An SoC Platform with On-Chip Web Interface for In-Field Monitoring

Tetsuya Iizuka[†], Daisuke Nakamura[‡*], Hiroaki Yoshida[†], Satoshi Komatsu[†],
Masahiro Sasaki[†], Makoto Ikeda[†‡], and Kunihiro Asada[†‡]

[†]VLSI Design and Education Center (VDEC), University of Tokyo
[‡]Department of Electronic Engineering, University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
Email: iizuka@vdec.u-tokyo.ac.jp

***Abstract*— As the VLSI technologies scale down to the nanometer regime, the circuit design and verification processes have become more and more complex and a reliable operation of VLSI becomes sensitive to the PVT (Process, Voltage, and Temperature) variations. Therefore, the LSI test only before the shipment to screen out the initial failures has been insufficient to ensure the reliable in-field operation of LSI.**

In this paper, we propose an SoC platform with on-chip web interface to realize an in-field LSI testing and an easy access to the on-chip LSI monitoring circuits such as scan registers, temperature sensors, and so on. We can control the on-chip monitoring systems through the web interface, and can monitor the LSI correct operations from remote locations using the proposed platform. Therefore, the proposed SoC platform realizes the LSI functionality monitoring even after the shipment and can test the in-field operation of LSI. This platform consists of 16-bit CPU, 64K words of instruction/data memory, and 10Base-T ethernet interface. A preliminary version of the proposed platform was implemented on 0.18μm standard CMOS process. The area overhead is 8.44mm^2 on 0.18μm process, and is estimated to scale down to about 1mm^2 on 65nm process.

I. Introduction

The improvement of VLSI process technologies over the last twenty years enables us to integrate a large number of transistors on a single chip, and significantly improves the circuit performance. Instead of this steady performance improvement, however, VLSI design and verification processes have become more and more complex. Moreover, a smaller feature sizes degrade the tolerance to the PVT (Process, Voltage, and Temperature) variations and increasing power densities also accelerate the impact of the stress-induced failures such as elector-/stress-migration, TDDB(Time Dependent Dielectric Breakdown), and so on[1]. These effects are found to have a great impact on VLSI lifetime reliability and degrade the device MTTF(Mean Time To Failure). Srinivasan *et al.* quantified the peak temperature and MTTF dependence on the process technologies in [1]. According to their estimation results, the worst case MTTF on 65nm process decreases to about two years whereas the one on 0.18μm is over fifteen years. Therefore, the LSI test only before the shipment to screen out the initial failures has been insufficient to ensure the reliable in-field operation of LSI and we have to test and monitor the operation of LSI by in-field monitoring to screen out the failures even after the shipment.

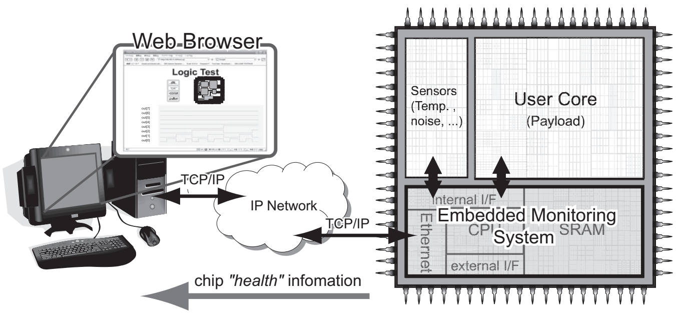


Fig. 1. The concept of the proposed platform

To realize the in-field testing and monitoring, we propose an SoC platform with on-chip web interface. The proposed platform provides a browser-based interface for designers and designers can test the LSI from remote locations through HTTP and TCP/IP connections. Therefore, the proposed SoC platform realizes the LSI functionality testing even after the shipment and can monitor the in-field operation. Moreover, we can also control various monitoring equipments such as scan registers, temperature sensors, etc.[2], [3] through this web interface when they are integrated on this platform.

The rest of this paper is organized as follows. In section II, we describe a concept of the proposed SoC platform to realize the in-field monitoring of the chip internal information. The architecture of the proposed platform is explained in section III, then section IV presents the experimental results and estimates the area overhead of the proposed platform. Finally, section V concludes this paper.

