M7A Ethernet MAC
Demo Example

User Guide

02/2014

Capital Microelectronics, Inc.
China

Contents

1 Introduction

This document describes a demo example exploiting M7A's internal Ethernet MAC hard IP. The detail of the design is as below.

- Function
  - Send out an ARP packet via 100Mbps MII interface after initialization
  - Receive Ethernet MAC frames(only broadcast frames and frames whose destination address matches with the Ethernet MAC's predefined MAC address 00-1E-C9-3B-11-F8 can be received) and send to PC via UART
- The UART is configured as below
  - 115200bps
  - 8 bits
  - No parity check
  - 1 stop bit
- MCU works at 200MHz
- Third-party software used in demo example
  - IPAnalyse(also called iptool, version: 1.2), capture frames transferred(involve transmitted and received)
  - CommView(version: 6.0), generate any frame in compliance to Ethernet specification
- Device: CME-M7A
- PHY: Marvell 88E1111-RCJ

2 Demo Design Introduction

2.1 Hardware Level Structure

Following is hardware block diagram of the example design.

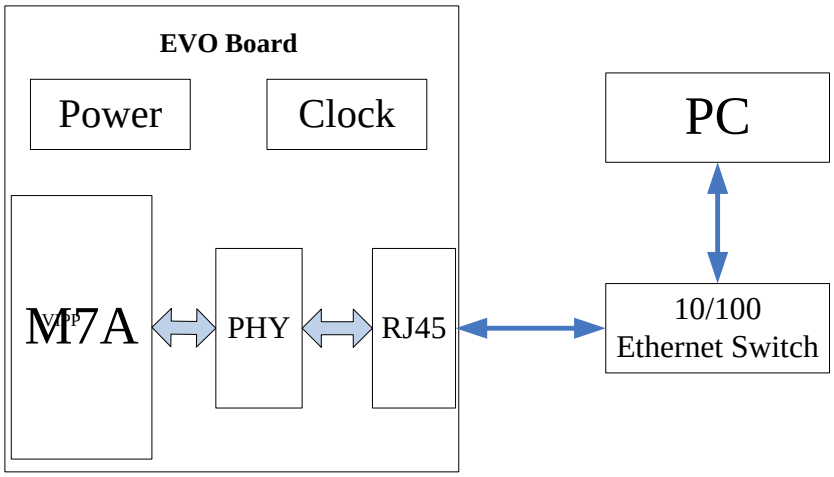

Figure 2-1 ***Hardware Block Diagram of the Example Design***

As shown in figure 2-1, there are three parts in the test environment, M7A evaluation board, 10/100 Ethernet switch and a PC. For evaluation board, the dedicated Ethernet MAC pins are connected to PHY via MII/GMII interface and the PHY transmits to and received from external device via RJ45. Two category 5 twisted pair cables are used, one connecting the evaluation board and the switch, one connecting the PC and the switch.

2.2 Detailed Design Introduction

2.2.1 Ethernet MAC Instance

For M7A, to utilize Ethernet MAC hard IP, users have to make use of ARM. The flow of adding ARM to the project is as below.

1) In Primace wizard, select ARM Cortex-M3.

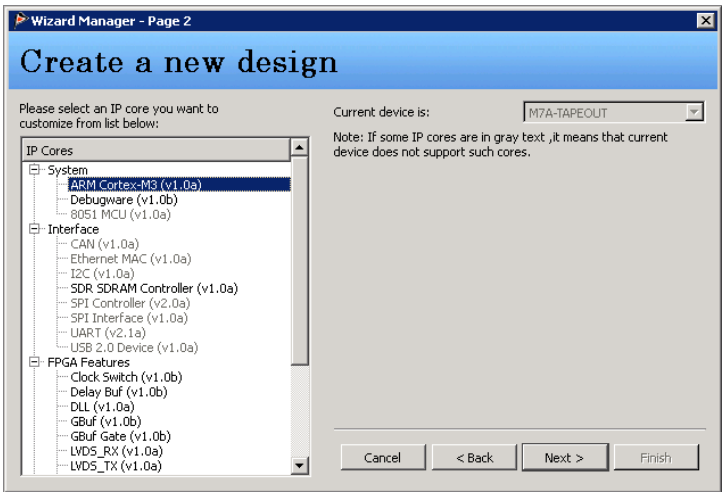

Figure 2-2 ***Phase 1 of Configuring ARM with Ethernet MAC***

2) Input the ARM module name and select its location.

