RECOVERY FROM INTERMITTENT FAULTS FOR INPUT/STATE ASYNCHRONOUS SEQUENTIAL MACHINES

Jung–Min Yang  
Department of Electrical Engineering  
Catholic University of Daegu  
330 Kumrak, Hayang, Gyeongsan  
Gyeongbuk 712-702, Republic of Korea  
email: jmyang@cu.ac.kr

Seong Woo Kwak  
Department of Electronic Engineering  
Keimyung University  
Sin-Dang-Dong 1000, Dal-Seo-Gu  
Daegu 704-701, Republic of Korea  
email: ksw@kmu.ac.kr (corresponding author)

ABSTRACT  
This paper presents a corrective control scheme that automatically counteracts the adverse effect of intermittent faults in the logic of asynchronous sequential machines. When an intermittent fault occurs to the machine, a set of state transitions defined in the machine suspends the normal behavior for a finite interval. The objective is to design a corrective controller that makes the closed-loop system show the normal input/state behavior despite the occurrence of an intermittent fault. We show that realization of fault tolerance depends on a certain reachability property of the asynchronous machine. For validating the applicability of the proposed control scheme, we implement the closed-loop system on VHDL code.

KEY WORDS  
Fault tolerance, corrective control, asynchronous sequential machines, intermittent faults.

1 Introduction  
Intermittent faults are a class of faults in dynamic systems where faulty behavior or a malfunction of the system occurs and endures for some finite time [1]. Several contributing factors, such as internal system faults or disturbance entities from the outer environment, combine their influence to invoke an intermittent fault. In hardware systems, intermittent faults are typically caused by bad electrical contacts, sticky components, overheating of chips, noisy measurements from sensors, power surges, and so forth [2].

In this paper, we address the problem of how to make asynchronous sequential machines immune to the attack of intermittent faults. Endowed with logic of clockless sequencing, asynchronous sequential machines have distinctive features such as stable/unstable states, instantaneous state transitions, and fundamental mode operation. Since asynchronous machines are still widely used as an important building block of digital systems [3], [4], we have to develop a robust schemes of fault tolerance for asynchronous machines against typical classes of faults, among which intermittent faults are given a high priority.

We use a control-theoretic approach to realizing fault tolerance. Corrective control is a novel automatic control scheme specified for asynchronous sequential machines. A corrective controller, also in the form of an asynchronous machine, is placed in front of the controlled machine and receives the external input and the output feedback from the machine to generate a control input. A remarkable advantage of corrective control is the capability that it can compensate the stable-state behavior of the asynchronous machine in a desirable way without resort to redesign of machine’s inner logic. A series of research results about corrective control has been presented during the past decade; refer to [5]–[10] and the references cited in these papers.

In our modeling, when an intermittent fault occurs to the considered asynchronous machine, a set of state transitions defined in the machine suspends the normal behavior for a finite interval. This setting reflects the property of intermittent faults that faulty behavior caused by the intermittent fault endures at intervals before its adverse effects diminishes. As a result, the asynchronous machine loses part of its reachability during the manifestation of the intermittent fault. We will address that if the machine has latent reachability enough to tolerate the adverse effect of the intermittent fault, one can design an appropriate corrective controller that materializes fault tolerance. Under the framework of corrective control, the closed-loop system can continue to show the normal behavior as if no intermittent fault happened to the machine.

The rest of this paper is organized as follows. In Section 2, we model the considered asynchronous sequential machine and the adverse effect of intermittent faults on the characteristics of state transitions. In Section 3, we present the condition for the existence of a corrective controller that realizes fault tolerance against intermittent faults. It is shown that realization of fault tolerance depends on a certain reachability property of the asynchronous machine in the failure mode. We also propose a novel control procedure to accommodate the occurrence of the intermittent fault and its self counteracting. In Section 4, for validating the applicability of the proposed control scheme, we implement the closed-loop system on VHDL code and conduct an experimental study using a fault scenario. Finally, conclusion and remarks on future studies are addressed in Section 5.
2 Preliminaries

2.1 Problem Statement

In this paper, we consider input/state asynchronous machines, where the present state of the machine is given as the output value. The corrective control system for an input/state asynchronous machine is illustrated in Figure 1. The closed-loop system represented by the diagram.

Figure 1. Corrective control system for the asynchronous machine \( \Sigma \) with the intermittent fault \( \omega \).

In this paper, we consider input/state asynchronous machines, where the present state of the machine is given as the output value. The corrective control system for an input/state asynchronous machine is illustrated in Figure 1. The closed-loop system represented by the diagram.

