



정적 교정 제어기를 이용한 비동기 순차 회로의 내 고장성 구현

Static Corrective Controllers for Implementing Fault Tolerance in Asynchronous Sequential Circuits

양정민* · 광성우**†

Jung-Min Yang and Seong Woo Kwak[†]

*경북대학교 전자공학부, **계명대학교 전자공학과

*School of Electronics Engineering, Kyungpook National University

**Department of Electronic Engineering, Keimyung University

요 약

비동기 순차 회로를 위한 교정 제어기는 회로를 재설계하지 않고도 회로 내에 존재하는 여러 고장을 탐지하고 극복하는 능력을 보인다. 이번 논문에서는 교정 제어기의 크기를 줄이기 위한 방법으로 정적 교정 제어기(static corrective controller)를 제안한다. 동적 제어기에 비해 정적 제어기는 제어기 상태가 필요 없으므로 조합 회로(combinational circuit)만으로 구현 가능하다. 본 논문에서는 상태 천이 고장에 대한 정적 내고장성 교정 제어기가 존재할 조건과 설계 과정을 규명한다. 또한 제안된 제어 기법을 FPGA로 구현된 SEU 오류 카운터에 적용하여 그 효율성을 실험적으로 검증한다.

키워드: 비동기 순차 회로, 교정 제어, 내고장성, 정적 제어기

Abstract

Corrective controllers enable fault diagnosis and tolerance for various faults in asynchronous sequential circuits without resort to re-design. In this paper, we propose a static corrective controller in order to decrease the size of the controller. Compared with dynamic controllers, static controllers can be made using only combinational circuits, as they need no inner states. We address the existence condition and design procedures for static corrective controllers that overcome state transition faults. To show the validity and advantage, the proposed controller is applied to an SEU error counter implemented on FPGA.

Key Words: Asynchronous Sequential Circuits, Corrective Control, Fault Tolerance, Static Controllers.

Received: Mar. 8, 2016
Revised: Mar. 11, 2016
Accepted: Apr. 14, 2016
[†]Corresponding authors
ksw@kmu.ac.kr

1. 서 론

교정 제어(corrective control)는 비동기 순차 회로의 과도 상태(transient state) 동작이 매우 빠르다는 성질을 이용하여 회로의 내부 로직(logic)을 바꾸지 않고도 페루프 시스템의 안정 상태(stable state) 특성을 변경할 수 있다[1]. 교정 제어는 비동기 순차 회로에서 발생하는 여러 가지 고장을 탐지하고 극복하는 데 이론 및 실험적 성취를 보였으며[2, 3], 최근에는 불리언(Boolean) 네트워크의 해석 방법과 결합되는 등[4] 다양한 형태로 확장되고 있다.

본 논문에서는 비동기 순차 회로에서 발생하는 상태 천이 고장을 극복할 수 있는 정적(static) 교정 제어 기법을 제안한다. 이전 연구[1~4]에서 개발된 교정 제어기는 모두 내부 상태가 필요한 동적(dynamic) 제어기 형태였다. 비동기 회로의 특성상 페루프 시스템의 속도는 제어 과정에서 거치는 제어기의 상태 개수와 상관없이 매우 빠르다. 하지만 고장이 존재하는 환경, 특히 실리콘 반도체에 고장을 야기하는 방사선 환경에서 동작하는 동적 교정 제어기는 제어 대상 시스템과 마찬가지로 고장의 영향을 받을 수밖에 없으므로 고장을 해결하기 위해 탑재한 제어기 자체가 고장 나는 문제가 생긴다.

이번 연구에서는 이러한 문제를 해결하기 위해서 조합 회로만으로 설계 가능한 정적 교정 제어기를 제안한다. 정적 제어기는 순차 머신으로 이루어진 동적 제어기보다 여러 가지 고장에 대한 영향을 덜

이 논문은 2015년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (No. NRF-2015R1A2A1A15054026). 이 논문은 2015년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. NRF-2015R1D1A1A01056764). 이 논문은 2015년도 정부(미래부)의 재원으로 한국연구재단 바이오·의료기술개발사업의 지원을 받아 수행된 연구임 (No. 2015M3A9A7067220).

