Printer Protocol Definition for IEEE-1394

Proposal Draft Ver. 0.2d
December 5, 1996

SEIKO EPSON Corporation

EPSON RESEARCH CENTER
EPSON Software Development Center

This is an internal draft proposal for printer working group of IEEE 1394 Trade association. Feedback may be sent to EPSON's TA reflector e-mail address PP1394@erc.epson.com.

The latest version of documents is available at http://www.ercipd.com.

Table of Contents

Figures

1. Overview

1.1 Scope

This proposal describes a printer access protocol which should be a standard for printing and has reliable connectivity on the High Performance Serial Bus IEEE 1394.
The protocol should be applied to any printers and plotters which generate bit images, any plotters and printers which receive vector data, portions of multiple functional printers and any software which emulate the printer behavior.

The protocol covers following systems and devices.

- Print Server System which has multiple connectivity for clients
- Software which emulates the printer behavior
- Printer which receives bit images
- Printer which interprets page description language source code
- Printer which perform halftoning internally
- Video printer
- Plotter which receive vector data
- Printer potion of multi-function device

1.2 Purpose

Specifications described in this document are intended to design a reliable standard protocol for printing and goals of the document are:

- To develop a method for multiple host devices to be able to share a printer effectively and safely on an IEEE 1394 serial bus.
- To provide a method for a host device to specify one printer and to use it, avoiding conflicts with multiple printers.
- To keep reliability for various types of printers and various types of commands and engines on a high performance serial bus.
- To avoid conflicts with host devices and other devices when image scanners and digital cameras are connected on an IEEE 1394 Serial Bus.
- To provide most suitable methods both for computers those have lots of memory and some types of digital camera which does not have enough memory.

2. Normative references

- IEEE 1284.4 Standard for Data Delivery and Logical Channels for IEEE Std. 1284 Interfaces Draft D0.01 June 26, 1996
- Serial Bus Protocol 2 (SBP-2) Working Draft, X3T10 Project 1155D Rev.1, July 17,1996
- ISO/IEC 13213:1994, Control and Status Register (CSR) Architecture for Microcomputer Buses
- IEEE Std. 1394-1995, Standard for a High Performance Serial Bus

3. Definitions and notations

3.1 Printer Types

To simplify the discussion and to clear the point of view, this document distinguishes printers into three classes.

Class A Printers:

Printers that have interpreters for page description languages would be identified as a Class A device. Examples include printers which implement the PostScript language, plotters that print vector data and print server systems.

In this class, a host device mainly uses text data to describe printing jobs. Printers and plotters parse input data, interpret them into images and return status information to the host device. Status information may consist of several arrays of character or binary data. The host device may query the printer to send current status or installed font information at any time. Each of these steps would be processed interactively. Servers allow multipe users to access the printer which requires some form of access control.

The implementation of reliable, bi-directional data communication is important. Bit errors can cause the interpreter to parse a syntax error, return incorrect data or image incorrect information.

Class B Printers:

Printers that receive bit image data with control commands or encoded data and convert them to image data are classified as 'Class B Printers'. The most popular types of printers in the 'SOHO' market now, are included in this class including page printers, ink-jet printers, and multi-function products. Conventional job processes of these printers return simple status.

A device using the Class B declaration will typically be easier to implement than a Class A device. Class B printers typically receive bit image data after half-toning has been done, and the bit image data is compressed. If data is lost or altered, errors will occur when it is uncompressed. Therefore reliable, data delivery is important in this class.

Class C Printers:

Printers that emulate video display devices should be classified as 'Class C Printers'. The printing data for these printers is often an array of RGB pixel data. Whether the printer is a black and white printer or a color printer, a printer in this class can receive pixel data which has brightness and darkness level, and can convert it into a printing dot. A sublimational color thermal printer is one of the most typical printer in this class. Various printers which perform halftoning internally could be included in this class. For these types of printers, even though some data may be damaged during transfer, the output quality may still be acceptable.

Multi-function printer with image scanners in one box may also be included in this class.

3.2 Definitions

PDL page description language
JCL job control language
MLC Multiple Logical Channels

3.3 Notation

4. Data Transfer Models

4.1 Agents and Memory Bus Model

4.1.1 Agents

An agent described in this specification is the software object that receive requests from producer and consume them to provide software services for requester. Each agent has a single 32-byte control register to which the host device may signal a request by means of a IEEE 1394 Serial Bus write transaction that contains an entire request.

IEEE 1394 printing protocol provides these agents.

- Login Agent
- Operation Agent
- Management Agent

- Plug and Play Agent

Login agent serves the delegation when the status request is received or when the connection channels are occupied.

- Delegation Agent

Operation Agent consists of these four agents.

- Conventional Agent
- Reverse direction Agent
- Interactive Agent
- Stream Control Agent

Hierarchies of these agents are shown in below.

- Login Agent
    - Delegation Agent
- Operation Agent
    - Conventional Agent
    - Reverse directional Agent
    - Interactive Agent
    - Stream Control Agent
- Management Agent
- Plug and Play Agent

Some of agents are unnecessary to implement for some class of printers. Table 1 shows the relations for class and agents. Each printer class has different implementation requirements. Table 1 shows the relations for printer class and agents.

Table 1. Agents implementations for each Printer Class

Agents	Class A	Class B	Class C
Login	Mandatory	Mandatory	Mandatory
Conventional	Mandatory	Mandatory	Mandatory
Reverse directional	Mandatory	Optional	Not implemented
Interactive	Mandatory	Optional	Not implemented
Plug and Play	TBD	TBD	TBD
Stream Control	Optional	Not implemented	Mandatory
Management	Mandatory	Mandatory	Mandatory

Only the login agent should be declared in the Configuration ROM. The protocol described in this proposal addresses communications between a print server system and clients. Print server systems allow multiple client connectivity and have capability to perform multiple printing jobs. Thus the printer server system would support multiple agents to receive requests from several clients.

The *login agent* is an unique agent whose address is declared in Configuration ROM. The *login agent* can service a single request at a time. When the agent is occupied with a request, subsequent writes to its control register shall be rejected with a busy acknowledge or a conflict error response.

The *conventional agent* serves traditional data flow from a host device to a printer. The agent is a fetch agent, thus the host device will write requests in its own memory space and the printer will fetch the requests via this agent.

The *reverse direction agent* allows host devices to read printer status at any time. A printer that support the *reverse directional agent* updates its status within its own memory space as needed.

