Available online at www.sciencedirect.com

SCIENCE DIRECT®

Electric Power Systems Research 72 (2004) 49–62

ELECTRIC POWER SYSTEMS RESEARCH

www.elsevier.com/locate/epsr

A condition-based preventive maintenance arrangement for thermal power plants

S.K. Yang*

Department of Mechanical Engineering, National Chin Yi Institute of Technology, Taichung 411, Taiwan, ROC

Received 5 December 2003; received in revised form 4 March 2004; accepted 17 March 2004

Available online 7 June 2004

Abstract

In the onsite operation phase, failures are the main causes of worsened performance and degraded reliability. Consequently, an effective maintenance is the main approach to failure reduction. According to the maintenance performed before or after a failure, maintenance can be sorted as preventive maintenance (PM) and corrected maintenance (CM). Preventive maintenance is an effective approach to improving reliability. Time-based and condition-based maintenance are two major categories of preventive maintenance. In contrast, condition-based maintenance can be a better and more cost-effective type of maintenance than time-based maintenance. To improve condition-based preventive maintenance, this study uses a hybrid Petri net modeling method coupled with fault-tree analysis and parameter trend to perform early failure detection and isolation. A Petri net arrangement, namely early failure detection and isolation arrangement (EFDIA), is employed that facilitates alarm, early failure detection, fault isolation, event count, system state description, and automatic shutdown or regulation. These functions are very useful for health monitoring and preventive maintenance of a system. Besides, the Petri net with these capabilities is not only done on paper but also actualized on an FPGA as an application-specific integrated circuit (ASIC) so that the proposed scheme is practicable. A thermal power plant is adopted as an example to demonstrate the method.

Keywords: Preventive maintenance; Failure prediction; Petri nets; Thermal power plant; ASIC

1. Introduction

In the onsite operation phase, failures are the main causes of worsened performance and degraded reliability. Accordingly, failure avoidance is the main approach to reliability assurance. To achieve failure reduction, an effective maintenance is the best way [1]. There are three main types of maintenance: improvement maintenance (IM), preventive maintenance (PM), and corrective maintenance (CM) [2]. The purpose of IM is to reduce or eliminate entirely the need for maintenance, i.e. IM is performed at the design phase of a system emphasizing elimination of failures. However, there are many restrictions for a designer, such as space, budget, market requirements, etc. Usually the reliability of a product is related to its price. On the other hand, CM is the repair performed after failure occurs. PM means all actions intended to keep equipment in good operating condition and to avoid failures [2]. PM should be able to indicate when a failure is about to occur, so that repair can be performed before such failure causes damage or capital investment loss. Hence, PM is an effective approach to promoting reliability [3]. Time-based and condition-based maintenance are two major approaches for PM. In contrast, condition-based maintenance can be a better and more cost-effective type of maintenance than time-based maintenance [4]. Irrespective of the approach adopted for PM, the key point is whether a failure can be detected early or even predicted. If the predicted parameters indicate a device is going to fail, then the failure can be prevented in time by PM. Nevertheless, the parameters should be accurately predicted at a reasonably long time ahead of failure occurrence [5,6]. Many methods have been proposed for failure prediction such as statistic skills [7,8], neural network [9], understanding the failure mechanism of damaged product [10], etc.

* Tel.: +886 423924505x7178; fax: +886 423930681.
E-mail address: skyang@chinyi.ncit.edu.tw (S.K. Yang).

0378-7796/$ – see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.epsr.2004.03.007

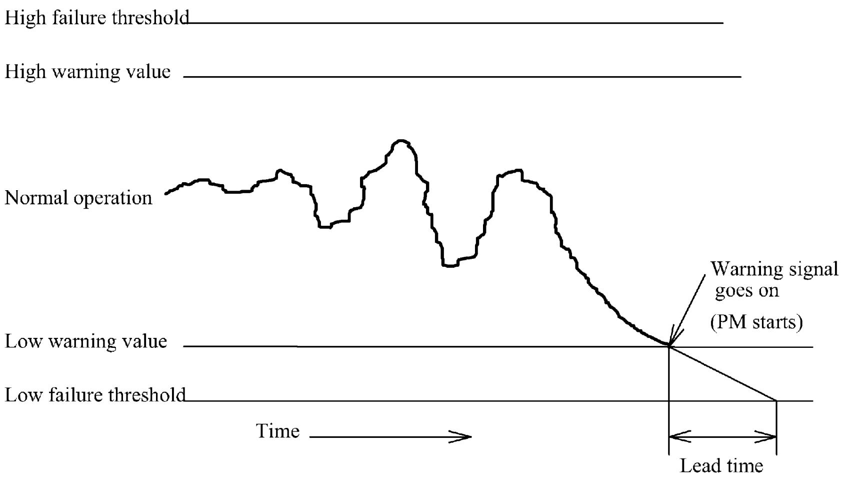


Fig. 1. A control chart.

Time-based maintenance is commonly adopted by the power plants in Taiwan recently. A scheduled maintenance is enacted based on a statistical average that is suggested by the equipment vendor or decided by the field-engineers. Therefore, time-based maintenance still retains the unavoidable risk that the system may fail before criteria are exceeded, i.e. a failure may occur unexpectedly. On the other hand, the actual duty-cycles for a certain part or module may be longer than those averages, so if they are replaced during scheduled maintenance, that is a waste of the investment. The condition-based scheme avoids those drawbacks. This study aims to promote the maintenance strategy for a thermal power plant from timed-based to condition-based.

Probabilistic risk assessment (PRA) is one of the effective methods for hazard reduction and maintenance strategy planning [11,12]. PRA is widely used in different areas concerning system safety, such as power plants [13–15], space shuttle [16], etc. Many extensions of the classic fault tree, for example: a probabilistic fault tree [17] and a dynamic fault tree [18], are employed as tools [19] to perform PRA. Expert system for PRA is developed [20] as well. In this study, a hybrid Petri net modeling method coupled with fault-tree analysis and parameter trend are used to perform early failure detection and isolation. First of all, a Petri net dealing with system failure, namely PNSF, has to be established, which can either be transformed from a system fault-tree or be constructed directly [4]. Each event in the PNSF is continuously monitored by an adequate sensor. Actual values of the event are acquired by the monitor sensors. Each event has a prescribed warning value, and the sensor-acquired value is compared with the prescribed warning value to judge whether the monitored event to be failed or not. Once the sensor-acquired value reaches the warning value, the failure is predicted. Accordingly, the current state is a warning state and the PM should be executed now.

Nowadays, ICs are becoming not only smaller and more powerful but also faster and cheaper. As a result, application-specific integrated circuits (ASICs) are widely used. In practice, Petri nets can be implemented as ASICs, so as to perform specific functions without user intervention. The Petri net used to perform early failure detection and isolation in this study is converted to logic circuit and actualized on an ASIC via an Field-Programmable Gate Array (FPGA).

2. Control chart and threshold

A failure threshold is a value used to judge an equipment failure occurs or not. It is prescribed as the measurement

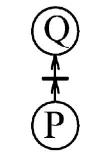
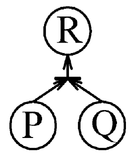
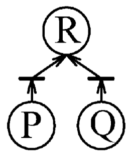
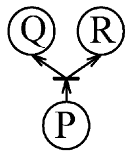
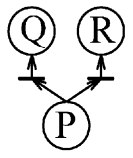
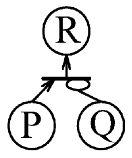
Logic relation	TRANSFER	AND	OR	TRANSFER AND	TRANSFER OR	INHIBITION
Description	If P then Q	If P AND Q then R	If P OR Q then R	If P then Q AND R	If P then Q OR R	If P AND Q' then R
Boolean function	Q=P	R=P*Q	R=P+Q	Q=R=P	Q+R=P	R=P*Q'
Petri nets						

Fig. 2. Basic structures of logic relations for Petri nets.

value that is taken just prior to or at the time of failure. Life testing is one method to obtain such data, and may be performed by field-engineers or users. Normally, the mean value of a failure-probability function that is established from tests of manufacturers is a theoretic value for the threshold.

Once the threshold has been determined, a margin of safety should be added to account for variations in early failure detection. The safety margin can be determined by the requirement of lead-time for PM or evaluation of the physical properties and actual operating conditions of different systems. The lower the warning value is set, the greater is the assurance that PM will be done prior to failure [2], whereas more labor manpower and cost will be expended. Theoretically, triple the standard deviation is one possible choice in prescribing a warning value [3]. On the basis of failure thresholds and warning values, a control chart can be constructed to conduct limit control, as illustrated in Fig. 1. The lead-time of early detection can be obtained by extrapolating the curve in a control chart with a line slope that is constructed by the last two sampled points on the curve [21]. The lead-time is the period between the time point where the warning value is exceeded and the intersection of the extended line and the time-axis. The lead-time obtained from the control chart is for the action of the PM for the monitored channel.

