LAPG-2: 가상 논리 분석기 및 패턴 생성기를 갖는 저비용 설계 검증 플랫폼

(LAPG-2: A Cost-Efficient Design Verification Platform with Virtual Logic Analyzer and Pattern Generator)

황 수 연 † 강 동 수 † 장 경 선 †† 이 강 †††
(Sooyun Hwang) (Dongsoo Kang) (Kyoungson Jhang) (Kang Yi)

요 약 본 논문에서는 FPGA 기반의 논리 회로를 에뮬레이션 하는 저비용 플랫폼인 LAPG-2의 구조 설계와 구현을 제안한다. 본 논문에서 제안한 에뮬레이션 플랫폼은 기존에 제안된 LAPG(Logic Analyzer and Pattern Generator)의 성능을 향상시키고, 더 많은 기능을 추가하였다. 따라서, LAPG-2는 기존 LAPG의 향상된 버전이라고 할 수 있다. 본 논문에서 제안한 LAPG-2는 크게 FPGA 기반 하드웨어 엔진과 에뮬레이션을 구동하고 결과를 모니터링 할 수 있는 소프트웨어 부분으로 구성된다. 호스트 컴퓨터와 FPGA 보드 사이의 양방향 직렬 통신 링크를 통한 새로운 통신 프로토콜을 제안함으로써 효과적인 상호 작용할 수 있는 검증 환경을 제공한다. 실험 결과, 본 논문에서 제안한 에뮬레이션 방법은 다른 방식들과 비교했을 때, 55%~99%의 통신 오버헤드 절감 효과를 얻었다. 하드웨어 면적의 경우는, 간단한 회로보다 입출력 포트 수가 많은 복잡한 회로에서 보다 더 효율적이었다.

키워드 : 검증, FPGA 프로토타이핑, 논리 에뮬레이션, 디지털 시스템 설계 테스트

Abstract This paper proposes a cost-efficient and flexible FPGA-based logic circuit emulation platform. By improving the performance and adding more features, this new platform is an enhanced version of our LAPG. It consists of an FPGA-based hardware engine and software element to drive the emulation and monitor the results. It also provides an interactive verification environment which uses an efficient communication protocol through a bi-directional serial link between the host and the FPGA board. The experimental results show that this new approach saves 55%~99% of communication overhead compared with other methods. According to the test results, the new LAPG is more area efficient in complex circuits with many I/O ports.

Key words : verification, FPGA rapid prototyping, logic emulation, digital system design test

1. Introduction

Design verification is becoming one of the critical and time-consuming aspects of the digital hardware design process. Traditionally, designers just use the software-based simulation tool to verify their design correctness. However, source level functional simulation by software is too slow for complex modern designs. Even though the size of a design is not very large, source-level functional simulation is inadequate for a reliable confirmation due to the imprecision between simulation and synthesis, and timing-related problems. These complications are cleared by lower-level (gate-level) simulation/emulation. The software-based gate-level simulation or

· 본 연구는 교육과학기술부의 과학기술위성 3호 개발사업의 지원을 받아 수행하였습니다.

† 학생회원 : 충남대학교 컴퓨터공학과
charisma95@cnu.ac.kr
atom@cnu.ac.kr
†† 종신회원 : 충남대학교 컴퓨터공학과 교수
sun@cnu.ac.kr
††† 종신회원 : 한동대학교 전산전자공학부 교수
yk@handong.edu

논문접수 : 2007년 6월 14일
심사완료 : 2008년 2월 4일

정보과학회논문지: 시스템 및 이론 제35권 제5호(2008.6)

timing simulation is available for further verification. Unfortunately, these have limitations on simulation performance.

The hardware-based emulation by the FPGA prototyping board [1,2] is one of the emerging alternative approaches to accelerate digital design verification while achieving both higher accuracy and better performance. For each specific application, different FPGA board designs and implementations are required to support each verification requirement. Each additional board design results in higher costs as well as more verification time. Even the existing general purpose rapid prototyping FPGA boards of high-end technology are not suitable for some of the applications that require special I/O. Furthermore, it is hard to learn how to use the generic multi-functional prototyping board for each specific demand. The Xilinx Chipscope [3], Altera SignalTap [4], or watch-point methods [5] may be alternatives to solve these challenges. Nevertheless, these tools also have their own drawbacks primarily because they just probe the signals in the FPGA by acting as embedded on-chip logic analyzers when what is needed is actual stimulus to the FPGA. JTAG [6] can be another alternative. JTAG is a generic testing method that uses a chain of serially linked I/O cells through which every stimulus (including clock signal and captured output signals) are shifted-in and read back. Because of the communication overhead, JTAG has limitations on efficiency.

