Nuclear Engineering and Design 241 (2011) 2597–2603

Contents lists available at ScienceDirect

Nuclear Engineering and Design

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

A Petri net design of FPGA-based controller for a class of nuclear I&C systems

Chang-Kuo Chen*

Institute of Nuclear Energy Research, No. 1000, Wenhua Road, Chiaan Village, Longtan Township, Taoyuan County 32546, Taiwan, ROC

ARTICLE INFO

Article history:
Received 11 October 2010
Received in revised form 29 March 2011
Accepted 1 April 2011

ABSTRACT

This study is concerned with a FPGA-based controller design for the lack of FPGA-based solutions in the nuclear industry. An efficient design procedure is proposed to achieve simpler and affordable verification and validation (V&V) of system efforts by explicitly modeling the interactions among processes. In the present approach, both of state diagram (SD) concept and Petri nets (PNs) are used to model the concurrent processes. An illustrative example of automatic seismic trip system (ASTS) is provided. Synthesis results demonstrate that the proposed design is feasible and easy to implement.

1. Introduction

Due to the effects of aging, the problems of obsolescence, environmental degradation, and mechanical failures in technologies of conventional relay and analog electronics have become an important issue for original Nuclear Power Plant (NPP) I&C systems. Therefore, the industrial base has greatly started to replace these older systems with digital-based systems such as microprocessor-based equipments and the needed analog spare parts are discontinued support by vendors progressively. So far, microprocessor-based systems are the major I&C designs for new plants because of they are extensively employed in the operation NPPs (EPRI, 2009; Zio and Di Maio, 2009).

Unfortunately, the safety applications and gaining regulatory approval are hardly and expensively realized by microprocessor-based systems in experiential due to the systems are complex as large amounts of software in operating systems and customized application software of other platforms. Also, microprocessors tend to become obsolete much more quickly than analog equipment they have replaced owing to the particular processor architecture cannot provide for long-term support. Hence, other electronic replace solutions for use in I&C systems should be proposed.

Recently, field-programmable gate arrays (FPGAs) and other programmable logic devices provide the capability to protect against obsolescence by circuit independent, i.e., portable to different FPGA architectures, and reduce complexity for regulatory approval as compared to conventional microprocessor-based systems with purely hardware architecture and high clock speeds. Therefore, FPGA-based systems and equipment are beginning to appear in new plant I&C applications, as well as in retrofits for operating plants, in particular for safety applications that makes verification and validation (V&V) efforts simpler and affordable (EPRI, 2009). Several international experiences in use of FPGA-based nuclear I&C systems are proposed (Clarkson, 2009; Zupan et al., 2008; Miyazaki et al., 2009; Yastrabenetsky et al., 2009; Nguyen, 2008).

FPGAs have been widely applied in other industries and installed in some nuclear power plants, and the FPGA-based ancillary circuits on the part of both regulators and licensees are obtained little direct attention. Although it started to become a core role in the implementation of nuclear plant I&C functions, it still not a perfect architectures. As it is a new technique and relative inexperience for nuclear industry, there has no international standard guidance and requirements been published for FPGA-based solutions over the past few years. Therefore, NRG/CR 7006 is proposed to address the gaps in existing regulatory documents that can be used as guidance to review FPGA-based safety systems in nuclear power plants (Bobrek et al., 2009). Also, based on the existing international documents and the lessons learned, the upcoming IEC standard, IEC 62566 is being developed that specifically addresses FPGAs and related devices (EPRI, 2009).

Commonly, the actions of computer systems can be depicted and analyzed by means of transition systems. The most frequently used transition systems are based on clear state enumeration namely state diagram (SD). SD becomes one of the key formalisms underlying the general approaches to hardware description and synthesis, which represent the transformation between inputs and outputs for sequential designs graphically.

In addition, Petri nets (PNs) are a prevailing modeling formalism at the discrete-event level in computer science, system engineering and many other subjects. It has long been confirmed to be suitable for describing and analyzing the discrete-event systems with significant advantages such as the graphical nature can visualize sequences of firing via token passing over the net, well-established formal mechanisms for modeling and analysis of systems and the mathematical base for analyzing structural and dynamic behaviors

* Tel.: +886 3 4711400x6312; fax: +886 3 4711415.
E-mail address: cahngkuochen@iner.gov.tw

0029-5493/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.nucengdes.2011.04.004

