A Network Security Processor Design Based on an Integrated SOC Design and Test Platform

Chen-Hsing Wang
Dept. Elec. Eng.
NTHU
Hsinchu, Taiwan 30013
chwang@larc.ee.nthu.edu.tw

Chih-Yen Lo
Dept. Elec. Eng.
NTHU
Hsinchu, Taiwan 30013
cylo@larc.ee.nthu.edu.tw

Min-Sheng Lee
Dept. Elec. Eng.
NTHU
Hsinchu, Taiwan 30013
mslee@larc.ee.nthu.edu.tw

Jen-Chieh Yeh
Dept. Elec. Eng.
NTHU
Hsinchu, Taiwan 30013
jcyeh@larc.ee.nthu.edu.tw

Chih-Tsun Huang
Dept. Comp. Sci.
NTHU
Hsinchu, Taiwan 30013
cthuang@cs.nthu.edu.tw

Cheng-Wen Wu
Dept. Elec. Eng.
NTHU
Hsinchu, Taiwan 30013
cww@ee.nthu.edu.tw

Shi-Yu Huang
Dept. Elec. Eng.
NTHU
Hsinchu, Taiwan 30013
syhuang@ee.nthu.edu.tw

ABSTRACT

In this paper we present a generic Network Security Processor (NSP) design suitable for a wide range of security related protocols in wired or wireless network applications. Following the platform-based design methodology, we develop four specific platforms, i.e., architecture platform, EDA platform, Design-for-Testability (DFT) platform, and prototyping platform, for our NSP design. With these platforms, design of the NSP chip becomes more efficient and systematic. A prototype chip of the NSP has been implemented and fabricated with a 0.18μm CMOS technology. The chip area is 5mm$\times$5mm (with 1M gates approximately), including I/O pads. The operating clock rate is 80MHz. The best performance of the crypto-engines is 1.025Gbps for AES, 1.652Mbps for RSA, 125.9/157.65Mbps for HMAC-SHA1/MD5, and 2.56Gbps for random number generator. Comparison result shows that our NSP is efficient in terms of performance, flexibility and scalability.

Categories and Subject Descriptors

B.7.1 [**Integrated Circuits**]: Types and Design Styles—*Algorithms implemented in hardware, VLSI(Very Large Scale Integration)*

General Terms

Design, Security

Keywords

AES, RSA, HMAC-MD5, HMAC-SHA1, RNG, AMBA

1. INTRODUCTION

With the rapid development of the wired and wireless communication technologies, the demand for data security is rising rapidly due to more and more new applications introduced. Most of these applications use insecure public networks to transmit private data, e.g., the personal data, identification, and password in electronic banking and e-commerce, confidential information, etc. These activities involve significant security risks. Without a robust security measure, many applications will be greatly limited due to privacy and security concern. Several security protocols, therefore, have been proposed to address such kind of issues, e.g., SSL (Secure Socket Layer) [1], IPSec (IP Security) [2], and VPN (Virtual Private Network) [3], are commonly used today to provide the protection of confidential information/data. In order to guarantee security, such communication protocols generally adopt different cryptographic algorithms for user authentication, data encryption/decryption, and data integrity check. For instance, in the IPSec protocol, asymmetric-key cryptographic algorithm (RSA or ECC) is used for user authentication and key management, symmetric-key cryptographic algorithm (DES/3DES or AES) is used for data encryption/decryption, and hash function (HMAC-SHA1 or HMAC-MD5) is used for data integrity check.

Traditionally, security processing can be done by software that is executed on a general-purpose processor [4].

