

Design and Implementation of a Video-Oriented Network-Interface-Card System

Ming-Chih Chen

Dept. of Computer Science and
Engineering
National Sun-Yet San Univ.
Kaohsiung, Taiwan, 80424
Tel : +886-7-5254336
Fax : +886-7-5254301
e-mail :
kenchen@garfield.cse.nsysu.edu.tw

Shen-Fu Hsiao

Dept. of Computer Science and
Engineering
National Sun-Yet San Univ.
Kaohsiung, Taiwan, 80424
Tel : +886-7-525-2000 ext 4314
Fax : +886-7-525-4301
e-mail :
sfhsiao@cse.nsysu.edu.tw

Cheng-Hsien Yang

Dept. of Computer Science and
Engineering
National Sun-Yet San Univ.
Kaohsiung, Taiwan, 80424
Tel : +886-7-5254336
Fax : +886-7-5254301
e-mail :
luke@garfield.cse.nsysu.edu.tw

Abstract - We design a specific Ethernet network interface card (NIC) for accelerating the video delivery by offloading the overheads of protocol headers identification/appending and CRC/checksums calculation, and speeding video bit streams with a dedicated video interface. Compared with the same operations of a 50MHz ARM micro-controller, the NIC system saves 47,000 ns per frame. This NIC card also supports the coexistence of the IPv4 and IPv6 standard for the future extension. Both FPGA prototyping and 0.35um cell-based design of the specific NIC system are given.

I. Introduction

The conventional Ethernet/Fast Ethernet dominate most LAN services of today's network. The Internet Protocols Suite, IPv4 was developed to allocate the addresses of networked hosts and communicate with other networks. IPv6 solved the shortage of IPv4 addresses by extending the address width from 32-bit to 128-bit. The coexistence of these two IP versions meets the future requirements. The transport protocols such as TCP and UDP are performed in the processor kernel, causing heavy burden for host processors. Optimizing these protocol routines can improve the performance of network. Thus offloading these protocol routines such as header recognition [1], checksum calculation [2], CRC (cyclic redundant check) computation [3],... etc. into specific hardwired circuits is very critical for many real-time networking applications.

Our newly developed Ethernet NIC can accelerating the transmission of video streams through a dedicated video interface connector and hardwired circuits of the NIC can reduce the service time by offloading the protocol routines.

II. Architecture of Our Ethernet NIC

Two major parts of our Ethernet NIC are the software drivers and hardware NIC controller. The software packages support the FPGA-based prototype system and perform the debugging and testing based on a Pentium processor (as a host). Our developed NIC software adopts the Linux kernel and three hierarchical driver stacks. The upper layer implements the PCI network device driver. The middle layer implements a set of device operation functions. The lower layer consists of a set of basic hardware utilities and macros. Fig. 1 shows the five major parts of the NIC controller:

- (1) a 10/100 Mbps Ethernet MAC, which is responsible for receiving and transmitting frames;
- (2) a packet classifier and decapsulation engine, which separates the video streams from other data streams and forward them to the specific FIFO queues;
- (3) a packet encapsulation engine and dispatcher, which dispatches the video and data streams respectively to the video interface and to the host.
- (4) a DMA controller, which supports the data movement between queues and PCI bus;
- (5) a dedicated video interface, which allows the identified video data to be forwarded to video connector directly;

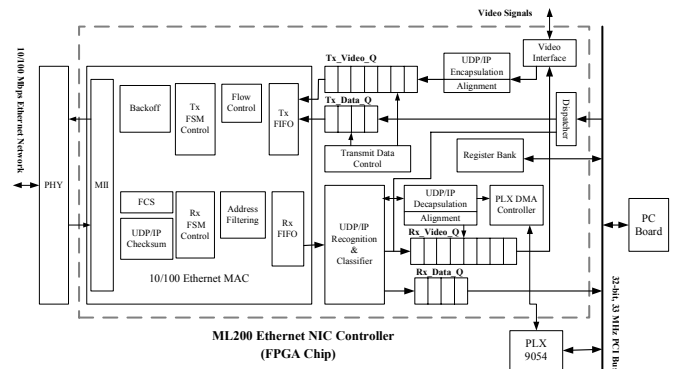


Fig. 1. The Hardware Architecture of Our Ethernet NIC

III. Performance Evaluation of HW/SW Co-Design

The performance evaluation is based on calculating the instruction counts of ARM micro-controller that spent in executing these major networking routines: header identification, checksum, and CRC computation.

A. Ethernet/IP/UDP Header Identification and Appending

The identification/appending of Ethernet/IP/UDP headers are subroutines in the receiving/transmitting phase separately. The service time of the process can be calculated except for the consideration of context switch when ARM performs other processes as shown in Table I.

B. IP and UDP Checksums Offloading

Checksums are used to detect errors of a packet. The service time of performing the checking/appending process of checksum in the ARM can be calculated as:

$$T_{Checksum} = (17 + \frac{9}{4}L) \times T_{Cycle} \text{ (ns/frame)} \quad (1)$$

where $T_{Checksum}$ is the service time spent in performing the checksum algorithm; L is the length (in unit of bytes) of a packet; T_{Cycle} is the cycle time of the ARM micro-controller.