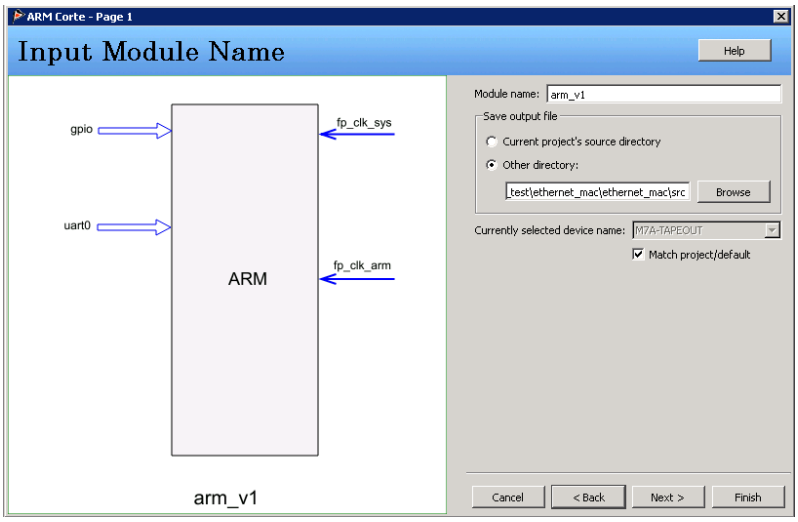

Figure 2-3 ***Phase 2 of Configuring ARM with Ethernet MAC***

3) Choose hard IPs used in user design(GPIO and UART2 are utilized for test).

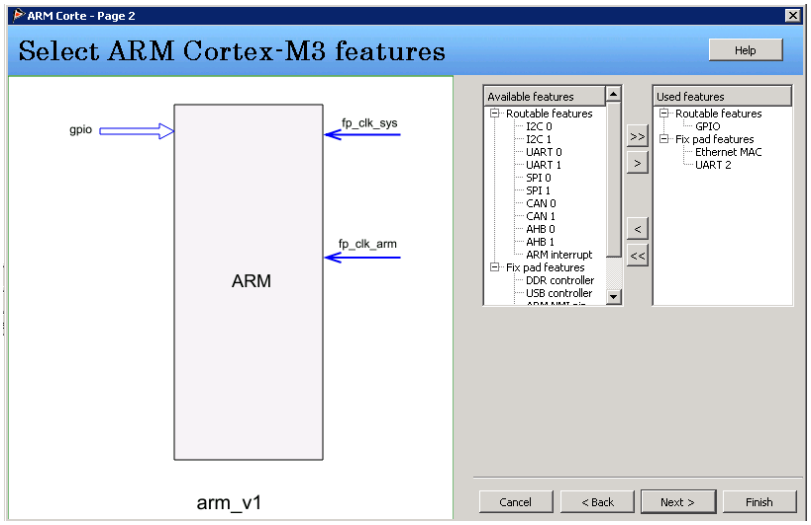

Figure 2-4 ***Phase 3 of Configuring ARM with Ethernet MAC***

4) Choose system clock frequency(in demo design, separated ARM clock is used).