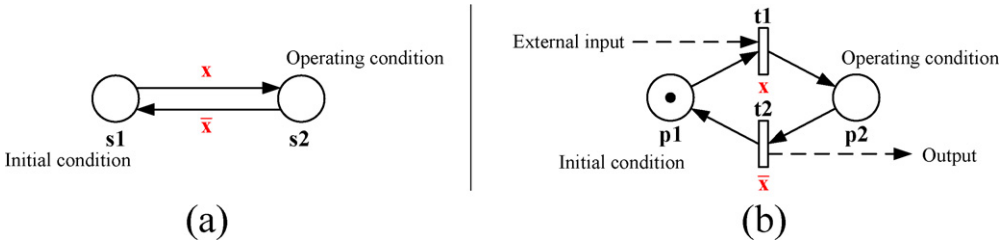

Fig. 1. Modeling of state diagram concept to equivalent labeled PN for local behavior in FPGA.

of a system. Consequently, PNs based on the theoretical analysis and realization have been thoroughly developed and studied (Lee and Seong, 2004; Németh et al., 2009; Murata, 1989; Zhou and Jeng, 1998; Lee, 2008; Giua and DiCesare, 1994).

In order to simplify the construction of Hardware Description Language (HDL) sequences or properties in use of comprehensive advanced simulation techniques including black box verification, assertion-based verification, and formal verification along with the code coverage analysis. The objective of this study is to propose a simple procedure of FPGA-based controller design for modeling and verifying systems of concurrent processes, which provide a precise textual representation of the design requirements and give unambiguous instructions to the logic synthesis with both SD concept and PNs.

The paper is organized as follows. In Section 2, the basic concepts of system modeling are presented. Section 3 describes the main result of FPGA-based controller design. An illustrated example of an automatic seismic trip system is given to show the feasibility in Section 4, and the brief conclusions are brought in Section 5.

2. System modeling

2.1. Basic Petri nets

A Petri net is a particular kind of bipartite directed graph by three kinds of objects. Those are places, transitions, and directed arcs connecting places to transitions and transitions to places. For the places and transitions, they are drawn by circles and bars/boxes, respectively. A basic Petri net is formally defined as

$$PN = (P, T, I, O) \tag{1}$$

where $P=\{p_1, p_2, \ldots, p_n\}$, $n>0$ is a finite set of places; $T=\{t_1, t_2, \ldots, t_m\}$, $m>0$ is a finite set of transitions with $P\cup T \neq \emptyset$ and $P\cap T=\emptyset$;

$I: P\times T\rightarrow Z^+$ is an input function defined the set of directed arcs from P to T in which Z^+ is a set of nonnegative integers.

$O: P\times T\rightarrow Z^+$ is an output function defined the set of directed arcs from T to P.

Furthermore, either none or a positive number of tokens is potentially held in each place as pictured by small solid dots. The distribution of tokens is referred as the marking. A Petri net which containing tokens is called as marked Petri net and its transitions can be enabled and fired immediately. In a basic Petri net, the enabling and firing rules are given as follows. $M: P\rightarrow Z^+$ is an marking in which i-th component represents the number of tokens in i-th place. A transition $t\in T$ is enabled, if and only if $M(p)>0$ when $I(p, t)=1$, $\forall p\in P$. Moreover, an enabled transition t removes tokens from its input places and deposits them on its output places when it fires at a marking M, the new marking M' is given as

$$M'(p_i) = M(p_i) + O(p_i, t) - I(p_i, t) \quad for \quad i = 1, 2, \ldots, n \tag{2}$$

In addition, Petri net has various properties which permit the system designer to identify the presence of lack of the specific functional properties under system design in the framework of the modeled system. Reachability, boundedness, conservativeness, liveness and reversibility are significant properties of Petri net. Those properties can be analyzed by some validation methods for example reachability graph, invariant analysis, reduction and simulation (Zhou and Jeng, 1998).

2.2. Modeling of SD concept to PN for FPGA design

In this study, one typical approach is proposed, that is applying SD concept and PNs for the modeling and verification of concurrent processes in FPGA design. Firstly, the SD concept is used to describe the local behavior of each process and then a global view of the system is given by PN for modeling the interactions in concurrent processes explicitly. SD concept is a basic component in FPGA design such which represents the transformation between inputs and outputs for sequential designs. A basic SD is defined as

$$SD = (S, Q, R, f, g) \tag{3}$$

where S is the set of state space, $S=\{s_1, s_2, \ldots, s_n\}$, $n>0$; Q and R denote the input and output symbols; f and g are the state transition and the output functions, respectively. Moreover, defined a labeled PN, lPN which is obtained from an ordinary $PN=(P, T, I, O)$.

$$lPN = (P, T, X, Y, I, O, l) \tag{4}$$

where P and T are the sets of places and transitions, respectively; I and O are denoted the functions giving the input and the output places of each transition by using a set X of input symbols and a set Y of output symbols. Moreover, a labeling function $l: T\rightarrow X\times Y$ mapping each transition onto the input symbol and output symbol. The semantics of the transitions are also syntactically apparent in labeled PNs. For example $l_x(t)=u$ and $l_y(t)=v$, or equivalently $l(t)=(u, v)$, means the transition t to consist in receiving the symbol $u\in X$ and in sending the symbol $v\in Y$. More detailed justification is illustrated in (Pattavina and Trigila, 1984).

