nSHIELD-Gateway
A Hybrid FPGA-Microprocessor based Architecture to Foster the Interconnection of Embedded Systems

Marco Aiello, Antonio Bruscino, Antonio Di Marzo and Michele Paragliola
SESM s.c.a.r.l. Via Circumvallazione Esterna, loc. Pontericcio c/o Selex ES, 80014 Giugliano in Campania (NA), Italy

Keywords: Design Methods, FPGA, Security, Reliability, Fault Tolerance, Reconfiguration, Embedded System, Avionics.

Abstract: The complexity of Embedded Systems (ESs) is greatly increased due to their hyper-connectivity. Nowadays resources and data are shared among ESs to create new systems, services and applications. The design of a dynamically composable ES has several pitfalls such as security, safety and privacy that are not easy to handle due to the shortage of tools and standard design methodologies. In such a context an FPGA IP core, based on nSHIELD methodology (EU-FP6-project) and on Xilinx Zynq SoC, has been developed and tested, to mitigate the aforementioned issues and to foster the evolution of the future dynamic composable embedded systems.

1 INTRODUCTION

ESs have been the driving force which has sparked the birth and pervasion of computing platforms such as microprocessors in almost every life aspect (Faggin et al., 1996),(Gercekci and Krueger, 1985). Nowadays industries are leaning toward the adoption of distributed embedded systems (Dabholkar et al., 2012): a class of ESs formed through a composition of predominantly legacy and closed subsystems. The formation of system-of-systems leads to a range of faults that manifest themselves at different granularities for which a statically defined fault tolerance scheme no longer applies at all: new methods comprising dynamic and adaptive mechanisms are needed though. Likewise, the design of complex and heterogeneous systems induces to more articulated security issues that are not easy to treat due to the lack of appropriate design methodologies and tools. So far, the aspects of security and dependability are analyzed during the designing stage and custom solution are carried out; any changes on the system will require a ”system security and dependability assessment” and eventually a system redesign. This approach is time and cost consuming, especially for the life critical systems, where the process of design testing and deployment is highly articulated. Looking at the ESs application domains, every vendor carries on its design with its own ”in house” methodology to design Secure, Safe and Dependable systems. Even though the final result is compliant to safety and security standards required by the application domain, not always it is interoperable with similar products from different vendors. Legacy system and components obsolescence are yet another problem (Hitt and Zwitch, 2002): especially in some specific application domain (such as avionics, railway systems, automotive etc.), aging of the systems has become a problem, mainly because the systems are being kept in service far longer than the original plan. The replacement of a subcomponent poses many challenges on the new substitutes due to the constraint of being fully interoperable in terms of security and dependability with the remaining system parts (Gaska, 2012).

nSHIELD project is co-funded by the ARTEMIS JOINT UNDERTAKING (Sub-programme SP6) (nSHIELD,), and it aims to overcome the aforementioned aspects, providing a complete framework and tools to ease the design of secure and dependable ES; where the security and dependability aspects are integrated into the system design flow. The project leads a pioneer investigation to address Security, Privacy and Dependability (SPD) in the context of ESs as ”built in” rather than as ”add-on” functionalities, proposing with this strategy the first step toward SPD certification program.

The nS-ESD-GW is a key component, it is in charge of binding the generic embedded components to the nSHIELD network. Conceptually this component can be assimilated to a network gateway; although they differ because of the network gateway dispatches messages and generic data, instead the nS-

Aiello M., Bruscino A., Di Marzo A. and Paragliola M..
nSHIELD-Gateway - A Hybrid FPGA-Microprocessor based Architecture to Foster the Interconnection of Embedded Systems.
DOI: 10.5220/0004713201360141
In *Proceedings of the 4th International Conference on Pervasive and Embedded Computing and Communication Systems* (PECCS-2014), pages 136-141
ISBN: 978-989-758-000-0

ESD-GW generates, dispatches and interprets messages belonging to the framework creating an interface among the layers. Several mechanisms and functions have been integrated in this component, such as fault detection algorithms, encryption mechanisms, etc. The ns-ESD-GW has been implemented using the Zynq chip endowed by a standard ARM®dual-core Cortex™-A9 MPCore™processing system with Xilinx 28nm programmable logic; this processor-centric architecture offers the flexibility and scalability of an FPGA with the performance and power consumption of ASIC. Albeit, this framework encompasses aspects of security privacy and dependability, in this paper the major focus has been given to the dependability.