In [7] we developed an FPGA-based verification platform named "LAPG (Logic Analyzer and Pattern Generator)" to meet the need for an interactive, flexible and yet cost-efficient hardware verification platform. This platform requires only a simple FPGA board with minimum features which enables a hardware emulation system. This system is not only economical, but it is also exactly what is needed. This approach is based on a virtual hardware wrapper that generates the stimulus for a user design as well as probes the outputs from the user design that is under test as shown in Fig. 1. The verification step is shown in Fig. 2.

The wrapper communicates with a host via serial link to send and receive test data. This approach

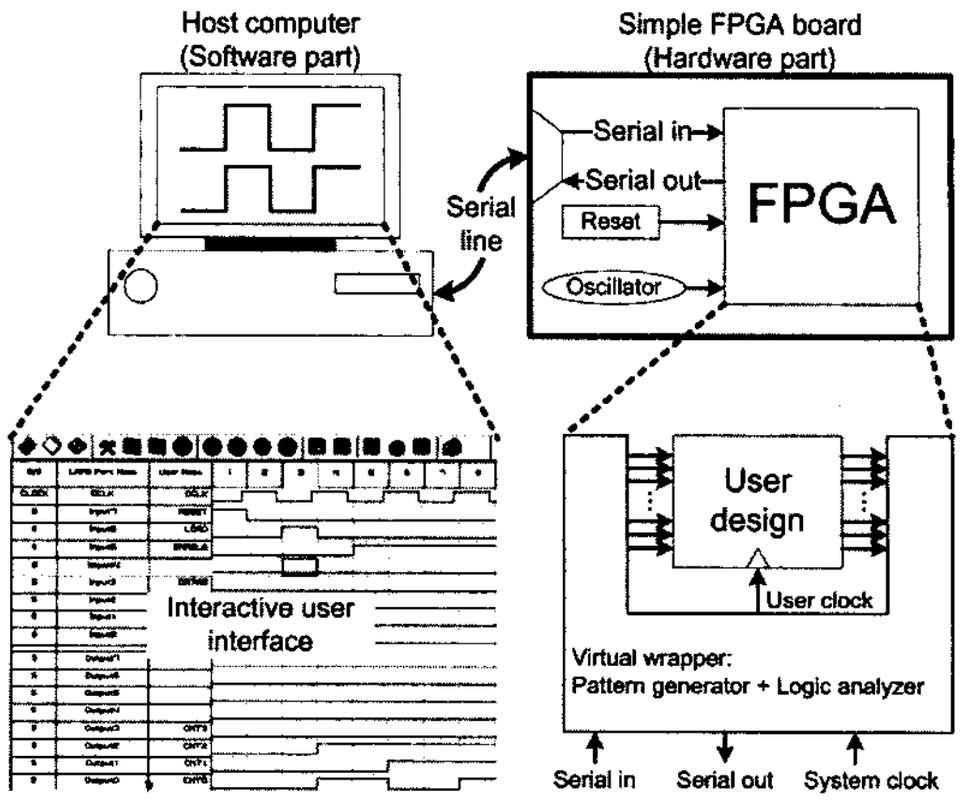


Fig. 1 LAPG: A proposed verification platform with the FPGA prototyping board connected to the host via serial link