This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

받기 때문에 고신뢰도를 요구하는 비동기 디지털 시스템을 운용할 때 유용할 것이다. 교정 제어의 목적은 고장에 의해서 원하지 않는 상태 천이를 겪은 대상 회로를 원래의 상태로 복귀시키는 일이다. 고장 탐지와 복구 동작이 모두 비동기적으로 이루어지므로 상태 천이 고장의 영향이 최소화될 수 있다.

본 논문에서는 먼저 교정 제어 이론을 바탕으로 정적 내고장성 교정 제어기가 존재할 도달가능성(reachability) 조건을 해석적으로 규명하고 제어기 설계과정을 제시한다. 동적 제어기에 비해 정적 제어기의 설계 및 동작은 더 간단하다. 또한 제안된 기법의 효용성을 보이기 위해서 SEU(single event upset) 고장이 존재하는 비동기 오류 카운터에 정적 교정 제어기를 부착한 페루프 시스템을 FPGA로 구현하고 제어기의 성능을 실험적으로 검증한다.

2. 상태 천이 고장이 있는 비동기 순차 회로

비동기 순차 회로 Σ 를 다음과 같이 정의하자.

$$\Sigma = (A, X, x_0, f)$$

A 는 입력 집합, X 는 상태 집합, x_0 는 초기 상태, $f: X \times A \rightarrow X$ 는 상태 천이 함수이다. 상태 천이 고장을 일으키는 외란 입력을 정의하기 위해서 $A = A_n \cup A_d$ 로 분류한다. 여기서 A_n 은 정상 입력이며 A_d 는 외란 입력이다. Σ 의 상태 $x \in X$ 는 입력 $v \in A$ 에 따라서 안정 상태 또는 과도 상태가 된다. $f(x, v) = x$ 이면 x 는 안정 상태이며 d 는 x 에 계속 머무른다. $f(x, v) \neq x$ 이면 x 는 과도 상태이며 d 는 x 에서 시작하여 일련의 과도 상태를 지나 다음 안정 상태에 도달한다. Σ 가 과도 상태에 머무르는 시간이 극히 짧기 때문에 보통 과도 상태를 생략하고 안정 상태 간의 동작만으로 Σ 의 동작을 표현한다. 이를 위해 'stable recursion 함수' $s: X \times A \rightarrow X$ 를 도입한다[1]. Σ 가 (x, v) 에서 시작하여 $f(x, v) = x_1, f(x_1, v) = x_2, \dots, f(x_{k-1}, v) = x_k, f(x_k, v) = x_k$ 의 천이 과정을 가진다면 $s(x, v) = x_k$ 로 정의되며 Σ 가 거치는 과도 상태 x_1, \dots, x_{k-1} 은 생략된다. 또 s 의 정의구역은 아래와 같이 길이 2 이상의 입력 스트링(string)으로 확장 가능하다.

$$s(x, v_1 v_2 \dots v_k) := s(s(x, v_1), v_2 \dots v_k), v_1 v_2 \dots v_k \in A^+$$

그림 1은 고장이 존재하는 비동기 교정 제어 시스템을 나타낸 것이다. Σ 는 제어 대상 비동기 회로, C 는 교정 제어기이며 Σ_c 는 Σ 와 C 가 결합된 페루프 시스템이다. $v \in A_n$ 은 외부 입력, $u \in A_n$ 은 C 가 생성하는 제어 입력, $x \in X$ 는 상태 피드백이고 $d \in A_d$ 는 상태 천이를 야기하는 고장 입력이다. 고장 입력 d 가 유입되는 순간 Σ 의 입력은

u 에서 d 로 바뀌며, Σ 는 원하지 않는 상태 천이를 겪을 수 있다. 이러한 고장을 복구하지 못하면 Σ 는 다음 외부 입력 변화에 따라서 비정상적인 동작을 계속할 것이다.