Although software implementation is flexible, the software approach may not be suitable for modern applications that require high data throughput rate. For that reason, hardware accelerators for certain cryptographic functions can be designed to enhance the processing performance, but a hardwired design lacks programmability and flexibility that is important in the consumer electronics market. In [5] a special purpose microprocessor is proposed to optimize the execution of cryptography algorithms. This approach provides better performance than the pure software implementation, and better programmability than the dedicated hardware approach. However, a special purpose microprocessor still suffers from long redesign cycle for heterogeneous security applications, which will increase the time-to-market, making it less competitive. In [6], we present an integrated architecture of a scalable Security Processor (SP) that provides four types of cryptographic functions. For different applications requiring different cryptographic functions, the architecture of our SP is designed to be capable of inserting new Crypto-Engines (CE) or removing unwanted ones with little effort. Based on the SP, we further develop the Network Security Processor (NSP) as reported in this paper. Using the platform-based design methodology, our NSP becomes more competitive in terms of flexibility and time-to-market, which supports various cryptographic functions and security requirements.

2. PROPOSED PLATFORMS FOR NSP

2.1 Target System Overview

Platform-based design has become a popular design style for system-on-chip (SOC) due to the fact that increasing circuit complexity, time-to-market pressure, and non-recurring engineering cost make the traditional ASIC design style infeasible [7]. Figure 1 shows a network processor system consisting of multiple Network Processor Cards (NPC) and the switch fabric. The NPC integrates a PHY module, a packet processor, our NSP, a host processor (e.g., an ARM core) and a switch fabric interface. We use the Advanced Microcontroller Bus Architecture (AMBA) system as our core integration platform. For many networking and communication applications, the host processor mainly manages the work flow of the system and executes system applications. The packet processor manipulates the ingress or egress packets from/to the PHY module or switching fabric interface; moreover, it also executes data compression, header modification, packet classification, packet framing, etc. Meanwhile, the NSP performs cryptography-intensive computation, such as user authentication, key management, and data encryption/decryption. The PHY module is a device to receive or transmit the electronic signal from/to the network transmission medium. Since the signal from/to the network transmission medium normally exists in analog form, the PHY module must handle signal transformation (between analog form and digital/binary form). The switch fabric is like a high speed switch providing the packet routing from one NPC to another.

One of the objectives of this work is to provide a scalable and extensible architecture platform. We also integrate the host processor and our previous security processor (SP) design into this platform. Our proposed EDA (Electronic Design Automation) platform supplies a complete software environment for the NSP design, making it more efficient and

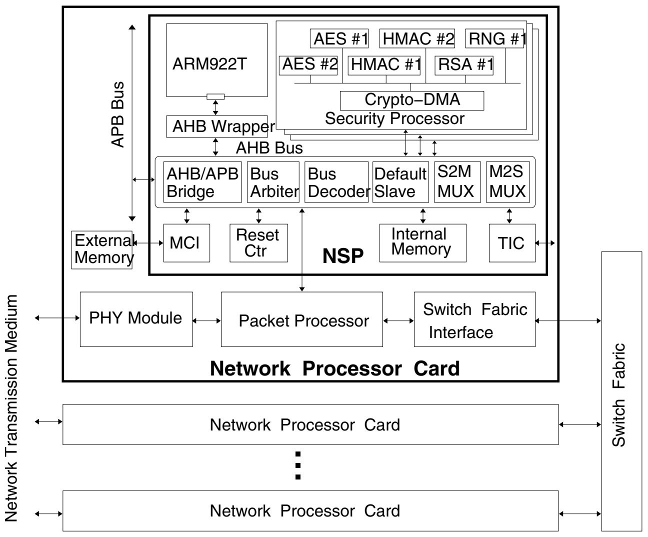

Figure 1: The generic network processor system.

competitive. The testing issues of the embedded logic and memory cores as well as the whole SOC are solved by the Design-for-Test (DFT) platform. Memory BIST (Built-In Self-Test) and SOC test circuits are inserted automatically by our tools, reducing the design time dramatically. The prototyping platform accelerates the verification of the NSP function in the design stage. Moreover, it demonstrates the feasibility of the NSP design.

