Computers and Electrical Engineering 61 (2017) 48–66

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

High-level modeling and synthesis of smart sensor networks for Industrial Internet of Things☆

Ching-Han Chen, Ming-Yi Lin*, Xing-Chen Guo

Department of Computer Science and Information Engineering, Machine Intelligence and Automation Technology (MIAT) Laboratory, National Central University, Taoyuan City 32001, Taiwan, ROC

ARTICLE INFO

Article history:
Received 24 September 2016
Revised 23 May 2017
Accepted 4 June 2017

Keywords:
Discrete-event modeling
High-level synthesis
Industrial Internet of Things (IIoT)
Smart sensor network

ABSTRACT

In this work, we use a high-level design methodology for the rapid hardware synthesis of a complex smart sensor network (SSN) system. The GRAFCET is then used to model the individual functional modules and the hierarchical behavior of the system. The behavior of each module is represented as a sequential–concurrent hybrid discrete event system. We apply high-level synthesis rules to generate a VHSIC hardware description language (VHDL)-target efficient hardware for a smart sensor controller and smart gateway controller. Finally, these embedded hardware controllers are generated automatically to integrate all intelligent functional modules into a complex embedded system, and a hardware circuit is then synthesized. The experimental results show that the hardware circuit can meet the definition of an SSN system for Industrial Internet of Things applications. Moreover, this methodology enables a coherent design quality, short design period, low development cost, and short time-to-market for complex industrial applications.

1. Introduction

Smart sensor network (SSN) systems show promise for building powerful Industrial Internet of Things (IIoT) applications. IIoT refers to the close integration of computation, networking, communication, and devices, and it is characterized by sensing, inferring, actuating, information exchange, data storage, and data processing capabilities.

A wide range of SSN applications have been developed in recent years, in which smart sensor devices are embedded in interconnected devices to sense, monitor, measure, communicate, and exchange information. This enables the collection, processing, analysis, and dissemination of valuable information gathered in various industrial environments.

SSN systems offer the ability to perform computations, make intelligent decisions and control industrial equipment to promote the progress of the enterprise or manufacturing unit [1–3]. Fig. 1 shows a generic model of an SSN system for industrial applications.

Most industrial SSN systems include a large number of smart sensors, actuators, gateways, electronic devices, and industrial equipment connected to the Internet or a cloud server for industrial environments. An industrial SSN system can be considered as a physically interconnected network because an increasing number of devices are equipped with smart sensors, electronic devices, and industrial equipment [5], in which things can be connected and controlled remotely.

☆ Reviews processed and recommended for publication to the Editor-in-Chief by Associate Editor Dr. Z. Jin.

* Corresponding author.
E-mail address: pk4881@gmail.com (M.-Y. Lin).

http://dx.doi.org/10.1016/j.compeleceng.2017.06.005
0045-7906/

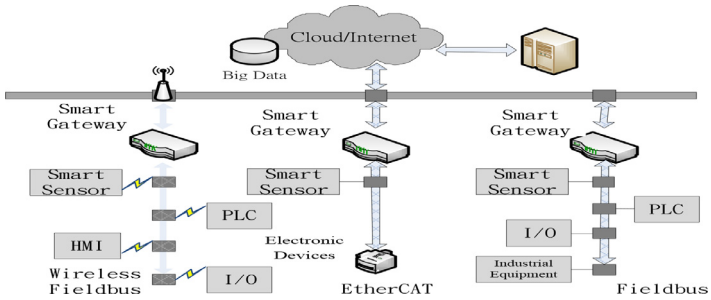

Fig. 1. SSN system for industrial applications.

Smart sensors must sense, exchange information, transmit useful collected information, and automatically assign roles to manage, deploy, and schedule the behaviors of industrial devices over a network. In an industrial SSN system, different things have different communication functionalities. A gateway must be able to facilitate the communication or interaction of various devices; furthermore, it must connect up to the cloud and down to smart sensors and existing controllers embedded in the network system. Consequently, smart sensors and gateways [6,7] play a crucial role in IIoT applications.

The authors of [2–4] have proposed that future industrial SSN systems should have characteristics such as self-configuration, self-optimization, self-protection, and self-healing because smart sensors will become more intelligent. In this paper, our definition of an SSN system for IIoT applications involves the following functionalities: nonlinear calibration, self-compensation, self-inspection, self-validation, and self-diagnosis.

Several field-programmable gate array (FPGA)-based implementations of SSN systems have been proposed for when high performance is required in the targeted IoT applications [11–18,23,24]. The potential benefits of FPGAs are mainly due to the high speed, super parallelism, and flexible configuration that can be achieved for high-performance operation. Hence, these hardware solutions generally overcome the low effectiveness, low efficiency, high power consumption, and slow response time of microprocessor-based software solutions [24].

To understand the development of SSN systems in industry, this work summarizes the foundational technologies of SSN systems and their key challenges, and identifies research constraints and system requirements. The significance and contribution of this manuscript are summarized as follows:

Generic hardware synthesis with embedded intelligence in things is difficult to achieve because most methods are applicable to hardware synthesis with a single function and sequential state transition, and they lack multiple functions, concurrent state transitions, and branching state transitions. Therefore, the main contribution of this manuscript is the focus on hierarchical modeling and designing. Embedded smart multiple functions satisfy SSN system requirements and are suitable for the behaviors of industrial SSN systems. We provide an effective and efficient hierarchical discrete-event modeling methodology by using hierarchical design, behavioral modeling, and automatic hardware synthesis to realize the interconnection, intercommunication, interaction, and interoperation requirements of a complex SSN system.

Therefore, a complex SSN system is decomposed into coarse-to-fine submodules, and each independent submodule is considered as a discrete-event system (DES). For each independent DES, we present a complete VHSIC hardware description language (VHDL) synthesized code to translate into a VLSI hardware architecture circuit for a smart sensor and gateway controller.

On the basis of research and development, this work focuses on a consistent and systematic design methodology using the GRAFCET language. Complex embedded system design and hardware synthesis tasks are straightforward and effective. Because of the hierarchical design, behavioral modeling, and automatic hardware synthesis paradigm, the SSN hardware controller is endowed with composability and software reusability instead of complexity. This reduces the development time and cost as well as the procedure of traditional trial-and-error approaches.

The rest of the paper is organized as follows: In Section 2, we explain our proposal and discuss related studies. In Section 3, the foundational technologies of SSN systems and their key challenges are summarized, and the research constraints and system requirements are identified. In Section 4, the effective and efficient hierarchical discrete-event modeling methodology is introduced in detail. Moreover, the benefits of using a consistent and systematic design methodology based on the GRAFCET language are discussed. Section 5 demonstrates the SSN design details, namely function and behavior definition, hierarchical modular design, discrete-event modeling, and high-level synthesis. Experimental results are presented in Section 6. Finally, in Section 7, the main conclusions of the work are explained and the limitations of the research are discussed.

2. Related work

In this section, we initially present an overview of SSN based on FPGA technology, followed by the state of the art SSN systems of several study cases and their application environments. Numerous recent studies have used FPGA in the design of SSN systems to improve the processing system performance [3,11–18]. Those researchers have proposed developing

reconfigurable sensor nodes for SSNs in IIoT environments, in which FPGA is adopted as the core controller. FPGA offers potential super-parallel processing capacity, flexible configuration benefits and it can read data in real time at high speed from multiple sensors, enabling smart sensor nodes with powerful embedded processors.

The project has proposed an FPGA-based SSN system that periodically measures the environmental temperature, relative humidity, solar radiation, CO_2, air pressure, and air flow [11]. This enables real-time signal processing and photosynthesis calculation through an FPGA-based photosynthesis smart sensor. In [12,13], Chi et al. have presented a reconfigurable intelligent sensor interface for IIoT applications, in which programmable hardware technology is adopted as the core controller. The sensor interface can process data in parallel, in real time, and at high speed. The proposed smart sensor device is based on the IEEE1451.2 intelligent sensor specification standard. Water environment monitoring revealed that the experimental results meet IoT application requirements. Perera et al. [14] also presented the design of a generalized, low-cost, reconfigurable, and reprogrammable SSN and used a ZigBee with FPGA to develop smart sensor node based on the IEEE1451 family of standards. In [15], Francisco et al. developed an alternative to the traditional reprogramming approach based on an on-chip learning scheme to implement the C-Mantec CoANN algorithm. They presented three case studies, namely a fire alarm, weather prediction, and fall detection. The solution overcomes hardware resource limitations to adapt the sensor node behavior to the environment conditions. Guesmi et al. [16] proposed an SSN system which combined with motor current signature analysis using fast Fourier transforms for the online faults diagnosis of induction motors. In [17], Luis proposed an SSN system to monitor an electrical installation. The system was able to detect power quality events and correlate them to calculate real-time indexes and big-data logging. These proposed SSN systems based on FPGA are good effects are achieved in practical application of IoT.

