NOC-centric security of reconfigurable SoC

Jean-Philippe Diguet, Samuel Evain, Romain Vaslin, Guy Gogniat, Emmanuel Juin
Université de Bretagne Sud / CNRS, LESTER lab., France
Jean-Philippe.Diguet@univ-ubs.fr

Abstract

This paper presents a first solution for NoC-based communication security. Our proposal is based on simple network interfaces implementing distributed security rule checking and a separation between security and application channels. We detail a four-step security policy and show how, with usual NOC techniques, a designer can protect a reconfigurable SOC against attacks that result in abnormal communication behaviors. We introduce a new kind of relative and self-complemented street-sign routing adapted to path-based IP identification and reconfigurable architectures needs. Our approach is illustrated with a synthetic Set-Top box, we also show how to transform a real-life bus-based security solution to match our NOC-based architecture.

1 Introduction

As a key issue for multiprocessor chip design, a lot of attention is being payed to communications architectures. With properties such as scalability, bandwidth increase and high-level formalization networks on chip have emerged as interesting alternatives to bus hierarchies. Many research efforts are still necessary, such as the management of QoS in the context of chip with heterogeneous clocks implementing applications with dynamic behaviors, however NoC could introduce new security challenges in future embedded systems. A multiplication of communication links within chips also means an increase of doors to program instructions and sensitive data. In this paper we show how this drawback can be turned into opportunities to efficiently implement security policy associated to communication controls. First, we introduce the context of our solution in the next section. In section 3, we present the main differences between recent work in bus-based security and our solution to face considered attacks. In section 4, we describe our architecture model for Secure Network Interface (SNI) and our new routing algorithms for path-based identification and backward path computation. Section 5 illustrates our solution with a Multimedia/Set-top box case study, we also apply our approach to a bi-processor example from [9]. Finally we conclude about security cost and perspectives in terms of monitoring.

2 Communication security

2.1 Model of threat

Authors of embedded devices attacks have different motivations and frameworks to proceed. End users may try to extract, from the firmware of their own device, information for hacking licenses or network accesses. Hackers aim to modify behaviors of remote systems for different reasons and operators may be interested by reading personal data for marketing or licensing purposes. All these attacks can be initiated through abnormal communications. Considering SOC complexity such behaviors can also be caused by bugs (HAL errors, hazardous pointers), thus security techniques can be used also for improving reliability and bug detection.

In this paper we don't address communication confidentiality that can be handled with complementary ciphering techniques [11]. We don't either consider physical attacks such as *fault injection* or *side channel attacks*, but our solution makes possible the implementation of counter-measures based on monitored reconfiguration. We focus our study on entity authenticity and data integrity, both attacks can be identified as abnormal communication behaviors. These attacks may be classified in the following three categories :

Hijacking Hijacking is a write access in the se-

Proceedings of the First International Symposium on Networks-on-Chip (NOCS'07)
0-7695-2773-6/07 $20.00 © 2007 IEEE

IEEE COMPUTER SOCIETY

Authorized licensed use limited to: UNIVERSITE DE BRETAGNE SUD. Downloaded on July 28, 2009 at 08:33 from IEEE Xplore. Restrictions apply.

cure area in order to modify the behavior or the configuration of the system, it also includes buffer overflow and internal registers reconfiguration.

Extraction of secret information This second aspect consists in read accesses to data stored in secure targets. The stolen information can be sensitive data (e.g. encryption keys), instructions from critical programs, IP configuration registers and so on.

Denial of service This kind of attacks aims to bring down the system performances. The network over utilization downgrades the operability of the system. We can distinguish four kinds of attack scenarios. The last three ones are specific to NoCs and can occur if NI are programmable or in case of multi-NoC systems with secure and unsecure areas.

Replay: intentional repeated requests imply wastes of bandwidth and cause higher latency transfers in the system resulting in deadline misses for instance. In the context of NoC new kinds of denial of service attacks may occur in order to overcrowd links with useless communications :

Incorrect path: it consists in introducing in the network a packet with erroneous paths with the aim to trap it into a dead end. The body of the trapped packet takes some channels and makes them unavailable for the others valid packets.

Deadlock: it means the use of packets with paths that intentionally disrespect deadlock-free rules of the routing technique with the intent to create deadlocks in the network. This leads to the contention of the channel and consequently of a part or the entire NoC.

Livelock: this is the introduction of a packet that can't reach its target and stay turning infinitely in the network, causing a waste of bandwidth, latency and power.

2.2 Claims for reconfiguration

Security is usually considered as a software issue based on fixed and so identified hardware targets. However we observe that needs for hardware reconfiguration are rising, this evolution is mainly driven by productivity gains. The first point is the hardware and software firmware upgrades on reconfigurable SoC for improving performances and implementing new standards and applications with dedicated hardware IP. The second point is relying on the optimization of design costs for hardware bug fixing as already done for software. Finally, reconfiguration is also a convenient way for chip reuse with a specialization of a single chip for different system integrations. Thus, we have considered a solution that may be implemented on reconfigurable SoC, our experiences are based on Xilinx Virtex2pro series. Both hardware for IP and NoC and Software (SNI) partial reconfigurations can be considered in the context of NoC-based SoC. Our secure reconfiguration protocol depicted in Fig.1 is described in section 3.2.