Without loss of generality, the simple modeling of SD concept to equivalent labeled PN for local behavior of submodule in FPGA is shown in Fig. 1. The marking conditions of places describe the status of the local behavior in modeling process. A transition includes a set of input and output places which represent the pre-conditions and pro-conditions of the event, respectively. An external input associated with its transition represents as a firing condition, which triggers an operation condition. The external input means that the related transition is fired with respect to the external input. Otherwise, the related transition is fired by itself such as source or generator. The output associated with its transition refers to the output of local behavior when the related transition is fired. Otherwise, the related transition is fired by itself such as sink. The associated condition is true or that the actions associated with this place are taken is depicted by the presence of a token in a given place. In Fig. 1(b), a token is in initial condition p1 initially. When the parameter x becomes active ($x=1$) the transition t1 will fire, then the token will move to p2 and remain. Similarly, when $x=0$ and the token will pass back to p1. Since the labeled PNs captured duplicate and hidden tasks naturally are introduced only for the sake of convenience to apply the subsequent interconnection algo-

rithm straightforwardly. For simplicity of presentation, a labeled PN is referred to as PN when no ambiguity can occur in the following sections.

3. FPGA-based controller design by Petri nets

3.1. Synthesis of concurrent processes by communication rules

In order to describe the concurrent processes among the local behavior of each process by modeling of SD concept to equivalent PN, this section will discuss the communication rules for concurrent processes synthesis. The following definitions are given for concurrent processes synthesis.

Definition 1. A set $F_1, F_2, \ldots, F_k$ of SDs is said to be communicating if for any each F_i, $i = 1, 2, \ldots, k$ there exists at least a SD_j, $j \neq i$ such that

$$(R_i \cap Q_j) \cup (R_j \cap Q_i) \neq 0 \tag{5}$$

For instance a pair F_1 and F_2 of SDs is called the communicating if some outputs from the former are input to the latter, or vice versa, i.e. $(R_1 \cap Q_2) \cup (R_2 \cap Q_1) \neq 0$.

Definition 2. Given two disjointed nets $N_1 = (P_1, T_1, I_1, O_1)$ and $N_2 = (P_2, T_2, I_2, O_2)$ with initial marking $M_{0,1}$ and $M_{0,2}$, respectively. Let $N_3 = (P_3, T_3, I_3, O_3)$ with initial marking $M_{0,3}$ and $T_3 \subset T_1 \cup T_2$ is a net of communication rules. A net $N = (P, T, I, O)$ with initial marking M_0, the concurrent composition of N_1, N_2 with N_3, is denoted as

$$N = N_1 || N_2 || N_3 \tag{6}$$

where

$$P = P_1 \cup P_2 \cup P_3;$$

$$T = T_1 \cup T_2;$$

$I(p, t) = I_i(p, t)$ if $(\exists i \in \{1, 2, 3\}, \exists j \in \{1, 2\})[p \in P_i \wedge t \in T_j]$, else $I(p, t) = 0$;
$O(p, t) = O_i(p, t)$ if $(\exists i \in \{1, 2, 3\}, \exists j \in \{1, 2\})[p \in P_i \wedge t \in T_j]$, else $O(p, t) = 0$;
$M_0(p) = M_{0,1}(p) \cup M_{0,2}(p) \cup M_{0,3}(p)$.

