Fault Tolerant Control for Spaceborne Dual Ring Counters with Selective Overriding

Jung–Min Yang1, Im Y. Jung1* and Seong Woo Kwak2*

1School of Electronics Engineering, Kyungpook National University, Daegu, 702-701, South Korea; iyjung@ee.knu.ac.kr
2Department of Electronic Engineering, Keimyung University, Daegu, 704-701, South Korea; ksw@kmu.ac.kr

Abstract
Among primary asynchronous sequential circuits, ring counters are used as an important building block in various spaceborne digital systems. As they are subject to adverse effects of radiation, it is imperative that they be given fault tolerance mechanism for robust operation. This paper presents a corrective control scheme for tolerating state transition faults in ring counters. The considered ring counter has the structure of Double Modular Redundancy (DMR), in which some selected memory bits of each ring counter module can override the counterpart values of the other module. With this structure, we can design a static corrective controller that tolerates any state transition fault occurring to the dual ring counter. Though the state information is unavailable, we can implement a simple fault tolerant control scheme using corrective control. The proposed control scheme is validated in VHDL experiments.

Keywords: Asynchronous Sequential Circuits, Double Modular Redundancy (DMR), Ring Counters, State Transition Faults

1. Introduction
High-performance digital systems have been increasingly employed in space environments. For instance, many experimental satellites are using Field-Programmable Gate Arrays (FPGAs) in implementing their core ingredients, mostly due to low cost, reconfigurability, and low design turn-around time of FPGAs. On the other hand, static RAM cells that are constituents of FPGAs and other digital systems are significantly susceptible to a category of radiation-related faults. As immediate repair by humans is impossible in space applications, fault diagnosis and tolerance in digital systems with SRAM cells or sequential circuits is a requisite for guaranteeing successful completion of their works.

In this paper, we focus on realizing fault tolerant asynchronous sequential circuits, in particular, a class of spaceborne ring counters. A ring counter is a kind of counters that consist of circular shift registers where the output of the last shift register is fed to the input of the first register. A single one bit encircles the ring, one step forward in response to the input. In spaceborne digital systems, ring counters are utilized as an important building block, e.g., a controller for finite state machines involved in memory scrubbing and an element of error detection and correction circuits (EDACs) for satellites. We assume that the dual ring counter is subject to state transition faults, where the system undergoes unauthorized state transitions by fault. Single Event Upsets (SEUs) occurring to SRAM cells are commonly observed state transition faults in space environments.

To provide fault tolerance capability, we endow the ring counter with the following two kinds of hardware redundancy:

1) Our ring counter is designed to have the structure of Double Modular Redundancy (DMR), that is, a pair of identical ring counters is assigned to do the same counting work in parallel. Since task duplication by DMR can be used to diagnose transient faults that cannot be detected internally, it increases the reliability of the system. In the states of the two processors are compared
to detect faults in normal task duplication procedures by DMR. However, the foregoing study assumes that the dual ring counter has a comparison module, which the system proposed in this paper does not use.

2) As a fault tolerance mechanism, a set of specific bits of one counter module can override the corresponding bits of another counter module and vice versa. For the simplicity of design, the overriding logic will be defined as minimal, i.e., all the designated bits are overridden simultaneously by a single command.

The main objective is to propose a corrective controller that diagnoses state transition faults and activates the overriding logic so that the closed-loop system can seem immune to state transition faults. The discussion of this paper is a continuation of 8,9. Compared with the former studies, however, the present paper has the following advantages. First, the dual ring counter developed in this paper has smaller dimensions of the input and output characters than the previous systems. Secondly, by virtue of the overriding logic, the procedure of fault diagnosis and tolerance of our scheme can be accelerated.

The proposed mechanism of fault diagnosis and tolerance is based on corrective control, a novel and efficient automatic control scheme that can change the stable-state behavior of an asynchronous sequential machine. Regarding the dual ring counter as a class of input/output asynchronous machines, we will develop an appropriate corrective controller that realizes the aforementioned fault diagnosis and tolerance. Recent studies dealing with the corrective control scheme can be found in 10,13.

