Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

Universe IID/IIB™ User Manual

May 12, 2010

6024 Silver Creek Valley Road, San Jose, California 95138
Telephone: (800) 345-7015 • (408) 284-8200 • FAX: (408) 284-2775
Printed in U.S.A.

GENERAL DISCLAIMER

Integrated Device Technology, Inc. reserves the right to make changes to its products or specifications at any time, without notice, in order to improve design or performance and to supply the best possible product. IDT does not assume any responsibility for use of any circuitry described other than the circuitry embodied in an IDT product. The Company makes no representations that circuitry described herein is free from patent infringement or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent, patent rights or other rights, of Integrated Device Technology, Inc.

CODE DISCLAIMER

Code examples provided by IDT are for illustrative purposes only and should not be relied upon for developing applications. Any use of the code examples below is completely at your own risk. IDT MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE NONINFRINGEMENT, QUALITY, SAFETY OR SUITABILITY OF THE CODE, EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. FURTHER, IDT MAKES NO REPRESENTATIONS OR WARRANTIES AS TO THE TRUTH, ACCURACY OR COMPLETENESS OF ANY STATEMENTS, INFORMATION OR MATERIALS CONCERNING CODE EXAMPLES CONTAINED IN ANY IDT PUBLICATION OR PUBLIC DISCLOSURE OR THAT IS CONTAINED ON ANY IDT INTERNET SITE. IN NO EVENT WILL IDT BE LIABLE FOR ANY DIRECT, CONSEQUENTIAL, INCIDENTAL, INDIRECT, PUNITIVE OR SPECIAL DAMAGES, HOWEVER THEY MAY ARISE, AND EVEN IF IDT HAS BEEN PREVIOUSLY ADVISED ABOUT THE POSSIBILITY OF SUCH DAMAGES. The code examples also may be subject to United States export control laws and may be subject to the export or import laws of other countries and it is your responsibility to comply with any applicable laws or regulations.

LIFE SUPPORT POLICY

Integrated Device Technology's products are not authorized for use as critical components in life support devices or systems unless a specific written agreement pertaining to such intended use is executed between the manufacturer and an officer of IDT.

1. Life support devices or systems are devices or systems which (a) are intended for surgical implant into the body or (b) support or sustain life and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any components of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

Contents

Figures

Tables

About this Document

This section discusses the following topics:

Scope

The *Universe IID/IIB User Manual* discusses the features, capabilities, and configuration requirements for the Universe II. It is intended for hardware and software engineers who are designing system interconnect applications with the device.

Document Conventions

This document uses the following conventions.

Signal Notation

Signals are either active high or active low. Active low signals are defined as true (asserted) when they are at a logic low. Similarly, active high signals are defined as true at a logic high. Signals are considered asserted when active and negated when inactive, irrespective of voltage levels. For voltage levels, the use of 0 indicates a low voltage while a 1 indicates a high voltage.

For voltage levels, the use of 0 indicates a low voltage while a 1 indicates a high voltage. For voltage levels, the use of 0 indicates a low voltage while a 1 indicates a high voltage.

Each signal that assumes a logic low state when asserted is followed by an underscore sign, "_". For example, SIGNAL_ is asserted low to indicate an active low signal. Signals that are not followed by an underscore are asserted when they assume the logic high state. For example, SIGNAL is asserted high to indicate an active high signal.

The asterisk sign "*" is used in this manual to show that a signal is asserted low and that is used on the on the VMEbus backplane. For example, SIGNAL* is asserted to low to indicate an active low signal on the VMEbus backplane.

Object Size Notation

- A *byte* is an 8-bit object.
- A *word* is a 16-bit object.
- A *doubleword* (Dword) is a 32-bit object.
- A quadword is a 64-bit (8 byte) object.
- A Kword is 1024 16-bit words.

Numeric Notation

- Hexadecimal numbers are denoted by the prefix *0x* (for example, 0x04).
- Binary numbers are denoted by the prefix *0b* (for example, 0b010).
- Registers that have multiple iterations are denoted by {x..y} in their names; where *x* is first register and address, and *y* is the last register and address. For example, REG{0..1} indicates there are two versions of the register at different addresses: REG0 and REG1.

