBMEMLIMIT(h, l) ((((pci_addr_t)(h) << 32) + ((l)<<16)) | 0xfffff)
- #define PCI_PPBIOBASE(h, l) ((((h)<<16) + ((l)<<8)) & ~0xfff)
- #define PCI_PPBIOLIMIT(h, l) ((((h)<<16) + ((l)<<8)) | 0xfff)

Typedefs

- typedef uint64_t pci_addr_t
- typedef pcicfg pcicfgregs

Functions

- STAILQ_HEAD (devlist, pci_devinfo)

Variables

- uint32_t pci_numdevs
- cdevsw pcicdev
- devlist pci_devq
- uint32_t pci_generation

7.15.1 Define Documentation

7.15.1.1 #define PCI_BUSMAX 255

Definition at line 37 of file pcivar.h.

7.15.1.2 #define PCI_FUNCMAX 7

Definition at line 39 of file pcivar.h.

7.15.1.3 #define PCI_MAXMAPS_0 6

Definition at line 42 of file pcivar.h.

Referenced by pci_hdrtypedata().

7.15.1.4 #define PCI_MAXMAPS_1 2

Definition at line 43 of file pcivar.h.

Referenced by pci_hdrtypedata().

7.15.1.5 #define PCI_MAXMAPS_2 1

Definition at line 44 of file pcivar.h.

Referenced by pci_hdrtypedata().

7.15.1.6 #define PCI_PPBIOBASE(h, l) ((((h)<<16) + ((l)<<8)) & ~0xfff)

Definition at line 147 of file pcivar.h.

Referenced by pcib_attach_common().

7.15.1.7 #define PCI_PPBIOLIMIT(h, l) ((((h)<<16) + ((l)<<8)) | 0xfff)

Definition at line 148 of file pcivar.h.

Referenced by pcib_attach_common().

7.15.1.8 #define PCI_PPBMEMBASE(h, l) ((((pci_addr_t)(h) << 32) + ((l)<<16)) & ~0xfffff)

Definition at line 145 of file pcivar.h.

Referenced by pcib_attach_common().

7.15.1.9 #define PCI_PPBMEMLIMIT(h, l) ((((pci_addr_t)(h) << 32) + ((l)<<16)) | 0xfffff)

Definition at line 146 of file pcivar.h.

Referenced by pcib_attach_common().

7.15.1.10 #define PCI_REGMAX 255

Definition at line 40 of file pcivar.h.

7.15.1.11 #define PCI_SLOTMAX 31

Definition at line 38 of file pcivar.h.

7.15.2 Typedef Documentation

7.15.2.1 typedef uint64_t pci_addr_t

Definition at line 46 of file pcivar.h.

7.15.2.2 typedef struct pcicfg pcicfgregs

7.15.3 Function Documentation

7.15.3.1 STAILQ_HEAD (devlist, pci_devinfo)

7.15.4 Variable Documentation

7.15.4.1 struct devlist pci_devq

Definition at line 217 of file pci.c.

Referenced by pci_find_bsf(), pci_find_device(), pci_freecfg(), pci_ioctl(), pci_modevent(), and pci_read_device().

7.15.4.2 uint32_t pci_generation

Definition at line 218 of file pci.c.

Referenced by pci_freecfg(), pci_ioctl(), pci_modevent(), and pci_read_device().

7.15.4.3 uint32_t pci_numdevs

Definition at line 219 of file pci.c.

Referenced by pci_freecfg(), pci_ioctl(), and pci_read_device().

7.15.4.4 struct cdevsw pcicdev

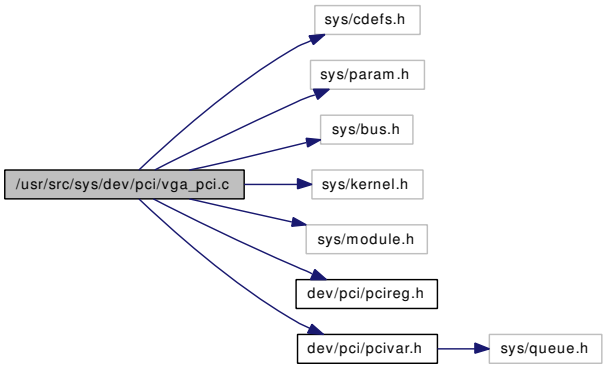
Definition at line 71 of file pci_user.c.

Referenced by pci_modevent().

7.16 /usr/src/sys/dev/pci/vga_pci.c File Reference

```
#include <sys/cdefs.h>
#include <sys/param.h>
#include <sys/bus.h>
#include <sys/kernel.h>
#include <sys/module.h>
#include <dev/pci/pcireg.h>
#include <dev/pci/pcivar.h>
```

Include dependency graph for vga_pci.c:



Functions

- __FBSDID ("$FreeBSD: src/sys/dev/pci/vga_pci.c,v 1.5 2006/02/01 15:45:29 jhb Exp $")
- static int vga_pci_probe (device_t dev)
- static int vga_pci_attach (device_t dev)
- static int vga_pci_suspend (device_t dev)
- static int vga_pci_resume (device_t dev)
- static int vga_pci_read_ivar (device_t dev, device_t child, int which, uintptr_t *result)
- static int vga_pci_write_ivar (device_t dev, device_t child, int which, uintptr_t value)
- static struct resource * vga_pci_alloc_resource (device_t dev, device_t child, int type, int *rid, u_long start, u_long end, u_long count, u_int flags)
- static int vga_pci_release_resource (device_t dev, device_t child, int type, int rid, struct resource *r)
- static uint32_t vga_pci_read_config (device_t dev, device_t child, int reg, int width)
- static void vga_pci_write_config (device_t dev, device_t child, int reg, uint32_t val, int width)
- static int vga_pci_enable_busmaster (device_t dev, device_t child)
- static int vga_pci_disable_busmaster (device_t dev, device_t child)
- static int vga_pci_enable_io (device_t dev, device_t child, int space)
- static int vga_pci_disable_io (device_t dev, device_t child, int space)
- static int vga_pci_set_powerstate (device_t dev, device_t child, int state)
- static int vga_pci_get_powerstate (device_t dev, device_t child)
- static int vga_pci_assign_interrupt (device_t dev, device_t child)
- static int vga_pci_find_extcap (device_t dev, device_t child, int capability, int *capreg)
- DRIVER_MODULE (vgapci, pci, vga_pci_driver, vga_devclass, 0, 0)

Variables

- static device_method_t vga_pci_methods [ ]
- static driver_t vga_pci_driver
- static devclass_t vga_devclass

7.16.1 Function Documentation

7.16.1.1 __FBSDID ("$FreeBSD: src/sys/dev/pci/vga_pci. *c*, v 1.5 2006/02/01 15:45:29 jhb Exp $")

7.16.1.2 DRIVER_MODULE (vgapci, pci, vga_pci_driver, vga_devclass, 0, 0)

7.16.1.3 static struct resource* vga_pci_alloc_resource (device_t *dev*, device_t *child*, int *type*, int * *rid*, u_long *start*, u_long *end*, u_long *count*, u_int *flags*) [static]

Definition at line 110 of file vga_pci.c.