2. Preliminaries and Problem Statement

We introduce modeling of asynchronous machines by adopting from the former studies 8,9,10. The dual ring counter discussed in this paper belongs to a class of asynchronous sequential machines termed input/output machines, where direct access to the present state is unavailable. A deterministic input/output asynchronous machine \( \Sigma \) is described as the following finite state machine:

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

where, \( A \) is the input set, \( Y \) is the output set, \( X \) is the state set, and \( x_0 \in X \) is the initial state. \( f: X \times A \to X \) and \( h: X \to Y \) (assuming a Moore machine) are the state transition function and the output function, respectively, defined as partial functions. To represent adversarial input characters that cause state transition faults, we divide the input set \( A \) into two mutually exclusive subsets:

\[
A = A_n \cup A_d,
\]

where, \( A_n \) includes the normal inputs and \( A_d \) is the set of all the adversarial inputs.

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

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

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

\[
\vdots
\]

until it reaches the next stable state

\[
x_k = f(x_{k-1}, v) = f(x_k, v).
\]

Due to the lack of a synchronizing clock, these transitions are executed instantaneously (in zero time, ideally). As a result, from an outer user’s viewpoint, \( \Sigma \) seems to move directly from \((x, v)\) to \((x_k, v)\). The stable recursion function \( s: X \times A \to X ^{14} \) represents this feature of asynchronous machines:

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

where, \( x_k \) is the next stable state of \((x, v)\). A chain of transitions from one stable combination to another, as described by \( s \), is called a stable transition. \( s \) is often extended from input characters to sequences recursively: for \( x \in X, v \in A \) and \( t \in A^* \),

\[
s(x, vt) := s(s(x, v), t),
\]

where, \( vt \) is the concatenation of \( v \) and \( t \).

2.1 Definition 1

A state \( x' \) is said to be stably reachable from another state \( x \) if there exists an input sequence \( t \in A^* \) such that \( x' = s(x, t) \). If \( t \in A_d^* \), \( x' \) is \( n \)-stably reachable from \( x^{10} \).

Stable reachability between the states is an essential property \( \Sigma \) must be endowed with for guaranteeing the existence of a corrective controller. When \( \Sigma \) experiences a stable transition from \( x \) to \( x_k \) between which there are transient states \( x_1, \ldots, x_{k-1} \), it generates a string of output values

\[
h(x)h(x_1)\cdots h(x_k) \in Y^*,
\]

called an output burst. A burst is a rapidly progressing string of output characters which, ideally, has duration of...
zero time. The use of bursts allows a corrective controller to estimate the present state more easily at the cost of higher controller complexity\textsuperscript{15, 16}. In our study, however, we will design the dual ring counter such that there is no transient state between any two stable states, that is, all the transient combinations end their transitions in one step ($k = 1$ in the former statement). Thus we do not consider the output burst for the rest of this paper.

Figure 1 is the corrective control system for the dual ring counter $\Sigma$ with the adversarial input (compare to\textsuperscript{9}). $C$ is another asynchronous machine that serves as the corrective controller. $w \in A_a$ is the normal external input, $u \in A_a$ is the control input generated by $C$, and $y \in Y$ is the output of the machine $\Sigma$ that is delivered to $C$ as the feedback. We denote by $\Sigma$, the closed-loop system represented by the diagram. In the former studies on input/output control of asynchronous machines\textsuperscript{16, 15}, the controller requests an observer to estimate the current state of the machine. But, since $y$ is supposed to be always a single character, no observation module is involved in the control configuration of Figure 1.