Printers with PDL or JCL interpreter provide interactive conversations with host devices. Host devices expect the printer to respond and to send data according to the command from the host device. The interactive agent can service this conversation with the host device.

The *Plug and Play agent* can service Microsoft's Plug and Play architecture. The specification of this agent is to be discussed.

The *stream control agent* services the Isochronous transfer channels.

The *management agent* services the process management of printing jobs.

4.1.2 Printer Aware Register

A host device must provide a *printer aware register* to receive notification writes from a printer when the printer provides the reverse direction agent. The printer shall signal a quadlet write when the printer updates the reverse direction data in its own memory. The host device may fetch data by means of an IEEE 1394 Serial Bus read transaction.

4.1.3 Doorbell Register

A printer shall have the *doorbell register* to accept update messages from a host device. The host device shall maintain a linked list of request blocks. When the host device adds a new request or remove some requests from the linked list, it must notify the printer by writing a value on the *doorbell register* via an IEEE 1394 Serial Bus write transaction.

4.1.4 Current Request Register

A printer shall provide the current request register to keep most recent value of the first request address of a linked list.

4.1.5 Status Base Register

A printer shall provide the status base register which indicates the base address of status block. The data structure of the status block is different for class A, B and C printers.

4.1.6 Memory Bus Model

The basic model of the IEEE 1394 printer protocol based on Memory Bus model. Figure 1 shows an overview of the architecture. Each request to an agent is listed in a linked list. The tail of a linked list is a terminated request shown in Figure 2.

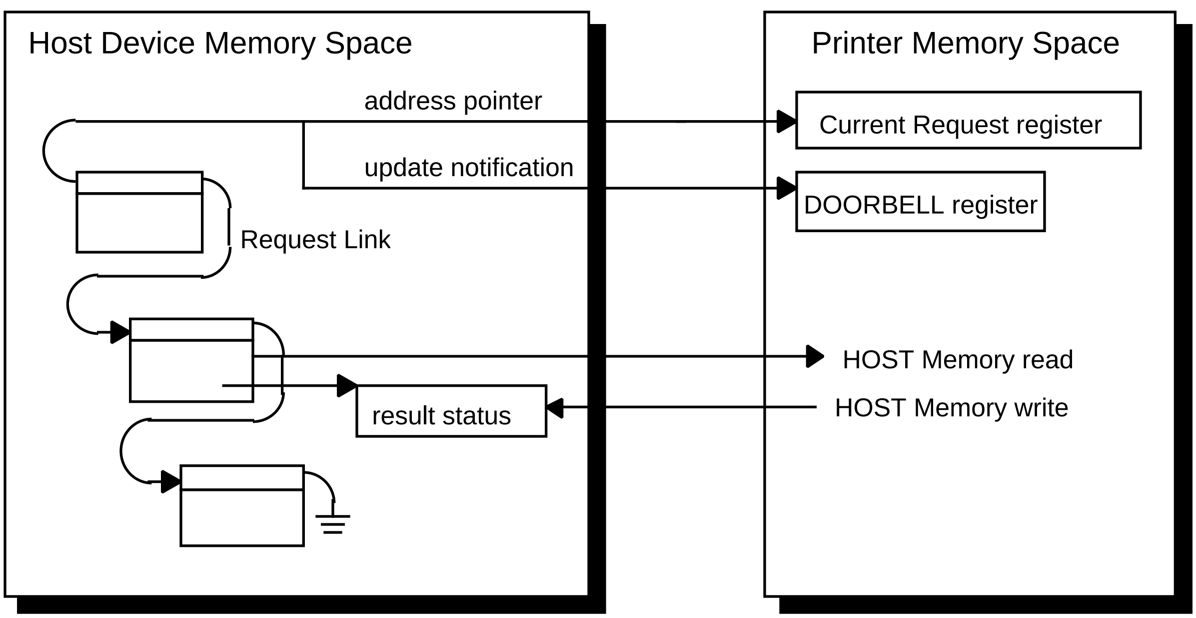

Figure 1. Overview of memory bus model

A printer reads each element of the linked list by means of an IEEE 1394 Serial Bus read transaction then writes the result status onto the status buffer for corresponding request block. Each request block provides an address offset to indicate the status buffer location in its own memory space.

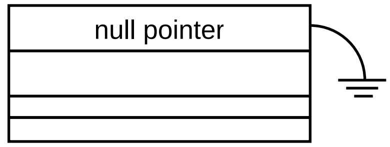

Figure 2. Terminated request block

4.1.7 Garbage collection

A host device may provide garbage collection job to purge insignificant request blocks and reuse memory space. When a printer completes the job requested by a request block, the printer invokes a quadlet write transaction and changes the contents of the status buffer within the corresponding request block. The host device may remove the request block when it detects a complete status in a request block. Then the host device shall update the linked list.

4.1.8 Logical channels

Conventional agents and reverse direction agents serve multiple logical channels. The data structure of each agent shall have a *Printer_info_block* which contains a 6 bit field that indicates logical channel number from 0 to 62. The value of 63 in the field is reserved as a broadcast channel. This feature may be used to address the multiple channel capability of IEEE 1284.4.

Logical channels serve vendor specific communication scheme. Assume that a color ink jet printer intends to return the printer status to a host device. Typical color ink jet printers have multiple ink cartridges, the printer may report empty status of cyan ink through logical channel 12, empty status of magenta ink through channel 13 and empty status of yellow ink through channel 14. Or, assume a multi-function printer intends to get printing data and data to send using facsimile engine. The host device may send printing data through channel 8, and send data to be sent by facsimile through channel 11.

4.2 Login

A host device shall provide this address for a printer at login process.

- Address offset of the register for printer aware register

A printer shall provide these addresses for a host device at login process.

- Address offset of the register from normal agent
- Address offset of the register from stream control agents
- Address offset of the register from management agent

The normal and the stream control agents shall specify the base address of the agent's CSR's. These registers have address offset value on CSR's.

- Address offset for DOORBELL register
- Address offset for Current_request_register
- Address offset for Status_base_register

4.2.1 Multiple connectivity

Login agent serves multiple connectivity through the agent. The printer may return login status which consist of some address offset of various agent. These agents are allocated dynamically. In other words, the printer may provide multiple address spaces for multiple clients. As the IEEE 1394 printer protocol is designed to cover print server systems, multiple connectivity is an important feature of the protocol. However consumer printing devices that intend to be used in home market may only need single connection capability.

