Readme File: Version 1.0 – November 14, 2000

1 Overview

This CodeKit software contains a low-level Ethernet driver for the PCnet™-*FAST III* controller on the NetSC520 demonstration board. The driver is written for protected mode operation using a flat memory model and is written to run from the 32-bit CAD-UL XDBx86 debug environment. This driver can be built to run one Ethernet controller on the NetSC520 board or the Élan™SC520 microcontroller customer development platform (CDP). The code defaults to running a loopback test on the NetSC520 where 255 packets are transmitted and received.

IMPORTANT: By loading this software or any portion thereof, you agree to all the terms of the License Agreement in the file LICENSE.PDF. Do not use this software until you have carefully read and agreed to the terms and conditions in the Agreement. If you do not agree to the terms of the Agreement, do not use this software or any portion thereof.

For support: e-mail epd.support@amd.com
or call 1-800-222-9323

1.1 Tool and System Requirements

The following target hardware and tools are required to build and execute this CodeKit software.\

ITEM	DESCRIPTION
Target Hardware	NetSC520 Demonstration Board or ÉlanSC520 microcontroller CDP, revision 1.0 or revision 1.1
Monitor	CAD-UL XDBx86, version 411P (or newer), or REMON (CodeKit #2803)
Workstation Operating System	Microsoft® Windows® 95, 98, or NT
Tool Suite	CAD-UL Workbench version V120B
Compiler	CAD-UL CXX386 - ANSI C/C++ cross optimizing compiler, version V701D
Assembler	AS386 - CAD-UL cross assembler 80386/80486, version V302A
Linker	LINK386 - optimizing cross linker and system builder, version V312F

NOTE: Older versions of the CAD-UL tools may work, but were not tested on this CodeKit.

1.2 Installing the CodeKit Software

This CodeKit software is delivered as a self-extracting executable file. Double-click on the EXE file to create a directory called C:\AMD\CK005400. The CK005400 directory contains the entire contents of this CodeKit.

1.3 Files and Directories

The EXE file contains the following files in the following directories.

1.0.1 CK005400 Directory

FILE	DESCRIPTION
ENETTEST.OPW	The main project file for the CAD-UL Workbench.
LICENSE.PDF	Licensing agreement for the software contained in the CodeKit.
README.PDF	This document in Adobe Acrobat PDF.

1.0.2 V100\ENETTEST.BD Directory

FILE	DESCRIPTION
LINKER.MAP	Project map file.
README.TXT	Gives description of CAD-UL directory structure.

1.0.3 V100\DEBUG Directory

FILE	DESCRIPTION
ENETTEST.BD	Binding file used by the debugger for loading the code to RAM.
ENETTEST.XDB	Cross debugger info.
ENETTEST.ABS	The Ethernet test application's absolute located file which is loaded to the target by REMON.
ENETTEST.HX	Motorola S-Record hex version of the software that is loaded by the CAD-UL debugger.

1.0.4 V100\ENETTEST.BD\BLD Directory

FILE	DESCRIPTION
LINK.BLD	Linker command file for locating the code into RAM.

1.0.5 V100\ENETTEST.BD\SOURCE\H Directory

FILE	DESCRIPTION
ALIB.H	Header file for assembly procedures used by the C Library.
AMDRIVE.H	Common structures used in the Ethernet driver.
GLOBAL.H	Global definitions used throughout code for loopback mode and building for the NetSC520 board or the ÉlanSC520 microcontroller CDP.
LIBRARY.H	Header file for C Library functions.
MMCR.H	Definitions for ÉlanSC520 microcontroller MMCR registers.

FILE	DESCRIPTION
PCI.H	Definitions used in PCI.C.
PCN.H	More structures used by the Ethernet driver.
PCNDRV.H	Definitions used in PCNDRV.C
PNPAPPN.H	Definitions for PNPAPPN.C
SC520.H	More ÉlanSC520 microcontroller register definitions.
TYPES.H	Type definitions used throughout the program.

1.0.6 V100\ENETEST.BD\SOURCE\CLIBRARY Directory

FILE	DESCRIPTION
ALIB.ASM	Helper functions for the C Library used in this CodeKit.
LIB.C	C Library functions.
PRINTF.C	Simple implementation of the C Printf() function.
SERIALIO.C	Serial IO functions for the UARTs
SSCANF.C	Minimal SSCANF function implementation.
TIMER.C	Timer tick functions for the ÉlanSC520 microcontroller.

1.0.7 V100\ENETTEST.BD\SOURCE\ENETTEST

FILE	DESCRIPTION
IPSCAN.C	Code to scan for PCnet-*FAST III* Ethernet controller on PCI bus.
MAIN.C	Code to perform Ethernet loopback test.
PCI.C	Helper functions for performing PCI bus accesses.
PCNDRV.C	Driver for the PCnet-*FAST III* controller
REG.C	Functions for accessing registers on the PCnet-*FAST III* controller

1.0.8 V100\ENETTEST.BD\STARTUP Directory

FILE	DESCRIPTION
INI386.ASM	Code for setting up data and code segments in memory
RESET.ASM	Startup code for the ÉlanSC520 microcontroller
STARTUP.ASM	Call to ‘C’ Main() function made from here

1.0.9 XDB Directory

FILE	DESCRIPTION
ENETTEST.XDB	Contains cross-debugger information for this application.

