

우주 방사능 누적에 의한 크리티컬 레이스가 존재하는 비동기 카운터를 위한 고장 탐지 및 극복

Fault Diagnosis and Tolerance for Asynchronous Counters with Critical Races Caused by Total Ionizing Dose in Space

곽성우* · 양정민***

Seong Woo Kwak and Jung-Min Yang

* 계명대학교 전자공학과

** 대구가톨릭대학교 전자공학과

요 약

전역 클럭 없이 외부 입력에 따라서 값이 변하는 비동기 카운터는 우주용 메모리 등 현대 디지털 시스템에서 널리 사용된다. 본 논문에서는 우주 방사능 누적에 기인하는 크리티컬 레이스 고장이 존재하는 비동기 카운터를 위한 고장 극복 기법을 제안한다. 크리티컬 레이스는 비동기 디지털 회로 설계 과정에서 발생하는 대표적인 고장으로서 회로의 다음 안정 상태가 고정되지 않고 여러 값 중 하나로 나오는 비결정적인 특성을 보인다. 이번 연구에서는 비동기 순차 머신에 대한 교정 제어 기법을 이용하여 크리티컬 레이스를 극복할 수 있는 상태 피드백 제어기의 설계 과정을 제시한다. 또한 비동기 카운터 교정 제어 시스템을 VHDL 코드로 구현하고 실험을 통하여 제안된 제어 시스템이 크리티컬 레이스 고장을 극복하는 과정을 예시한다.

키워드 : 비동기 카운터, 크리티컬 레이스, 방사능 누적, 교정 제어, 고장 극복.

Abstract

Asynchronous counters, where the counter value is changed not by a synchronizing clock but by outer inputs, are used in various modern digital systems such as spaceborne electronics. In this paper, we propose a scheme of fault tolerance for asynchronous counters with critical races caused by total ionizing dose (TID) in space. As a typical design flaw of asynchronous digital circuits, critical races cause an asynchronous circuit to show non-deterministic behavior, i.e., the next stable state of a state transition is not a fixed value but may be any value of a state set. Using the corrective control scheme for asynchronous sequential machines, this paper provides an existence condition and design procedure for a state feedback controller that can invalidate the effect of critical races. We implement the proposed control system in VHDL code and conduct experiments to demonstrate that the proposed control system can overcome critical races.

Key Words : Asynchronous counters, Critical races Total Ionizing Dose (TID), Corrective control, Fault tolerance.

1. 서 론

디지털 공학의 비약적인 발전 덕분에 디지털 시스템의 적용 환경이 심해, 원전, 우주 공간 등으로 점점 확대되었다. 예를 들어 예전에는 우주에서 사용되지 않았던 FPGA(Field Programmable Gate Array)가 인공위성, 우주 탐사선 등 우주 개발 기술에 많이 적용되고 있다. 특히 저비용, 설계 기간 단축 등의 요구 조건에 따라서 인공위성이 궤도에 진입한 후에도 회로를 재구성하거나 고장을 자가

복구할 수 있는 소자에 대한 관심이 급증하였다[1, 2].

비동기 카운터(asynchronous counter)는 전역 클럭(clock) 없이 외부 입력에 따라서 카운터 변수 값을 변경시키는 순차 회로(sequential machine)이다. 비동기 카운터는 동기 카운터에 비해서 반응 속도가 빠르고 특정 입력에 대한 카운팅 동작을 즉시 행할 수 있기 때문에 우주용 디지털 시스템 등에서 널리 사용되고 있다. 하지만 비동기 카운터는 비동기 순차 머신 설계 과정에서 발생할 수 있는 여러 가지 고장의 영향을 받을 수 있는 위험이 존재한다[3].

한편 우주 환경에서는 우주선(cosmic ray)과 태양에서 나오는 방사능이 전자회로에 많은 영향을 준다. 디지털 소자에서의 방사능 누적(total ionizing dose:TID)은 소자의 반응 속도를 느리게 하며 비동기 순차 회로에 과도 고장[4] 등, 여러 가지 문제를 일으킨다[5]. 크리티컬 레이스(critical race)는 반응 속도의 변화 때문에 비동기 회로에서 생기는 고장 중 가장 대표적인 예이다.

크리티컬 레이스는 비동기 머신의 다음 안정 상태(stable

접수일자: 2011년 6월 17일

심사(수정)일자: 2011년 9월 15일

게재확정일자 : 2011년 9월 30일

* 교신저자

이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임 (No. 2010-0021366).

state)가 하나로 고정되지 않고 여러 상태 값 중 임의의 한 값으로 나오는 비결정적인(non-deterministic) 동작을 의미한다. 비동기 머신의 설계 과정에서 미리 여분(redundant)의 상태 비트를 추가하여 크리티컬 레이스를 없앨 수 있다. 하지만 우주 공간에서 동작하는 디지털 회로가 우주선을 받으면 방사능 누적에 의해 설계 기술과 상관없이 크리티컬 레이스 고장을 야기할 가능성이 있다[6].

본 논문에서는 크리티컬 레이스가 존재하는 비동기 카운터가 레이스 영향에 관계없이 정상적인 동작을 가지게 하는 고장 극복 기법을 다룬다. 본 논문에서 제안하는 기법은 비동기 순차 머신에 대한 교정 제어(corrective control)에 기반을 두고 있다[7-9]. 교정 제어는 자동 제어의 원리를 이용하여 비동기 순차 머신의 안정 상태 동작을 보정하는 제어 기법이다. 교정 제어는 재설계를 하지 않고 비동기 머신의 동작을 바꿀 수 있기 때문에 비동기 머신의 고장 진단이나 고장 극복 능력을 구현하는 데 활용될 수 있다.