4.2.2 Login arbitration

When the login was unsuccessful, the login agent of the printer returns an address offset of a delegation buffer to the requester. The host device whose login request was rejected shall place the request block on the buffer addressed by the delegation buffer offset. The request block shall include a time stamp indicating when the request was rejected for the first time. The printer shall provide several delegation buffer spaces dynamically when login requests are received.

The login agent shall determine the highest priority pending requests in the delegation buffer when it starts a new session. If there were several login requests in the delegation buffer, the login agent shall define the criteria used to accept the next request.

If the printer detects a pending request on delegation buffer, the printer shall start the solicitation process and shall write specific quadlet data on *the printer_aware_register* of the host device chosen by the login agent criteria for the next print job.

When the host device is notificated by quadlet write transaction on the *printer_aware_register*, it shall re-send login request block to the login agent of the printer.

The current strategy uses a time stamp which allows the login agent the opportunity to provide equal chances for multiple host devices when they are in race condition of requests.

4.2.3 Login-less conectivity and status monitor ring

Printers are particular devices from some mechanical points of view. Because a printer consumes paper and supplies, but, a HDD does not. This means when a host device requests a printing job, the printer could be out of paper, supplies or a combination of errors. Thus the host device may ask the printer to report current status, before it starts printing. Or the host device may spawn monitoring software to know the current status of the printer while the user is using other application software to prepare the document.

A status monitoring software shall ask the printer to report status even while the printer is processing other job requests. This means that the printer protocol shall serve login-less request for status monitoring software. The protocol serves login-less connection through the delegation agent.

4.2.4 Plug and Play agent and delegations

The plug and play agent is a form of delegation agent which serves login-less connections to fetch PnP ID from the target printer.

4.3 Conventional data flow

A conventional data flow model is a traditional data flow model that serves a host device to be a data producer and makes a printer to be a data consumer. Figure 3 shows a typical operation of the model.

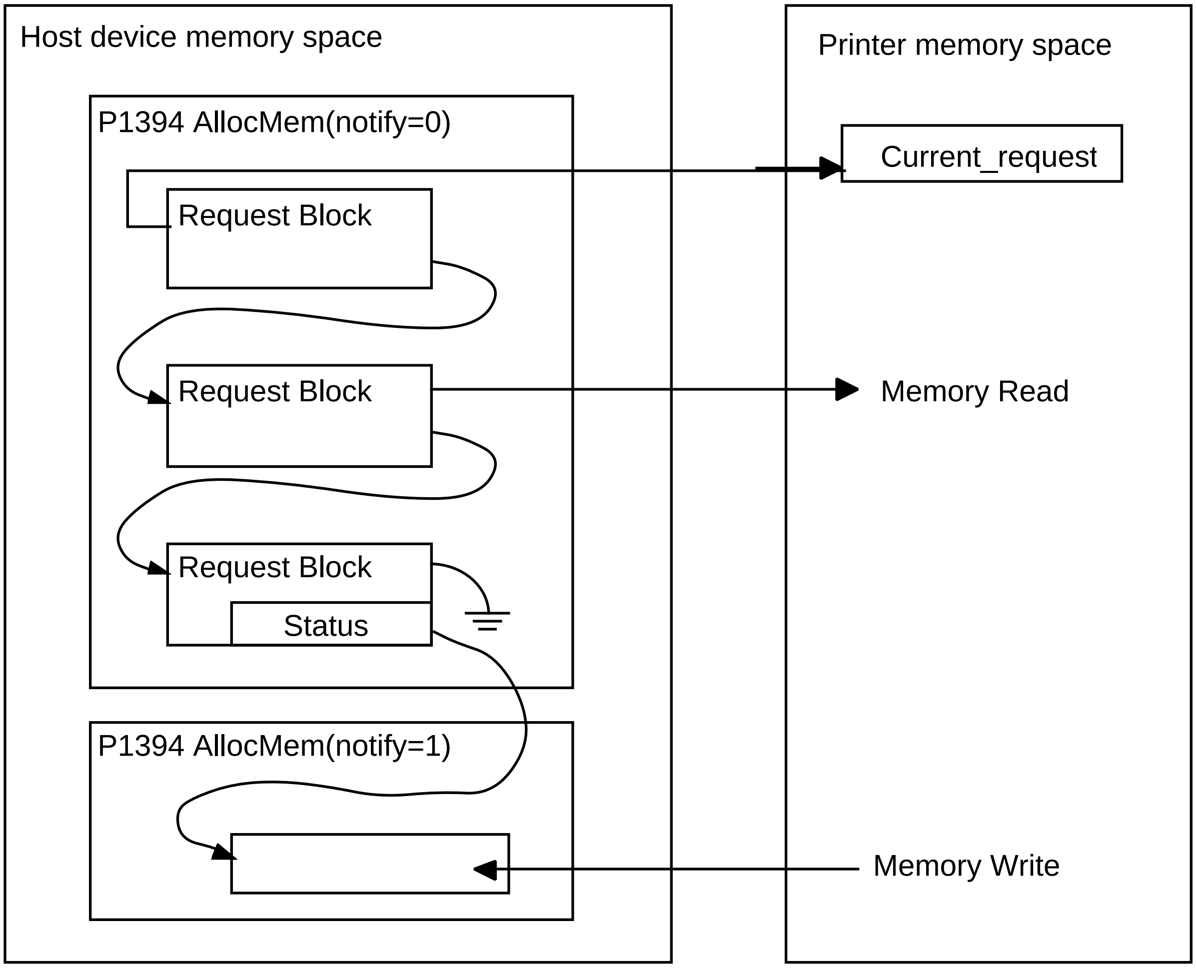

Figure 3. Data flow of conventional Agent

The host device may provide request blocks on its own memory and shall make them linked in a list when it has multiple requests. Each request shall include an address offset to the status buffer to receive the job result from the printer. The printer may fetch requests at its own pace.

4.4 Reverse directional data flow

Each class of printer can provide data for a host device at any time. However data may not be sent immediately to the host device. The printer writes data on its own memory, and the host device fetch data at its own pace.

The host device gives an address offset for the printer aware register at login process. The printer may notify the host that it provided reverse direction data by changing the value of the register with a quadlet write transaction of the IEEE 1394 Serial Bus.