Symbols

Tip This symbol indicates a basic design concept or information considered helpful.

📝 This symbol indicates important configuration information or suggestions.

⚠ This symbol indicates procedures or operating levels that may result in misuse or damage to the device.

Document Status Information

- Advance – Contains information that is subject to change, and is available once prototypes are released to customers.
- Preliminary – Contains information about a product that is near production-ready, and is revised as required.
- Formal – Contains information about a final, customer-ready product, and is available once the product is released to production.

Revision History

May 12, 2010, Formal

This document fixed a number of minor typographical errors. No technical changes were made.

October 2009, Formal

This document was rebranded as IDT. No technical changes were made.

June 2009, Formal

There have been changes throughout the manual.

August 2007, Formal

There have been numerous edits throughout the manual. The formatting of the document has also been updated.

November 2002, Formal

This document information applies to both the Universe IIB and the Universe IID devices. The Universe IID is recommended for all new designs. For more information about the two devices, see the *Universe IID/IIB Differences Summary.*

The following chapter was updated for the release of this manual:

- "Reliability Prediction" on page 355

October 2002, Formal

This document information applies to both the Universe IIB and the Universe IID devices. The Universe IID is recommended for all new designs. For more information about the two devices, see the *Universe IID/IIB Differences Summary.*

There was an erratum found in the 361 DBGA package drawing.

1. Functional Overview

This chapter outlines the functionality of the Universe II. This chapter discusses the following topics:

- "Overview" on page 17
- "VMEbus Interface" on page 22
- "PCI Bus Interface" on page 22
- "Interrupter and Interrupt Handler" on page 23
- "DMA Controller" on page 24