Next, we review studies on intrusion detection and prevention systems [19–22]; issues related to the security of data sensing and communications for SSN system in IoT are also discussed. The cloud-of-things is the latest IoT evolution; it incorporates a diverse range of things, in which cloud computing can be generally regarded as the technological enabler for IoT in cloud-of-things. An integrated IoT and cloud computing applications enable the creation of smart environments. Big data is increasingly accessible and stored on a range of cloud-of-things infrastructure using cloud technologies without security consideration, such as SSN systems. In fact, cloud-of-things, mobile OS devices, and IoT services are vulnerable to malicious cyber threats as they cannot be given the intrusion protection. Researchers have posited that the cloud technologies in IoT (i.e., cloud-of-things) will cause security incidents and privacy risks [19–22]. Therefore, any vulnerability can be exploited by cyber-criminals seeking to exfiltrate information and attacking.

Bertino et al. [20] proposed that IoT services will evolve along with the emergence of new vulnerabilities and threats. IoT systems introduced a large number of vulnerabilities, because each device represents a potential risk and attack threat as a backdoor for unauthorized access. DOrazio et al. [21] demonstrated that an attacker could exfiltrate data from an operating system of a mobile device by abusing a library and a command line tool distributed with the application program. To avoid similar attacks in the future, the authors presented recommendations for mobile OS devices. Teing et al. [22] attempted to determine the data remnants from the use of newer BitTorrent Sync applications. Findings from this research using mobile and computer devices running a variety of operating systems suggested that artefacts relating to the installation, uninstallation, log-in, log-off, and file synchronization could be recovered, which are potential sources of IoT forensics. Cahyani et al. [19] demonstrated the physical acquisition of a Windows phone and revealed the limitations of the current mobile forensic tool support for Windows Phone 8. Furthermore, the results provided an overview of the current ability of mobile forensic tools and the challenges for successfully extracting evidence from the Windows phone platform for cloud-of-things.

3. Design of complex smart sensor network system

An SSN system can be considered as a physically interconnected network infrastructure comprising numerous connected smart sensors that employ networking, communication, information exchange, data storage, and data processing technologies.

A fundamental technology for an SSN system is a smart sensor and a smart sensor gateway. A smart sensor enables microcontrollers to transmit identification information to a user through communication, and a smart sensor gateway uses interconnected sensor devices for sensing and monitoring.

These key technologies significantly improve the ability of an SSN system to sense and identify industrial things or the environment. Because of these conditions, SSN systems have many interconnection, intercommunication, interaction, and interoperation problems. This section focuses on identifying the design processing, implementation processing, and business processing of existing SSN system studies in industrial areas and highlights the challenges, constraints, demands, and process requirements.

3.1. Design processing

In [4], the authors categorized system-level design into three domains: upstream system specification, midstream algorithm elaboration, and downstream electronic device design.

The design of a complex smart sensor controller is challenging because it must satisfy the following conditions:

(a) Intercommunication between midstream and downstream: Coordination between different standards and the supervision of various relationships are necessary.

(b) The interconnection of complex computing systems makes decomposition difficult because of the tight coupling between subsystems and submodules, and achieving collaborative teamwork is difficult.
(c) If a problem is identified in the design flow, the system development cycle will be prolonged and the cost will increase.

3.2. Implementation processing

Recent technological advancements in smart sensors have increased the viability of using IIoT systems consisting of numerous intelligent sensors to sense, monitor, measure, communicate, and exchange information. Sensor data are shared among sensor nodes and gathered in a distributed or centralized system for analytics in various enterprise applications and industrial environments. A sensor device is required for extending the lifetime of the network and for ensuring reliable data collection from the sensor.

(a) Sensor node hardware: A sensor node has limited processing capability. Typically, a sensor node contains sensor interfaces, processing units, transceiver units, a power supply, and A–D converters. Node failure and energy are common characteristics of a sensor network, and the network topology should be able to heal itself. Saving energy is a critical design goal for IIoT devices such as wireless sensors [9].
(b) Network communication stack [8]: Sensor nodes are deployed in an ad hoc manner for most applications. Designing an appropriate topology, routing, and media access control (MAC) layer is critical for the scalability and longevity of the deployed network. Sensor nodes in SSN systems must communicate among themselves to transmit data in single or multiple hops to a base station. Node drop outs and consequent degraded network lifetimes are frequent.
(c) A smart gateway should be able to interact through the IIoT network, enabling easy connectivity to electronic devices and industrial equipment for automated production and control systems. Connectivity is imperative for realizing the power of an IIoT network, which can enable insights to be gained from the data provided by the connected things.

The smart gateway for an IIoT network should offer proven solutions for the following problems:

(1) Connectivity up to the cloud and enterprises and down to smart sensors and existing controllers embedded in the system.
(2) Local decision making and computing for in-device analytics.

Furthermore, a smart gateway for IIoT networks should meet the definition of a smart gateway with the following intelligent functionalities: nonlinear calibration, self-compensation, self-inspection, self-validation, and self-diagnosis.

3.3. Business processing

Industrial SSN systems are becoming increasingly popular in industry and organizations because they have the potential to engender significant personal, professional, and economic benefits.

However, SSN systems may include numerous smart sensors, electronic devices, and industrial equipment that are developed by different manufacturers, and may not always follow the same standards.

As more connected smart things become available, more standards and solutions will emerge.

To provide high-quality services to users, SSN technical standards must be designed to define the specifications for communications, processing, and information exchange between things. However, a sensor network evolves into connecting existing objects and embedding intelligence into the industrial environment. In [4], the authors presented standard requirements for SSN systems:

(1) Interoperation: Software architectures and communication networks should ensure interoperability between things and the system, enable easy service deployment and application development, and process and convey contextual information.
(2) Interaction: In an SSN system that aims for autonomous and smart behavior, a shared understanding of the situation of users and their appliances should be realized.

With these two fundamental requirements, smart connectivity and context-aware computation can be accomplished. Furthermore, SSN systems are considered the building blocks of IIoT, and modular architectures must consider scalability, extensibility, and interoperability among heterogeneous devices.

4. High-level modeling and synthesis methodology

This section emphasizes the importance of embedded hardware design and automatic synthesis in a complex computing system design and provides an effective and efficient design methodology for complex embedded system design and hardware synthesis. In this methodology, which is based on the GRAFCET formalism for hierarchical design, behavioral modeling and the automatic hardware synthesis of complex SSN systems that satisfy the definition of an intelligent sensor with multiple functional modules are performed.

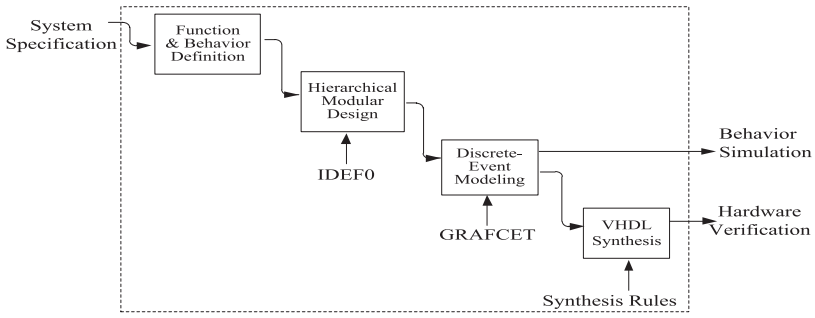

Fig. 2. High-level modeling and synthesis methodology.

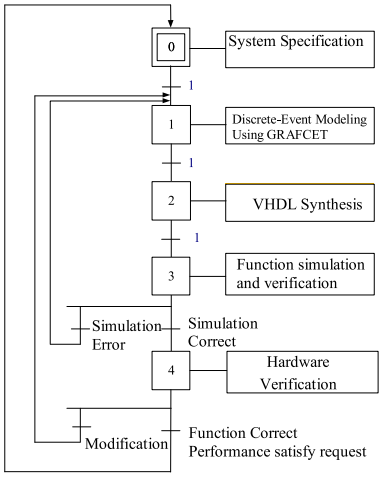

Fig. 3. Workflow of high-level modeling and synthesis methodology.

Fig. 2 shows the top-level schema of the high-level modeling and synthesis methodology. Furthermore, Fig. 3 shows the GRAFCET workflow model of a methodology that contains five macro-steps, namely system specification, discrete-event modeling, VHDL synthesis, function simulation and verification, and hardware verification.

First, we define system specifications that must be satisfied for the functions and behaviors of an SSN system and hierarchical and modular design. We then use the function modeling tool IDEF0 and perform the hierarchical and modular design of the functional architecture for the SSN system. The behavior of each independent functional module is represented as a sequential–concurrent hybrid DES by using the GRAFCET discrete-event modeling tool and is then synthesized as a VLSI hardwired circuit. The graphical GRAFCET language was adopted to model the individual DES module and the hierarchical, pipelined behaviors of the system. Each GRAFCET structure can be synthesized in a segment of the VHDL code block. We propose a systematic design methodology based on the GRAFCET formalism for the hierarchical design, behavioral modeling, and automatic hardware synthesis of SSN systems that generally have multiple functional modules.

4.1. Hierarchical design of complex system: IDEF0

As shown in Fig. 4, IDEF0 is an effective graphical analysis and functional modeling tool. It can help to analyze the structure of a complex system and facilitate communication between the developer and the designer from a functional perspective. An IDEF0 model consists of functional blocks and arrows. Each functional block represents an (single) activity, process, or operation in a system. Arrows represent relationships between functional blocks and describe (the data) required for executing the system. There are four arrow types: input, output, control, and mechanism. These are collectively called ICOM; IDEF represents ICOM DEFinition.