2.2 Architecture Platform

The NSP consists of a host processor, multiple SPs, a TIC (Test Interface Controller) module, an MCI (Memory Control Interface) module, a system reset controller, and basic components of the AHB bus (see Fig. 1). The SP reported in our previous work [6] provides four popular cryptographic functions: AES, RSA, HMAC-SHA1/MD5 and random number generation. A DMA-like controller, called crypto-DMA, is responsible for controlling the internal CEs and to transfer the processing data between CEs and internal/external memory. The control method of the crypto-DMA is designed like a DMA device, hence it inherits the advantages of the descriptor-based DMA devices. The control overhead of the host processor is alleviated dramatically to the need of creating a descriptor for each request of the cryptographic operation. The descriptor will then guide the SP to process the cryptographic operations. In order to manipulate multiple integrated CEs, a micro-controller is implemented in the crypto-DMA to manage the control flows and interpret the descriptors. For newly integrated CEs, the designer only needs to add micro-codes in the micro-program ROM without modification of the crypto-DMA module. The TIC is used to provide core-vendor supplied test patterns for the ARM core. Finally, the MCI is a memory management unit that processes the external memory accesses.

The platform is flexible when handling various protocols since this part is done by software. Thus, the same hardware architecture is capable of processing heterogeneous network applications by implementing different software programs. In addition to the flexibility, the platform is also extensible and scalable with respect to performance. Multiple SPs can be used in parallel to speed up data processing. Furthermore, the architecture of the SP itself is also scalable. Design parameters in the SP, such as the number of CEs, the number of crypto-channels, and the number of internal AHB buses are all scalable and configurable to meet different performance/area trade-offs for different applications. To further enhance the system performance, the packet processor can be integrated to speed up the packet processing. It also reduces CPU workload.

2.3 EDA Platform

The proposed EDA platform is shown in Fig. 2, which provides an integrated environment for the CAD tools and other software used in the NSP chip implementation. The platform can be divided into three major parts: (1) ASIC design flow, (2) FPGA prototyping flow, and (3) Memory BIST & SOC test insertion flow. The EDA platform provides a complete software environment for the NSP design, which significantly reduces the development time and effort.

We will describe the FPGA prototyping flow and Memory BIST & SOC test circuitry insertion flow in the following sections. The ASIC design flow starts from the design specification phase and ends at the GDSII sign-off phase. It includes the commercial CAD tools and our in-house tools (highlighted in Fig. 2). Once the design specification is determined, architecture evaluation is performed. A fast evaluation method using analytical models is proposed in [8], which is used to help determine the optimized configuration parameters of the SP—the number of CEs, the number of crypto-channels, and the number of internal AHB buses—in the early design stage. From the evaluation result, we can obtain an implementation specification of the hardware/software partitions.

2.3.1 *Software Partition*

The software partition is the software program to generate the descriptors for cryptographic operation requests. Different cryptographic operations are implemented in different function calls to produce the relative descriptors. For example, in some applications requiring security process to protect private data, the user can invoke the function calls to perform cryptographic operations. Accordingly, the descriptors are created to guide the SP to accomplish the work. During the RTL simulation and verification phase, we write a test program to check all cryptographic functions. The program is compiled, linked, and assembled into an image file, then fed into the ARMulator to run the the overall system simulation with other system components.

2.3.2 *Hardware Partition*

The hardware partition of the NSP involves modification of the SP, system integration, and simulation. According to the implementation specification, we perform minor modifications on the SP to meet the specification. Since our own CEs are all wrapped with the AHB bus interface, inserting them into or removing them from the on-chip AHB bus becomes straightforward. Therefore, the integration and verification of the SP requires little effort. To further minimize the design overhead, we also develop an automatic AHB wrapper generator. The AHB interface can then be easily generated for both in-house and third-party cores. In RTL simulation, we can dump the simulation waveform to observe errors and correct the SP function and that of the overall system. In addition, we check the coding style and test bench coverage by nLint and VNavigator. Since RTL simulation of the entire system takes unacceptably long time, we use the prototyping platform to speed up the system verification process.