3 From Bus to enhanced NoC-based solution

3.1 Relative work

Few work exist in the domain of communication-based security for SoC out of specific secure processor cores such as ARM/Trustzone [2] or Stanford/XOM [5]. Design of security-aware communication architectures has been introduced by [9]. Their solution named SECA relies on both a single Security Enforcement Module (SEM) and a Security Enforcement Interface (SEI) for each slave device connected to the bus. Their solution which has been demonstrated through the AMBA bus can be used to implement features such as (*a*) *address-based protection*, (*b*) *data-based protection* and (*c*) *sequence-based protection*. Such a solution is interesting since it can detect a large number of attacks from access control violation up to slave device configuration corruptions. However there are still some limitations that should be considered when dealing with multiprocessor systems where a large number of resources will have to exchange data in parallel. Even if some security features, such as peripheral register write protection, are distributed within the SEI most of the monitoring is centralized and performed by the SEM especially the *address-based protection* which is based on a ternary content addressable memory (TCAM). Such an approach leads to an exponential increase of the cost when adding new slave devices on the bus and strongly limits the scalability. When one device is added to the bus, the whole TCAM needs to be modified. A NoC based solution mitigates this cost since the *address-based protection* can be performed within network interfaces (NI) that are intrinsic to NoC implementations. Thus, *Address-based protection* can be distributed and so performed in parallel like

IEEE COMPUTER SOCIETY

Authorized licensed use limited to: UNIVERSITE DE BRETAGNE SUD. Downloaded on July 28, 2009 at 08:33 from IEEE Xplore. Restrictions apply.

data transfers. In the domain of NoC, ARTERIS [3] proposes the introduction of Firewall relying on fixed communications schemes and software security based on a secure boot configuration. In [12] we present an overview of NoC security issues. Gebotys, in [6], extends classical key exchange protocols for IP identification within NoC.

3.2 NoC-centric proposal

We consider reconfigurable architectures based on 2.2 motivations. Then our objective is not exactly to secure NoC-based communications, but rather to select a NoC in order to implement secure communications based on the range of opportunities it offers. They are mainly: spatial parallelism enabling path-based identification and distributed and dedicated implementation of security functions. Our solution for a secured NoC regarding abnormal traffic detection is based on the following five points:

I. Single, centralized, dedicated and secured master IP. The **Security and Configuration Manager (SCM)** is in charge of safe hardware and communication configurations and counter-attacks monitoring. This is the most sensitive resource, so its configuration must be based on an initial boot and secured network communications for online updates. The SCM can be for instance a usual processor core dedicated to this task.

II. Enhanced Network Interface for access control. **Secure Network Interface (SNI)** handles in a distributed and dedicated way the following attack symptoms: (i) Denial of service, (ii) Unauthorized Read Access (information extraction) and Unauthorized Write Access (Hijacking). The main points are firstly the use of NoC scalability, basically the large number of I/Os that offer NoC enables to personalize and dedicate access control to specific and small memory areas (memory partitions and peripheral configuration registers). Moreover NI proceeding delays can be usefully exploited for implementing security control in parallel with data-flow. The second point is that we don't need costly probes in router [4] since all NoC traffics can be analyzed thought Secured NI. As a consequence to scalability use, the NI cost may be minimized.

III. Two distinct (virtual) networks for a separation between application and security data. Two virtual channels (VC) are implemented, the first one is dedicated to elementary best-effort communications (BE) for inter IP transactions and the second one is a priority best-effort (PBE) channel dedicated to SNI / SCM communications (security monitoring and configuration). For deadlock protection, the transport layer implements connexion-oriented transactions with end to end flow control and the network layer implements source-routing preventing from the introduction of incorrect paths. Note that security messages are very short, typically one flit is enough to transmit a denial of service alert or an unauthorized access attempt. Initially our intention was to use PBE channels for End-To-End flow controls, namely for credits exchanges which correspond to inter-SNI communications but this idea has been left out in order not to create potentially malicious PBE traffic besides SCM communications.

IV. Four steps access control. The verification of access rights and the detection of denial of service can be simplified compared to bus solution thanks to scalability NoC property. Basically a NoC provides a distributed address control that enables to separate IP identification (i.e. path) and local short address checking while bus-based approaches are centralized and lead an exponential cost.

1. Overflow checking. On the emitter SNI side, the first packet contains the message size, namely the number of words to be transmitted within the upcoming transaction. If the bound is reached and the FIFO is not empty then the FIFO is flushed and the transaction ended. Moreover the packet length is limited to the input FIFO size and the network layer control bits of the last flit are automatically set by the NI controller.