The nS-ESD-GW is subject to a strict company policy, It has been developed, within the context of nshield research project, to support the NSHIELD's avionc demonstrator. Because of this, it is not possible to provide any detailed information about the design, source code or the testing environment.

This paper is structured as follows: in section 2 an introduction to the framework is given; then the section 3 tackles the proposed nS-ESD-GW, as a means to bridge the embedded systems and nSHIELD network components. Furthermore, the section 3 includes a description of the proposed application scenario and of the bench demonstrator. The last section 4, comprises the conclusions.

2 nSHIELD

The main focus of the nSHIELD project is to develop a new methodology to boost and to facilitate the design process for complex embedded systems, with a particular emphasis to Security, Privacy and Dependability aspects (nSHIELD,). In Figure 1, the logical architecture of the framework is depicted. Referring to the framework guideline a system shall be structured in a number of bottom-up logical layers, hereafter described:

Node Layer: they represent the hardware components that constitute the bottom side of this layered architecture. In this layer are placed all the Embedded Systems, which perform a specific function, in the context of the global system. According to the computational capabilities of these devices. The framework defines three types of nodes: nano, micro and power nodes.

Gateway (nS-ESD-GW): makes a bridge to facilitate the exchange of the framework messages. Further details about this type of node will be given in next section.

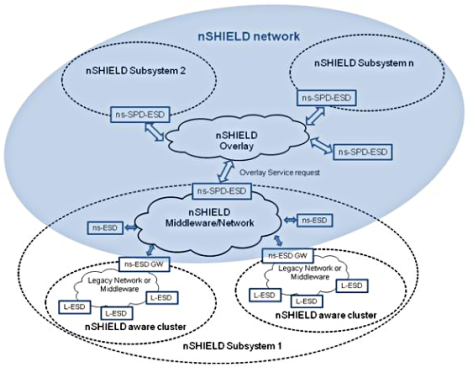

Figure 1: nSHIELD Conceptual Architecture.

Network Layer: includes the communication technologies, procedures, algorithms and protocols that allow the exchange of the data among the components of the framework. This is a heterogeneous layer since it must interconnect different kind of devices.

Middleware Layer: is a software layer that provides functionalities as discovery, composition and execution of the basic services necessary to guarantee SPD required.

Overlay: is a logical vertical layer that collects semantic information coming from the Nodes, Network and Middleware layers and uses them to compute the adequate actions in order to drive the composition of the SHIELD elements and to meet the desired level of SPD. This is a software layer as well.

An ES designed in compliance to the nSHIELD will be natively able to sense and control the SPD attribute. In particular the middleware will execute the three following atomic actions to preserve the SPD level required by the application (Figure 2):

- Measure: nSHIELD/components will measure the current system SPD level by means of nodes, network and gateways;
- Compare: nSHIELD/components will continuously compare the current value of SPD with desired SPD level;
- React: nSHIELD/components will take an appropriate countermeasure to keep an adequate level of SPD. As countereaction SHIELD, eventually, reassembles the whole system by means of discovery, composability and SPD policies.

