

# DN2000K10 ASIC Prototyping Engine Instruction Manual

<http://www.dinigroup.com>

Updated 2/5/01

This document describes the configuration and operation of the DN2000K10 series of logic emulation devices.

- 1.0 Device Overview
  - 1.1 Jumper Descriptions
  - 1.2 LED Descriptions
- 2.0 Programming Instructions
  - 2.1 JTAG Programming
  - 2.2 Serial Programming
  - 2.3 Flash Programming
  - 2.4 Programming Thru FPGA 'F'
- 3.0 Miscellaneous
  - 3.1 FPGA 'F' Revision Number
  - 3.2 Using Block Memory in FPGA 'F'



## 1.0 Device Overview

The DN2000K10 ASIC Prototyping Engine is a PCI-based logic emulation device available with one, four, or six Xilinx Virtex or Virtex-E field-programmable gate arrays (FPGA's). The device contains an 8 MB flash memory that can be used to store designs and load them into the FPGA's, two clock oscillator sockets with oscillators (1 and 14.318 MHz), and an array of headers that allow for signal probing and for stacking multiple DN2000K10's to increase the amount of logic available.

The DN2000K10 is currently available with the following FPGA's:

FPGA Model	Brand	Package	PCI Voltage
XCV1000	Xilinx Virtex	BG560	+5V/+3.3V
XCV1000E	Xilinx Virtex-E	BG560	+3.3V
XCV2000E	Xilinx Virtex-E	BG560	+3.3V

**Warning:** Installing a 3.3V device in a 5V PCI slot can damage your device, your computer, and you. All DN2000K10's are keyed to prevent improper installation; do not attempt to subvert this keying feature.

This manual contains basic operating instructions for this device. For more technical information, the schematic, net lists, and source code are available on the CD supplied with your DN2000K10.

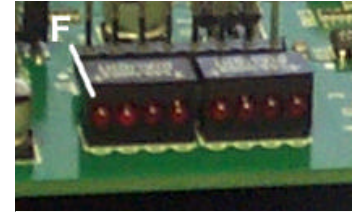
## 1.1 Jumper Descriptions

Jumper	Affects	Installed	Uninstalled	Default *	
J2	JP_PCI_CFG	Will configure FPGA_F (U10) only, first 1 MB of flash will be loaded into FPGA (depending on J4)	Will configure all FPGA's, lower 7 MB of flash will be loaded into FPGA's (depending on J4)	Install	
J4	JP_HDR_CFG	Uses the header for configuration (Xilinx Xchecker or parallel download cable)	Downloads from the onboard 8 MB flash.	Uninstall	
J1 pins 1, 2	Reset button	If board is configured, reset button asserts reset. If board is not configured, reset reconfigures board.	Reset reconfigures board.	Uninstall	
J1 pins 3, 4	unused				
J1 pins 5, 6	Polarity of FL4K_RESET_	Active high (always driving)	Active low (open drain output)	Uninstall	
JP27	NC/OE/OE_ selection for oscillators in X1, X2	NC/OE/OE_ pin connected to GND	NC/OE/OE_ pin connected to +5V	Uninstall	
JP18-JP23	CLKA select	These jumpers control the source of the clock signals for the FPGA's and CPLD. CLKA is required for proper operation of CPLD.		Install	
JP15-JP17	CLKB select			Install	
JP24, JP26	CLKCPLD select			Uninstall	
JP16, JP70	PCICLK settings	Always install JP70, uninstall JP16		Install JP70, uninstall JP16	
JP38, JP39, JP40	M66EN settings for PCI	Only install one: JP38 for +5V, 66 MHz capable; JP39 for +3.3V, 66 MHz capable; JP40 for non-66 MHz capable.		Install JP40	
JP36, JP37	Presence indicators for PCI	For no board, uninstall JP36 and JP37 For 25W max, uninstall JP36 and install JP37 For 15W max, install JP36 and uninstall JP37 For 7.5W max, install JP36 and JP37		Install JP37, JP36	
J9, J10, J11	M0, M1, M2 settings for FPGA programming mode	Unused, do not install		Uninstall	
JP4	Flash write enable	Writing to flash disabled	Writing to flash enabled	Uninstall	

\* Defaults denote settings required for loading and running a PCI target once it is loaded into the first MB of onboard flash.

## 1.2 LED Descriptions

With the LED's facing you and the FPGA side of the board pointing up, the left-most LED indicates whether FPGA\_F is programmed if J2 is installed. The second from the left LED indicates whether the other FPGA's are programmed. Off indicates programmed.