II. Concept of the Proposed Platform

The concept of the proposed SoC platform is depicted in Figure 1. As shown in the right part of Figure 1, the proposed platform has the user core region, the on-chip sensors to monitor the chip internal informations, and the embedded monitoring system. The on-chip sensors include temperature sensors like [3], and so on. Some type of sensors should be distributed in the user core region. The embedded monitoring

[*] He is now with Toshiba Corporation.

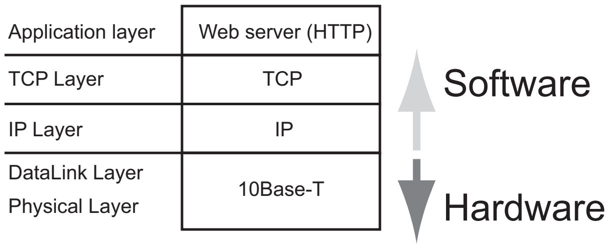


Fig. 2. Hardware and software functionalities of our embedded web server

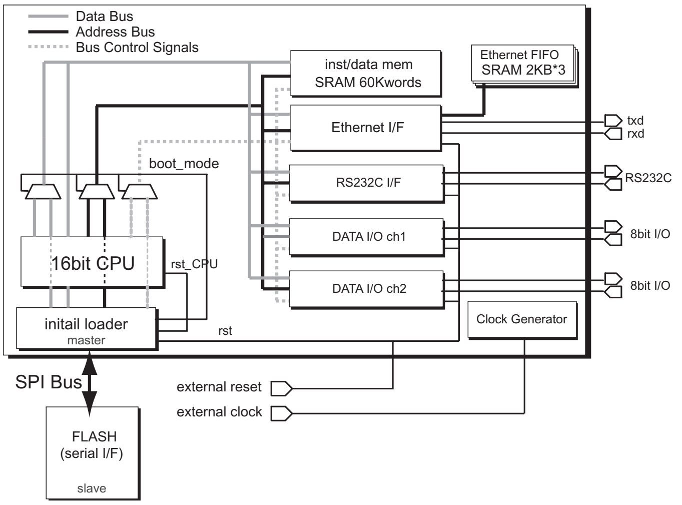


Fig. 3. Block diagram of the hardware architecture of the proposed platform

system offers a web interface through which designers access to the chip functionality and its internal information (so-called chip *health* information) such as internal digital signal states, and temperature distributions etc. The designers can access to these informations from remote locations by using HTTP and TCP/IP connections. A client PC depicted on the left part of Figure 1 communicates with the embedded monitoring system on the proposed platform using web browser to access to the user core functionality and *health* informations.

The embedded monitoring system in the proposed platform has an ethernet interface to transmit and receive packets, and a tiny CPU to control monitoring functions and TCP/IP communication. The on-chip monitoring operation is executed as follows. First, a client PC sends an HTTP request to the platform and the on-chip ethernet interface receives the HTTP request. Then, the HTTP server running on the on-chip CPU analyzes the received packets and gathers the requested chip information from on-chip monitoring sensors. Finally, the HTTP server generates the HTML code of the requested information on the memory, and sends back this HTML code to the client PC.

By employing the proposed platform with on-chip web interface, the following advantages can be obtained.

1) The operation of the chip can be easily tested and monitored from remote locations as long as the chip is connected to the internet. We can keep monitoring the in-field reliable operation of the chip due to this feature.
2) No dedicated equipment is required on the client PC to use the proposed monitoring system since ethernet interface is very common in PC.
3) A common web browser can be used as a monitoring interface. This feature provides an intuitive operation to users of the proposed platform.

Although the proposed platform has these advantages, the integration of the embedded monitoring system has disadvantages in terms of area overhead. In this paper, the area of the proposed platform is not negligible compared to the user core area since this version is implemented on the 0.18μm process. On the actual target processes that may pose significant challenges to lifetime reliability such as 65nm, however, the area of the embedded monitoring system scales down accordingly and the estimated area becomes small enough to be negligible compared to the user area. In addition, if the network appliances become widespread and an on-chip ethernet interface is commonly implemented[4], the area overhead of the monitoring system is expected to become extremely small. Therefore, the area overhead of the proposed platform will be small in the future.

III. Architecture of the Proposed Platform

We implemented the preliminary version of the proposed platform to confirm the correct operation and to quantify the area overhead. The proposed platform was implemented on an ASIC chip using 0.18μm standard CMOS process. In this version, we implemented a web server and a simple on-chip testing equipment, and a full-functional *health* information processing feature was not implemented in this version yet.