2. Path Based Identification. An IP identification can be based on tags inserted in the packet header or payload, this tag can be fixed if an ASIC solution is considered. However intrinsic NOC properties provide a simpler solution, basically paths can be used to identify request and response communications. Our solution is based on this property that enables hardware reconfiguration and does not need extra-information like the ID or the backward path to the remote IP. Moreover we implement a new routing algorithm that simplifies path coding and backward path computation, this point is detailed in section 4.2. Thus the remote IP (request or response) is filtered according the Read, Write rights stored in the acceding SNI and the path instruction extracted from the packet header.

3. Local Address checking. From the payload of the first transaction packet is extracted the local

IEEE COMPUTER SOCIETY

Authorized licensed use limited to: UNIVERSITE DE BRETAGNE SUD. Downloaded on July 28, 2009 at 08:33 from IEEE Xplore. Restrictions apply.

base address and the message size. These values are compared to bounds stored in the SNI. The verification is limited to a reduced number of local address bits.

4. Statistics. Alerts are transmitted to the SCM when traffic bounds are out of predefined normal behavior bounds. Available credits allocations are accumulated and compared to upper and lower bounds over a given observation period.

V. Configuration protocol and SNI configuration. All security policies are ineffective if the system can be configured by the attacker, so the (re)configuration protocol and order is a key issue. In software-oriented security, this point is guaranteed by boot configuration stored in an embedded ciphered memory [7]. In the case of NoC on reconfigurable SoC, we distinguish four phases.

INIT: SOC hardware initial safe configuration. At the boot time, the complete hardware initial configuration is loaded from an external ciphered memory. The reset configuration depends on the reconfigurable architecture. In case of a tile-based architecture built around an hard NoC core only IPs are configured but if we consider a FPGA, then both IP and NoC cores are configured at reset time. In such a case, a dedicated decryption core is available to run this task and a specific strategy can be elaborated to secure this first step as described in [10]. In this paper we consider this stage as already performed.

SNI: initial security configuration. In the very first configuration two kinds of communications are authorized, the first one is a BE read operation from SCM to a ciphered memory containing SNI configurations according to IP access policy and secondly PBE SCM / SNI communications. The available channels are used by SCM to manage a SW configuration of SNI (an example is given on Fig.8). Note that after the SNI step is achieved, a new configuration may be dynamically changed only by the SCM since reconfiguration control resource (e.g. ICAP) is only connected to SCM.

RUN: runtime monitoring based on the current hardware and software configuration. If alerts are detected, different strategies can be implemented and lead to SNI software reconfigurations or to hardware reconfiguration.

DPR: dynamic and partial reconfiguration. Based on monitoring data (adaptation to traffic), Attack detection, or Upgrades requirements the SCM can decide at run-time partial hardware reconfiguration. These reconfigurations are available as bitstreams initially available or downloaded from a secured network connection. It has a consequence on the configuration of the SNI connected to the SCM which is slightly different to IP ones. Actually, it needs only one virtual channel, which can be connected to the BE VCs during reconfiguration in order to access to configuration memories whereas it is only connected to priority VCs at runtime. Thus they are no possible connections with other IPs. This point is described in section 5.1. Online reconfiguration management based on network download a is not a trivial issue from a security perspective, a dedicated protocol still remains to be specified. According to FIPS 140-2 level 3 recommendations, the ultimate solution would consist in a physical separation between secure and unsecure network ports and so to a SCM with dedicated network connection capabilities. Such an implementation would be safe but costly, another one can be based on a pooling procedure that enables the SCM to get ciphered tags authenticating new hardware upgrades from a ciphered memory. This last solution requires four conditions: i) a new BE Read access from SCM to a ciphered memory, ii) a new BE Write access to network processor FIFO to launch secured bitstream downloads, iii) a safe network link (e.g. SSL) and iv) a certification procedure [1] to authenticate upgrade requirements.

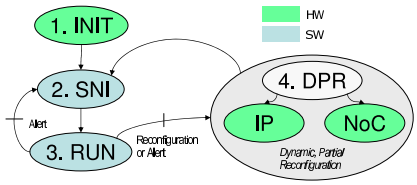

Figure 1: Reconfiguration protocol

4 Architecture models

4.1 SNI

Standard NI Basic design of standard NI is quite simple, since it is based on a single channel with end to end flow controls based on credits exchanges between IPs (see Fig.2-a)). The design is based on a restriction, which is that transactions interleaving is not allowed at emission. Namely the API developed in C for Xilinx microblaze doesn't permit to initiate a transaction before the end

IEEE COMPUTER SOCIETY

Authorized licensed use limited to: UNIVERSITE DE BRETAGNE SUD. Downloaded on July 28, 2009 at 08:33 from IEEE Xplore. Restrictions apply.

packet of the previous transaction is loaded into the FIFO. At reception, packets from different transactions can be interleaved since message sizes are loaded at transaction initialization in SNI registers. These design decisions result in two major changes compared to our previous work where different FIFOs were associated to the different channels. First, there are only one Input FIFO and one Output FIFO, it means that credits and FIFO status are checked during the NI access protocol. Namely, a new transaction is initiated only if the input FIFO is empty and if credits are available. Moreover the read/write address is transformed into the associated path in packet header and local address stored in the first packet. At reception, a simple address generator unit (AGU) produces the local address by summing basis address and offset.

