NAME sysbus, isa, eisa, mca – configuration files for ISA, EISA, and MCA bus device drivers

AVAILABILITY x86

DESCRIPTION Solaris for x86 platforms support the ISA, EISA, and MCA buses as the system bus. Drivers for devices on these buses use driver configuration files to inform the system that the device hardware may be present (see **driver.conf**(4) for more information). The configuration file must specify the device I/O port addresses, any interrupt capabilities that the device may have, any DMA channels it may require, and any memory-mapped addresses it may occupy.

Configuration files for ISA, EISA, and MCA device drivers should identify the parent nexus driver implicitly using the **class** keyword. This removes the dependency on the name of the particular nexus driver involved since most drivers can operate device controllers attached to any of those buses. The ISA, EISA, and MCA nexus drivers all belong to class **sysbus**. See **driver.conf**(4) for further details of configuration file syntax.

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

intr 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 device's relative priority. This is the priority that this device's interrupt handler will receive relative to the interrupt handlers of other drivers. The priority is an integer from **1** to **16**. Generally, disks are assigned a priority of **5**, while mice and printers are lower, and serial communication devices are higher, typically **7**. **10** is reserved by the system and must not be used. Interrupts **11** and greater are high level interrupts and are generally not recommended (see **ddi_intr_hilevel**(9F)).

The second integer of each pair denotes the hardware interrupt request (IRQ) number.

The driver can refer to the elements of this array by index using **ddi_add_intr**(9F). The index into the array is passed as the *inumber* argument of **ddi_add_intr( )**.

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

reg An arbitrary-length array where each element of the array consists of a 3-tuple of integers. Each array element describes a contiguous memory address range associated with the device on the bus.

The first integer of the tuple is reserved.

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( )**.

All **sysbus** 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**. If the device has memory-mapped addresses, the first integer should be **0** and the second integer should specify the physical address; if the device does not have memory-mapped addresses, the first integer should specify a unique identifier and the second integer should be **0**. A recommended unique identifier is the **ioaddr** value; that is, specify the I/O address in both the **ioaddr** field and the first integer of the **reg** field. The third integer of each 3-tuple specifies the size, in bytes, of the mappable region.

It is recommended that drivers for devices connected to the system bus recognize the following standard property names:

ioaddr An integer that describes the I/O port base address for this device.

dmachan An integer that specifies the DMA channel used by this device. Only devices that use a DMA channel need to provide **dmachan** properties.

EXAMPLES

Here are three sample entries from three different **driver.conf** files:

name="fdc" class="sysbus" intr=5,6 ioaddr=0x3f0 dmachan=2 reg=0x3f0,0,0;

name="logi" class="sysbus" intr=1,4 ioaddr=0x23c reg=0x23c,0,0;

name="sbpro" class="sysbus" ioaddr=0x220 intr=5,7 dmachan=1 reg=0x220,0,0;

SEE ALSO

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

Writing Device Drivers