In order to have a formal and quantitative estimation of SPD level associated to the system several ESs, the framework employs the concept of SPD metrics. The SPD metric is a triple (x,y,z) where each number is a dimensional number within 0 and 100 (higher

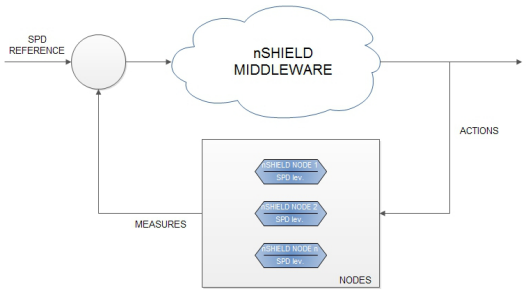

Figure 2: nSHIELD Middleware Functionalities.

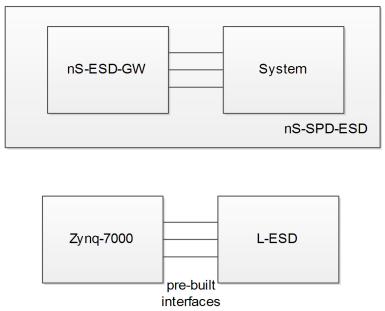

Figure 3: nS-ESD-GW integrations.

number correspond to higher dependability respectively security and privacy).

3 CASE STUDY: DESIGNING THE GATEWAY

To foster the diffusion of this methodology, several framework components, such as the nS-ESD-GW and the middleware, have been deployed as open code risp. open hardware. The nS-ESD-GW is a key component of the nSHIELD methodology. This component is in charge of binding the generic ESs to the nSHIELD network. This component can be used in two different cases (Figure 3):

New systems: it can be integrated during the design process into an hosted embedded system in order to let it become a nSHIELD compliant device;

Legacy systems: it is possible to interface the nS-ESD-GW to a legacy embedded system that is already deployed endowing it the properties required by the framework.

Therefore, considering systems such as avionics, railways, automotive and many others, they encompass several components that, for some reason or other, can't be replaced nor even easily redesigned in order to adhere to the nSHIELD methodology. Upon such considerations the needs of the nS-ESD-GW is based. The nS-ESD-GW finally operates as a smart adapter between legacy components and the nSHIELD network.

3.1 The Gateway

Seeing the flexibility and computational power required by the nS-ESD-GW a suitable solution is based on a SoC FPGA centric architecture. In this way the reliability, security and real-time requirements can be guaranteed exploiting the potentiality and the reconfigurability given by this SoC outlined.

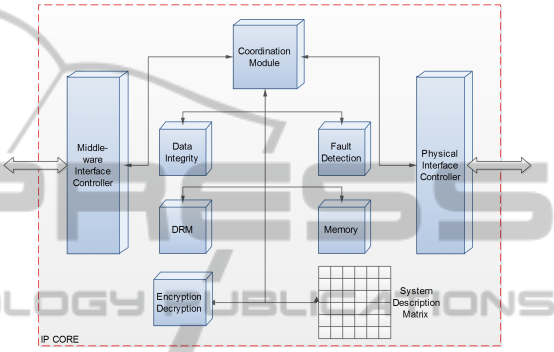

Figure 4: nS-ESD-GW Gateway Logical View.

The integration of the aforementioned nS-ESD-GW into a preexisting system or into a new system implies the development of custom hardware controllers and software drivers, because of the heterogeneous nature of legacy nodes. However, the use of a predefined set of module templates can boost the integration process. As shown in the Figure 4, the gateway architecture includes different modules whose synergy will foster the integration of legacy embedded systems in the SHIELD architecture. This architecture encompasses three main modules:

- Dispatcher modules;
- Coordination module;
- Security module.

3.1.1 Dispatcher Modules

The dispatcher module encompasses two controllers dedicated to the integration and communications of the framework components. In the Figure 4 these controllers are identified as middleware interface controller and physical interface controller. Through the middleware interface controller the nS-ESD-GW converts a service or command request from the component to a logical format that the legacy systems in the lower layer can understand and vice versa. Likewise, the physical interface controller supplies proxy services for nodes with a non-standard physical commu-

nication. According to this, it can be seen as a low level interface to the legacy embedded systems.

The physical interface controller includes a subset of common ready-to-use interfaces as: ETH, SPI, I2C, etc.

