

CONPUTER – A FRAMEWORK OF INTRUSION-FREE SECURE COMPUTER ARCHITECTURE

Shuangbao Wang

George Mason University
Fairfax, Virginia, U.S.A.

Fengjing Shao

Qingdao University
Qingdao, China

Robert S. Ledley

Georgetown University
Washington, DC, U.S.A.

Abstract - We propose a new type of secure computer architecture that can enable computers to prevent intruders from obtaining data stored in the computer system. Based on the modified Neumann model as we proposed with a recent pending patent, the network communication component is separated from the other parts of the computer system with a separate system bus. Data exchange between the two system buses can only be performed through the Bus Controller via a command issued by the computer operator. At any time, data stored in this computer can only be accessed by the computer operator. The system ensures that unauthorized operations and communications cannot access system and user data. The main system resources cannot be compromised or taken over from the outside network. Identity theft cannot happen on this new type of secure computer system. Preliminary tests show that the system can secure data and prevent all Spyware programs we have tested from obtaining information from the computer.

Keywords: security, computer architecture, Neumann model

1. INTRODUCTION

Computers nowadays are very easy to be intruded via network especially through Internet. Therefore, information stored in a computer such as ssn, credit cards, bank accounts, and personal privacy information etc. is vulnerable to computer hackers.

Firewalls in some extends can prevent information stored in a computer be stolen. However it can only effective in a certain period of time. Some firewalls are mere software; some others even though use “hardware” to setup a “wall” between the computer and the outside world, the core components are based on algorithms or in other word software. On the other hand, a commercial firewall is not designed to use on personal computers or handheld devices. Using firewall cannot guarantee that the information stored on a computer will never be stolen.

Privacy is one of the biggest concerns nowadays. Some employers use centralized monitoring software to

monitor employee’s emails and other private information.

Identity theft is a more serious problem which draws attentions recently by the Congress. Nearly 10 million people were victimized by identity theft in the year 2004, according to Time magazine, the lost reached 5 billion. In early March 2005, the nation’s largest data miner ChoicePoint with 19 billion data files include driver’s license, ssn, credit history, birth certificate, real estate deed, and even thumbprint and DNA was broken into and some 145,000 people’s data was extracted.

As U.S. Senator Orrin Hatch said, “Identity theft is a serious problem that has drawn much attention recently in Congress. As we know, the damage caused can go beyond money and privacy and become a real threat to our national security.”

2. RELATED WORK

There are many researches related to the secure computer architecture area. Largman et al. [1] proposed “automatically create multiple sequentially or concurrently and intermittently isolated and/or restricted computing environments method to prevent viruses, malicious, or even computer or device corruption and failure”. According to this method, untrusted content is only exposed in the user processor logic environment in a temporary storage. The question remains for this method is how to determine which content is trusted and which is not. There might be a pre-determination process. Another problem is “concurrently computing environment” reduces the processing power dramatically. It is generally not suitable for PCs.

Anderson [2] put “removable trusted (hardware) gateway devices” between each of the inputs/outputs and the bus to secure the file transmission. As described, the approval of access the data is depended on a so-called “LOCK”. Once the lock is stolen, intercepted or hacked, sensitive data is then open to those hackers.

Hewlett-Packard has been working on a new type of Secure Platform Architecture (SPA) [3]. It is a set of software interfaces built on top of HP’s Itanium-based product line. SPA will enable operating systems and device drivers to be run as unprivileged tasks and will allow services to be authenticated and identified. The

architecture will also let enterprises run multiple operating system images--for public and private infrastructure, for example--on the same security platform. The problem exists in the SPA is that, as the company described, it uses a set of software interfaces to authenticate and identify the tasks. Once the system is compromised, SPA will not be able to function well.

Sean W. Smith and Steve Weingart developed a prototype using a high-performance, programmable secure coprocessor [4]. It is a type of software, hardware and cryptographic architecture. The research involved in the occurrence of configuration and maintenance in a hostile environment, the recovery from the vulnerabilities that emerge in complex software, the inability of the device to be opened or examined, different and mutually suspicious origins and the need for a reloadable on-card software. It addressed some issues especially how to secure programs running on coprocessors and system recovery. In term of secure information and data, there are lots of works need to be done.

Recently, MIT researchers proposed secure processors that enable new applications by ensuring private and authentic program execution even in the face of physical attack [6].

So far, many current researches may have some impacts to reduce the risk of information theft in one way or another. However those solutions have not solved the information security problems thoroughly due to the limitation of the computer architecture they used. We have found that there is a problem exists in John von Neumann computer architecture model – the foundation of computer architecture. If this problem is not solved, information stored on a computer will hardly be secure.

