

7711 to 7751 MIGRATION

Application Note





Hi/fnTM supplies two of the Internet's most important raw materials: compression and encryption. Hi/fn is also the world's first company to put both on a single chip, creating a processor that performs compression and encryption at a faster speed than a conventional CPU alone could handle, and for much less than the cost of a Pentium or comparable processor.

As of October 1, 1998, our address is:

**Hi/fn, Inc.
750 University Avenue
Los Gatos, CA 95032
info@hifn.com
http://www.hifn.com
Tel: 408-399-3500
Fax: 408-399-3501**

**Hi/fn Applications Support Hotline:
408-399-3544**

Disclaimer

Hi/fn reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

Hi/fn warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with Hi/fn's standard warranty. Testing and other quality control techniques are utilized to the extent Hi/fn deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

HI/FN SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of Hi/fn products in such critical applications is understood to be fully at the risk of the customer. Questions concerning potential risk applications should be directed to Hi/fn through a local sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

Hi/fn does not warrant that its products are free from infringement of any patents, copyrights or other proprietary rights of third parties. In no event shall Hi/fn be liable for any special, incidental or consequential damages arising from infringement or alleged infringement of any patents, copyrights or other third party intellectual property rights.

"Typical" parameters can and do vary in different applications. All operating parameters, including "Typicals," must be validated for each customer application by customer's technical experts.

The use of this product may require a license from Motorola. A license agreement for the right to use Motorola patents may be obtained through Hi/fn or directly from Motorola.

AN-0002-00 (12/98) © 1997-1998 by Hi/fn, Inc., including one or more U.S. patents No.: 4,701,745, 5,003,307, 5,016,009, 5,126,739, 5,146,221, 5,414,425, 5,414,850, 5,463,390, 5,506,580, 5,532,694. Other patents pending.

This product is NOT FOR EXPORT outside the U.S. without prior authorization from the U.S. Government

Table of Contents

1	Overview	5
2	PCI Bus Differences	5
2.1	Hardware PCI differences	5
2.2	Software PCI differences	6
3	DMA differences	6
3.1	Hardware DMA differences	6
3.2	Software DMA differences	6
4	Miscellaneous differences	7
4.1	Miscellaneous Hardware differences	7
4.2	Miscellaneous Software differences	7

Figures

Figure 1.	7751 Security Levels	7
-----------	----------------------------	---

THIS PAGE INTENTIONALLY BLANK

1 Overview

The 7751 and 7711 devices are encryption processors used for implementing security with compression in applications such as data communications. Both devices implement security and compression algorithms for the popular communication security protocols, and support multiple concurrent sessions per device. Both devices perform the same packet processing functions as the 7711. In addition, the 7751 supports PCI bus I/O and contains a bus DMA master.

The 7711 requires a companion device to facilitate the DMA and bus interface functions. Typically, this companion device can be an off the shelf bridge chip, a separate FPGA or incorporated in a system ASIC. Hi/fn developed a Xilinx-based gate array, called “Jupiter”, which performs the PCI interface and bus master DMA functions.

This application note describes the migration issues for upgrading from a 7711 solution, using Jupiter for the PCI and DMA bus master functions, to a 7751 single-chip compression, encryption, and authentication solution with PCI bus master DMA capability.

This application note is organized to describe the hardware and software differences between the PCI and DMA portions of the two solutions, as well as some miscellaneous issues.

This document assumes that the reader is already familiar with the operation of the 7711, Jupiter, and 7751.

2 PCI Bus Differences

This section compares the differences in the PCI bus operation between a system that utilizes Jupiter and 7711, and a system that utilizes the 7751. This section addresses hardware and software PCI Bus issues when migrating from a 7711/Jupiter PCI based design to a 7751 PCI based design.

2.1 Hardware PCI differences

- 1) 7751's PCI bus burst size is 64 bytes (16 transfers). Jupiter's PCI bus bursts size is 32 bytes. 7751 better utilizes the PCI bus.
- 2) 7751 performs individual reads and writes within a burst in a single clock. Jupiter performs individual reads within a burst in a single clock and individual writes within a burst in two clocks. 7751 writes to the PCI bus more efficiently.
- 3) 7751 supports the PCI Read Multiple command. Jupiter does not support the Read Multiple PCI bus command. If the PCI system controller allows the Read-Multiple command, but only supports smaller burst sizes, then the Read Multiple command significantly reduces bus overhead for 64 byte 7751 transfers.
- 4) 7751 will function in either 3.3V and 5V PCI buses. Jupiter only functions in 5V PCI buses.
- 5) 7751 supports a 64x16 EEPROM to load PCI configuration information at reset. Jupiter does not support an EEPROM.

