NAME

ddi_iopb_alloc, ddi_iopb_free – allocate and free non-sequentially accessed memory

SYNOPSIS

```
#include <sys/ddi.h>
#include <sys/sunddi.h>

int ddi_iopb_alloc(dev_info_t *dip, ddi_dma_lim_t *limits, u_int length,
        caddr_t *iopbp);

void ddi_iopb_free(caddr_t iopb);
```

INTERFACE LEVEL

Solaris DDI specific (Solaris DDI).

ARGUMENTS

ddi_iopb_alloc( )

dip	A pointer to the device's **dev_info** structure.
limits	A pointer to a DMA limits structure for this device (see **ddi_dma_lim_sparc**(9S) or **ddi_dma_lim_x86**(9S)). If this pointer is **NULL**, a default set of DMA limits is assumed.
length	The length in bytes of the desired allocation.
iopbp	A pointer to a **caddr_t**. On a successful return, **iopbp* points to the allocated storage.

ddi_iopb_free( )

iopb	The *iopb* returned from a successful call to **ddi_iopb_alloc( )**.

DESCRIPTION

ddi_iopb_alloc( ) allocates memory for DMA transfers and should be used if the device accesses memory in a non-sequential fashion, or if synchronization steps using **ddi_dma_sync**(9F) should be as lightweight as possible, due to frequent use on small objects. This type of access is commonly known as *consistent* access. The allocation will obey the alignment and padding constraints as specified in the *limits* argument and other limits imposed by the system.

Note that you still must use DMA resource allocation functions (see **ddi_dma_setup**(9F)) to establish DMA resources for the memory allocated using **ddi_iopb_alloc( )**.

In order to make the view of a memory object shared between a CPU and a DMA device consistent, explicit synchronization steps using **ddi_dma_sync**(9F) or **ddi_dma_free**(9F) are still required. The DMA resources will be allocated so that these synchronization steps are as efficient as possible.

ddi_iopb_free( ) frees up memory allocated by **ddi_iopb_alloc( )**.

RETURN VALUES

ddi_iopb_alloc( ) returns:

DDI_SUCCESS	Memory successfully allocated.
DDI_FAILURE	Allocation failed.

CONTEXT

These functions can be called from user or interrupt context.

SEE ALSO **ddi_dma_free**(9F), **ddi_dma_setup**(9F), **ddi_dma_sync**(9F), **ddi_mem_alloc**(9F), **ddi_dma_lim_sparc**(9S), **ddi_dma_lim_x86**(9S), **ddi_dma_req**(9S)

Writing Device Drivers

NOTES This function uses scarce system resources. Use it selectively.