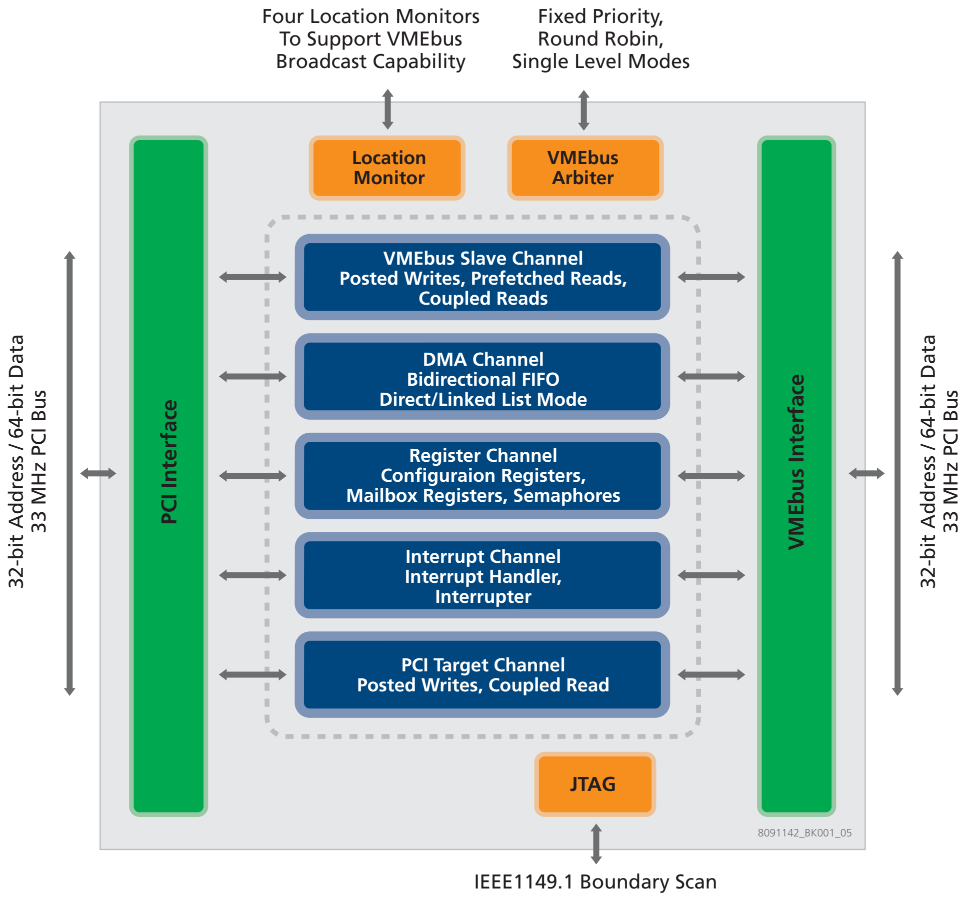
1.1 Overview

The IDT Universe II is the industry's leading high performance PCI-to-VMEbus interconnect. Universe II is fully compliant with the VME64 bus standard, and tailored for the next-generation of advanced PCI processors and peripherals. With a zero-wait state implementation, multi-beat transactions, and support for bus-parking, Universe II provides high performance on the PCI bus.

The Universe II eases development of multi-master, multi-processor architectures on VMEbus and PCI bus systems. The device is ideally suited for CPU boards functioning as both master and slave in the VMEbus system, and that require access to PCI systems. Bridging is accomplished through a decoupled architecture with independent FIFOs for inbound, outbound, and DMA traffic. With this architecture, throughput is maximized without sacrificing bandwidth on either bus.

With the Universe II, you know that as your system becomes more complex, you have proven silicon that continues to provide everything you need in a PCI-to-VME bridge.

Figure 1: Universe II Block Diagram



1.1.1 Universe II Features

The Universe II has the following features:

- Industry-proven, high performance 64-bit VMEbus interconnect
- Fully compliant, 32-bit or 64-bit, 33 MHz PCI bus interconnect
- Integral FIFOs for write posting to maximize bandwidth utilization
- Programmable DMA controller with Linked-List mode (Scatter/Gather) support
- Flexible interrupt logic
- Sustained transfer rates up to 60-70 Mbytes/s
- Extensive suite of VMEbus address and data transfer modes
- Automatic initialization for slave-only applications
- Flexible register set, programmable from both the PCI bus and VMEbus ports
- Full VMEbus system controller
- Support for RMWs, ADOH, PCI LOCK_ cycles, and semaphores

- Commercial, industrial, and extended temperature variants
- IEEE 1149.1 JTAG
- Available packaging:
  - 35mm x 35mm, 313-contact plastic BGA (PBGA) package

1.1.2 Universe II Benefits

The Universe II offers the following benefits to designers:

- Industry proven device
- Reliable customer support with experience in hundreds of customer designs

1.1.3 Universe II Typical Applications

The Universe II is targeted at today’s technology demands, such as the following:

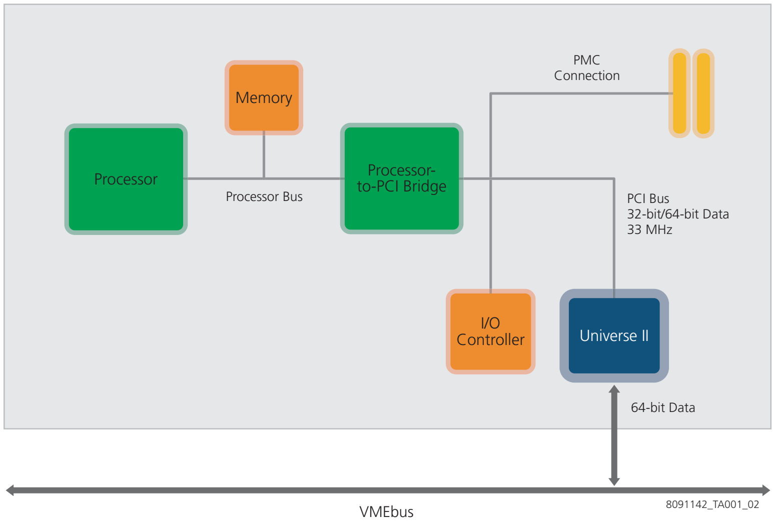
- Single-board computers
- Telecommunications equipment
- Test equipment
- Command and control systems
- Factory automation equipment
- Medical equipment
- Military
- Aerospace

1.1.3.1 Typical Application Example: Single Board Computers

The Universe II is widely used on VME-based Single Board Computers (SBC) that employ PCI as their local bus and VME as the backplane bus, as shown in the accompanying diagram. These SBC cards support a variety of applications including telecom, datacom, medical, industrial, and military equipment.