2.3.3 *NSP Synthesis Flow*

The synthesis phase is partitioned into three steps with the consideration of inserting the SOC test circuitry and applying the vendor-provided synthesis/design scripts. In the first step, the individual CE in the SP is synthesized with scan chains. Meanwhile the ATPG tool is used to generate the scan patterns. The CEs with netlist and test information are submitted to our STEAC (SOC TEst Aid Console) framework [9] to generate their IEEE 1500 test wrappers and patterns. In the second step, the SP module is synthesized excluding its internal CEs, with SP-level scan chains inserted. The internal CEs are integrated in a core-based style in this step. In the final step the integration script is used to synthesize the system-level components, e.g., the ARM wrapper, bus arbiter, bus decoder, TIC, etc. System-level (glue-logic) scan chains are inserted excluding the consideration ARM core and SP module.

One of the important in-house tools in the EDA platform is the mixed-level power estimator, called PowerMixer [10]. The PowerMixer allows fast and accurate power estimation from the RTL to gate-level netlist. With the cooperation of the static and dynamic power/timing analysis tools at the gate level, our PowerMixer facilitates the power evaluation of the entire design efficiently in the early design stage. The tool also helps rapid reconvergence in power optimization.

The physical implementation of the NSP also follows the core-based design methodology. Floorplanning, placement, clock tree synthesis and automatic routing are accomplished by Synopsys Astro. Physical verification such as DRC/ERC check and LVS check is done by Mentor Graphics Calibre. Both the power and timing analyses are performed again to justify the final implementation result. The core-based design and DFT style of the NSP effectively minimizes the complexity of the system-level integration and optimization.

2.4 DFT Platform

The DFT platform provides an SOC test integration methodology, based on our previously proposed STEAC framework [9]. We adapt the framework for the NSP architecture. There are four different test requirements: (1) The ARM processor core: In our DFT platform the ARM core is tested via the TIC because this is the recommended test methodology suggested by the core vendor. The test vectors for the processor core can be applied by the AHB bus via the TIC. (2) The SP and its cores: All of the CEs in the SP are DFT ready after the fist synthesis phase. In addition, the SP has its own test architecture [6]. In this case we almost reuse the legacy test architecture and test patterns, only moving the TACS (Test Access Control System) unit from the SP module to the system level. The STEAC is utilized for test integration and test pattern translation. Test reuse effectively reduces the test development time. (3) The system-level components: The system-level components, including the TIC, MCI, system reset controller, AHB bus components, and other user-defined modules use scan test thanks to their direct accessibility. (4) Memory cores: For all small memory instances in the CEs and other IP cores,