SNI, IP configuration SNI architecture is configured for IP as depicted in Fig.2-b). Compared to standard configuration, we add a second high priority Best Effort channel (PBE) for monitoring data. Security is based on a complete separation of channels, there are no connexion between IP and BPE FIFOs, which are reserved exclusively to SNI communications.

It results firstly in the implementation of a decoder and arbiter for channel selection. Secondly counters are added for statistics updates. Thirdly, a couple of registers for offset bounds checking are implemented for the survey a each inter-IP communication. Finally, the NI controller is enhanced with security controls for offset checking, path identification and alert message generation based on statistics. On the one hand as security control can be performed in parallel with data-flows there is no impact on processing delay, on the other hand the cost increased is mainly dominated by the introduction of a second virtual channel that also strongly impacts the router costs.

SNI, SCM configuration SNI configuration for SCM/NOC interface is specific since the SCM is the only IP allowed to communicate with SNIs as the only trusted IP. These communications can be performed sequentially, so only one couple of input/output FIFO is necessary as described in Fig.2-c). The SCM can access to SNI local registers in order to modify VC ID inserted in packet headers and switch between PBE and BE channels. The SCM uses BE VC to access to configuration memories in order to get SNI configuration for all IPs, the SCM uses the PBE VC to configure IP SNIs. At run time, SCM SNI is configured with PBE VC to prevent any intrusion from remote IP and to transmit security alerts to the SCM. Finally we observe that offset and access configuration registers are useless and can be removed. In section 5.1 we detail access along with the different phases of configuration protocols.

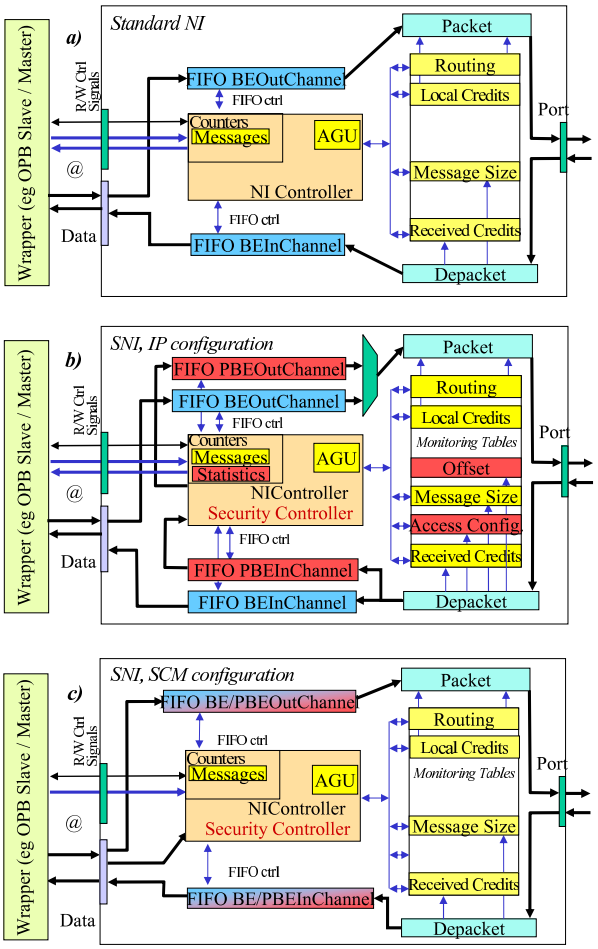

Figure 2: SNI configurations

SCM At reset time, SNIs are configured in such a way that no IPs communication are allowed except for the SCM that can access to its own program and data memories, and to configuration memories. SCM is a processor dedicated to security control, it is in charge of two main tasks. The first one is devoted to SNI configurations, the second

IEEE COMPUTER SOCIETY

Authorized licensed use limited to: UNIVERSITE DE BRETAGNE SUD. Downloaded on July 28, 2009 at 08:33 from IEEE Xplore. Restrictions apply.

one is active at run time and consists in listening alert messages from SNIs, the reaction to attacks is decided by software designers. It can consist in closing affected SNIs and transmit an alert through a secured network communication. Another role of counter-measure management can be assigned to the SCM, against side-channels attacks.

4.2 Optimized path coding for identification and reconfiguration

Concept We propose a simple and low cost solution to allow a receiving SNI to identify the sending SNI of received packet. This one is based on a smart use of routing instructions in packet header. Classical X-Y and street-sign routing coding techniques don't preserve routing information, so we propose a routing technique [8] that we call self-complemented path coding (SCP). The choice of the output port in a router is given by the turn number in counter-clockwise from the considered input port (Fig. 4). This feature induces some key improvements detailed hereafter. The instruction is relative to both input and output ports considered. It allows preserving routing instruction information in packet header to identify the sender from the destination. Moreover this information constitutes not only a single key but also the backward path compliant with reconfigurable architectures where master IP location can be modified. It is a reliable identification key because a sender can't use the routing instructions of another one to usurp its identity because its topological position is different and requires an other path and so other routing instructions.