C. Parallel CRC Computation

Parallel CRC unit is designed for checking the errors of a frame. The service time of performing the checking/ appending of CRC in the ARM can be calculated as:

$$T_{CRC} = (32081 + 2288 \times L) \times T_{Cycle} \text{ (ns/frame)} \quad (2)$$

where T_{CRC} is the service time of performing CRC process.

D. Overall Improvements

Table II shows the service time of processing these operations and can be saved by the parallel dedicated circuits of our NIC system.

Table I. SERVICE TIME OF THE IDENTIFICATION/APPENDING PROCESS

Item	Cycles	Service Time (ns) (Proc. Freq.= 50 MHz)
A. Total Service Time of Identification Process		
(1) Ethernet/IPv4/UDP	152	3,040
(2) Ethernet/IPv6/UDP	170	3,400
B. Total Service Time of Appending Process		
(3) Ethernet/IPv4/UDP	59	1,180
(4) Ethernet/IPv6/UDP	75	1,500

Table II. COST TIME OF VARIOUS SERVICE TYPES

Service Type	Consumed Cycles (ns/frame) (L: Frame Length, bytes)
Saved Rx Time: $T_{Rx,cost} = T_{Ident.} + T_{Checksum} + T_{CRC}$	
(a) Rx: Ethernet/IPv4/UDP	32550+230.25L
(b) Rx: Ethernet/IPv6/UDP	32551+228L
Saved Tx Time: $T_{Tx,cost} = T_{Append.} + T_{Checksum} + T_{CRC}$	
(c) Tx: Ethernet/IPv4/UDP	32157+230.25L
(d) Tx: Ethernet/IPv6/UDP	32156+228L

IV. Implementation of Our Ethernet NIC

The FPGA-based implementation of our NIC system is designed with the Altera EP20K400EBC652 FPGA chip as shown in Fig. 2. The device utilization of our NIC is given in Table III. The entire NIC controller, excluding the video and data queues, uses 55.51 % of the FPGA logic cells. The working clock frequency is 36.6 MHz. Video and data queues use 180,320 bits of memory cells in the FPGA chip. Another portion of the NIC card is the peripheral circuit that contains a video connector supporting the communication of 8-bit parallel data and control commands between the NIC and the specific video resource. Fig.3 shows the layout-graph of our NIC using 0.35 μm cell library and the die size is 5,745 $\mu\text{m} \times 5,745 \mu\text{m}$ (33.005025 mm^2).

Table III. DEVICE UTILIZATION OF ALTERA CHIP

Resource	Used	Available	Utilization (%)
IOs	142	488	29.10
LCs	9237	16640	55.51
Memory bits	180320	212992	84.66



Fig. 2. The FPGA-based Prototype of Our NIC Card

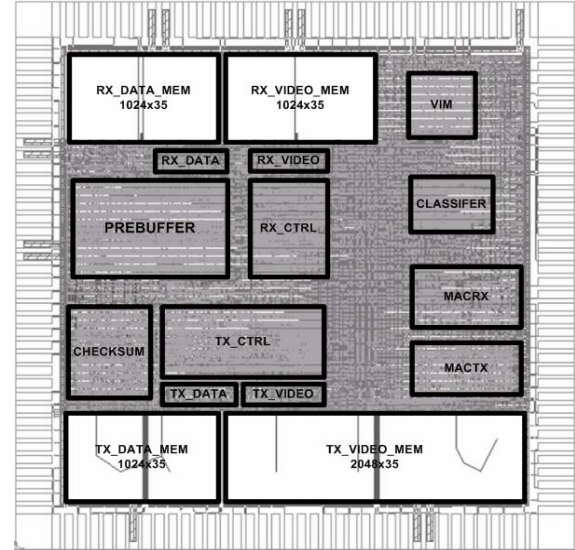


Fig. 3. The Layout-graph of Our Ethernet NIC

V. Conclusions

In this paper, we present a newly developed network interface card that combines the IPv4 and IPv6 protocols, offloads the subroutines of protocol software executed in the microprocessor, and adds a specific video channel. The coexistence of IPv4 and IPv6 extends the network utilization and the connectivity power. The offloaded subroutines include header identification, checksum calculation and parallel CRC computation. A special video interface is included to deliver the specific video streams more quickly. Our NIC provide three hardwired circuits that can effectively reduce the burden of the executed protocol routines in a microprocessor.

References

- [1] C. Partridge and S. Pink, "A Faster UDP", IEEE/ACM Transaction on Networking, Vol. 1, No. 4, pp. 429-440, Aug. 1993.
- [2] M. Ross, A. Bechtolsheim, M. T. Le, and J. O'Sullivan, "FX1000: A High Performance Single Chip Gigabit Ethernet NIC", in IEEE proceedings of Compton'97, pp.218-223, 1997.
- [3] M.D. Shieh, M.H. Sheu, C.H. Chen, and H.F. Lo, "A Systematic Approach for Parallel CRC Computations", Journal of Information Science and Engineering, Vol. 17, No. 3, pp. 445-461, May 2001.