이번 연구에서는 [7]에서 제안된 크리티컬 레이스를 없애는 교정 제어 시스템을 바탕으로 비동기 카운터에 적용될 수 있는 제어기를 개발한다. 크리티컬 레이스가 야기하는 비결정적인 상태 천이(state transition) 결과에 관계없이 정상적인 동작을 유지하기 위해서는 카운터가 여분의 도달가능성(redundant reachability)을 가지고 있어야 한다. 비동기 카운터가 크리티컬 레이스를 탐지하고 고장 극복 교정 제어기가 존재할 필요충분조건을 만족시킨다는 사실을 증명하고 제어기의 설계 과정도 제시한다.

기존 연구 [7-9]와 비교하여 이번 연구가 가지는 기여도는 크리티컬 레이스의 발생과 고장 극복 제어기의 작동 과정을 실험적으로 보인다는 사실이다. 본 논문에서는 비동기 카운터와 교정 제어기가 결합된 폐루프 시스템(closed-loop system)을 VHDL 코드로 구현한다. 또한 VHDL 코드 상에서 크리티컬 레이스를 인위적으로 발생시키는 모듈을 개발하고 폐루프 시스템에 장착한다. 실험을 통하여 제안된 비동기 카운터 제어 시스템이 크리티컬 레이스 고장을 극복 하면서 정상적인 카운팅 동작을 보인다는 사실을 입증한다.

2. 모델링 및 문제 설정

2.1 크리티컬 레이스가 있는 비동기 카운터

n차 비동기 카운터를 Σ 라 하자. Σ 는 십진수 0에서 n-1까지를 $0 \rightarrow 1 \rightarrow \dots \rightarrow n-1 \rightarrow 0$ 의 순서로 순차적으로 카운팅한다. Σ 가 가지는 n개의 상태를 x_0, x_1, \dots, x_{n-1} 이라 정의한다. x_i 는 카운터 변수 i에 해당한다. Σ 를 상태 x_i 에서 x_{i+1} 로 천이시키는 입력을 a_i 라 하자($0 \leq i \leq n-1$). a_{n-1} 은 Σ 를 상태 x_{n-1} 에서 x_0 으로 천이시킨다. Σ 는 입력/상태 비동기 순차 머신이므로 다음과 같은 유한 상태 머신(finite state machine)으로 표현된다.

$$\Sigma = (A, X, x_0, f) \quad (1)$$

$A = \{a_0, a_1, \dots, a_{n-1}\}$ 는 카운터의 입력 집합, $X = \{x_0, x_1, \dots, x_{n-1}\}$ 은 상태 집합, x_0 은 초기상태, 그리고 $f: X \times A \rightarrow X$ 는 상태 천이 함수(state transition function)이다. 앞에서 정의한 입력과 상태 변수의 동작을 f를 이용하여 표시하면 다음과 같다.

$$\begin{aligned} f(x_i, a_i) &= x_{i+1} \\ f(x_{i+1}, a_i) &= x_{i+1} \end{aligned}$$

$$x_n := x_0, \forall i=0,1,\dots,n-1 \quad (2)$$

위 식은 비동기 카운터의 전형적인 특성을 보여준다. 전역 클럭이 존재하는 동기 카운터와 달리 비동기 카운터 Σ 는 외부 입력이 변화할 때만 카운팅 동작을 실시한다. Σ 가 안정 상태 x_i 에 머물러 있다고 가정한다면 이때의 입력은 a_{i-1} 이다. 즉 Σ 가 상태 x_{i-1} 에서 입력 a_{i-1} 을 받아서 x_i 로 한 스텝(step) 천이하였고, 입력이 바뀌지 않아서 현재 상태 x_i 에서 대기 중이다. 외부 입력이 a_{i-1} 에서 a_i 로 바뀌는 순간 Σ 는 한 스텝을 증가시키는 카운팅 동작을 하기 위해 다음 상태 x_{i+1} 로 천이하며, 외부 입력이 다시 바뀌기 전까지 x_{i+1} 에 머무른다. 이때 (x_i, a_{i-1}) 과 (x_{i+1}, a_i) 는 Σ 의 안정 조합(stable combination)이며 (x_i, a_i) 는 과도 조합(transient combination)이다. 비동기 순차 회로는 안정 조합에서 입력이 바뀌지 않는 한 현재 상태 값을 그대로 유지하며, 과도 조합에서는 현재 상태에 머무르지 못하고 순식간에 연쇄적인 상태 천이를 시작한다[9]. 또한 과도 조합 (x_i, a_i) 가 일으키는 상태 천이 $f(x_i, a_i) = x_{i+1}$ 의 진행 시간은 매우 짧으며 이론적으로 0이다.

비동기 순차 머신의 동작에서 과도 조합이 일으키는 상태 천이는 순식간에 벌어지므로 외부 사용자에게는 안정 상태 동작만 관측된다. Σ 의 안정 상태 동작만을 따로 표현하기 위해서 ‘stable-state 머신 Σ_{is} ’를 아래와 같이 정의한다[7].

$$\Sigma_{is} = (A, X, x_0, s)$$

위 식에서 f 대신 사용되는 ‘stable recursion 함수’ s는 다음과 같이 정의된다.

$$s: X \times A \rightarrow X, s(x, u) := x', (x, u) \in X \times A$$

x' 은 상태-입력 조합 (x, u) 의 다음 안정 상태(next stable state), 즉 현재 안정 상태 x에 있는 머신 Σ 에 입력 u가 들어왔을 때 머신이 도달하는 차기 안정 상태를 의미한다[7]. 예를 들어 안정 상태 x에서 입력 a가 들어와 $f(x, a) = y$, $f(y, a) = z$, $f(z, a) = z$ 와 같이 천이된다면 $s(x, a) = z$ 가 된다. 한편 식 (2)를 가지는 비동기 카운터에서 임의의 상태 천이 f는 모두 안정 상태 천이 s와 동일하므로 $\Sigma = \Sigma_{is}$ 이다.

단위 입력 대신 입력 스트링(string)을 s의 변수로 설정하면 다음과 같이 일반화할 수 있다.

$$s(x, ut) = s(s(x, u), t), x \in X, u \in A, t \in A^+$$

위 식에서 A^+ 는 A에 속한 원소로 이루어지는 길이 1 이상의 스트링 집합을 말한다.