Architecture Packet header is made of an instruction field constituted of instructions for crossed routers on the path of this packet to travel from the source to the destination. Moreover after proceeding instructions, routers compute backward instructions and shift instructions as indicated on Fig. 5. Note that a small field indicates the number of remaining instructions to proceed. This one is decreased by each crossed router. It allows router to detect a live-lock path (an infinite loop imprisoning packet in the network conducing to a denial of service). A zero count must occur only at destination SNI. Thus, if before router processing it is detected, it is abnormal and conducts the router to delete the entire packet. Destination SNI needs

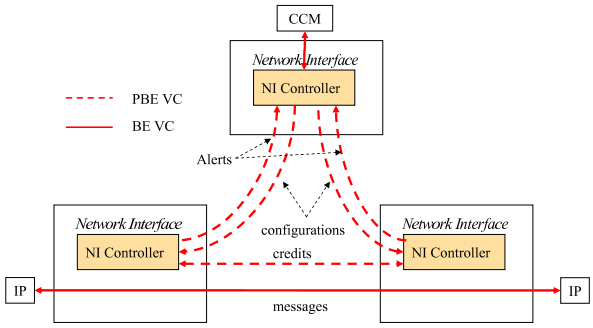

Figure 3: The separate use of virtual channels

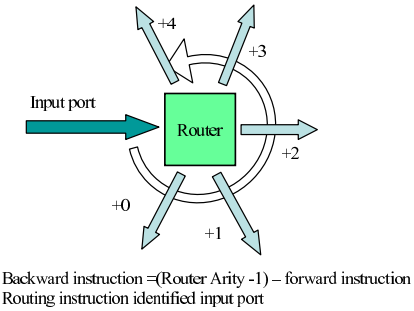

Figure 4: Relative street-sign

only to invert instruction order to determine the backward path. So it can compare it with its own path table, and check corresponding access rights.

Bit level optimization To reduce instruction field width, or to maximize the number of instructions to dispose of longer paths we propose to reduce individual instruction width to the minimum and so to use heterogeneous bit widths for instruction coding. Routers have different arities. Thus, the instruction coding width may be different for each one. However, the problem is the reordering of instructions at destination SNI. The destination doesn't know the width of each of them, so, it can't do the reordering (Fig. 6). We propose a suited

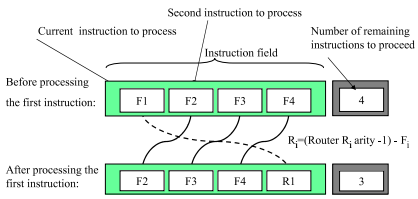

Figure 5: Instruction complement and shifting

IEEE COMPUTER SOCIETY

Authorized licensed use limited to: UNIVERSITE DE BRETAGNE SUD. Downloaded on July 28, 2009 at 08:33 from IEEE Xplore. Restrictions apply.

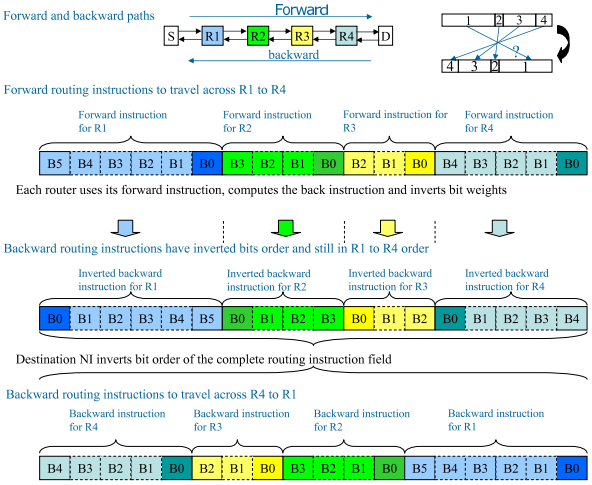

Figure 6: Inversion technique

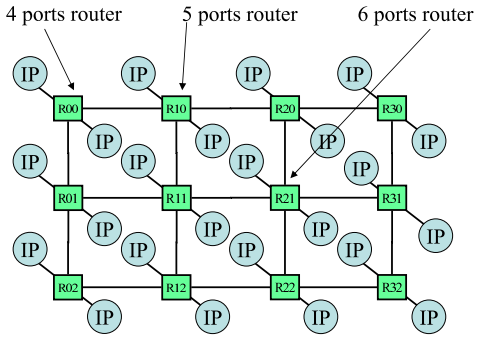

Figure 7: NoC topology

process on instructions to perform this reordering. This process is done at two levels. Back instruction bit order is inverted by router after computing. This is done by each crossed router. Finally, the destination SNI inverts the entire instruction field to obtain the backward instruction field. This process is detailed in Fig. 6