$w \in A_d$ denotes the adversarial input that attacks $\Sigma$, invoking state transition faults. As an asynchronous machine is driven by changes of its input variables, $u$ and $w$ override each other. The real input to $\Sigma$ is thus defined as one of $u$ and $w$ whose value changes at the last. The controller $C$ receives as its input the external input $v$ and the output feedback $y$ from the machine $\Sigma$. $C$ perceives an occurrence of a state transition fault whenever it observes that $y$ changes while $v$ remains unchanged. In particular, assume that $\Sigma$ has been staying at a stable combination $(x, y) \in X \times A$ with the output value $h(x)$. $C$ asserts a fault occurrence when the output $y$ changes to another value $y' \neq h(x)$ whereas $v$ is not changed.

Referring to Figure 1, $C$ is described by an input/output asynchronous machine of the form

$$ C := (Y \times A_n, A_n, \Xi, \xi_0, \varphi, \eta), $$

where, $Y \times A_n$ is the input set, $A_n$ is the output set, $\Xi$ is the state set, $\xi_0 \in \Xi$ is the initial state, $\varphi : \Xi \times Y \times A_n \rightarrow \Xi$ is the recursion function, and $\eta : \Xi \rightarrow A_n$ is the output function.

### 3. Modeling of Dual Ring Counters

Adopting from the former result of\textsuperscript{6} and\textsuperscript{7}, we present a specific model of the $n$-bit dual ring counter $\Sigma = (A, Y, X, x_0, f, h)$. Assume that $n$ is an even number, namely $n = 2m$. Let $P$ and $Q$ be two separate ring counter modules, and let $p_i$ and $q_i$, $i = 1, \ldots, n$, be the $i$th counter bit of each module, respectively. We define a state of the dual ring counter $\Sigma$ as its $2n$ counter bits, each $n$ bits assigned to $P$ and $Q$. Thus the state set is

$$ X = \{(p_1, p_2, \ldots, p_n, q_1, q_2, \ldots, q_n) | p_i, q_i \in \{0, 1\} \}, $$

with $x_n = (10\ldots0, 10\ldots0)$.

The normal input set $A_n$ is defined as

$$ A_n = \{SO, SE, OV_P, OV_Q\}. $$

The elements of $A_n$ are classified according to their functionality. $SO$ and $SE$ are the external counter inputs. In response to these input characters, logic ‘one’ values of counter modules $P$ and $Q$ advance one step forward simultaneously, maintaining their positions at the same with each other. $SO$ is applied to $\Sigma$ when logic one is at an odd bit ($p_{2k+1}$ and $q_{2k+1}$), and $SE$ is applied at an even bit ($p_{2k}$ and $q_{2k}$). $OV_P$ and $OV_Q$ are overriding commands used only by the controller when implementing fault tolerance. $OV_P$ overrides all the pre-specified positions of the counter bits of module $P$ with the values of the corresponding bits of module $Q$. In this paper, we select as the overridden positions all even bits of each counter module. In response to $OV_P$, hence, all even bits of $P$ are overridden by the logic values of the corresponding even bits of $Q$, namely

$$ p_i := q_i, \text{ by } OV_P, \quad i = 2, 4, \ldots n. $$

$OV_P$ is the overriding command that changes the values of the counter module $Q$.

$$ q_i := p_i, \text{ by } OV_Q, \quad i = 2, 4, \ldots n. $$

Figure 2 illustrates the normal commands of the dual ring counter $\Sigma$.

When expressing a state transition fault or SEU occurring to a counter bit, we must designate which bit is
corrupted and which logic value is upset by fault. To this end, let us define the adversarial input set \( A_d \) as
\[
A_d = \{e_{h,i}, z_{h,i} \mid h \in \{P, Q\}, i \in \{1, \ldots, n\}\},
\]
where, \( e_{h,i} \) designates the state transition fault that upsets the \( i^{\text{th}} \) bit of counter module \( h \) from 1 to 0, and \( z_{h,i} \) from 0 to 1. For example, \( e_{h,i} \) denotes the state transition fault occurring to the first bit \( p_1 \) of counter module \( P \) when the logic value of \( p_1 \) is one.