비동기 카운터 Σ 에서 크리티컬 레이스를 일으키는 상태-입력 조합이 한 쌍 존재한다고 가정하고 (x_r, a_r) 이라 하자 ($0 \leq r \leq n-1$). 기존 연구 [10]과 유사한 방법으로 본 논문의 결과를 확장하면 크리티컬 레이스를 일으키는 상태-입력 조합이 여러 개 있는 문제도 다룰 수 있다. (x_r, a_r) 이 가지는 다음 안정 상태는 한 개로 고정되지 않고 m개의($m \geq 2$) 후보 상태 중 임의의 한 개로 결정되며, Σ 가 (x_r, a_r) 를 가질 때마다 그 결과는 계속 바뀌는 비결정적인 특성을 보인다. stable recursion 함수 s를 이용하여 (x_r, a_r) 의 크리티컬 레이스를 다음과 같이 표현한다.

$$s(x_r, a_r) = \{z_1, \dots, z_m\}, x_{r+1} \in s(x_r, a_r) \quad (3)$$

z_1, \dots, z_m 은 $s(x_r, a_r)$ 이 가질 수 있는 다음 안정 상태들이다. Σ 가 정상(normal) 동작을 한다면 다음 안정 상태는 x_{r+1} 이어야 하므로 $x_{r+1} \in s(x_r, a_r)$ 인 관계가 성립해야 한다. 또

z_1, \dots, z_m 이 다음 안정 상태들이므로 입력 a_r 과 안정 조합을 이룬다. 즉

$$s(z_i, a_r) = z_i, i=1, \dots, m$$

이다. (x_r, a_r) 에서 크리티컬 레이스가 발생하여 Σ 가 x_{r+1} 이 아닌 다른 상태 $z_i := x_q$ 로 천이하였다고 가정하자. Σ 에 고장 극복 모듈이 없다면 Σ 는 x_{r+1} 로 가지 못하고 x_q 에 머무르게 된다. 이때 외부 입력 a_q 가 들어온다면 Σ 는 x_{q+1} 로 천이하게 되며 결과적으로 정상적인 한 스텝 증가 카운팅 동작을 하지 못하는 고장 사건을 일으킨다.

2.2 교정 제어 시스템 구조

그림 1은 이번 연구에서 사용하는 비동기 순차 머신에 대한 교정 제어의 기본 구조를 나타낸다. Σ 는 제어 대상 비동기 머신, 즉 식 (1)의 비동기 카운터이며, C 는 역시 비동기 머신으로 구현되는 교정 제어기이다. $v \in A$ 는 외부 입력, $u \in A$ 는 C 가 만드는 제어 입력이며, $x \in X$ 는 Σ 로부터 나오는 상태 피드백 값이다. Σ_c 는 제어기 C 와 비동기 카운터 Σ 로 구성된 페루프 시스템을 가리킨다. 그림 1의 교정 제어기 C 는 다음과 같은 입력/출력 비동기 머신으로 표현된다.

$$C = (X \times A, A, \Xi, \xi_0, \phi, \eta) \quad (4)$$

$X \times A$ 는 C 의 입력 집합, A 는 C 의 출력 집합을 가리킨다. 또 Ξ 는 C 의 상태 집합, $\xi_0 \in \Xi$ 는 초기 상태이며, $\phi: \Xi \times X \times A \rightarrow \Xi$ 와 $\eta: \Xi \times X \times A \rightarrow A$ 는 각각 C 의 상태 천이 함수와 출력 함수이다.

제어기 C 는 상태 피드백 x 를 보고 비동기 카운터 Σ 에 크리티컬 레이스가 발생하는 순간을 관측한다. 그런 다음 외부 입력 v 가 다른 값으로 바뀌기 전에 적절한 제어 입력 스트링을 생성하여 Σ 가 정상적인 카운팅 동작을 취할 수 있게 제어한다. 이러한 제어 동작은 모두 비동기적으로 매우 빨리 이루어지기 때문에 외부 사용자에게는 관측되지 않으며, 페루프 시스템 Σ_c 는 크리티컬 레이스 고장 없이 정상적으로 움직이는 것처럼 보인다.

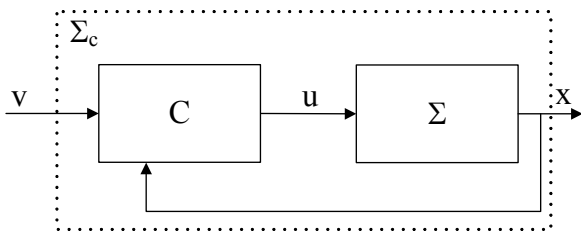


그림 1. 고장 극복 교정 제어 시스템
Fig. 1. Fault tolerant corrective control system

3. 고장 탐지 및 극복 교정 제어기

본 장에서는 [7, 9]에서 제안된 기법을 바탕으로 식 (1), (2)의 n -차 비동기 카운터가 크리티컬 레이스 발생에 상관 없이 정상적인 카운팅 동작을 할 수 있도록 해주는 교정 제어기가 존재함을 보인다.

교정 제어기 존재 조건을 기술하는 데 필요한 성질은 두 상태 사이의 ‘안정적 도달가능성(stable reachability)’[7]이다. x 와 y 가 비동기 머신 $\Sigma=(A, X, x_0, f)$ 가 가지는 임의의 두

상태라고 하자. 이때 아래 조건을 만족시키는 입력 스트링 $t := u_1 u_2 \dots u_k \in A^+$ 가 존재하면 상태 y 는 x 로부터 안정적으로 도달가능하다고 한다.

$$s(x, t) = y, t \in A^+$$

제어기는 입력 스트링 t 를 이용하여 안정 상태 x 에서 y 로의 교정 동작을 실시할 수 있다. 예를 들어 머신 Σ 가 안정 상태 x 에 있을 때 외부 입력 $b \in A$ 가 들어오면 $z \neq y$ 인 상태 z 로 천이하도록 정의되었다고 하자. 즉 $s(x, b) = z$ 라고 가정하자. 제어 목적은 그림 1의 페루프 시스템 Σ_c 가 안정 상태 x 에서 외부 입력 b 에 반응하여 z 대신 y 로 천이하도록 교정하는 일이다.