7.16.1.4 static int vga_pci_assign_interrupt (device_t *dev*, device_t *child*) [static]

Definition at line 197 of file vga_pci.c.

7.16.1.5 static int vga_pci_attach (device_t *dev*) [static]

Definition at line 68 of file vga_pci.c.

7.16.1.6 static int vga_pci_disable_busmaster (device_t *dev*, device_t *child*) [static]

Definition at line 152 of file vga_pci.c.

7.16.1.7 static int vga_pci_disable_io (device_t *dev*, device_t *child*, int *space*) [static]

Definition at line 170 of file vga_pci.c.

7.16.1.8 static int vga_pci_enable_busmaster (device_t *dev*, device_t *child*) [static]

Definition at line 143 of file vga_pci.c.

7.16.1.9 static int vga_pci_enable_io (device_t *dev*, device_t *child*, int *space*) [static]

Definition at line 161 of file vga_pci.c.

7.16.1.10 static int vga_pci_find_extcap (device_t *dev*, device_t *child*, int *capability*, int * *capreg*) [static]

Definition at line 206 of file vga_pci.c.

7.16.1.11 static int vga_pci_get_powerstate (device_t *dev*, device_t *child*) [static]

Definition at line 188 of file vga_pci.c.

7.16.1.12 static int vga_pci_probe (device_t *dev*) [static]

Definition at line 50 of file vga_pci.c.

References PCIC_DISPLAY, PCIC_OLD, and PCIS_OLD_VGA.

7.16.1.13 static uint32_t vga_pci_read_config (device_t *dev*, device_t *child*, int *reg*, int *width*) [static]

Definition at line 128 of file vga_pci.c.

7.16.1.14 static int vga_pci_read_ivar (device_t *dev*, device_t *child*, int *which*, uintptr_t * *result*) [static]

Definition at line 96 of file vga_pci.c.

7.16.1.15 static int vga_pci_release_resource (device_t *dev*, device_t *child*, int *type*, int *rid*, struct resource * *r*) [static]

Definition at line 118 of file vga_pci.c.

7.16.1.16 static int vga_pci_resume (device_t *dev*) [static]

Definition at line 87 of file vga_pci.c.

7.16.1.17 static int vga_pci_set_powerstate (device_t *dev*, device_t *child*, int *state*) [static]

Definition at line 179 of file vga_pci.c.

7.16.1.18 static int vga_pci_suspend (device_t *dev*) [static]

Definition at line 80 of file vga_pci.c.

7.16.1.19 static void vga_pci_write_config (device_t *dev*, device_t *child*, int *reg*, uint32_t *val*, int *width*) [static]

Definition at line 135 of file vga_pci.c.

7.16.1.20 static int vga_pci_write_ivar (device_t *dev*, device_t *child*, int *which*, uintptr_t *value*) [static]

Definition at line 103 of file vga_pci.c.

7.16.2 Variable Documentation

7.16.2.1 devclass_t vga_devclass [static]

Definition at line 253 of file vga_pci.c.

7.16.2.2 driver_t vga_pci_driver [static]

Initial value:

```
{
        "vgapci",
        vga_pci_methods,
        1,
}
```

Definition at line 247 of file vga_pci.c.

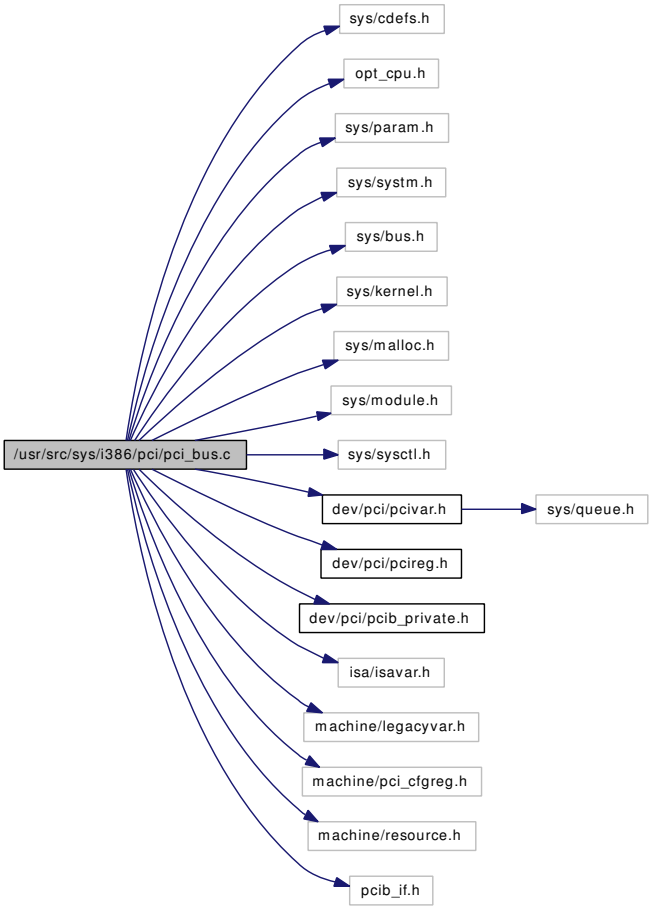
7.16.2.3 device_method_t vga_pci_methods[ ] [static]

Definition at line 213 of file vga_pci.c.

7.17 /usr/src/sys/i386/pci/pci_bus.c File Reference

```
#include <sys/cdefs.h>
#include "opt_cpu.h"
#include <sys/param.h>
#include <sys/systm.h>
#include <sys/bus.h>
#include <sys/kernel.h>
#include <sys/malloc.h>
#include <sys/module.h>
#include <sys/sysctl.h>
#include <dev/pci/pcivar.h>
#include <dev/pci/pcireg.h>
#include <dev/pci/pcib_private.h>
#include <isa/isavar.h>
#include <machine/legacyvar.h>
#include <machine/pci_cfgreg.h>
#include <machine/resource.h>
#include "pcib_if.h"
```

Include dependency graph for pci_bus.c:



Functions

- __FBSDID ("$FreeBSD: src/sys/i386/pci/pci_bus.c,v 1.126 2007/01/22 21:48:44 jhb Exp $")
- static int pcibios_pcib_route_interrupt (device_t pcib, device_t dev, int pin)
- int legacy_pcib_maxslots (device_t dev)
- u_int32_t legacy_pcib_read_config (device_t dev, int bus, int slot, int func, int reg, int bytes)
- void legacy_pcib_write_config (device_t dev, int bus, int slot, int func, int reg, u_int32_t data, int bytes)
- static int legacy_pcib_alloc_msi (device_t pcib, device_t dev, int count, int maxcount, int *irqs)
- static int legacy_pcib_alloc_msix (device_t pcib, device_t dev, int index, int *irq)
- static const char * legacy_pcib_is_host_bridge (int bus, int slot, int func, uint32_t id, uint8_t class, uint8_t subclass, uint8_t *busnum)
- static void legacy_pcib_identify (driver_t *driver, device_t parent)
- static int legacy_pcib_probe (device_t dev)
- static int legacy_pcib_attach (device_t dev)
- int legacy_pcib_read_ivar (device_t dev, device_t child, int which, uintptr_t *result)
- int legacy_pcib_write_ivar (device_t dev, device_t child, int which, uintptr_t value)
- SYSCTL_DECL (_hw_pci)
- TUNABLE_ULONG ("hw.pci.host_mem_start",&legacy_host_mem_start)
- SYSCTL_ULONG (_hw_pci, OID_AUTO, host_mem_start, CTLFLAG_RDTUN,&legacy_host_mem_start, 0x80000000,"Limit the host bridge memory to being above this address. Must be\n\ set at boot via a tunable.")