A class A printer can have large amount of reverse directional data using a page table structure. The overview of this model is shown in Figure 4.

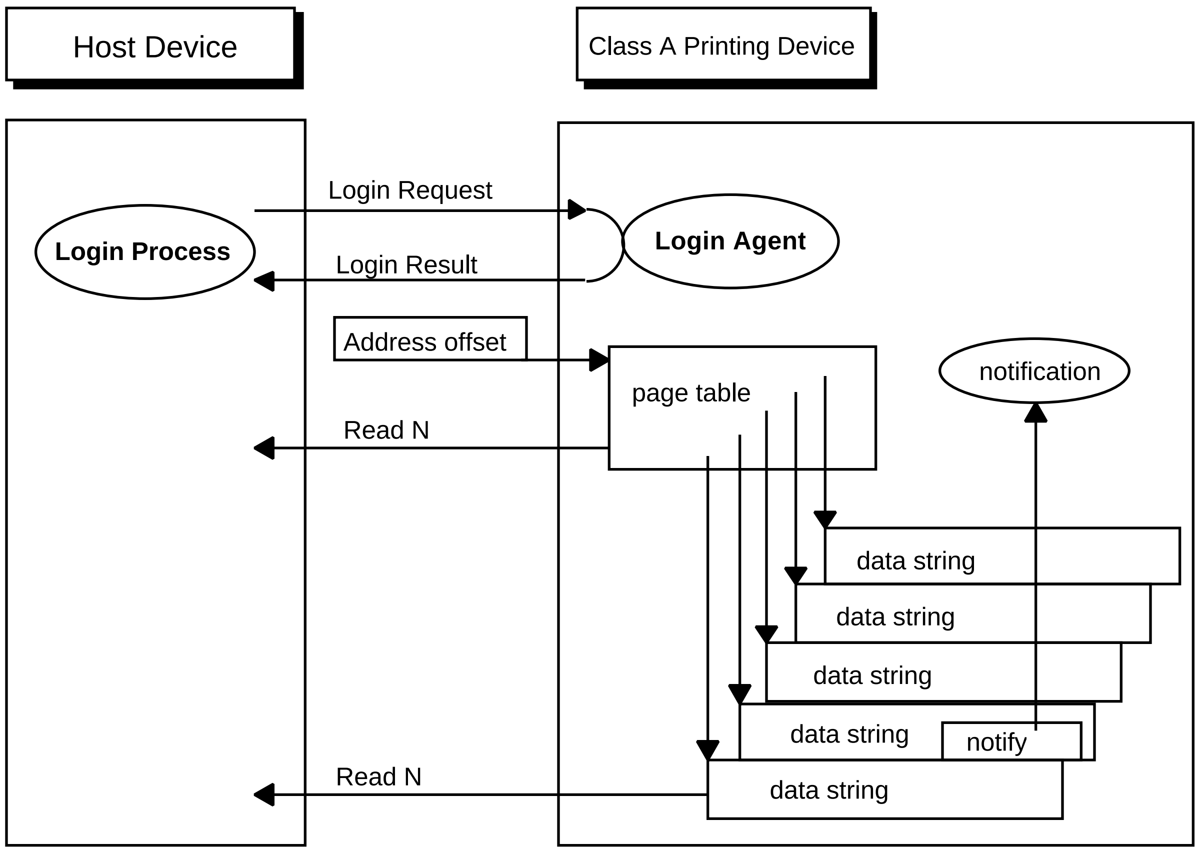

Figure 4. Reverse directional data flow of a class A printer

The maximum size of each page is 512 byte. As described in section 5, each element of the page table has an address offset value to specify the data buffer for the corresponding element. The base address for these offset values is specified by the *status_base_offset* field of the page table. (See 5.4)

Data flow model of the reverse directional agent of class A printers serves multiple logical channels. Each element of the page table has a *printer_info_block* field that indicates the number of the logical channel and any element specifies the logical channel independently. The agent also provides broadcast for all of logical channels.

If the device layer of the kernel software for the printer can provide notification mechanism, or in other words, if the printer can provide a special address space which cause a notification when data is written inor read from that space, an address offset value of an element may be allocated on the memory space which cause memory access notification. Thus the printer can detect the event when the host device read that space.

The reverse direction data flow of a class B printer is simpler than that of a class A printer. Figure 5 shows an overview of data flow for a class B printer. The status block is a single data structure which specifies the address offset value of the status string.

The reverse direction agents of class B printers are suitable for devices that do not have enough memory to support large messages in a limited space. The data flow model of the reverse direction agent for class B printers can service multiple logical channels. However, the status block of the class B printer can only specify one logical channel as one response data. The class B printer shall update the status block when it has a new message for other logical channels.

