International Journal of Smart Home
Vol.8, No.5 (2014), pp.95-102
http://dx.doi.org/10.14257/ijsh.2014.8.5.09

Ethernet Controller Module Design based on ARM Technology

Zheng Peng[1], Zhang Zhaochen[1] and Peng Yanjun[2]

[1]*Taishan Medical University, Taian 271016, China;*
[2]*Shandong University of Science and Technology, QingDao 2665906, China)*

Abstract

With the rapid development of ARM technology, its application is also involved in the project to all spheres of society. This paper designs an Ethernet controller module based on ARM technology, through the actual verification, its operation is stable and can be very convenient to realize embedded system networking; at the same time the system communication and debugging fast, reliable, has the very high real-time performance.

Keywords: *The ARM technology, Ethernet controller, Module design*

1. Introduction

With the rapid development of science and technology, Internet has permeated all aspects. In embedded systems, and network has become the necessity of developing embedded systems. In the ARM system, Ethernet interface (Ethernet Port) is to communicate with the remote machine and debug, the basis of extended Ethernet front-end ports module can make internal communications between LAN and the Internet. System without Ethernet interface based on ARM and its application value will be discounted. Therefore, in terms of the whole system, Ethernet interface circuit should be necessary, but at the same time also is relatively complex.

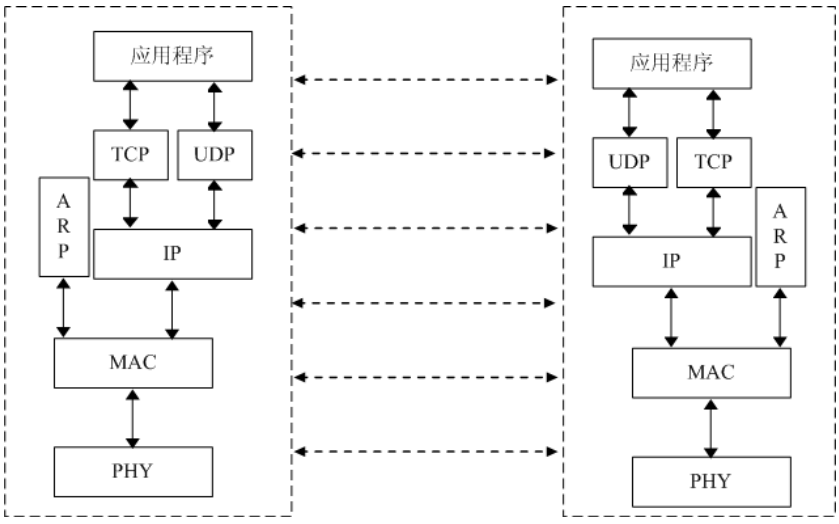

Figure 1. 802.3 Model Structure between the Layers

From the hardware point of view, as shown in Figure 1 of the ITU, T802.3 model structure between the layers, Ethernet interface circuit is mainly composed of medium access control MAC controller and Physical Layer interface, Physical Layer, PHY)

ISSN: 1975-4094 IJSH

composed of two parts. At present common Ethernet interface chip, such as the RTL8019/8029/8039, CS8900, DM9008 and DWL650 wireless network card and so on, its internal structure is mainly include the two parts. The module selects a specially designed for the third generation of fast Ethernet connection and RTL8019AS 10 m / 100 MBPS compatible with Ethernet interface chip, it supports a variety of embedded processor chip, embedded FIFO buffer is used to send and receive data.

2. Ethernet Front-end Ports Working Principle

Most ARM embedded an Ethernet controller, Interface to support Independent Media (Media Independent Interface, MII) and with the DMA buffer Interface (Buffered DMA Interface, BDI), can be in half-duplex and full-duplex mode provides 10m/100Mbps Ethernet connection. In half duplex mode, the controller support agreement CSMA/CD protocol; in full duplex mode, supporting the IEEE802.3 MAC layer protocol control.