Generally, $N = (P, T, I, O)$ obtained through the above construction does not make the proper communication rule from which a final state cannot be approached due to the unpredictable conditions as deadlock. Therefore, the PN analysis should be included to further refine and restrict the behavior of net N. Consequently, the design procedure for a set of r communicating SDs converts into a global PN by applying the following steps:

Step 1: Convert each SD, $F_h = \{S_h, Q_h, R_h, f_h, g_h\}$, $h = 1, 2, \ldots, r$ into a labeled PN, $lPN_h = \{P_h, T_h, X_h, Y_h, I_h, O_h, l_h\}$;
Step 2: Compose the specific communication rules for each disjointed net pair $lPN_h, lPN_k, h, k = 1, 2, \ldots, r; h \neq k$ by creating new places u_{hk} and v_{kh} for all $u \in R_h \cap Q_k$ and $v \in R_k \cap Q_h$, respectively.
Step 3: Make $O(t_h) := O(t_h) \cup \{u_{hk}\}$ and $I(t_k) := I(t_k) \cup \{u_{hk}\}$ by specific communication rules for all $t_h \in T_h \,|\, u = l_y(t_h)$ and $t_k \in T_h \,|\, u = l_x(t_k)$, respectively.
Step 4: Perform $O(t_k) := O(t_k) \cup \{v_{kh}\}$ and $I(t_k) := I(t_k) \cup \{v_{kh}\}$ by specific communication rules for all $t_k \in T_k \,|\, v = l_y(t_k)$ and $t_h \in T_h \,|\, v = l_x(t_h)$, respectively.

3.2. Elementary communication rules for Petri nets

At the modeling stage, FPGA design is needed to have well compliance with the relationship of communication rules in their sequential or precedent, concurrent, or conflicting relationship. The basic relations among these processes or operations of local behavior are classified as follows:

a. *Sequential*: The places and transitions are formed by a sequential or cascade relation in PNs when one operation follows the other as shown in Fig. 2(a).
b. *Conflicting*: Two of more transitions form the outputs from the same place when either of two or more processes can follow an operation as shown in Fig. 2(b).
c. *Concurrent*: Two or more places are formed by a parallel structure starting with a transition when two or more processes are initiated by an event as shown in Fig. 2(c).
d. *Cyclic*: A cyclic structure is formed by a sequence of processes follow one after another and resumes the first one when the completion of the last one as shown in Fig. 2(d).
e. *Mutually Exclusive*: Two processes cannot be performed at the same time due to constraints on the usage of share resources called mutually exclusive as shown in Fig. 2(e). A common place marked with one token plus multiple output and input arcs to activate these processes is realized this structure.

In this study, PN models of the FPGA-based controller design will be constructed based on these elementary communication rules.

3.3. Implementation by means of FPGA

When a concurrent process is modeled as a Petri net, a set of places represents its local process and the maximal subset of simultaneously marked places defines the global the local process of the controller. The local process changes mostly forced by the external inputs are described by transitions. Based on a set of event-oriented (Petri net transition-oriented) *if–then* conditional statements, one-to-one encoding (Wegrzyn, 2003) was developed to translate into FPGA specification format such as HDL or netlist format. The flip-flop-based one-to-one encoding of Petri net is regarded as the simplest case of more general mapping that produces fast designs with a simple combinational part, particularly for implementations of field programmable logic (FPL) in FPGA. Therefore, it is applied to realize the concurrent process as composed PN model directly mapping into specification format for FPGA-based controller design in this study. Consequently, the details of the proposed FPGA-based controller design by Petri nets are described thoroughly as follows:

Step 1: Construct the SD for the local behavior of each process;
Step 2: Convert each graphic SD into a PN model;
Step 3: Define the specific communication rules for concurrent process of overall system.
Step 4: Compose each disjointed PN model pair with specific communication rule to synthesize the preliminary PN model.
Step 5: Analyze and verify the properties of the composed PN model.
Step 6: Refine the model to obtain a deadlock-free, bounded, and reversible model according to the defined communication rules.
Step 7: Implement the local behavior of each process by using the most commonly HDLs, VHDL or Verilog in accordance with corresponding SD.
Step 8: Utilize the one-to-one encoding method to translate communication rules of each process into FPGA specification format directly for conjoining the each process based on the composed PN model.