Our objective is to present the existence condition and design procedure for \( C \) that diagnoses and tolerates the adverse effect of intermittent faults. The closed-loop system \( \Sigma_c \) will seem to maintain the normal behavior even though part of the machine’s state transition characteristics is invalidated by fault for a certain interval. This is made possible by virtue of latent redundancy in the reachability of the machine and the feature of asynchronous operation of the closed-loop system. To this end, the controller \( C \) should achieve the following specific functions.

i) First, \( C \) should recognize fault diagnosis. In other words, it has to determine whether an intermittent fault occurs to the machine \( \Sigma \) without violating the normal input/state specification of the machine.

ii) After diagnosing an occurrence of the fault, \( C \) must generate an appropriate control input \( u \) in response to the change of the external input \( v \) so that \( \Sigma_c \) can continue to show the normal behavior. This is the critical component of fault tolerance in our study.

iii) Since the adverse effect of the intermittent fault vanishes after finite steps in our model, \( C \) should also perceive whether \( \Sigma \) returns to the normal status, i.e., whether part of state transitions that degenerated by fault regain their functionality.

In this paper, the event that the adverse effect of the intermittent fault vanishes is termed **self counteracting**.

We assume that the control configuration of Figure 1 abides by the principle of **fundamental mode operation** [11], an operating policy that prohibits the simultaneous change of two or more variables. This policy helps to prevent uncertainties arising from simultaneous changes in two or more variables. According to fundamental mode operation, both the occurrence of the intermittent fault \( \omega \) and its self counteracting should occur when \( \Sigma \) is at a stable combination.

2.2 System Modeling

An input/state asynchronous machine \( \Sigma \) is modeled by a 4-tuple

\[
\Sigma := (A, X, x_0, f),
\]

where \( A \) is the input set, \( X \) is a set of \( n \) states, \( x_0 \in X \) is the initial state, and \( f : X \times A \to X \) is the state transition function defined on \( X \times A \). \( \Sigma \) operates according to a recursion of the form

\[
x_{k+1} = f(x_k, u_k), \quad k = 0, 1, 2, \ldots
\]

The current state \( x_k \) goes to the next state \( x_{k+1} \) asynchronously in response to a switch of the input to \( u_k \), where \( k \) represents the step counter of the machine, which advances by one upon a change of the machine’s input or state.

Any valid state–input pair \( (x, v) \in X \times A \) is a **stable combination** if \( f(x, v) = x \), where \( x \) is called a **stable state** with \( v \); otherwise, it is a **transient combination** and \( x \) is an **unstable state** with \( v \). A transient combination \( (x, v) \) induces a chain of transient transitions, e.g.,

\[
f(x, v) = x_1, f(x_1, v) = x_2, \ldots,
\]

until it reaches the next **stable state** \( x' = f(x', v) \).

Owing to the lack of a synchronizing clock, the transition time between transient combinations is very fast (ideally, in zero time). Thus from a user’s viewpoint, only a transition from a stable combination \( (x, v) \) to the next stable combination \( (x', v) \), termed a **stable transition**, appears. The stable recursion function \( s : X \times A \to X \) extracts this stable transition from \( \Sigma \). For state–input pair \( (x, v) \), \( s \) is defined as

\[
s(x, v) := x',
\]

where \( x' \) is the next stable state of \( (x, v) \). When an input sequence \( t = v_1v_2 \cdots v_k \in A^+ \) is applied to a state \( x \), \( s \) is extended recursively as \( (A^+ \text{ the set of all non-empty strings of characters of } A) \)

\[
s(x, t) := s(s(x, v_1), v_2 \cdots v_k).
\]

If there exists an input sequence \( t \in A^+ \) such that \( x' = s(x, t) \), \( x' \) is said to be **stably reachable** from \( x \). It is known that any stably reachable state can be reached in at most \( n - 1 \) steps, where \( n \) is the cardinality of the state set \( X \) [5].
the machine not to respond to any invalid input character. In terms of state transitions, receiving an input character that makes an invalid pair with the present state is equivalent to the behavior of a stable combination—staying at the present state. Hence, by assigning a stable combination to every invalid state-input pair, we can make \( f \) a total function. If a state \( x \in X \) has transient combinations with all the elements of a set \( A \subseteq A \), then we define

\[
f(x, v) = x, \quad \forall v \in A \setminus A_x,
\]

where \( \setminus \) denotes the set difference.