5 Case study

5.1 SetTop Box SoC example

We consider a reconfigurable SOC for SetTop Box applications, described in Fig.8-10, with one SCM, 6 and then 7 master IPs (general purpose or specialized processors including one crypto-processor), and 13 slave memories for data (clear or ciphered), programs and configuration (ciphered security configuration and bitstreams). In this example scalability is completely exploited with 22 SNIs enabling separate access controls. The resulting 2D mesh network is composed of 4x3 routers having various numbers of ports as indicated in Fig. 7 where SNIs are not present for clarity sake. Tab.1 shows how the instruction width may be different for those routers. In such a chip, a strict security policy is necessary to control access to sensitive data such as private keys and crypto-processor programs.

location	#routers	#ports	instrc bitwidth
Corner	4	4	2
Border	6	5	2
Center	2	6	3

Table 1: Instruction bit width for routers with different numbers of ports

5.1.1 Configuration protocol

Step 1 Fig.8 shows the system status after reset. The configuration results from a ciphered bitstream stored in an external memory, the initial bitsream contains IP contexts including NOC IP. The initial configuration only authorizes one BE communication, a read from SCM to ROM memory containing SNI configurations. PBE communications are also instantiated between SCM and SNIs for NoC configuration according to security policy.

Step 2 Fig.8 figures out PBE VC connecting the SCM with SNIs. The SCM uses these channels to configure SNIs, the resulting scheme is given in Fig.9. Plain red arrows indicate BE communications related to sensitive data. In our example the sensitive accesses are related to the crypto processor and to its instruction memory and data memory from which ciphering key can be extracted, to ciphered data memories and their communications with network processor, GPP processor and DMA. Dotted red arrows corresponding to SCM/SNI PBE communications controlled by SCM are not indicated for clarity sake but are identical to those specified in Fig.8. Note that the SCM has only two possible Read BE communications, the first one in Boot ROM occurs at reset time and the second one in ciphered Data Memory 2 is conditioned by a tag authentication as explained in step 4. It means that no read or write operations are possible from any IP to SCM and that no SNI accesses are feasible out of SCM PBE capabilities.

IEEE COMPUTER SOCIETY

Authorized licensed use limited to: UNIVERSITE DE BRETAGNE SUD. Downloaded on July 28, 2009 at 08:33 from IEEE Xplore. Restrictions apply.

Step 3 At run time, alerts are transmitted to the SCM through the PBE VC as indicated in Fig.9. If no abnormal behavior is detected, these channels remain unused or if no explicit read access for monitoring are programed in SCM. It means, that PBE impact on BE traffic is null out of any trouble detection.

Step 4 In case of online reconfigurations for firmware updates or bug fixing, a secure connection can be established by the network processor under the control of the SCM. Different possibilities may be implemented. In this example we base our implementation on the four conditions described in 3.2-V., thus the SCM BE communications are limited to a Read access to a ciphered memory and a write access to the network processor. Thus, based on a pooling, processed with a period that depends on the application characteristics (e.g. every day for a set-top box), the SCM switches to BE traffic mode and reads an upgrade flag from (ciphered) Data Memory 2. After an authentication procedure, if the flag reveals that an upgrade is available then the SCM writes a command to the input buffer of the network processor to launch a safe new configuration download. Otherwise, the SCM returns to the run state. When a new configuration is available (after upgrade, for adaptation or for security reasons), the SCM can load new contexts and partially reconfigure the SOC through the ICAP controller. Fig.10 gives the resulting partial reconfiguration with new DSP and Turbo decoder IPs.

5.1.2 Preliminary results

We have estimated the cost of security while considering a NOC with standard NIs and a second one implementing SNIs. Results have been obtained with our NOC CAD tool (μSpider), the VHDL hierarchy of files is produced according to designer parameters. The selected configuration is the following, Data width is 32 bits, each link between connected routers is composed of two unidirectional opposite links, Buffer size is 8 words in input queue of routers for the message virtual channel, and only 4 words for secure virtual channels. This last one requires minor buffer size because packets are short and contentions between packet on secure VC are short and rare. The security controller used is still elementary, but we have added required resources

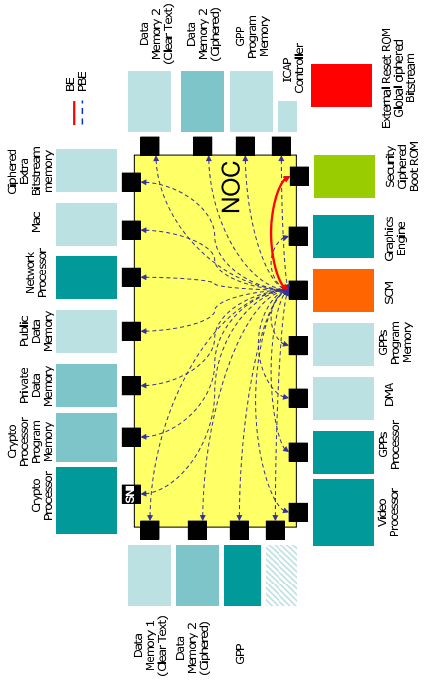

Figure 8: NoC configuration, monitoring links

in order to estimate quite accurately the final cost. Synthesis has been performed with Xilinx/ISE tool. Tab. 2 shows a comparison between a NoC with and without secure features, results include (S)NI, links and routers. As also observed in processor-based work, security has a price, we note that the cost increase is around 45%. This is mainly due to the additional VC for secure transmissions.

