

제어 불능 입력이 존재하는 비동기 순차 머신의 상태 피드백 제어 및 오류 카운터로의 응용

State Feedback Control of Asynchronous Sequential Machines with Uncontrollable Inputs: Application to Error Counters

양 정 민, 곽 성 우*
(Jung-Min Yang and Seong Woo Kwak)

Abstract: The model matching problem of asynchronous sequential machines is to design a corrective controller such that the stable-state behavior of the closed-loop system matches that of a prescribed model. In this paper, we address model matching when the external input set consists of controllable inputs and uncontrollable ones. Like in the frame of supervisory control of Discrete-Event Systems (DES), uncontrollable inputs cannot be disabled and must be transmitted to the plant without any change. We postulate necessary and sufficient conditions for the existence of a corrective controller that solves model matching despite the influence of uncontrollable events. Whenever a controller exists, the algorithm for its design is outlined. To illustrate the physical meaning of the proposed problem, the closed-loop system of an asynchronous machine with the proposed control scheme is implemented in VHDL code.

Keywords: asynchronous sequential machines, corrective control, model matching, uncontrollable inputs

I. 서론

비동기 순차 머신(asynchronous sequential machine)은 전역 클럭(clock) 없이 입력의 변화에 따라서만 동작이 결정되는 순차 머신이다. 비동기 순차 머신은 동기 순차 머신(synchronous sequential machine)에 존재하는 clock-skew, 전력 과다 소비, worst-case timing 등의 문제점이 없으며 무엇보다 동작 속도가 빠르기 때문에 디지털 시스템이나 통신 프로토콜 등의 분야에서 여전히 널리 사용되고 있다[1,2].

비동기 머신에 대한 교정 제어(corrective control)는 비동기 머신의 약점으로 지적되어 온 복잡한 재설계 과정을 해결하기 위해서 제안된 방법이다[3-5]. 교정 제어는 비동기 순차 머신을 제어 대상으로 보고 전통적인 피드백 제어(feedback control) 기법을 이용하여 폐루프(closed-loop) 시스템의 동작을 보상한다. 비동기 머신 앞에 설치된 제어기는 외부 입력(external input)과 출력 피드백을 받아서 제어 입력(control input)을 생성하여 비동기 머신으로 보냄으로써 머신이 나타내는 올바르지 않은 동작을 교정한다. 이미 설계된 비동기 순차 머신의 안정 상태 동작(stable-state behavior)을 올바르게 동작하는 기준 모델(reference model)의 동작과 일치시킨다는 의미로 교정 제어의 결과를 모델 정합(model matching)이라 부르기도 한다[6,7].

본 논문의 목적은 제어 불능 입력(uncontrollable inputs)이 존재하는 비동기 순차 머신을 위한 교정 제어기를 제안하는 것이다. 이전 연구들에서[3-5] 머신이 받는 외부 입력은 모두 제어 가능하다고 가정하였다. 즉 제어기가 외부 입력 값을 임의로 변경하여 제어 입력을 생성한 후 머신으로 보

낼 수 있다고 가정하였다. 하지만 값을 절대 변경하지 못하고 그대로 시스템으로 전달되어야 하는 입력 집합은 이산 사건 시스템(discrete-event systems)[8]에서 항상 존재할 수 있다. 생산 시스템에서의 고장 및 복구 사건[9], 실시간 시스템(real-time system) 태스크(task)의 시작(invoke) 및 완료(finish) 동작[10] 등이 제어 불능 입력의 대표적인 예이다.

본 논문에서는 먼저 비동기 머신을 위한 교정 제어 원리를 간략하게 기술한 후 제어 불능 입력이 존재하는 머신을 위한 교정 제어 시스템 구조를 제안한다(II 장). 그런 다음 주어진 기준 모델에 대한 모델 정합 문제를 풀 수 있는 교정 제어기가 존재할 조건 및 제어기 설계 과정을 제시한다(III 장). 또한 사례 연구로서 비동기 오류 카운터(asynchronous error counter)를 VHDL 모의 코드로 작성한다(IV 장). 제안된 교정 제어기는 제어 불능 입력의 영향에도 불구하고 정확한 오류 카운팅을 할 수 있는 폐루프 시스템을 구현한다.

II. 모델 정합 조건

1. 제어 불능 입력에 대한 모델 정합

비동기 머신이 과도 상태(transient state)에 머무르는 시간은 극히 짧으며(이론적으로는 0) 외부 사용자는 머신의 안정 상태, 즉 외부 입력의 변화가 없으면 천이가 일어나지 않는 상태만 관측할 수 있다. 교정 제어는 비동기 머신의 이러한 성질을 이용하여 폐루프 시스템의 안정 상태 동작을 바꾼다.