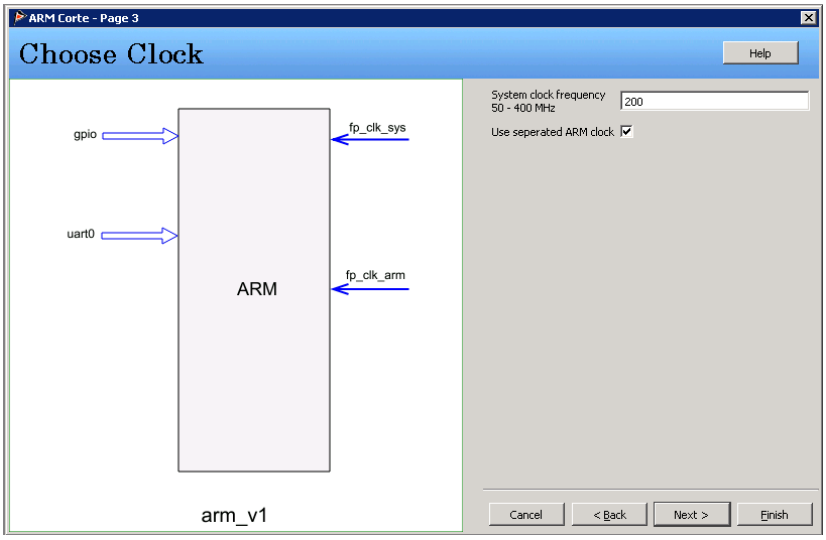

Figure 2-5 ***Phase 4 of Configuring ARM with Ethernet MAC***

5) Add related hex file to ARM.

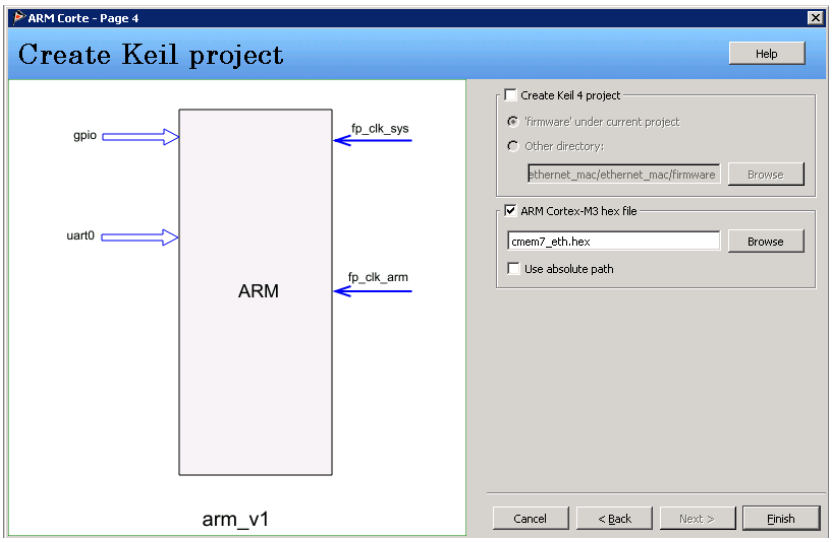

Figure 2-6 ***Phase 5 of Configuring ARM with Ethernet MAC***

2.2.2 Software Flow and Result

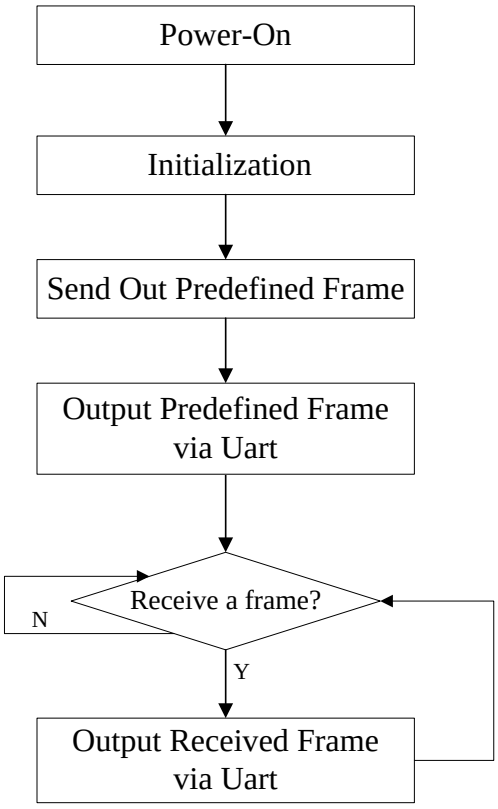

Figure 2-7 ***Demo Design Flow Chart***

As shown in figure 2-7, in this demo design,

1) After power-on, the evaluation board gets into initialization process. During this process, the PHY auto-negotiates with that of Ethernet switch or PC and as the result ARM gets the status of link speed and duplex status(for demo design, the status is 100Mbps, full-duplex). After that, ARM configures Ethernet MAC's internal registers and UART's parameters.
2) After initialization and configuration, M7A begins to transmit a predefined frame for one time. At the same time, the predefined frame is sent to PC via UART. It is seen in figure 2-8.