The different arrows are identified by the side of the activity block. Inputs are on the left, and they describe the necessary data for a function to be performed and transformed into outputs. Controls are at the top, and they specify directions, conditions, controls, constraints, and forms of input. Mechanisms are at the bottom, and they specify the resources, methods, and tools required to complete the process. Outputs are on the right, and they describe the data produced. The IDEF0 graphical functional decomposition tool can be used to describe the system architecture more completely and in a detailed manner.

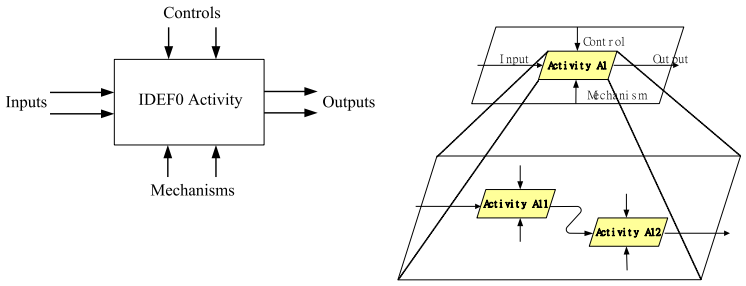

Fig. 4. IDEF0 (ICOM DEFinition 0).

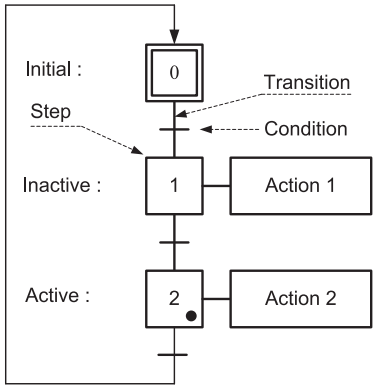

Fig. 5. Elements of GRAFCET graphical language.

A complex SSN system can be divided into independent submodules; therefore, the task of collaborative design of a complex SSN system can be accomplished. The behavior of each IDEF0 module is considered as an independent DES. The behavior of each IDEF0 module is represented as a sequential/concurrent hybrid DES. GRAFCET is used to model the evolution of discrete events.

4.2. Discrete-event modeling using GRAFCET

A DES is a discrete-state and event-driven system that includes the space of discrete states and the events of changing the states. Our methodology uses GRAFCET because it can describe the evolution order of a large number of states in the DES. In addition, GRAFCET provides a simple and intuitive formalism for a DES. GRAFCET contains a sequence of steps with a transition to connect two consecutive steps and a directed connection.

As shown in Fig. 5, the initial step of a DES is represented by a double-lined square associated with a unique number; a single-lined square represents a normal step. A normal step is either active or inactive and is usually connected to an action. The triggered action is executed only when the corresponding step is activated, and the square is marked by a black dot indicating the acquisition of a token. If the step is not activated, the token will be transferred to the next step and the action will not be allowed. A vertical line is used for connecting two consecutive steps, and a transition represents a possibility of change in the behavior of the system. A condition is represented by a horizontal line provided in between. Each directed connection links a step to a transition or a transition to a step; a step and a transition always appear alternately.

When the above step is active and the transition condition is fulfilled, the step becomes inactive and the step below simultaneously becomes active. A condition can be associated with these actions, and the action is executed only if the step is active and the condition is satisfied.

4.3. Hierarchical GRAFCET modeling

A hierarchical GRAFCET model can be represented by a specific hierarchical structure that is similar to that of the main program and its subfunction in the software architecture. Sub-GRAFCETs are similar to subfunctions that represent independent DESs interacting with GRAFCET, which are similar to the main program. In Fig. 4, a sub-GRAFCET accepts the transition signal for activating the subsystem and transmits the fulfillment signal to GRAFCET while all functional steps are performed.

In Fig. 6, G1 is a GRAFCET module that includes G11 and G12 sub-GRAFCET modules; X0, X10, and X20 are the initial steps of G1, G10, and G20, respectively. If step X1 is activated, X0 is switched to inactive. Simultaneously, the active state in sub-GRAFCET G11 is also transferred from X10 to X11, and the G11 module initiates autonomous evolution.

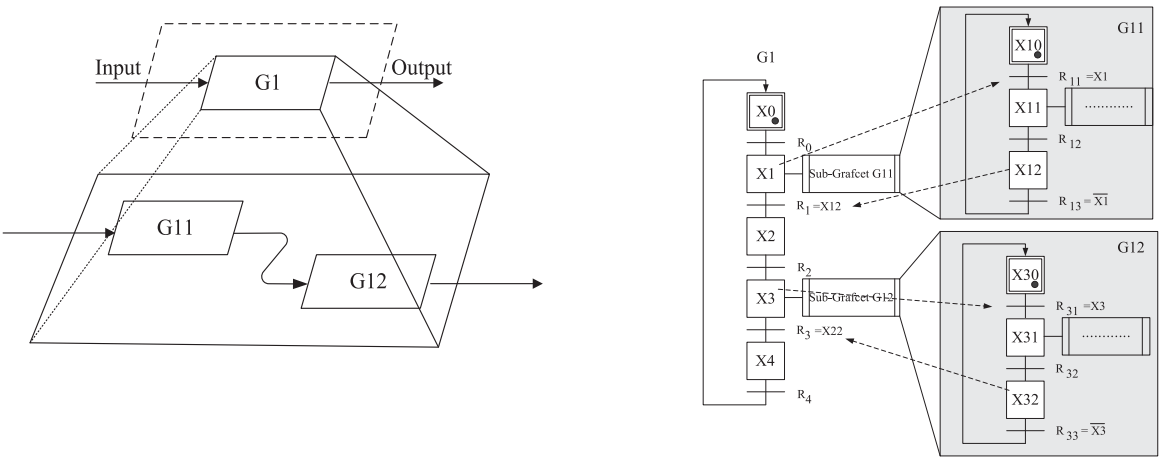

Fig. 6. Hierarchical discrete-event modeling by GRAFCET.

The hierarchical structure in Fig. 6 can be extended consecutively to the lower layer following the aforementioned protocol. Thus, a system is decomposed into coarse-to-fine submodules, and each submodule is independently designed. Because of the simple master–slave communication protocol, the design of a complex computing system can be divided into a set of decoupled tasks whose results will ultimately be integrated.

4.4. Hardware synthesis of GRAFCET model and behavior simulation

Because GRAFCET is a rigorous language structure with states and transitions, formulating a set of synthesis rules for mapping each GRAFCET block into the hardware architecture is not difficult. Fig. 7 shows a set of VHDL synthesis rules corresponding to the GRAFCET blocks. Each GRAFCET structure can be synthesized in a segment of VHDL code block. Fig. 7(a) and (c) show VHDL code blocks corresponding to the concurrent structures, and Fig. 7(e) shows the behavior-simulated timing diagram of concurrent state transitions. Fig. 7(b) and (d) show VHDL code blocks corresponding to the branching structures, and Fig. 7(f) presents the behavior-simulated timing diagram of branching state transitions. Because of these generic synthesis rules, the transformation of the GRAFCET model to VHDL code is fast and straightforward.

DEM01 is the identifier for the GRAFCET module. In the entity unit, we declare the action and transition signals for the DES model as well as the essential clocking (CLK) and RESET signals. In the design unit of ARCHITECTURE, we declare state signals X1, X2, and X3, which indicate whether the internal state is active or inactive. The evolutionary behavior of the GRAFCET model is described in the process triggered by CLK'EVENT and the CLK signal. The state signals are assigned to action outputs: A1 <= X1; A2 <= X2; and A3 <= X3. Based on these synthesis rules, the VHDL synthesis template and behavior-simulated timing diagram are shown in Fig. 8.

5. System design of the smart sensor network

FPGA-based [10] hardware realizations of a smart sensor device enable very high-speed inference but they are generally characterized by high cost and time expenditure, lack of a multiple-function design, and fail to meet SSN system requirements. In this section, we demonstrate the FPGA hardware implementation of a smart sensor controller and smart gateway controller by applying the high-level design methodology. The synthesized hardware controllers can be obtained rapidly by using the synthesis rules provided in Section 4.4. This reduces the development time and cost as well as the procedure of traditional trial-and-error approaches.

5.1. Hierarchical modeling using IDEF0

This section presents the hierarchical modeling of a smart sensor controller and smart sensor gateway using IDEF0.

As shown in Fig. 9, the hierarchical modeling of the smart sensor node is integrated with a **fieldbus communication** module (**RTUReceive** and **RTUSend** submodules**), sensor controller module (sensor interface), function assistance** module (**PortEvent, Baudrate**, and **Timer** submodules), and **smart sensor controller** module.

The **smart sensor controller** module is the core of the smart sensor device and is responsible for packaging processing tasks and controlling communication with other modules, including the **function assistance** module that assists the master control module in completing specific tasks.