- resource * legacy_pcib_alloc_resource (device_t dev, device_t child, int type, int *rid, u_long start, u_long end, u_long count, u_int flags)
- DEFINE_CLASS_0 (pcib, legacy_pcib_driver, legacy_pcib_methods, 1)
- DRIVER_MODULE (pcib, legacy, legacy_pcib_driver, hostb_devclass, 0, 0)
- static int pcibus_pnp_probe (device_t dev)
- static int pcibus_pnp_attach (device_t dev)
- DEFINE_CLASS_0 (pcibus_pnp, pcibus_pnp_driver, pcibus_pnp_methods, 1)
- DRIVER_MODULE (pcibus_pnp, isa, pcibus_pnp_driver, pcibus_pnp_devclass, 0, 0)
- static int pcibios_pcib_probe (device_t bus)
- DEFINE_CLASS_0 (pcib, pcibios_pcib_driver, pcibios_pcib_pci_methods, sizeof(struct pcib_softc))
- DRIVER_MODULE (pcibios_pcib, pci, pcibios_pcib_driver, pcib_devclass, 0, 0)

Variables

- static unsigned long legacy_host_mem_start = 0x80000000
- static device_method_t legacy_pcib_methods [ ]
- static devclass_t hostb_devclass
- static struct isa_pnp_id pcibus_pnp_ids [ ]
- static device_method_t pcibus_pnp_methods [ ]
- static devclass_t pcibus_pnp_devclass
- static device_method_t pcibios_pcib_pci_methods [ ]
- static devclass_t pcib_devclass

7.17.1 Function Documentation

7.17.1.1 __FBSDID ("$FreeBSD: src/sys/i386/pci/pci_bus. *c*, v 1.126 2007/01/22 21:48:44 jhb Exp $")

7.17.1.2 DEFINE_CLASS_0 (pcib, pcibios_pcib_driver, pcibios_pcib_pci_methods, sizeof(struct pcib_softc))

7.17.1.3 DEFINE_CLASS_0 (pcibus_pnp, pcibus_pnp_driver, pcibus_pnp_methods, 1)

7.17.1.4 DEFINE_CLASS_0 (pcib, legacy_pcib_driver, legacy_pcib_methods, 1)

7.17.1.5 DRIVER_MODULE (pcibios_pcib, pci, pcibios_pcib_driver, pcib_devclass, 0, 0)

7.17.1.6 DRIVER_MODULE (pcibus_pnp, isa, pcibus_pnp_driver, pcibus_pnp_devclass, 0, 0)

7.17.1.7 DRIVER_MODULE (pcib, legacy, legacy_pcib_driver, hostb_devclass, 0, 0)

7.17.1.8 static int legacy_pcib_alloc_msi (device_t *pcib*, device_t *dev*, int *count*, int *maxcount*, int * *irqs*) [static]

Definition at line 83 of file pci_bus.c.

7.17.1.9 static int legacy_pcib_alloc_msix (device_t *pcib*, device_t *dev*, int *index*, int * *irq*) [static]

Definition at line 94 of file pci_bus.c.

7.17.1.10 struct resource* legacy_pcib_alloc_resource (device_t *dev*, device_t *child*, int *type*, int * *rid*, u_long *start*, u_long *end*, u_long *count*, u_int *flags*)

Definition at line 511 of file pci_bus.c.

References legacy_host_mem_start.

7.17.1.11 static int legacy_pcib_attach (device_t *dev*) [static]

Definition at line 456 of file pci_bus.c.

References pci_pir_probe().

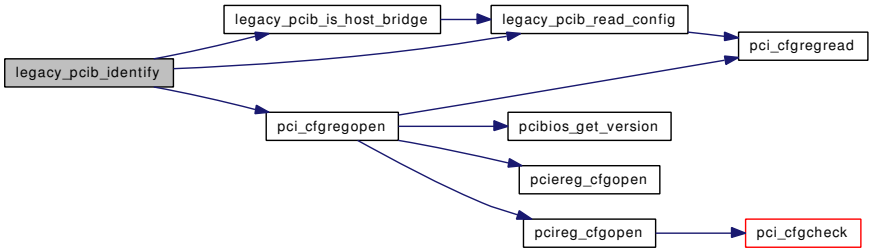
Here is the call graph for this function:

7.17.1.12 static void legacy_pcib_identify (driver_t * *driver*, device_t *parent*) [static]

Definition at line 326 of file pci_bus.c.

References legacy_pcib_is_host_bridge(), legacy_pcib_read_config(), pci_cfgregopen(), pci_devclass, PCI_FUNCMAX, PCI_MAXHDRTYPE, PCI_SLOTMAX, PCIM_HDRTYPE, PCIM_MFDEV, PCIR_-DEVVENDOR, PCIR_HDRTYPE, PCIR_SUBCLASS, and subclass.

Here is the call graph for this function:



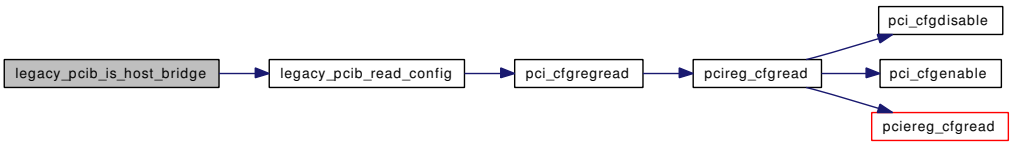
7.17.1.13 static const char* legacy_pcib_is_host_bridge (int *bus*, int *slot*, int *func*, uint32_t *id*, uint8_t *class*, uint8_t *subclass*, uint8_t * *busnum*) [static]

Definition at line 103 of file pci_bus.c.

References legacy_pcib_read_config(), PCIC_BRIDGE, and PCIS_BRIDGE_HOST.

Referenced by legacy_pcib_identify().

Here is the call graph for this function:



7.17.1.14 int legacy_pcib_maxslots (device_t *dev*)

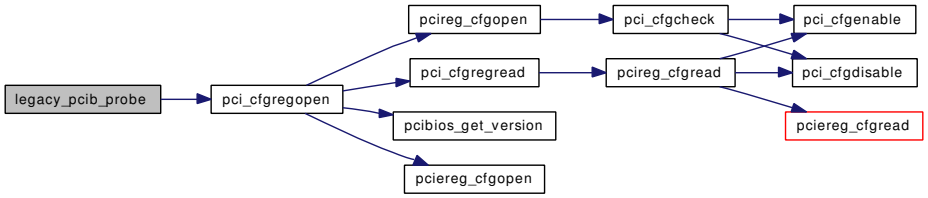
Definition at line 57 of file pci_bus.c.

7.17.1.15 static int legacy_pcib_probe (device_t *dev*) [static]

Definition at line 447 of file pci_bus.c.