Once a state transition fault occurs to \( \Sigma \), the corrective control mechanism will be executed instantaneously, returning \( \Sigma \) to normal state. As long as an appropriate controller exists for \( \Sigma \), no more than one fault manifests itself in the counter bits, and the logic values of the other bits are zero. Hence it is convenient to use the following simpler notation:
\[
x := [p_1 = 1, q_1 = 1]
\]
that represents a normal state where logic one stays at the \( i^{\text{th}} \) bit of both counter modules. If \( e_{h,i} \), occurs to counter module \( P \) so that bit \( p_i \) is upset to 0, the resulting state \( x' \) is described by:
\[
x' := [0, q_i = 1].
\]
Similarly, if \( z_{h,i} \) occurs so that another bit \( p_i \) is upset from 0 to 1, we denote the resulting state, say \( x'' \), by:
\[
x'' := [p_i = 1, q_i = 1].
\]

With a slight abuse of notation, we denote by \( [\cdot]_p \) and \( [\cdot]_q \) the state bits of counter modules \( P \) and \( Q \), respectively. For instance, the state \( x'' \) induces \( [p_i = 1, q_i = 1]_p \) and \( [q_i = 1]_q \).

The output character \( y \) of \( \Sigma \) consists of the following four bits.
\[
y := (BO_pBE_p, BO_QBE_Q)
\] (1)

\( BO_p \) and \( BE_p \) are the exclusive-OR of all the odd and even bits of counter module \( P \), respectively:
\[
BO_p = p_1 \oplus p_3 \oplus \cdots \oplus p_{n-1},
BE_p = p_2 \oplus p_4 \oplus \cdots \oplus p_n
\]
\( BO_Q \) and \( BE_Q \) are the corresponding output bits for counter module \( Q \).

An advantage of the above definition of output characters is that the output dimension is always constant without regard to the bit size \( 2n \) of the dual ring counter \( \Sigma \). In the former study, on the other hand, the output character has five bits, where another bit is obtained by exclusive-O Ring all the real counting outputs. Hence the present study facilitates the ring counter design by decreasing its output size by one.

4. Overriding Logic

With the output \( y \) defined in (1), the controller \( C \) obviously has no direct access to the present state \( x \) of \( \Sigma \). Still, \( C \) can discern whether a state transition fault has occurred only by observing \( y \). Further, using the overriding command, immediate fault tolerance can be made for any fault situation, because each normal state \( [p_i = 1, q_i = 1] \) of the ring counter \( \Sigma \) is \( n \)-stably reachable from any faulty state. To validate the latter arguments, assume that \( \Sigma \) has been staying at a normal state \( x = [p_i = 1, q_i = 1] \), when a state transition fault occurs. Without loss of generality, we suppose that the fault happens to a bit of counter module \( P \); the case of counter module \( Q \) can be easily derived using symmetry.

\( A \). Fault \( e_{p,i} \)

An occurrence of \( e_{p,i} \) implies that the present logic value ‘one’ of counter module \( P \) is upset by fault. First, assume that \( i \) is an even number. Then, the output of the state \( x = [p_i = 1, q_i = 1] \) is
\[
y = h(x) = (01, 01).
\]

By \( e_{p,i} \), \( \Sigma \) is forced to move to a faulty state
\[
x' = [0, q_i = 1]
\]
with the corresponding output
\[
y' = h(x') = (00, 01).
\]

Upon observing that the output character changes from \( y \) to \( y' \), the controller \( C \) discerns the occurrence of \( e_{p,i} \). Since the counting bit (logic one) is attacked, \( C \) knows the exact location of the state transition fault.
The mechanism of fault tolerance is straightforward. As soon as receiving the output feedback $y'$, the controller $C$ generates the overriding command $OV'$. In response to $OV'$, all the logic values of even bits of counter module $P$ will be replaced by nominal values of counter module $Q$. To realize this behavior in a well-defined form, $C$ first moves to a state $ξ_i$ in response to $y'$, i.e., we set the recursion function $φ$ as

$$φ(ξ_i, (00, 01), v) = 01 \quad \forall v \in A_n$$

$$φ(ξ_i, (00, 01), v) = 01 \quad \forall v \in A_n \quad (2)$$

As no control action is executed at the initial state $ξ_v$,

$$η(ξ_v) = v, \quad (3)$$

where, $v$ is the present external input. At $ξ_v$, $C$ provides the control input $OV'$.