The **Timer** submodule enables the communication modules to judge the ending signals of the fieldbus frame and activate the master control module for error processing. The **PortEvent** submodule reports the control state signals of the master

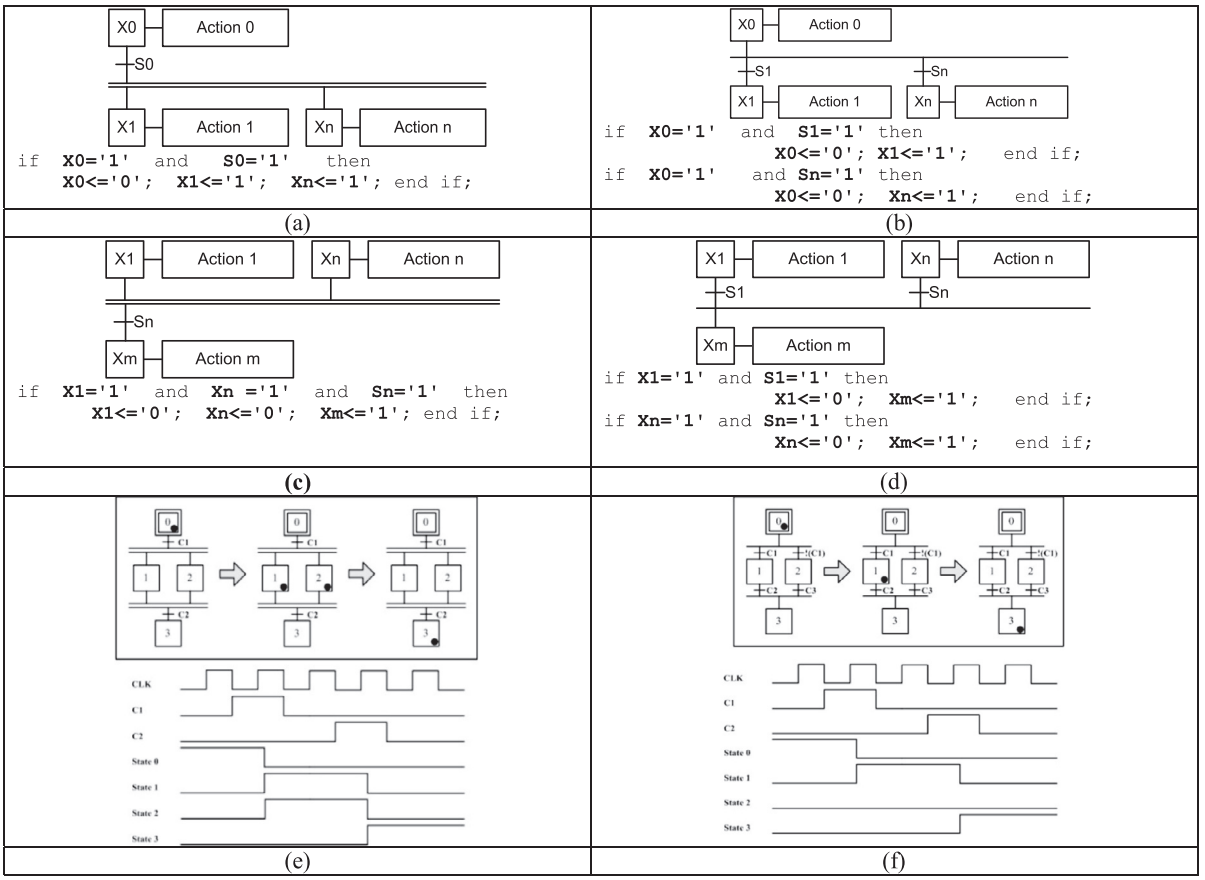

Fig. 7. Synthesis of the VHDL code segment from GRAFCET model.

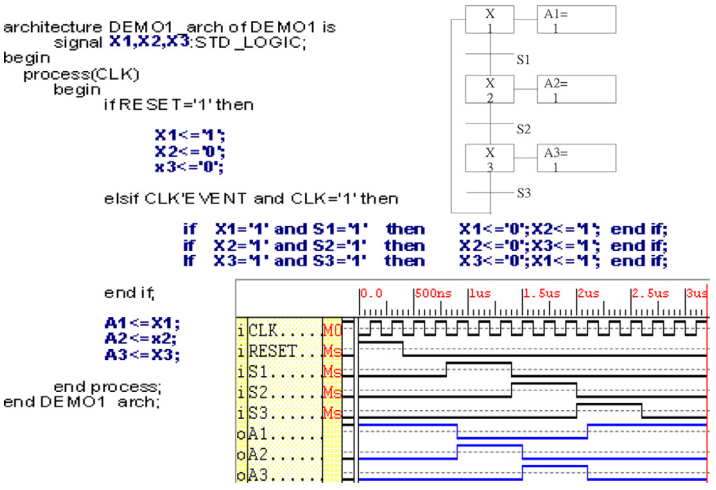

Fig. 8. Hardware synthesis in the development process.

control module to facilitate the task of controlling the master control module and other modules. The **Baudrate** submodule is used to generate specific pulse signals for other modules. The **Fieldbus communication** module is responsible for transmitting and receiving signals.

These submodules integrate the self-test and self-diagnosis functions, thereby satisfying the definition of smart sensors. The **sensor controller (sensor interface)** submodule from IDEF0 is depicted in GRAFCET modeling for the sensor controller in Fig. 9. It comprises the inter-integrated Circuit **(I²C)** and **Dataprocess** submodules, and it is responsible for management

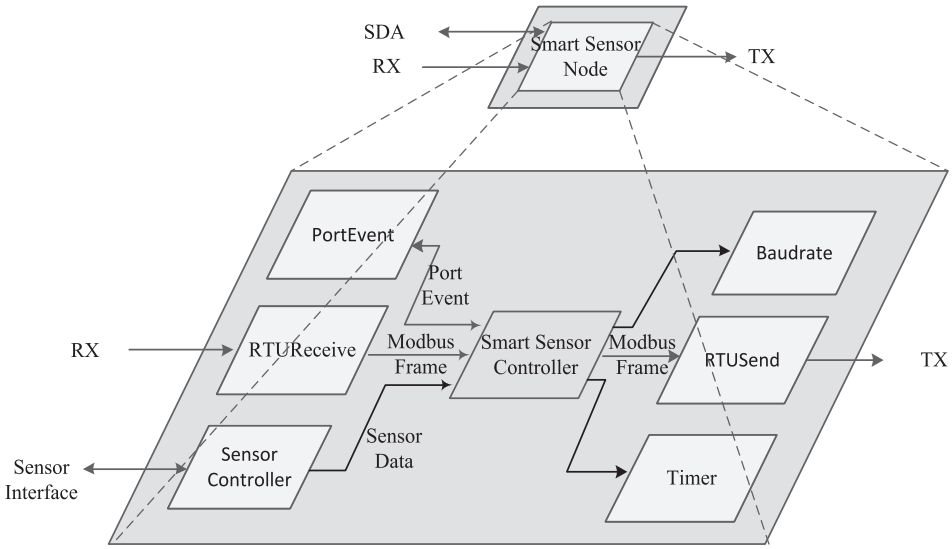

Fig. 9. Hierarchical decomposition of functions for the smart sensor node using IDEF0.

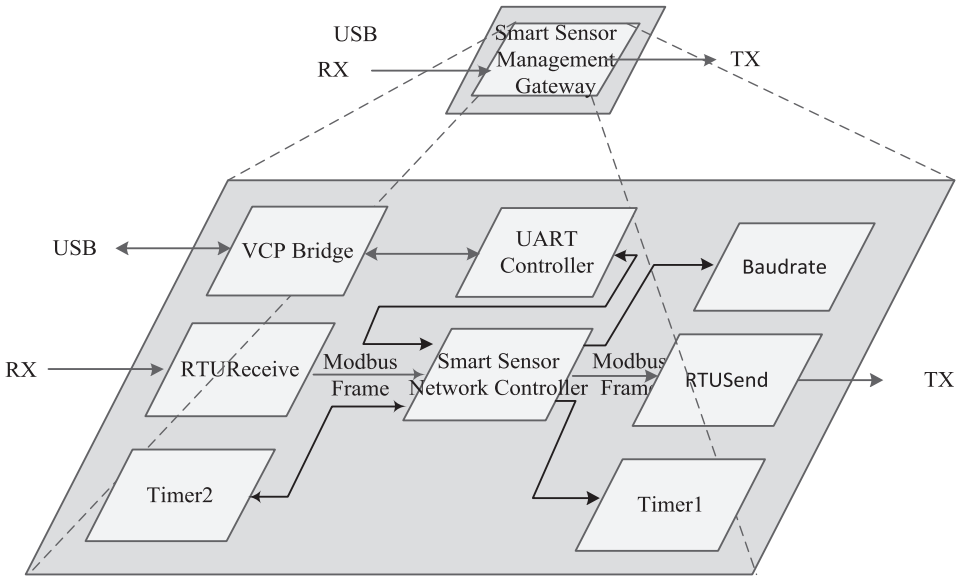

Fig. 10. Hierarchical decomposition of functions for smart sensor management gateway using IDEF0.

and sensor communication, correcting received sensor data, compensating data, and identifying error data. **I²C** is responsible for accessing sensor data. The **Baudrate** submodule of the **function assistance** module provides clock rate signals. The **Dataprocess** submodule processes the received sensor data, including information on nonlinear correction, self-compensation, and data diagnosis.

SSN systems are often associated with industrial devices that communicate with one another and with people through the network. For SSN devices installed in unattended, remote, and harsh environments, a reliable and stable smart sensor management gateway is required between the end device and the cloud server for enabling intelligent big data analysis and intelligent decision making. In this work, we develop a smart management engine (SME) for a smart sensor management gateway that enables users to share and filter data for analysis, thus helping to ensure federated data generated by things, fulfilling the requirements of the SSN system.