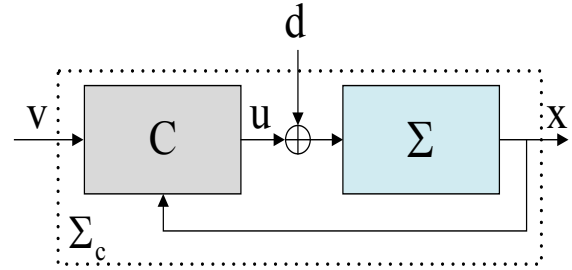


그림 1. 고장이 존재하는 비동기 교정 제어 시스템
Fig. 1. Corrective control system with faults

C 의 역할은 상태 천이 고장을 탐지하고 Σ 를 고장 전 상태로 즉시 복귀시키는 일이다. 본 논문에서는 기존 연구(6)와 차별화하여 C 를 정적 제어기로 설계한다. 메모리가 없는 정적 제어기 C 는 순차 머신(sequential machine)이 아니라 아래와 같이 외부 입력 v 와 상태 피드백 x 에 대한 단순 함수 형태이다.

$$C: X \times A \rightarrow A, C(x, v) = u \tag{1}$$

Σ 가 정상 동작을 한다면 C 는 외부 입력 v 를 그대로 제어 입력 $u = v$ 로 해서 Σ 에 전달한다. Σ 가 x 에 있을 때 상태 천이 고장을 일으키는 입력 d 가 발생했다고 하자. C 는 제어 입력 u 를 변경시키지 않은 상태에서 상태 피드백이 x 에서 $s(x, d)$ 로 변하는 것을 보고 고장 발생을 감지한다. Σ 를 고장 발생으로 도달할 수 있는 임의의 상태 $s(x, d)$ 에서 원래 상태 x 로 되돌리는 일이 교정 제어의 목적이 된다. 본 연구에서는 (1)의 형태를 지닌 정적 내고장성 교정 제어기 C 가 존재할 조건과 C 의 설계 과정을 제시한다.

Σ_c 는 비동기 머신의 기본 모드(fundamental mode) 동작 원리를 만족시키도록 설계한다. 즉 C 와 Σ 중 한 회로가 상태 천이를 할 때 다른 회로는 안정 상태에 있어야 하며, Σ_c 의 두 개 이상의 변수 값이 동시에 변하지 말아야 한다. 기본 모드 원리에 따르면 d 가 발생할 때 Σ 는 안정 상태에 있어야 한다. Σ 의 과도 상태 속도가 매우 빠르므로 이것은 일반성을 잃지 않는 가정이라 말할 수 있다.

3. 정적 교정 제어기

그림 1의 Σ 가 안정 상태 x 에서 고장 d 에 의해 $s(x, d)$ 로 천이하였다고 하자. Σ 를 원래 상태 x 로 되돌리는 동적 내고장성 교정 제어기가 존재할 조건은 Σ 가 $s(x, d)$ 에서 x 까지 도달가능성을

지녀야 한다는 것이다. 이 조건을 명시적으로 표현하면 아래와 같다
[6].

$$\forall d \in A_d \text{ s.t. } f(x,d), \exists t_d \in A_n^+ \text{ s.t. } s(s(x,d), t_d) = x \quad (2)$$

$f(x,d)$ 는 f 가 (x,d) 에서 정의된다는 의미이다. f 가 (x,d) 에서 정의되지 않으면 $f(x,d)$ 라고 표기한다. 입력 스트링 t_d 를 이용하여 내고장성 제어 과정이 구현되며, 이 과정에서 동적 제어기는 t_d 의 길이 $|t_d|$ 만큼의 보조 상태를 정의해야 한다.

(2)는 그림 1의 정적 제어기 C를 설계하기 위해서도 반드시 필요한 조건이다. 하지만 C는 내부 상태 없이 만들어져야 하므로 (2) 이외에도 조건이 더 필요하다. (1)의 정의를 보면 C는 인수 $(x,v) \in (X, A_n)$ 를 가지는 함수이다. 이것은 주어진 (x,v) 에 대해서 함수 값(=제어 입력) u 가 반드시 한 개만 대응되어야 한다는 의미이다. 그런데 Σ 가 정상동작을 할 때는 u 가 반드시 외부 입력 v 와 동일해야 한다. 즉 (x,v) 가 Σ 의 유효한 상태/입력 조합이라면, 즉 $f(x,v)$ 이면 C의 함수 값은 (x,v) 에 대해 무조건 아래와 같이 할당되어야 한다.