The Universe II high performance architecture seamlessly bridges the PCI and VME busses, and is the VME industry's standard for single board computer interconnect device.

Figure 2: Universe II In Single Board Computer Application



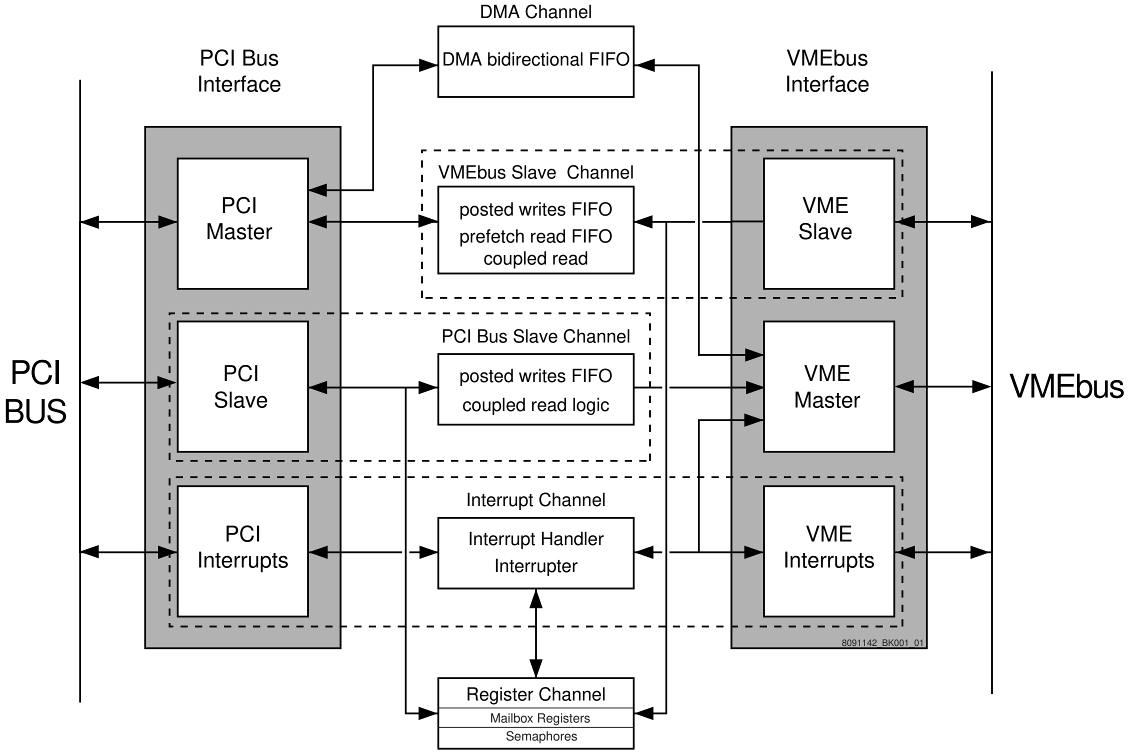
1.2 Main Interfaces

The Universe II has two main interfaces: the PCI Bus Interface and the VMEbus Interface. Each of the interfaces, VMEbus and PCI bus, there are three functionally distinct modules: master module, slave module, and interrupt module. These modules are connected to the different functional channels operating in the
Universe II. The device had the following channels:

- VMEbus Slave Channel
- PCI Bus Target Channel
- DMA Channel
- Interrupt Channel
- Register Channel

Figure 3 shows the Universe II in terms of the different modules and channels.