안정 상태 x 에서 머신 Σ 가 입력 스트링 $t = u_1 u_2 \dots u_k$ 를 받은 후 다음 안정 상태 y 까지 옮겨갈 때 거치는 중간 상태들을 y_1, y_2, \dots, y_{k-1} 이라 하면 아래 관계가 성립한다.

$$\begin{aligned} s(x, u_1) &= y_1 \\ s(y_1, u_2) &= y_2 \\ &\vdots \\ s(y_{k-1}, u_k) &= y \end{aligned}$$

그림 1의 교정 제어기 C 의 동작은 다음과 같다. 머신 Σ 가 안정 상태 x 로 천이하면 C 는 교정 동작을 준비한다. 이때 외부 입력이 $v = b$ 로 바뀌면 C 는 교정 동작을 실시한다. C 는 입력 b 대신 t 의 원소들을 순차적으로 Σ 에 전달한다. t 의 첫번째 원소 u_1 을 입력하면 Σ 는 $s(x, u_1) = y_1$ 로 천이한다. 상태 피드백 값이 y_1 로 바뀌는 순간 C 는 t 의 두번째 원소 u_2 를 입력시킨다. u_2 를 받은 Σ 도 역시 두번째 중간 상태 $s(y_1, u_2) = y_2$ 로 천이한다. t 의 마지막 원소 u_k 까지 이 동작이 실행되면 Σ 는 목적 상태 y 로 도달한다. 이러한 일련의 교정 제어 과정은 모두 클럭 없이 비동기적으로 이루어지므로 실행 속도는 매우 빠르다. 따라서 페루프 시스템 Σ_c 는 안정 상태 x 에서 외부 입력 b 를 받고 즉시 다음 안정 상태 y 로 천이하는 것처럼 관측된다. 즉 $s_c(x, b) = y$ 라는 제어 목적이 완성된다(s_c 는 Σ_c 의 stable recursion 함수이다).

위에서 설명한 교정 제어를 크리티컬 레이스가 존재하는 비동기 카운터에 적용시킨다. 먼저 식 (2)로부터 비동기 카운터의 임의의 상태는 다른 임의의 상태에서부터 모두 안정적으로 도달가능하다는 사실을 알 수 있다. 예를 들어 $0 < i < j \leq n-1$ 인 두 상태 x_i 와 x_j 를 생각하자. 식 (2)를 이용하여 Σ 가 x_i 에서 x_j 까지 안정적으로 도달가능 하도록 하는 입력 스트링을 다음과 같이 찾는다.

$$s(x_i, a_i a_{i+1} \dots a_{j-1}) = x_j \quad (5)$$

또한 x_j 에서 출발하여 x_i 까지 도달가능 하게 하는 입력 스트링도 식 (2)로부터 아래와 같이 존재한다.

$$s(x_j, a_j a_{j+1} \dots a_{n-1} a_0 \dots a_{i-1}) = x_i \quad (6)$$

비동기 카운터 Σ 는 임의의 두 상태간의 안정적 도달가능성을 모두 보장하므로 크리티컬 레이스를 없앨 수 있는 고장 극복 교정 제어기 C 를 꾸밀 수 있다.

크리티컬 레이스를 일으키는 상태-입력 조합 (x_r, a_r) 을 다시 생각하자. 레이스의 결과로 Σ 가 가질 수 있는 다음 안정 상태는 식 (3)에서 정의한 집합 $\{z_1, \dots, z_m\}$ 중의 임의의 원소 z_i 이다($1 \leq i \leq m$). $z_i = x_{r+1}$ 이면 카운터가 정상 동작한 결과가므로 아무런 교정 제어가 필요하지 않다. $z_i \neq x_{r+1}$ 이면 앞에서 설명했듯이 고장 극복 과정을 거치지 않고 후속 입력이

계속 들어온다면 비동기 카운터가 오류를 범하게 된다. 따라서 크리티컬 레이스를 없애기 위해서는 카운터 Σ 가 (x_r, a_r) 에서 (z_i, a_r) 로 천이한 직후 교정 제어기가 작동하여 Σ 를 안정 상태 z_i 에서 목적 상태 x_{r+1} 로 다시 천이시켜야 한다. z_i 에서 x_{r+1} 까지의 도달가능성은 식 (5)와 (6)이 보장하므로 임의의 크리티컬 레이스 결과에 대한 교정 제어기가 항상 존재한다.

고장 극복 교정 제어기 C의 구성과 동작은 다음과 같다.

- 1) 비동기 카운터 Σ 를 z_i 에서 x_{r+1} 로 천이시키는 교정 제어기를 C_i 라 하고 C_1, \dots, C_m 을 모두 설계한다.
- 2) C_1, \dots, C_m 을 join 연산[10] “ \vee ”로 결합하여 $C=C_1 \vee \dots \vee C_m$ 을 만든다.
- 3) C는 다음과 같이 동작한다.
 - 가) Σ 가 안정 조합 (x_r, a_{r-1}) 에 도달하면 C는 교정 동작을 준비한다. 하지만 크리티컬 레이스 발생 전이므로 제어 동작 없이 외부 입력 v 를 그대로 제어 입력 u 로 전달한다.
 - 나) 외부 입력 $v=a_r$ 가 들어와 Σ 가 과도 조합 (x_r, a_r) 에서 천이하면 C는 상태 피드백 x 를 통해 다음 상태를 관측한다.
 - 다) Σ 의 다음 안정 상태가 x_{r+1} 로 관측되면 Σ 가 정상 동작한 것이므로 C는 아무 제어 동작을 실행하지 않는다.
 - 라) Σ 의 다음 안정 상태가 $z_i \neq x_{r+1}$ 인 z_i 로 관측되면 C는 C_i 부분을 구동하여 페루프 시스템을 z_i 에서 x_{r+1} 로 이동시켜 크리티컬 레이스를 극복한다.