References pci_cfgregopen().

Here is the call graph for this function:



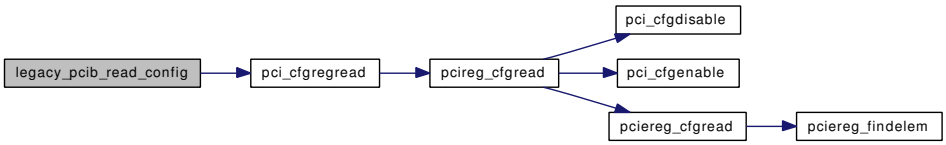
7.17.1.16 u_int32_t legacy_pcib_read_config (device_t *dev*, int *bus*, int *slot*, int *func*, int *reg*, int *bytes*)

Definition at line 65 of file pci_bus.c.

References pci_cfgregread().

Referenced by legacy_pcib_identify(), and legacy_pcib_is_host_bridge().

Here is the call graph for this function:



7.17.1.17 int legacy_pcib_read_ivar (device_t *dev*, device_t *child*, int *which*, uintptr_t * *result*)

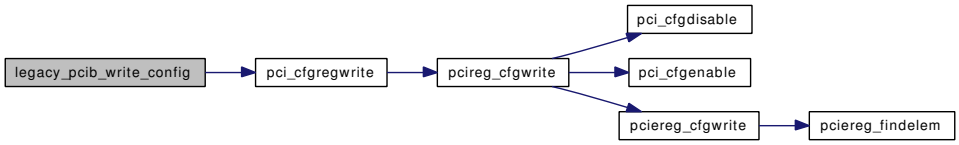
Definition at line 476 of file pci_bus.c.

7.17.1.18 void legacy_pcib_write_config (device_t *dev*, int *bus*, int *slot*, int *func*, int *reg*, u_int32_t *data*, int *bytes*)

Definition at line 74 of file pci_bus.c.

References pci_cfgregwrite().

Here is the call graph for this function:



7.17.1.19 int legacy_pcib_write_ivar (device_t *dev*, device_t *child*, int *which*, uintptr_t *value*)

Definition at line 489 of file pci_bus.c.

7.17.1.20 static int pcibios_pcib_probe (device_t *bus*) [static]

Definition at line 669 of file pci_bus.c.

References pci_pir_probe(), PCIC_BRIDGE, PCIR_SECBUS_1, and PCIS_BRIDGE_PCI.

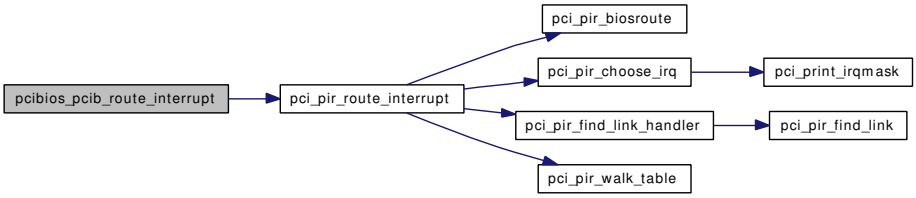
Here is the call graph for this function:

7.17.1.21 static int pcibios_pcib_route_interrupt (device_t *pcib*, device_t *dev*, int *pin*) [static]

Definition at line 686 of file pci_bus.c.

References pci_pir_route_interrupt().

Here is the call graph for this function:



7.17.1.22 static int pcibus_pnp_attach (device_t *dev*) [static]

Definition at line 600 of file pci_bus.c.

7.17.1.23 static int pcibus_pnp_probe (device_t *dev*) [static]

Definition at line 590 of file pci_bus.c.

References pcibus_pnp_ids.

7.17.1.24 SYSCTL_DECL (_hw_pci)

7.17.1.25 SYSCTL_ULONG (_hw_pci, OID_AUTO, host_mem_start, CTLFLAG_RDTUN, & *legacy_host_mem_start*, 0x80000000, "Limit the host bridge memory to being above this address. Must be\n\set at boot via a tunable.")

7.17.1.26 TUNABLE_ULONG ("hw.pci.host_mem_start", & *legacy_host_mem_start*)

7.17.2 Variable Documentation

7.17.2.1 devclass_t hostb_devclass [static]

Definition at line 568 of file pci_bus.c.

7.17.2.2 unsigned long legacy_host_mem_start = 0x80000000 [static]

Definition at line 503 of file pci_bus.c.

Referenced by legacy_pcib_alloc_resource().

7.17.2.3 device_method_t legacy_pcib_methods[ ] [static]

Definition at line 534 of file pci_bus.c.

7.17.2.4 devclass_t pcib_devclass [static]

Definition at line 662 of file pci_bus.c.

7.17.2.5 device_method_t pcibios_pcib_pci_methods[ ] [static]

Definition at line 629 of file pci_bus.c.

7.17.2.6 devclass_t pcibus_pnp_devclass [static]

Definition at line 616 of file pci_bus.c.

7.17.2.7 struct isa_pnp_id pcibus_pnp_ids[ ] [static]

Initial value:

```
 {
        { 0x030ad041 , "PCI Bus" },
        { 0 }
}
```

Definition at line 584 of file pci_bus.c.

Referenced by pcibus_pnp_probe().

7.17.2.8 device_method_t pcibus_pnp_methods[ ] [static]

Initial value:

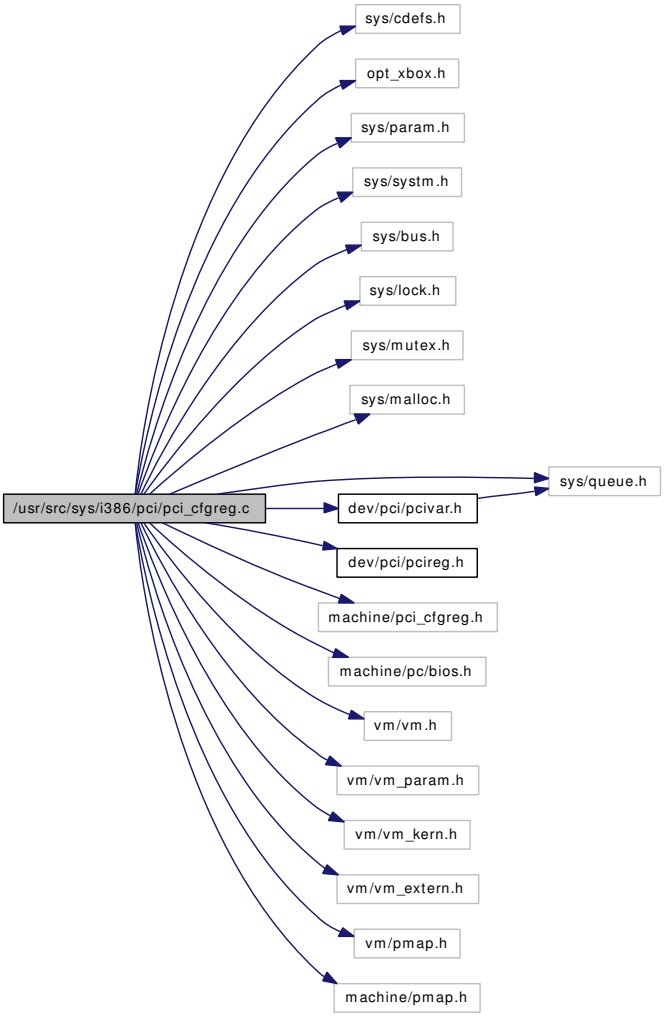
```
{

        DEVMETHOD(device_probe,         pcibus_pnp_probe),
        DEVMETHOD(device_attach,        pcibus_pnp_attach),
        DEVMETHOD(device_detach,        bus_generic_detach),
        DEVMETHOD(device_shutdown,      bus_generic_shutdown),
        DEVMETHOD(device_suspend,       bus_generic_suspend),
        DEVMETHOD(device_resume,        bus_generic_resume),
        { 0, 0 }
}
```

Definition at line 605 of file pci_bus.c.