Therefore, ARM inside actually contains the Ethernet MAC control, but did not provide the physical layer interface, therefore, need a physical layer chip converter to provide Ethernet access channel. And the commonly used high-speed stand-up 10m/100Mbps Ethernet physical layer interface devices provide MII interface and traditional wire network interface, can easily interface with ARM. Ethernet physical layer interface devices main function generally includes: physical coding sublayer, physical media attachment, twisted pair 10 base - physical media layer, TX encode/decoder and twisted-pair media access unit, *etc*.

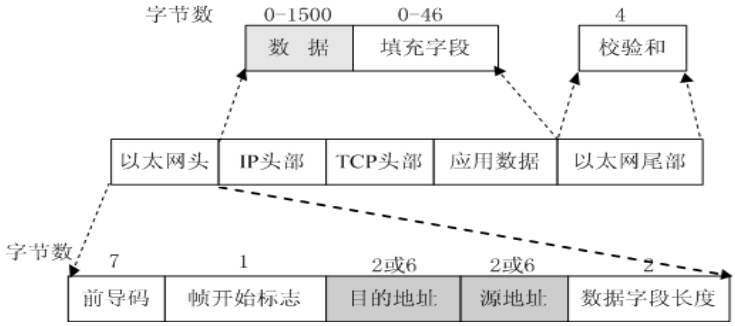

Figure 2. 802.3 Frame Format

Using RTL8019AS as the Ethernet physical layer interface, its basic working principle is: after receiving datagram from all parts of the host (from the destination address field into the data fields, as shown in figure 2), listen for network lines. If the line is busy, it will wait until the line idle, otherwise, immediately send the data frames. In the process of sending, it add the Ethernet frame header (including leading field and frame start sign), and then generate CRC check code, finally send the data frames to the Ethernet.

In the process of receiving, it will be received from the Ethernet data frame after the decoding, to frame the head and address inspection steps after cached on chip. After the CRC check through, it will according to the initialization configuration situation, notify RTL8019AS received data frame, in the end, with some kind of transport mode (I/O mode, the Memory mode, the DMA mode) to the ARM of the store.

3. The Hardware Circuit Design

3.1. The Principle Diagram of the Circuit

With Ethernet controller RTL8019AS chip design related circuit, the system can through the RJ - 45, connected to the Ethernet network communication part of its block diagram as shown in Figure 3.

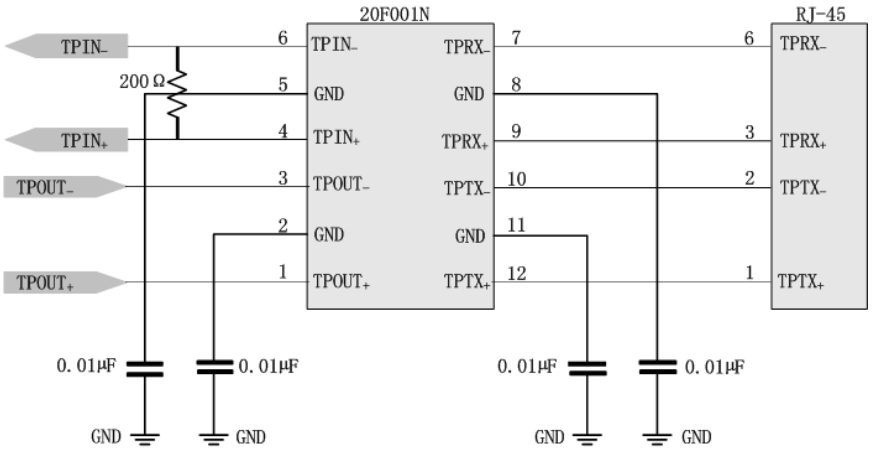

Figure 3. Ethernet Controller Module (RJ 45 I/O)

RTL8019AS IOS pins with the base address of the corresponding relation, as shown in Table 1.