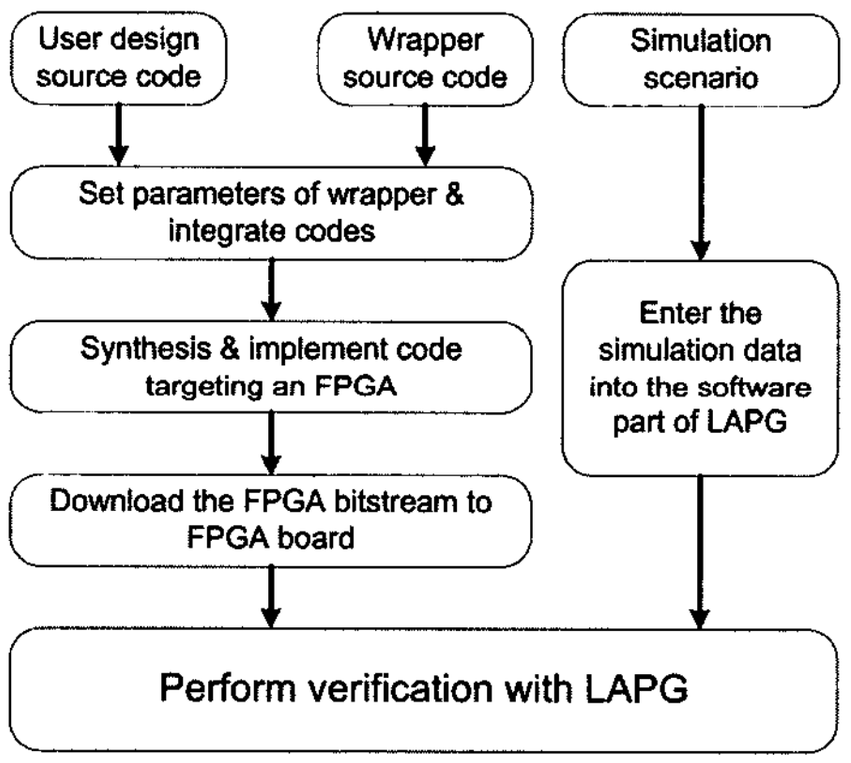


Fig. 2 The LAPG Design Verification Flowchart

eliminates the I/O restrictions of the FPGA board and provides higher testing capacity. It seems to be similar to the JTAG [6] scheme because of the serial-link based input pattern injection and output signal monitoring. However, it is different in two aspects: (1) it separates the clock signal from data and other control signals, (2) it uses the separated links for input and output signals in order to reduce the communication overhead via a serial link.

The hardware and the software of LAPG communicate with each other to perform verification of user design. However, LAPG is inefficient due to the communication overhead between the host machine and the FPGA board as every stimulus and clocking injection needs to send all the unre-

lated input/output data through the serial link. To deal with this issue, an enhanced version of the LAPG, the LAPG-2 has been developed by devising a new communication protocol to minimize the communication overhead through the serial link as well as to add more convenient features. The LAPG-2 has the following advantages: (1) it saves the verification time by using the same test vectors as used in the functional simulation; (2) it saves cost as there is no need to buy any additional testing equipment, and (3) it is able to perform tests in an interactive manner.

2. Enhancement of the Communication Protocol

The enhanced communication protocol of the LAPG-2 gives less communication overhead and more verification capacity. With LAPG-2 we can control the iteration patterns of data injection and sampling as freely as possible to obtain the most efficient communication. The number of clock cycles needed to get the valid results can be specified without additional input data, consequently saving the traffic by unnecessary data injection and output sampling data transmission. Additionally, experimentation with various types of clocking schemes with LAPG-2 can be performed.

The new communication protocol consists of four types of packets: (1) commands, (2) ACK (Acknowledge), (3) injection data, and (4) sampling data. The commands provide instructions on how to apply stimulus and monitor the output data. The ACK is the response from the hardware to the software which confirms that the hardware received the proper command and is ready for the next command or data from the host computer. The command packet formats are shown in Fig. 3.

The reset and execution command packets are distinguished by the MSB in the command packet. The reset command configures parameters that determine injection and sampling data width, active clock edge type, sampling intervals, and the number of clock cycles needed until sampling. The execution command dictates whether to perform injection and/or sampling with the subsequent data packets as well as specifies how many data packets

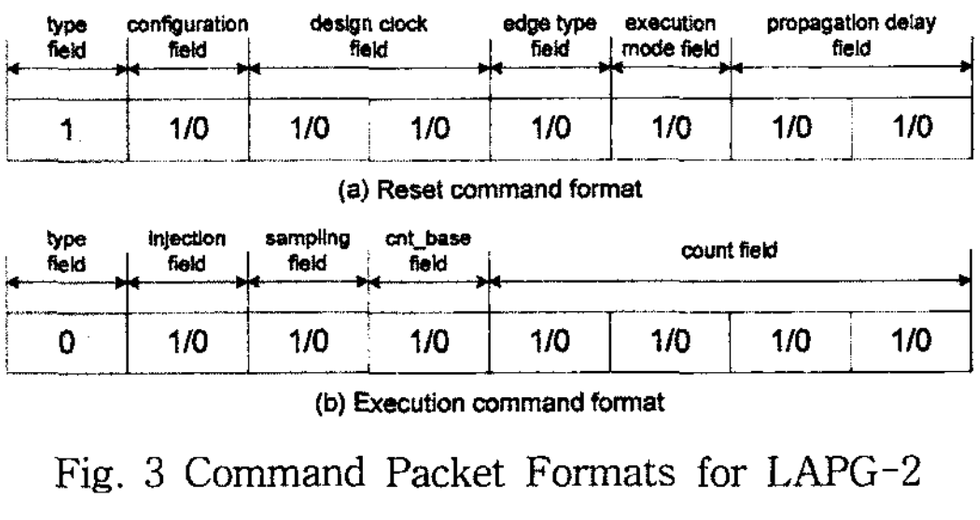


Fig. 3 Command Packet Formats for LAPG-2

are required.