교정 제어기의 상세한 설계 및 구동 과정은 다음 장 사례 연구에서 기술한다. 식 (7)의 교정 제어기는 C_1, \dots, C_m 가 합쳐진 것이므로 상태 집합이 커질 가능성이 있다. 하지만 비동기 머신의 상태를 줄이는 일반적인 기법[9]을 사용하거나 한 상태를 중복 이용하여 C_1, \dots, C_m 의 동작을 함께 구현하는 방법[11] 등을 활용하면 C의 상태 집합의 크기를 줄일 수 있다.

4. FPGA 실험

4.1 5차 비동기 카운터

본 논문에서 제안한 크리티컬 레이스 극복을 위한 교정 제어기의 활용성을 입증하기 위해서 VHDL을 이용하여 실험을 실시한다. 사례 연구로서 5차 비동기 카운터를 구현한다. 그림 2는 5차 비동기 카운터 $\Sigma=(A, X, x_0, f)$ 의 상태 흐름도(state flow diagram)이다.

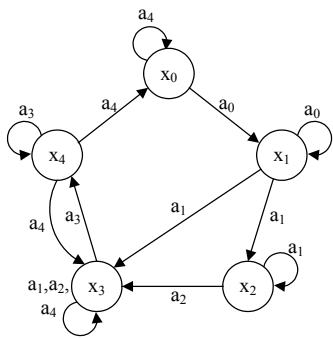


그림 2. 크리티컬 레이스가 있는 5차 비동기 카운터
Fig. 2. 5-step asynchronous counter with critical race

카운터 Σ 의 입력 집합은 $A=\{a_0, a_1, a_2, a_3, a_4\}$ 이며 상태 집합은 $X=\{x_0, x_1, x_2, x_3, x_4\}$ 이다. 또 그림에 나온 대로 Σ 는 과도 조합 (x_1, a_1) 과 (x_4, a_4) 에서 크리티컬 레이스를 가진다. (x_1, a_1) 과 (x_4, a_4) 로부터 Σ 가 다다를 수 있는 다음 안정 상태는 각각 $\{x_2, x_3\}$ 과 $\{x_0, x_3\}$ 중 하나이며 카운터의 작동 시마다 그 값이 계속 바뀐다. 서론에서 논했듯이 크리티컬 레이스의 발생은 우주 방사능 누적의 원인으로 비동기 카운터 상태를 구성하는 상태 비트가 동시에 변하지 않고 불규칙한 시간 지연(time delay)을 가지는 데 기인할 수 있다. Σ 의 스텝(step)이 다섯 개 있으므로 VHDL 코드에서 각 상태는 세 비트(bit)로 표현한다. 본 실험에서는 $x_0=000$, $x_1=011$, $x_2=101$, $x_3=001$, $x_4=111$ 로 상태 비트를 정의한다. 그런데 Σ 가 x_1 에서 입력 a_1 을 받아서 x_2 로 천이할 때에는 $011 \rightarrow 101$ 로서 상위 상태 비트 두 개가 동시에 바뀌어야 한다. 만약 불규칙한 시간 지연이 발생하여 최상위 비트가 중간 비트보다 더 늦게 그 값이 변한다면 Σ 는 $011 \rightarrow 001$ 의 상태 천이를 순간 경험한다. 따라서 $x_2=101$ 대신에 $x_3=001$ 을 다음 안정 상태로 가지게 되어 레이스가 일어날 수 있다. 이번 실험에서는 (x_1, a_1) 의 다음 안정 상태가 x_2 와 x_3 중 무작위로 임의의 값이 나오도록 설정해주었다. (x_4, a_4) 에서 일어나는 크리티컬 레이스도 비슷하게 해석된다.

(x_1, a_1) 을 예로 들어 교정 제어기 설계 과정을 설명한다. (x_1, a_1) 의 다음 안정 상태로 x_2 가 나오면 정상적인 카운팅 동작과 일치하므로 고장 극복 모듈의 구동이 필요 없다. 반면 (x_1, a_1) 의 다음 안정 상태가 x_3 이라면 Σ 를 x_3 에서 x_2 로 보내는 교정 제어 동작을 구동해야 한다. 식 (6)과 그림 2로부터 Σ 를 x_3 에서 x_2 로 안정적으로 도달하게 하는 입력 스트링 $t \in A^*$ 를 찾으면 $t=a_3 a_4 a_0 a_1$ 이다. 또 Σ 가 안정 상태 x_3 에서 t 를 입력받은 후 x_2 까지 갈 때 거치는 중간 상태는 x_4, x_0, x_1 이다. t 를 이용하여 크리티컬 레이스를 극복하는 식 (4)의 교정 제어기 C_1 을 다음과 같이 설계한다.

초기 상태 ξ_0 에 있던 C_1 은 Σ 가 안정 상태 x_1 에 진입하면 상태 $\xi_1 \in \Xi$ 로 이동하여 크리티컬 레이스의 발생에 대비한다. 그림 2에서 x_1 이 안정 상태일 때 외부 입력은 a_0 이다. 크리티컬 레이스가 아직 일어나지 않았으므로 C_1 은 외부 입력 v 를 그대로 비동기 카운터 Σ 에 전달한다. 따라서 ξ_0 에서 C_1 의 상태 천이 함수 ϕ 와 출력 함수 η 는 아래와 같이 정의된다.

$$\begin{aligned} \phi(\xi_0, x, v) &= \xi_0, \quad \forall (x, v) \in X \times A \setminus \{(x_1, a_0)\} \\ \phi(\xi_0, x_1, a_0) &= \xi_1 \\ \phi(\xi_1, x_1, a_0) &= \xi_1 \\ \eta(\xi_0, x, v) &= v, \quad \forall (x, v) \in X \times A \end{aligned} \tag{7}$$