7.18 /usr/src/sys/i386/pci/pci_cfgreg.c File Reference

```
#include <sys/cdefs.h>
#include "opt_xbox.h"
#include <sys/param.h>
#include <sys/systm.h>
#include <sys/bus.h>
#include <sys/lock.h>
#include <sys/mutex.h>
#include <sys/malloc.h>
#include <sys/queue.h>
#include <dev/pci/pcivar.h>
#include <dev/pci/pcireg.h>
#include <machine/pci_cfgreg.h>
#include <machine/pc/bios.h>
#include <vm/vm.h>
#include <vm/vm_param.h>
#include <vm/vm_kern.h>
#include <vm/vm_extern.h>
#include <vm/pmap.h>
#include <machine/pmap.h>
```

Include dependency graph for pci_cfgreg.c:



Data Structures

- struct pcie_cfg_elem

Defines

- #define PRVERB(a)
- #define PCIE_CACHE 8
- #define PCIE_PADDR(bar, reg, bus, slot, func)

Enumerations

- enum { CFGMECH_NONE = 0, CFGMECH_1, CFGMECH_2, CFGMECH_PCIE }

Functions

- __FBSDID ("$FreeBSD: src/sys/i386/pci/pci_cfgreg.c,v 1.124 2006/12/12 19:23:52 jhb Exp $")

- static TAILQ_HEAD (pcie_cfg_list, pcie_cfg_elem)
- static u_int16_t pcibios_get_version (void)
- int pci_cfgregopen (void)
- u_int32_t pci_cfgregread (int bus, int slot, int func, int reg, int bytes)
- void pci_cfgregwrite (int bus, int slot, int func, int reg, u_int32_t data, int bytes)
- static int pci_cfgenable (unsigned bus, unsigned slot, unsigned func, int reg, int bytes)
- static void pci_cfgdisable (void)
- static int pcireg_cfgread (int bus, int slot, int func, int reg, int bytes)
- static void pcireg_cfgwrite (int bus, int slot, int func, int reg, int data, int bytes)
- static int pci_cfgcheck (int maxdev)
- static int pcireg_cfgopen (void)
- static int pciereg_cfgopen (void)
- static __inline struct pcie_cfg_elem * pciereg_findelem (vm_paddr_t papage)
- static int pciereg_cfgread (int bus, int slot, int func, int reg, int bytes)
- static void pciereg_cfgwrite (int bus, int slot, int func, int reg, int data, int bytes)

7.18.1 Define Documentation

7.18.1.1 #define PCIE_CACHE 8

Definition at line 63 of file pci_cfgreg.c.

Referenced by pciereg_cfgopen().

7.18.1.2 #define PCIE_PADDR(bar, reg, bus, slot, func)

Value:

```
((bar)                          |       \
        (((bus) & 0xff) << 20)          |       \
        (((slot) & 0x1f) << 15)         |       \
        (((func) & 0x7) << 12)          |       \
        ((reg) & 0xfff))
```

Definition at line 530 of file pci_cfgreg.c.

Referenced by pciereg_cfgread(), and pciereg_cfgwrite().

7.18.1.3 #define PRVERB(a)

Value:

```
do {                                                    \
        if (bootverbose)                                        \
                printf a ;                                      \
} while(0)
```

Definition at line 58 of file pci_cfgreg.c.

Referenced by pci_cfgregopen(), and pcibios_get_version().

7.18.2 Enumeration Type Documentation

7.18.2.1 anonymous enum

Enumerator:

CFGMECH_NONE
CFGMECH_1
CFGMECH_2
CFGMECH_PCIE

Definition at line 70 of file pci_cfgreg.c.

7.18.3 Function Documentation

7.18.3.1 __FBSDID ("$FreeBSD: src/sys/i386/pci/pci_cfgreg. *c*, v 1.124 2006/12/12 19:23:52 jhb Exp $")

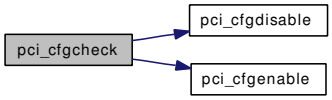
7.18.3.2 static int pci_cfgcheck (int *maxdev*) [static]

Definition at line 354 of file pci_cfgreg.c.

References pci_cfgdisable(), and pci_cfgenable().

Referenced by pcireg_cfgopen().

Here is the call graph for this function:



7.18.3.3 static void pci_cfgdisable (void) [static]

Definition at line 279 of file pci_cfgreg.c.

References CFGMECH_1, and CFGMECH_2.

Referenced by pci_cfgcheck(), pcireg_cfgread(), and pcireg_cfgwrite().

7.18.3.4 static int pci_cfgenable (unsigned *bus*, unsigned *slot*, unsigned *func*, int *reg*, int *bytes*) [static]

Definition at line 216 of file pci_cfgreg.c.

References CFGMECH_1, CFGMECH_2, PCI_BUSMAX, PCI_FUNCMAX, and PCI_REGMAX.

Referenced by pci_cfgcheck(), pcireg_cfgread(), and pcireg_cfgwrite().

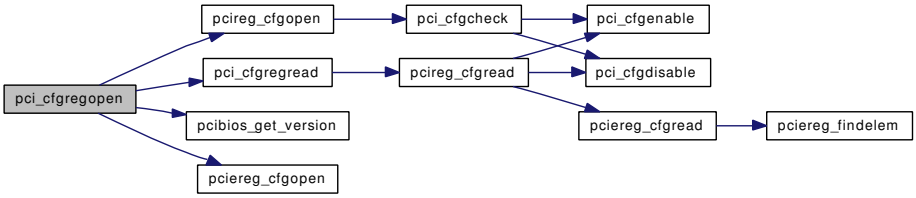
7.18.3.5 int pci_cfgregopen (void)

Definition at line 133 of file pci_cfgreg.c.

References pci_cfgregread(), pcibios_get_version(), pciereg_cfgopen(), PCIR_DEVICE, PCIR_-VENDOR, pcireg_cfgopen(), and PRVERB.

Referenced by legacy_pcib_identify(), and legacy_pcib_probe().

Here is the call graph for this function:



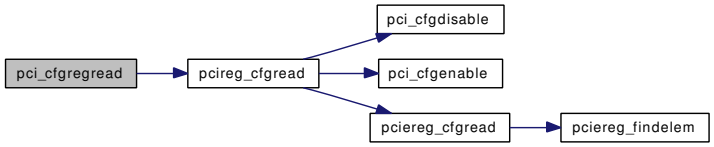
7.18.3.6 u_int32_t pci_cfgregread (int *bus*, int *slot*, int *func*, int *reg*, int *bytes*)

Definition at line 184 of file pci_cfgreg.c.