3. TECHNICAL OBJECTIVES

The main goal of this research is to propose a new type of secure computer architecture that can not only monitor the system security but also enable computers to prevent intruders from getting data stored in the computer system. In a recent pending patent, we proposed a new computer architecture model – *Connputer*. Based on this new model, the network communication component is separated form the other parts of a computer system with a separate system bus. In a computer system, all components except network reside on one system bus, and the network interface resides on a separate system bus. Data exchange between those two system buses can only be performed through the Bus Controller via a command issued by the computer operator. So, data stored in this computer (main storage) can only be accessed by the computer operator. In other words, user data is isolated form outside networks and therefore cannot be accessed even

the computer is compromised or taken over from outside networks.

In addition to preventing information theft, the system contains a security agent that can monitor and report any security related events. The recorded security events can be transmitted to or viewed by the central monitoring system in real-time.

A prototype of this computer - *Connputer* has been developed and initial experiments show that the system is very promising. The major technology breakthrough is that it can prevent unauthorized access of any information in the computer system. Security is guaranteed as the system is implemented using this secure computer architecture.

4. IMPLEMENTATION OF THE CONNPUTER

Before starting our research, we first studied the widely used John Neumann computer architecture model. Then, we modified the Neumann model and proposed a new secure computer architecture model - *Connputer*. We further discussed the technical details of the design and how to implement the system. We used a co-processor, an FPGA and other digital circuits together with kernel software to construct the add-on security agent. In combing the add-on board with the current computer system, we designed multiport I/O and multiport memory interfaces. The process of taking over system buses for a CPU is performed through a Bus-Controller. The design of the Bus-Controller is also discussed in detail.

4.1 John von Neumann Computer Architecture Model

John von Neumann wrote "First Draft of a Report on the EDVAC" [5] in which he outlined the architecture of a stored-program computer. He proposed a concept that has characterized mainstream computer architecture since 1945. Fig. 1 shows the Neumann model.

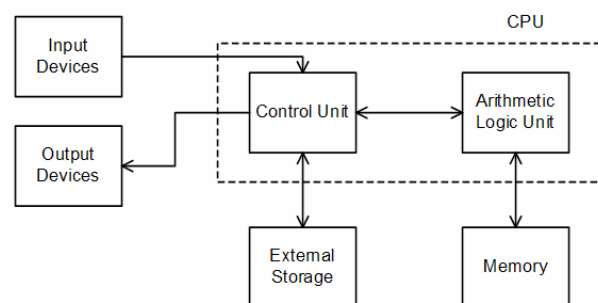


Figure 1: Block diagram of John von Neumann's computer architecture model.

A “system bus” representation of Neumann model is shown in Fig. 2. This is just another view of the Neumann model, with the introduction of the concept of Direct Memory Access (DMA).

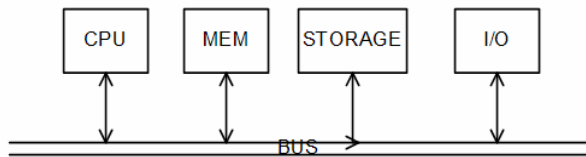


Figure 2: A “system bus” representation of the Neumann model. It is equivalent to Fig. 1 with the introduction of DMA.

Since 1990s, computer networks especially Internet has been wide spread around the world. Computers are no longer only being used to compute as a stand alone machine. The feature of information exchange through network is a vital component in today’s computers. Unfortunately John Neumann was not able to foresee this change. One can argue that we can consider network is part of input/output device which is already included in the Neumann model. However, the network interface is so important that it is not appropriate to classify it as in the general I/O device category. Furthermore, an I/O device in Neumann model refers to those devices such as a keyboard, a display and a printer etc. which are used for direct interact with the computers. Now, the way people use a computer is quite different than that of sixty years ago. So a modification of Neumann’s computer architecture model is necessary to reflect this change. Fig. 3 shows the modified Neumann model. In Fig. 3, a network unit (interface) is added to the computer system bus so that the I/O unit only deals with input and output devices such as keyboard, mouse, display etc. Separating network unit from the general I/O offers great advantages.

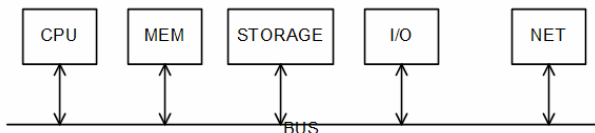


Figure 3: Modified Neumann computer architecture model. Here network interface is added to the Neumann model and is separated from the general input and output devices.