Figure 2 shows the functionalities of the embedded web server that is implemented in hardware and software. Although a hardware TCP/IP protocol processor can process packet faster than CPU and reduce CPU load, this kind of hardware is too complex and the area overhead becomes too large. Therefore, our ethernet interface processes only physical and MAC layers, and IP, TCP, and HTTP layers are processed by the software.

The minimum functions of both hardware and software required to realize the on-chip monitoring with web interface are implemented on this version of the proposed platform, since a full-featured TCP/IP routine requires a large memory and area.

A. Hardware Architecture

Figure 3 shows the hardware block diagram of the proposed platform. This platform has 16-bit, non-Harvard architecture tiny CPU and 64K words SRAM for instruction and data memory. 10Base-T[5] ethernet interface, RS-232C interface, 2-channel 8-bit input ports, 2-channel 8-bit output ports are also implemented. All I/O ports including ethernet interface are implemented as memory-mapped I/O and are connected to the 16-bit data and 16-bit address bus. We developed the 16-bit non-Harvard architecture tiny CPU to minimize the hardware

TABLE I

INSTRUCTION SET OF OUR 16-BIT PROCESSOR

Category	Mnemonic Code	Explanation
Arithmetic	ADDF RA, RB, RC	RC = RA + RB
	SUBF RA, RB, RC	RC = RA - RB
Shift	SHR RA, RC	RC = {1'b0, RA[15:1]}
	SHL RA, RC	RC = {RA[14:0], 1'b0}
	SAR RA, RC	RC = {RA[15], 1'b0, RA[14:1]}
	SAL RA, RC	RC = {RA[15], RA[13:0], 1'b0}
	ROR RA, RC	RC = {RA[0], RA[15:1]}
	ROL RA, RC	RC = {RA[14:0], RA[15]}
	RCR RA, RC	RC = {C, RA[15:1]}
	RCL RA, RC	RC = {RA[14:0], C}
	SHR8 RA, RC	RC = {8'h00, RA[15:8]}
	SHL8 RA, RC	RC = {RA[7:0], 8'h00}
	ROR8 RA, RC	RC = {RA[7:0], RA[15:8]}
Logic	AND RA, RB, RC	RC = RA & RB
	OR RA, RB, RC	RC = RA — RB
	XOR RA, RB, RC	RC = RA R̂B
	XNOR RA, RB, RC	RC = (RA R̂B)
Load/Store	ST RA, RB, RC	write RB to memory[RA+RC]
	LD RA, RB, RC	read memory[RA+RB] to RC
Jump/Branch	JS RA, RB, RC	jump to [RA+RB], save PC to RC
	BRC RA, IMM6	Branch to [PC+IMM6] if Carry == 1
	BRNZ RA, IMM6	Branch to [PC+IMM6] if Zero == 0
Set	SET IMM8, RC	Set {8'h00, IMM8} to RC
	SETHI IMM8, RC	Set {IMM8, RC[7:0]} to RC
Interruption	STFG RC	Store flags to RC before int.
	LDFG RC	Load flags from RC after int.
	STPC RC	Store PC to RC before int.
	LDPC RC	Load PC from RC after int.
	INTE	Set int. flag 1
	INTD	Set int. flag 0
	RETI	Return from int.