References PCIR_INTLINE, and pcireg_cfgread().

Referenced by legacy_pcib_read_config(), pci_cfgregopen(), and pci_pir_search_irq().

Here is the call graph for this function:



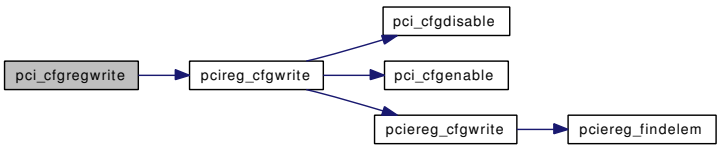
7.18.3.7 void pci_cfgregwrite (int *bus*, int *slot*, int *func*, int *reg*, u_int32_t *data*, int *bytes*)

Definition at line 204 of file pci_cfgreg.c.

References pcireg_cfgwrite().

Referenced by legacy_pcib_write_config().

Here is the call graph for this function:



7.18.3.8 static u_int16_t pcibios_get_version (void) [static]

Definition at line 109 of file pci_cfgreg.c.

References PRVERB.

Referenced by pci_cfgregopen().

7.18.3.9 static int pciereg_cfgopen (void) [static]

Definition at line 481 of file pci_cfgreg.c.

References CFGMECH_PCIE, and PCIE_CACHE.

Referenced by pci_cfgregopen().

7.18.3.10 static int pciereg_cfgread (int *bus*, int *slot*, int *func*, int *reg*, int *bytes*) [static]

Definition at line 573 of file pci_cfgreg.c.

References PCIE_PADDR, and pciereg_findelem().

Referenced by pcireg_cfgread().

Here is the call graph for this function:

7.18.3.11 static void pciereg_cfgwrite (int *bus*, int *slot*, int *func*, int *reg*, int *data*, int *bytes*) [static]

Definition at line 605 of file pci_cfgreg.c.

References PCIE_PADDR, and pciereg_findelem().

Referenced by pcireg_cfgwrite().

Here is the call graph for this function:

7.18.3.12 static __inline struct pcie_cfg_elem* pciereg_findelem (vm_paddr_t *papage*) [static]

Definition at line 544 of file pci_cfgreg.c.

Referenced by pciereg_cfgread(), and pciereg_cfgwrite().

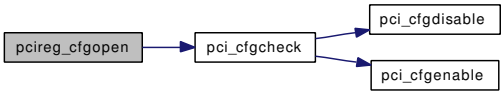
7.18.3.13 static int pcireg_cfgopen (void) [static]

Definition at line 401 of file pci_cfgreg.c.

References CFGMECH_1, CFGMECH_2, CFGMECH_NONE, and pci_cfgcheck().

Referenced by pci_cfgregopen().

Here is the call graph for this function:



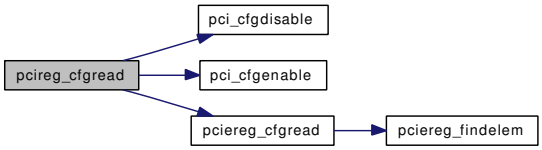
7.18.3.14 static int pcireg_cfgread (int *bus*, int *slot*, int *func*, int *reg*, int *bytes*) [static]

Definition at line 293 of file pci_cfgreg.c.

References CFGMECH_PCIE, pci_cfgdisable(), pci_cfgenable(), and pciereg_cfgread().

Referenced by pci_cfgregread().

Here is the call graph for this function:



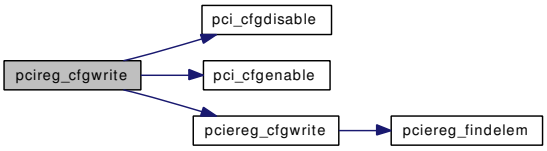
7.18.3.15 static void pcireg_cfgwrite (int *bus*, int *slot*, int *func*, int *reg*, int *data*, int *bytes*) [static]

Definition at line 324 of file pci_cfgreg.c.

References CFGMECH_PCIE, pci_cfgdisable(), pci_cfgenable(), and pciereg_cfgwrite().

Referenced by pci_cfgregwrite().

Here is the call graph for this function:



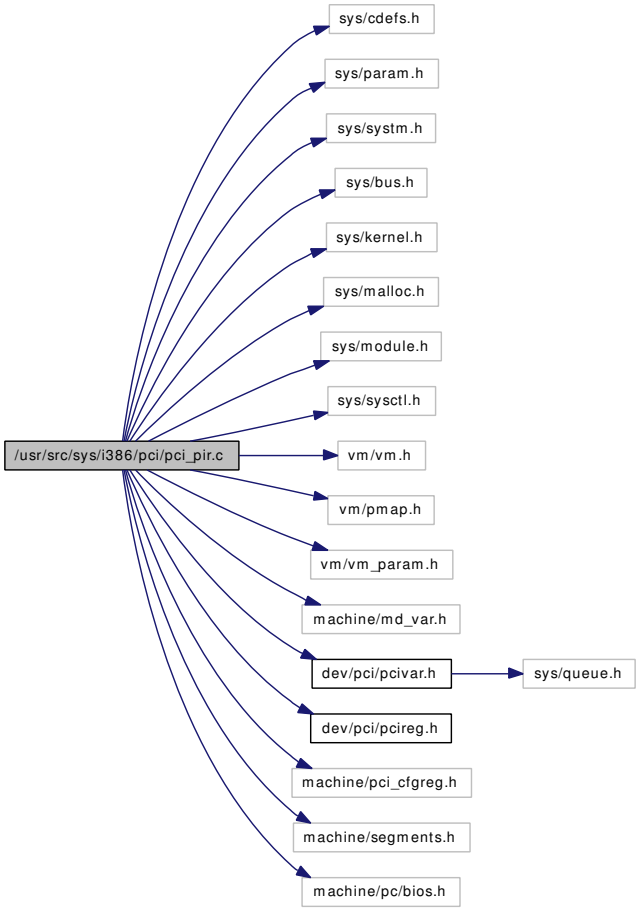
7.18.3.16 static TAILQ_HEAD (pcie_cfg_list, pcie_cfg_elem) [static]

Definition at line 77 of file pci_cfgreg.c.