As shown in Fig. 10, the hierarchical model of the smart sensor management gateway integrates the **fieldbus communication** module **(RTUReceive** and **RTUSend** submodules**), function assistance** module **(PortEvent, Baudrate**, and **Timer** submodules**), smart sensor network controller** module, and **UART controller** module **(UART controller** and **VCP Bridge** submodules**),** and it mainly controls communication with other modules and packaging processing tasks.

Table 1
Resources used for implementing the SSN. The numbers in brackets indicate the device utilization.

	Smart sensor node controller	Smart sensor gateway controller
Device	Cyclone III EP3C25F256	Cyclone IV EP4CE115F29C7
Total logic element	2,942/24,624 (12%)	3,239/114,480 (3%)
Total pins	10/157 (6%)	12/529 (2%)
Total PLLs	1/4(25%)	1/4(25%)

Our smart sensor management gateway has a modular design, and the gateway can be flexibly configured with different protocol modules to ensure interoperability between systems, thus enabling various applications.

The **smart sensor network controller** module is the core of the smart sensor management gateway, and it mainly controls the communication with other modules, including the **function assistance module** that assists the master control module in completing specific tasks. The **Timer** submodule enables the communication modules to judge the ending signals of the Fieldbus frame and activate the master control module for error processing.

The **Timer** submodule provides regular triggering signals. The completion of the **Timer** is determined through consistently accumulated sensed values and a comparison between input parameters. The external module can set the time through the control port settings. The **PortEvent** submodule records the master control module state signals in the smart sensor nodes. There are four states: preparation, frame reception, frame processing, and frame transmission. Other modules can access the state signals through the control port. The **Baudrate** submodule can set the **Baudrate** through the baud rate parameter input port. The **Baudrate** submodule parameters are input to determine the accumulated counts and to output the corresponding pulse signals.

The **Fieldbus communication module** is responsible for transmitting and receiving signals. The **UART controller** module consists of a parallel-to-serial converter, a serial-to-parallel converter, and Hamming code submodules.

The **smart sensor network controller** module embedded in the SME integrates functions such as signal extraction, system mode translation, sensor addition, insertion, removal, fault detection, error report generation, and automatic ID configuration, thereby satisfying the definition of SSN systems.

5.2. High-level modeling using GRAFECT

In this section, for each independent DES module, we present the GRAFCET model for translation into a VLSI hardware architecture for the smart sensor and gateway controller. The smart sensor node model controls each stage to complete its task in each operation cycle. Fig. 11 shows the GRAFCET modeling corresponding to the DES specification of the smart sensor controller module in the top-level architecture of the smart sensor node.

5.3. High-level synthesis

The smart sensor controller and smart gateway controller are easily obtained from the proposed template model by GRAFCET-to-VHDL synthesis. Fig. 13 shows the VLSI hardware circuit for the smart sensor controller. The VLSI hardware circuit for the smart sensor gateway controller can be obtained through the same synthesis process (see Fig. 14).

6. System integration and performance evaluation

6.1. Experiment

The system hardware architecture of the SSN consists of three major modules, as shown in Fig. 15. The prototype of the SSN system constructed in this work includes a smart sensor node, a smart sensor gateway with an embedded SME, and a PC user interface. Both the smart sensor node and the SSN gateway were implemented and tested on a real FPGA platform (see Fig. 16).

For this study, we also designed a graphical user interface to support the interaction with the SSN system deployed on the network (see Fig. 17). The administrator and users can analyze and mode control the real-time and trend data from the systems to optimize them for parameters such as power efficiency, performance, operational life, and mode control.

6.2. Performance evaluation

In this study, the prototypes of the smart gateway and smart sensor were implemented using the resources shown in Table 1. We validated the hardware system performance improvement through comparison with a previous work; this manuscript compared the embedded system (Xilinx XC5VSX50T) executed using command codes on an FPGA and measured the power consumption, execution times, and instruction cycle of the SSN system by using the controllers. The set of parameters included the number and time required for operating one data set. The results are listed in Table 2.

Table 2
Comparison of the computational costs of the proposed smart sensor controller and gateway controller and those of XC5VSX50T.

Parameter	FPGA-based multiprocessor XC5VSX50T	Smart sensor node controller EP3C25F256	Smart sensor gateway controller EP4CE115F29C7
Power consumption	3.720 W	1.510 W	1.610 W
Execution times	120 ns	50 ns	95 ns
Instruction cycle	12 (Clock cycle)	2 (Clock cycle)	4 (Clock cycle)

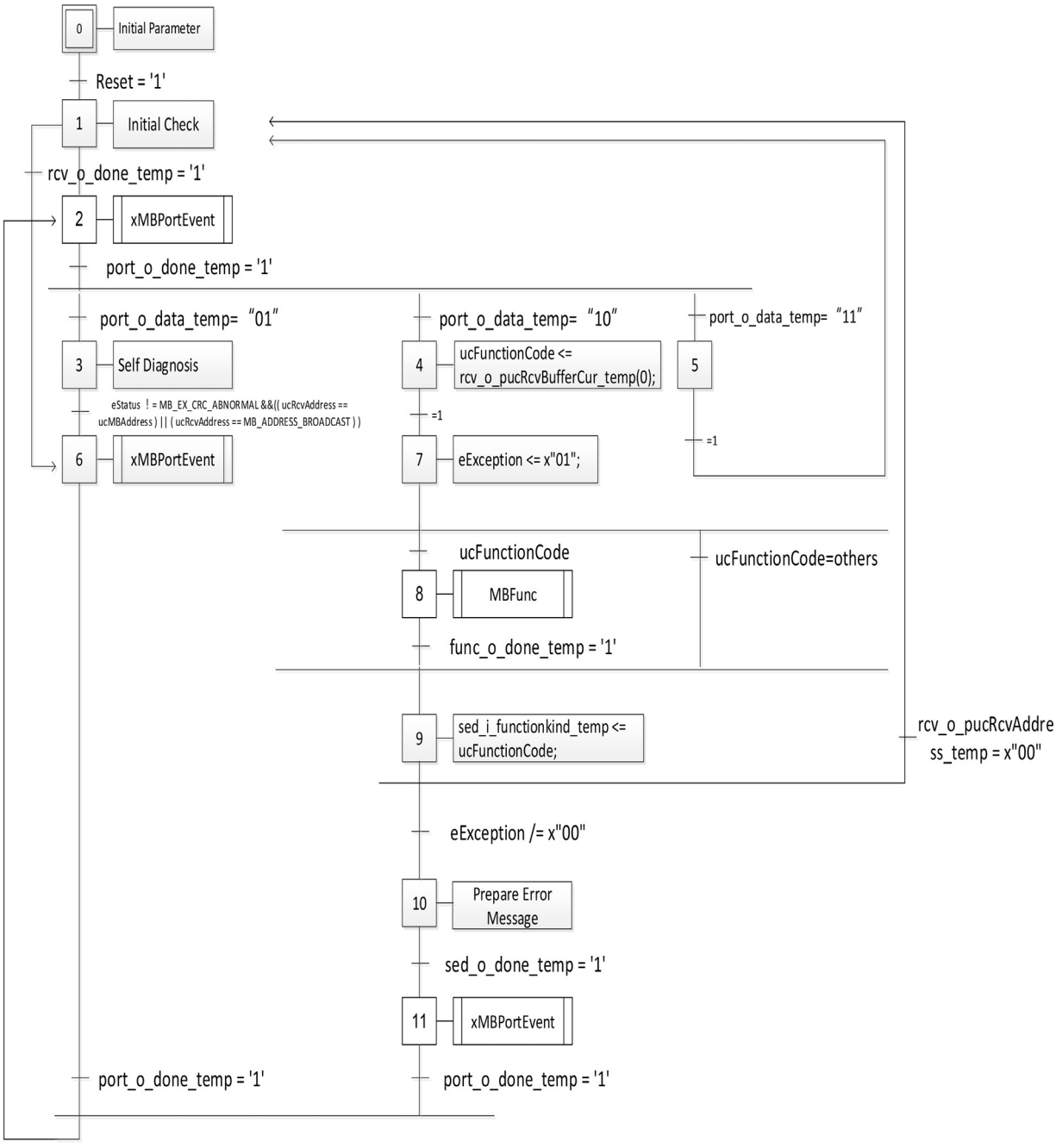

Fig. 11. GRAFCET modeling for smart sensor controller module.

Table 2 compares the implementation of one order of calculations in the XC5VSX50T [3] with the proposed smart sensor controller and gateway controller. To evaluate our coherent and systematic design methodology, an FPGA-reconfigurable SSN system was prototyped and related experimental data were collected. We also demonstrated benchmark testing graphs to determine the performance characteristics of the proposed SSN system.

For the testing, the smart sensor node and gateway were load tested; several performance indices such as power consumption, execution times, and instruction cycle were evaluated. The experiments revealed that adopting our design methodology reduced the power consumption by up to 56.7% with a 56.7% performance improvement compared with another similar system [3] (Figs. 18 and 21). An execution time test indicated that the sensor node controller and smart gateway controller obtained improvements in speed of 58.3% and 20.8%, respectively, with our design methodology (Figs. 19 and 21).