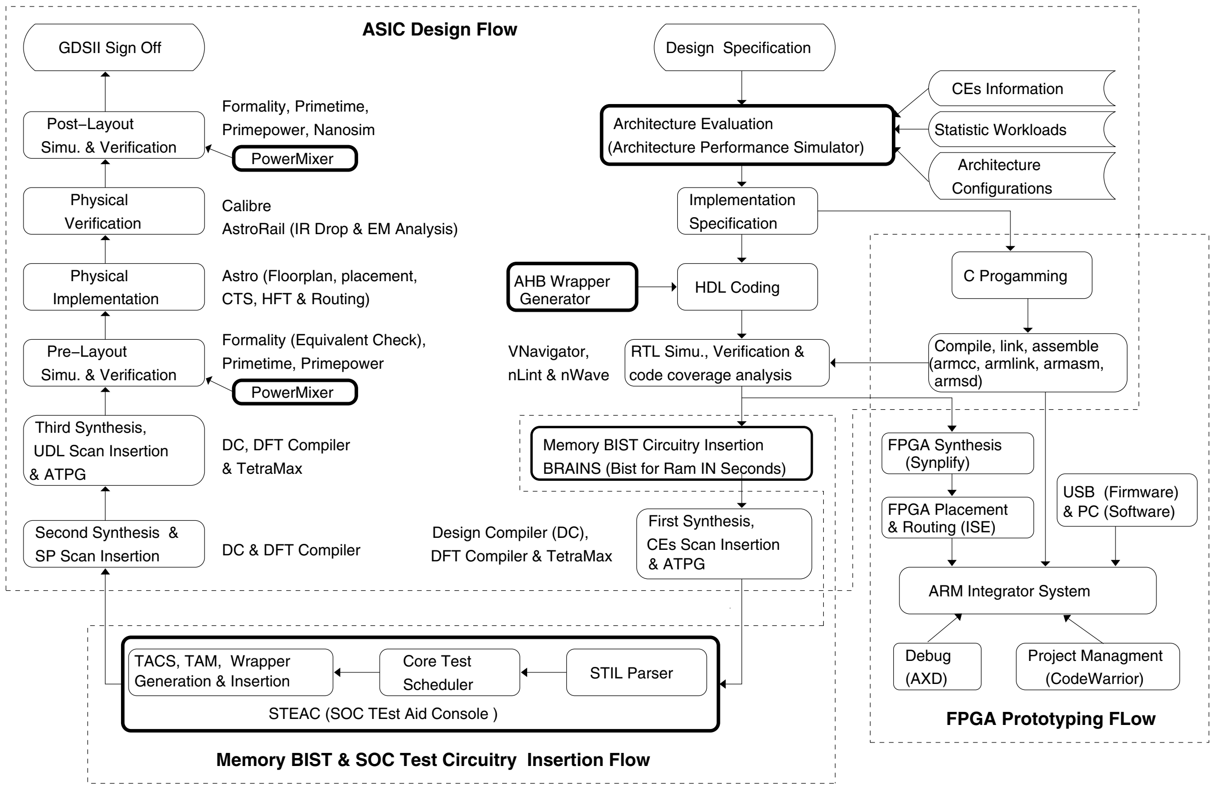
ASIC Design Flow
GDSII Sign Off
Post–Layout Simu. & Verification
Formality, Primetime, Primepower, Nanosim
PowerMixer
Physical Verification
Calibre
AstroRail (IR Drop & EM Analysis)
Physical Implementation
Astro (Floorplan, placement, CTS, HFT & Routing)
Pre–Layout Simu. & Verification
Formality (Equivalent Check), Primetime, Primepower
PowerMixer
Third Synthesis, UDL Scan Insertion & ATPG
DC, DFT Compiler & TetraMax
Second Synthesis & SP Scan Insertion
DC & DFT Compiler
Design Specification
CEs Information
Statistic Workloads
Architecture Configurations
Architecture Evaluation (Architecture Performance Simulator)
Implementation Specification
AHB Wrapper Generator
HDL Coding
VNavigator, nLint & nWave
RTL Simu., Verification & code coverage analysis
Memory BIST Circuitry Insertion BRAINS (Bist for Ram IN Seconds)
Design Compiler (DC), DFT Compiler & TetraMax
First Synthesis, CEs Scan Insertion & ATPG
C Progamming
Compile, link, assemble (armcc, armlink, armasm, armsd)
FPGA Synthesis (Synplify)
FPGA Placement & Routing (ISE)
USB (Firmware) & PC (Software)
ARM Integrator System
Debug (AXD)
Project Managment (CodeWarrior)
FPGA Prototyping FLow
TACS, TAM, Wrapper Generation & Insertion
Core Test Scheduler
STIL Parser
STEAC (SOC TEst Aid Console)
Memory BIST & SOC Test Circuitry Insertion Flow

Figure 2: EDA platform.

internal memories, and the large external memory, BIST is use as the test solution. The BRAINS framework [11] is used to generate the memory BIST circuitry automatically. Finally, the TACS is implemented as the SOC test controller to manipulate the test application at the system level.

2.5 Prototyping Platform

A prototyping platform is constructed to demonstrate the feasibility of the proposed architecture platform. It also accelerates the verification of the NSP with new configuration parameters or newly developed CEs. Figure 3 depicts the block diagram of the prototyping platform, consisting of the hardware, software, and firmware components. The hardware includes a desktop PC, a USB development board and an ARM Integrator system with the logic module (FPGA) and the core module (ARM922T processor). The software on PC includes the USB development environment, and the ARM Developer Suite (ADS) includes a project management tool (CodeWarrior) and a debugger (AXD). The firmware component is mainly the USB firmware. The GPIF (General Programmable Interface) provides a highly configurable and glueless IO interface that allows the highest possible bandwidth between the USB board and our system. We design a GPIF-AHB wrapper to bridge the PC and the AHB bus. Then, data transmission between the PC and the platform can be done by the USB interface. In our experiment, the ARM core handles the generation of the descriptors to control the SP. The ARM program is compiled and downloaded into the ARM 922T processor by the MultiICE. During the experiment, AXD provides a debug window that helps monitor the values in the memory and the status of the SP when the entire system is running. With this platform, we can successfully accelerate the encryption and decryption operations while comparing with the security applications using pure software.