7.19 /usr/src/sys/i386/pci/pci_pir.c File Reference

```
#include <sys/cdefs.h>
#include <sys/param.h>
#include <sys/systm.h>
#include <sys/bus.h>
#include <sys/kernel.h>
#include <sys/malloc.h>
#include <sys/module.h>
#include <sys/sysctl.h>
#include <vm/vm.h>
#include <vm/pmap.h>
#include <vm/vm_param.h>
#include <machine/md_var.h>
#include <dev/pci/pcivar.h>
#include <dev/pci/pcireg.h>
#include <machine/pci_cfgreg.h>
#include <machine/segments.h>
#include <machine/pc/bios.h>
```

Include dependency graph for pci_pir.c:



Data Structures

- struct pci_link
- struct pci_link_lookup
- struct pci_dev_lookup

Defines

- #define NUM_ISA_INTERRUPTS 16
- #define PCI_IRQ_OVERRIDE_MASK 0xdef8

Typedefs

- typedef void pir_entry_handler (struct PIR_entry *entry, struct PIR_intpin *intpin, void *arg)

Functions

- __FBSDID ("$FreeBSD: src/sys/i386/pci/pci_pir.c,v 1.120 2006/11/09 18:03:36 jhb Exp $")
- static void pci_print_irqmask (u_int16_t irqs)
- static int pci_pir_biosroute (int bus, int device, int func, int pin, int irq)

- static int pci_pir_choose_irq (struct pci_link *pci_link, int irqmask)
- static void pci_pir_create_links (struct PIR_entry *entry, struct PIR_intpin *intpin, void *arg)
- static void pci_pir_dump_links (void)
- static struct pci_link * pci_pir_find_link (uint8_t link_id)
- static void pci_pir_find_link_handler (struct PIR_entry *entry, struct PIR_intpin *intpin, void *arg)
- static void pci_pir_initial_irqs (struct PIR_entry *entry, struct PIR_intpin *intpin, void *arg)
- static void pci_pir_parse (void)
- static uint8_t pci_pir_search_irq (int bus, int device, int pin)
- static int pci_pir_valid_irq (struct pci_link *pci_link, int irq)
- static void pci_pir_walk_table (pir_entry_handler *handler, void *arg)
- static MALLOC_DEFINE (M_PIR,"$PIR","$PIR structures")
- static TAILQ_HEAD (pci_link)
- int pci_pir_route_interrupt (int bus, int device, int func, int pin)
- int pci_pir_probe (int bus, int require_parse)
- static int pir_probe (device_t dev)
- static int pir_attach (device_t dev)
- static void pir_resume_find_device (struct PIR_entry *entry, struct PIR_intpin *intpin, void *arg)
- static int pir_resume (device_t dev)
- DRIVER_MODULE (pir, legacy, pir_driver, pir_devclass, 0, 0)

Variables

- static struct PIR_table * pci_route_table
- static device_t pir_device
- static int pci_route_count
- static int pir_bios_irqs
- static int pir_parsed
- static device_method_t pir_methods [ ]
- static driver_t pir_driver
- static devclass_t pir_devclass

7.19.1 Define Documentation

7.19.1.1 #define NUM_ISA_INTERRUPTS 16

Definition at line 50 of file pci_pir.c.

7.19.1.2 #define PCI_IRQ_OVERRIDE_MASK 0xdef8

Referenced by TAILQ_HEAD().

7.19.2 Typedef Documentation

7.19.2.1 typedef void pir_entry_handler(struct PIR_entry *entry, struct PIR_intpin *intpin, void *arg)

Definition at line 79 of file pci_pir.c.

7.19.3 Function Documentation

7.19.3.1 __FBSDID ("$FreeBSD: src/sys/i386/pci/pci_pir. *c*, v 1.120 2006/11/09 18:03:36 jhb Exp $")

7.19.3.2 DRIVER_MODULE (pir, legacy, pir_driver, pir_devclass, 0, 0)

7.19.3.3 static MALLOC_DEFINE (M_PIR, "$PIR", "$PIR structures") [static]

7.19.3.4 static int pci_pir_biosroute (int *bus*, int *device*, int *func*, int *pin*, int *irq*) [static]

Definition at line 481 of file pci_pir.c.

Referenced by pci_pir_route_interrupt(), and pir_resume().

7.19.3.5 static int pci_pir_choose_irq (struct pci_link * *pci_link*, int *irqmask*) [static]

Definition at line 571 of file pci_pir.c.

References pci_print_irqmask().

Referenced by pci_pir_route_interrupt().

Here is the call graph for this function:

7.19.3.6 static void pci_pir_create_links (struct PIR_entry * *entry*, struct PIR_intpin * *intpin*, void * *arg*) [static]

Definition at line 235 of file pci_pir.c.

References pci_pir_find_link().

Referenced by pci_pir_parse().

Here is the call graph for this function:

7.19.3.7 static void pci_pir_dump_links (void) [static]

Definition at line 622 of file pci_pir.c.

References pci_print_irqmask().

Referenced by pci_pir_parse().

Here is the call graph for this function:

7.19.3.8 static struct pci_link * pci_pir_find_link (uint8_t *link_id*) [static]

Definition at line 174 of file pci_pir.c.

Referenced by pci_pir_create_links(), pci_pir_find_link_handler(), and pci_pir_initial_irqs().

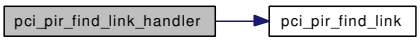
7.19.3.9 static void pci_pir_find_link_handler (struct PIR_entry * *entry*, struct PIR_intpin * *intpin*, void * *arg*) [static]

Definition at line 189 of file pci_pir.c.

References pci_link_lookup::bus, pci_link_lookup::device, pci_link_lookup::pci_link_ptr, pci_pir_find_link(), and pci_link_lookup::pin.

Referenced by pci_pir_route_interrupt().

Here is the call graph for this function:



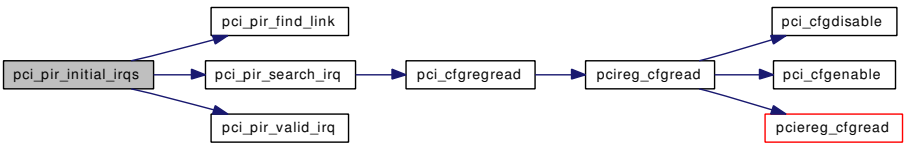
7.19.3.10 static void pci_pir_initial_irqs (struct PIR_entry * *entry*, struct PIR_intpin * *intpin*, void * *arg*) [static]

Definition at line 311 of file pci_pir.c.

References pci_pir_find_link(), pci_pir_search_irq(), and pci_pir_valid_irq().

Referenced by pci_pir_parse().

Here is the call graph for this function:



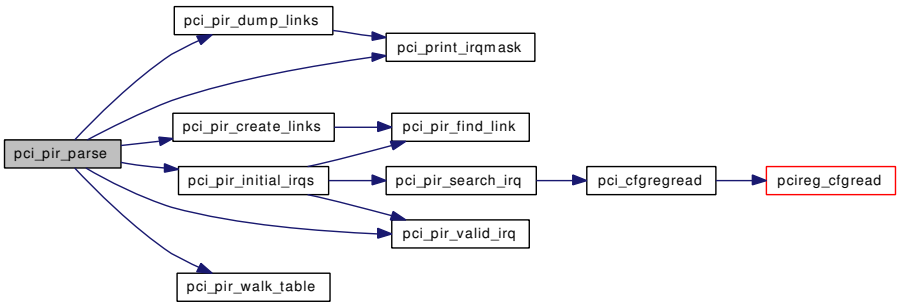
7.19.3.11 static void pci_pir_parse (void) [static]

Definition at line 382 of file pci_pir.c.

References pci_pir_create_links(), pci_pir_dump_links(), pci_pir_initial_irqs(), pci_pir_valid_irq(), pci_pir_walk_table(), pci_print_irqmask(), pir_bios_irqs, and pir_parsed.

Referenced by pir_attach().