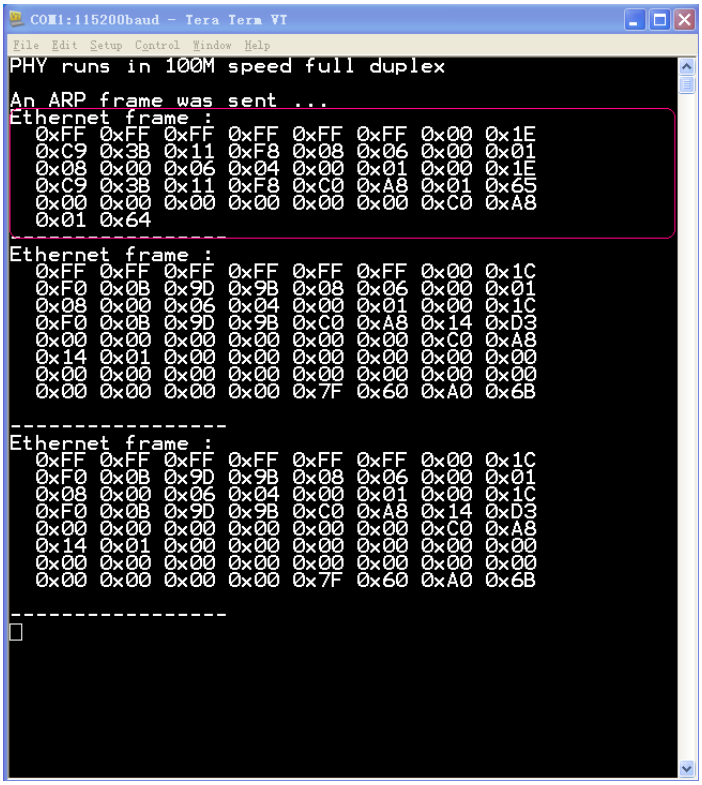

Figure 2-8 ***Predefined Frame Received via UART***

Also, with iptool(Ethernet frame capture tool), the frame can be captured as shown in figure 2-9.