$$η(ξ_v) = OV' \quad (4)$$

When, $C$ receives the original output character $y = (01, 01)$, $C$ knows that fault tolerance is completed, and returns to $ξ_v$.

$$φ(ξ_v, (01, 01), v) = 00 \quad (5)$$

Secondly, assume that $i$ is an odd number. By the definition of (1), the normal output of $Σ$ at $x = [p_i, q_i = 1]$ is

$$y = h(x) = (10, 10).$$

Once bit $q_i$ is upset from 1 to 0 by the fault $ε_{p, i}$, $Σ$ reaches the next stable state $x' = [0, q_i = 1]$ with the corresponding output

$$y' = h(x') = (00, 10).$$

Since an odd bit of counter module $P$ is corrupted, direct fault tolerance using the overriding command $OV'$ is not possible. In this case, hence, the controller $C$ actuates multiple steps of the correction procedure. First, upon receiving the output feedback $y' = (00, 10)$, $C$ transfers to a state $ξ_v$. To this end, set $φ$ as

$$φ(ξ_v, (00, 10), v) = 00 \quad \forall v \in A_n$$

$$φ(ξ_v, (00, 10), v) = 00 \quad \forall v \in A_n \quad (6)$$

Since the faulty bit has an odd position $i$, the corrupted value is not removed by $OV'$, which only overrides all the even bits of counter module $P$. Instead, $C$ makes $Σ$ advance one step forward by generating the control input $SO$:

$$η(ξ_v) = SO \quad (7)$$

In response to $SO$, the normal one bit of counter module $Q$ moves from $q_i$ to $q_{i+1}$, while all the bits of $P$ remain 0's. The output character thus becomes $(00, 01)$. Receiving this output, $C$ transfers to the next state $ξ_v$.

$$φ(ξ_v, (00, 01), v) = 00 \quad \forall v \in A_n$$

$$φ(ξ_v, (00, 01), v) = 00 \quad \forall v \in A_n \quad (8)$$

At $ξ_v$, $C$ provides the overriding command $OV'$, which makes equal all the corresponding bits of $P$ and $Q$.

$$η(ξ_v) = OV' \quad (9)$$

The output character further changes to the normal value $(01, 01)$. Notice that though $(01, 01)$ is the normal output character, it is not the same as the original one $(10, 10)$, since logic one value is at the $(i + 1)$th bit of $P$ and $Q$. To move logic one value to the initial position, the $ith$ bit, $C$ must provide $Σ$ with the external counter input $n−1$ times, i.e., $SE, SO, …, SE$. Thus $C$ requires $n−1$ more auxiliary states, termed $ξ_{v,1}, …, ξ_{v,n−1}$. The following recursive behavior realizes the correction trajectory.

$$φ(ξ_v, (01, 01), v) = 01 \quad \forall v \in A_n$$

$$φ(ξ_v, (01, 01), v) = 01 \quad \forall v \in A_n \quad (10)$$

At $ξ_{v,n−1}$, finally, fault tolerance is completed when $C$ receives the output character $(10, 10)$, which implies that the dual ring counter $Σ$ reaches the original state $x = [p_i, q_i = 1]$. After ensuring the end of the correction procedure, $C$ returns to its initial state $ξ_v$.

$$φ(ξ_{v,n−1}, (10, 10), v) = 00 \quad (11)$$

B. Fault $z_{p, j}$, $j \neq i$

$z_{p, j}$ is the state transition fault that upsets a bit value of counter module $P$ from 0 to 1. As logic one stays at bit $p_j$, this fault must occur to another bit, namely bit $p_j$, where $j \neq i$. The procedure of fault diagnosis and tolerance for $z_{p, j}$ is similar to the case of $ε_{p, i}$, and some fault cases are already solved by the pre-assigned behavior of $C$. For the sake of simplicity, we suppose that logic one is at an even position, i.e., $i$ is an even number. The corresponding output character is $y = (01, 01)$.