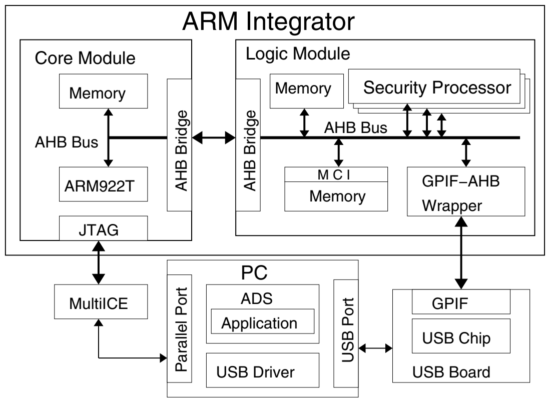

Figure 3: Prototyping platform.

The NSP demo system is shown in Fig. 4. The SP that is originally implemented by FPGA (see Fig. 3) is now replaced with the ASIC chip developed in [6]. The SP chip is put in the socket on the customized daughter board connected to the logic module. The logic module is at the bottom of the SP board. In order to prove the feasibility of our NSP, a real network application (secure web service) is modified and ported to the demo system. The demo system runs an Apache-SSL web service. The operating system and the application are executed on the PC, while the cryptographic functions are manipulated by the NSP prototype system. When there is a secure data access from the remote client (e.g., a PC), the encryption and decryption will be accelerated by the hardware. The host processor (ARM core) in the NSP is in charge of the communication between the PC and the SP by generating the descriptors to control the cryptographic engines. Once the embedded operating system is ported, the NSP can demonstrate the entire Apache-SSL service. Other network security protocols and applications can be implemented easily as well in this demo system.

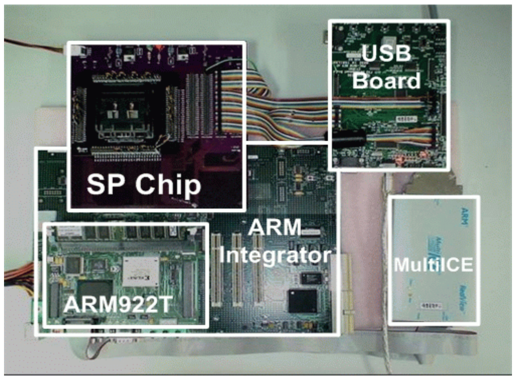

Figure 4: Demo system.

We next show the evaluated performance improvement for the SSL security protocol through the use of our prototyping platform. The SSL, similar to IPSec, uses a combination of symmetric-key and asymmetric-key algorithms to secure the data transmission between a client and a server. The SSL utilizes asymmetric-key algorithm, such as RSA, for the client authentication. Then, the server creates the symmetric keys that are used for fast encryption/decryption of the bulk data. In our evaluation, we choose AES and RSA as the symmetric-key and asymmetric-key algorithms, respectively. The operating clock rate of our prototyping platform is 10MHz while the software program is executed on a 2.8GHz computer with 1GB memory. Given the different session sizes from 1KB to 32KB, we evaluate the execution times using the pure software method and ours. Figure 5 shows the evaluation result. Our prototyping platform improves the overall performance about 1.99× to 3.22× although the operating clock rate is only 10MHz.