The Neumann model is so dominant that few people dare challenge it since its birth in 1945. However if we look into the Neumann model from security perspective, we could find out that it does have some drawbacks.

In the Neumann model, CPUs, Memory, I/O, external storage and network interface are all connected to one single system bus that includes control bus, data bus and address bus. Once intruders break into the system from any network locations, they can totally take over the computer system and do whatever they want.

For the Neumann model, the concept of CPU is a centralized control and arithmetic unit. Even though nowadays a computer with multiprocessors is very common, however those processors are merely coordinated together by software to perform one task or a series of tasks. In other words, they share the same system bus. Intruders can still take over the whole system once they break into the system from any network ports.

4.2 Architecture of Computer

The pilot computer we implemented is an intrusion-free, information and data secure computer system. It consists of:

1. Two zones (red zone and green zone) with two separated system buses,
2. The network interface that is only attached on one bus in red zone,
3. Each bus have its own CPU and private memory,
4. Main (protected) external storage is attached only on one bus in green zone,
5. One cache storage (temporary external storage or dual-port external storage) is connected to both internal system buses via a Bus Controller,
6. A Bus Controller connects two internal system buses between the red zone and green zone,
7. Input and output devices such as keyboard, mouse and display etc.

Even though a network interface can be considered as an input/output device, add this interface to the system bus and separate it from other parts (even the I/O port) has many advantages. The modification made it possible for this research to isolate network from other parts within a computer system while data can still be transmitted through the network.

Fig. 4 depicts a functional block diagram of such intrusion-free, information and data secure computer system architecture. Normally the computer is in the state of green zone where all computation works are performed. In green zone, network is disabled. When data transmission is needed, the Bus Controller switches to red zone where another CPU is taken over the job. In red zone, there is no external storage, all data is stored on cache storage via the Bus Controller. The Bus Controller is managed by the computer operator.

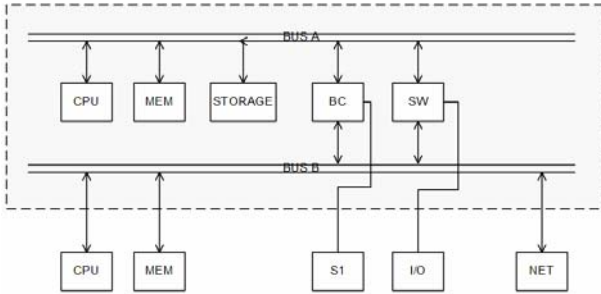


Figure 4: Block diagram of intrusion-free computer architecture. User data is stored on the main storage which will never expose to the network.

Looking from network side (outside), this intrusion-free, information and data secure computer has one or more CPUs, internal memory, input/output devices such as a keyboard and a mouse, network ports (Ethernet or wireless) and cache storage. Because the red zone only deal with the network communication, suppose a hacker break into the system from the Internet, what the hacker will see is just the temporary data on the cache storage and maybe some of the system data. It is impossible for the intruder to see data on the main (protected) storage.

Fig. 5 is the block diagram of the Bus Controller. Bus A in green zone can access the cache storage only if the EN 1 signal is enabled. Similarly, the Bus B from the red zone can access the cache storage only if the EN 2 signal is enabled. Notice that EN 1 and EN 2 are controlled by the computer operator. Intruders can not make any enable actions without direct operating the computer.

Computer operators can automatically enable the data access to the cache storage. To automatic enable the data access to the cache storage, an operator set default to Bus A (green zone) so that data can be accessed directly from/to the cache storage. When network communication is needed such as launching an internet explorer, the EN 2 is automatically enabled so that Bus B is connected and Bus A is disconnected from the system so that main storage is isolated from the system.

A multiport interface is used to switch the keyboard/mouse and display devices between those two buses automatically. For automatic switching, the switching process is synchronized with the Bus Controller.

Combining the cache storage or temporary external storage with the Bus Controller forms the dual-port storage which can be accessed by two computer system buses. It is different from so-called dual-port external storage devices which for example have one USB port and one FireWare port. In our case you cannot just attach the device to two system buses without synchronize them.

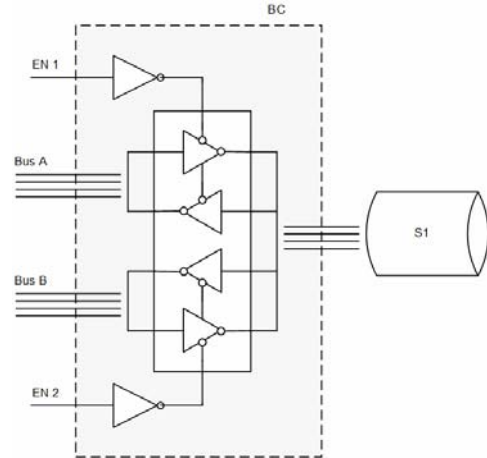


Figure 5: Block diagram of Bus Controller that connects two buses and a cache (dual-port) storage device for data exchange between the red zone and green zone.