First, assume that $z_{p, j}$ occurs to an even bit of counter module $P$, or $j$ is an even number. $Σ$ then moves to a state $x''$:

$$x'' = [p_i, p_j = 1, q_i = 1].$$
Since \( i \) is supposed to be an even number, the occurrence of \( z_{p,j} \) causes parity bit \( BE_p \) to be zero, so the resulting output value \( y'' \) is

\[
y'' = h(x'') = (00, 01).
\]

The former design (2)-(5) of the controller C can deal with this situation. By (2), C moves to the state \( \xi \) in response to the output feedback \((00, 01)\). Then, by the definition of (4), C generates the overriding command \( OV_p \), which resets all the even bits of P (including \( i \) and \( j \)) with the values of \( Q \). In this way, fault tolerance is accomplished and \( C \) returns to \( \xi \) (see (5)).

Next, assume that \( z_{p,j} \) occurs to an odd bit of counter module \( P \), or \( j \) is an odd number. As before, \( \Sigma \) is driven to the state \( x'' = [p_j = 1, q_j = 1] \). Since both odd and even bits of \( P \) contain logic one value, parity bits \( BO_p \) and \( BE_p \) become one, leading to the output value

\[
y'' = h(x'') = (11, 01).
\]

The correction procedure for this situation is very similar to the former assignment (6)-(11). In association with the states \( \xi', \xi_2', ..., \xi_{n-1}' \), we define another set of auxiliary states \( \xi''', \xi''_2', ..., \xi''_{n-1}' \in \Xi \). Upon observing the output character \((11, 01)\), let \( C \) move to the state \( \xi''_2' \).

\[
\begin{align*}
\phi(\xi''_2', (11, 01), v) &= c''_2 \quad \forall v \in A_n \\
\phi(\xi''_2', (11, 01), v) &= c''_2 \quad \forall v \in A_n 
\end{align*}
\]

As logic value ‘one’ stays at an even bit \( i \), \( C \) gives \( \Sigma \) the counter input \( SE \) at \( \xi''_2' \).

\[
\eta(c''_2) = SE. \quad (13)
\]

Observing that the output bits of \( Q \) change from 01 to 10, \( C \) moves to \( \xi_3' \) (output bits of \( P \) remain unchanged).

\[
\begin{align*}
\phi(\xi''_3', (11, 01), v) &= c''_3 \quad \forall v \in A_n \\
\phi(\xi''_3', (11, 01), v) &= c''_3 \quad \forall v \in A_n 
\end{align*}
\]

At \( \xi''_3' \), \( C \) resets the faulty bit by providing the overriding command.

\[
\eta(c''_3) = OV_p, \quad (15)
\]

which will induce the normal output \((10, 10)\). Since logic one value has advanced one step forward to the \((i + 1)\)th bit, \( C \) must provide \( \Sigma \) with the external counter input \( n-1 \) times starting from \( SO \), i.e, \( SO, SE, ..., SO \). Imitating \((10)\) and \((11)\), we complete the controller construction by assigning \( \varphi \) and \( \eta \) at \( \xi''_2', ..., \xi''_{n-1}' \) as follows.

\[
\begin{align*}
\phi(\xi''_2', (10, 10), v) &= c''_2 \\
\eta(\xi''_2') &= SO \\
\phi(\xi''_3', (10, 10), v) &= c''_2 \\
\eta(\xi''_3') &= SE \\
&\vdots \\
\phi(\xi''_{n-2}, (10, 10), v) &= c''_{n-1} \\
\eta(\xi''_{n-2}) &= SO \\
\phi(\xi''_{n-1}, (01, 01), v) &= \xi_0 
\end{align*}
\]

### Conclusion

The proposed corrective controller \( C \) ensures fault tolerance by employing a selective overriding scheme in VHDL. We have validated the applicability of the proposed fault tolerance scheme in VHDL experiments. To further enhance the fault tolerance capability, we have conducted two experiments on fault diagnosis and tolerance.