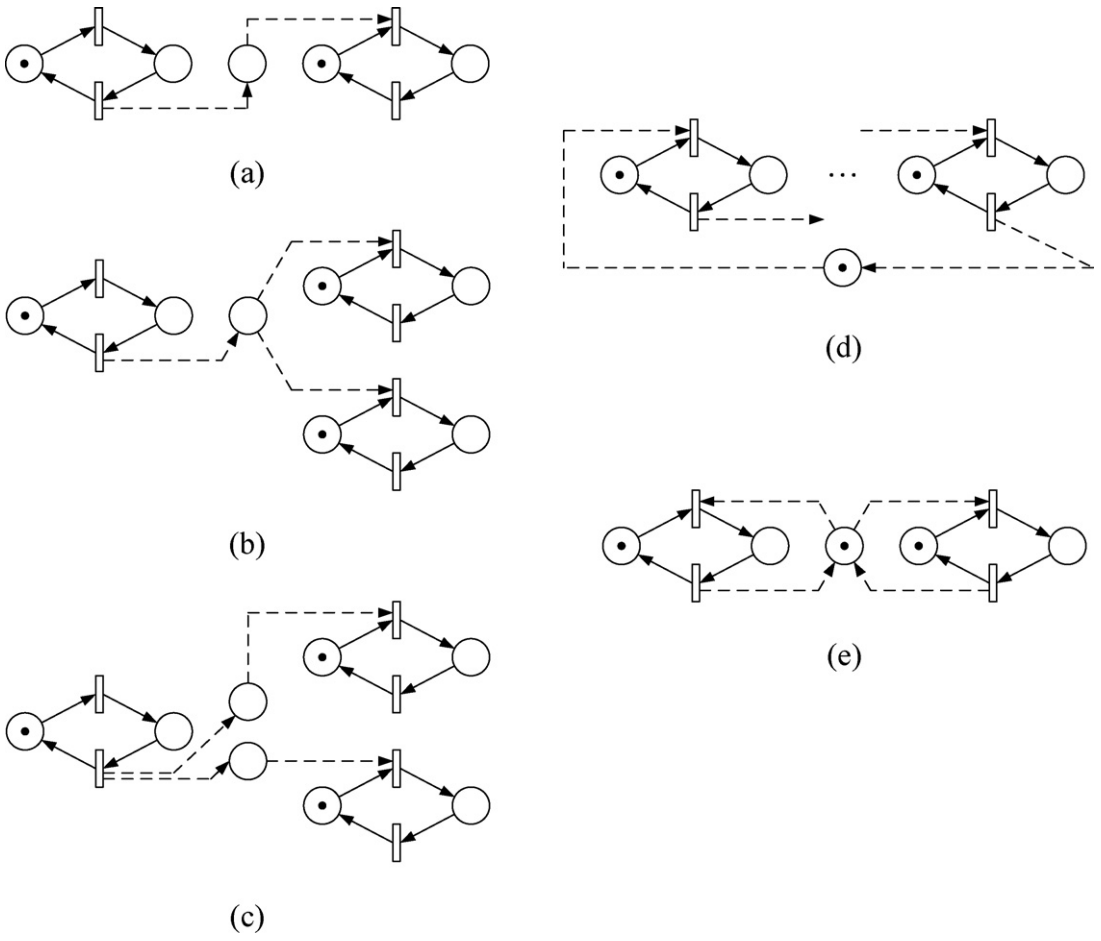

Fig. 2. Basic communication rules of PN modeling for (a) sequential, (b) concurrent, (c) conflicting, (d) cyclic, and (e) mutually exclusive in FPGA.

4. Illustrated example: automatic seismic trip system

4.1. System description

The automatic seismic trip system (ASTS) is applied to Reactor Protection System (RPS) in nuclear power plant (NPP), which consists of sensor box and controller cabinet as shown in Fig. 3. The FBA-23s produced by Kinemetrics is used as earthquake sensors in sensor box. It is composed of three force-balance accelerometers inside a single cast-aluminum case, capable of measuring ground motion in the transverse, vertical and longitudinal planes simultaneously. Signal conditioner includes band pass filter (BPF), bipolar to unipolar converter (BUC) and voltage to current converter (VAC), is also set in sensor box. BPF is proposed to eliminate the noise other than seismic signal for avoiding the system malfunction, particularly for the harmonic interference from power supply. BUC is used to convert FBA signal to positive signal only for set-point comparison and VAC is designed to change FBA output to current for long distance transmission support, respectively. Controller cabinet is constructed by Foxboro Spec-2000 controller with several modules.

Firstly, signals from three planes are measured by earthquake sensors periodically. Secondly, the measured signals are dealt with BPF, BUC and VAC for elimination of noise, voltage regulation, and signal conversion, respectively. Finally, the trip logic is performed by Foxboro Spec2000 controller, and then the trip signal is sent to the output relay for operation of the reactor trip when one of earthquake signals is exceeded the set-point value.

In this study, the FPGA is developed to replace the parts of Foxboro Spec2000 controller as shown in the gray area of Fig. 3. The real works include data acquisition unit, average calculator, set-point comparison unit, trip signal generator, and monitor unit. The concurrent process of overall system is described as follows.