NoC	FPGA slices
NI based 4x3 2DMesh	23818
SNI based 4x3 2DMesh	34568

Table 2: Security overhead

This main overhead is due to routers since a router with two VCs is nearly two times the cost of router with a single VC. The cost observed in NI for additional FIFO and registers is not really significant. It means that minimizing routers cost is an important point, that we address with our SCP routing technique. Our method doesn't increase the architecture cost since it is close to the classical street sign and consists only of the replacement of the forward instruction by the reverse instruction and bit inversions by crossing wires. Moreover, it

IEEE COMPUTER SOCIETY

Authorized licensed use limited to: UNIVERSITE DE BRETAGNE SUD. Downloaded on July 28, 2009 at 08:33 from IEEE Xplore. Restrictions apply.

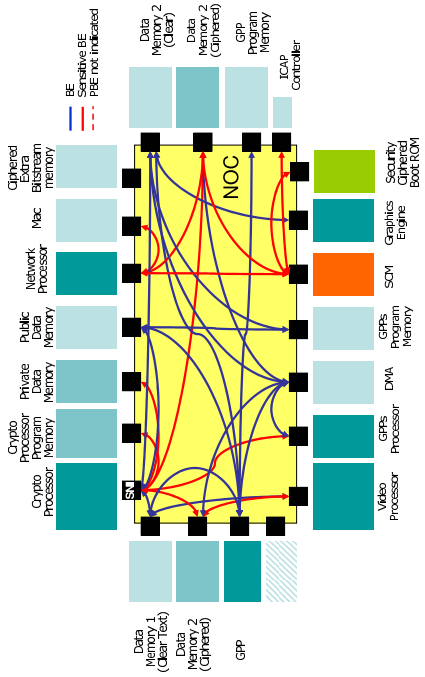

Figure 9: NoC Run-time configuration

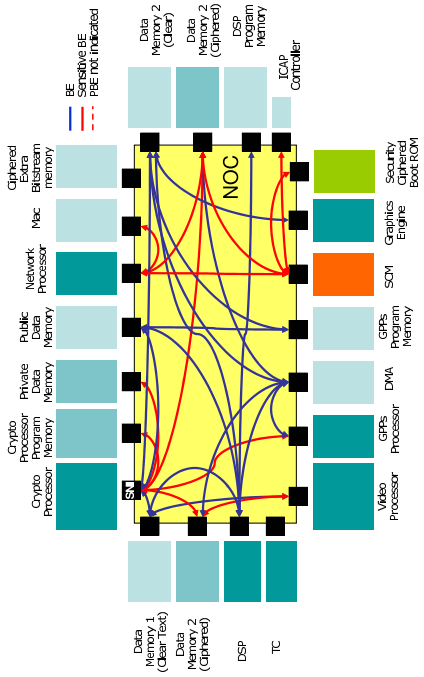

Figure 10: Dynamic IP and NoC reconfiguration

enables to reduce routing instruction field to the minimum. To travel across the network in diagonal with a shortest path, 6 routing instructions are needed. If equal size would be chosen, the width of the instruction field would be 6x3, so 18 bits. With our reduction coding, this width is 6x2, 12 bits, or in worst case 4x2bits + 2x3bits = 14 bits. This coding reduction allows reducing instruction field cost in packet headers, or permits longer paths. Routers with many ports require more bits for instruction coding, and so reduce path length but offer also more output port choices and so opportunities to find a path. Finally note that our VC implementation doesn't introduce any delay overhead since VC (de)coding is performed in parallel with other header bit processing.

5.2 NEC case study

To illustrate our solution we have mapped to a NoC the DRM architecture for portable playback of multimedia content application given in [9] with a bus-based security implementation. In this example two processors CPU A and CPU B (crypto processor) are required. Processors A and B have different rights regarding accesses to the memory mapping. The resulting mapping is described in Fig.11 where N/RW/W/R respectively mean *not accessible, read-write, write only and read only.* We observe that the previous NoC can be reused for this example. SNIs have been connected to Flash, ROM and peripherals, which have homogeneous access rules. This is not the case for the SDRAM which is divided in five areas having distinct access rules. Different solutions may be chosen regarding SDRAM SNIs. Based on our solution, we have used different SNIs for each area which enables a clear security policy for each memory access. An opposite solution would consist in using a single SNI considering that SDRAM accesses can not be performed simultaneously as it is with the initial architecture. The first solution means an increase of NoC cost mainly in terms of routers compared to a solution using a single SNI implementation. However, it also provides a very simple solution if the NoC is already available, in such a case SNI security rules are extremely basic. The availability of small distributed scratch pad memories can also justify this kind of implementation. The second choice fits if a NoC architecture can be decided, however the SNI security must distinguish five areas. Actually the best solution depends on performance constraints

IEEE COMPUTER SOCIETY

Authorized licensed use limited to: UNIVERSITE DE BRETAGNE SUD. Downloaded on July 28, 2009 at 08:33 from IEEE Xplore. Restrictions apply.