Table 1. RTL8019AS IOS Pins with the Base Address of the Corresponding Relation Table

IOS3	IOS2	IOS1	IOS0	I/O BASE
0	0	0	0	300H
0	0	0	1	320H
0	0	1	0	340H
0	0	1	1	360H
1	0	0	0	380H
1	0	0	1	3A0H
1	0	1	0	3C0H
1	0	1	1	3E0H
0	1	0	0	200H
0	1	0	1	220H
0	1	1	0	240H
0	1	1	1	260H

1	1	0	0	280H
1	1	0	1	2A0H
1	1	1	0	2C0H
1	1	1	1	2E0H

Now discuss the corresponding relationship of connection details. I/O base address, choose one of the base address in some situations there may be, when the interface with the ARM, after many validation choose any of them are not obvious influence on network CARDS work, only the corresponding connection is different, when programming the address is different. To note is that the pin definition corresponds to an ios in another definition name is BD0, IOS2 corresponding BD1, IOS1 corresponds to 802, IOS0 corresponds to BD3, if not carefully especially easy to cause mistake, cause the network card can't work normally.

As an example to elaborate on below. Can be seen in table 1 for each addressable unit of base address is 32 (20 h), such as choose I/O base address is 360H, then the corresponding ios - IOS0 logic level is 0011, 0 represents the grounding, 1 represents the + 5VDC (VDD). Base address written binary is 001101100000B, 360H, 360H is the addressing space - 37FH. 37FH binary is written as 001101111111B, compared with 360H of binary, can find seven front the same and only after five different, but the network card chip from SA0 are used along the address of the line, there are 20, comprehensive, in front of the binary complement into 20, can be written as 000000000011011 XXXXXB, x said here can be 0 or 1. So you can see, are used along - SA0 this 20 address thread pin connection is in turn are used along - SA10, SA7 are received digital DGND, SA9, SA8, SA6, SA5 receives a + 5VDC, the rest of the SA4 - A4 - A0 SA0 received address bus. The address bus and IOS connection is finished.

3.2. Card Chip Debugging

Net long chip does not work alone, also must have a network transformer in the RJ-45 interface and network cards - intermediate level transform chip. Of various components of the welding is completed, will be ready to test. Card chip has two LED indicator is used to indicate a state of receiving and sending, if you have Internet connection and normal send and receive packets, the LED will be flashing. When judge card chip is working correctly, there are two standard: one is to see if there is a flashing status indicator LED. The second is to use special software to monitor network monitoring tools, such as can use Sniffer software, listening to the network card constantly send out the specific test data suggests that card to work normally.

4. The Software Design

4.1. Ethernet Front-end Ports Storage and Initialized

RTL8019AS internal RAM, address range from 0x0000-0x7FFF, where 0x4000-0x7FFF used to send and receive buffers.

1. How to read or write RAM

RTL8019AS internal RAM is double Port RAM, for it to support two separate operation, the user is the CPU reads the contents of the RAM, the operating

RTL8019AS provide a mouth, speaking, reading and writing, also is the Remote DMA Port of register; Another is RTL8019AS internal control circuit to receive from the network data is written to RAM, RAM then referred to as the Local DMA. RTL8019AS through Local DMA write RAM is without user intervention, it through the Remote Port DMA read-write RAM.

Read RAM see RTLReadRam function. This function indicates that reading size bytes starting address address content to buff pointer to memory. Set the CR register to:

```
WriteReg(CR, (0x00 | CR_REMOTE_READ | CR_START_COMMAND));
```

Then read from the Remote DMA Port size times, and get the required data.

Write RAM using RTLWriteRam functions, operation and reading RAM basically almost, as long as it will set the CR register statement to the following statement.

```
WriteReg(CR, (0x00 | CR_REMOTE_WRITE | CR_START_COMMAND));
```

The last step of reading the size time to write the size is ok.

