

Subject: : AmigaOS4

Topic: : DMA buffer for PCI busmaster headache

DMA buffer for PCI busmaster headache

Author: : geennaam

Date: : 2021/2/10 23:54:31

URL:

I want to allocate a DMA buffer for a device driver but I am completely lost how to accomplish this.

The PCI device wants to fetch commands from a command ring buffer located in main memory.

I have to provide the base address of this command buffer to the PCI device and the offset pointer where I wrote my last command. The PCI device will then fetch commands from this command buffer until its read offset pointer will match my write offset pointer.

I have allocated the command buffer with AllocMemTags in the following manner:

```
pCmdBuf= IExec->AllocVecTags((uint32)(buffer_size), // 1kB
    AVT_Type,          MEMF_SHARED,
    AVT_Lock,          TRUE, // Don't allow to be moved in memory
    AVT_Contiguous,   TRUE, // DMA buffer must be contiguous
    AVT_Alignment,    128, // Align on 128 bytes boundary
    AVT_PhysicalAlignment, 128, // Physical align on 128 bytes boundary
    AVT_ClearWithValue, 0,
    TAG_END);
```

I wrote the address of pCmdBuf to the command buffer base address register inside the PCI device. Wrote the command to the next entry in pCmdBuf and updated the Write pointer register to match the location of this new command.

I can see that the DMA fetched something because his offset readpointer increased with 1 and matches my offset write pointer. But I do not get a response. This could mean that the DMA controller fetched some bogus command and chose to ignore it. I am sure that the command is valid. I am also sure that the response part of the PCI device is also setup correctly.

I did an experiment with disabling caches for this memory buffer in hypervisor mode but that didn't help. On top of that I acquired the physical address of the buffer and wrote this to the PCI device command buffer base address. This also didn't work.

I noticed a IExec functions StartDMA/EndDMA. But from what I understand these are for devices that support scatter/gather functionality which this pci device doesn't support.

And there is also an older and deprecated functions combinations called CachePreDMA/CachePostDMA/CacheClearE. This is supposed to flush caches and provide a physical

address.

AFAIK these function are supposed to deal with the fact that the Amigaos4 memory architecture will not provide a contiguous buffer. But I provided the tags in AllocMemVecs to force contiguous memory block and both virtual and physical alignment. And since disabling caching didn't help either, I am right back at where I started this post: completely lost in what is the correct way to handle these kind of buffers. So I hope that someone can help me.

I also do not fully understand the PCI mechanism. Will the MMU handle DMA request to virtual address spaces? Or will a DMA access from a PCI device go to a physical memory address?

The PCI device expects little endian format so naturally everything is manually converted BE<-> LE by me in either direction when it is read/written through the PCI ibus. Only the functions in the PCIIFace for PCI config space are byte swapped automatically.