Altera DE2:
DM9000A Ethernet Controller
Application Notes

By: Brent Erickson
Mavis Chan
Sydney Bitner

Table of Contents

Introduction

This app note provides revisions from the 2013 Ethernet App Notes by Tarek Kaddoura and Jigar Nahar.
The Ethernet controller on the DE2 board is a Davicom DM9000A Fast Ethernet Controller. The controller consists of a 16 KB SRAM, a MAC unit, a 10/100 PHY transceiver, and an interface to communicate with it. This document will explain the steps necessary to add the DM9000A to the hardware configuration, and the drivers needed to get it working.

Hardware Configuration

In order to simplify the hardware configuration process, Terasic has provided the HDL and TCL files necessary to add the component to a system through SOPC builder. These files can be downloaded from
https://www.ualberta.ca/~delliott/local/ece492/appnotes/2013w/Ethernet_DM9000A/DM9000A.tar

QSys System

The Ethernet controller will need to be supplied with a 25 MHz clock. This can be accomplished by creating a new clock, or dividing the system clock by 2. The system clock is assumed to run at 50 MHz. The second approach will be taken here by using an ALTPLL.
Note: The following steps only show what needs to be added to get the DM9000A working. They assume that a NIOSII processor and other components (such as RAM) have already been added to the system.

1. In the downloaded file, there is a folder named hardware. Copy the contents of this folder to the root of your Quartus project.
2. In Quartus, go to Tools → QSys.
3. Add an Avalon ALTPLL if one doesn't already exist.
4. Create a clock of 25MHz and export the signal.
5. In the component list in QSys, there should be a new category: “Terasic Technologies Inc.”. Under this category is the DM9000A component. Add this component to the system.
6. Export the conduit signal under the DM9000A component.
7. Connect the Component as shown in Figure 1 below
8. Generate the Qsys system.

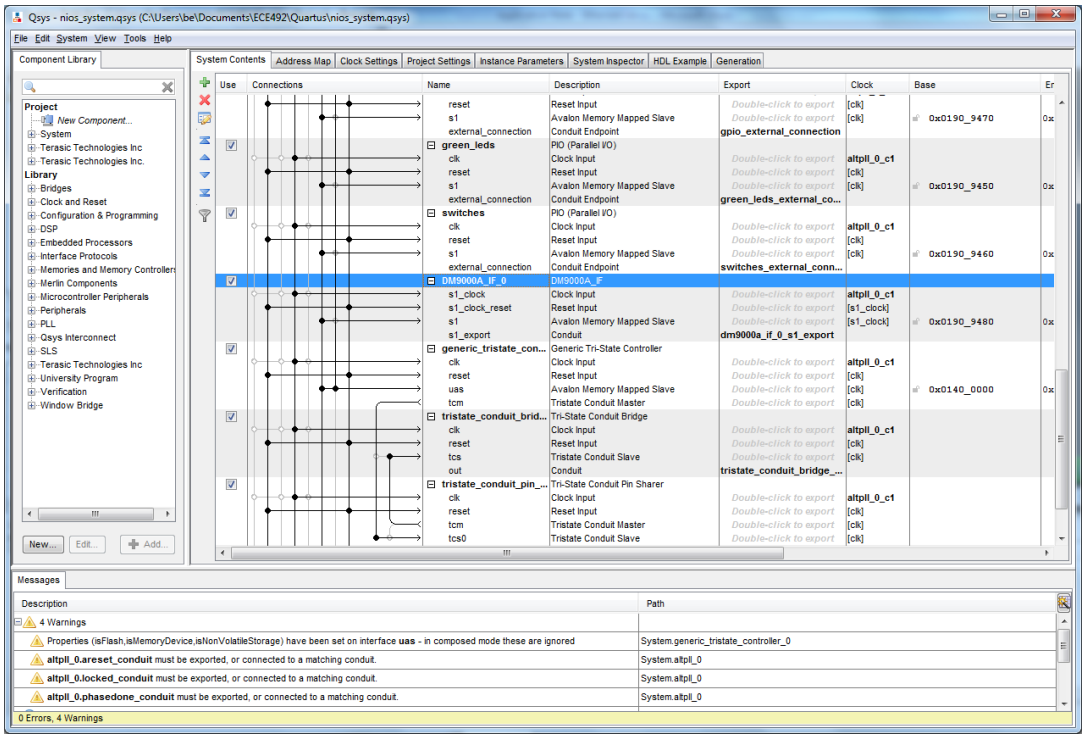

Figure 1 Qsys System

Top Level Modifications

At this point, the system contains the DM9000A component. Now, the component needs to be connected through the top level HDL file. Also, the Ethernet clock needs to be connected to the 25 MHz PLL clock created above.

Note: The following instructions assume that the DM9000A component was named "DM9000A_IF_0" in QSys Builder and the 25 MHz PLL clock is called atpll_0_c2. Also, all necessary code is shown in VHDL.