When the cache storage is attached onto Bus A in green zone, the files are displayed and then the trusted files are ready to be copied to the main storage. After the operation, the cache storage is erased. User data can then be copied to the cache storage if network transmission is further required. When the cache storage is switched to the Bus B in red zone, the data is displayed and is ready to be transmitted. Data download from network or Internet can then be stored on the cache storage. All data have to pass through the Bus Controller which is controlled by the computer operator.

4.3 FPGA and Co-processor Board

The coprocessor board is used to monitor system security and protect main storage areas from being accessed by unauthorized uses. It contains a coprocessor, an FPGA (Field Programmable Gate Array), flash memory, multiport memory interface, multiport I/O interface, a Bus Controller and kernel program. The kernel program also coordinates the communication between the add-on board and the current computer system.

LatticeXP FPGA devices provide security by eliminating the need for an external configuration bit-stream and by providing non-volatile security features.

The LatticeXP architecture contains an array of logic blocks surrounded by programmable I/O Cells (PIC). Interspersed between the rows of logic blocks are rows of sysMEM bedded Block RAM (EBR).

On the left and right sides of the PFU array, there are Non-volatile Memory Blocks. In configuration mode this nonvolatile memory is programmed via the IEEE 1149.1 TAP port or the sysCONFIG™ peripheral port. On power up, the configuration data is transferred from the Non-volatile Memory Blocks to the configuration SRAM. With this technology, expensive external

configuration memories are not required and designs are secured from unauthorized read-back. This transfer of data from non-volatile memory to configuration SRAM via wide busses happens in microseconds, providing an “instant-on” capability that allows easy interfacing in many applications.

There are two kinds of logic blocks, the Programmable Functional Unit (PFU) and Programmable Functional unit without RAM/ROM (PFF). The PFU contains the building blocks for logic, arithmetic, RAM, ROM and register functions. The PFF block contains building blocks for logic, arithmetic and ROM functions. Both PFU and PFF blocks are optimized for flexibility, allowing complex designs to be implemented quickly and efficiently.

Each PIC block encompasses two PIOs (PIO pairs) with their respective sysIO interfaces. PIO pairs on the left and right edges of the device can be configured as LVDS transmit/receive pairs. sysMEM EBRs are large dedicated fast memory blocks. They can be configured as RAM or ROM. The PFU, PFF, PIC and EBR Blocks are arranged in a two-dimensional grid with rows and columns. The blocks are connected with many vertical and horizontal routing channel resources. The place and route software tool automatically allocates these routing resources.

Every device in the family has a JTAG Port with internal Logic Analyzer (ispTRACY) capability. The sysCONFIG port which allows for serial or parallel device configuration. The LatticeXP devices are available for operation from 3.3V, 2.5V, 1.8V and 1.2V power supplies, providing easy integration into the overall system.

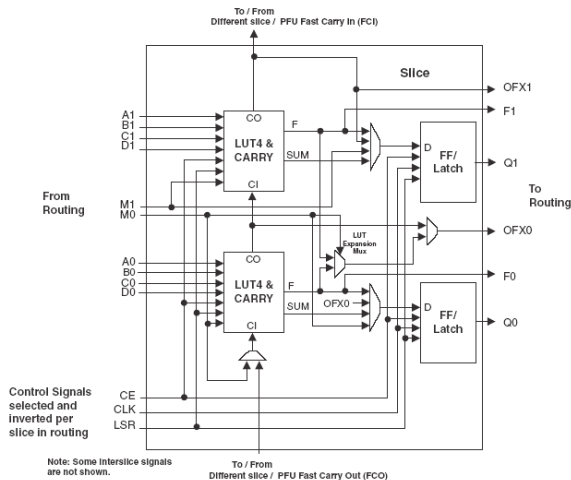


Figure 6: Slice Diagram

PFU and PFF Blocks

The core of the LatticeXP devices consists of PFU and PFF blocks. The PFUs can be programmed to perform Logic, Arithmetic, Distributed RAM and Distributed ROM functions. PFF blocks can be programmed to perform Logic, Arithmetic and ROM functions. Except where necessary, the remainder of the data sheet will use the term PFU to refer to both PFU and PFF blocks.