To deal with intermittent faults, we partition the input set \( A \) into

\[
A := A_N \cup A_F,
\]

where \( A_N \) is the normal input set and \( A_F \) is the set of intermittent faults. As addressed before, the occurrence of an intermittent fault causes a temporary halt of a set of state transitions. For brevity’s sake, this paper assumes that there is only one intermittent fault \( \omega \), i.e., \( A_F := \{ \omega \} \), with the associated set of state transitions \( F \subseteq X \times A_N \):

\[
F := \{(z_i, v_i)|1 \leq i \leq |F|\}.
\]

Without loss of generality, we assume that every \( (z_i, v_i) \in F \) is a transient combination; stable combinations would not be influenced by the intermittent fault.

As addressed before, we stipulate that once the intermittent fault attacks \( \Sigma \), its adverse effect persists only for some finite steps, after which it attenuates or vanishes. Here, a step implies a change of the external input character \( v \) that enters into the asynchronous machine \( \Sigma \) (see Figure 1). Define \( l \) as the maximum number of steps within which \( \Sigma \) stays under the influence of the intermittent fault \( \omega \), that is, all the transitions of \( F \) degenerate. This means that after the fault occurrence, the transitions of \( F \) may go back to the normal mode any time within \( l \) steps. After \( l \) steps, they will definitely be back to the normal mode. Note that the exact moment that \( F \) returns to the normal mode is nondeterministic; only the maximum duration \( l \) of the failure mode is available. We limit the range of \( l \) to

\[
1 \leq l \leq n - 1,
\]

where \( n = |X| \). If \( l > n - 1 \), \( \Sigma \) can traverse the entire states before returning to the normal mode, which becomes equivalent to the case of permanent changes of the characteristics of state transitions [12]. Using the stable recursion function \( s \), we summarize as follows the situation of the fault occurrence discussed so far.

a) For all \( (z_i, v_i) \in F \), \( i = 1, \ldots, |F| \), \( s(z_i, v_i) = z_i \) if \( \omega \) occurs to \( \Sigma \) and its effect does not vanish yet. Staying at the stable state \( z_i \), \( \Sigma \) would not respond to the input character \( v_i \).

b) Else if the effect of \( \omega \) vanishes, or \( l \) steps elapse after the occurrence of \( \omega \), \( s(z_i, v_i) = z_i' \), where \( z_i' \) is the next stable state of \( (z_i, v_i) \).

### 3 Corrective Controller

#### 3.1 Stable Reachability

In the problem of model matching between the asynchronous machine and a reference model, a corrective controller exists if and only if stable reachability of the machine is greater than or equal to that of the model [5], [6]. Our problem of fault tolerance can be interpreted as another model matching problem. We regard the asynchronous machine under attack of the intermittent fault as the system to be controlled, and the original machine with the nominal behavior as the model. Then, we can design a corrective controller that realizes fault tolerance, provided that stable reachability of the machine that loses part of its state transitions (\( F \) defined above) is equal to that of the nominal machine. Note that stable reachability of the machine under attack of the intermittent fault can be never greater than that of the nominal machine.

Let us define the state set as \( X := \{x_1, x_2, \ldots, x_n\} \). Remind that stable reachability from a state \( x_i \) to another one \( x_j \) means the existence of an input string \( t \in A_N \) with \( 1 \leq |t| \leq n - 1 \) such that \( s(x_i, t) = x_j \). In our modeling, the adverse effect of the intermittent fault \( \omega \) terminates at most in \( l \) steps. After the occurrence of \( \omega \), \( \Sigma \) can reach from a state to those states that are reachable from the original state in \( l \) or less steps, instead of \( n - 1 \) steps. In our study, hence, we only have to consider stable reachability with the length of \( l \) steps.

In the former studies [5], [8], we used the skeleton matrix to describe stable reachability of the machine \( \Sigma \). In this paper, we define \( l \)-skeleton matrix, termed \( K_l(\Sigma) \), to represent stable reachability of \( \Sigma \) measured in \( l \) steps. \( K_l(\Sigma) \) is an \( n \times n \) matrix whose \((i, j)\) element \( K_{i,j}(\Sigma) \) is defined as

\[
K_{i,j}(\Sigma) := \begin{cases} 
1 & \exists r \in A^+_N \text{ s.t. } 1 \leq |r| \leq l \text{ and } s(x_i, r) = x_j \\
0 & \text{otherwise, } i, j = 1, \ldots, n
\end{cases}
\]

\( K_{l}(\Sigma) \) provides in a compact way the information on whether a state is stably reachable from another state in \( l \) or less steps.