Failure detection can be carried out by comparing actual with nominal quantities, and fault isolation by comparing actual with fault quantities [22]. Consequently, an instrumentation system should be set up for PM, to acquire actual quantities at measurement points. In addition to being used for comparison, acquired quantities can be stored to establish a database for modifying predetermined failure thresholds and warning values. The performance of some systems depends on external conditions. For example, the output current of a power generator varies with the load, which changes with time during the day. Hence, thresholds and warning values may be varied according to a scheduled scheme that accomplishes adaptive adjustment for those values. Referring to Fig. 1, the situation is called ’error’ in this paper whenever the acquired quantity exceeds the prescribed low (high) warning value but falls within the low (high) warning value and low (high) failure threshold. An error is sometimes referred to as an incipient failure [23]. Therefore, PM action is taken when the system is still at an error condition, i.e. within acceptable deviation and before failure occurs. Thus, through the technique of PM, failure can be early processed so that the reliability is improved.

3. Petri nets and EFDIA

3.1. Petri nets

A Petri net is a general-purpose graphical tool for describing relations existing between conditions and events [24]. The basic symbols of Petri nets include [25]:

○	Place, drawn as a circle, denotes event
—	Immediate transition, drawn as a thin bar, denotes event transfer with no delay time
▬	Timed transition, drawn as a thick bar, denotes event transfer with a period of delay time
↑	Arc, drawn as an arrow, between places and transitions
●	Token, drawn as a dot, contained in places, denotes the data
⫯	Inhibitor arc, drawn as a line with a circle end, between places and transitions

Places contain dots, the representation of tokens, being the specific marking of a Petri net [26].

The transition is said to fire, if input places satisfy an enabling condition. Transition firing will remove one token from all of its input places and put one token into all of its output places [27].

Basic structures of logic relations for Petri nets are listed in Fig. 2, where there are two types of input places for the transition; namely, specified and conditional [4]. The former has a single output arc whereas the latter has multiples. Tokens in a specified-type place have only one outgoing destination, i.e. if the input place(s) holds a token then the transition fires and gives the output place(s) a token. However, tokens in the conditional-type place have more than one outgoing paths, which may lead the system to different situations. For the ‘TRANSFER or’ Petri net in Fig. 2, whether Q or R takes over a token from P depends on which output-transition of P is fired earlier.

There are three types of transitions that are classified based on time [24]. Transitions with no time delay are called immediate transitions, while those that need a certain constant period of time for transition are called timed transitions. The third type is called a stochastic transition and is used for modeling a process with random time [28]. Hence the Petri net is a powerful tool for modeling various systems.

3.2. The EFDIA

An early failure detection and isolation arrangement (EFDIA) [4] is employed in this paper. It is a hybrid Petri net that includes three kinds of Petri sub-nets: ordinary, inhibitor-arc type, and timed. In an PNSF, for PM optimization, each place with a monitor sensor will be equipped with an EFDIA that facilitates alarm, early failure detection, fault isolation, event count, system state description, and automatic shutdown or regulation. In this context, a cause-consequence type of PNSF is drawn in fault-tree style with basic events at the bottom and the final undesirable event at the top. EFDIA is shown in Fig. 3, and all the symbols are defined as follows:

1. n: total number of sensing points in a PNSF.
2. i: sequence number, $1 \leqq i \leqq n$.

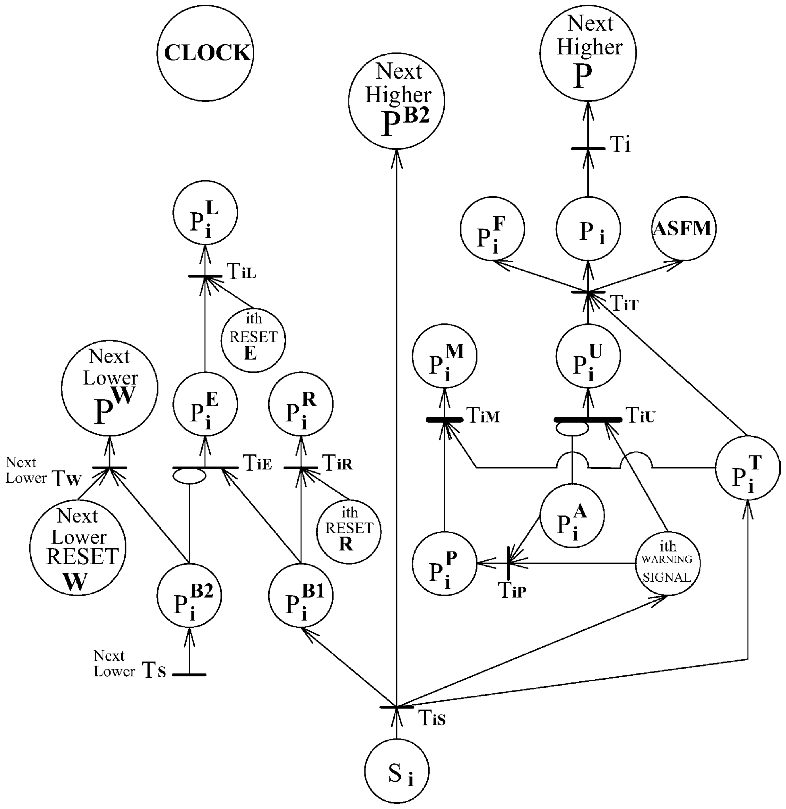


Fig. 3. Early failure detection and isolation arrangement (EFDIA).

3. $M(P)_k$: marking of place P at time kT, representing the token quantity of place P at time kT, $k = 1, 2, 3, \ldots$
4. P_i: ith place of PNSF, $M(P_i) = 1$ if the failure represented by P_i occurs.
5. T_i: ith transition of PNSF, representing the time duration.
6. S_i: sensing signal place of P_i, S_i generates a token such that $M(S_i) = 1$ if the signal of S_i exceeds the warning value, i.e. an abnormal situation (error) occurs.
7. $T_{i\mathrm{E}}$: error transition of P_i; an immediate transition.
8. $T_{i\mathrm{L}}$: error times log transition of P_i; an immediate transition.
9. $T_{i\mathrm{M}}$: maintained transition, representing the transitional time from when the PM action for P_i is taken to when P_i is maintained, a timed transition.
10. $T_{i\mathrm{P}}$: processing transition of P_i; an immediate transition.
11. $T_{i\mathrm{R}}$: reset transition of P_i; an immediate transition.
12. $T_{i\mathrm{S}}$: sensing transition of P_i; an immediate transition.
13. $T_{i\mathrm{T}}$: transfer transition of P_i; an immediate transition.
14. $T_{i\mathrm{U}}$: unprocessed transition of P_i, representing the transitional time from when the ith warning signal appears to when P_i failure occurs, a timed transition.
15. Next Lower T_{W}: warning times log transition of the corresponding next lower level P^{W}; an immediate transition; the number of the Next Lower T_{W} should equal the number of inhibitor arcs of transition $T_{i\mathrm{E}}$.
16. P_i^{A}: PM action taken place of P_i; $P_{\mathrm{i}}^{\mathrm{A}}$ generates a token such that $M(P_{\mathrm{i}}^{\mathrm{A}}) = 1$ if the PM action for P, is taken.
17. $P_i^{\mathrm{B}j}$ jth buffer place of P_i for tokens to stay temporarily, $j = 1$ to x; x is the number of input arcs for P_i. $P_i^{\mathrm{B}j}$ is unnecessary when P_i is a basic place in a PNSF. A basic place is a place that there is no place lower than it in a Petri net.
18. P_i^{E}: error indication place of P_i; $M(P_i^{\mathrm{E}}) = 1$ after $T_{i\mathrm{E}}$ fires if the $M(S_i) = 1$ situation is generated by P_i itself but not aroused by lower-level places (for fault isolation).
19. P_i^{F}: failure counter place of P_i; $M(P_i^{\mathrm{F}})$ represents failure times log number of P_i; $M(P_i^{\mathrm{F}})$ increases by one when P_i failure occurs.
20. P_i^{L}: error counter place of P_i; $M(P_i^{\mathrm{L}})$ represents the error times log number of P_i; $M(P_i^{\mathrm{L}})$ increases by one when P_i error occurs.
21. P_i^{M}: maintenance counter place of P_i; $M(P_i^{\mathrm{M}})$ represents maintenance times log number of P_i; $M(P_i^{\mathrm{M}})$ increases by one when the $M(S_i) = 1$ situation is maintained.
22. P_i^{P}: processing place of P_i, representing P_i in being maintained situation.
23. P_i^{R}: reset counter place of P_i; $M(P_i^{\mathrm{R}})$ represents the warning times log number of P_i that are aroused by lower-level places, i.e. the reset times of the ith RESET R.