Table 1 shows some examples of communication scenarios for different types of emulation operations. For example, we can perform a number of injection operations without any sampling, a number of sampling operations without any injections, or alternating injection and sampling operations.

With the proper packet mix, we can minimize the communication overhead, thereby achieving better performance. Fig. 4 presents a waveform for a DES cryptography circuit test.

This is a typical case that requires multiple clock cycles to get a valid output for each input data.

Table 1 Examples of Communication Scenarios

Operations	Communication Sequences
Configuration	{reset, ack}+
Injection only	{exec, ack, injection+}+
Sampling only	{exec, ack, sampling+}+
Injection-Sampling	{exec, ack, {injection, sampling}+}+

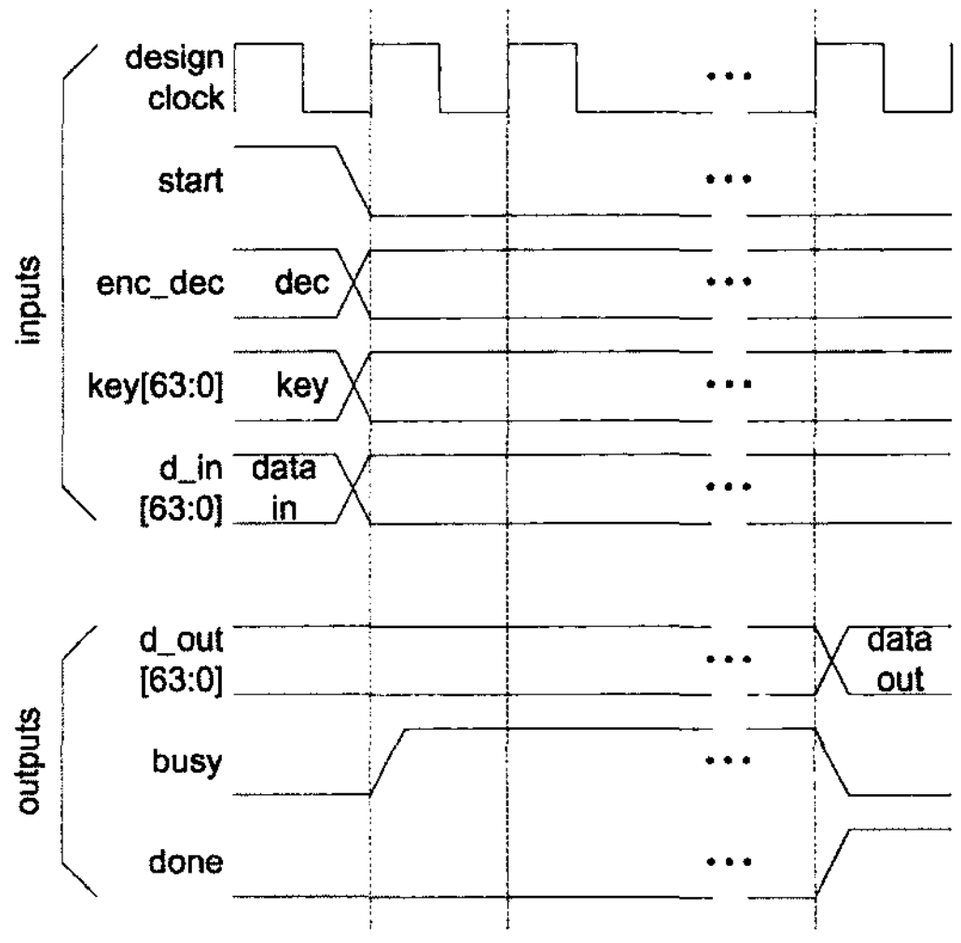


Fig. 4 DES Circuit Simulation Waveform

Fig. 5 shows the corresponding communication scenario between a host computer and the FPGA board for DES circuit design verification.

Fig. 6 shows the internal structure of a virtual wrapper circuit for LAPG-2 with enhanced communication protocols.

The command register as well as the data injection and sampling registers are shown in Fig. 6. The controller module in the virtual wrapper manages the serial communication procedures for the handshaking protocol. It also controls data injection registers, output data acquisition registers and clock triggering operation for a design under test.