3. IMPLEMENTATION RESULTS

Following our integrated SOC design and test methodology, an NSP test chip has been implemented and fabricated with a 0.18μm CMOS technology. The internal/external memories are not integrated into this test chip due to the limited area budget. Table 1 summarizes the area statistics of the NSP, excluding the hard ARM core. The SP in [6] is reused with minor reduction of the I/O buffer size in the AES engines. Therefore the area of the SP listed in Table 1 is a little smaller than the previous one [6]. For the demonstration consideration, we also integrate the GPIF-AHB wrapper into this chip.

In Table 2 we show the comparison among different cryptographic processors. In [4], the authors report the performance results of the symmetric-key algorithms executed on

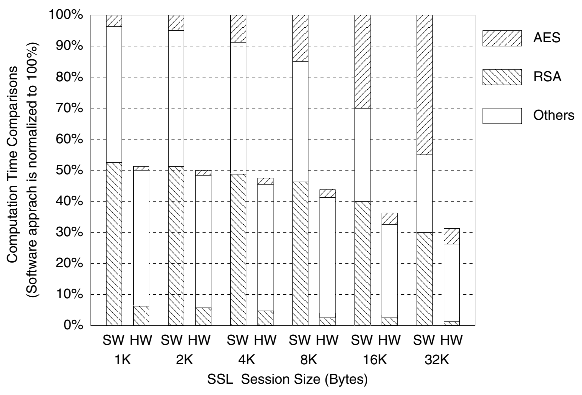

Figure 5: Performance comparison for various SSL session lengths.

Table 1: Area Statistics of NSP Chip

Module Name	Gate Count	%
SP	486,461.67	93.43%
GPIF-AHB	2,789.08	0.54%
TACS & TAM	667.53	0.13%
Test Wrapper	22,507.47	4.32%
AHB Bus Components	8,239.08	1.58%
Total	520664.83	100.00%

the Alpha 21264 processor and their proposed co-processor (CryptoManiac). In [5] a special purpose microprocessor is proposed to support both the symmetric-key and asymmetric-key cryptographies. Compared with previous works, our design outperforms in terms of performance for both the AES and RSA cryptographies. In addition, the scalable architecture allows the extension of multiple heterogeneous CEs. Our test chip consists of four different kinds of CEs, e.g., two AES, one RSA, two HMAC SHA-1/MD5, and one random number generator.

A layout view of the the NSP chip and its floorplan is depicted in Fig. 6. The total chip area is 5mm×5mm, including I/O pads. The core area is only 4.1mm×4.1mm. The equivalent gate count is about 1M gates including the ARM core. The operating frequency is 80MHz as measured by ATE. Total power consumption is about 296mW, excluding the ARM core.

Table 2: Performance Comparison

	Function	Clock	Performance
[4]	AES	360MHz	512Mbps (CryptoManiac)
		600MHz	227.6Mbps (Alpha 21264)
[5]	AES	58MHz	390Mbps
	RSA	28MHz	153Kbps (16-bit Key)
			17Kbps (1024-bit Key)
Ours	AES×2 (ECB/CBC Modes)	80MHz	1025Mbps×2 (128-bit Key)
			856Mbps ×2 (192-bit Key)
			731Mbps×2 (256-bit Key)
	RSA	80MHz	1652Kbps (16-bit Key)
			26Kbps (1024-bit Key)
	HMAC×2	80MHz	101-126Mbps×2 (HMAC-SHA1)
		80MHz	126-158Mbps×2 (HMAC-MD5)
	RNG	80MHz	2.56Gbps

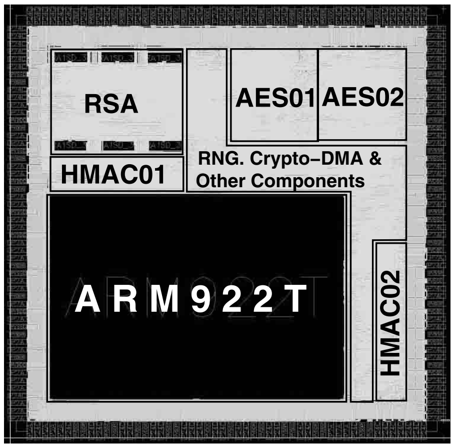

Figure 6: Layout view of NSP.