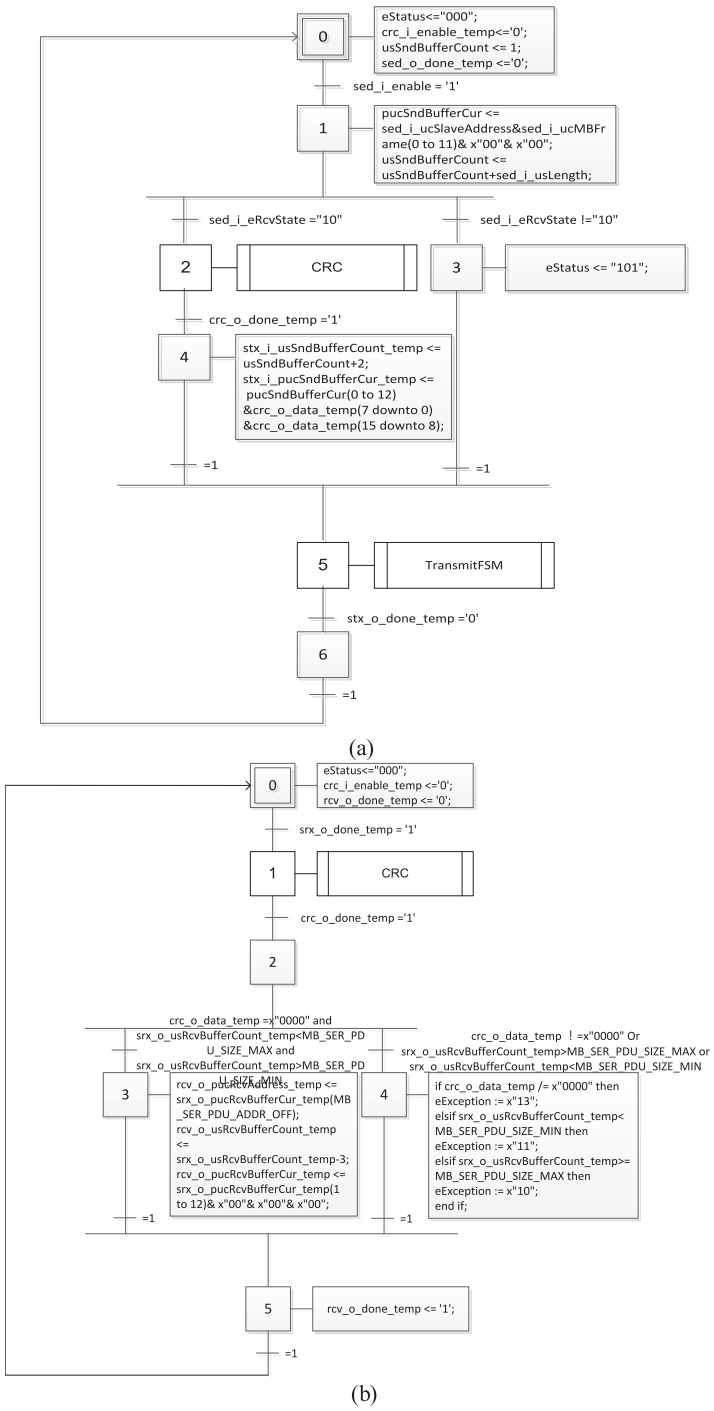

Fig. 12. (a) GRAFCET modeling of the frame transmission submodule and (b) GRAFCET modeling of the frame reception submodule.

As shown in Figs. 20 and 21, the system from [3] required 12 clock cycles over 120 ns. The proposed smart sensor node controller required only 2 clock cycles over 50 ns, and the smart gateway controller required only 4 clock cycles over 95 ns (more than 66.7% reduction in the clock cycle). The measurement results revealed that our proposed solution can potentially outperform an FPGA-based multiprocessor [3]. Furthermore, the results verified that our coherent and systematic design methodology is effective and efficient, and that it is suitable for complex embedded SSN system design.

(1) Real-time analysis: In high-performance synchronous motion control, the required real-time response time was less than 100 ns for server motion control. For a baud rate of 100 Mbps, the mean system response time was 95 ns, which

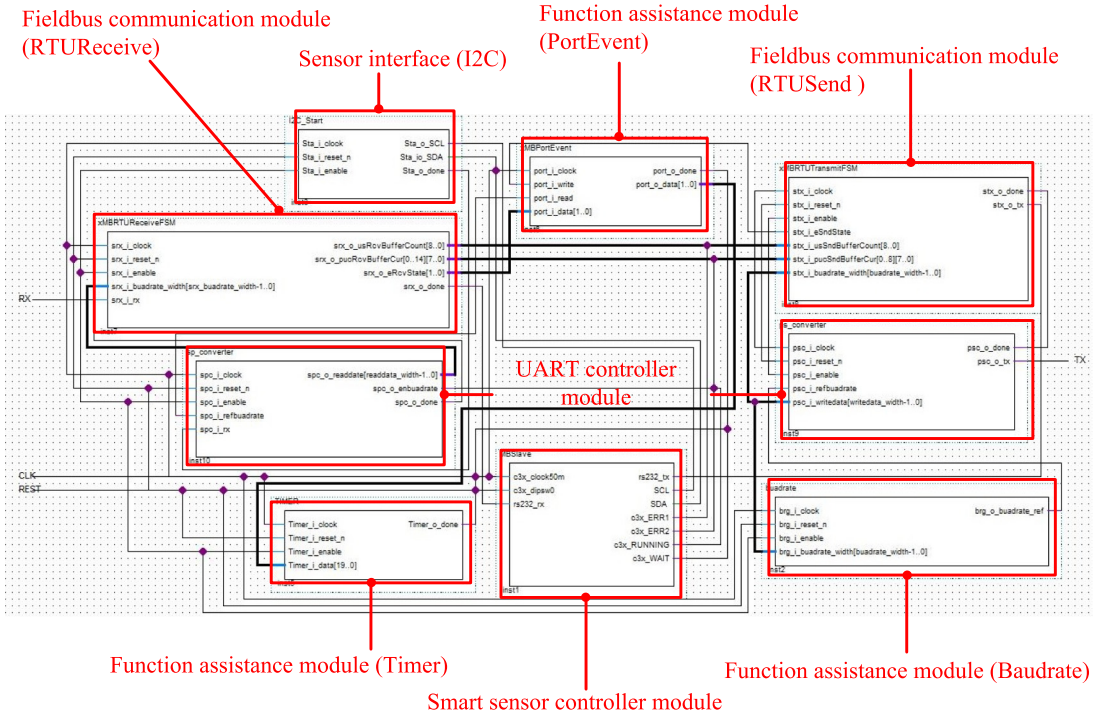

Fig. 13. Synthesized hardware block diagram of the smart sensor controller.

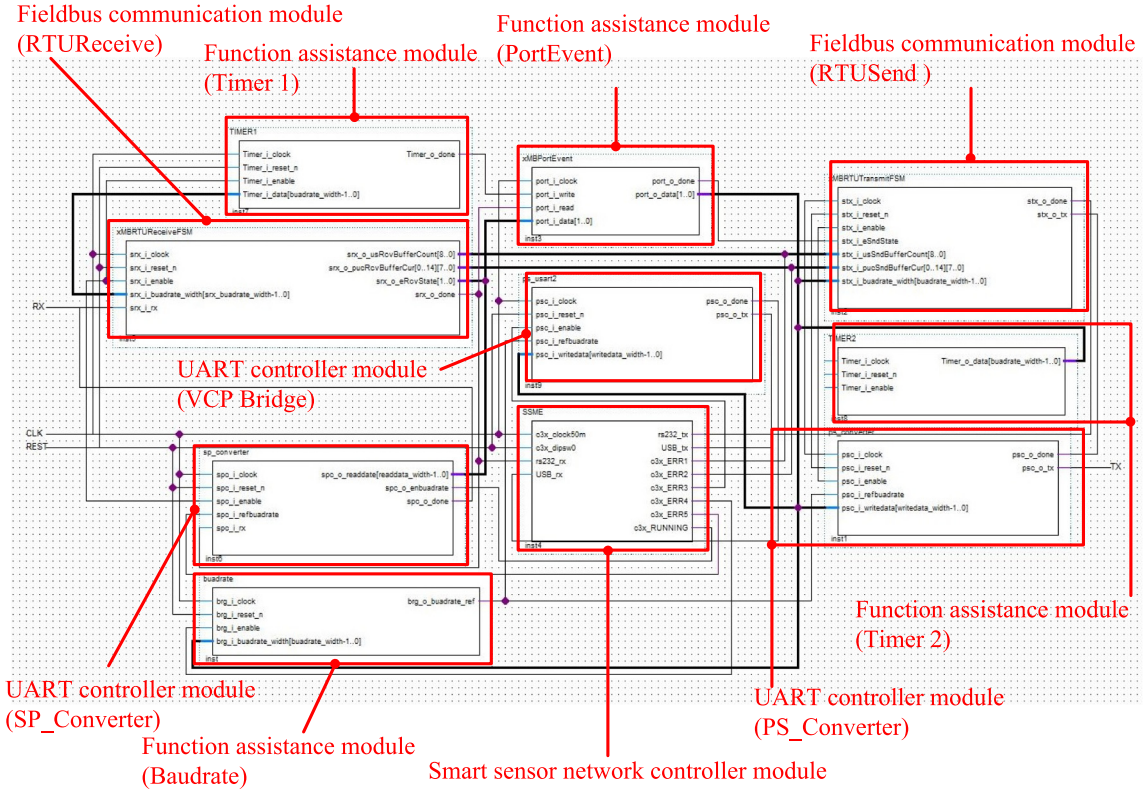

Fig. 14. Synthesized hardware block diagram of the smart gateway controller.