2.2 Software PCI differences

- 1) 7751 contains two additional PCI configuration registers, located at 0x40: Retry Timeout and TRDY timeout. Retry Timeout sets the number of retries 7751 PCI master will perform. TRDY Timeout sets the number of PCI clocks the 7751 PCI master will wait for TRDY. Both fields should be set to zero upon initialization and after a reset.
- 2) 7751's Device ID is 0x0005. Jupiter's Device ID is 0x0004. The 7751 value can be configured in the EEPROM, since the 7751 supports loadable configuration parameters in an EEPROM.
- 3) 7751 allows the system to access the PCI configuration space while the device is polling for descriptors. Jupiter's polling must be disabled before accessing the PCI configuration space.
- 4) The 7751 supports Read Multiple PCI bus commands by using the Cache Line size PCI configuration register setting as a threshold to switch between Read and Read-multiple bus operations. Jupiter does not support the Cache Line Size PCI configuration register. Read-Multiple may allow up to 64 byte 7751 bursts for system controllers that only support smaller burst sizes
- 5) 7751 allows byte addressing of the DMA registers pointed to by Base Address Register 1 (BAR1). Jupiter only allows 32-bit accesses to BAR1 registers pointed to by both Base Address Registers. Both devices only allow 32-bit accesses to 7751 Processing Unit registers (pointed to by Base Address Register 0).

3 DMA differences

This section compares the differences in the DMA operation between a system that utilizes Jupiter and 7711, and a system that utilizes the 7751. This section addresses hardware and software DMA issues when migrating from a 7711/Jupiter DMA based design to a 7751 DMA based design.

3.1 Hardware DMA differences

- 1) 7751 allows the command and source data descriptor rings to point to byte-aligned system addresses. (7751 generates 32-bit aligned destination data and results.) Jupiter requires 32-bit aligned descriptor pointers.
- 2) 7751 features a second poll timer utilized when 7751 encounters an invalid descriptor. This timer, called the Invalid Poll Scalar field in the DMA Configuration register, limits the number of bus accesses 7751 performs while there is no work to be done. When 7751 encounters a valid descriptor, the polling increases to the value in the Polling Frequency field of the DMA Configuration register. This flexibility decreases bus utilization for periods of inactivity and decreases bus latency for periods of activity. Jupiter only has one poll timer for both valid and invalid descriptors.

3.2 Software DMA differences

- 1) The Last bit in the 7751's four DMA descriptors can be configured to be an input to the destination or results descriptor (set by the system), as well as an output (set by the device). This flexibility allows the system to allocate the memory boundary. Jupiter treats the Last bit in its four DMA descriptors as an output only. So, Jupiter's DMA operations will continue until all destination bytes are transferred.
- 2) The 7751 features the Mask Done Interrupt bit in all 4 DMA descriptor rings. This feature allows the software to select which descriptor will cause

- a DONE interrupt in the descriptor ring. This allows the system to limit CPU overhead only to critical descriptors.
- 3) 7751 features the Over bit in the destination data and results descriptor rings. This signals that more data was to have been written to the area to which this descriptor points (i.e. destination overrun with respect to the DMA controller). Jupiter does not provide this feature.
 - 4) 7751 allows the four ring address register values to be changed any time if 7751 polling is disabled. Jupiter pre-fetches descriptors, and only allows changing the destination and result ring address register after software reset.

4

Miscellaneous differences

This section compares the other differences in the operation between a system that utilizes Jupiter and 7711, and a system that utilizes the 7751. This section addresses other hardware and software issues when migrating from a 7711/Jupiter DMA based design to a 7751 DMA based design.

4.1 Miscellaneous Hardware differences

- 1) The 7751's chip ID returns multiple chip IDs depending on the security level for which the device is configured. The 7711's chip ID returns a single chip ID. Both 7751 and 7711 chip IDs also contain revision information.
- 2) The 7751's revision ID represents the stepping level of the DMA portion of the 7751. Jupiter does not have a revision ID.
- 3) Although the Hi/fn 7751 and 7711 processors are produced in the same package, they are not pin compatible.
- 4) Both the 7711 and 7751 devices support 3.3V or 5V system; however, the 7751 interfaces directly to 5V or 3.3V PCI buses. The 7711's companion device determines which bus level signaling is supported. In the case of Hi/fn implementations, Jupiter does not support 3.3V PCI signaling.

4.2 Miscellaneous Software differences

- 1) The 7751 powers on with encryption disabled. In order to enable encryption capabilities, an unlock software mechanism must be followed, utilizing a key that is both programmed into the EEPROM and resident in the software mechanism. Thus the serial EEPROM is required to enable the encryption capabilities of the 7751.

There are 3 security levels that can be enabled via this unlock procedure. Figure 1 summarizes these security levels, along with the different 7751 Chip IDs and algorithms supported.

Details of this unlock software mechanism are available from Hi/fn.

Security Level	7751 Chip ID	Algorithm Support
Security Level 0 (default)	0x30XX	LZS, MPPC
Security Level 1	0x10XX	LZS, MPPC, MD5, SHA, DES
Security Level 2	0x11XX	LZS, MPPC, MD5, SHA, DES, 3-DES, RC4

Figure 1. 7751 Security Levels

Jupiter has no system security enabling feature.

- 2) 7751 does not have a data register; 7711 does. So, the CPU cannot be used to transfer commands, data, and results directly through the 7751. The 7711 allows the CPU to directly transfer commands, data, and results; however, this may be significantly less efficient.