3.1.2 Coordination Module

The coordination module is composed by a processor running some balancing algorithms and safety algorithms and by several specific processing blocks, e.g. cryptography. Albeit, it is possible to execute the same process on a softcore processor, the current solution relies on a dual core microprocessor. The processor will be in charge to manage the node resources allocation and to balance services with the aim to ensure the system reliability with a dynamic load distribution.

3.1.3 Security Module

The component identified as DRM (Dynamic Reconfiguration Module) is in charge of managing the reconfiguration of the processing blocks and software modules based on the system context and status. The purpose of this functional block is to increase security and dependability aspects of the node allowing the device to switch between different operational modes. Furthermore this module allows to run multiple applications on the same device enhancing the dependability and redundancy.

nS-ESD-GW architecture includes a Data Integrity controller to process data by implementing cyclic redundancy check. In conjunction with a Encrypt/Decrypt controller it will be possible to ensure a long term storage of sensitive data improving security aspects of the node. Cryptographic algorithms impose tremendous processing power demands that can be a bottleneck in high speed and real time networks. To follow the variety and the rapid changes in algorithms and standards, a cryptographic implementation must also support different algorithms and should be upgradeable in field. For all these reasons, a reconfigurable/upgradable FPGA based HW accelerator will provide software-like flexibility with hardware-like performances. The Fault Detection module will encompass the nSHIELD and application domain specific fault detection algorithms.

3.2 Application Scenario

To verify the validness of the framework and its components, an avionic demonstrator has been employed. Due to framework complexity and to its large amount of features, this demonstrator is not enough to verify all of its features. Therefore the demonstrator is focused on some of them and more in detail on verifying the following concepts:

- Dynamic composability;
- Fault detection and recovery;
- SPD Interoperability and Integration of heterogeneous system.

An avionic system is a critical or even life-critical one, for these reasons keeping it in fully operational status is of vital importance. To mitigate effects, such as loss of services or loss of lives, caused by failure of system components, the system redundancy is commonplace in the avionic field. In this way, in the occurrence of a fault, spare parts can substitute damaged ones. In such a context we employ system supervisor based on nSHIELD methodologies, that is capable to identify and recover a fault by means of composability. At the heart of the chosen avionic application scenario there is the Unmanned Aerial Vehicle (UAV), a system composed by an aircraft controlled by a pilot located on the ground, through a remote control unit. Our target is to keep constant the level of safety of the overall mission through the cooperation of heterogeneous tools and related technologies, in order to protect staff and aircrafts from accidental and/or malicious harm, crime and any other possible threat. In order to achieve our goal, we can identify some critical issues:

- difficulties to coordinate heterogeneous systems to achieve an appropriate level of security;
- difficulties to reassemble the system and to react to a non-predictable and known events;
- difficulties to react promptly to threats based on context changing.

The avionic scenario refers to an UAV flock flight mission. The main actors of the scenario are: two UAVs (scaled down to the Integrated Modular Avionic (IMA) systems), the nS-ESD-GW gateway, a Software Defined Radio (SDR), a GPS sensor board and a Remote Control Unit (RCU). In the course of an UAV flock flight mission, where the adequate level of security is kept due to the coordination between the RCU and the UAV, two events of failure will be tackled: a fault solved at UAV's system level, and a fault settled at system of systems level.

With the aim to demonstrate the fault detection and the system dynamic reconfigurability, an initial configuration of the scenario is made up of a single UAV and its RCU. During a normal operational status, the first fault occurs: suddenly the GPS positioning module experiences a failure. When this event

is identified, it's isolated and recovered by the IMA with the activation of a spare unit. The second fault occurs on the GPS positioning spare module: suddenly it stops working, putting at risk the mission outcome. Due to the lack of another spare unit, such a fault cannot be solved via traditional methods. Using the nSHIELD technology instead, the fault is identified as a reduction of SPD level and is managed via the adoption of the feature composability provided by the nS-ESD-GW. The overlay receives data supplied by the gateway and decides to involve the second UAV to share its measurements (camera, GPS position, radar) and to perform the position estimation task with which the faulty UAV keeps on carrying on its mission. The outcome of this process is a partial restoring of SPD level.

