Writing Device Drivers for Zorro Devices

Written by Geert Uytterhoeven <geert@linux-m68k.org>
Last revised: September 5, 2003

1. Introduction

The Zorro bus is the bus used in the Amiga family of computers. Thanks to AutoConfig(tm), it's 100% Plug-and-Play.

There are two types of Zorro busses, Zorro II and Zorro III:

- The Zorro II address space is 24-bit and lies within the first 16 MB of the Amiga's address map.

- Zorro III is a 32-bit extension of Zorro II, which is backwards compatible with Zorro II. The Zorro III address space lies outside the first 16 MB.

2. Probing for Zorro Devices

Zorro devices are found by calling `zorro_find_device()', which returns a pointer to the `next' Zorro device with the specified Zorro ID. A probe loop for the board with Zorro ID `ZORRO_PROD_xxx' looks like:

```
    struct zorro_dev *z = NULL;

    while ((z = zorro_find_device(ZORRO_PROD_xxx, z))) {
        if (!zorro_request_region(z->resource.start+MY_START, MY_SIZE,
                                  "My explanation"))
        ...
    }
```

`ZORRO_WILDCARD' acts as a wildcard and finds any Zorro device. If your driver supports different types of boards, you can use a construct like:

```
    struct zorro_dev *z = NULL;

    while ((z = zorro_find_device(ZORRO_WILDCARD, z))) {
        if (z->id != ZORRO_PROD_xxx1 && z->id != ZORRO_PROD_xxx2 && ...)
            continue;
        if (!zorro_request_region(z->resource.start+MY_START, MY_SIZE,
                                  "My explanation"))
        ...
    }
```

3. Zorro Resources

Before you can access a Zorro device's registers, you have to make sure it's not yet in use. This is done using the I/O memory space resource management functions:

```
    request_mem_region()
    release_mem_region()
```

Shortcuts to claim the whole device's address space are provided as well:

```
    zorro_request_device
    zorro_release_device
```

4. Accessing the Zorro Address Space

The address regions in the Zorro device resources are Zorro bus address regions. Due to the identity bus-physical address mapping on the Zorro bus, they are CPU physical addresses as well.

The treatment of these regions depends on the type of Zorro space:

- Zorro II address space is always mapped and does not have to be mapped explicitly using z_ioremap().

  Conversion from bus/physical Zorro II addresses to kernel virtual addresses and vice versa is done using:

  ```
      virt_addr = ZTWO_VADDR(bus_addr);
      bus_addr = ZTWO_PADDR(virt_addr);
  ```

- Zorro III address space must be mapped explicitly using z_ioremap() first before it can be accessed:

  ```
      virt_addr = z_ioremap(bus_addr, size);
      ...
      z_iounmap(virt_addr);
  ```

5. References

linux/include/linux/zorro.h
linux/include/asm-{m68k,ppc}/zorro.h
linux/include/linux/zorro_ids.h
linux/drivers/zorro
/proc/bus/zorro