2. Send and Receive Buffers

0x4000-0x7FFF send and receive buffers, can be divided into the send buffer and the receive buffer. Buffer is according to the management page, 256 bytes to a page, so send buffer page from 0x40 to 0x7F. Send buffer start page of the Settings in the TPSR registers, receive buffer of the start page in the PSTART register Settings, PSTART actually page also indicates the end of the send buffer, the end of the receive buffer page is PSTOP. So send buffer page from TPSR to PSTART-1, of the receive buffer from the PSTART PSTOP-1.

The next question is to send and receive buffer how much of each is appropriate. Here setting as follows:

```
#define RECEIVE_START_PAGE    0x4C
#define RECEIVE_STOP_PAGE     0x60
#define SEND_START_PAGE0      0x40
#define SEND_START_PAGE1      0x47
```

The sending buffer can accommodate two maximum Ethernet frame (maximum Ethernet frame size is 1514 bytes), the first frame on SEND_START_PAGE0 start page, the second frame on SEND_START_PAGE1 start page, the rest of the buffer as the receive buffer.

3. Ethernet Front-end Ports Initialization

Initialize the first step is to reset the Ethernet front-end ports. Ethernet front-end ports divided into software and hardware reset reset reset. Hardware reset by RTL8019AS the RST pin of a pulse rifle an Ethernet front-end ports. Software reset by writing ResetPort to reset, that is, to 18-1 f any register writes between any number and makes Ethernet front-end ports reset. The second step is to set up some initial value of the register. Register save native Ethernet physical address, only the physical address and register save the same Ethernet frames to be received (if in RCR PRO = 0).

Ethernet front-end ports hardware reset must be reset for the first time, that is to say use Ethernet front-end ports must be before the hardware reset. In addition, the hardware reset after go through about 10 ms can wait for Ethernet front-end ports operation, especially to send and receive operation.

4. Ethernet Front-end Ports Driver

Ethernet front-end ports of the drivers is/arm/kernel/Linux/drivers/net/this file. Modify the files in the line 500, change to the following form:

For (offset = 1；offset < (buf[0] & 0xff)/2；offset++) {

Such changes, because the cycle of the cycle number depends on the Ethernet interface module, matching the capacity of the memory, the driver not to memory capacity to detect, but uses Ethernet interface default matching capacity directly, but the hardware board adopted by the storage capacity is half the default capacity, so there will be cycles were reduced by half.

4.2. Send the Packet

Send the basic steps are as follows:

(1) choose the start page of the first, is generally within the send buffer pages (0x40-0x4b), written to StartPage variable. To send data to address to StartPage began to buffer, then wait for the last time to send over. For packet is too large or too small, don't send; Ethernet is the largest minimum frame size, too small for the frame, to fill in when sending.TPSR start register for sending, will write TPSR StartPage registers, high byte write TBCRH (TBCR1), and the low byte write TBCRL (TBCR0). When write to send command, RTL8019AS from TPSR < < 8 address to send the size bytes of data. Command as follows:

```
WriteReg(CR,((PrePage&0xC0) | CR_ABORT_COMPLETE_DMA | CR_TXP |
                    CR_START_COMMAND));
```

If send in mishan package Medium reservoir black area of the store as shown in Figure 4, the RTL8019AS cannot automatically connected two areas, namely, the current page sent to RECEIVE_START_PAGE, it won't turn to SEND_START_PAGE, but send the shadow part of the content.

