NAME

pci – configuration files for PCI device drivers

AVAILABILITY

SPARC

DESCRIPTION

The Peripheral Component Interconnect (PCI) bus is a little endian bus. PCI devices are *self-identifying* — that is to say the PCI device provides configuration parameters to the system which allows the system to identify the device and its driver. The configuration parameters are represented in the form of name-value pairs that can be retrieved using the DDI property interfaces. See **ddi_prop_lookup**(9F) for details.

The PCI bus properties are derived from PCI Configuration Space, or supplied by the Fcode PROM if it exists. Therefore, driver configuration files are not necessary for these devices.

However, on some occasions, drivers for PCI devices may use driver configuration files to provide driver private properties. This can be done through global property mechanism. See **driver.conf**(4) for further details.

All bus drivers of class **pci** recognize the following properties:

reg
An arbitrary length array where each element of the array consists of a 5-tuple of 32-bit values. Each array element describes a logically contiguous mappable resource on the PCI bus.

The first 3 values in the 5-tuple describe the PCI address of the mappable resource. The first tuple contains the following information:

Bits 0 - 7	**8-bit Register number**
Bits 8 - 10	**3-bit Function number**
Bits 11 - 15	**5-bit Device number**
Bits 16 - 23	**8-bit Bus number**
Bits 24 - 25	**2-bit Address Space type identifier**

The Address Space type identifier may be interpreted as follows:

0x0	**Configuration Space**
0x1	**I/O Space**
0x2	**32-bit Memory Space address**
0x3	**64-bit Memory Space address**

The Bus number is a unique identifying number assigned to each PCI bus within a PCI domain.

The Device number is a unique identifying number assigned to each PCI device on a PCI bus. Note that a Device number is only unique within the set of Device numbers for a particular bus.

Each PCI device can have 1 to 8 logically independent functions, each with its own independent set of configuration registers. Each function on a device is assigned a Function number. For a PCI device with only one function, the Function number must be **0**.

The Register number field selects a particular register within the set of configuration registers corresponding to the selected function.

The second and third values in the **reg** property 5-tuple specify the 64-bit address of the mappable resource within the PCI address domain. The second 32-bit tuple corresponds to the high order 4 bytes of the 64-bit address. The third 32-bit tuple corresponds to the low order bytes.

The fourth and fifth 32-bit values in the 5-tuple **reg** property specify the size of the mappable resource. The size is a 64-bit value where the fourth tuple corresponds to the high order bytes of the 64-bit size and the fifth corresponds to the low order.

The driver can refer to the elements of this array by index, and construct kernel mappings to these addresses using **ddi_regs_map_setup**(9F). The index into the array is passed as the *rnumber* argument of **ddi_regs_map_setup**(9F).

interrupts This property consists of a single integer element array. Valid interrupt property values are **#INTA**, **#INTB**, **#INTC**, and **#INTD**. This value is derived directly from the contents of the device's Configuration Interrupt Pin register.

A driver should use an index value of **0** when registering its interrupt handler with **ddi_add_intr**(9F).

All PCI devices support the reg property. The Device number and Function number as derived from the **reg** property are used to construct the address part of the device name under **/devices**.

Only devices that generate interrupts support an **interrupts** property.

EXAMPLES

An example configuration file called **ACME,scsi-hba.conf** for a PCI driver called **ACME,scsi-hba** follows:

```
#
# Copyright (c) 1995, ACME SCSI Host Bus Adaptor
#
# ident   "@(#)ACME,csi-hba.conf  1.1  95/02/04"

hba-advanced-mode="on";
hba-dma-speed=10;
```

Two global driver properties are created: **hba-advanced-mode** (which has the string value **on**) and **hba-dma-speed** (which has the value **10** M bit/s). These properties apply to all device nodes of the **ACME,scsi-hba**.

SEE ALSO

driver.conf(4), **ddi_add_intr**(9F), **ddi_prop_lookup**(9F), **ddi_regs_map_setup**(9F)

Writing Device Drivers

IEEE 1275 PCI Bus Binding