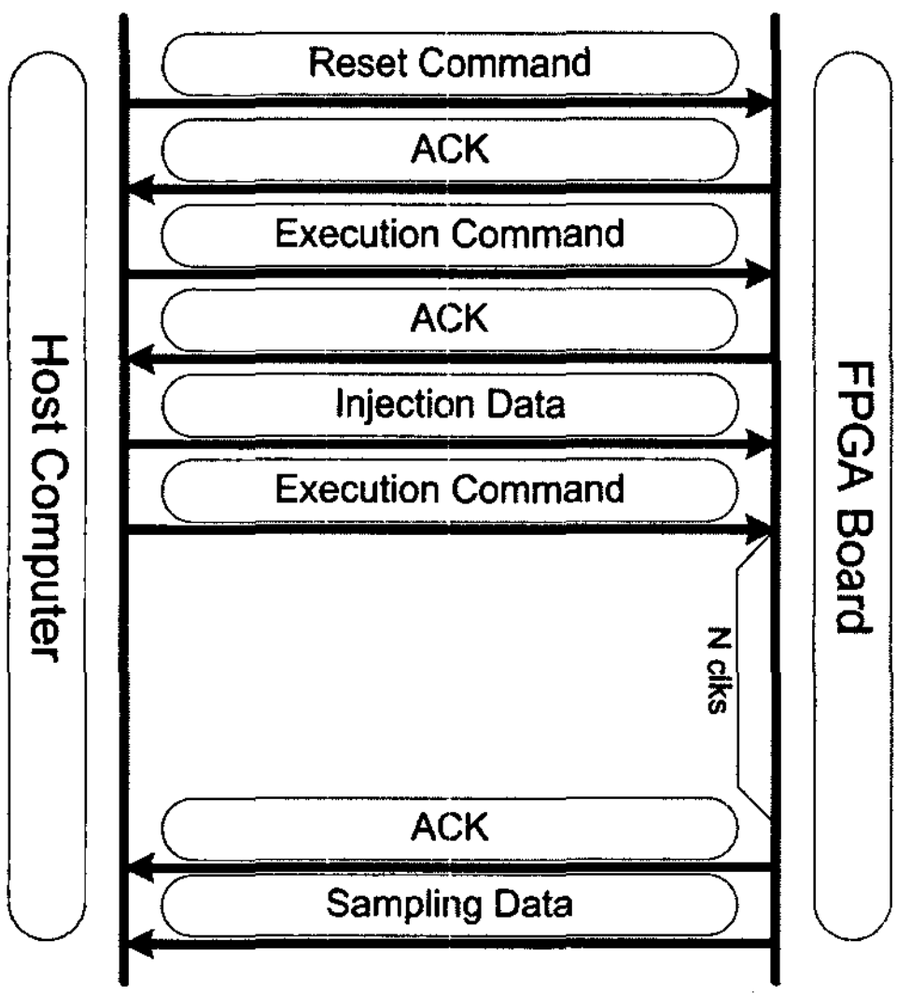


Fig. 5 Emulation Scenario for a DES Circuit

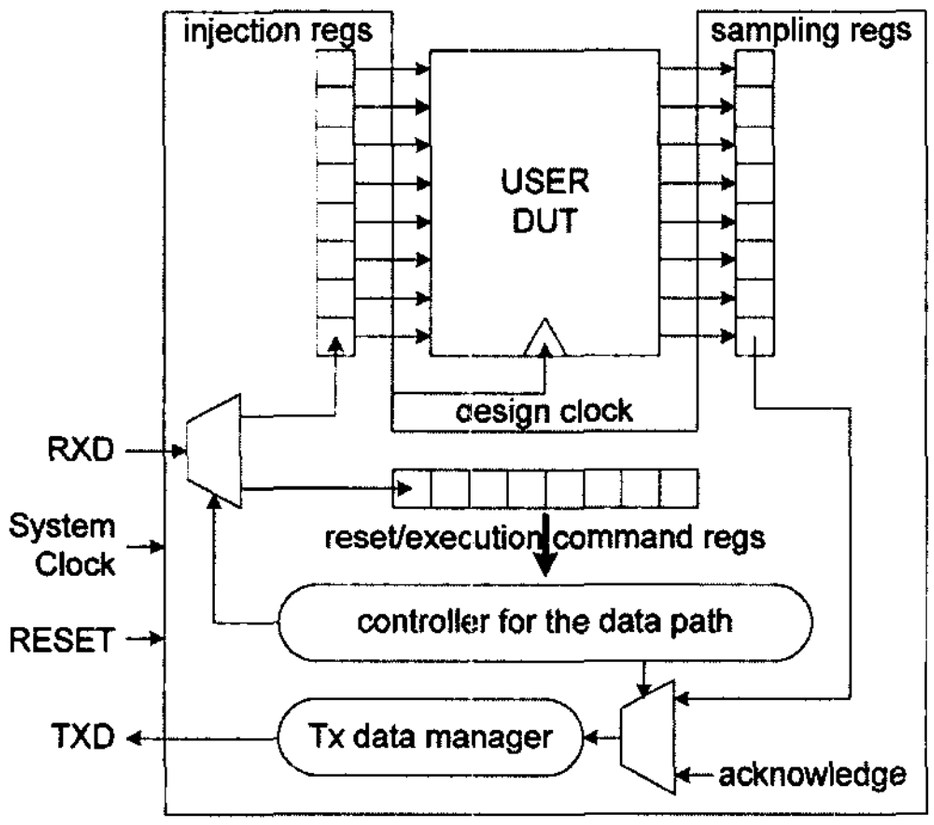


Fig. 6 Improved Structure of the LAPG Wrapper

3. Evaluation

We evaluated the benefits and cost of our method in terms of communication and area overhead by the proposed platform implementation with Xilinx FPGA board. To compare the efficiency of the methods, we applied LAPG-2, LAPG-1 and JTAG on several circuit examples. Table 2 summarizes the circuit examples used for the experiments.

Fig. 7 compares the communication overhead including commands as well as data traffic for different methods with different circuits.

The graph shows that LAPG-2 saves about 55%~99% of the communication overhead. This large amount of communication traffic saving owes to the higher