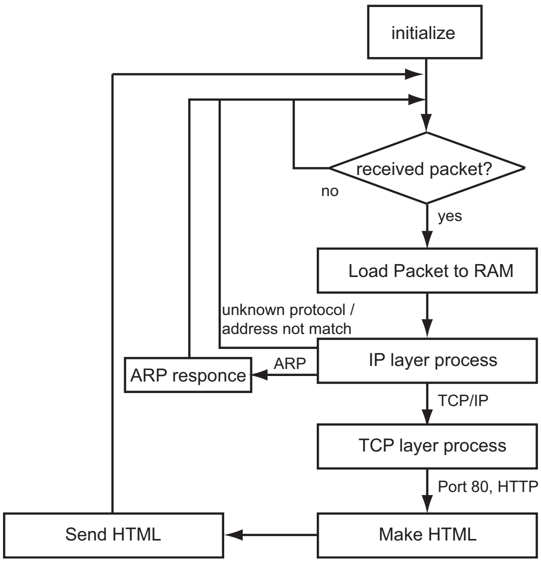


Fig. 4. Flowchart of the web server program

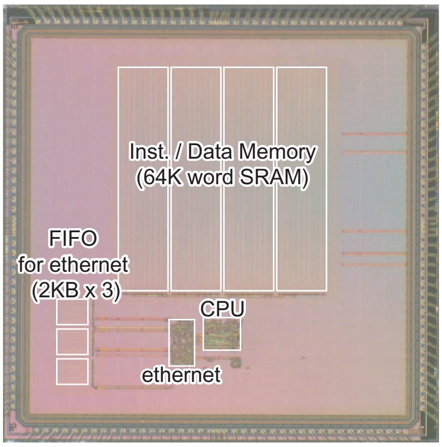


Fig. 5. Chip micrograph (0.18μm CMOS process)

area overhead, since our target software is only a monitoring system controller, TCP/IP, and web server program that do not require high performance processing. Our CPU consists of a decoder, ALU which supports the minimum required instruction set, and 16 x 16-bit register file. Table I shows the instruction set supported by our CPU, where R(A/B/C) means the register, and IMM*n* means *n*-bit immediate value, respectively. To further minimize the area overhead, we implemented a small 10Base-T controller which supports the minimum required functions. This ethernet interface has own FIFO buffers, 2KByte for transmitter and 2 x 2KByte (double buffering) for receiver.

The software running on the CPU is stored on RAM instead of ROM to preserve the flexibility of software. The initial program loader loads the software from external FLASH memory connected with SPI bus and then copy it into RAM after reset signal.

B. Software Architecture

We ported gcc program[6] to our CPU instruction set described in III-A to develop TCP/IP routines and web server programs in C language. We also developed a dedicated software emulator to verify the operation of the C programs written for our CPU. All the programs for our platform are developed in C language and verified its operation using these dedicated softwares.

In this version, TCP/IP processing routine and web server program are implemented. Figure 4 is the flowchart of the web server program. This web server employs polling method. First, the program checks whether any packets are received or not. If some packets are received, the program loads the packet contents from the ethernet interface and processes the TCP/IP packet. Finally, the program makes HTML code and sends it to the client PC, then go back to the start point of the polling loop.

IV. EXPERIMENTAL RESULTS

Our proposed platform design was implemented on 0.18μm standard CMOS process. Figures 5 and 6 represent the micrograph of the fabricated chip and the evaluation board, respectively.

Figure 7 shows a block diagram of the experimental setup. The transmitted and received signals are connected to on-board differential line driver and receiver respectively. The output and input ports of the driver and receiver are connected to a client PC through a switching hub. A web browser runs on the client PC, while the web server program runs on the CPU implemented on the fabricated chip.

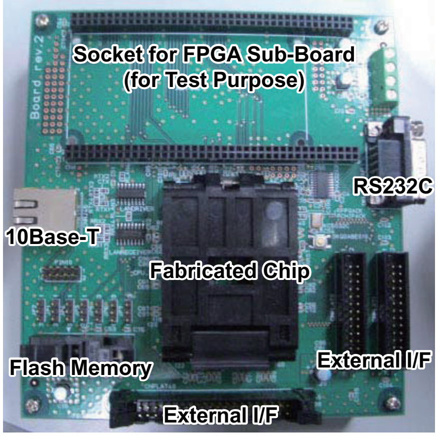


Fig. 6. Evaluation board for the fabricated chip

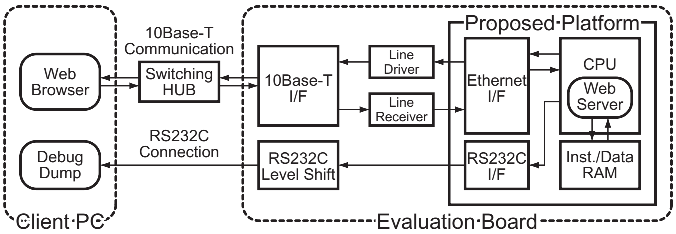


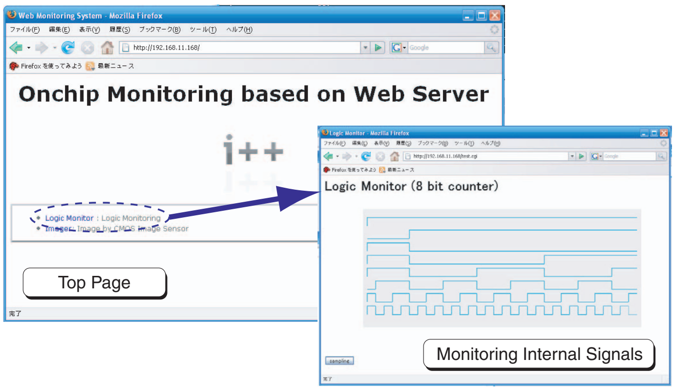
Fig. 7. Experimental setup of the functional verification