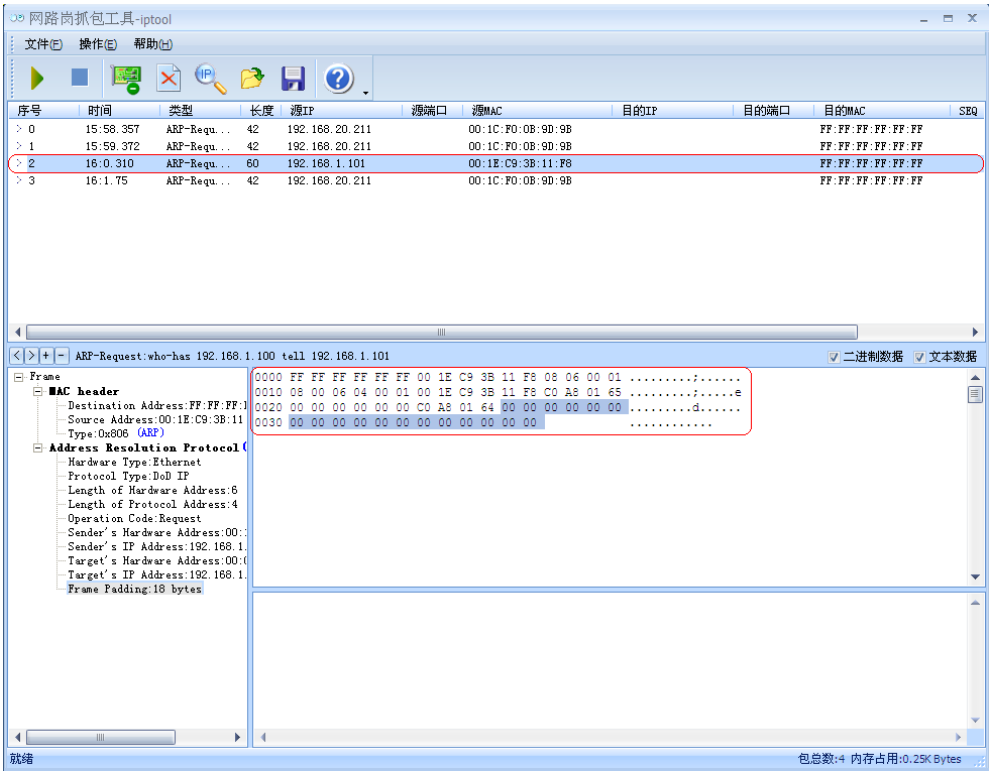

Figure 2-9 ***Predefined Frame Captured by iptool***

3) ARM continuously checks whether there is a frame received(in the demo design, only broadcast frames and frames whose destination address matches with the Ethernet MAC's predefined

MAC address 00-1E-C9-3B-11-F8 can be received). If a frame is received, ARM transmits it to PC via UART which is shown in figure 2-10.

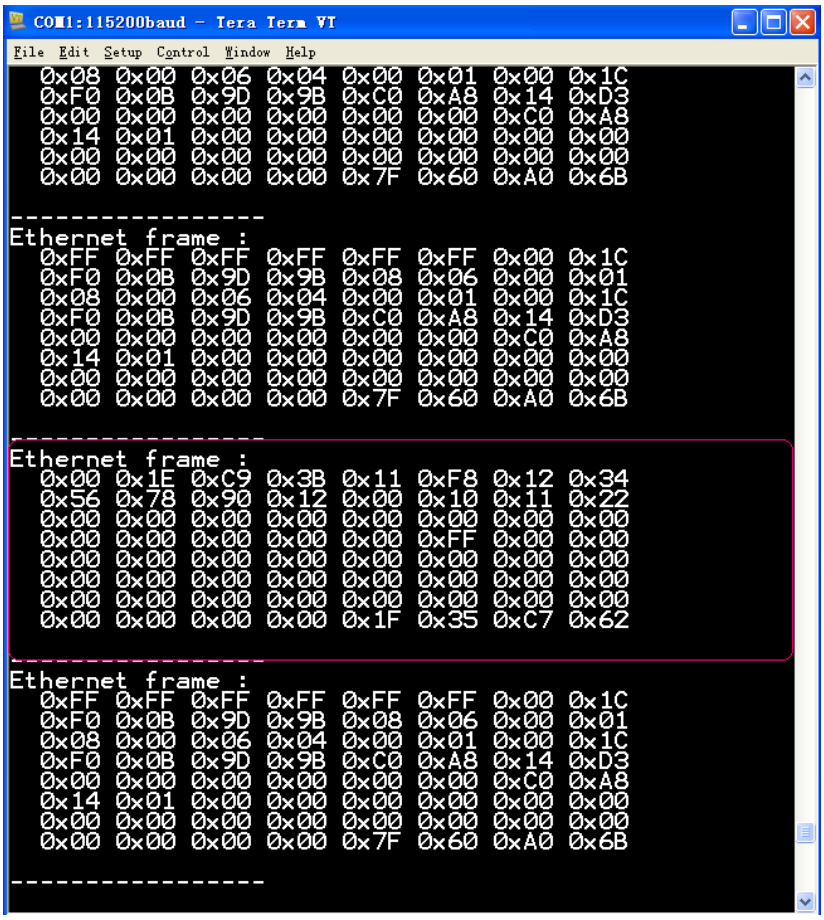

Figure 2-10 ***Predefined Frame Received via UART***

In figure 2-10, the frame highlighted with red rectangle is from PC generated by CommView(virtual Ethernet packet generation tool). And those frames with "FF-FF-FF-FF-FF-FF" as head are broadcast frames.