Step 1: Receive the earthquake signals by data acquisition unit.
Step 2: Calculate the average value for eliminating noise by average calculator.
Step 3: Compare the average value with set-point by set-point comparison unit.
Step 4: Generate trip signal for operation of the reactor trip by trip signal generator and provide the system information via the monitor unit simultaneously.

4.2. PN modeling for ASTS

By applying the SD concept to PN concept with communication rules as well as based on the system description, the overall PN model of the trip controller for ASTS is constructed as shown in Fig. 4. It consists of 14 places and 10 transitions with a synchronized clock, respectively. Corresponding notation of the PN model is described in Table 1.

Note that the communication rule places pc1–pc4 are one-bounded places and the arcs of pc1–pc4 are shown with hidden lines. A communication rule place is modeled as an input place of the transitions that require such a resource, and as an output place of the transitions that release this resource. For example, pc3 and pc4 depict both trip signal generator and monitor unit

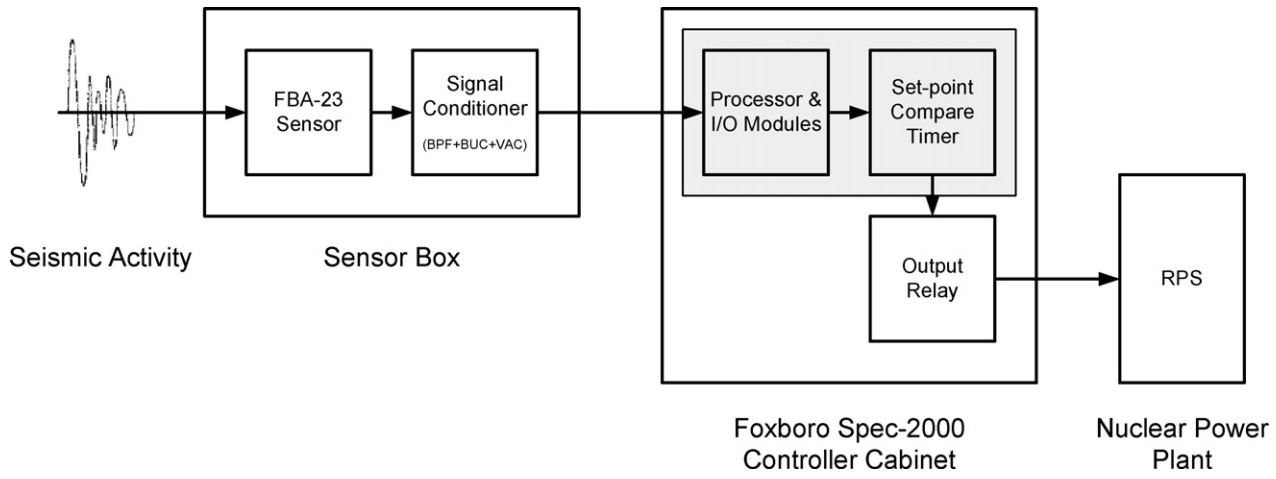

Fig. 3. ASTS system function block.

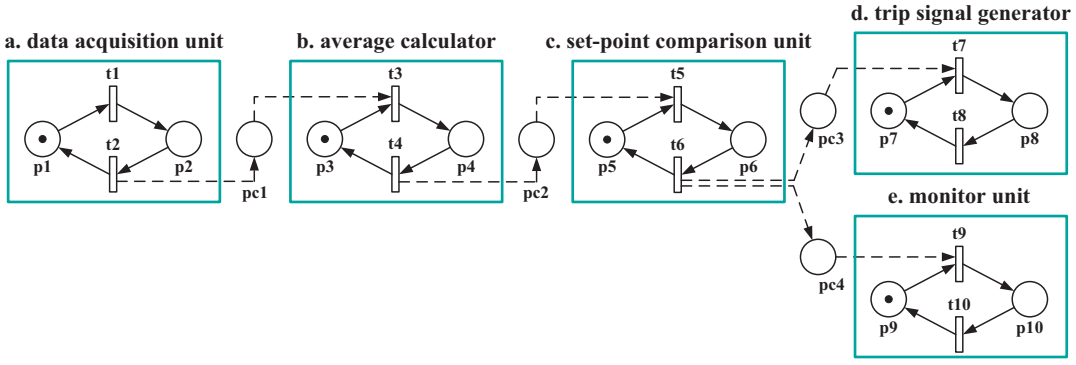

Fig. 4. Composed PN model of ASTS. (a) Data acquisition unit, (b) average calculator, (c) set-point comparison unit, (d) trip signal generator and (e) monitor unit.

being available simultaneously after the transition t6 has been fired, i.e. the end of process of set-point comparison. Furthermore, The communication rule places pc1–pc4 are used to prevent undesired conditions and regulate the concurrent process in order to avoid the signal conflicts on the part of the system.