Figure 3: Universe II Data Flow Diagram



1.2.1 VMEbus Interface

The VME Interface is a *VME64 Specification* compliant interface.

1.2.1.1 Universe II as VMEbus Slave

The Universe II VMEbus Slave Channel accepts all of the addressing and data transfer modes documented in the *VME64 Specification* - except A64 and those intended to augment 3U applications. Incoming write transactions from the VMEbus can be treated as either coupled or posted, depending upon the programming of the VMEbus slave image (see “VME Slave Image Programming” on page 67). With posted write transactions, data is written to a Posted Write Receive FIFO (RXFIFO), and the VMEbus master receives data acknowledgment from the Universe II. Write data is transferred to the PCI resource from the RXFIFO without the involvement of the initiating VMEbus master (see “Posted Writes” on page 33 for a full explanation of this operation). With a coupled cycle, the VMEbus master only receives data acknowledgment when the transaction is complete on the PCI bus. This means that the VMEbus is unavailable to other masters while the PCI bus transaction is executed.

Read transactions may be either prefetched or coupled. A prefetched read is initiated when a VMEbus master requests a block read transaction (BLT or MBLT) and this mode is enabled. When the Universe II receives the block read request, it begins to fill its Read Data FIFO (RDFIFO) using burst transactions from the PCI resource. The initiating VMEbus master then acquires its block read data from the RDFIFO instead of directly from the PCI resources.

As VMEbus slave, the Universe II does not assert RETRY* as a termination of the transaction.

1.2.1.2 Universe II as VMEbus Master

The Universe II becomes VMEbus master when the VMEbus Master Interface is internally requested by the PCI Bus Target Channel, the DMA Channel, or the Interrupt Channel. The Interrupt Channel always has priority over the other two channels. Several mechanisms are available to configure the relative priority that the PCI Bus Target Channel and DMA Channel have over ownership of the VMEbus Master Interface.

The Universe II’s VMEbus Master Interface generates all of the addressing and data transfer modes documented in the *VME64 Specification* - except A64 and those intended to augment 3U applications. The Universe II is also compatible with all VMEbus modules conforming to pre-VME64 specifications.

As VMEbus master, the Universe II supports Read-Modify-Write (RMW), and Address-Only-with-Handshake (ADOH) but does not accept RETRY* as a termination from the VMEbus slave. The ADOH cycle is used to implement the VMEbus Lock command allowing a PCI master to lock VMEbus resources.

1.2.2 PCI Bus Interface

The PCI Interface is a *PCI 2.1 Specification* compliant interface

1.2.2.1 Universe II as PCI Target

Read transactions from the PCI bus are always processed as coupled transactions. Write transactions can be either coupled or posted, depending upon the setting of the PCI bus target image (see "PCI Bus Target Images" on page 70). With a posted write transaction, write data is written to a Posted Write Transmit FIFO (TXFIFO) and the PCI bus master receives data acknowledgment from the Universe II with zero wait-states. Meanwhile, the Universe II obtains the VMEbus and writes the data to the VMEbus resource independent of the initiating PCI master (see "Posted Writes" on page 60 for a full description of this operation).

The Universe II has a Special Cycle Generator that enables PCI masters to perform RMW and ADOH cycles. The Special Cycle Generator must be used in combination with a VMEbus ownership function to guarantee PCI masters exclusive access to VMEbus resources over several VMEbus transactions (see "Special Cycle Generator" on page 61 and "Using the VOWN bit" on page 64 for a full description of this functionality).

1.2.2.2 Universe II as PCI Master

The Universe II becomes PCI master when the PCI Master Interface is internally requested by the VMEbus Slave Channel or the DMA Channel. There are mechanisms provided which allow the user to configure the relative priority of the VMEbus Slave Channel and the DMA Channel.

1.2.3 Interrupter and Interrupt Handler

The Universe II has both interrupt generation and interrupt handling capability.