교정 제어의 원리를 설명하기 위해서 제어 대상 비동기 머신이 그림 1과 같이 안정 상태 x 에 있다고 가정하자. 이때 외부 입력이 a 에서 b 로 바뀌면 머신은 다음 안정 상태 x_1 로 천이한다. (x 와 x_1 사이에는 과도 상태들이 존재할 수 있지만 앞서 기술했듯이 외부 사용자에게는 감지되지 않는

* 책임저자(Corresponding Author)

논문접수: 2009. 7. 22., 수정: 2009. 8. 11., 채택확정: 2009. 8. 12.

양정민: 대구가톨릭대학교 전자공학과(jmyang@cu.ac.kr)

곽성우: 계명대학교 전자공학과(ksw@kmu.ac.kr)

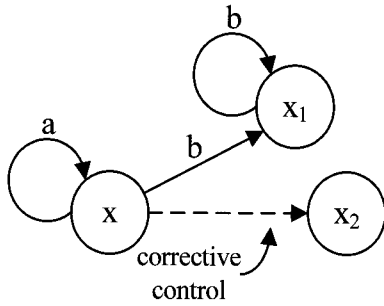


그림 1. 교정 제어의 원리.
Fig. 1. Principle of corrective control.

다.) 그런데 기준 모델은 안정 상태 x 에서 외부 입력이 b 를 받으면 x_2 로 천이한다고 가정하자. 따라서 모델 정합을 위해서는 폐루프 시스템이 상태 x 에서 입력 b 에 반응하여 x_1 대신 x_2 로 천이하도록 만들어주어야 한다(교정 제어에 관한 더 자세한 기술은 [3-6] 참조).

모델 정합을 위한 교정 제어기가 존재하기 위해서는 제어 대상 비동기 머신에서 x_2 가 x 로부터 ‘안정적으로 도달가능(stably reachable)’해야 한다[3]. x_2 가 x 로부터 안정적으로 도달가능하다는 것은 머신이 x 에서 출발하여 다른 안정 상태들을 거쳐서 x_2 에 도달하도록 하는 입력 스트링(string)이 존재한다는 뜻이다. 교정 제어기는 안정 상태 x 에서 외부 입력이 b 로 바뀌는 순간 b 를 머신에 전달하는 대신 입력 스트링의 변수를 하나씩 제어 입력으로 보냄으로써 폐루프 시스템의 동작을 바꾼다. 교정 제어기 역시 비동기 머신이므로 교정 제어 과정은 클럭 없이 순식간에 일어나며 따라서 외부 사용자에게는 외부 입력 b 에 대해서 머신이 x_2 로 즉시 천이하는 동작만이 관측된다.

제어 불능 입력이 존재할 때 모델 정합에 필요한 조건을 나타내기 위해서 그림 2와 같은 상황을 가정하자. 그림 1과 마찬가지로 비동기 머신은 안정 상태 x 에 머물러 있다. 이때 제어 불능 입력 c 가 외부 입력으로 들어온다고 하자. c 는 제어 불능이므로 교정 제어기는 일단 c 를 그대로 머신으로 전달해야 한다. 입력 c 가 들어와서 천이되는 다음 안정상태가 x_1 이라면 비동기 머신의 상태는 x 에서 x_1 로 바뀐다. (기준 모델은 입력 c 에 대해서 x_2 로 천이한다.) 하지만 x_1 에서 x_2 까지 도달가능하다고 한다면 머신이 x_1 에 다다른 순간 교정 제어기가 작동하여 머신을 다시 x_2 로 보낼 수

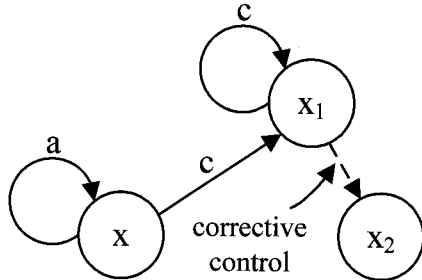


그림 2. 제어 불능 입력에 대한 교정 제어 원리.
Fig. 2. Principle of corrective control considering the uncontrollable input.

있다. 비동기적으로 실행되는 교정 동작 때문에 외부 사용자지는 머신이 x_1 을 잠깐 거쳤다는 사실을 인지하지 못한다. 즉 제어 대상 비동기 머신이 x_1 에서 x_2 까지 도달가능성을 보유한다면 모델 정합이 가능하다. 본 시스템은 클럭 없는 비동기로 작동되기 때문에 기본 모드 동작(fundamental mode operation) 원리[11]를 만족시킨다. 따라서 머신이 x_1 에 도달하는 순간 외부 입력이 추가로 변하기 전에 제어 동작을 바로 실행시키는 메커니즘을 구현할 수 있다.

그런데 제어 불능 입력이 존재하는 비동기 머신의 도달가능성은 제어 가능 입력만으로 이루어진 입력 스트링이 두 상태 사이에서 존재할 때만 유효하다. 제어 불능 입력은 그 특성상 제어기나 관리자(supervisor)가 생성하지 못하는 값이므로[8] 교정 제어에 사용될 수 없다.