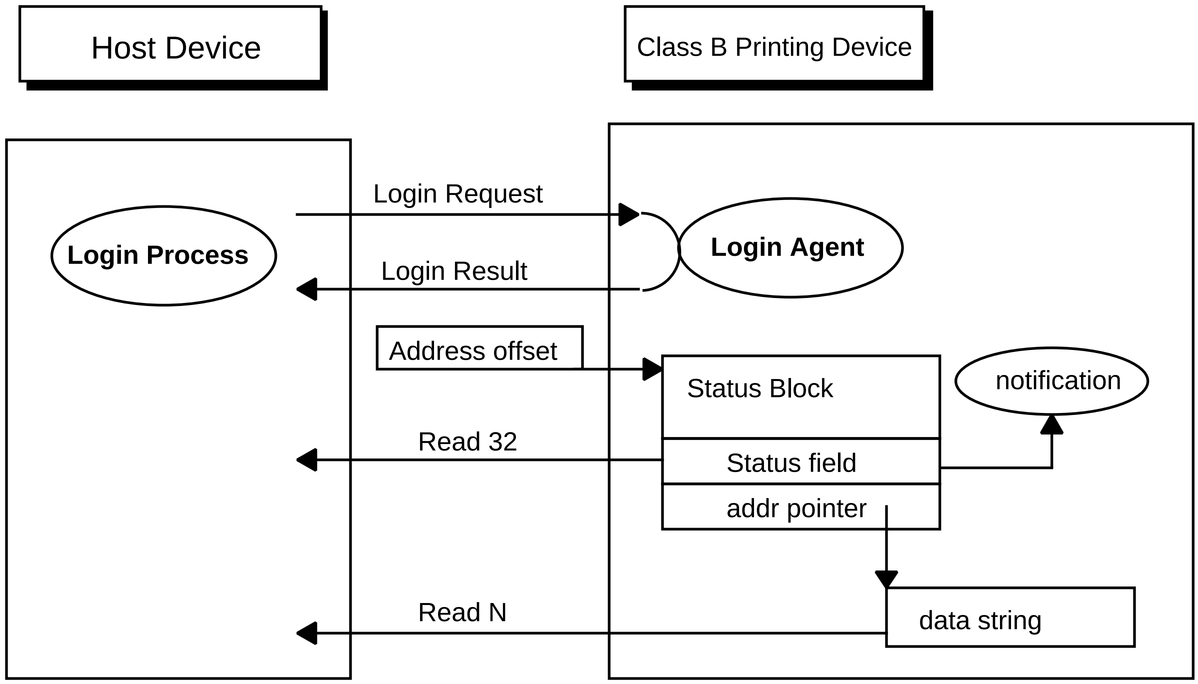

Figure 5. Reverse directional data flow for a class B printer

4.5 Interactive data flow

An interactive agent provides bi-directional communication protocol through an IEEE-1394 Serial Bus. A host device places request blocks that are intended to be interpreted by the printer in host device's memory space. Each request block has a address offset which is pointing to a buffer on memory of the host device. The printer shall store the result of the request in host space after it interpreted the request

4.6 Print process management

4.6.1 Reconnection

This proposal defines job steps which are held sequentially from login to logout as a 'session'. If a bus reset has happened while a printer and a host device is executing a session, the printer shall invoke the reconnection transaction.

5. Data Structures

5.1 Data structures for Login

5.1.1 Login request blocks

Before any other requests can be made of an IEEE 1394 printing device, the host device shall invoke a login request of which data structure shown below.

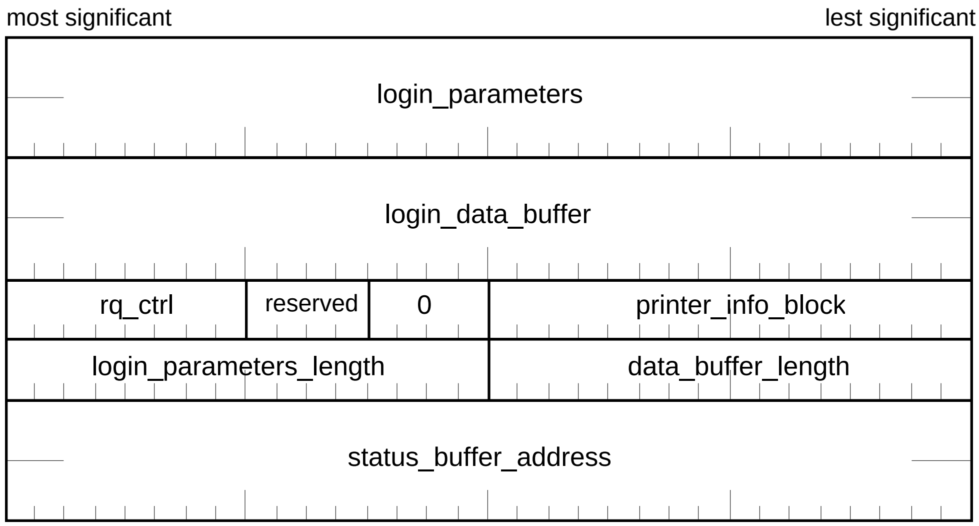

Figure 6. Login request block

The *login_parameters* and *login_parameters_length* fields specify the address and size of a buffer in system memory that contains parameters shown in Figure 6 in this section.

The *login_data_buffer* and *data_buffer_length* fields specify the address and size of a buffer allocated for the return of login data.