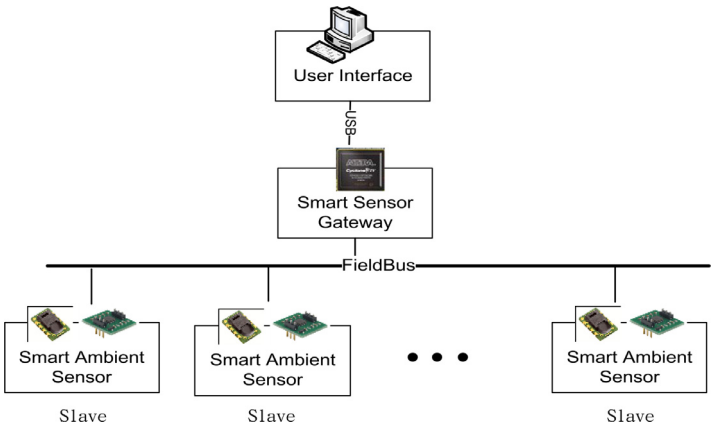

Fig. 15. Hardware architecture of the SSN system.

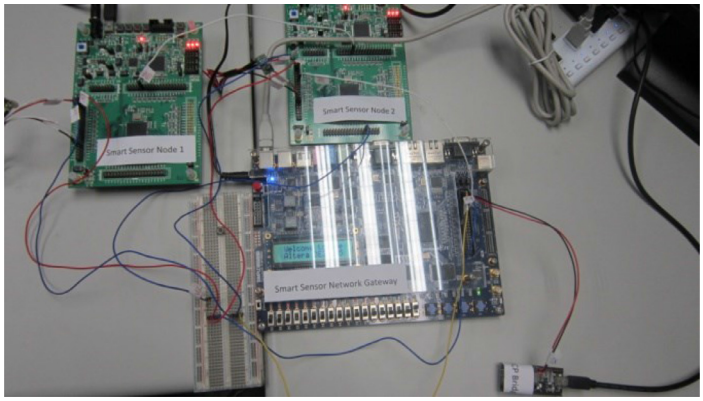

Fig. 16. Experimental platform.

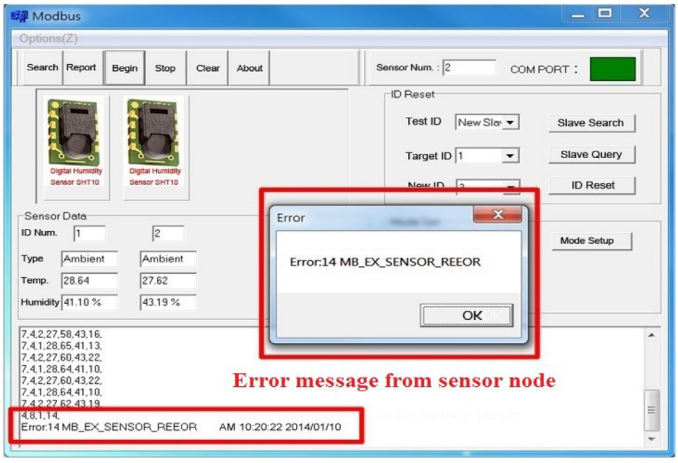

Fig. 17. Graphical user interface.

was considerably less than that in the case of the multiprocessor (120 ns) [3]; thus, the SSN meets the industrial real-time requirement.

(2) Reliability analysis: In the network topology structure, according to the Fieldbus communication protocol, the proposed sensor network uses a master–slave model. The network logic topology structure is shown in Fig. 15. Because all nodes are connected in parallel on the bus, when a problem occurs on the bus, none of the subsequent nodes affect other nodes. In the communication serial link layer, a single-byte Hamming code error correction and a mecha-

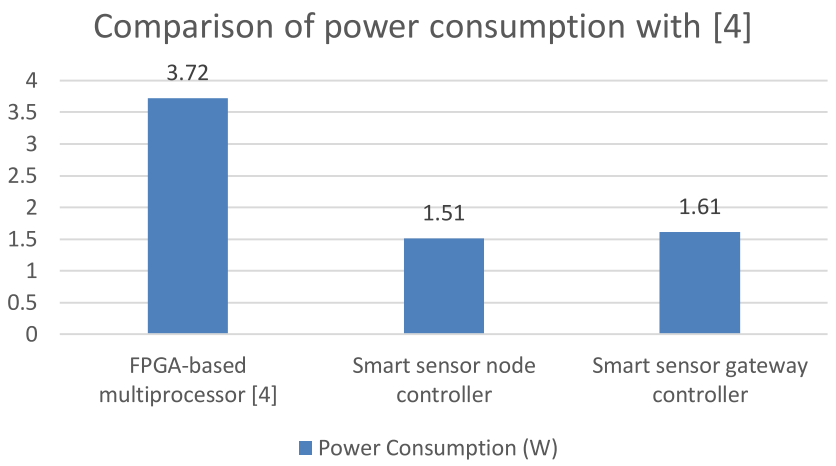

Fig. 18. Comparison of the power consumption of the proposed smart sensor controller and gateway controller and that of XC5VSX50T.

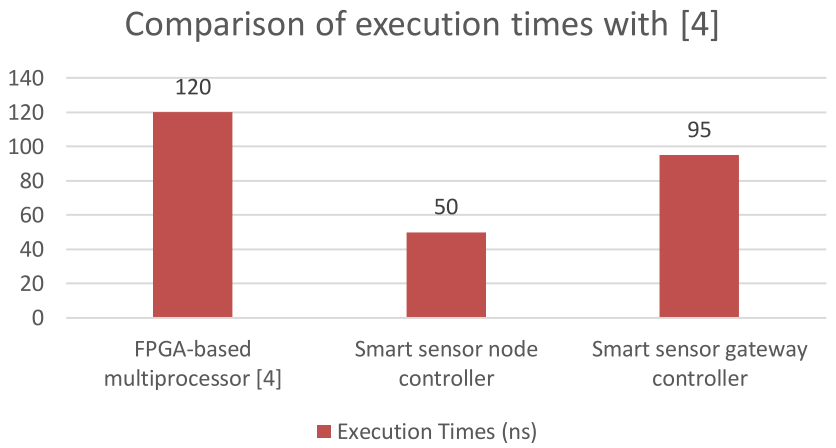

Fig. 19. Comparison of the execution times of the proposed smart sensor controller and gateway controller and that of XC5VSX50T.

nism combining data frame (cyclic redundancy check) CRC and automatic repeat-request were adopted as innovative methods for increasing system reliability.

(3) Power consumption: Our hardware controller is not limited by the number of processing cores available because it is truly parallel. Unlike the multiprocessor in [3], different processing operations have to compete for the same resources. The power consumption of the proposed design was analyzed using the Altera Power Estimator tool and was found to be 3.120 W for the entire system; the power consumptions of the sensor node controller and gateway controller were 1.510 W and 1.610 W, respectively. This is considerably lower than that in the case of the multiprocessor (3.720 W) [3]. The experimental results revealed that the communication reliability of the proposed system meets the required standards.

The hierarchical modular design and functional modules of the SSN system based on the reconfigurable FPGA platform can be used to flexibly add and remove function codes, thus reducing the fabrication cost and time. Moreover, the hardware design of the SSN containing the smart sensor and smart gateway uses a fully hardware-electric circuit without an additional CPU to avoid using software on top of the controller. This can reduce the power loss efficiency of the system as well as its computational complexity. Furthermore, this enables convenient module expansion and future modifications. This work attempted to use hardware methods to fabricate an SSN system that achieves the goals of communication reliability, minimum power consumption, and energy conservation.

7. Conclusions and perspectives

In this work, we first reviewed the background on SSN systems and analyzed research challenges in SSN systems used for IIoT. On the basis of a coherent and systematic design methodology, we subsequently formulated a hierarchical design according to the modular functional decomposition of an SSN system. By using GRAFCET modeling and VHDL synthesis,

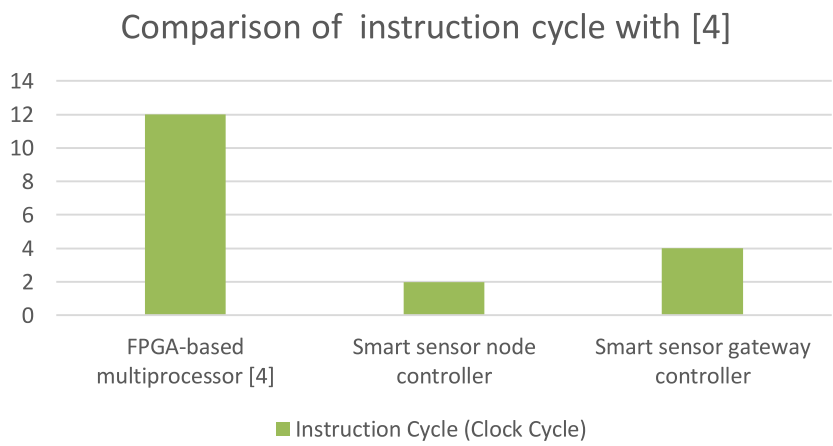

Fig. 20. Comparison of the instruction cycle of the proposed smart sensor controller and gateway controller and that of XC5VSX50T.