3 Pin and Design Source Description

3.1 Pin Description

*Table 3-1 **Top Module Pin Description***

Name	Direction	Width	Description
clk_20	Input	1	Clock input (20MHz)
gpio0	Inout	8	GPIO inout signal(only output direction is used), for test
gpio_in	Input	8	GPIO input signal, for test

Note: gpio0 and gpio_in in design are for test and they are optional.

3.2 Pin Assignment

The following figure shows the detailed pin assignments in IO Editor of Primace. For Ethernet MAC hard IP, pins are dedicated and their location are fixed. Primace will automatically assign the Ethernet MAC IOs(as seen in the following figure, the “Port Name” item that is gray is Ethernet MAC IO), users just need to assign available pins for other user IO.

	Port Name	Location	Pin Name	Direction	Operation Mode
15	gpio_in[5]	P6	IO11_2	input	normal
16	gpio_in[6]	P5	IO12_2	input	normal
17	gpio_in[7]	R6	IO13_2	input	normal
18	ETH_CLK_RX_I	AB9	IO_07N_ETH_CLK_RX_I_4	input	dedicated
19	ETH_CLK_TX_I	W12	IO_13P_ETH_CLK_TX_I_4	input	dedicated
20	ETH_GMII_MDC_O	AB11	IO_12N_ETH_GMII_MDC_O_4	output	dedicated
21	ETH_GMII_MDO_IO	AB8	IO_07P_ETH_GMII_MDO_IO_4	inout	dedicated
22	ETH_PHY_COL_I	Y12	IO_13N_ETH_PHY_COL_I_4	input	dedicated
23	ETH_PHY_CRS_I	AB5	IO_03N_ETH_PHY_CRS_I_4	input	dedicated
24	ETH_PHY_INTF_SEL_I1	AA8	IO_06P_ETH_PHY_INTF_SEL_I1_4	input	dedicated
25	ETH_PHY_INTF_SEL_I2	Y8	IO_05N_ETH_PHY_INTF_SEL_I2_4	input	dedicated
26	ETH_PHY_RXDV_I	AA11	IO_12P_ETH_PHY_RXDV_I_4	input	dedicated
27	ETH_PHY_RXD_I0	Y11	IO_11N_ETH_PHY_RXD_I0_4	input	dedicated
28	ETH_PHY_RXD_I1	W11	IO_11P_ETH_PHY_RXD_I1_4	input	dedicated
29	ETH_PHY_RXD_I2	AB10	IO_10N_ETH_PHY_RXD_I2_4	input	dedicated
30	ETH_PHY_RXD_I3	AA10	IO_10P_ETH_PHY_RXD_I3_4	input	dedicated
31	ETH_PHY_RXD_I4	AA6	IO_03P_ETH_PHY_RXD_I4_4	input	dedicated
32	ETH_PHY_RXD_I5	AB4	IO_02N_ETH_PHY_RXD_I5_4	input	dedicated
33	ETH_PHY_RXD_I6	W8	IO_05P_ETH_PHY_RXD_I6_4	input	dedicated
34	ETH_PHY_RXD_I7	AB6	IO_04N_ETH_PHY_RXD_I7_4	input	dedicated
35	ETH_PHY_RXER_I	AA7	IO_04P_ETH_PHY_RXER_I_4	input	dedicated

*Figure 3-1 **IO Assignment in Primace’s IO Editor***

3.3 Design Source

The Ethernet MAC demo example RTL source files and related firmware are shown in the following table 3-2.

Table 3-2 ***Source Files Description***

File	Description
RTL ./src/	ethernet_mac.zip/Ethernet_mac/src/
/ethernet_mac_top.v	The top module, implements PLL and ARM
/arm_v1.v	The ARM processor core implemented by ARM Wizard
/pll_v1.v	Phase-locked loop, implemented by PLL Wizard
Firmware ./user/	ethernet_mac.zip/Ethernet_mac/firmware/cmem7.uvproj
/main.c	Main function
/app_bufferpool.c	Create a buffer pool to deal with multiple frames received
/app_bufferqueue.c	Deal with multiple frames received in time sequence
/app_phy.c	Configure phy register and monitor phy's work status
/cmem7_extern_it.c	Interrupt functions for DMA

Revision History

Revision	Date	Comments
1.0	2014-02-14	Initial release