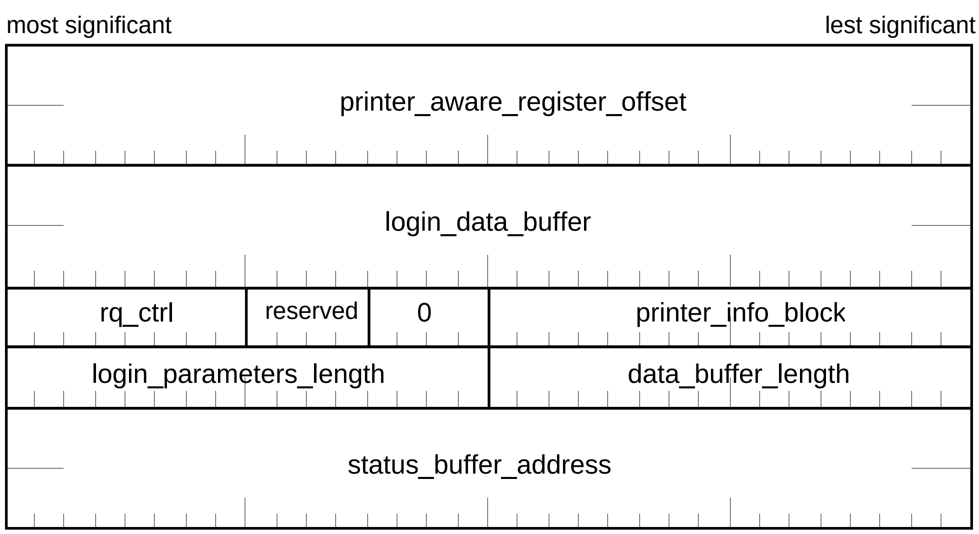

Figure 7. Login parameters

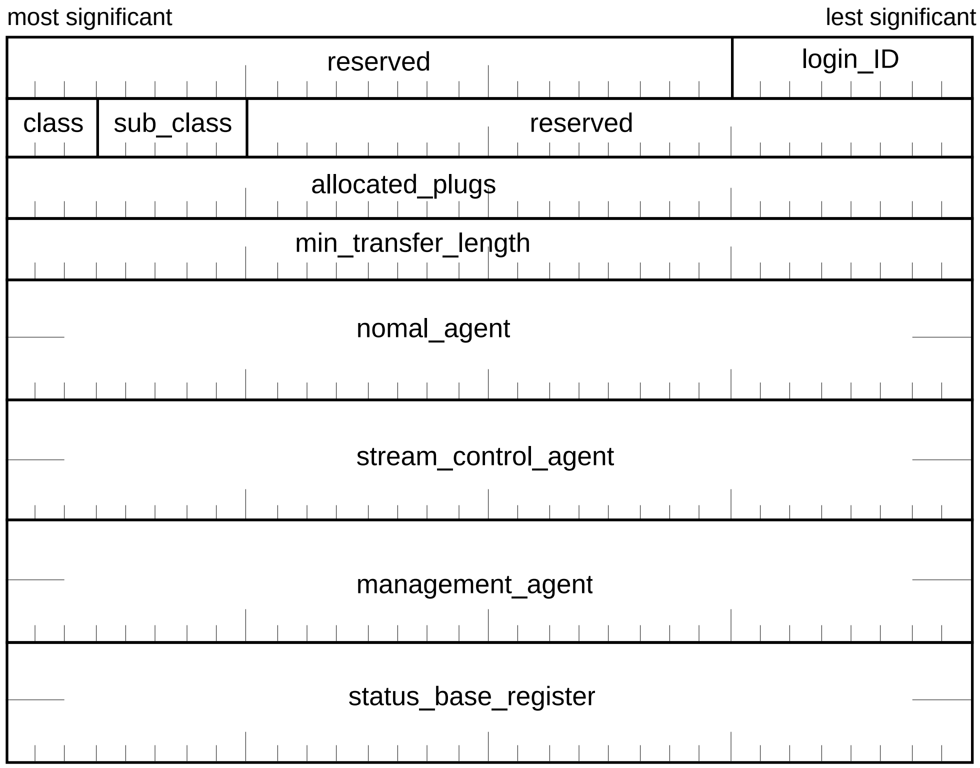

Figure 8. Login data

The *class* field specify the class of a printer.

value	Class of the printer
0	class A
1	class B
2	class C
3 - 7	reserved for future standardization

The sub_class field is reserved for future standardization.

The r (reverse) bit shall specify whether or not the reverse directional data flow is supported in this printer class. This field is insignificant for class A and C printers.

The I (interactive) bit shall specify whether or not the interactive data flow is supported in this printer class. This field is insignificant for class A and C printers.

5.2 Data structure for Logout

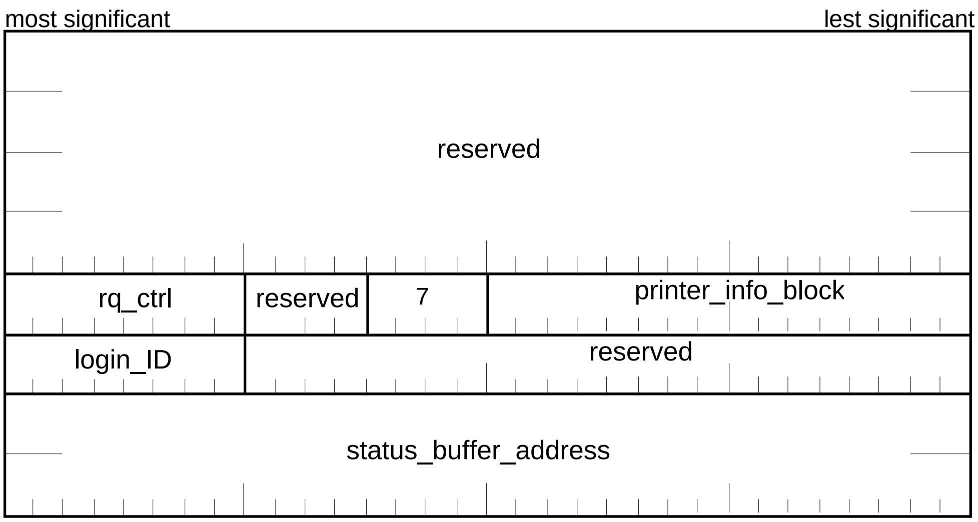

Figure 9. Logout request block

5.3 Data structures for Conventional data flow

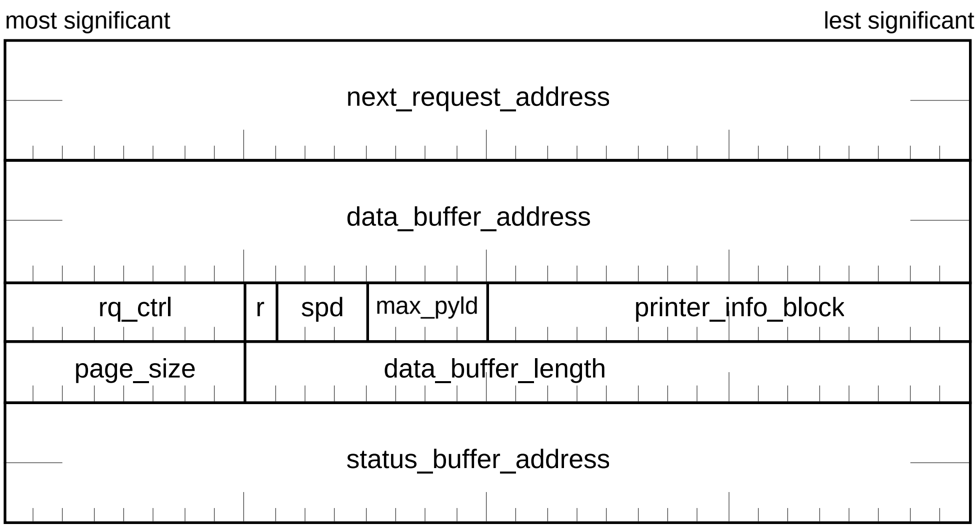

Figure 10. Conventional Data Request

5.3.3 Printer Info Block

The data structure of the printer_info_block is shown in Figure 11.

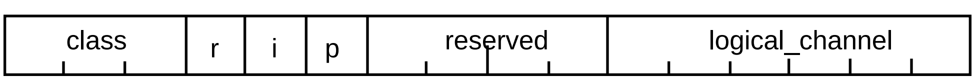

Figure 11. Printer_info_block

The *logical_channel* field shall specify input logical channel of a printing device.