The operational verification was performed by sending HTTP request from a client PC to the platform on the chip, and checking returned HTML. Figure 8 is the returned HTML from the on-chip CPU. In this example, the internal signals of a simple 8-bit synchronous counter are monitored and their waveforms are displayed on a web browser. This result shows the correct operations of the internal signal monitoring sequence and web interface. The implementation of the various *health* monitoring circuits and the realization of the full-functional monitoring system are our future works. Table II summarizes the area overhead of each hardware component. The most significant area overhead is found to be instruction and data memory, which occupies nearly 90% of the total area overhead. The total area overhead of this version of the proposed platform implemented on 0.18μm process is 8.44mm^2 as shown in Table II. Although this area overhead is not negligible compared to the user core area, this overhead is estimated to scale down to about 1mm^2 on 65nm process which is our actual target processes that may pose significant challenges to lifetime reliability.

V. Conclusions

We proposed an SoC platform with web interface to realize an in-field LSI testing and an easy access to the on-chip LSI monitoring circuits. We can control the on-chip monitoring systems through the web interface, and can monitor the LSI correct operations from remote locations using the proposed platform. Therefore, the proposed SoC platform realizes the LSI functionality monitoring even after the shipment and can test the in-field operation of LSI.

The preliminary version of the proposed platform was implemented on an ASIC chip using 0.18μm CMOS process, and its functionality was verified. The area overhead of the proposed platform is about 8.44mm^2 on the 0.18μm process and is estimated to scale down to about 1mm^2 on 65nm process.


Web Monitoring System - Mozilla Firefox

ファイル(F) 編集(E) 表示(V) 履歴(S) ブックマーク(B) ツール(T) ヘルプ(H)

http://192.168.11.168/

Firefox を使ってみよう 最新ニュース

Onchip Monitoring based on Web Server

- Logic Monitor : Logic Monitoring
- Image : Image by CMOS Image Sensor

完了

Logic Monitor - Mozilla Firefox

http://192.168.11.168/test.cgi

Logic Monitor (8 bit counter)

sampling

完了

Fig. 8. Returned HTML from the on-chip CPU of the proposed platform

TABLE II
Area overhead on 0.18μm CMOS process

Process	CMOS 0.18μm
Chip size	4.9mm x 4.9mm
CPU	0.18mm^2
Ethernet (Core)	0.19mm^2
Ethernet (Memory)	0.51mm^2
Inst. / Data memory	7.55mm^2
Other	0.02mm^2
Total area overhead	8.44mm^2

Acknowledgments

This research project has been supported by STARC. The VLSI chip in this study has been fabricated in the chip fabrication program of VLSI Design and Education Center(VDEC), the University of Tokyo in collaboration with Rohm Corporation and Toppan Printing Corporation.

References

[1] J. Srinivasan, S. V. Adve, and P. Bose, “Lifetime Reliability: Toward an Architectural Solution,” *IEEE Micro*, Vol. 25, No. 3, pp. 70–80, 2005.

[2] “IEEE standard test access port and boundary-scan architecture”,IEEE Std 1149.1-2001

[3] M. Sasaki, M. Ikeda, and K. Asada, “A Temperature Sensor With an Inaccuracy of −1/+0.8°C Using 90-nm 1-V CMOS for Online Thermal Monitoring of VLSI Circuits,” *IEEE Transactions on Semiconductor Manufacturing*, Vol. 21, No. 2, pp. 201–208, May. 2008.

[4] R. Kango, P.R. Moore, and J. Po, “Networked Smart Home Appliances Enabling Real Ubiquitous Culture”, in *Proceedings of IEEE International Workshop on Networked Appliances*, pp. 76–80, 2003.

[5] “Part3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications”, IEEE Std 802.3-2005

[6] “gcc, the GNU Compiler Collection” http://gcc.gnu.org