4.3. Analysis, verification and implementation of PN model for ASTS

In order to confirm the behavioral properties of the composed PN model, the software package Platform Independent Petri Net Editor (PIPE) 2.5 is used for analysis and verification. PIPE is an open source, platform-independent tool which implemented entirely by Java for creating and analyzing generalized stochastic Petri nets (GSPN). The analysis and verification results demonstrate that the proposed PN model of ASTS is nondeadlock, bounded, safe and reversible which implies that this model can be performed correctly without deadlock and overflows, one boundedness, as well as the initial state is reachable all the time, respectively. Furthermore, the correctness of signal transmissions is guaranteed in the proposed communication rule framework for the ASTS.

After analysis and verification, the composed PN model is directly mapped into specification format by one-to-one mapping method which is considered as an explicit hierarchy included in netlist formats, or hierarchical constructs of HDL. Fig. 5 shows a part of Verilog code of local behavior and concurrent process, where the

Table 1
Notations for composed PN model of ASTS.

Place	Description	Transition	Description
p1	Initialize data acquisition unit	t1	Start data acquisition
p2	Data acquisition	t2	End data acquisition
p3	Initialize average calculator	t3	Start calculating average
p4	Calculate average value	t4	End calculating average
p5	Initialize set-point comparison unit	t5	Start set-point comparison
p6	Set-point comparison	t6	End set-point comparison
p7	Initialize trip signal generator	t7	Start monitor
p8	Generate trip signal	t8	End monitor
p9	Initialize monitor unit	t9	Start generate trip signal
p10	Monitor	t10	End generate trip signal
pc1	Notify of end of data acquisition		
pc2	Notify of end of calculating average		
pc3	Notify of end of set-point comparison for trip signal generator		
pc4	Notify of end of set-point comparison for monitor		

```
// Definition part
`define p3 CurrentState[1]
// …

module SubModule_b (clk, reset,
                    t3, t4,
                    smb_i, smb_o);
// Declarative part
input clk, reset;
// …

// Set marking
// Place 3 – initial marking
always @ (posedge clk)
    begin
        if (reset) `p3 <= 1;
        else `p3 <= (smb_i & `p3 & ~ t3)
                  | (`p4 & t4);
    end
// …

// Set outputs
always @ (CurrentState)
    begin
        if (`p3 == 1'b1) smb_o <= 1'b1;
        else smb_o <= 1'b0;
        // …
    end
endmodule
```

(a)

```
// Definition part
`define pc1 CommunicationRule[1]
// …

module MainModule (clk, reset, …);

// Declarative part
input clk, reset;
// …

// Set marking
// Communication Rule Place 1
always @ (posedge clk)
    begin
        if (reset) `pc1 <= 0;
        else `pc1 <= sma_o;
    end
// …

// Set outputs
always @ (CommunicationRule)
    begin
        if (`pc1 == 1'b1) smb_i <= 1'b1;
        else smb_i <= 1'b0;
        if (`pc2 == 1'b1) smc_i <= 1'b1;
        else smc_i <= 1'b0;
        // …
    end

// Submodule part
SubModule_a(.clk(clk), .reset(reset), ...);
SubModule_b(.clk(clk), .reset(reset), ...);
// …
endmodule
```

(b)

Fig. 5. A part of Verilog code (a) local behavior of Submodule_b and (b) concurrent process of overall system.

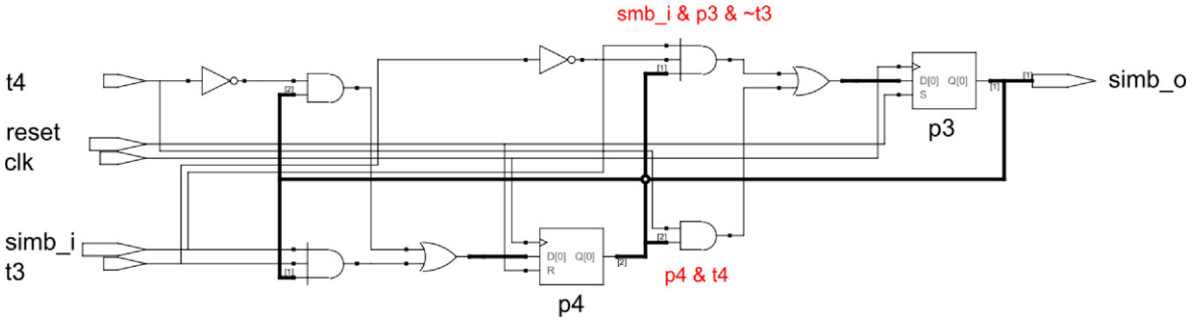

Fig. 6. A part of synthesis result of local behavior.