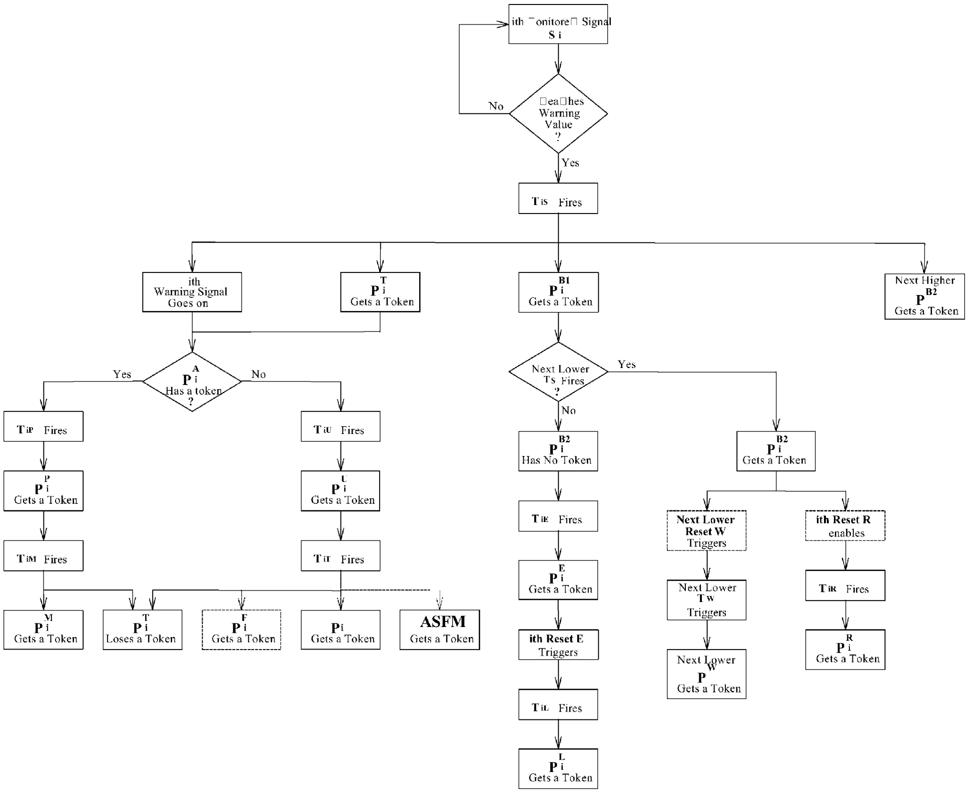


Fig. 4. Flowchart of the EFDIA.

24. P_i^{T}: transitional place of P_i, representing a transitional state inserted between S_i and P_i, which is the original path from S_i to P; without EFDIA constructed.
25. P_i^{U}: unprocessed place of P_i, representing the error of P_i not corrected.
26. Next Lower P^{W}: warning counter place of the next lower P; M(Next Lower P^{W}) represents the warning times log number of the next lower P no matter from where warning cause arises. The number of the Next Lower P^{W} should equal number of inhibitor arcs of transition $T_{i\mathrm{E}}$.
27. ith RESET E: reset E place of P_i, representing a reset signal for P_i^{E}, generates a token once it is triggered.
28. ith RESET R: reset R place of P_i, representing a reset signal for P_i^{B1}, generates a token when it is triggered.
29. Next Lower RESET W: reset W place of the corresponding Next Lower level P^{W}; representing a reset signal for P_i^{Bj} (j is the sequence number of the next lower place); generates a token once it is triggered; the number of the Next Lower RESET W should equal number of inhibitor arcs of transition $T_{i\mathrm{E}}$.
30. ASFM: automatic shutdown or feedback mechanism place; for instance, an air-conditioning or ventilation system is a feedback mechanism for an over temperature module.
31. ith WARNING SIGNAL: warning signal place for P_i.
32. Clock: a clock is embedded to indicate and record the time of the occurrence for each event.

To explain EFDIA more clearly, based on the aforementioned definition for each symbol, the operation of EFDIA is depicted step by step as follows.

1. As defined in the previous paragraph, $M(P)$ is the marking of place P. Thus, $M(P_i^{\mathrm{T}}) = 1$ represents the S_i monitored subsystem (module) at a transitional state. Transition $T_{i\mathrm{S}}$ fires if $M(S_i) = 1$. Subsequently, each of P_i^{B1}, ith WARNING SIGNAL, P_i^{T}, and next higher P^{B2} obtains a token. Similarly, M(ith WARNING SIGNAL) $= 1$ represents that the ith warning signal goes on, which may be a light indication, a beep or some other form, to remind the user that the value of the monitored signal has reached the prescribed warning value.
2. There are two paths to follow:
   (1) $T_{i\mathrm{P}}$ fires if P_i^{A} generates a token, i.e. PM action is taken. The tokens in P_i^{A} and the ith WARNING SIGNAL move to P_i^{P}, i.e. the subsystem (module) is being maintained. Otherwise, $T_{i\mathrm{U}}$ fires if P_i^{A} does not generate a token during the transition time of $T_{i\mathrm{U}}$ such that P_i^{U} acquires a token.
   (2) $T_{i\mathrm{E}}$ fires if P_i^{B2} has no token, i.e. this error is not caused by the next lower subsystem (module) but by the ith-level subsystem (module) itself, such that P_i^{E} obtains a token. On the other hand, if P_i^{B2} holds a token, i.e. this error results from the next lower subsystem (module), then $T_{i\mathrm{E}}$ does not fire, such that

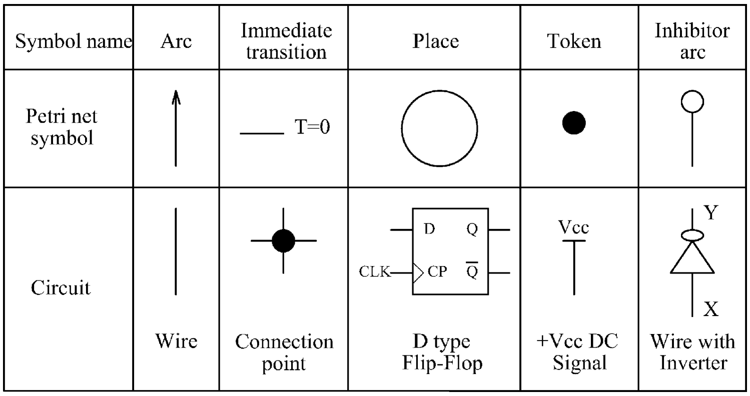


Fig. 5. Counter circuit converted form Petri net.

the token from S_i will be held in P_i^{B1}. The error is hence isolated.

3. There are again two paths to follow:
   (1) $T_{i\mathrm{M}}$ fires if the PM action is finished, such that the token in P_i^{P} together with the token in P_i^{T} move to P_i^{M}, i.e. this error has been corrected. Otherwise, $T_{i\mathrm{T}}$ fires if P_i^{U} obtains a token resulting from the firing of $T_{i\mathrm{U}}$, i.e. PM action was not taken in time, such that tokens in P_i^{U} and P_i^{T}, move to P_i, i.e. a failure indicated by the marking of P_i occurs. As a consequence, both P_i^{F} and ASFM also obtain a token. Accordingly, the failure times log number increases by one and the ASFM is triggered. The ASFM can be optional for different systems.
   (2) $T_{i\mathrm{L}}$ fires if P_i^{E} holds a token and the ith reset E is triggered, such that P_i^{L} obtains a token, i.e. the error times log number of P_i increases by one. Otherwise, the token in buffer place P_i^{B1} will move to P_i^{R} when $T_{i\mathrm{R}}$ fires by triggering the ith RESET R, i.e. this error is not caused by P_i and the reset times log number of P_i increases by one. Similarly, the Next Lower RESET W triggers to fire Next Lower T^{W} such that the token in the other buffer place, P_i^{B2}, moves to the Next Lower P^{W}, i.e. the warning times log number of the next lower P increases by one.

Conventionally, a flowchart is an easy visual representation for understanding the operational steps. Therefore, the above descriptions are summarized into a flowchart for clarity, as shown in Fig. 4.

3.3. Capabilities of EFDIA

1. Alarm: EFDIA provides alarm capability whenever an over-warning-value situation occurs, by triggering the ith WARNING SIGNAL for the associated place.
2. Early failure detection: EFDIA is capable of early failure detection, since the alarm function operates whenever the sensor-acquired value reaches the corresponding prescribed warning value. This means that the abnormal situation is detected before failure occurs.

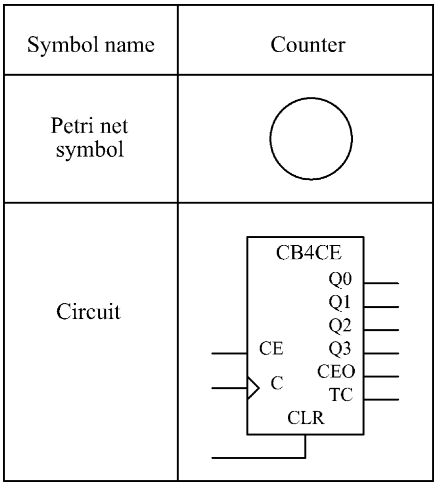


Fig. 6. Corresponding circuits for basic Petri net symbols.