flexibility of LAPG-2 protocol compared to the LAPG-1 protocol. That is, LAPG-1 requires the input ports and output ports should be the same size while LAPG-2 allows different inputs and outputs size. Thus, it is clear that in any case LAPG-2 has the least communication overhead than the other approaches. It is also observed that more overhead savings are definitely achieved with complex circuits with many I/O ports.

Table 2 Example Circuits

Circuit Name	Inputs × Outputs	# of FPGA Slices
Seq. Multiplier	10 × 8	11
Modified Booth Multiplier	10 × 8	17
Seq. Divider	14 × 9	18
DES	131 × 66	320
AMBA Bus Controller	15 × 4	47
SEED	259 × 128	647

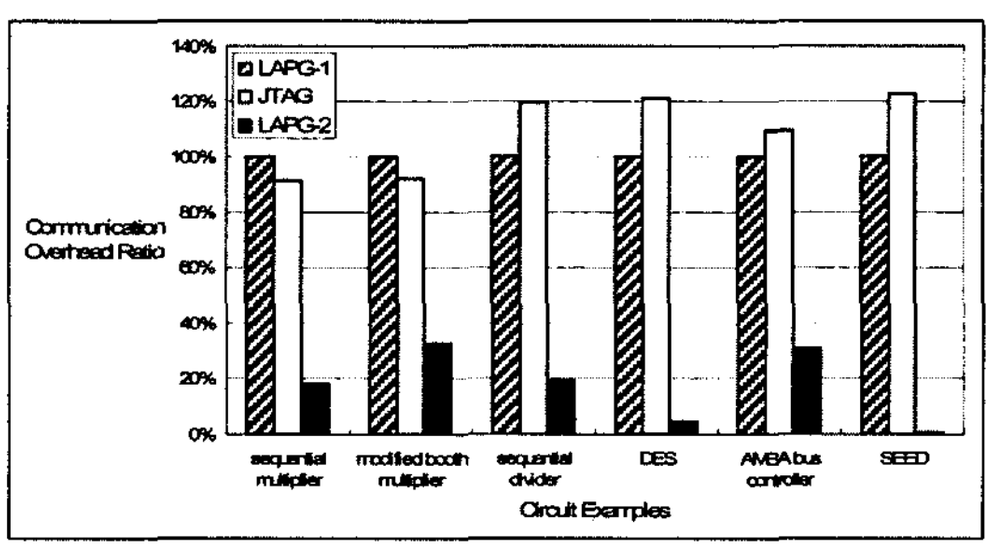


Fig. 7 Communication Overhead Comparison for Different Verification Methods (assume LAPG-1 communication overhead = 100%)