To describe stable reachability of \( \Sigma \) with faulty behavior, we denote by \( \Sigma_F \) the asynchronous machine \( \Sigma \) in which all the state transitions of \( F \) lose their functionality, as designated in item a). The corresponding skeleton matrix, termed \( K(\Sigma_F) \), has its \((i, j)\) element defined as follows.

\[
K_{i,j}(\Sigma_F) := \begin{cases} 
1 & \exists r \in A^+_N \text{ s.t. } 1 \leq |r| \leq n - 1 \text{ and } s(x_i, r) = x_j \\
0 & \text{otherwise, } i, j = 1, \ldots, n
\end{cases}
\]

Note that we do not have to impose the length limit \( l \) on \( K(\Sigma_F) \) because in the operation of corrective control, the stable transitions between the states of the machine will be made instantaneously [5], [8]. Thus the corrective controller \( C \) can utilize the maximal reachability of the machine by employing input strings with the length up to \( n - 1 \).
Using the skeleton matrices, we now postulate the following existence condition for a corrective controller that realizes fault tolerance against the intermittent fault $\omega$.

**Proposition 1.** An input/state asynchronous sequential machine $\Sigma = (A, X, x_0, f)$ has the intermittent fault $\omega$ and the associated set of state transitions $F \subset X \times A$. Then, there exists a corrective controller $C$ for which the closed-loop system $\Sigma_c$ of Figure 1 is fault tolerant against the adverse effect of $\omega$ if and only if

$$K^f(\Sigma) = K(\Sigma_F),$$

where the equality is taken entry-by-corresponding-entry.

The proof of the above proposition can be induced in a manner similar to the problem of model matching (e.g., refer to [5], [7], [8]).

### 3.2 Control Mechanism

In view of Figure 1, the controller $C$ perceives an occurrence of the intermittent fault $\omega$ by observing the state feedback $x$ from $\Sigma$. Assume that $\Sigma$ stays at a stable combination with the state $z_i$ when the external input $v$ changes to $v_i$. Since $(z_i, v_i) \in F$, we expect that one of the following two events will occur if $C$ relays $v_i$ to $\Sigma$ as the control input $u$: If the intermittent fault $\omega$ has not infiltrated into $\Sigma$ yet, $\Sigma$ would have the normal behavior, transferring to the next stable state $s(z_i, v_i)$. On the other hand, if $\omega$ has occurred to $\Sigma$ and all the transitions of $F$ have stopped their normal functionality, $\Sigma$ would not respond to the input $v_i$. Either of the two events can be recognized by $C$ with the state feedback $x$. $x$ is observed to be $s(z_i, v_i)$ or $x = z_i$ depending on the occurrence of $\omega$.

Once certifying the occurrence of the intermittent fault $\omega$, $C$ begins corrective control action for realizing fault tolerance. $C$ has two inputs—the external input $v$ and the state feedback $x$—and one output $u$ that serves as the control input to $\Sigma$. Hence $C$ is a deterministic input/output asynchronous machine whose form is

$$C = (X \times A, A, \Xi, \xi_0, \phi, \eta),$$

where $\Xi$ is the state set, $\xi_0 \in \Xi$ is the initial state, $\phi : \Xi \times X \times A \to \Xi$ is the recursion function, and $\eta : \Xi \times X \times A \to A$ is the output function.

Assume again that $\Sigma$ has been at the stable state $z_i$ and the intermittent fault $\omega$ has occurred to $\Sigma$, when the external input changes to $v_i$. To determine whether $\omega$ has really happened, $C$ delivers $v_i$ to $\Sigma$ as the control input by setting $u = v_i$. Upon receiving the incorrect state feedback $x = z_i$, $C$ knows that all the transitions of $F$, including $(z_i, v_i)$, degenerate by fault. Note that this does not mean that $\omega$ has occurred just before the external input $v_i$ is transmitted. It may have occurred several steps before; the only information available is that it has occurred before less than $l$ steps.

Let $z_i := x_p$ and $s(z_i, v_i) := x_q$. In view of the skeleton matrix $K^f(\Sigma)$, this means $K^f_p(\Sigma) = 1$. Assuming that Proposition 1 is guaranteed, $K^f_p(\Sigma_F) = 1$ leads to $K^f_p(\Sigma_F) = 1$. Hence there exists an input sequence $r := u_1 u_2 \cdots u_l \in A^+_l$ such that $s(x_p, r) = x_q$. $r$ serves as the control input sequence for realizing fault tolerance from $z_i = x_p$ and $s(z_i, v_i) = x_q$.