LED indicators

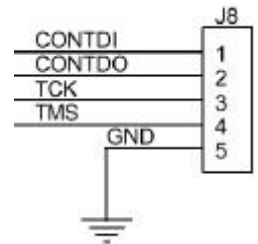
## 2.0 Programming Instructions

The FPGA's may be programmed using the onboard JTAG or serial interfaces as well as downloaded across the PCI bus into the onboard flash memory using the supplied AETest utility. Suggested tools are Xilinx's JTAG Programmer and Hardware Debugger.

Please note that boards with 4 or more XCV2000-E FPGA's may only be programmed using the JTAG method unless you are only programming the 'F' FPGA.

### 2.1 JTAG Programming (all board configurations)

1. Install J4 (and uninstall J2, if you want to program all FPGA's). Power down host system and power back up.
2. Access JTAG signals using J8 (not the labeled signals on J6, these are for programming the CPLD). See figure for pin locations on J8. If the board has XCV1000 FPGA's on it, use +5V on J6 for power. **If the board has XCV1000E or XCV2000E FPGA's, YOU MUST use JP8 pin 19 for power (+3.3V).**
3. Program using JTAG Programmer.



#### Hints for creating JTAG bit files:

Disable "Enable Internal Done Pipe" and use JTAG startup clock.

### 2.2 Serial Programming (all but 4 or 6 XCV2000E configurations)

#### For a single FPGA

1. Install J2 (JP\_PCI\_CFG) and J4 (JP\_HDR\_CFG)
2. Attach cable to serial interface on J6.
3. Use Hardware Debugger to download a .bit file.
4. Reset host system to reinitialize PCI configuration.

#### For multiple FPGA's

1. Uninstall J2 (JP\_PCI\_CFG) and install J4 (JP\_HDR\_CFG)
2. Attach cable to serial interface on J6.
3. Use Hardware Debugger to download an .exo file.
5. Reset host system to reinitialize PCI configuration.

### 2.3 Flash Programming (all but 4 or 6 XCV2000E configurations)

Warning: Using AETest to download a .bit file will result in the first 1 MB of flash being overwritten by that .bit file. The first 1 MB normally contains the default PCI target (fpga\_f1.bit, fpga\_f1e.bit)

or fpga\_f2e.bit, depending on FPGA model) that is written to FPGA 'F' when J4 is not installed. This PCI target, when loaded into 'F', allows AEttest to talk to the board. Overwriting this default target in the flash will prevent you from simply loading 'F' from flash memory to be able to run AEttest; instead you will need to program the appropriate PCI target using serial or JTAG methods described above. If you put your design in a .hex file, AEttest will not overwrite the first 1 MB of flash.

1. Install J2. Uninstall J4.
2. Press reset button on board. Wait until the 'F' LED goes out (see 1.2).
3. Reset host system to reinitialize PCI configuration. FPGA 'F' now has the default PCI target loaded.
4. Place a .bit or .hex file of your design in the directory that AEttest will be run from.
5. Run AEttest and download your design by selecting "Download a hexfile" from the Flash menu and following instructions. Your design is now contained in the upper 7 MB of flash.
6. Uninstall J2.
7. Press reset button on board. When the second LED goes out, your device is programmed.
8. Reset host system to reinitialize PCI configuration.

#### **Hints for creating .hex files:**

Use Xilinx's PROM File Formatter. Disable the "swap bits" option. Please note that you must have a design for each FPGA, even if it is blank. Add designs to the project in this order: F-A-E-B-D-C. Put all hexfiles in one daisy chain.

## **2.4 Programming Thru FPGA 'F' (all board configurations)**

FPGA's 'A' thru 'E' may be programmed using the standard PCI target in 'F' and the AEttest utility.

1. Make a hex file for A-E-B-D-C as you would for all 6 FPGA's and place it in the directory that AEttest will be run from.
2. Configure 'F' with the standard PCI target: Install J2 and uninstall J4. Press reset button on board. fpga\_f.bit will be loaded from flash into FPGA 'F'.
3. Reset host system to reinitialize PCI configuration.
4. Run AEttest and download your design by selecting "Download to FPGAs ABCDE from FPGA F" and select your design. When the second LED goes out, your device is programmed.

## **3.0 Miscellaneous**

### **3.1 FPGA 'F' Revision Number**

The 'F' design (fpga\_f?.bit) supplied with your board contains a revision number. From AEttest select menu option 0 to read the revision number of the PCI target. The revision number is stored in PCI BAR1, offset 0x00078000.

### **3.2 Using Block Memory in FPGA 'F'**

The 'F' design (fpga\_f?.bit) supplied with your board contains a block of memory. It is located at PCI BAR1, offset 0x00079000, and is dword (32 bit) accessible only. The memory is 32 bits

wide and 1024 entries deep. You may test the presence and functionality of this memory by running Aetest and selecting option 'n' from the memory menu.