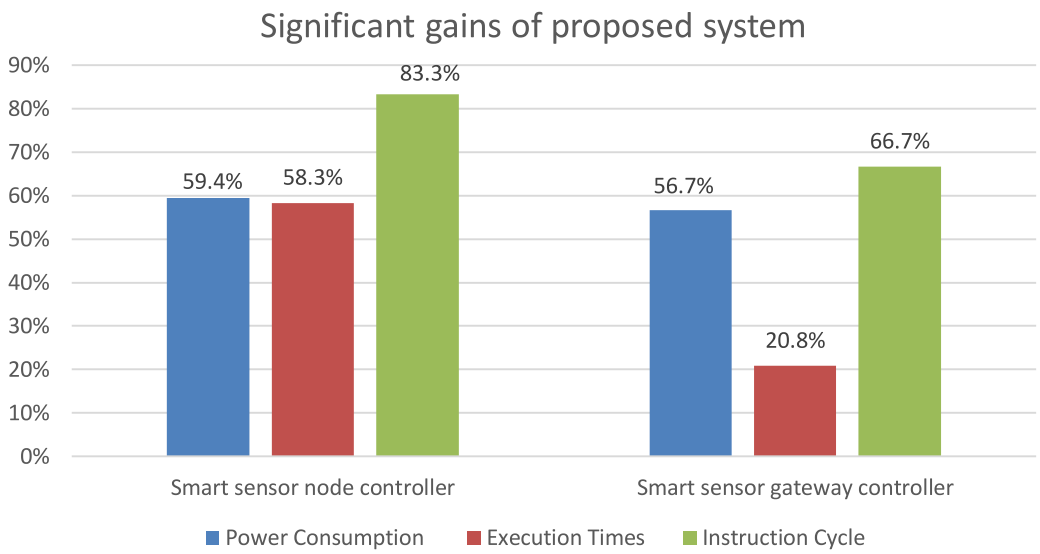

Fig. 21. Significant gains of the proposed smart sensor controller and gateway controller and those of XC5VSX50T.

efficient controller hardware for the SSN system can be obtained rapidly. The synthesized controllers were implemented on an FPGA, and they exhibited excellent real-time performance.

The smart gateway controller with an embedded SME module also performs functions such as signal extraction, mode control, node addition, node insertion, node removal, fault detection, error report generation, and automatic ID configuration, thereby satisfying the definition of an SSN system. Therefore, the automatic generation of the SSN controller in a concurrent system is advantageous for the rapid design and synthesis of a complex SSN system.

Moreover, unlike previous methods, the proposed approach is advantageous for complex applications in the consumer electronics and communications industries, which generally require low development cost and short time-to-market.

Because of the limitations of the FPGA hardware implementation, our SSN sacrifices portability and re-configurability to achieve high performance. In the future, a processor could be added to this SSN architecture to realize reconfigurable SSN functionalities for providing greater design flexibility for application developers.

References

[1] Gubbi J, Buyya R, Marusic S, Palaniswami M. Internet of Things (IoT): a vision, architectural elements, and future directions. Future Gen Comput Syst 2013;29:1645–60.
[2] Da Xu L, He W, Li S. Internet of things in industries: a survey. IEEE Trans Ind Informat 2014;10:2233–43.
[3] Echanobe J, delCampo I, Basterretxea K, Martinez M, Doctor F. An FPGA-based multiprocessor-architecture for intelligent environments. Microproc Microsyst 2014;38:730–40.
[4] Kortuem G, Kawsar F, Fitton D, Sundramoorthy V. Smart objects as building blocks for the internet of things. IEEE Internet Comput 2010;14:44–51.
[5] Wu Y, Sheng QZ, Zeadally S. RFID: opportunities and challenges. In: Chilamkurti N, editor. Next-generation wireless technologies; 2013. p. 105–29.
[6] Sanislav T, Mois G, Miclea L. An approach to model dependability of cyber-physicalsystems. Microproc Microsyst 2016;41:67–76.

[7] Xia F. Wireless sensor technologies and applications. Sensors 2009;9(11):8824–30.
[8] Mujica G, Portilla J, Riesgo T. Performance evaluation of an AODV-based routing protocol implementation by using a novel in-field WSN diagnosis tool. Microproc Microsyst 2015;39:920–38.
[9] Ghosh A, Das SK. Coverage and connectivity issues in wireless sensor networks: a survey. Pervasive Mobile Comput 2008;4:303–34.
[10] Barba J, Rincón F, Moya F, Dondo JD, López JC. A comprehensive integration infrastructure for embedded system design. Microproc Microsyst 2012;36:383–92.
[11] Millan-Almaraza JR, Torres-Pachecob I, Duarte-Galvanb C, Guevara-Gonzalezb RG, Contreras-Medinab LM, Romero-Troncosoc RJ. FPGA-based wireless smart sensor for real-time photosynthesis monitoring. Comput Electron Agri 2013;95:58–69.
[12] Chi Q, Yan H, Zhang C, Pang Z, Xu LD. A reconfigurable smart sensor interface for industrial WSN in IoT environment. IEEE Trans. Ind. Informat. 2014;10:1417–25. doi:10.1109/TII.2014.2306798.
[13] Bao S, Yan H, Chi Q, Pang Z, Sun Y. A FPGA-based reconfigurable data acquisition system for industrial sensors. IEEE Trans Ind Informat 2016. doi:10.1109/TII.2016.2641462.
[14] Perera MDR, Meegama RGN, Jayananda MK. FPGA based single chip solution with 1-wire protocol for the design of smart sensor nodes. J Sensors 2014;2014. http://dx.doi.org/10.1155/2014/125874.
[15] Ortega-Zamoranoa F, Jereza JM, Subiratsa JL, Molinab I, Francoa L. Smart sensor/actuator node reprogramming in changing environments using a neural network model. Eng Appl Artif Intell 2014;30:179–88.
[16] Guesmi H, Salemc SB, Bachac K. Smart wireless sensor networks for online faults diagnosis in induction machine. Comput Elect Eng 2015;41:226–39.
[17] Morales-Velazqueza L, Romero-Troncosob RJ, Herrera-Ruiza G, Morinigo-Soteloc D, Osornio-Rios RA. Smart sensor network for power quality monitoring in electrical installations. Measurement 2017;103:133–42.
[18] Chen CH, Lin MY, Lin WH. Designing and implementing a lightweight WSN MAC protocol for smart home networking applications. J Circuit Syst Comp 2017;26. doi:10.1142/S0218126617500438.
[19] Cahyani NDW, Martini B, Choo KKR, Al-Azhar AKBPMN. Forensic data acquisition from cloud-of-things devices: windows smartphones as a case study. Concurr Comput 2016. doi:10.1002/cpe.3855.
[20] Bertino E, Choo KKR, Georgakopolous D. Internet of Things (IoT): smart and secure service delivery. ACM Trans Internet Technol (TOIT) 2016;16. http://dx.doi.org/10.1145/3013520.
[21] DOrazio CJ, Choo KKR, Yang LT. Data Exfiltration from Internet of Things devices: iOS devices as case studies. IEEE Internet of Things J 2016. doi:10.1109/JIOT.2016.2569094.
[22] Teing YY, Dehghantanhab A, Choo KKR, Yang LT. Forensic investigation of P2P cloud storage services and backbone for IoT networks: Bit Torrent Sync as a case study. Comput Elect Eng 2016. In Press http://dx.doi.org/10.1016/j.compeleceng.2016.08.020..
[23] Chen CH, Kuo CM, Chen CY, Dai JH. The design and synthesis using hierarchical robotic discrete-event modeling. J Vib Control 2012. doi:10.1177/1077546312449645.
[24] Chen CH, Lin MY, Guo XC. High-performance fieldbus application-specific integrated circuit design for industrial smart sensor networks. J Supercomput 2017:1–19. doi:10.1007/s11227-017-2010-1.

Ching-Han Chen received the D.E.A and Ph.D. degree in 1992 and 1995 from the Franche-Comte University, France. He was an associate professor in the department of electrical engineering, I-Shou University from 1995. Since 2006, he is now an associate professor in the department of CSIE, NCU, Taiwan. His research interests include embedded system, machine vision, and robotics.

Ming-Yi Lin received the M.S. degrees in Department of Computer Science and Engineering from Yuan-Ze University, Taiwan R.O.C., in 2011. He is currently pursuing the Ph.D. degree in Department of Computer Science and Information Engineering at National Central University, Taiwan. His research interests include embedded systems, Wireless Sensor Networks, and Internet of Things.

Xing-Chen Guo received Master degree of computer science from National Central University in 2013. His research interests include embedded system, Fieldbus protocol and smart sensor network.