$$C(x,v) = v \quad \forall (x,v) \in X \times A_n, f(x,v) \quad (3)$$

Σ 가 안정 상태 x 에 있다가 고장 입력 d 에 의해 $s(x,d)$ 로 천이하였다고 다시 가정하자. 또 고장 발생 순간 x 와 안정 조합을 이루고 있었던 외부 입력을 $a \in A_n$ 라 하자($s(x,a)=x$). 기본 모드 원리에 의해 C가 고장 극복 교정 제어를 하는 동안 외부 입력은 변하지 않는다. 다시 말하면 고장 탐지부터 고장 극복 완료 시까지 외부 입력은 $v=a$ 가 계속 유지된다. 또 x 와 d 에 대해서 조건 (2)가 성립한다면 $s(s(x,d), t_d)=x$ 인 입력 스트링 $t_d:=u_1 u_2 \dots u_m$ 이 존재한다 ($|t_d|=m$). Σ 가 t_d 를 입력 받으면 $s(x,d)$ 에서 출발하여 $m-1$ 개의 안정 상태를 거치면서 x 에 도달한다. Σ 가 거치는 안정 상태를 z_1, \dots, z_{m-1} 라 하면 아래와 같은 관계를 얻는다.

$$\begin{aligned} s(z_{i-1}, u_i) &= z_i, i=1, 2, \dots, m \\ z_0 &:= s(x,d), z_m := x \end{aligned} \quad (4)$$

그런데 z_i 와 z_{i+1} 사이에도 여러 개의 과도 상태 $z_i^{(1)}, \dots, z_i^{(n)}$ 가 존재할 수 있다. 여기서 $n(i) \in \mathbb{N}$ 는 z_i 와 z_{i+1} 사이에서 Σ 가 거치는 과도 상태의 개수를 말한다. 이 성질을 일반화하여 Σ 가 어떤 안정 상태에서 주어진 입력 스트링에 반응하여 연쇄적 상태 천이를 할 때 거치는 모든 상태를 다음과 같은 함수 τ 로 표기하자(마지막 안정 상태 $z_m=x$ 는 제외).

$$\tau : X \times A_n^+ \rightarrow P(X)$$

$$\tau(s(x,d), t_d) := \{z_0, z_0^{(1)}, \dots, z_0^{(n(0))}, \dots, z_{m-1}, z_{m-1}^{(1)}, \dots, z_{m-1}^{(n(m-1))}\}$$

정적 제어기 C가 $s(x,d)$ 에서 x 까지의 교정 동작에 t_d 를 실제로 이용했다고 하자. 앞에서 외부 입력은 a 로 유지된다고 하였다. 그런데 $\tau(s(x,d), t_d)$ 에 속한 어떤 상태 x' 가 a 와 유효한 조합을 이룬다면 식 (3)에 의해서 제어 입력은 반드시 $C(x', a)=a$ 이어야 한다. 그런데 Σ 가 x' 를 거칠 때 나오는 제어 입력은 a 가 아니라 u_1, u_2, \dots, u_m 중의 하나이어야 하므로 식 (3)의 정의와 충돌이 생긴다. 동적 제어기는 t_d 의 입력 값을 한 개씩 생성할 때마다 새로운 상태로 이동하기 때문에 [6] 이러한 문제를 일으키지 않는다.

위에서 설명한 문제를 일반적으로 해결하기 위해서는 고장 극복 과정 중 Σ 가 거치는 상태 집합 $\tau(s(x,d), t_d)$ 가 고장 발생 시 가지는 외부 입력과 유효한 상태/입력 조합을 만들지 말아야 한다. d 는 Σ 가 안정 상태 x 에 있을 때 발생하기 때문에 d 의 발생 시 Σ 는 x 와 안정 조합을 만드는 임의의 외부 입력을 가질 수 있다. 이 입력 집합을 $U(x) \subset A_n$ 로 표시하면 $U(x)$ 는 다음과 같이 정의된다.

$$U(x) := \{a \in A_n \mid s(x, a) = x\}$$

조건 (2)에 덧붙여 $\tau(s(x,d), t_d)$ 와 $U(x)$ 를 이용하여 정적 내고장성 교정 제어기가 존재할 조건을 기술하면 다음과 같다.

조건 1: 정적 내고장성 교정 제어기 존재 조건

- $\forall d \in A_d \text{ s.t. } f(x,d), \exists t_d \in A_n^+ \text{ s.t.}$
- (i) $s(s(x,d), t_d) = x$
- (ii) $\forall x' \in \tau(s(x,d), t_d) \quad \forall a \in U(x), f(x', a) \neq \emptyset$

조건 1.(i)은 (2)와 같은 의미이며 조건 1.(ii)는 정적 교정 제어기를 설계하기 위해서 더 필요한 Σ 의 성질이다. 이와 같이 메모리를 쓰지 않는 간단한 구조의 정적 제어기가 존재하려면 동적 제어기에 비해 더 제한적인 존재 조건이 만족되어야 함을 알 수 있다.