3. Fault isolation: The cause(s) of malfunction of a system can be located anywhere within the system. However, since malfunction causes are constrained by the logic relations of the PNSF, they can be isolated by the inhibit

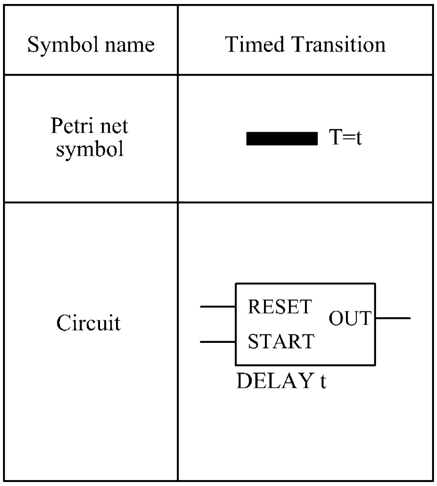


Fig. 7. Timer circuit converted form Petri net.

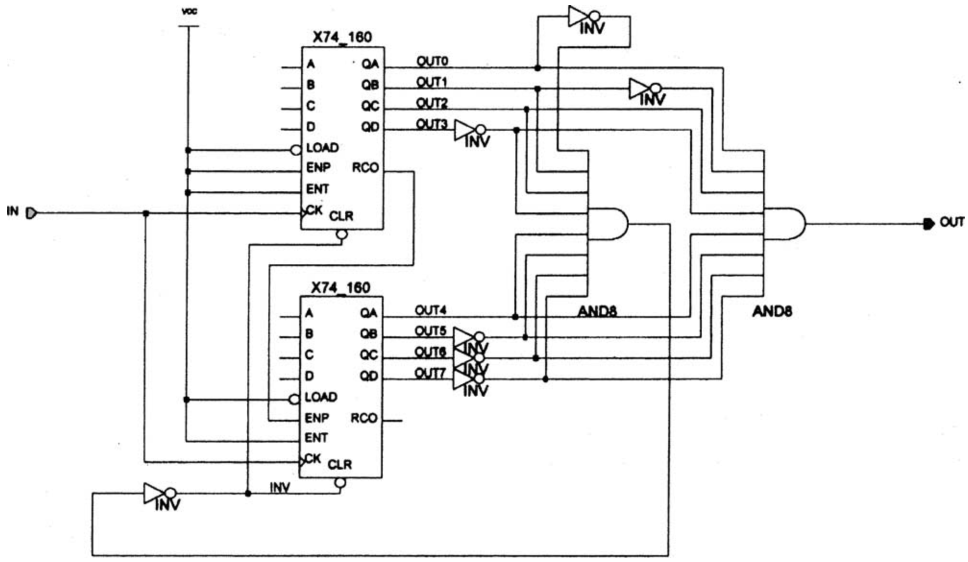


Fig. 8. Circuit of FRE1DIV15.

transition $T_{i\mathrm{E}}$ via the indication of the event flag P_i^{E}. The error is located at the ith place if $M(P_i^{\mathrm{E}}) = 1$. Otherwise, the error of the ith place arises from the lower-level place(s) even if the ith warning signal appears.

4. Event count: All the counters designated in EFDIA record the associated occurrence multiplicities of events. By incorporating a time clock, the associated rates can be obtained at the same time. The following items can be derived from EFDIA:
   (1) Failure rate of the ith place: $M(P_i^{\mathrm{F}})/t$.
   (2) Error rate of the ith place: $M(P_i^{\mathrm{L}})/t$.
   (3) Maintenance rate of the ith place: $M(P_i^{\mathrm{M}})/t$.
   (4) Alarm rate of the ith place: $M(P_i^{\mathrm{W}})/t$.

   From these rates, two advantages can be obtained:

   (1) If the ith subsystem is maintained whenever a failure is predicted, the failure rate of the ith place can be minimized such that the system reliability is improved.
   (2) All the rates can be recorded as historical data so as to perform statistical prediction of system failure (by failure rate and error rate), and the time needed for maintenance (by maintenance rate) of each subsystem can be derived.
5. System state description: The system state is clearly visible by the indication of every place in EFDIA. The following parameters are defined to account for system state:
   (1) $\mathbf{M}_k$: marking of the PNSF at time kT, $\mathbf{M}_k = [M(P_1), M(P_2), \ldots, M(P_n)]^{\mathrm{T}}$.
   (2) $\mathbf{S}_k$: predicted sensing signal matrix at time kT, $\mathbf{S}_k = [M(S_1), M(S_2), \ldots, M(S_n)]^{\mathrm{T}}$.
   (3) $\mathbf{L}_k$: maintenance log matrix at time kT, $\mathbf{L}_k = [M(P_1^{\mathrm{M}}), M(P_2^{\mathrm{M}}), \ldots, M(P_n^{\mathrm{M}})^{\mathrm{T}}$.

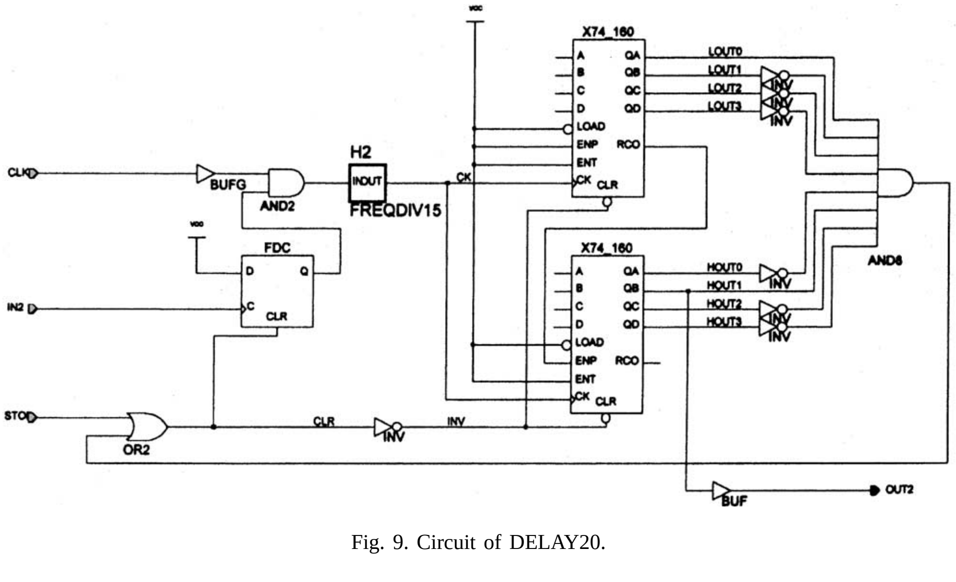


Fig. 9. Circuit of DELAY20.

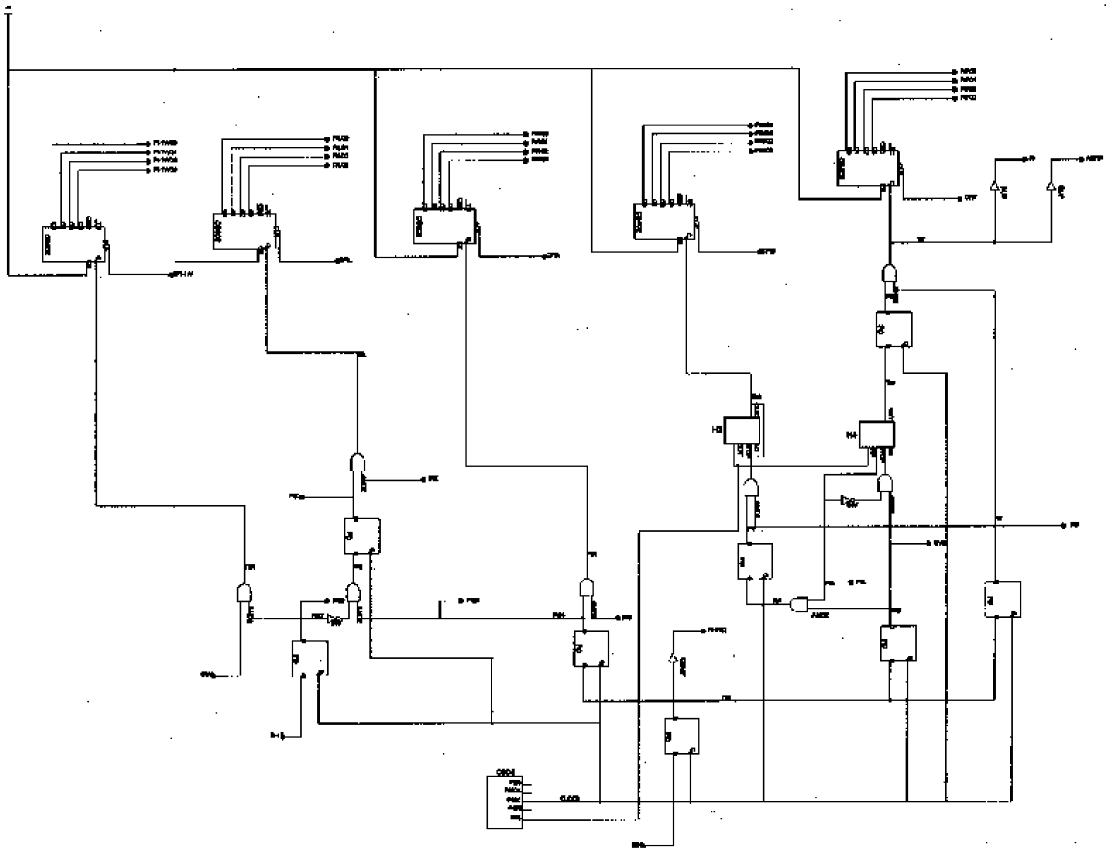


Fig. 10. Circuit of EFIDIA.

