NAME

vme – configuration files for VMEbus device drivers

AVAILABILITY

SPARC

DESCRIPTION

Some Solaris platforms support the VMEbus as a peripheral expansion bus to allow VME devices to be connected to the system. Drivers for these devices need to use driver configuration files to inform the system that the device hardware may be present. The configuration file also must specify the device addresses on the VMEbus and any interrupt capabilities that the device may have.

Configuration files for VMEbus device drivers should identify the parent bus driver implicitly using the *class* keyword. This removes the dependency on the name of the particular bus driver involved since this may be named differently on different platforms. See **driver.conf**(4) for further details of configuration file syntax.

All bus drivers of class **vme** recognise the following properties:

reg
An arbitrary length array where each element of the array consists of a 3-tuple of integers. Each array element describes a logically contiguous mappable resource on the VMEbus.

The first integer of the tuple specifies the type of access. The value is derived from the size of transfer and the address modifier bits used to access the locations. The table below shows the values used for common VME devices accessed in supervisor mode:

Address space	Value
A16D16	**0x2d**
A24D16	**0x3d**
A32D16	**0xd**
A16D32	**0x6d**
A24D32	**0x7d**
A32D32	**0x4d**

The second integer of each 3-tuple specifies the offset in the address space identified by the first element. The third integer of each 3-tuple specifies the size, in bytes, of the mappable region.

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

interrupts
An arbitrary length array where each element of the array consists of a pair of integers. Each array element describes a possible interrupt that the device might generate.

The first integer of each pair specifies the VMEbus interrupt level. The second integer of each pair specifies the VMEbus vector number. The driver can refer to the elements of this array by index, and register interrupt handlers with the system using **ddi_add_intr**(9F). The index into the array is passed as the *inumber* argument of **ddi_add_intr( )**.

All VMEbus device drivers must provide **reg** properties. The first two integer elements of this property are used to construct the address part of the device name under **/devices**.

Only devices that generate interrupts need to provide **interrupts** properties.

EXAMPLES

Here is a configuration file called **SUNW,diskctrl.conf** for a VMEbus disk controller card called **SUNW,diskctrl**.

The device provides two sets of registers, both should be accessed with supervisor accesses and the A16D32 address modifier bits (16 bits of address, 32 bit data transfers). Both registers occupy 32 bytes; one register set starts at address 0xee80, the other is at 0xef00. The device can generate interrupts at VME level 2 with a VME vector number of 0x92.

```
#
# Copyright (c) 1992, by Sun Microsystems, Inc.
#
#ident  "@(#)SUNW,diskctrl.conf 1.4     92/05/11 SMI"

name="SUNW,diskctrl" class="vme"
        reg=0x6d,0xee80,32,0x6d,0xef00,32
        interrupts=2,0x92;
```

SEE ALSO

driver.conf(4), **ddi_add_intr**(9F), **ddi_map_regs**(9F), **ddi_prop_op**(9F)

Writing Device Drivers

ANSI/IEEE Std 1014-1987: IEEE Standard for a Versatile Backplane Bus: VMEbus