Let us describe in detail the procedure of corrective control. In the beginning, $C$ is at the initial state $\xi_0$. The controller remains in its initial state until it detects a stable combination with the state $z_i$. $C$ then moves to $\xi_i$, termed the transition state [7]. In fundamental mode operation, an input change or occurrence of the intermittent fault can occur only when the machine stays at a stable combination. At $\xi_i$, hence, $C$ anticipates that the input character $v_i$ may enter into the system, thereby preparing fault tolerance.

When the external input $v$ switches to $v_i$ at $\xi_i$, $C$ relays it to the machine. Upon receiving the incorrect state feedback $x = z_i$ (and identifying the occurrence of $\omega$), $C$ initiates corrective action by generating the first control input $u_1$ to $\Sigma$. $\Sigma$ will be driven to $x^1 := s(z_i, u_1)$, namely the first intermediate state of the machine. $C$ then receives the state feedback $x^1$ and generates the second control input $u_2$, which will further drive $\Sigma$ to $x^2 := s(x^1, u_2)$, the second intermediate state and so forth until $\Sigma$’s reaching the goal state $s(z_i, v_i)$. Since all these interactions between $C$ and $\Sigma$ are made asynchronously, the entailed stable transitions seem transient and invisible from the outer user’s viewpoint. In this way, the closed-loop system $\Sigma$ accomplishes fault tolerance, i.e., in response to the external input $v_i$, it transits to the desired next stable state $s(z_i, v_i)$ despite the occurrence of $\omega$.

Following the above notations, we denote all the intermediate states $\Sigma$ passes through by $x^1, x^2, \ldots, x^{k-1}$ with the following relation:

$$\begin{align*}
x^1 &= s(z_i, u_1) \\
x^2 &= s(x^1, u_2) \\
&\vdots \\
x^{k-1} &= s(x^{k-2}, u_{k-1}) \\
s(z_i, v_i) &= s(x^{k-1}, u_k)
\end{align*}$$

Then, $T(z_i, v_i) \subset X \times A_N$, namely the set of all the transient combinations traversed by $\Sigma$ in the correction behavior for $(z_i, v_i)$, is deduced to

$$T(z_i, v_i) = \{ (z_i, u_1), (x^1, u_2), \ldots, (x^{k-1}, u_k) \}.$$
Identifying self counteracting, or finding the moment the state transitions of \( F \) regain their normal behavior, is solved automatically when the controller \( C \) is faced with any state–input pair belonging to \( F \). Remind that even after the occurrence of \( \omega \), we specify the behavior of \( C \) such that it relays the external input \( v_t \) to the machine \( \Sigma \) whenever \( \Sigma \) is at the stable state \( z_s \). If \( \Sigma \) stays still in the failure mode, the state feedback remains unchanged from \( z_s \). \( C \) will then continue to carry out the correction behavior as described above. On the contrary, if the state feedback is observed to be \( x = s(z_s, v_t) \), \( C \) concludes that the adverse effect of the intermittent fault \( \omega \) has vanished. After identifying, the closed-loop system \( \Sigma_c \) will return to the normal mode in which \( C \) does not conduct any control action; it just relays the external input \( v \) to the machine by setting \( u = v \).

4 VHDL Experiment

Consider an input/state asynchronous sequential machine \( \Sigma \) whose state flow diagram is shown in Figure 2. Here, \( \mathcal{A}_\Sigma = \{a, b, c, d\} \), \( \mathcal{A}_F = \{\omega\} \) (unseen in Figure 2), and \( X = \{x_1, x_2, x_3, x_4\} \) with \( x_0 := x_1 \). Solid arrows represent normal state transitions and dashed arrows are the elements of \( F \), the set of the state transitions associated with \( \omega \):

\[
F = \{(x_1, a), (x_2, b)\}.
\]

When \( \omega \) occurs to \( \Sigma \), the transitions of \( F \) becomes

\[
s(x_1, a) = x_1
\]

\[
s(x_2, b) = x_2.
\]

We assume \( l = 3 \), namely the adverse effect of \( \omega \) vanishes at most 3 steps after its occurrence.