1. Add the pins necessary in your top level entity.

```
-- Ethernet
ENET_CLK        :       out std_logic;
ENET_CMD        :       out std_logic;
ENET_CS_N       :       out std_logic;
ENET_INT        :       in std_logic;
ENET_RD_N       :       out std_logic;
ENET_WR_N       :       out std_logic;
ENET_RST_N      :       out std_logic;
ENET_DATA       :       inout std_logic_vector(15 downto 0);
```

2. Add the necessary signals into your system component (assuming the DM9000A was called DM9000A_0 in SOPC Builder).

```
dm9000a_if_0_s1_export_DATA     : inout std_logic_vector(15 downto 0) := (others => 'X'); -- DATA
 dm9000a_if_0_s1_export_CMD     : out   std_logic;                                        -- CMD
```

dm9000a_if_0_s1_export_RD_N : out std_logic; -- RD_N
dm9000a_if_0_s1_export_WR_N : out std_logic; -- WR_N
dm9000a_if_0_s1_export_CS_N : out std_logic; -- CS_N
dm9000a_if_0_s1_export_RST_N : out std_logic; -- RST_N
dm9000a_if_0_s1_export_INT : in std_logic := 'X'; -- INT
dm9000a_if_0_s1_export_CLK : out std_logic;

3. Add the pll clock to the top level created above.
altpll_0_c2_clk : out std_logic; -- clk

4. Connect the pins as shown below in the VHDL code.
altpll_0_c2_clk => ENET_CLK, -- altpll_0_c2.clk
dm9000a_if_0_s1_export_DATA => ENET_DATA -- dm9000a_if_0_s1_export.DATA
dm9000a_if_0_s1_export_CMD => ENET_CMD, -- .CMD
dm9000a_if_0_s1_export_RD_N => ENET_RD_N, -- .RD_N
dm9000a_if_0_s1_export_WR_N => ENET_WR_N, -- .WR_N
dm9000a_if_0_s1_export_CS_N => ENET_CS_N, -- .CS_N
dm9000a_if_0_s1_export_RST_N => ENET_RST_N, -- .RST_N
dm9000a_if_0_s1_export_INT => ENET_INT, -- .INT

NOTE: dm9000a_if_0_s1_export_CLK is left unconnected

The DM9000A component should now be connected. The final step is to program the board with the new configuration.

Software Configuration

After adding the component to the system, the software drivers are not added automatically in the NIOS II IDE. The following additions and initializations need to be done in software.
1. Add the files from the "software" folder of the downloaded tar file to the root of your software project.

2. Open the dm9000a.h file and add these lines:

```
#define DM9000A_0_BASE DM9000A_IF_0_BASE
#define DM9000A_0_IRQ DM9000A_IF_0_IRQ
#define DM9000A_0_NAME DM9000A_IF_0_NAME
```

3. In web_server.c, define the green leds base and lcd display name if they are different than the default ones in the file.
*#define LED_PIO_BASE GREEN_LEDS_BASE*3
#define LCD_DISPLAY_NAME LCD_0_NAME

4. Define the flash name and base in network_utilities.c. Add the following lines:
#define EXT_FLASH_NAME CFI_FLASH_0_NAME
#define EXT_FLASH_BASE CFI_FLASH_0_BASE

5. Initialize the controller by adding the following lines in the main() function in web_server.c: (Add dm9000a.h as the first include-file at the top of web_server.c)

DM9000A_INSTANCE(DM9000A_0, dm9000a_0);
DM9000A_INIT(DM9000A_0, dm9000a_0);

Application Example: Web Server

Altera has provided an example web server that is really easy to set up. Running the web server will confirm that the DM9000A is indeed working correctly.
In order for the web server to work, the system needs be non-volatile (contains tri-state controller, conduit bridge and conduit pin sharer). The web server's files will be stored in this flash memory. The IP address will be displayed via serial once connected successfully. Finally, the board needs to be connected to a router through an Ethernet cable, or to a computer using a crossover Ethernet cable. First, open the NIOS II IDE and follow the instructions below:

1. Go to File → New Project → Web Server.
2. Apply the software modifications mentioned in Software Configuration.
3. Build the web server project.
4. Make sure the board is connected through Ethernet and run the project on the board.
5. Monitor the NIOS II console.

Now that the board is connected and the IP is known, the board's web site can be accessed by accessing the IP through a browser. A simpler test can be performed by simply pinging the IP of the board from a computer.

Issues and Notes

Running Off Flash

Running the web server off SDRAM works very nicely: the DM9000A is detected, the MAC Address is read from the CFI Flash, and the IP Address shows up on the LCD.
Unfortunately, trying to run the software off of flash causes complications. The DM9000A is still detected, but the MAC address is not read correctly from the flash. This stalls the program and never displays the IP.

Further Reading

The web server and simple socket server provided by Altera are extremely good examples on how to use the Ethernet controller as well as how to set up a whole web server. The Altera directory provided on the lab computers has a DE2 demonstration named SE2_NIOS_HOST_MOUSE_VGA that utilizes the Ethernet.

Note: The demonstration projects provided on the University labs are quite old, and will likely lead to hardware or other compilation errors. Newer versions of the demonstrations updated to support Quartus II 10.0 are the DE2_70 demonstrations. These demonstrations are not for the Cylone II, but

they contain excellent examples of updated code and drivers for the newer Quartus. These demonstrations can be downloaded from here: http://www.terasic.com/downloads/cd-rom/de2_70/DE2_70_demonstrations_V10.rar

The above rar file contains three projects that utilize the Ethernet: DE2_70_NET, DE2_70_NIOS_HOST_MOUSE_VGA and DE2_70_SD_Card_Audio_Player.

References

Previous application notes from winter2012 for running a web server:
http://www.ece.ualberta.ca/~elliott/ece492/appnotes/2012w/Webserver/

DM9000A Application note from 2013:
https://www.ualberta.ca/~delliott/local/ece492/appnotes/2013w/Ethernet_DM9000A/
Non-Volatile Lab Tutorial

Datasheet for the DM9000A:
http://www.cs.columbia.edu/~%20sedwards/classes/2013/4840/Davicom-DM9000A-Ethernet-controller.pdf

Author Information

Based on application notes from 2013 Ethernet App Notes by Tarek Kaddoura and Jigar Nahar. Compiled by Group 9: Brent Erickson, Mavis Chan & Sydney Bitner.