4. CONCLUSIONS

We have presented a network security processor (NSP) design, which is implemented by using the proposed integrated SOC design and test environment that consists of the architecture, EDA, DFT, and prototyping platforms. The architecture platform provides a flexible and extensible architecture for security related communication applications. The EDA platform supports core-based SOC design, from early-stage performance and architecture evaluation to timing and power analysis during RTL and circuit design, as well as the verification in physical implementation. The EDA platform also integrates our in-house tools into the design flow with commercial tools. Test automation for SOC, such as memory BIST insertion and SOC test integration, is implemented by our DFT platform. Finally, the prototyping platform facilitates system development and verification. A demo system running the Apache-SSL web server application has been shown to justify the feasibility of the NSP. A test chip has been implemented using a 0.18 μm CMOS technology, whose area is 5mm$\times$5mm, including I/O pads (1M gates approximately), and the clock rate is 80MHz. Results show that our NSP outperforms others in terms of performance, flexibility, and scalability.

5. REFERENCES

[1] A. Frier, P. Karlton, and P. Kocher, *The SSL Protocol Version 3.0*, http://wp.netscape.com/eng/ssl3/draft302.txt, Netscape, Nov. 1996.

[2] S. Kent and R. Atkinson, *Security Architecture for the Internet Protocol*, IETF Network Working Group, 1998, rFC 2401.

[3] P. Fergguson and G. Huston, "What is a VPN?—Part I," *The Internet Protocol Jour.*, vol. 1, no. 1, pp. 2–19, June 1998, http://www.cisco.com/warp/public/759/.

[4] L. Wu, C. Weaver, and T. Austin, "CryptoManiac: A fast flexible architecture for secure communication," in *Proc. 28th Ann. Int'l Symp. on Computer Architecture*, 2001, pp. 110–119.

[5] H.-W. Kim and S. Lee, "Design and implementation of a private and public key crypto processor and its application to a security system," *IEEE Trans. on Consumer Electronics*, vol. 50, no. 1, pp. 214–224, Feb. 2004.

[6] C.-P. Su, C.-H. Wang, K.-L. Cheng, C.-T. Huang, and C.-W. Wu, "Design and test of a scalable security processor," in *Proc. Asia and South Pacific Design Automation Conf. (ASP-DAC)*, Shanghai, Jan. 2005, pp. 372–375.

[7] A. Sangiovanni-Vincentelli, L. Carloni, F. D. Bernardinis, and M. Sgroi, "Benefits and challenges for platform-based design," in *Proc. IEEE/ACM Design Automation Conf. (DAC)*, San Diego, June 2004, pp. 409–414.

[8] Y.-C. Lin, C.-W. Huang, and J.-K. Lee, "System-level design space exploration for security processor prototyping in analytical approaches," in *Proc. Asia and South Pacific Design Automation Conf. (ASP-DAC)*, Shanghai, Jan. 2005, pp. 372–375.

[9] C.-W. Wang, J.-R. Huang, K.-L. Cheng, H.-S. Hsu, C.-T. Huang, C.-W. Wu, and Y.-L. Lin, "A test access control and test integration system for system-on-chip," in *Sixth IEEE Int'l Workshop on Testing Embedded Core-Based System-Chips (TECS)*, Monterey, California, May 2002, pp. P2.1–P2.8.

[10] Y.-F. Lee, S.-Y. Huang, S.-Y. Hsu, I.-L. Chen, C.-T. Shieh, J.-C. Lin, and S.-C. Chang, "Power estimation starategies for a low-power security processor," in *Proc. Asia and South Pacific Design Automation Conf. (ASP-DAC)*, Shanghai, Jan. 2005, pp. 367–371.

[11] C. Cheng, C.-T. Huang, J.-R. Huang, C.-W. Wu, C.-J. Wey, and M.-C. Tsai, "BRAINS: A BIST complier for embedded memories," in *Proc. IEEE Int'l Symp. on Defect and Fault Tolerance in VLSI Systems (DFT)*, Yamanashi, Oct. 2000, pp. 299–307.