조건 1의 (i)과 (ii)가 임의의 $x \in X$ 와 $d \in A_d$ 에 대해서 모두 성립한다면 그림 1의 정적 교정 제어기 C의 설계가 가능하다. 먼저 $f(x,v) \neq \emptyset$ 인 모든 유효 상태/입력 조합 (x,v) 에서는 이미 (3)의 설계 지침을 제시하였다. 고장 발생 시의 C의 동작도 앞에서 기술한 분석을 따른다. 원하지 않는 상태 천이 $x \rightarrow s(x,d)$ 에 대해 조건 1.(i)과 1.(ii)를 만족시키는 입력 스트링 $t_d:=u_1 u_2 \dots u_m$ 이 존재한다고 하였다. 또 Σ 는 x 에 도달하기 전까지 식 (4)에서 나타난 안정 상태 $z_0(=s(x,d)), z_1, \dots, z_m(=x)$ 을 거친다고 하였다. 현재 외부 입력이 $a \in U(x)$ 라면 C는 z_i 에서 아래와 같이 정의된다.

$$\begin{aligned}
 C(s(x,d), a) &= u_i \\
 C(z_i, a) &= u_{i+1}, i=1, \dots, m-1 \\
 C(z_m, a) &= u_{i+1}, j=1, \dots, n(i)
 \end{aligned} \tag{5}$$

위 식에서 z_1, \dots, z_m 는 앞에서 도입했듯이 z_i 와 z_{i+1} 사이에 존재하는 과도 상태이다. 조건 1.(ii)에 의해서 (5)에 나오는 모든 상태/입력 조합은 Σ 에서 유효하지 않으므로 (5)의 정의는 (4)와 상충하지 않는다. 식 (4)와 (5)에서 볼 수 있듯이 C의 설계는 간단한 함수 값 할당 과정으로서 동적 내고장성 교정 제어기의 설계[6]에 비해 매우 단순하다. 하지만 다음 장에서 나오듯이 정적 교정 제어기의 성능은 동적 제어기에 비해 차이가 없으며, 제어기를 만드는 데 필요한 회로의 크기는 더 줄어드는 장점을 보인다.

4. FPGA 실험: 비동기 오류 카운터

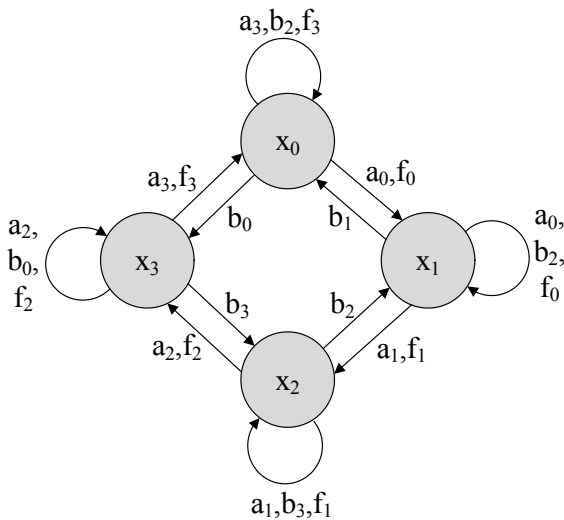


그림 2. 2-비트 비동기 오류 카운터
Fig. 2. 2bit asynchronous error counter

그림 2는 위성 메모리에서 발생하는 방사선 고장 중의 하나인 SEU(Single Event Upset) 고장 개수를 세기 위한 2-비트 비동기 오류 카운터(error counter)에 대한 상태 흐름도이다. 그런데 SEU 고장 개수를 세는 오류 카운터 역시 메모리 소자를 포함하기 때문에 방사선에 의한 SEU의 영향을 받는다. 상태 $x_i(0 \leq i \leq 3)$ 에서 오류 카운터를 한 스텝 전진시키는 정상 입력을 $a_i \in A_i$ 라고 정의하자. 예를 들어 상태 x_0 에서 a_0 가 입력되면 상태 x_1 로 카운터의 상태가 천이한다. 반면 상태 x_i 에서 발생하는 SEU 고장 입력 $b_i \in A_i$ 는 카운터를 한 스텝 후진시키며, SEU 고장 입력 $f_i \in A_i$ 는 한 스텝 전진시킨다. 그림 2는 정상 입력(a)과 고장 입력(b, f)을 포함하여 2-비트 오류 카운터의 전체 상태 천이를 나타낸 것이다.