3.3 Bench Demonstrator

In order to put at test our ideas, a bench demonstrator has been implemented. Albeit not being a fully functional system, this choice allows the practical highlighting of key concepts previously introduced: dynamical system reconfigurability and composability, efficient and robust SPD management, legacy systems lifespan extension. The adaptation of a reduced bench demonstrator, rather than a fully functional system, is necessary to conciliate conflicting constraints such as: shortage of development time, budget restrictions, manpower limitations, demonstrator practical value and usefulness, IP cores reusability. The demonstrator main actors are:

- IMAs: they depict the on-board embedded legacy systems responsible of the functioning of the UAVs.
- nSHIELD gateway: being a bridge among the legacy avionic system (already equipped with built-in features of reconfigurability), it allows the extension of dynamic system reconfigurability at the wider system of systems level.
- Network infrastructure.
- nSHIELD Middleware: it orchestrates the overall system by purpose of control the SPDs levels, given the constraints dictated by the available system resources and current threats levels.
- GPS positioning modules: they are used to show the nSHIELD methodology behavior in the management of a fault at the system of systems level.
- SDRs (Software Defined Radios): their purpose is to allow the data exchange among the UAVs in a way that allows to change dynamically the level of SPD on the basis of the decisions taken by the nSHIELD overlay.

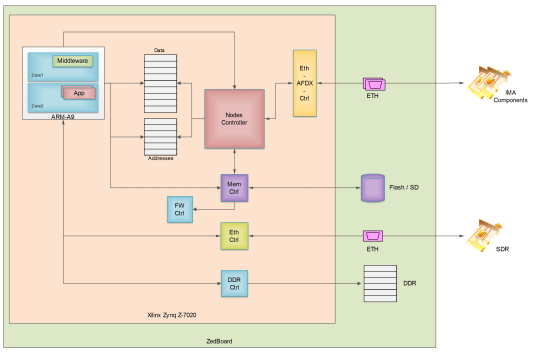

Figure 5: Gateway Architecture.

In Figure 6 a subsection of the whole test scenario is depicted: the IMA itself is made up of an IMA Central Unit plus a RIU board (which is responsible of data acquisition from sensors signals and actuation of commands toward on-board electromechanical systems). Both IMA and RIU are redundant for a matter of dependability as required by the avionic standards. The IMA Central Unit interacts with the RIU via an ETH/AFDX DATA link. The transmission protocol used by the IMA is the RTPS (Real Time Publish Subscribe). This ensemble constitutes the legacy avionic system. The nS-ESD-GW gateway is linked by one side to the legacy embedded system by a standard Ethernet data link and by the other one with the SDR. By the IMA perspective, the gateway acts as an RTPS node, its main purpose is to render the overall system nSHIELD compliant with the minimum impact. All this equipment is replicated, being two UAVs involved. The overlay SW runs on a COTS PC which is equipped with an SDR to perform data exchange with the UAVs. Going further, the gateway has a Zynq-7000 AP SoC at its backbone: it embodies a dual core ARM CPU plus the relative cache memories, an FPGA fabric, DSP data paths, I/O functions. On the first ARM core runs the middleware , on the other CPU there are bare metal applications which allows the interfacing with the IP cores implemented on FPGA fabric.