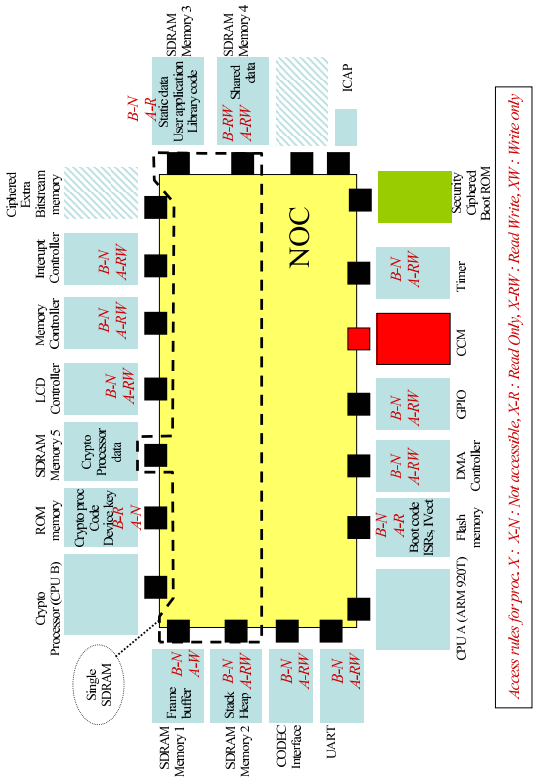

Figure 11: Configuration for NEC DRM [9]

and parallel access opportunities allowed by applications.

6 Conclusion

In this paper we have presented a solution to take benefit of NoC properties for improving security of reconfigurable SOC in a simple and efficient way. Our approach is based on a security manager and secure NIs implementing communications with dedicated virtual channels that separate application and security/monitoring/control data-flows. We have defined a four-step hierarchical access control policy that encompasses attacks corresponding to abnormal communication behaviors. The cost overhead using FPGA slices is 45%, but it must be nuanced since FGPA technology introduces slice waste and doesn't favor required optimizations. Otherwise the overhead it is mainly due to the introduction of virtual channels but some improvements can be performed. A first direction is a selection of a limited number of routers used for PBE traffics, which require low bandwidth compared to applications. Another issue could be the optimization of FIFO length and security message bitwidth. NoC security based on SNI has a double advantage compared to bus-based solution, first security checking are performed in parallel thanks to distributed and local implementations. Secondly, path identification and local offset enable to limit the verification to minimum amount of address bits.

Future work will address monitoring algorithm implemented in the SCM, namely it means a study of different strategies for counter-attacks and counter-measures. Finally, data protection can also be implemented in the SNI especially for peripheral configurations.

References

[1] A.J.Menezes, P. C. Oorschot, and S. A.Vanstone. *Handbook of Applied Cryptography*. CRC Press, 2001.

[2] ARM. Trustzone. www.arm.com.

[3] ARTERIS. A comparison of network-on-chip and buses. whitepaper, Arteris.com, 2006.

[4] C.Ciordas, K.Goossens, A. Radulescu, and T.Basten. Noc monitoring: Impact on the design flow. In *IEEE Int. Symp.on Circuits and Systems (ISCAS)*, 2006.

[5] D.Lie. XOM project. www-vlsi.stanford.edu, 2003.

[6] C. H. Gebotys and R. J. Gebotys. A framework for security on noc technologies. In *ISVLSI '03: Proceedings of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI'03)*, page 113, Washington, DC, USA, 2003. IEEE Computer Society.

[7] G.E.Suh, C. O'Donnell, I.Sachdev, and S.Devadas. Design and implementation of the aegis single-chip secure processor usingphysical random functions. In *32nd Annual Int'l Symposium on Computer Architecture*, 2005.

[8] J-Ph.Diguet and S.Evain. Patent FR 05/53280, oct 2005.

[9] J.Coburn, S.Ravi, A.Raghunathan, and S.Chakradhar. Seca: security-enhanced communication architecture. In *CASES '05: Proceedings of the 2005 Int. Conference on Compilers, architectures and synthesis for embedded systems*, pages 78–89, New York, NY, USA, 2005. ACM Press.

[10] L.Bossuet, G.Gogniat, and W.Burleson. Dynamically configurable security for sram fpga bitstreams. In *18th Int. Parallel and Distributed Processing Symp.(IPDPS)*, 2004.

[11] R.Elbaz, L.Torres, G.Sassatelli, P.Guillemin, M.Bardouillet, and A.Martinez. A parallelized way to provide data encryption and integrity checking on a processor-memory bus. In *DAC'06*, pages 506–509, 2006.

[12] S.Evain and J-Ph.Diguet. From NoC security analysis to design solutions. In Ieee *Work. on Signal Processing Systems (SIPS)*, Athens, Greece, Oct. 2005.

IEEE COMPUTER SOCIETY

Authorized licensed use limited to: UNIVERSITE DE BRETAGNE SUD. Downloaded on July 28, 2009 at 08:33 from IEEE Xplore. Restrictions apply.