모든 정상 입력과 고장 입력에 대해서 앞 절에서 제시된 정적 제어기의 존재 조건(조건 1)이 만족되므로 정적 교정 제어기를 설계할 수 있다. 상태 x_0 의 예를 들어 보자. x_0 에서 고장 입력 b_0 과 f_0 이 발생하면 오류 카운터는 각각 x_3 과 x_1 로 천이한다. 그런데 $s(x_3, a_3) = x_0$, $s(x_1, a_1, a_2, a_3) = x_0$ 이므로 조건 1의 (i)이 만족된다. $\tau(x_3, a_3) = \emptyset$ 이므로 b_0 에 대해서는 조건 1의 (ii)가 당연히 만족된다. 또한

$$\tau(x_1, a_1, a_2, a_3) = \{x_2, x_3\}$$

이고 $U(x_0) = \{a_3\}$ 이다. 그림 2를 보면 x_2 에서는 a_3 이 정의되지 않으며 x_3 에서는 $s(x_3, a_3) = x_0$ 이므로 내고장성 제어 경로와 a_3 이 만드는 상태 천이가 일치하다. 따라서 f_0 에 대해서도 조건 1의 (ii)가 성립한다. 다른 상태에 대해서도 동일하게 분석하면 모두 조건 1이 만족됨을 알 수 있다.

상태 x_i 에서 발생할 수 있는 고장 입력은 b_i 와 f_i 두 개이므로 고장 극복을 위한 교정 제어기도 각 상태별로 두 개가 필요하다. 각각의 고장 극복 교정 제어기를 C_{bi} 와 C_{fi} 라고 하자. 예를 들어 상태 x_1 에서 발생할 수 있는 고장 입력 b_1 에 대한 정적 교정 제어기 C_{b1} 을 설계하면 다음과 같다.

$$\begin{aligned}
 C_{b1}(x_0, a_0) &= a_0, \\
 C_{b1}(x_1, a_0) &= a_0
 \end{aligned}$$

또한 x_1 에서 일어나는 고장 입력 f_1 에 대한 정적 교정 제어기 C_{f1} 은 다음과 같다.

$$\begin{aligned}
 C_{f1}(x_2, a_0) &= a_2, \\
 C_{f1}(x_3, a_0) &= a_3, \\
 C_{f1}(x_0, a_0) &= a_0, \\
 C_{f1}(x_1, a_0) &= a_0
 \end{aligned}$$

식 (3)에서 주어진 대로 a_0 과 안정 상태를 이루는 x_1 에서는 $C_{b1}(x_1, a_0) = C_{f1}(x_1, a_0) = a_0$ 이 된다. 다른 고장 입력 $b_i, f_i(0 \leq i \leq 3)$ 에 대해서도 위 예제와 유사한 방법으로 정적 교정 제어기를 설계할 수 있다.

그림 3은 2-비트 비동기 오류 카운터와 본 논문에서 제시된 정적 교정 제어기, 그리고 고장 발생을 모사하기 위한 고장 주입기(fault injector)를 FPGA에서 함께 구현한 후 고장 복구 실험을 실시한 결과이다. 오류 카운터의 각 상태는 2-비트 값 $y_1 y_0$ 을 이용하여 $00(x_0)$, $01(x_1)$, $11(x_2)$, $10(x_3)$ 로 나타낸다. 오류 카운터는 초기에 $x_0(y_1 y_0 = 00)$ 에 머물러 있다가 시각 $t_1 = 68\text{ns}$ 에서 외부 입력 a_1 에 의해 $x_1(y_1 y_0 = 01)$ 로 천이한다. 실험에서 사용된

모든 외부 입력과 제어 입력은 상승 에지(edge)에서 트리거(trigger) 된다. 즉 입력이 0→1(상승 에지)로 된 후 다시 1→0로 하강하더라도 계속 입력이 활성화되어 있는 것으로 간주된다.