(4) $\mathbf{F}_k$: failure number log matrix at time kT, $\mathbf{F}_k = [M(P_1^F), M(P_2^F), \ldots, M(P_n^F)]^T$.

(5) $\mathbf{E}_k$: error indication matrix at time kT, $\mathbf{E}_k = [M(P_1^E), M(P_2^E), \ldots, M(P_n^E)]^T$; the *i*th entry indicates that the error is located at the *i*th place if the *i*th entry value is unity.

6. Auto shutdown or regulation: Automatic shutdown or regulation capability can be provided by EFDIA through triggering the ASFM place. ASFM is intended to prevent a higher-level fault or system breakdown by automatic shutdown or regulation. It should be incorporated with the places that may cause a safety problem in an PNSF.
7. Time recording: The time at which each event occurs can be indicated and recorded by the embedded CLOCK. This is required for failure analysis.

4. Implementation of EFDIA

System can be modeled into Petri net to express not only static behaviors such as logical relations between components of the system, but also dynamic behaviors such as operating sequence or failure occurrence of the system. Because Petri nets are state machines [29], it is feasible to realize Petri nets to perform those capabilities. Hardware implementation of Petri nets actualizes state machines that are converted from Petri nets to logic circuits. Mainly because of the programmable capability, FPGAs are suitable for hardware implementation of Petri nets. This study employs a Xilinx FPGA [30] as the design tool to implement Petri nets.

4.1. Petri net symbols

By using Xilinx Foundation [30], each of the Petri net-converted circuits can be generated as a macro symbol for the schematic toolbox. Detail circuits of corresponding macro symbols can be observed by hierarchy push-and-pop functions. The corresponding circuits for the five basic symbols of Petri nets are listed in Fig. 5.

1. *Place*, a place can be converted to a D-type flip–flop, which represents the associated event occurrence by output high Q. Q is high if D is high at the rising edge of the clock pulses.
2. *Token*, a token can be represented as a logic high signal.
3. *Arc*, arcs are connection wires between components.
4. *Immediate transition*, a connection point represents it.
5. *Inhibitor arc*, an inhibitor arc can be converted to a connection wire with an inverter. It inverts the relation between input (X) and output (Y).

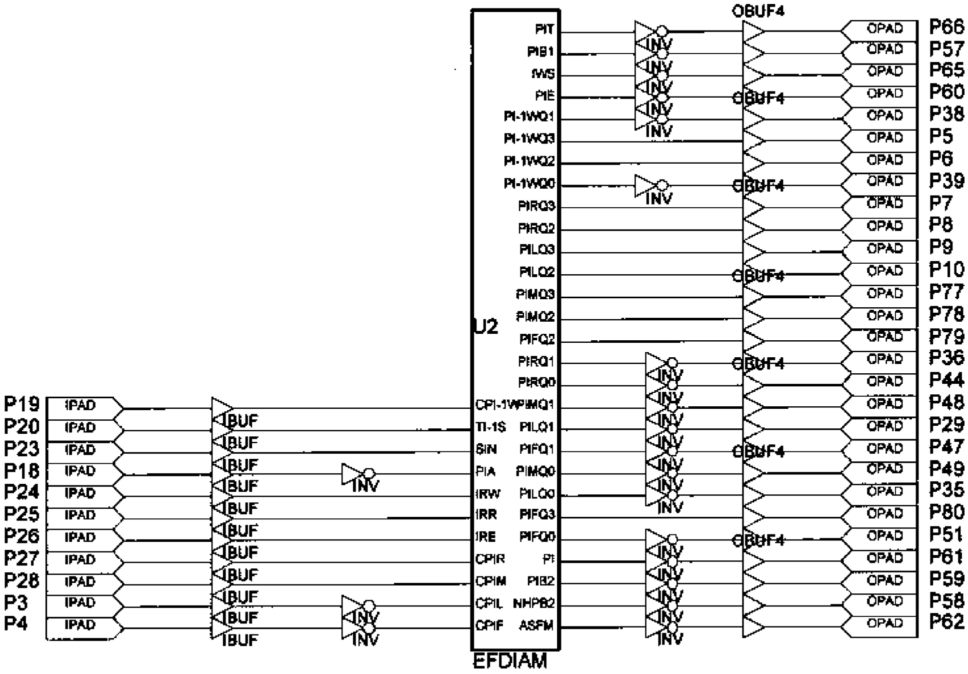


Fig. 11. EFIDIA macro symbol.

4.2. Specific function arrangements

1. *Reset*: This function is used to release the token that is hold in a place by generating a token to fire the output transition of the place. Since a token is implemented by a logic high signal, the reset function can be implemented as a push button with a Vcc input.
2. *Counter*: It is used to count and record event occurrence times. There are various types of counters in Xilinx XACT libraries [31]. In a Petri net dealing with system failures, the counter should count up to a sufficient number and be able to be cleared asynchronously. Therefore, a 4-bit cascade binary counter with clock enable and asynchronous clear (CB4CE) is adopted in this study. The CB4CE is shown in Fig. 6 and pin functions are described as follows:

   (1) CE is the clock enable input, which is used to enable the counter itself.
   (2) C stands for the clock.
   (3) Q0, Ql, Q2, and Q3 constitute four data output bits. They increment when the CE is high during the low-to-high clock transition.
   (4) CEO is the counter-enable output, which is used to enable the next stage counter.
   (5) TC denotes terminal count. It is high when all Qs are high.
   (6) CLR is the asynchronous clear. When CLR is high, all other outputs are ignored and all Qs and TC outputs go to logic level zero, independent of clock transition.

3. *Timed transition*: It denotes event transfer with delay time *t*. As shown in Fig. 7, it is implemented by a timer with delay time *t* and start–reset functions. The timer output becomes high at t time later than the arrival of a logic high signal at the timer input. To achieve this function, a two-level hierarchy configuration circuit is used. The lower level is a frequency divider, namely FREQDIV15 in this study, dividing the input clock frequency by 15. The FREQDIV15 circuit is shown in Fig. 8, where the X74_160 is a 4-bit BCD counter [31]. The FREQDIV15 sends a clock pulse out for every 15 input clock pulses and clears X74_160 at the positive edge of the 16th clock pulse. The FREQDIV15 is generated to a macro symbol, as shown in Fig. 9, for the design toolbox of this project file. The upper-level circuit of the timer configuration is shown in Fig. 9. There is an existing oscillator in Xilinx XACT library, namely OSC4, which supplies five different frequencies of clock, i.e. 15 Hz, 490 Hz, 16 kHz, 500 kHz, and 8 MHz. The FREQDIV15 outputs a 1 Hz clock by feeding the 0SC4 15 Hz clock into the FREQDIV15. The 1 Hz clock is used as a base time to generate the *N*-sec time delay for timed transition merely follows the FREQDIV15 a MOD-N frequency divider. A MOD-20 frequency divider circuit, for example, follows the FREQDIV15 in Fig. 9. The D flip–flop and the AND gate construct a switch to start and stop counting delay time of timed transitions by the IN2 trigger signaland the STOP signal, respectively. The STOP signal also resets the timer. Using the technique similar to the DELAY20 and the five clock frequencies provided by OSC4, a variety of delay times can be implemented.