Here is the call graph for this function:



7.19.3.12 int pci_pir_probe (int *bus*, int *require_parse*)

Definition at line 643 of file pci_pir.c.

References pci_route_count, pci_route_table, and pir_parsed.

Referenced by legacy_pcib_attach(), and pcibios_pcib_probe().

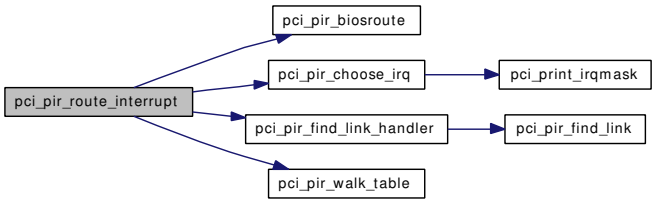
7.19.3.13 int pci_pir_route_interrupt (int *bus*, int *device*, int *func*, int *pin*)

Definition at line 496 of file pci_pir.c.

References pci_link_lookup::bus, pci_link_lookup::device, pci_link_lookup::pci_link_ptr, pci_pir_biosroute(), pci_pir_choose_irq(), pci_pir_find_link_handler(), pci_pir_walk_table(), pci_route_table, pci_link_lookup::pin, pir_bios_irqs, and pir_device.

Referenced by pcibios_pcib_route_interrupt().

Here is the call graph for this function:



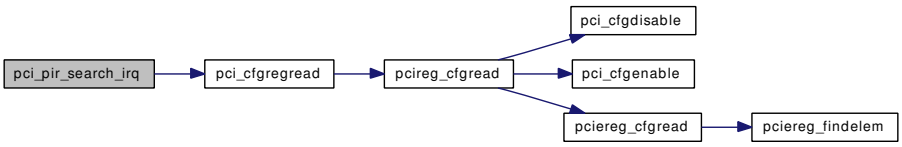
7.19.3.14 static uint8_t pci_pir_search_irq (int *bus*, int *device*, int *pin*) [static]

Definition at line 268 of file pci_pir.c.

References pci_cfgregread(), PCI_FUNCMAX, PCI_MAXHDRTYPE, PCIM_HDRTYPE, PCIM_MFDEV, PCIR_DEVVENDOR, PCIR_HDRTYPE, PCIR_INTLINE, and PCIR_INTPIN.

Referenced by pci_pir_initial_irqs().

Here is the call graph for this function:



7.19.3.15 static int pci_pir_valid_irq (struct pci_link * *pci_link*, int *irq*) [static]

Definition at line 205 of file pci_pir.c.

Referenced by pci_pir_initial_irqs(), and pci_pir_parse().

7.19.3.16 static void pci_pir_walk_table (pir_entry_handler * *handler*, void * *arg*) [static]

Definition at line 219 of file pci_pir.c.

References pci_route_count, and pci_route_table.

Referenced by pci_pir_parse(), pci_pir_route_interrupt(), and pir_resume().

7.19.3.17 static void pci_print_irqmask (u_int16_t *irqs*) [static]

Definition at line 599 of file pci_pir.c.

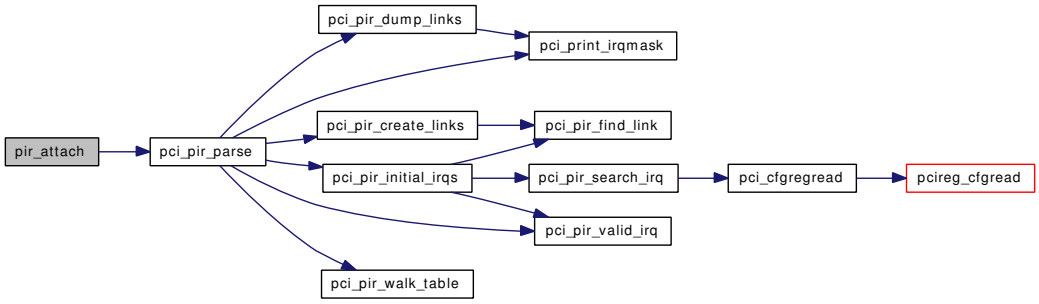
Referenced by pci_pir_choose_irq(), pci_pir_dump_links(), and pci_pir_parse().

7.19.3.18 static int pir_attach (device_t *dev*) [static]

Definition at line 671 of file pci_pir.c.

References pci_pir_parse(), and pir_device.

Here is the call graph for this function:



7.19.3.19 static int pir_probe (device_t *dev*) [static]

Definition at line 660 of file pci_pir.c.

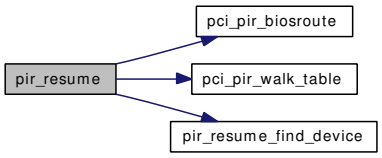
References pci_route_count.

7.19.3.20 static int pir_resume (device_t *dev*) [static]

Definition at line 695 of file pci_pir.c.

References pci_dev_lookup::bus, pci_dev_lookup::device, pci_dev_lookup::link, pci_pir_biosroute(), pci_pir_walk_table(), pci_dev_lookup::pin, and pir_resume_find_device().

Here is the call graph for this function:



7.19.3.21 static void pir_resume_find_device (struct PIR_entry * *entry*, struct PIR_intpin * *intpin*, void * *arg*) [static]

Definition at line 681 of file pci_pir.c.

References pci_dev_lookup::bus, pci_dev_lookup::device, pci_dev_lookup::link, and pci_dev_-lookup::pin.

Referenced by pir_resume().

7.19.3.22 static TAILQ_HEAD (pci_link) [static]

Definition at line 104 of file pci_pir.c.

References PCI_IRQ_OVERRIDE_MASK.

7.19.4 Variable Documentation

7.19.4.1 int pci_route_count [static]

Definition at line 103 of file pci_pir.c.

Referenced by pci_pir_probe(), pci_pir_walk_table(), and pir_probe().

7.19.4.2 struct PIR_table* pci_route_table [static]

Definition at line 101 of file pci_pir.c.

Referenced by pci_pir_probe(), pci_pir_route_interrupt(), and pci_pir_walk_table().

7.19.4.3 int pir_bios_irqs [static]

Definition at line 103 of file pci_pir.c.

Referenced by pci_pir_parse(), and pci_pir_route_interrupt().

7.19.4.4 devclass_t pir_devclass [static]

Definition at line 747 of file pci_pir.c.

7.19.4.5 device_t pir_device [static]

Definition at line 102 of file pci_pir.c.

Referenced by pci_pir_route_interrupt(), and pir_attach().

7.19.4.6 driver_t pir_driver [static]

Initial value:

```
{
        "pir",
        pir_methods,
        1,
}
```

Definition at line 741 of file pci_pir.c.

7.19.4.7 device_method_t pir_methods[ ] [static]

Initial value:

```
{

        DEVMETHOD(device_probe,         pir_probe),
        DEVMETHOD(device_attach,        pir_attach),
        DEVMETHOD(device_resume,        pir_resume),

        { 0, 0 }
}
```

Definition at line 732 of file pci_pir.c.

7.19.4.8 int pir_parsed [static]

Definition at line 103 of file pci_pir.c.

Referenced by pci_pir_parse(), and pci_pir_probe().

Index