1.4 Setting Up the Target Hardware

To run the demonstration software, you need an RJ-45 Ethernet loopback connector plugged into the onboard Ethernet connector on the NetSC520 board.

You will also need to have a Raven Wiggler connected to your NetSC520 board and have REMON (CodeKit #2803) running on a desktop machine.

1.5 Building the CodeKit Software

To build the software, open the project by double clicking on the ENETTEST.OPW file in the CK005400 directory. This launches the CAD-UL workbench. From here, use the compile options supported by the CAD-UL workbench.

NOTE: If you intend to run this code from the CAD-UL debugger then you need to comment out the '#define REMON_SERIAL' line at the top of the LIBRARY.H file to redirect the program output to UART 2 which is connected to the serial connector closest to the Ethernet connector on the NetSC520 board. You will also have to have a terminal program like Hyperterm setup in ASCII mode at 115200 bit/s to see the output from the NetSC520 board.

1.6 Loading the Code with REMON

This is the way this CodeKit is intended to be used. Get REMON running on the target system and make sure you have the ENETTEST.ABS file copied to the same directory as REMON resides in and enter the following sequence of REMON commands to load and execute the code.

```
Z

Yn

L ENETTEST.ABS

GS
```

This sequence resets the NetSC520 board, initializes its memory, loads the ENETTEST.ABS program to memory, and executes the code while having the program IO redirected to the AMDebug™ technology virtual UART (the program output will show up in the REMON debug console on your desktop machine).

1.7 Loading the Code using the CAD-UL Debugger

From the CAD-UL workbench, you can click the debugger icon, which looks like a magnifying glass, to launch the debugger. When the monitor launches, it indicates if a successful link has been established with the NetSC520 board. You can click the load icon to bring up the load menu. Click the download box to select it, then click OK to load your software to the NetSC520 board over the Raven connection. If you don't select the download box (i.e., leave it empty), no code is loaded to the NetSC520 board. When the code is loaded, you can click on the run icon, which looks like a person running, to execute the program.

2 About the Software

The software is designed to support the PCnet-*FAST III* Ethernet controller on the NetSC520 board. When the code is executed, it will display the base address used for IO to the Ethernet controller and information about which IRQ the controller is using and the Ethernet controller's chip identification number. The user will be prompted to hit any key to execute the loopback test. Before hitting a key, make sure that a RJ-45 loopback plug is plugged into the NetSC520 board's Ethernet connector. When you hit a key, the program will attempt to transmit and receive 255 1500-byte IPV4 packets to itself. If the loopback is successful the program will printout that 255 packets were transmitted and 255 packets were received. If a failure occurs, the software will indicate how many packets were lost or corrupted during the loopback.

If the EEPROMs for your Ethernet controller(s) are not programmed, the media access code (MAC) address of your controller will be 00 01 02 03 40 50. If the software fails, it may be because of a improperly programmed EEPROM on your NetSC520 board. If you believe this is the case, download CodeKit #4700 which is a utility for programming EEPROMs on the NetSC520 board to reburn the EEPROM.

3 How the Software Works

The MAIN() function includes steps to initialize and use the driver. The first step is to call IPSCAN() to scan the PCI bus for PCnet Ethernet controllers and store the number and type of controllers in the PCNINFO data structure. This structure is passed to PCNINIT() in PCNDRV.C, where all the controllers are initialized.

The next step is to call PCNOUTPUT() to send packets or PCNRPROCESS() to check if any packets were received. This process is repeated 255 times and the results are saved and displayed when the program it done.

For more information about the PCnet-*FAST III* Ethernet controller, refer to the *Am79C973/Am79C975 PCnet*™-FAST *III Single-Chip 10/100 Mbps PCI Ethernet Controller with Integrated PHY Data Shee*t, order #21510.

NOTE: More information about the code is supplied in the inline documentation.

3.1 Tweaking the Code

Global.H contains some definitions that can affect how the code runs. These definitions are described below.

NETSC520 - Build the code to run on the NetSC520 board, this is set by default, and if the comment is removed, the code will be built for the ÉlanSC520 microcontroller CDP.

FORCE_IT - This definition will force an Ethernet controller to transmit, even if the controller does not have a loopback connector plugged into it.

LOOPBACK - This definition places the controller(s) in loopback mode.

MAX_SPEED - This definition places the software into don't-wait-for-interrupt mode. In this mode, after transmitting a packet, the controllers do not wait for a transmit interrupt to occur before allowing code execution to continue. This mode can result in your transmit buffers being filled continuously for maximum throughput, and may prevent code hangs from an unsuccessful transmit attempt; however, this mode can also cause loss of data.