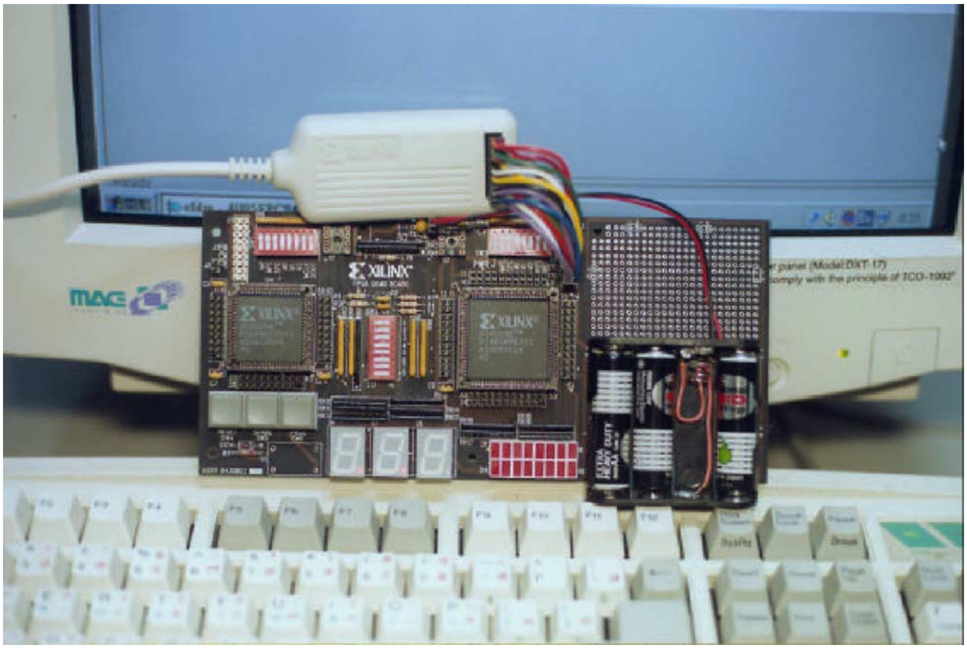


Fig. 12. I/O assignment on Demonstration Board for EFDIA implementation.

4.3. Circuit of EFDIA

Using circuits described in Sections 4.1 and 4.2, the logic circuit of the EFDIA is constructed as shown in Fig. 10. Each of H3 and H4 in Fig. 10 is a timer composed of DELAY20. The EFDIA circuit can be integrated into a 39-pins ASIC. Fig. 11 shows the macro symbol for EFDIA. Hence, the EFDIA Petri net is realized to become an ASIC as long as downloading this EFDIA macro to a Xilinx FPGA board [32], The correspondence between EFDIA pin names (Fig. 11) and EFDIA Petri net symbol names (Fig. 3) are listed below:

1. Input pins:
   (1) CPI-1W: clear signal, which is implicit in Fig. 3, for Next Lower P^{W} counter
   (2) TI-1S: Next Lower T_{S}
   (3) SIN: S_i
   (4) PIA: P_i^{A}
   (5) IRW: Next Lower Reset W
   (6) IRR: *i*th Reset R
   (7) IRE: *i*th Reset E
   (8) CPIR: clear signal, which is implicit in Fig. 3, for P_i^{R} counter
   (9) CPIM: clear signal, which is implicit in Fig. 3, for P_i^{M} counter
   (10) CPIL: clear signal, which is implicit in Fig. 3, for P_i^{L} counter
   (11) CPIF: clear signal, which is implicit in Fig. 3, for P_i^{F} counter
2. Output pins:
   (1) PIT: P_i^{T}
   (2) PIB1: P_i^{B1}
   (3) IWS: *i*th WARNING SIGNAL
   (4) PIE: P_i^{E}
   (5) PI-1WQ0–PI-1WQ3: Next Lower P^{W} counter
   (6) PIRQ0–PIRQ3: P_i^{R} counter
   (7) PILQ0–PILQ3: P_i^{L} counter
   (8) PIFQ0–PIFQ3: P_i^{F} counter
   (9) PIMQ0–PIMQ3: P_i^{M} counter
   (10) PI: P_i
   (11) PIB2: P_i^{B2}
   (12) NHPB2: Next Higher P^{B2}
   (13) ASFM: ASFM

4.4. Implementation

The EFDIA logic circuit is implemented by downloading its schematic diagram to a Xilinx FPGA Demonstration Board. The board is a stand-alone board for experimenting and developing prototypes using the Xilinx FPGA architecture. Two FPGA devices, namely XC3020A and XC4003E have been installed on the board. The XC4003E has higher density and more input/output blocks and flip–flops than the XC3020A [33]. Hence, the XC4003E is adopted in this study to implement EFDIA. The configuration of the XC4003E for implementing the EFDIA is described as follows:

1. *Power supply*: The power for the Demonstration Board is supplied by a battery set, which has 3 AA(UM-3) batteries in series to supply +5 V through the connector J9 of the board.
2. *Downloading interface*: The EFDIA schematic diagram for configuring the XC4003E is downloaded from a personal computer through an Xchecker cable [32] which

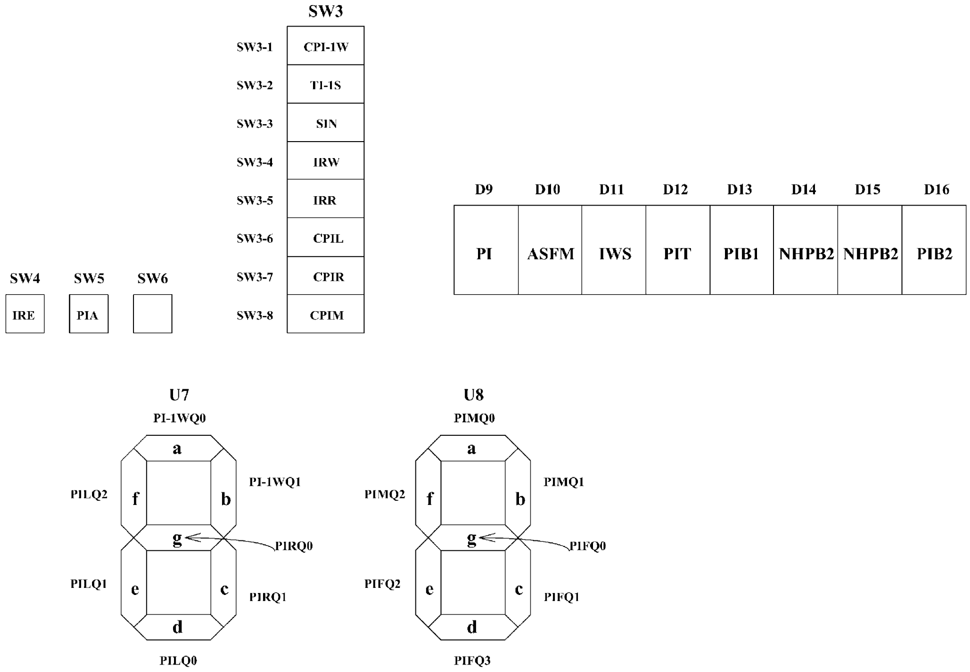


Fig. 13. The downloaded Demonstration Board.

connects either the COM1 or COM2 port of the computer to the J2 connector of the board.

3. *Input terminals*: Switches SW3, SW4 and SW5 provide input signals for the XC4003E to implement the EFDIA circuits. The SW3 is a switch set with eight switches connecting to eight general-purpose inputs on XC4003E input pins. An XC4003E input pin is set to logic 1 when the corresponding switch is on, and logic 0 when the corresponding switch is off. The SW4, namely Reset Pushbutton, can apply an active-Low reset signal to the XC4003E via pin 56 when the SW2-7 switch is on. As for the SW5, namely Spare Pushbutton, applies also an active-Low signal to the XC4003Eviapinl8.
4. *Output terminals*: Three seven-segment displays are included with the U6 connect to the XC3020A, and U7 and U8 connect to the XC4003E. Each LED segment is turned on by driving the corresponding FPGA pin Low with logic 0. Decimal points serve as state and error indicators. Besides, there are eight LEDs connected to the I/O pins in each FPGA. LEDs Dl through D8 connected to the XC3020A, while D9 through D16 connect to the XC4003E. Each LED is also turned on by driving its corresponding FPGA pin Low with logic 0. There are extra 16 I/O lines that connect each FPGA.

Fig. 12 shows the picture of the downloaded Demonstration Board, and the I/O assignment on the Demonstration Board for the EFDIA implementation is shown in Fig. 13.

5. Examples

A thermal power plant is employed as an example for PM by using the method introduced in this study. The system block diagram for the thermal power plant is shown in Fig. 14. In order to construct the PNSF, eight sensors are selected to be installed at the associated test points to acquire data. Sensor types, locations, and associated sensing signals are depicted in Fig. 14. Fig. 15 depicts the resultant PNSF of this system. However, the PNSF only describes the cause-consequence relations among events that are shown in the PNSF. For example, P_7 causes P_8 but P_8 may be caused by other than P_7 such as excitation system failure, short circuit, loading conditions, etc. that are not included in the PNSF. A complete failure-cause list is the prerequisite of a complete PNSF. Nevertheless, the situation of P_8 caused by other than P_7 can be identified by the capability of EFDIA that was depicted in Section 3.3.