To check the existence of a corrective controller, we first calculate skeleton matrices. A slight examination of Figure 2 leads to

\[
K^3(\Sigma) = \begin{pmatrix}
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]

\[
K(\Sigma_F) = \begin{pmatrix}
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]

Since \( K^3(\Sigma) = K(\Sigma_F) \), by Proposition 1 there exists a corrective controller \( C \) that accomplishes fault tolerance. Technically speaking, \( K^3(\Sigma) = K(\Sigma_F) \) implies that even though the transitions of \( F \) degenerate by the intermittent fault \( \omega \), the overall stable reachability of \( \Sigma \) remains intact.

The control input sequences used in corrective control for each state–input pair of \( F \) are selected as follows.

\[
(x_1, a) \:: r_1 = bd
\]

\[
(x_2, b) \:: r_2 = cb
\]

Clearly, neither \( r_1 \) nor \( r_2 \) induces an element of \( F \) in the corrective control path. For example, \( T(x_2, b) \) is derived as

\[
T(x_2, b) = \{(x_2, c), (x_1, b)\}
\]

and \( T(x_2, b) \cap F = \emptyset \).

Figure 2. State flow diagram of the example machine \( \Sigma \).

We have implemented on VHDL code the closed-loop system \( \Sigma_c \) comprising the asynchronous machine \( \Sigma \) and the corrective controller \( C \), and have conducted an experiment using a fault scenario. It is assumed that the intermittent fault \( \omega \) occurs to \( \Sigma \) when the machine stays at the stable combination \((x_2, a)\). In our scenario, after 3 steps, i.e., after the external input values changes 3 times, the adverse effect of \( \omega \) vanishes.

Figure 3 shows the result of the VHDL experiment illustrating the control activity of \( C \). \( b_1b_0 \) are state bits with the assignment of \( x_1 = 00, x_2 = 01, x_3 = 11, \) and \( x_4 = 10 \). Note that all the signals are interpreted as rising edge-triggered. The asynchronous machine \( \Sigma \) is at the initial state \( x_1 \) at \( t = 0 \,\text{ns} \), and in response to the change of the input to \( a \), it moves to the next stable state \( s(x_1, a) = x_2 \) at \( t = 40 \,\text{ns} \). At this instant, the intermittent fault \( \omega \) occurs to \( \Sigma \), freezing the state–input pairs of \( F \). As marked in Figure 3, the external input changes to \( b \) at \( t = t_1 \). But, since \( \Sigma \) is in the failure mode and \((x_2, b) \in F \), the transition does not give the correct result; instead, the machine \( \Sigma \) remains stuck at the state \( x_2 \). Observing this situation, \( C \) perceives the occurrence of \( \omega \) and begins the corrective control activity.

As the control input sequence for \((x_2, b)\) is chosen to be \( r_2 = cb \), \( C \) provides \( \Sigma \) with the first control input character \( c \) at time \( t_2 \). \( \Sigma \) then transfers to \( x_1 \), the first intermediate state in the correction path. Next, \( C \) gives the second control input \( b \) to \( \Sigma \), which moves to the goal state \( x_3 \) at time \( t_3 \), completing fault tolerance. Since this corrective control is carried out asynchronously, an outer user will observe that \( \Sigma \) moves directly from \( x_2 \) to \( x_3 \) in response to the external input \( b \).

After this control phase, the external input changes again to \( d \) (at time \( t_4 \)) and \( c \), in response to which \( \Sigma \) moves to \( x_2 \) and \( x_1 \). Because the external input is switched three
times in total, the adverse effect of $\omega$ vanishes right after the external input changes to $c$, i.e., when $\Sigma$ reaches the stable combination $(x_1, c)$. We see at time $t_5$ of Figure 3 that the next external input is $a$ for which $(x_1, a) \in F$. According to the proposed control scheme, $C$ relays $a$ to $\Sigma$ and upon receiving the right state feedback $x_2$ at time $t_6$, it ensures that the transitions of $F$ return to the normal status.

Considering the instantaneous transition characteristics of the closed-loop system shown in Figure 3, we assert that this experiment result validates the applicability of the proposed control scheme to real-world digital systems.

5 Conclusion

In this paper, a corrective control law has been proposed in order to tolerate the intermittent faults occurring to input/state asynchronous sequential machines. The considered intermittent fault has the feature that its influence lasts for a finite time. Realizing fault tolerance depends on the remaining stable reachability of the machine given after part of its transitions degenerates by fault. Also, the corrective controller can know both the occurrence and termination of the intermittent fault whenever the machine enters into the degenerated state-input pairs. The result of the VHDL experiment validates the applicability of the proposed fault tolerance scheme.

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (No. 2012-003419).

References