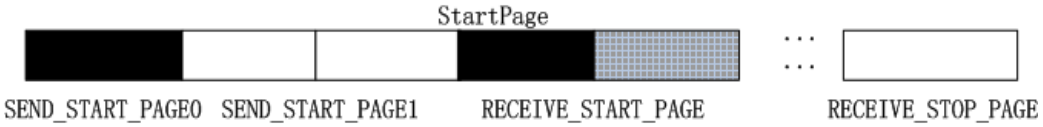

Figure 4. Packets Sent to Store

(2) sending data frames send buffer can store two maximum Ethernet frame. Send buffer can be stored for every frame of the two, a start page for SEND_START_PAGE0, another start page for SEND_START_PAGE1, two alternate use. If send the command is:

WriteReg (CR,((PrePage&0xC0)|CR_SEND_PACKET|CR_TXP|CR_START_COMMAND))；

RTL8019AS internal RAM crash at the next, speaking, reading and writing. Will send order to:

```
WriteReg (CR,((PrePage&0xC0)|CR_ABORT_COMPLETE_DMA|CR_TXP|CR_START_COM
                    MAND))；
```

There is no issue, but also can correct to send data.

4.3. Receiving Packets

Receives the packet, complete the following steps:

(1) receiving buffer operation. When RTL8019AS receives a packet, it will automatically receive packets into the CURR page. If a page not put, add 1 ; CURR If the CURR = RECEIVE_STOP_PAGE CURR automatically become RECEIVE_START_PAGE, continue to write the received data.

(2) the user read the packet received. RTL8019AS through Local DMA put the receive data written to the buffer, buffer and automatically change the CURR and identification of boundaries, all without user intervention. The next is how the user read the packet within the receive buffer.

When an error-free data reception and trigger the interrupt handler. The next step is to read packets to the allocated memory, read how many can learn from ReceiveByteCount. Here are buried in a situation: if receive packet storage is not continuous, as shown in Figure 5. This case needs two times to read a complete packet.

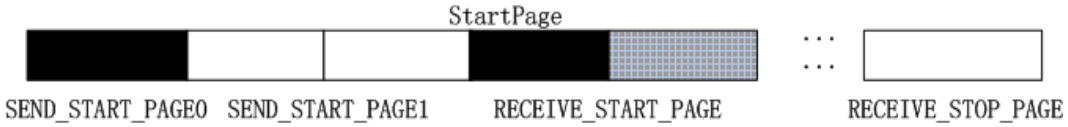

Figure 5. Packets Storage of Discontinuity

Determine whether stored discontinuous condition is:

BnryAdd1 > Head[1] && Head[1]!=RECEIVE_START_PAGE

BnryAdd1 is the start page of the packet, the Head [1] is a bag under the start page. And then will receive the data written to the network interface layer of the input queue, if write failed memory is released. After writing the upper agreement will extract the packet. Finally nic to the RAM through the interrupt controller the interrupt response, interrupt is clear interrupt flag, enables the later and low-level interrupts at the same level to response.

5. Summary

Above introduction of Ethernet interface module based on ARM technology is to construct a common ARM embedded system based on network, the basis of the interface module's main task is to complete information interaction with the outside world, in order to achieve the purpose of network monitoring. In practice, it runs stable, can very easily achieve embedded system network interconnection. In the ARM system adopts high performance of the Ethernet controller, system communication and debugging fast, reliable, has the very high real-time performance.

References

[1] P. H. W. Wong, O. C. Au and Y. M. Yeung, “A Novel Blind Multiple Watermarking Technique for Images”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, no. 8, **(2003)** August, pp. 813-830.
[2] Y. Jianxin, “An embedded system based tutorial”, Machinery industry press, **(2008)** March.
[3] G. Wei, “SoC design methods and implementation”, Electronic industry press, **(2007)** June.
[4] D. Jianguo, “Verilog hardware description language HDL cholesterol [C]”, **(2004)** January.
[5] X. Yinghui, M. Zhongmei and W. lei, ”ARM9 embedded system design [C]”, Beijing university of aeronautics and astronautics press, **(2008)** May.
[6] Z. Yamin and D. Xiaozhou, “Programmable logic devices to develop software QuartusII”, national defense industry press, **(2006)** September.

Author

Zheng Peng, Male, he was born in 1978, lecturer, master, Graduated from Shandong University of Science and Technology, software engineering. The main research direction for the data of medical image processing, medical information.