1.2.3.1 Interrupter

The Universe II Interrupt Channel provides a flexible scheme to map interrupts to the PCI bus or VMEbus Interface. Interrupts are generated from hardware or software sources (see "Interrupt Generation" on page 111 and "Interrupt Handling" on page 116 for a full description of hardware and software sources). Interrupt sources can be mapped to any of the PCI bus or VMEbus interrupt output pins. Interrupt sources mapped to VMEbus interrupts are generated on the VMEbus interrupt output pins VIRQ_ [7:1]. When a software and hardware source are assigned to the same VIRQ_ pin, the software source always has higher priority.

Interrupt sources mapped to PCI bus interrupts are generated on one of the INT_ [7:0] pins. To be fully PCI compliant, all interrupt sources must be routed to a single INT_ pin.

For VMEbus interrupt outputs, the Universe II interrupter supplies an 8-bit STATUS/ID to a VMEbus interrupt handler during the IACK cycle. The interrupter also generates an internal interrupt in this situation if the SW_IACK bit, in the PCI Interrupt Status (LINT_STAT) register, is set to 1 (see "VMEbus Interrupt Generation" on page 113).

Interrupts mapped to PCI bus outputs are serviced by the PCI interrupt controller. The CPU determines which interrupt sources are active by reading an interrupt status register in the Universe II. The source negates its interrupt when it has been serviced by the CPU (see "PCI Interrupt Generation" on page 111).

1.2.3.2 VMEbus Interrupt Handling

A VMEbus interrupt triggers the Universe II to generate a normal VMEbus IACK cycle and generate the specified interrupt output. When the IACK cycle is complete, the Universe II releases the VMEbus and the interrupt vector is read by the PCI resource servicing the interrupt output. Software interrupts are ROAK, while hardware, and internal interrupts are RORA.

1.2.4 DMA Controller

The Universe II has an internal DMA controller for high performance data transfer between the PCI and VMEbus. DMA operations between the source and destination bus are decoupled through the use of a single bidirectional FIFO (DMAFIFO). Parameters for the DMA transfer are software configurable in the Universe II registers (see "DMA Controller" on page 85).

The principal mechanism for DMA transfers is the same for operations in either direction (PCI-to-VMEbus, or VMEbus-to-PCI), only the relative identity of the source and destination bus changes. In a DMA transfer, the Universe II gains control of the source bus and reads data into its DMAFIFO. Following specific rules of DMAFIFO operation (see "FIFO Operation and Bus Ownership" on page 101), it then acquires the destination bus and writes data from its DMAFIFO.

The DMA controller can be programmed to perform multiple blocks of transfers using linked-list mode. The DMA works through the transfers in the linked-list following pointers at the end of each linked-list entry. Linked-list operation is initiated through a pointer in an internal Universe II register, but the linked-list itself resides in PCI bus memory.

2. VMEbus Interface

This chapter explains the operation of the VMEbus Interface.This chapter discusses the following topics:

2.1 Overview

The VMEbus Interface incorporates all operations associated with the VMEbus. This includes master and slave functions, VMEbus configuration and system controller functions.

2.2 VMEbus Requester

There are different channels in the Universe II which require the use of the VMEbus. They are referred to as VMEbus requesters and are described in the following sections.

2.2.1 Internal Arbitration for VMEbus Requests

Different internal channels within the Universe II require use of the VMEbus: the Interrupt Channel, the PCI Target Channel, and the DMA Channel. These three channels do not directly request the VMEbus, instead they compete internally for ownership of the VMEbus Master Interface.

2.2.1.1 Interrupt Channel

The Interrupt Channel always has the highest priority for access to the VMEbus Master Interface (see Figure 3 on page 21). The DMA and PCI Target Channel requests are handled in a fair manner. The channel awarded VMEbus mastership maintains ownership of the VMEbus until it is has completed the transaction. The definition of a complete transaction for each channel is in "VMEbus Release" on page 27.

The Interrupt Channel requests the VMEbus master when it detects an enabled VMEbus interrupt line asserted and must run an interrupt acknowledge cycle to acquire the STATUS/ID.