Slice

Each PFU block consists of four interconnected slices, numbered 0-3. Each slice contains two LUT4 lookup tables feeding two registers (programmed to be in FF or Latch mode), and some associated logic that allows the LUTs to be combined to perform functions such as LUT5, LUT6, LUT7 and LUT8. There is control logic to perform set/reset functions (programmable as synchronous/asynchronous), clock select, chip-select and wider RAM/ROM functions. Fig. 6 shows an overview of the internal logic of the slice. The registers in the slice can be configured for positive/negative and edge/level clocks.

The control logic provides the co-processor board with real-time security monitor and bus control.

4.4 Multi-port Memory Interface

Motorola’s MPC8260 is a chip that contains a 64-bit PowerPC microprocessor and a versatile communications processor module (CPM). The MPC8260 is used in a wide array of applications, especially those in the communications and networking markets. Examples include remote access servers, regional office routers, cellular base stations, and SONET transmission controllers.

A Lattice’s ispGDX2™ Generic Digital Crosspoint Switch is used as a multiport interface. The ispGDX2 device can interface the MPC8260 with an external master and a number of slaves including SDRAM and FLASH. The control logic for the SDRAM and FLASH is built in a CPLD which is used to interface the MPC8260 to the ispGDX2 device and to control the read/write to the memory. This function can be implemented in Lattice CPLDs.

The PowerPC core of the 8260 (the PowerPC 603e) can be replaced by other processors or ASIC. The memory controller within the MPC8260 is utilized in this design.

Fig. 7 shows in detail the function, internal logic and cross-connections that the ispGDX2 performs in the design. This section includes the signal list and descriptions of all signals used in this design and also provide a functional description of the design.

The multi-port interface coordinates two or more processors accessing the same I/O devices.

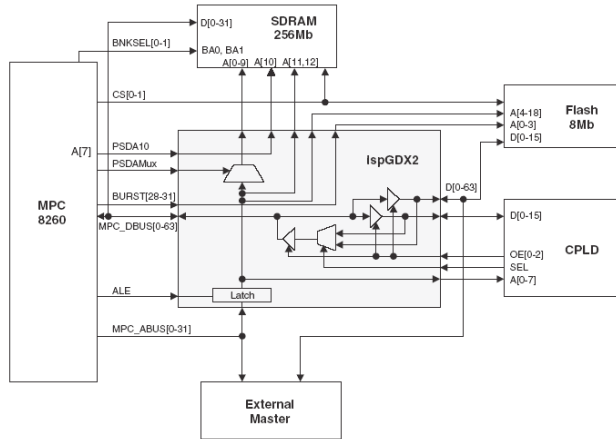


Figure 7: Detailed Functional Block Diagram

5. TESTING AND FURTHER DISCUSSION

We have designed and built a model machine – *Connputer* using the secure computer architecture we proposed. Preliminary tests show that it meets our design goals. We intentionally removed the Windows XP firewall and decline all security update. After one month period, we haven't found any incidents showing that it has been attacked by any hacker programs and tools such as Key Loggers, Spyware, Spyware cookies, Trojans, and Worms etc. We are now preparing tests by inviting some senior security personnel to remote "attack" this system. We expect the results will prove that the *Connputer* secure computer architecture model can enhance the computer security. And hopefully it can be adopted to modern personal computers.

We will further improve the architecture and look for the possibility to extend its scope from personal computers to server systems. In order to make the system to be widely used in the market, we will continue developing software that can make the system not only prevent intruder from getting information but also monitor the system security and capture and report any security related events.

REFERENCES

- [1] Kenneth Largman, Anthony B. More, Jeffrey Blair, "Computer system architecture and method providing operating-system independent virus-, hacker-, and cyber-terror-immune processing environments", U.S. patent: US 2004-0236874, USPTO, 2004
- [2] Anderson; Mark Stephen, "Secure computer architecture", U.S. patent, US 6,115,819, USPTO, 2000.
- [3] HP-Compaq Sets Platform Security, eWeek, June 3, 2002
- [4] Sean W. Smith, Steve Weingart, "Building a high-performance, programmable secure coprocessor", Computer Networks Vol.31, pp. 831–860, 1999
- [5] John von Neumann, "First Draft of a Report on the EDVAC", Moore School of Electrical Engineering, University of Pennsylvania, June 30, 1945
- [6] G.E. Suh, C.W O'Donnell, Ishan Sachdev, Srinivas Devadas, "Design and implementation of the AEGIS single-chip secure processor using physical random functions", Proceedings of 32nd International Symposium on Computer Architecture, ISCA '05, pp.25–36, 2005
- [7] Shuangbao Wang, "Intrusion-free Secure Computer Architecture for Information and Data Security", U.S. patent pending, March 2005