그림 3의 $t_2=160\text{ns}$ 에서 SEU 고장 f_1 이 발생하여 오류 카운터가 $x_2(y_1y_0=11)$ 로 천이한다. 고장 탐지기에 의해 f_1 고장이 탐지되면 정적 교정 제어기가 작동한다. $t_3=165\text{ns}$ 에서 최초 제어 입력 a_2 를 발생시켜 카운터의 상태를 $x_3(y_1y_0=10)$ 으로 이동시킨다. $t_4=170\text{ns}$ 에서 제어 입력 a_3 과 a_0 을 연이어 발생시켜 카운터의 상태를 $x_3(10) \rightarrow x_0(00) \rightarrow x_1(01)$ 로 이동시킨다. $t_5=178\text{ns}$ 에서 외부 입력 a_0 과 x_1 이 안정 상태를 이루므로 제어기는 a_0 을 발생시킨 이후 새로운 제어 입력을 생성하지 않고 고장 복구 과정을 종료한다. 그림 3을 보면 정적 교정 제어기에 의해 오류 카운터의 상태가 SEU 고장 f_1 이 발생하기 이전 상태 x_1 로 신속히 복구함을 볼 수 있다. 고장 복구에 사용된 제어 입력은 $a_2a_3a_0$ 이며, 실제로 고장 극복에 걸린 시간은 $t_5-t_2=18\text{ns}$ 로 매우 짧다.

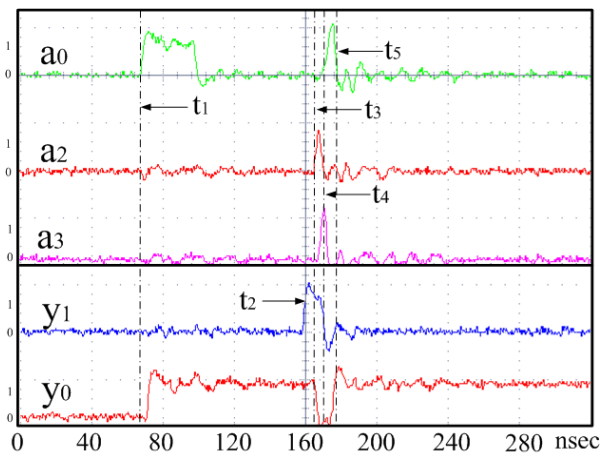


그림 3. 정적 제어기 고장 복구 실험 결과
Fig. 3. Fault recovery experiments by the static controller

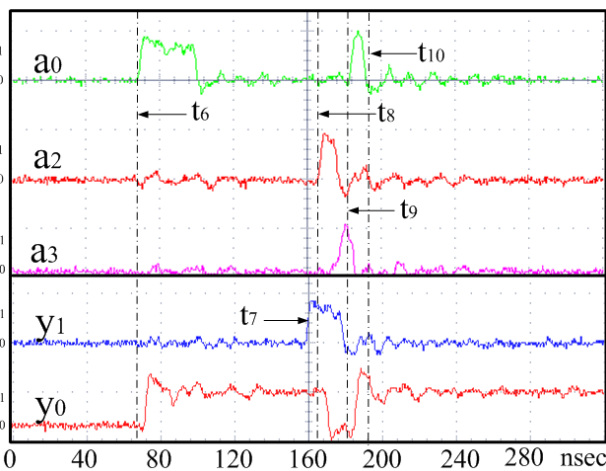


그림 4. 동적 제어기 고장 복구 실험 결과
Fig. 4. Fault recovery experiments by the dynamic controller

그림 4는 기존 결과와의 비교를 위해서 앞의 사례 연구에서 다른 고장을 똑같이 극복하는 동적 교정 제어기[3]를 설계한 후 수행한 실험 결과이다. $t_7=160\text{ns}$ 에서 고장이 발생하여 $t_{10}=192\text{ns}$ 에 고장 복구가 완료되었다. 따라서 고장 복구에 걸린 시간은 $t_{10}-t_7=32\text{ns}$ 이다. 동적 제어기는 상태 머신으로 구현되어야 하므로 제어 입력을 발생시키기 위해 제어기의 상태를 변화시키는 과정이 필요하다[3]. 하지만 정적 제어기는 조합회로만으로 구성되므로 제어기의 상태 천이 과정이 없어서 동적 제어기보다 더 신속히 제어 입력 신호를 발생시킬 수 있다. 따라서 고장 복구가 더 빠르게 진행되는 것을 볼 수 있다.