Submodule_b is equivalent to the local behavior of average calculator with external input smb_i and output smb_o. Similarly, the sma_i and sma_o are the external input and output of the local behavior of data acquisition, respectively. Others are known as reasoning by analogy. The synthesis result of Submodule_b shown as Fig. 6 is obtained. Obviously, the PN model of average calculator can be correctly mapped into specification format in FPGA by one-to-one mapping method. From the above design procedure, it can be evidently observed that the proposed design method is feasible and easy to implement as well as can be applied to a class of nuclear power plant I&C system.

5. Conclusion

This study deals with the problems of a FPGA-based controller design for the lack of relative experiences in the nuclear industry. Based on both SD concept and PNs modeling techniques, the concurrent process of system is explicitly described from the local behavior of each process to a global view of the interactions among processes of the system. An efficient design procedure is also proposed to achieve verification and validation (V&V) of system efforts simpler and affordable. An illustrative example of ASTS is given to demonstrate result obtained in previous sections. Synthesis results

have confirmed that the proposed scheme is feasible and easy to implement. In the future, it may have a technique that will attempt to integrate the diagnostic and safe mechanisms into a single framework to provide a completed protection scheme for nuclear I&C system.

References

Clarkson, G., 2009. Implementing the ALS at Wolf Creek. In: Proceedings of the 6th NPIC&HMIT , Knoxville, TN, USA.

Bobrek, M., et al., 2009. Review Guidelines for Field-Programmable Gate Arrays in Nuclear Power Plant Safety Systems. NUREG/CR-7006.

EPRI, 2009. Guidelines on the Use of Field Programmable Gate Arrays (FPGAs) in Nuclear Power Plant I&C Systems. EPRI-1019181.

Giua, A., DiCesare, F., 1994. Petri net structural analysis for supervisory control. IEEE Trans. Robot. Autom. 10, 185–195.

Lee, J.S., 2008. A Petri net design of command filters for semiautonomous mobile sensor networks. IEEE Trans. Ind. Electron. 55, 1835–1841.

Lee, S.J., Seong, P.H., 2004. Development of automated operating procedure system using fuzzy colored Petri nets for nuclear power plants. Ann. Nucl. Energy 31, 849–869.

Miyazaki, T., et al., 2009. Qualification of FPGA-Based Safety-Related PRM System. In: Proceedings of the 6th NPIC&HMIT , Knoxville, TN, USA.

Murata, T., 1989. Petri nets: properties, analysis and applications. Proc. IEEE 77, 541–579.

Németh, E., et al., 2009. Verification of a primary-to-secondary leaking safety procedure in a nuclear power plant using coloured Petri nets. Reliab. Eng. Syst. Saf. 94, 942–953.

Nguyen, T., 2008. EDF's projects with FPGAs. In: Proceedings of the 1st IAEA Workshop on Applications of Field-Programmable Gate Arrays in Nuclear Power Plants , Chatou, France.

Pattavina, A., Trigila, S., 1984. Combined use of finite-state machines and Petri nets for modeling communicating processes. Electron. Lett. 20, 915–916.

Platform Independent Petri net Editor (PIPE) 2.5. http://pipe2.sourceforge.net.

Wegrzyn, M., 2003. Implementation of safety critical logic controller by means of FPGA. Annu. Rev. Control. 27, 55–61.

Yastrabenetsky, M., et al., 2009. Safety assessment of FPGA-based ESFAS for Kozloduy NPP. In: Proceedings of the 6th NPIC&HMIT , Knoxville, TN, USA.

Zhou, M.C., Jeng, M.D., 1998. Modeling, analysis, simulation, scheduling, and control of semiconductor manufacturing systems: a Petri net approach. IEEE Trans. Semicond. Manuf. 11, 333–357.

Zio, E., Di Maio, F., 2009. Processing dynamic scenarios from a reliability analysis of a nuclear power plant digital instrumentation and control system. Ann. Nucl. Energy 36, 1386–1399.

Zupan, A., et al., 2008. Darlington digital control computer system replacement approach and experience. In: Proceedings of the IAEA Technical Meeting on Impact of Digital I&C on the Operation and Licensing of Nuclear Power Plants , Beijing, China.