5.4 Data structures for Reverse directional data flow

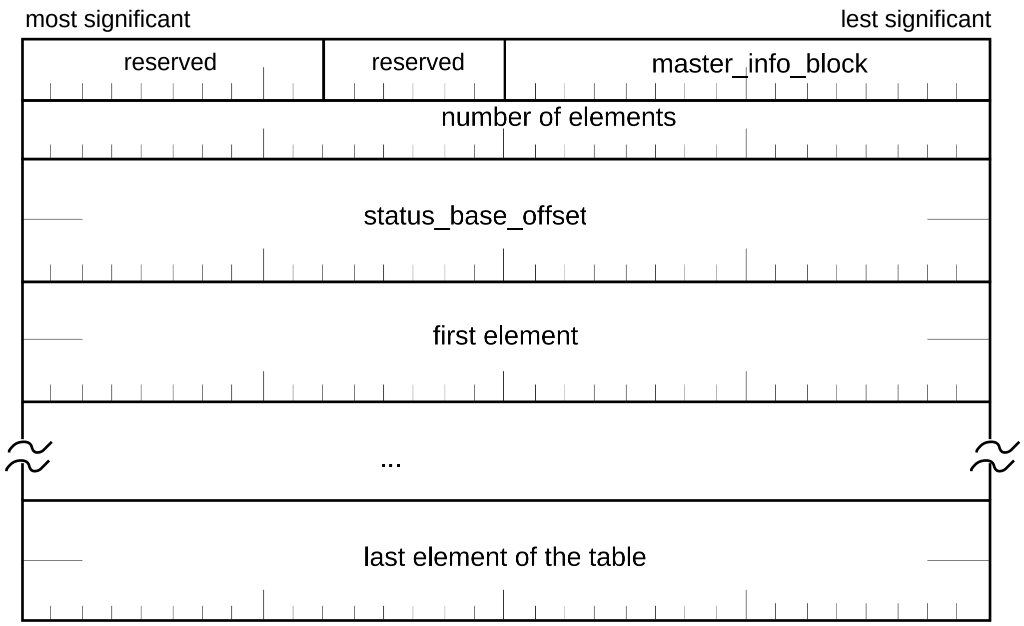

Figure 12. Class A page table

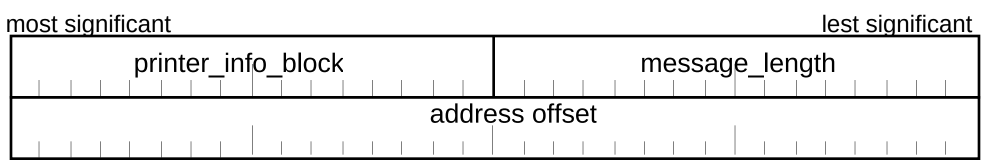

Figure 13. Class A element of page table

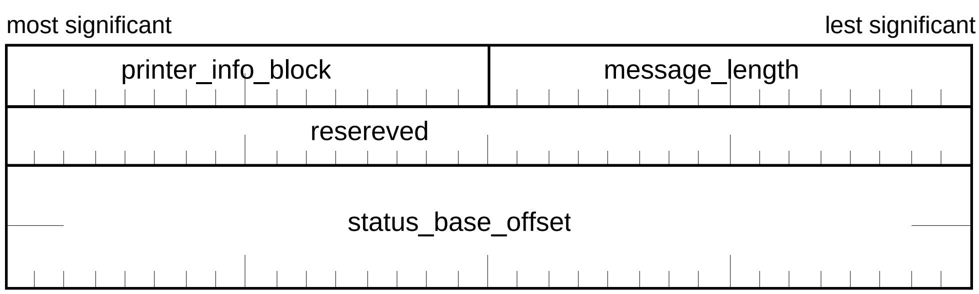

Figure 14. Class B status block

The *status_base_offset* field directly addresses the data buffer associated with the status block. In class B printers' status block the data buffer shall be contiguous within Serial Bus address space of which length is specified by the *message_length* field.

5.5 Data structures for Interactive data flow

Figure 14 shows the data structure of a request block. The host device can place two types of buffer spaces for the printer. The first buffer space is data buffer space in which the host device place command to be interpreted by the printer. Second buffer space is status buffer space in which the printer returns interpreted result.

The *data_buffer_address* shall specify the address of a buffer allocated for the place holder of command from the host device. The length of this buffer is specified by the *data_buffer_length* field and the *page_size* field.

The *page_size* field shall have a value of zero if *data_buffer_address* directly addresses the data buffer associated with the request block. Otherwise, if the *page_size* have a nonzero value, the *data_buffer_address* specifies the top address of the page table, and the value of the *page_size* field defines the page size of elements described by the page table.

The *status_buffer_address* field shall specify the address of a buffer allocated for the return of interpreted result from the printer. If *stat_page_size* has a value of zero, the *status_buffer_length* shall specify the size in bytes. Otherwise, when *stat_page_size* has a nonzero value, the value indicates the size of page table.