### 5. Case Study

To validate the fault tolerance scheme, we have implemented the 4-bit dual ring counter \( \Sigma \) and the proposed corrective controller \( C \) in VHDL code and have conducted two experiments on fault diagnosis and tolerance.

#### A. Fault \( e_{p,2} \)

First, we achieve fault tolerance against \( e_{p,2} \), the state transition fault occurring to bit \( p_2 \) of counter module \( P \) while its logic value is one. Figure 3 shows the experiment result for this case study. \( \Sigma \) starts from the initial state \( x_0 = [p_1 = 1, q_1 = 1] \) and in response to the external counter input, it experiences the normal transition to the next state \([p_2 = 1, q_2 = 1]\) at \( t_1 = 48 \) nsec. \( e_{p,2} \) happens to \( \Sigma \) at \( t_2 = 84 \) nsec, causing the unauthorized state transition to \([p_2 = 0, q_2 = 1]\) at \( t_3 \).

As addressed in the previous section, the controller \( C \) perceives the fault occurrence by observing that the output changes from \((01, 01)\) to \((00, 01)\). We can see that at \( t_3 \), \( BE_p \) drops to logic zero in Figure 3. According to (2), \( C \) transfers to the state \( \xi \), and generates the control input \( OV_p \) at \( t_4 \) (see (4)). At \( t_5 = 101 \) nsec, finally, \( \Sigma \) returns to the original state \([p_2 = 1, q_2 = 1]\), which we validate by observing

![Figure 3. Experiment result for tolerating fault \( e_{p,2} \).](image-url)
that $BE_p$ recovers to logic one. $\tau(e_{p,2}) \in A^+_{\alpha}$, namely the control input sequence induced by $C$ for solving $e_{p,2}$, is

$$\tau(e_{p,2}) = OV_p.$$ 

The duration between the fault occurrence and the completion of fault tolerance is

$$t_4 - t_2 = 17nsec.$$ 

Since 17 nsec is shorter than the frequency of a common synchronizing clock, we assert that the proposed scheme materializes instantaneous fault tolerance for the dual ring counter.

**B. Fault $z_{p,3}$**

Next, consider the case of $z_{p,3}$, the state transition fault occurring to bit $p_2$ of counter module $P$ while its logic value is zero. For clarity, we continue to assume that the logic value ‘one’ has stayed at $p_3$ and $q_3$ at the moment of the fault occurrence. Figure 4 is the corresponding experiment result. In Figure 4, $z_{p,3}$ occurs at time $t_9$, when the (unobserved) state changes from $[p_2 = 1, q_2 = 1]$ to $[p_2 = p_3 = 1, q_2 = 1]$, and the output from (01, 01) to (11, 01) (see the change of $BO_p$ in the figure). According to (12) and (13), the controller C first moves to the state $K'_p$, and provides $Z$ with the counter input $SE$ at $t_6$. Observing that the output feedback changes to (11, 10), $C$ moves to the state $K'_p$, generating the overriding command $OV_p$ at $t_7$ (by (14) and (15)). Then, from (16) and the fact that $n = 4$, three more counter inputs are generated at $t_8$, $t_9$ and $t_10$. Finally, at $t_11$, the dual ring counter returns to the original state $[p_2 = 1, q_2 = 1]$, completing the fault tolerance procedure. $\tau(z_{p,3}) \in A^+_{\alpha}$, namely the control input sequence induced by $C$ for solving $z_{p,3}$, is

$$\tau(z_{p,3}) = SE, OV_p, SO, SE, SO.$$ 

![Figure 4. Experiment result for tolerating fault $z_{p,3}$](image)