상태 ξ_1 에서 C_1 은 크리티컬 레이스의 발생을 기다린다. 외부 입력이 a_0 에서 a_1 로 바뀐 후 상태 피드백 값이 x_2 로 나온다면 정상적인 카운팅 동작이 실행되었다는 의미이므로 C_1 은 다시 초기 상태 ξ_0 으로 돌아간다. 또 상태 ξ_1 에서 C_1 은 계속 외부 입력 v 를 변경 없이 제어 입력 u 로 그대로 전달(relay)해준다.

$$\begin{aligned} \phi(\xi_1, x_1, a_1) &= \xi_1 \\ \phi(\xi_0, x_2, a_1) &= \xi_0 \\ \eta(\xi_1, x, v) &= v, \quad \forall (x, v) \in X \times A \end{aligned} \tag{8}$$

C_1 이 상태 ξ_1 에 있을 때 외부 입력이 a_0 에서 a_1 로 바뀐 후 상태 피드백 값 x_3 을 받았다고 하자. C_1 은 크리티컬 레이스의 발생을 인식하고 위에서 정의한 입력 스트링 t 의 원소들

을 차례차례 제어 입력으로 Σ 에 전달하여 교정 동작을 실행한다. $|t|=4$ 이므로 이 과정에서 C_1 의 보조 상태(auxiliary state) 네 개가 필요하다[11]. C_1 의 보조 상태를 $\xi_2, \xi_3, \xi_4, \xi_5$ 라 정의한다.

크리티컬 레이스가 일어난 후 C_1 이 받는 입력은 (x_3, a_1) 이다. 상태 ξ_1 에서 입력 (x_3, a_1) 을 받은 C_1 은 상태 ξ_2 로 천이한 후 t 의 첫번째 원소 a_3 을 제어 입력으로 출력한다. ξ_2 로 천이한 후에도 C_1 이 가지는 외부 입력은 a_1 로 유지된다.

$$\begin{aligned} \phi(\xi_1, x_3, a_1) &= \xi_2 \\ \phi(\xi_2, x_3, a_1) &= \xi_2 \\ \eta(\xi_2, x_3, a_1) &= a_3 \end{aligned} \quad (9)$$

C_1 로부터 제어 입력 a_3 을 받은 비동기 카운터 Σ 는 첫번째 중간 상태 x_4 로 천이한다. 또 x_4 를 상태 피드백으로 받은 C_1 은 ξ_2 에서 ξ_3 으로 옮겨간 후 t 의 두번째 원소 a_4 를 제어 입력으로 낸다. 즉 상태 ξ_2 에서 C_1 의 ϕ 와 η 는 다음과 같이 정의된다.

$$\begin{aligned} \phi(\xi_2, x_4, a_1) &= \xi_3 \\ \phi(\xi_3, x_4, a_1) &= \xi_3 \\ \eta(\xi_3, x_4, a_1) &= a_4 \end{aligned} \quad (10)$$

이런 식으로 일련의 피드백과 제어 입력 적용 과정이 두 번 더 실행되면 비동기 카운터 Σ 는 최종 목적 상태 x_2 에 도달하고 C_1 은 초기 상태 ξ_0 으로 복귀한다. 지금까지 정의한 C_1 의 모든 교정 동작은 전역 클럭 없이 비동기적으로 순식간에 이루어진다. 그러므로 외부 사용자에게는 그림 1의 페루프 시스템 Σ_c 가 과도 조합 (x_1, a_1) 에서 크리티컬 레이스가 발생해도 정상적으로 다음 안정 상태 x_2 로 천이하는 카운팅 동작을 수행하는 모습만이 관측될 것이다.

한편 (x_4, a_4) 에서 발생하는 크리티컬 레이스를 없애는 제어기를 C_2 라 하자. (x_4, a_4) 에서 발생하는 크리티컬 레이스를 없애기 위해서는 레이스 발생 후 카운터가 이동하는 상태 x_3 에서 출발하여 정상 상태 x_0 까지 도달가능하게 하는 입력 스트링을 찾아야 한다. 식 (6)과 그림 2로부터 이 입력 스트링을 찾으면 $t'=a_3a_4$ 이다. t' 를 이용한 C_2 의 교정 제어 동작은 (x_1, a_1) 에 대한 제어기 C_1 의 설계 과정과 유사하다.

4.2 실험 결과

그림2의 비동기 카운터와 본 연구에서 제안한 교정 제어기를 VHDL로 구성한 후 Altera Cyclone FPGA로 구현하여 실험을 실시하였다. 그림 3은 $s(x_1, a_1)=x_3$ 레이스가 발생했을 때 제어기 C_1 이 제어 입력 $t=a_3a_4a_0a_1$ 을 전달해서 카운터 Σ 가 x_4, x_0, x_1 을 거쳐 목적 상태 x_2 로 도달하는 실험 결과를 보여준다. 그림에서 $y_2y_1y_0$ 은 카운터의 상태 비트를 나타낸다. 그림 3(a)는 실험 결과를 100nsec 스케일(scale)로 표시한 그림이며 그림 3(b)는 40nsec 스케일로 나타낸 것이다. 그림 3(b)에서는 또한 교정 제어에서 제어 입력으로 사용된 a_0, a_1, a_3, a_4 신호의 발생을 오실로스코프로 측정하여 나타내었다.