The PNSF for the thermal power plant endowed with EFDIA is shown in Fig. 16. It is a result of appending an EFDIA to each place with a monitor sensor in Fig. 15, i.e. P_1–P_8 The logic relations among all places in Fig. 15 are still retained in Fig. 16 (the shadowed portion). At basic places of the PNSF, i.e. P_1, P_4, and P_5, the function for testing whether the error cause is from the next lower place or not becomes unnecessary. The following two situations are used to demonstrate the function of EFDIA in this system:

1. Suppose the value of the monitored signal for fuel flow, i.e. S_1 in Fig. 16, reaches the prescribed warning value. Subsequently, T_{1S} fires such that the 1st WARNING SIGNAL is produced and each of P_1^{T}, P_2^{B2} and P_1^{E} obtains a token. $M(P_1^{\mathrm{T}}) = 1$ represents the fuel flow rate that is at an error situation, and is a transitional state between normal and faulty. There is a lead-time from then until P_1 failure really happens. If the PM action takes

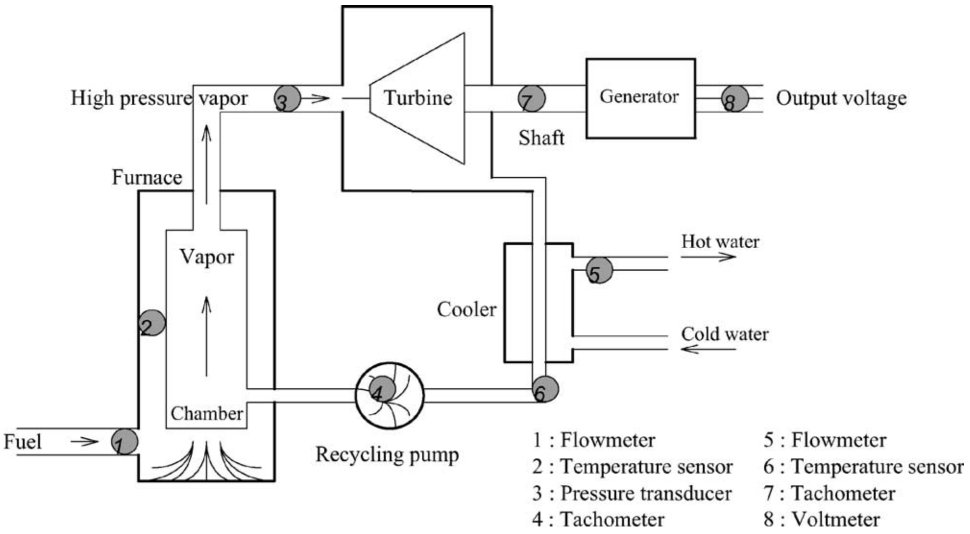


Fig. 14. System block diagram of a thermal power plant.

place during the lead-time, then P_1^{A} generates a token such that $T_{i\mathrm{P}}$ fires so as to make the token in P_1^{A} together with the token in the 1st WARNING SIGNAL move to P_i^{P}. The subsystem is being maintained and the 1st WARNING SIGNAL goes off at this moment. Tim fires if the PM action is finished. Subsequently, the tokens in P_i^{P} and P_1^{T} move to P_1^{M}, i.e. this error has been corrected. The marking of P_1^{M}, i.e. the maintenance times log number for P_1, increases by one. On the other hand, if the PM action does not take place in time, then $T_{1\mathrm{U}}$ fires such that P_1^{U} obtains a token. Consequently, the tokens in P_1^{U} and P_1^{T} move to P_1. Hence, P_1 failure occurs. At the same time, P_1^{F} obtains a token, i.e. the failure times log number for P_1 increases by one. Because of the logic relation between P_1 and P_2, the value of the monitored signal for P_2, i.e. S_2, exceeds the prescribed warning value due to P_i failure. Accordingly, the 2nd WARNING SIGNAL is produced and each of P_2^{T}, P_3^{B2} and P_2^{B1}, obtains a token. $T_{2\mathrm{E}}$ is inhibited by the token in P_2^{B2}, such that the tokens in P_2^{B1} and P_2^{B2} move to P_2^{R} and P_2^{W} after triggering the 2nd RESET R and the 1st RESET W respectively. Hence this error is located at P_i, whereas $M(P_2^{\mathrm{L}})$ does not increase.

2. Suppose the value of the monitored signal for shaft rotation speed, i.e. S_7 in Fig. 16, exceeds the prescribed warning value spontaneously while all S_3, S_4, and S_6 are at normal condition. As a result, $T_{7\mathrm{S}}$ fires such that the 7th WARNING SIGNAL goes on. Simultaneously, each of P_7^{T}, P_7^{B1} and P_8^{B2} obtains a token. In a similar manner, as S_1 exceeds the prescribed warning value, $M(P_7^{\mathrm{M}})$ increases by one if the PM action for P_7 takes place in time. Otherwise, P_7 failure occurs such that $M(P_7^{\mathrm{F}})$ increases by one. However, since P_7^{B2}, P_7^{B3} and P_7^{B4} are empty, $T_{7\mathrm{E}}$ fires such that P_7^{E} obtains a token. As a result, $M(P_7^{\mathrm{L}})$, i.e. the error times log number for P_7, increases by one after triggering the 7th RESET E. It indicates that this error is caused by the 7th monitored signal, i.e. the shaft rotation speed, but not by next lower signals, i.e. vapor pressure, recycling pump rotation speed, or return flow temperature.

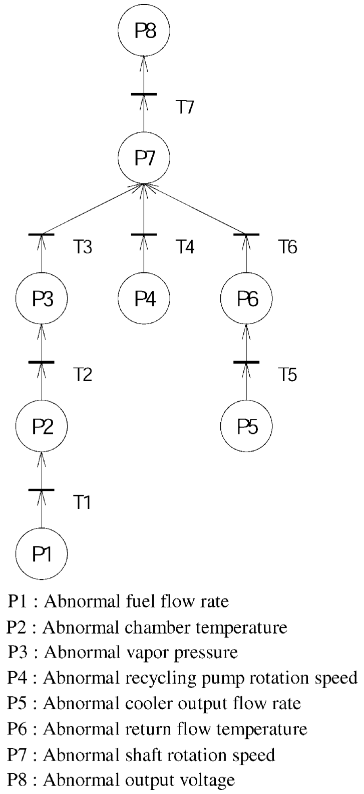

P8
T7
P7
T3
T4
T6
P3
P4
P6
T2
T5
P2
P5
T1
P1
P1 : Abnormal fuel flow rate
P2 : Abnormal chamber temperature
P3 : Abnormal vapor pressure
P4 : Abnormal recycling pump rotation speed
P5 : Abnormal cooler output flow rate
P6 : Abnormal return flow temperature
P7 : Abnormal shaft rotation speed
P8 : Abnormal output voltage

Fig. 15. The PNSF for the thermal power plant.

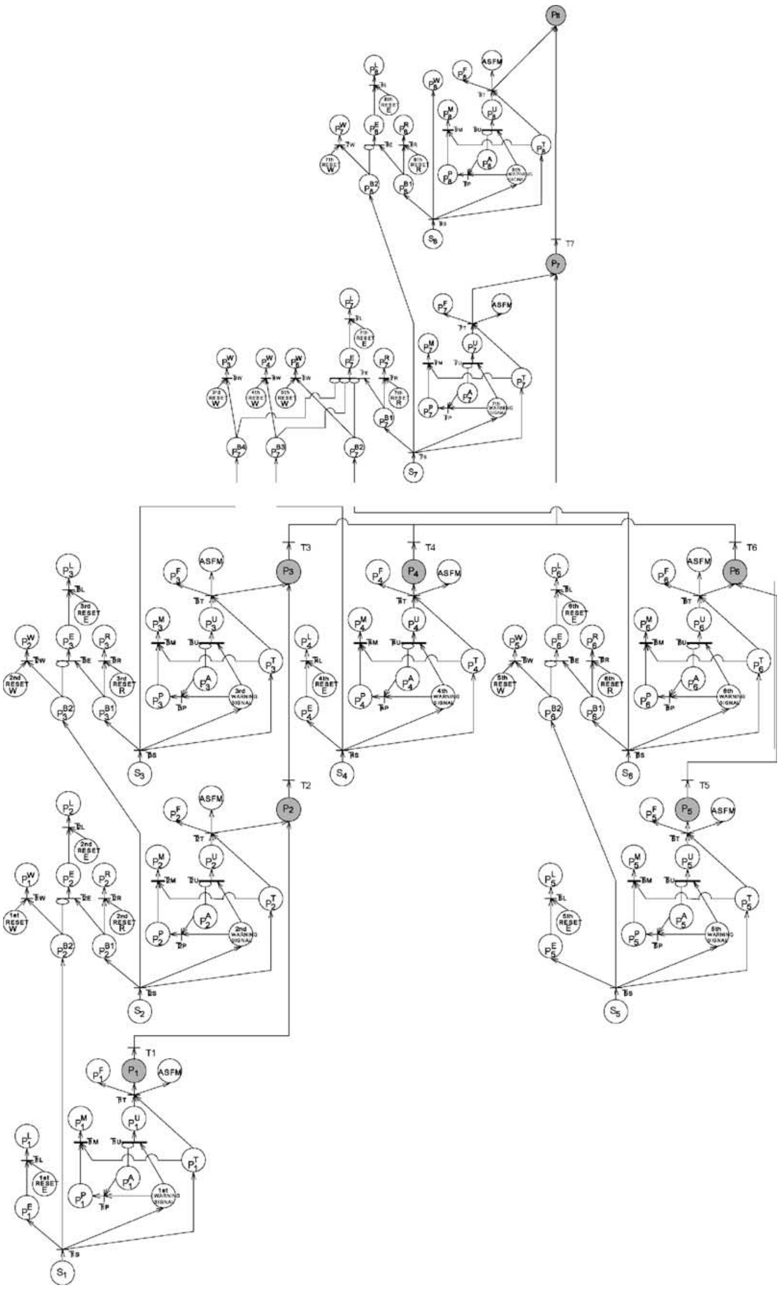


Fig. 16. The PNSF for the thermal power plant endowed with EFDIA.