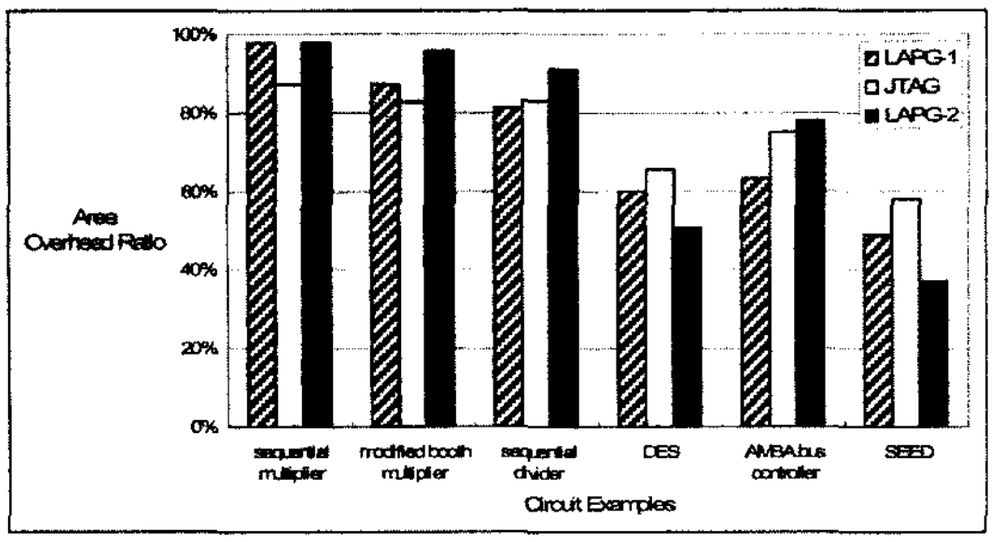


Fig. 8. Area Overhead Comparison for Different Verification Approaches

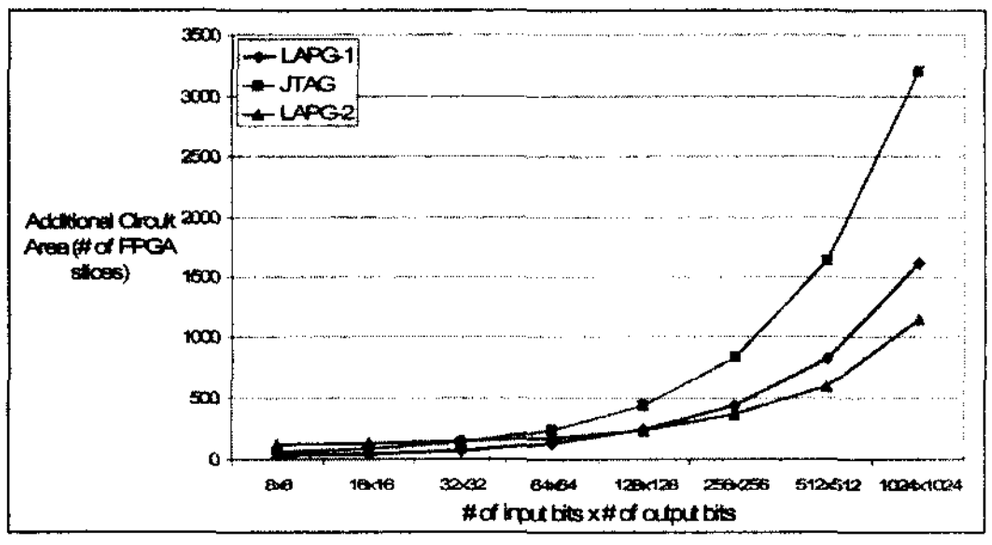


Fig. 9 Area Overhead by Different Verification Methods with Different Input × Output Port Sizes

What is the cost for these benefits? Fig. 8 shows the area overhead for each method. Fig. 8 demonstrates that LAPG-2 uses more area than any of the other methods if the circuits are relatively simple, while it uses less area than other methods if the circuits are complex and have many I/O ports. Fig. 9 shows the area overhead of three verification methods according to the increase of the number of I/O ports. From these graphs it is evident that for a small number of I/Os, LAPG-2 occupies more area than both LAPG-1 and JTAG, but as the I/O size increases over 64x64, LAPG-2 uses the least area and JTAG uses the most area.