시각 $t=400\text{nsec}$ 에서 카운터 Σ 가 $x_1(=011)$ 에 있을 때 a_1 이 입력되었다. 하지만 카운터는 크리티컬 레이스에 의해 목적 상태 x_2 가 아닌 $x_3(=001)$ 로 천이한다. 이때 그림 3(b)의 시각 t_1 에서 제어기 C_1 이 고장을 탐지하고 제어 입력 a_3 을 발생시킨다. 시각 t_1 직후 입력 a_3 신호가 발생하는 것을 관측할 수 있으며 이는 제어기 C_1 이 작동하기 시작했다는 사실을 의미한다. a_3 에 의해 카운터의 상태는 $x_4(=111)$ 로 천

이하며, C_1 은 시각 t_2 에 이를 인식하고 다음 제어 입력 a_4 를 투입하여 카운터를 $x_0(=000)$ 로 천이하게 한다. 제어기 C_1 은 다시 시각 t_3 에서 제어 입력 a_0 를 발생시켜 카운터는 $x_1(=011)$ 로 천이한다. 마지막으로 시각 t_4 에 C_1 에 의해 다시 입력 a_1 이 투입됨으로써 카운터가 목적 상태인 $x_2(=101)$ 에 도달하고 고장 극복 제어 과정은 종료된다. 그림 3(b)는 제어 입력 스트링 $t=a_3a_4a_0a_1$ 의 단위 입력이 순차적으로 발생하여 카운터를 목적상태 $x_2(=101)$ 로 도달하게 하는 과정을 잘 보여준다.

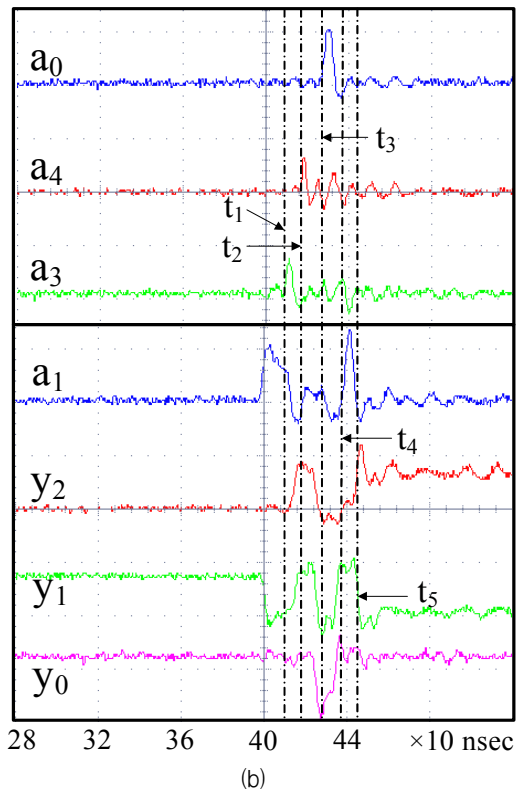
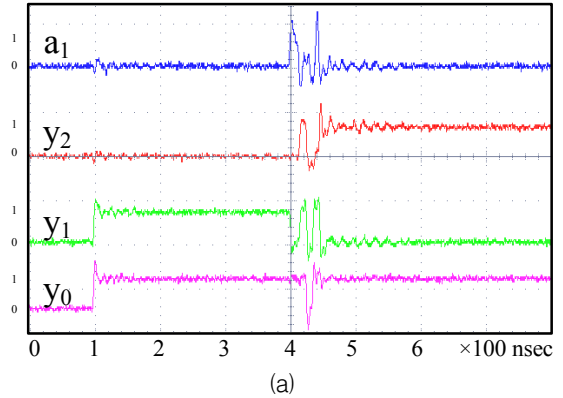


그림 3. $s(x_1, a_1)=x_3$ 레이스가 발생한 경우의 고장 극복 교정 제어 실험 결과: (a) 100nsec 스케일, (b) 40 nsec 스케일

Fig. 3. Experimental result of corrective control for the critical race of $s(x_1, a_1)=x_3$: (a) 100nsec scale and (b) 40nsec scale

제어기 C₁이 고장을 인지하고 제어 입력을 투입하는 과정은 비동기적으로 순식간에 일어난다. 그림 3의 실험 결과에서 제어기 C₁이 4개의 제어 입력($t=a_3a_4a_0a_1$)을 투입하는데 걸린 시간은 $t_5-t_1=448-408=40nsec$ 이다. 따라서 1개의 제어 입력을 내보내는 시간은 10nsec정도이며, 이는 수 개의 논리 게이트 전파시간(gate delay)에 해당하는 시간이다. 만약 전파시간이 없다면 이상적으로 “0”시간에 제어가 완성된다.

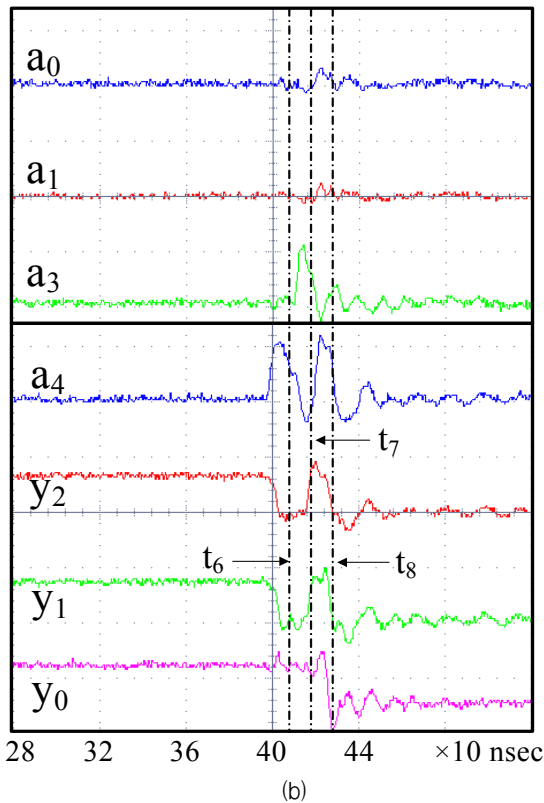
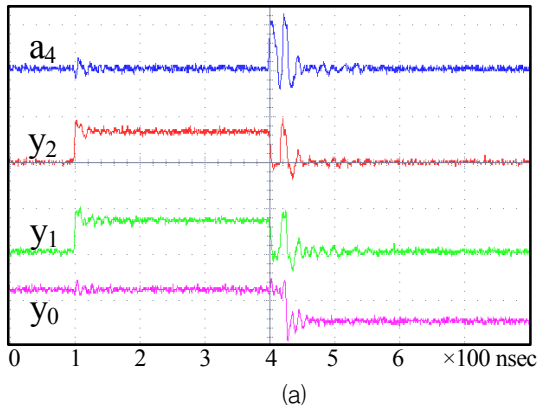