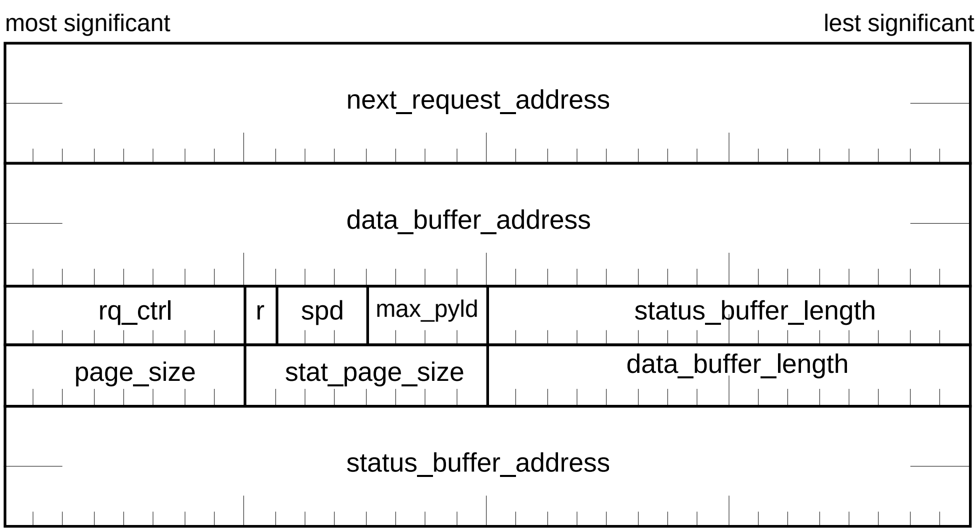

Figure 15. Interactive Request Block

5.5 Data structures for Print process management

6. Control and Status Registers

6.1 IEEE 1394 Address space usage

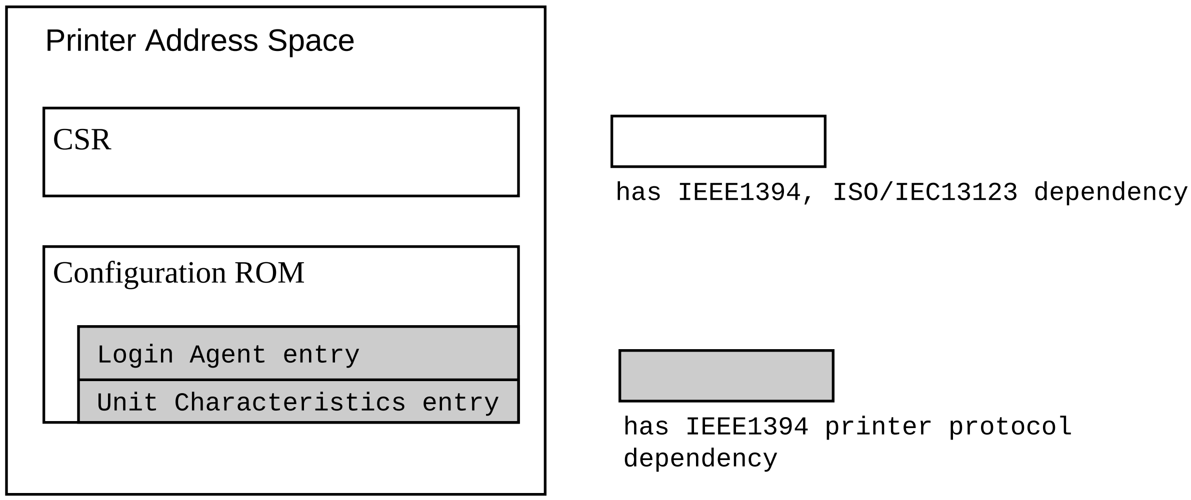

Figure 16. Unknown

6.2 Control and Status Registers

6.3 Configuration ROM

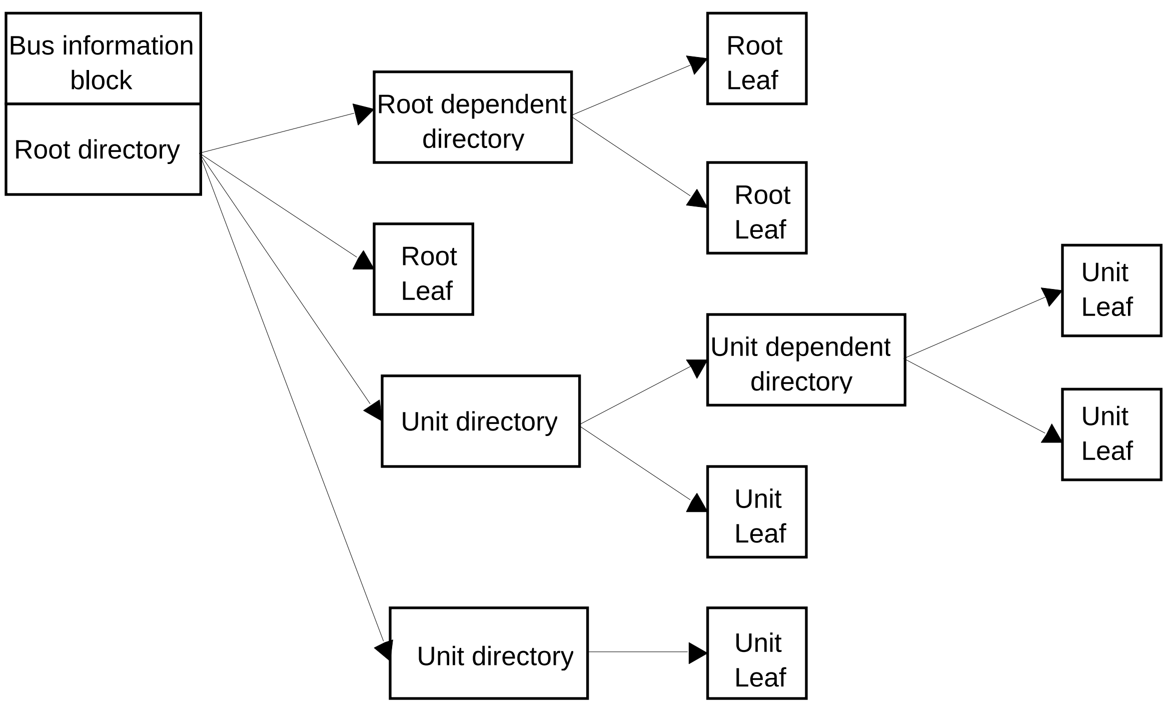

Figure 17. Configuration ROM hierarchy

6.3.1 Root directory

Offset	0-7	8-15	16-23	24-31
400h	02h	FFh	rom_crc_value	
404h	31h	33h	39h	34h
408h	0 0 1 0 rsv	FFh	4h	rsv
40Ch	node_uniq_ID hi			
410h	node_uniq_ID lo			
414h	root_length		CRC	
418h	03h	module_vender_ID		
41Ch	0Ch	rsv	0	
420h	8Dh	node_unique_id leaf offset		
424h	D1h	unit_directory_offset		

6.3.2 Unit directory

Offset	0-7	8-15	16-23	24-31
0h	0003h		CRC	
4h	12h	unit_spec_ID (TBD)		
8h	13h	unit_sw_version (TBD)		
Ch	78h	Login_Agent entry		
10h	39h	Unit_Characteristics entry		

6.3.2.1 Login_Agent entry

most significant … lest significant

78_{16}	csr_offset

6.3.2.1 Unit_Characteristics entry

most significant … lest significant

39_{16}	qm	o	reserved	i	x	e	r	m	device_type

6.3.3 Node unique ID leaf

Offset	0-7	8-15	16-23	24-31
0h	0002h		CRC	
4h	node_vender_ID			chip_ID_hi
8h	chip_ID_lo			

Figure 18 Node Unique ID Leaf

Offset	0-7	8-15	16-23	24-31
0h	0003h		CRC	
4h	node_vender_ID			chip_ID_hi
8h	chip_ID_lo			
Ch				