4. Conclusions

In this paper, we proposed an efficient hardware verification platform based on a virtual wrapper and serial communication protocol. We believe that *our approach can contribute to the reduction of the* price of a FPGA prototyping platform because it eliminates the necessity of expensive I/O peripherals on the FPGA board. Designers can have controls over the input signals and monitor outputs from the design circuits without using any expensive, complex pattern generators or logic analyzers. Furthermore, verification time may be saved since the same test vectors for both software simulation and FPGA- based hardware verification may be used. We believe that LAPG-2 can meet needs that require powerful and yet a low-cost verification platform.

References

[1] Butts, M., Batcheller, J., Varghese, J., "An Efficient Logic Emulation System," Proc. Int. Conf. on Computer Design, Oct. 1992, pp. 138-141.

[2] Lo, W.Y., Choy, C.S., and Chan, C.F., "Hardware Emulation Board Based on FPGAs and Programmable Interconnections," Proc. Int. Workshop on Rapid System Prototyping, June 1994, pp. 126-130.

[3] Oltu, O., Milea, P.L., and Simion, A., "Testing of Digital Circuitry using Xilinx Chipscope Logic Analyzer," Proc. Int. Conf. on Semiconductor, Oct. 2005, pp. 471-474.

[4] SignalTap II Embedded Logic Analyzer, http://www.altera.com.

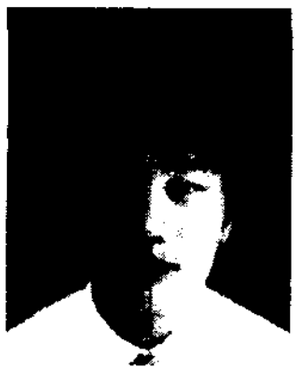
[5] Tiwari, A., Tomko, K.A., "Scan-chain Based Watchpoints for Efficient Run-time Debugging and Verification of FPGA Designs," Proc. Asia and South Pacific, Conf. on Design Automation, Jan. 2003, pp. 705-711.

[6] Kenneth P. Parker, "The Boundary Scan Handbook," Kluwer Academic Publisher, May 2003.

[7] S.Y. Hwang, K.S. Jhang, K. Yi, Y.S. Han, and S.Y. Ohm, "A Low Cost Verification Method for Digital System Design Targeting Rapid Prototyping FPGA Board," Proc. Int. Conf. on APIS, Jan. 2004, pp. 20-24.

황 수 연

2002년 한남대학교 컴퓨터공학과 졸업(학사). 2004년 충남대학교 대학원 컴퓨터공학과 졸업(석사). 2004년~현재 충남대학교 대학원 컴퓨터공학과 박사과정 2004년 9월~2006년 2월 한국전자통신연구원 IT융합/부품연구소 멀티미디어 SoC 설계팀 근무(온칩 네트워크 기반 SoC Platform 개발). 2006년 4월~2007년 12월 한국전자통신연구 원 이동통신연구단 초고속단말모뎀연구팀 근무(3GE 초고속 단말모뎀 개발). 관심분야는 시스템 온 칩 설계, 온 칩 버스 설계 및 검증, 이동 통신, 초고속 단말모뎀 설계 및 검증



강 동 수

2005년 충남대학교 정보통신공학부 졸업(학사). 2007년 충남대학교 대학원 컴퓨터공학과 졸업(석사). 2007년~현재 충남대학교 대학원 컴퓨터공학과 박사과정 관심분야는 시스템 온 칩 설계, 인공위성용 마이크로 프로세서 설계 및 검증, 컴퓨터 구조, 설계자동화

장 경 선

1986년 서울대학교 전자계산기공학과 졸업(학사). 1988년 서울대학교 대학원 컴퓨터공학과 졸업(석사). 1995년 서울대학교 대학원 컴퓨터공학과 졸업(박사). 1996년 3월~2001년 8월 한남대학교 컴퓨터공학과 교수. 2001년 9월~현재 충남대학교 컴퓨터공학과 교수. 관심분야는 컴퓨터 구조, 시스템 온 칩 설계 및 자동화

이 강

1990년 서울대학교 컴퓨터공학과 졸업(학사). 1992년 서울대학교 대학원 컴퓨터공학과 졸업(석사). 1997년 서울대학교 대학원 컴퓨터공학과 졸업(박사). 1998년 3월~1999년 2월 인제대학교 정보 통신공학과 전임강사. 1999년 3월~2001년 2월 한동대학교 전산전자공학부 전임강사. 2001년 3월~2005년 2월 한동대학교 전산전자공학부 조교수. 2005년 3월~현재 한동대학교 전산전자공학부 부교수. 2005년 9월~2006년 8월 미국 UCI 방문교수. 관심분야는 설계 자동화, Reconfigurable Computing, SoC Design