표 1은 논문에서 제안한 정적 교정 제어기와 동적 교정 제어기로 오류 카운터를 위한 교정 제어 시스템을 각각 구현한 후 로직 및 상태 사용량을 비교한 결과이다. 정적 제어기로 시스템을 구현하면 동적 제어기에 비해 로직 사용량이 대략 절반으로 줄었다. 이것은 제안된 정적 교정 제어기가 더 적은 로직 사용량으로 동적 제어기와 동일한 성능을 가진다는 사실을 입증한다.

표 1. 정적 제어기와 동적 제어기의 구현 자원 비교
Table 1. Implementation load comparison between static controllers and dynamic controllers

Number	Logic elements	States
Static controller	43	4
Dynamic controller	98	28

5. 결론

본 논문에서는 비동기 순차 회로에서 발생하는 상태 천이 고장의 영향을 극복하는 내고장성 교정 제어기를 제안하였다. 특히 기존의 동적 제어기에 비해 메모리를 사용하지 않는 정적 교정 제어기의 존재 조건을 규명하고 설계과정을 제안하였다. 제안된 기법을 SEU 고장의 영향을 받는 비동기 오류 카운터의 고장 탐지 및 극복 문제에 적용하였다. 정적 제어기는 동적 제어기보다 더 적은 로직으로 구현되고 고장 탐지 및 극복 제어의 속도 또한 동적 제어기보다 빠르다는 것을 실험적으로 입증하였다.

References

- [1] T. E. Murphy, X. Geng, and J. Hammer, "On the control of asynchronous machines with races," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 1073-1081, 2003.
- [2] J. Peng and J. Hammer, "Bursts and output feedback control of non-deterministic asynchronous sequential machines,"

- European Journal of Control*, vol. 18, no. 3, pp. 286-300, 2012.
- [3] S. W. Kwak and J.-M. Yang, "Fault diagnosis and tolerance for asynchronous counters with critical races caused by total ionizing dose in space," *Journal of Korean Institute of Intelligent Systems*, vol. 22, no. 1, pp. 49-55, 2012.
- [4] X. Xu and Y. Hong, "Matrix approach to model matching of asynchronous sequential machines," *IEEE Transactions on Automatic Control*, vol. 58, no. 11, pp. 2974-2979, 2013.
- [5] L. Sterpone and M. Violante, "Analysis of the robustness of the TMR-architecture in SRAM-based FPGAs," *IEEE Transactions on Nuclear Science*, vol. 53, no. 5, pp. 1545-1549, 2005.
- [6] J.-M. Yang and S. W. Kwak, "Corrective control for transient faults with application to configuration controllers," *IET Control Theory and Applications*, vol. 9, no. 8, pp. 1213-1220, 2015.
- [7] Z. Kohavi and N. K. Jha, *Switching and Finite Automata Theory*, 3rd ed., Cambridge UK: Cambridge University Press, 2010.

저자 소개



양정민(Jung-Min Yang)

1993년 : 한국과학기술원 전기및전자공학과 (공학사)

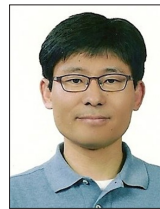
1995년 : 한국과학기술원 전기및전자공학과 (공학석사)

1999년 : 한국과학기술원 전기및전자공학과 (공학박사)

2013년~ 현재 : 경북대학교 전자공학부 교수

관심분야 : 비동기 머신 교정 제어, 실시간 시스템 고장 진단 및 극복, 불리언 제어 네트워크

E-mail : jmyang@ee.knu.ac.kr



곽성우(Seong Woo Kwak)

1993년 : 한국과학기술원 전기및전자공학과 (공학사)

1995년 : 한국과학기술원 전기및전자공학과 (공학석사)

2000년 : 한국과학기술원 전기및전자공학과 (공학박사)

2003년~ 현재 : 계명대학교 전자공학과 교수

관심분야 : 실시간 시스템, 교정제어, 우주용 시스템 설계, 무인 자율주행 제어

E-mail : ksw@kmu.ac.kr