CHRP™ ISA Floppy Controller Device Binding

to:

IEEE 1275-1994

Standard for Boot

(Initialization, Configuration)

Firmware

Revision: 1.0 Unapproved DRAFT

Date: May 6, 1996

Table of Contents

1. Purpose of this Device Binding

This document specifies the application of Open Firmware to the PowerPC Common Hardware Reference Platform (CHRP™) ISA Floppy Controller, including device-specific requirements and practices for initialization, properties, and methods. This device shall be a child of an ISA or EISA Bus Node.

2. Revision History

Revision 1.0 Unapproved DRAFT May 6, 1996	Initial revision. Jordan Brown, Sunsoft and John Kingman, IBM editors

3. References

This Open Firmware System binding standard *shall* be used in conjunction with the following publications. When the following standards are superseded by an approved revision, the revision *shall* apply.

[1] *IEEE Std 1275-1994 Standard for Boot (Initialization, Configuration) Firmware, Core Practices and Requirements*.

[2] *Core Errata, IEEE P1275.7/D4.*

[3] *ISA/EISA/ISA-PnP binding to: IEEE Std 1275-1994, Standard for Boot (Initialization, Configuration) Firmware*.

[4] *Device Support Extensions to: IEEE Std 1275-1994, Standard for Boot (Initialization, Configuration) Firmware*.

[5] *PowerPC Microprocessor Common Hardware Reference Platform: I/O Device Reference*. This document describes the PowerPC Common Hardware Reference Platform (CHRP™) System Standard I/O Devices; hardware registers, register locations, and hardware attributes.

[6] *PowerPC Microprocessor Common Hardware Reference Platform binding to: IEEE Std 1275-1994, Standard for Boot (Initialization, Configuration) Firmware*.

[7] *Open Firmware Recommended Practice: Interrupt Mapping*.

[8] *Open Firmware Recommended Practice: Generic Names.*

4. Definition of Terms

EISA: Extended Industry Standard Architecture

ISA: Industry Standard Architecture

5. Device Characteristics (Informative)

The ISA Floppy Controller generally controls one or two floppy disk drives. The controller supports both high- and low-density diskette media. Floppy controllers in the ISA world have traditionally supported two operational modes - PC/AT and PS/2. CHRP[5] requires PS/2 mode controllers.

5.1. Address Formats and Representations

This section describes the physical address representation.

5.1.1. Physical Address Format: Numerical Representation

The numerical representation of a floppy device address is a binary integer, in the range ’0’ to ’1’.

5.1.2. Physical Address Format: Text Representation

The textual representation of a floppy device address is an ASCII “0” or an ASCII “1.”

6. Device-specific Configuration Variables

None.

7. Device Nodes

7.1. Properties

As specified in [1], [3] and [6], with the following additions or modifications. This node shall not have a “`ranges`” property.

7.1.1. Open Firmware-defined Properties for Device Nodes.

`"name"` **S**

Standard *property name*, specifies the generic name of the device.

The meaning of this property is as defined in Open Firmware core document [1], as modified by the *Generic Names Recommended Practice* [8]. The value for nodes described by this specification shall be “`fdc`”.

`"device_type"` **S**

Standard *property name* to define the device’s implemented interface.

The meaning of this property is as defined in the Open Firmware core document [1]. The value for nodes described by this specification shall be “`fdc`”.

`"compatible"` **S**

Standard *property name*, specifies device names with which this device is compatible.

The meaning of this property is as defined in Open Firmware, as modified by the *Generic Names Recommended Practice* [8]. As described in those documents, the entries are a list of device names with which this device is compatible, starting with the name of the device itself and progressing through successively less precise and possibly less functional compatible devices.

The value of this property shall include “`pnpPNP,700`.” If the controller supports auto-eject this property shall also include “`chrp,fdc`” preceding “`pnpPNP,700`.”

Additional entries may be supplied, at their appropriate position in the list, to describe devices with which this device is compatible.

`"reg"` **S**

Standard *property name* to define the package’s registers.

The meaning of this property is as defined in the Open Firmware core document [1]. It describes the device's register set. The values which shall be assigned to this property are explained in the *ISA/EISA/ISA-PnP binding*[3] and the *I/O Device Reference*[5]

`"interrupts"` **S**

Standard *property name* to define a package’s interrupts.

The meaning of this property is as defined in the *Interrupt Mapping Recommended Practice* [7]. The values assigned to this property are explained in the *ISA/EISA/ISA-PnP binding*[3] and the *I/O Device Reference*[5].

`"#address-cells"` **S**

Standard *prop-name* to define the number of cells required to represent the physical addresses for the children of this node.

prop-encoded-array: Integer constant 1, encoded as with **`encode-int`**.

The value of "`#address-cells`" for this node *shall* be `1`.

`"#size-cells"` **S**

Standard *prop-name* to define the number of cells necessary to represent the length of a physical address range.

prop-encoded-array: Integer constant 0, encoded as with **encode-int.**

The value of "`#size-cells`" for this node shall be 0 because the children of this node do not consume any physical address space.

`"dma"` **S**

Standard *property name* to define a package's DMA channels.

The values assigned to this property are explained in the *ISA/EISA/ISA-PnP binding*[3] and the *I/O Device Reference*[5]. This device shall have one DMA channel assigned to it.

`"auto-eject"` **S**

property name to identify that the auto-eject functionality is supported.

This property shall be present if the controller supports the auto-eject function, otherwise it shall be absent.

7.1.2. Device-specific Properties for Device Nodes

As specified in [3], without addition or modification.

7.1.3. Open Firmware-defined Methods for Device Nodes

As specified in [1] and [3], without addition or modification.

7.1.3.1. Device Arguments for "Open" Method

As specified in [4] and [6], without addition or modification.

7.1.4. Device-specific Methods for Device Nodes

As specified in [4] and [6], without addition or modification.

8. Child Nodes

8.1. Properties

As specified in [1] and [3], with the following additions or modifications.

8.1.1. Open Firmware-defined Properties for Child Nodes

`"name"` **S**

Standard *property name*, specifies the generic name of the device.

The meaning of this property is as defined in Open Firmware core document [1], as modified by the Generic Names Recommended Practice. The value for nodes described by this specification shall be "`disk`".

`"device_type"` **S**

Standard *property name* to define the device's implemented interface.

The meaning of this property is as defined in the Open Firmware core document [1]. The value for nodes described by this specification shall be "`block`".

"reg" S

prop-name, specifies the number of this floppy drive.

prop-encoded-array: An integer, encoded with **encode-int**.

The value of this is property is the drive number, in the range of 0-1.

8.1.2. Device-specific Properties for Child Nodes

None.

8.1.3. Device-specific Methods for Child Nodes

As specified in [4], with the following addition for a controller which supports auto-eject:

eject (--) M

eject shall activate the media eject function in the drive.

9. User Interface Commands

9.1. Open Firmware-defined User Interface Commands

None.

9.2. Device-specific User Interface Commands

None.

10. Device State

10.1. Device State When Client is Started

This device shall be initialized with its address, DMA and interrupt assigned and enabled when the client is started.

Refer to [5] for more information on the state of this device when the client is started.

10.2. Device State Required When Client Calls Open Firmware

For devices not selected as Open Firmware's "console input device" or "console output device" device, there is no requirement. Typically, this device is unsuitable for use as an Open Firmware console input or console output device.

11. Other Commentary