ASFM should be incorporated with the places that may cause safety problem in a PNSF. In this example, each of P_1–P_8 failures can trigger ASFM.

6. Conclusions

Knowing when and where a system needs maintenance and economizing capital investment are two of the major problems of maintenance. The aforementioned approach improves the maintenance problem in the following aspects:

1. Before a failure of a system occurs, the approach is able to indicate where and when the failure is going to be.
2. It makes the health condition and the historical record of maintenance for a system clear at a glance.
3. It avoids the drawbacks of a timed-based maintenance, i.e. unavoidable risks of failure occurrence and investment waste.

This paper has presented an early failure detection and isolation scheme for PM via the thermal power plant example, by using a hybrid Petri net modeling method endowed with fault-tree analysis and parameter trend. The PNSF has to be constructed beforehand. The next task is to obtain control charts for all fault places in the PNSF in order to prescribe thresholds and allowable margins. With these prerequisites, the present method can be applied to any system. The introduced Petri net approach not only can achieve early failure detection and isolation for fault diagnosis but also facilitates event count, system state description, and automatic shutdown or regulation. These capabilities are very useful for health monitoring and preventive maintenance of a system. Besides, the Petri net with these capabilities, i.e. EFDIA, is not only done on paper but also actualized on an ASIC so that the proposed scheme is practicable. The presented method promotes the maintenance strategy for a thermal power plant from timed-based to condition-based.

Acknowledgements

The author is grateful to the National Science Council in Taiwan for supporting this study under grant number NSC 87-TPC-E009-005 and NSC 89-2213-E167-015.

References

[1] E.A. Elsayed, Reliability Engineering, Addison Wesley Longman, Taipei, 1996.

[2] J.D. Patton Jr., Preventive Maintenance, Instrument Society of America, New York, 1983.

[3] S.S. Rao, Reliability-Based Design, McGraw-Hill, New York, 1992.

[4] S.K. Yang, T.S. Liu, A Petri net approach to early failure detection and isolation for preventive maintenance, Quality Reliability Eng. Int. 14 (3) (1998) 319–330.

[5] R. Isermann, Process fault detection based on modeling and estimation methods—a survey, Automatica 20 (4) (1984) 387–404.

[6] A.S. Willsky, A survey of design methods for failure detection in dynamic systems, Automatica 12 (1976) 601–611.

[7] O.T. Ogunyemi, P.I. Nelson, Prediction of Gamma failure times, IEEE Trans. Reliability 46 (3) (1997) 400–405.

[8] W. Nelson, Weibull prediction of a future number of failures, Quality Reliability Eng. Int. 16 (1) (2000) 23–26.

[9] E.A. Rietman, M. Beachy, A study on failure prediction in a plasma reactor, IEEE Trans. Semicond. Manufacturing 11 (4) (1998) 670–680.

[10] S.N. Kher, G.M. Bubel, Predicting system-failure risk from unanticipated fiber-breaks in manufacturing, IEEE Trans. Reliability 47 (2) (1998) 126–130.

[11] J.C. Duthie, M.I. Robertson, A.M. Clayton, D.P.G. Lidbury, Risk-based approaches to ageing and maintenance management, Nuclear Eng. Design 184 (1) (1998) 27–38.

[12] L.J. Perryman, N.A.S. Foster, D.R. Nicholls, Using PRA in support of maintenance optimisation, Int. J. Pressure Vessels Piping 61 (2–3) (1995).

[13] B.J. Garrick, R.F. Christie, Probabilistic risk assessment practice in the USA for nuclear power plants, Safety Sci. 40 (1–4) (2002) 177–201.

[14] F. Niehaus, Use of Probabilistic Safety Assessment (PSA) for nuclear installations, Safety Sci. 40 (1–4) (2002) 153–176.

[15] H.D. Brewer, K.S. Canady, Probabilistic safety assessment support for the maintenance rule at Duke Power Compant, Reliability Eng. Syst. Safety 63 (3) (1999) 243–249.

[16] E. Pate-Cornell, R. Dillon, Probabilistic risk analysis for NASA space shuttle: a brief history and current work, Reliability Eng. Syst. Safety 74 (3) (2001) 345–352.

[17] F.I. Khan, A. Iqbal, N. Ramesh, S.A. Abbasi, SCAP: a new methodology or safety management based on feedback from credible accident-probabilistic fault tree analysis system, J. Hazard. Mater. 87 (1–3) (2001) 23–56.

[18] M. Cepin, B. Mavko, A dynamic fault tree, Reliability Eng. Syst. Safety 75 (1) (2002) 83–91.

[19] J.M. Koren, J. Caertner, CAFTA: a fault tree analysis tool designed for PSA, in: Proceedings of Probabilistic Assessment and Risk Management: PSA'87, vol. 2, Zurich, Switzerland, May 1987, pp. 588–592.

[20] G. Degen, J. Mertens, S.B. Reer, ESAP, an easy-to use expert system for the systematic analysis of operator actions within the scope of probabilistic risk assessment, Nuclear Eng. Design 159 (2–3) (1995) 259–264.

[21] T. Kumamaru, T. Utsunomiya, Y. Yamada, A fault diagnosis system for district heating and cooling facilities, in: Proceedings of the IECON'91 International Conference on Industrial Electronics, Control, and Instrumentation, vol. 1, Kobe, Japan, 28 October–l November 1991, pp. 131–136.

[22] A. Baccigalupi, A. Bernieri, A. Pietrosanto, A digital-signal-processor-based measurement system for on-line fault detection, IEEE Trans. Instrum. Measurement 46 (3) (1997) 731–736.

[23] M. Rausand, K. Oien, The basic concept of failure analysis, Reliability Eng. Syst Safety 53 (2) (1996) 73–83.

[24] R. David, H. Alia, Petri nets for modeling of dynamic systems–a survey, Automatica 30 (2) (1994) 175–202.

[25] G.S. Hura, J.W. Atwood, The use of Petri nets to analyze coherent fault trees, IEEE Trans. Reliability 37 (5) (1988) 469–474.

[26] W.G. Schneeweiss, Petri net for reliability modeling, Lilole-Verlag, Hagen, 1999.

[27] S.K. Yang, T.S. Liu, Failure analysis for an airbag inflator by Petri nets, Quality Reliability Eng. Int. 13 (2) (1997) 139–151.

[28] G. Ciardo, J. Muppala, K. Trivedi, SPNP: stochastic petri net package, in: Proceedings of the 3rd International Workshop on Petri Nets and Performance Models, Kyoto, Japan, December 1989, pp. 142–151.

[29] A.W. Shaw, Logic Circuit Design, Saunder College Publishing, Fort Worth, 1993.

[30] Xilinx, Foundation Series Quick Start Guide Version F1.4, The Programmable Logic Company, San Jose, 1998.

[31] Xilinx, XACTLibraries Guide, The Programmable Logic Company, San Jose, 1994.

[32] Xilinx, Hardware User Guide, The Programmable Logic Company, San Jose, 1998.

[33] CIC, Training Course 12, Chip Implementation Center of Nation Science Council, Republic of China, 1998.

Shang-Kuo Yang was born in Taiwan. He received the B.S. in 1982 and the M.S. in 1985 in automatic control engineering from Feng Chia University, Taiwan. From 1985 to 1991, he was an assistant researcher and instrumentation system engineer of Flight Test Group, Aeronautic Research Laboratory, Chung Shan Institute of Science and Technology, Taiwan. Since 1991, he has been with the Department of Mechanical Engineering at National Chin Yi Institute of Technology, Taiwan, where he is an associate professor. He received the Ph.D. in 1999 in Mechanical Engineering from National Chiao Tung University, Taiwan. His research interests are in reliability, data acquisition, and automatic control.