NAME

ddi_dma_lim_x86 – x86 DMA limits structure

SYNOPSIS

#include <sys/ddidmareq.h>

INTERFACE LEVEL

Solaris x86 DDI specific (Solaris x86 DDI)

DESCRIPTION

A **ddi_dma_lim** structure describes in a generic fashion the possible limitations of a device or its DMA engine. This information is used by the system when it attempts to set up DMA resources for a device. When the system is requested to perform a DMA transfer to or from an object, the request will be broken up, if necessary, into multiple sub-requests, each of which conforms to the limitations expressed in the **ddi_dma_lim** structure.

This structure should be filled in by calling the routine **ddi_dmae_getlim**(9F)**,** which sets the values of the structure members appropriately based on the characteristics of the DMA engine on the driver's parent bus. If the driver has additional limitations, it may *further restrict* some of the values in the structure members. A driver should take care to not *relax* any restrictions imposed by **ddi_dmae_getlim( )**.

STRUCTURE MEMBERS

```
u_long   dlim_addr_lo;      /* low range of 32 bit addressing capability */
u_long   dlim_addr_hi;      /* inclusive upper bound of addressing capability */
u_int    dlim_minxfer;      /* minimum effective dma transfer size */
u_int    dlim_version;      /* version number of this structure */
u_int    dlim_adreg_max;    /* inclusive upper bound of incrementing addr reg */
u_int    dlim_ctreg_max;    /* maximum transfer count minus one */
u_int    dlim_granular;     /* granularity (and min size) of transfer count */
short    dlim_sgllen;       /* length of DMA scatter/gather list */
u_int    dlim_reqsize;      /* maximum transfer size in bytes of a single I/O */
```

The **dlim_addr_lo** and **dlim_addr_hi** fields specify the address range the device's DMA engine can access. The **dlim_addr_lo** field describes the lower 32 bit boundary of the device's DMA engine; **dlim_addr_hi** describes the inclusive upper 32 bit boundary. The system will allocate DMA resources in a way that the address for programming the device's DMA engine (see **ddi_dma_cookie**(9S) or **ddi_dma_segtocookie**(9F)) will be within this range. For example, if your device can access the whole 32 bit address range, you may use [**0,0xFFFFFFFF**].

The **dlim_minxfer** field describes the minimum effective DMA transfer size (in units of bytes). It must be a power of two. This value specifies the minimum effective granularity of the DMA engine. It describes the minimum amount of memory that can be touched by the DMA transfer. As a resource request is handled by the system, the **dlim_minxfer** value may be modified contingent upon the presence (and use) of I/O caches and DMA write buffers in between the DMA engine and the object that DMA is being performed on. After DMA resources have been allocated, the resultant minimum transfer value can be retrieved using **ddi_dma_devalign**(9F).

The **dlim_version** field specifies the version number of this structure. This field should be set to **DMALIM_VER0**.

The **dlim_adreg_max** field describes an inclusive upper bound for the device's DMA engine address register. This handles a fairly common case where a portion of the address register is simply a latch rather than a full register. For example, the upper 16 bits of a 32 bit address register may be a latch. This splits the address register into a portion which acts as a true address register (lower 16 bits) for a 64 kilobyte segment and a latch (upper 16 bits) to hold a segment number. To describe these limits, you would specify **0xFFFF** in the **dlim_adreg_max** structure member.

The **dlim_ctreg_max** field specifies the maximum transfer count that the DMA engine can handle in one segment or cookie. The limit is expressed as the maximum count minus one. This transfer count limitation is a per-segment limitation. It is used as a bit mask, so it must be one less than a power of two.

The **dlim_granular** field describes the granularity of the device's DMA transfer ability, in units of bytes. This value is used to specify, for example, the sector size of a mass storage device. DMA requests will be broken into multiples of this value. If there is no scatter/gather capability, then the size of each DMA transfer will be a multiple of this value. If there is scatter/gather capability, then a single segment will not be smaller than the minimum transfer value, but may be less than the granularity; however the total transfer length of the scatter/gather list will be a multiple of the granularity value.

The **dlim_sgllen** field specifies the maximum number of entries in the scatter/gather list. It is the number of segments or cookies that the DMA engine can consume in one I/O request to the device. If the DMA engine has no scatter/gather list, this field should be set to one.

The **dlim_reqsize** field describes the maximum number of bytes that the DMA engine can transmit or receive in one I/O command. This limitation is only significant if it is less than (**dlim_ctreg_max** +1) ∗ **dlim_sgllen**. If the DMA engine has no particular limitation, this field should be set to **0xFFFFFFFF**.

SEE ALSO

ddi_dmae(9F), **ddi_dma_addr_setup**(9F), **ddi_dma_buf_setup**(9F), **ddi_dma_devalign**(9F), **ddi_dma_setup**(9F), **ddi_dma_lim_sparc**(9S) **ddi_dma_req**(9S),