**6. Conclusion**

In this paper, we have proposed a structure of spaceborne ring counters with fault tolerance capability against state transition faults. As a kind of hardware redundancy, the ring counter has DMR and the function of overriding logic for the even bits of each counter module. Based on the corrective control scheme, we have presented a novel fault diagnosis and tolerance process in which the corrective controller compensates any state transition fault under asynchronous mechanism. The proposed structure of the dual ring counter has the virtue that the output dimension remains constant regardless of the size of the counter bits, and no state observer is required for estimating the inner state of the counter. We have validated the applicability of the proposed fault tolerance scheme in VHDL experiments.

**7. Acknowledgement**

The work of J.-M. Yang was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. NRF-2012R1A2A2A01003419), and in part by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. NRF-2010-0007271). The work of S. W. Kwak was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. NRF-2010-0021366).

**8. References**

1. Samudrala PK, Ramos J, Katkoori S. Selective triple modular redundancy (STMR) based single-event upset (SEU) tolerant synthesis for FPGAs. IEEE Trans Nucl Sci. 2004; 51(5):2957–69.
2. Sterpone L, Violante M. Analysis of the robustness of the TMR architecture in SRAM-based FPGAs. IEEE Trans Nucl Sci. 2005; 52(5):1545–9.
3. Karp S, Gilbert BK. Digital system design in the presence of single event upsets. IEEE Trans Aero Electron Syst. 1993; 29(2):310–6.
4. Umanesan G, Fujiwara E. A class of random multiple bits in a byte error correcting and single byte error detecting (St/b EC-Sb ED) codes. IEEE Trans Comput. 2003; 52(7):835–47.
5. Reyes P, Reviriego P, Maestro JA, Ruano O. New protection techniques against SEUs for moving average filters in
a radiation environment. IEEE Trans Nucl Sci. 2007; 54(4):957–64.

6. Kwak SW, Yang J-M. A fault detection and self-recovery system for space-borne dual ring counters. Trans Kor Inst Electr Eng (in Korean). 2013; 62(1):120–6.

7. Kwak SW, You KH, Yang J-M. Checkpoint management with double modular redundancy based on the probability of task completion. J Comput Sci Tech. 2012; 27(2):273–80.

8. Ziv A, Bruck J. Performance optimization of checkpointing schemes with task duplication. IEEE Trans Comput. 1997; 46(12):1381–6.

9. Yang J-M, Kwak SW. Applying input/output control of asynchronous sequential machines to dual ring counters. Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics; 2012. p. 2407–12.

10. Yang J-M, Kwak SW. Realizing fault-tolerant asynchronous sequential machines using corrective control. IEEE Trans Contr Syst Tech. 2010; 18(6):1457–63.

11. Yang J-M, Kwak SW. Corrective control of asynchronous machines with uncontrollable inputs: application to SEU error counters. IET Control Theory and Applications. 2010; 4(11):2454–62.

12. Yang J-M, Kwak SW. Fault diagnosis and fault-tolerant control of input/output asynchronous sequential machines. IET Control Theory and Applications. 2012; 6(11):1682–9.

13. Peng J, Hammer J. Bursts and output feedback control of non-deterministic asynchronous sequential machines. Eur J Contr. 2012; 18(3):286–300.

14. Murphy TE, Geng X, Hammer J. On the control of asynchronous machines with races. IEEE Trans Automat Contr. 2003; 48(6):1073–81.

15. Geng X, Hammer J. Input/output control of asynchronous sequential machines. IEEE Trans Automat Contr. 2005; 50(12):1956–70.

16. Yang J-M. Corrective control of input/output asynchronous sequential machines with adversarial inputs. IEEE Trans Automat Contr. 2010; 55(3):755–61.