그림 4. $s(x_4,a_4)=x_3$ 레이스가 발생한 경우의 고장 극복 교정 제어 실험 결과: (a) 100nsec 스케일, (b) 40 nsec 스케일

Fig. 4. Experimental result of corrective control for the critical race of $s(x_4,a_4)=x_3$: (a) 100nsec scale and (b) 40nsec scale

그림 4는 $s(x_4,a_4)=x_3$ 레이스가 발생했을 때 제어기 C₂가 제어 입력 $t'=a_3a_4$ 를 전달해서 카운터 Σ 를 x_4 를 거쳐서 목적 상태 x_0 로 보내는 실험 결과를 보인 것이다. 카운터가 $x_4(=111)$ 에 있을 때 a_4 가 입력되었으나 크리티컬 레이스에 의해 $x_3(=001)$ 로 천이한다. 이때 C₂가 고장을 인지(시각 t_6)한 후 제어 입력 a_3a_4 를 투입시켜 카운터가 목적 상태 x_0 (시각 t_8)에 도달하도록 교정한다. 2개의 제어 입력 a_3a_4 를 내보내는 데 걸린 시간은 $t_8-t_6=430-410=20nsec$ 으로 앞의 실험과 마찬가지로 페루프 시스템의 비동기성을 보장할 만큼 충분히 짧다.

5. 결론

본 논문에서는 크리티컬 레이스가 존재하는 비동기 카운터의 고장 극복을 위한 제어 기법을 제안하였다. 크리티컬 레이스는 우주 방사능의 누적으로 인해 디지털 회로의 반응 시간이 달라짐으로 인해 발생할 수 있다. 본 논문에서 사용한 제어 기법은 비동기 순차 머신의 안정 상태 동작을 보장하는 교정 제어이다. 모델 정합(model matching), 무한 순환(infinite cycle) 제거 등의 목적으로 사용되었던 교정 제어기는 크리티컬 레이스를 없애는 고장 극복 모듈로도 적절하게 이용할 수 있음을 보였다. 비동기 카운터의 특성상 발생하는 크리티컬 레이스를 모두 제거할 수 있는 상태 피드백 교정 제어기가 항상 존재한다. 이번 논문에서는 특히 비동기 카운터에서 일어나는 크리티컬 레이스의 발생과 교정 제어기의 작동 등을 VHDL 코드로 구현하고 실험적으로 입증하였다.

참고 문헌

- [1] C. Carmichael, E. Fuller, P. Blain, and M. Caffrey, "SEU mitigation techniques for Virtex FPGAs in space applications," *Proceedings of MAPLD99*, 1999.
- [2] T. Vladimirova, "Reconfigurable system-on-a-chip based platform for satellite on-board computing," *Scientific Conference "Space, Ecology, Safety" with International Participation (SES 2005)*, pp. 10-13, 2005.
- [3] S. H. Unger, "Hazards, critical races, and metastability," *IEEE Transactions on Computers*, vol. 44, no. 6, pp. 754-768, 1995.
- [4] 광성우, 양정민, "확률 최적화를 이용한 비선점형 Rate Monotonic 스케줄링의 체크포인트 구간 결정," *한국지능시스템학회 논문지*, 제21권 1호, pp. 120-127, 2011.
- [5] R. Koga, J. George, G. Swift, C. Yui, L. Edmonds, C. Carmichael, T. Langley, P. Murray, K. Lanes, and M. Napier, "Comparison of Xilinx Virtex-II FPGA SEE sensitivities to protons and heavy Ions," *IEEE Transactions on Nuclear Science*, vol. 51, no. 5, pp. 2825-2833, 2004.
- [6] 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.
- [7] 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.
- [8] J.-M. Yang, "State feedback control of asynchronous machines with nondeterministic models," *IEEE Transactions on Automatic Control*, vol. 54, no. 5, pp. 1072-1076, 2009.
- [9] Z. Kohavi, *Switching and Finite Automata Theory* (2nd ed.), New York: McGraw-Hill, 1978.
- [10] J. Peng and J. Hammer, "Input/output control of asynchronous sequential machines with races," *International Journal of Control*, vol. 83, no. 1, pp. 125-144, 2010.
- [11] J.-M. Yang, "Corrective control of asynchronous sequential machines in the presence of adversarial input," *IET Control Theory and Applications*, vol. 2, no. 8, pp. 706 - 716, 2008.

저 자 소 개



곽성우(Seong-Woo Kwak)

1993년 : 한국과학기술원 전기및전자공학과 (공학사)
 1995년 : 한국과학기술원 전기및전자공학과 (공학석사)
 2000년 : 한국과학기술원 전기및전자공학과 (공학박사)
 2003년~현재 : 계명대학교 전자공학과 부교수

관심분야 : 실시간 시스템 고장 진단 및 극복, 우주용 디지털시스템 설계, 비동기 머신 교정 제어

E-mail : ksw@kmu.ac.kr



양정민(Jung-Min Yang)

1993년 : 한국과학기술원 전기및전자공학과 (공학사)
 1995년 : 한국과학기술원 전기및전자공학과 (공학석사)
 1999년 : 한국과학기술원 전기및전자공학과 (공학박사)
 2001년~현재 : 대구가톨릭대학교 전자공학과 부교수

관심분야 : 비동기 머신 교정 제어, 실시간 시스템 고장 진단 및 극복, 보행로봇 걸음새 연구

E-mail : jmyang@cu.ac.kr