Now follows a detailed description of the aforementioned faults. Starting with the single system level fault, the GPS positioning module malfunctioning is excited merely unplugging the power to the module. The GPS module is directly plugged to the RIU and the IMA can perform settings and readings using the IMA itself. The IMA Central Unit periodically acquires information needed to apply actions defined by the avionic standards. When the fault is detected, being available spare resource, the IMA substitutes the faulty part with a replacement. In this context the nS-ESD-GW continuously gathers these information and computes, through the middleware, the

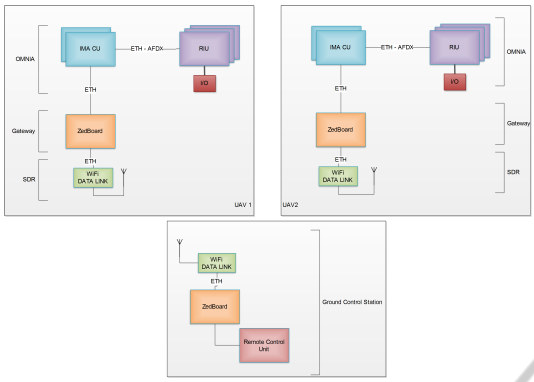

Figure 6: Overall Architecture.

SPD values. Looking at the second malfunctioning, due to the fault of the spare GPS positioning module, the IMA cannot overcome the replacement without the nSHIELD methodology intervention. Looking at Figure 5, the nodes controller writes information acquired by the IMA about the system status on the data tables hosted in the Gateway. The middleware running on the ARM CPU reads these memories and uses their content in order to guide its decisions. The SDR will distribute to the Remote Control the information about the occurred fault Unit, consequently the Overlay computes the action to be performed and orders the involvement of another UAV. The middleware running on this second UAV reads its data tables in order to query for the presence of the requested resources. Now the middleware involves the nodes controller to gather the requested GPS positioning data. Once collected these information are written in the data table. At this stage the middleware of the second UAV involves the SDR resource in order to start a data reply with the estimated GPS data positioning information of the first UAV. In this way the middleware on the first UAV detects a pending transmission on its SDR, recognizes the requested GPS data, writes it on its data table. Now the nodes controller is used to start a publish transaction via the RTPS protocol. Hereon the IMA detects the message and can sustain the lack of the faulty GPS module.

4 CONCLUSIONS AND FUTURE WORK

In this paper we have shown as the nSHIELD methodology is a viable solution to tackle the management the dependability requirements in heterogeneous and distributed ES. Albeit, the silicon market is overcrowded by a multitude of chips adequate to the purpose, the Zynq SoC has represented a perfect match with our needs thanks to its tight integration of typical embedded systems components such as multicore ARM CPUs, a powerful FPGA fabric, a wealth of IP peripherals and a tight integration among its subcomponents, accompanied by a huge capacity data channel. Its huge versatility has greatly eased the practical application of the nSHIELD methodology in the context of the gateway implementation. Albeit, this methodology offers a first step toward the implementation of a design standard in the ES landscape, it has yet to experience a wide diffusion. In the years to come such a void could be filled via the adoption of this methodology and its evolutions as a standard in the design stage of heterogeneous and distributed embedded systems.

REFERENCES

Dabholkar, A., Dubey, A., Gokhale, A., Karsai, G., and Mahadevan, N. (2012). Reliable distributed real-time and embedded systems through safe middleware adaptation. In *Reliable Distributed Systems (SRDS), 2012 IEEE 31st Symposium on*, pages 362–371.

Faggin, F., Hoff, M., Mazor, S., and Shima, M. (1996). The history of the 4004. *Micro, IEEE*, 16(6):10–20.

Gaska, T. (2012). Optimizing an incremental modular open system approach (mosa) in avionics systems for balanced architecture decisions. In *Digital Avionics Systems Conference (DASC), 2012 IEEE/AIAA 31st*, pages 7D1–1–7D1–19.

Gercekci, A. and Krueger, A. (1985). Trends in microprocessors. In *Solid-State Circuits Conference, 1985. ESSCIRC '85. 11th European*, pages 233–233i.

Hitt, E. and Zwitch, B. (2002). Aging avionics: the problems and the challenges. *Aerospace and Electronic Systems Magazine, IEEE*, 17(9):16–21.

nSHIELD. http://www.newshield.eu/.