2. 제어기 존재 조건

제어 불능 입력을 가지는 비동기 머신의 모델 정합을 실현하는 교정 제어기가 존재할 조건을 수학적으로 표현하면 다음과 같다. 먼저 본 논문에서는 머신의 출력이 현재 상태와 동일한 입력/상태 비동기 머신에 대해서 논하기로 한다. 입력/상태 머신 Σ 을 유한 상태 머신(finite state machine)으로 표현하면 아래와 같다[3].

$$\Sigma = (A, X, X, x_0, f, h) \tag{1}$$

위 식에서 A 는 입력 집합, X 는 상태 집합(출력 집합과 동일하므로 두 번 표기됨), x_0 은 초기 상태이다. 제어 가능 입력을 A_c , 제어 불능 입력 집합을 A_u 라고 하면 입력 집합은 $A=A_c \cup A_u$ 로 나뉜다. $f: X \times A \rightarrow X$ 는 상태 천이 함수이며 출력 함수 $h: X \rightarrow X$ 는 항등 함수(identity function)이다. Σ 의 안정 상태 동작을 표현하기 위해서 ‘stable-state 머신 Σ_s ’를 아래와 같이 정의한다[3,11].

$$\Sigma_s = (A, X, X, x_0, s, h) \tag{2}$$

위 식에서 상태 천이 함수 f 대신 사용되는 ‘stable recursion 함수’ s 는 다음과 같이 정의된다.

$$s: X \times A \rightarrow X, s(x,u) = x', x, x' \in X, u \in A \tag{3}$$

x' 은 상태와 입력 조합 (x,u) 의 다음 안정 상태, 즉 초기 안정 상태 x 에 있던 머신 Σ 에 입력 u 가 들어왔을 때 머신이 도달하는 차기 안정 상태를 의미한다[3,11]. 예를 들어 안정 상태 x 에서 입력 a 가 들어와 $f(x,a)=x_1$, $f(x_1,a)=x_2$, $f(x_2,a)=x_2$ 와 같이 천이된다면 $s(x,a)=x_2$ 가 된다. 즉 s 는 외부에 인식되지 않는 과도 상태 x_1 을 제거하고 안정 상태 사이의 천이만 기록한다.

단위 입력 대신 입력 스트링을 s 의 변수로 설정하면 다음과 같이 일반화할 수 있다.

$$s(x,ut) = s(s(x,u),t), x \in X, u \in A, t \in A^+ \tag{4}$$

위 식에서 A^+ 는 A 에 속한 알파벳으로 이루어지는 길이 1 이상의 스트링 집합을 말한다. $s(x,t)=x'$ 인 입력 스트링 $t \in A^+$ 가 존재하면 상태 x' 는 상태 x 로부터 안정적으로 도달가능하다[3,11]. 그런데 앞서 기술했듯이 머신에 제어 불능 입력이 존재할 때에는 제어 가능 입력만으로 도달가능성을 표현해야 한다. 본 논문에서는 상태 x 와 x' 가 아래 조건을

만족할 때 x' 는 x 로부터 '제어 가능 입력만으로 안정적으로 도달가능하다'라고 정의한다.

$$\exists t \in A_c^+ \text{ s.t. } s(x,t)=x' \quad (5)$$

t 는 제어 가능 입력 집합 A_c 의 원소만으로 이루어진 스트링이다.

Σ 가 추종해야 하는 기준 모델을 $\Lambda := (A, X, X, x_0, s', h)$ 이라고 정의한다. s' 은 Λ 의 stable recursion 함수이다. 모델 정합은 안정 상태 동작만 비교하면 되므로 기준 모델 자체를 stable-state 머신으로 정의해도 일반성을 잃지 않는다.

모델 정합의 조건은 상태 천이를 일으키는 외부 입력이 제어 가능한지 제어 불가능인지에 따라서 다르게 표현된다. Σ 와 Λ 가 어떤 상태-입력 조합 $(x,a) \in X \times A$ 에서 모델 부정합이 생긴다고 가정하자. 즉 Σ 에서는 $s(x,a)=y$, Λ 에서는 $s'(x,a)=z(\neq y)$ 로 천이가 일어난다.

i) $a \in A_c$.

a 가 제어 가능 입력이라면 그림 1과 같이 기존의 교정 제어 기법을 사용하여 모델 정합을 이룰 수 있다. 즉 머신이 x 에 있을 때 입력 a 를 받는 즉시 y 대신 z 로 상태 천이가 일어나게 교정 동작이 실행된다. 제어기 존재 조건을 도달가능성으로 기술하면 다음과 같다.

조건 1: Λ 에서 z 가 x 로부터 안정적으로 도달가능 $\Rightarrow \Sigma$ 에서 z 가 x 로부터 제어 가능 입력만으로 안정적으로 도달가능, $\forall x, z \in X$.

모델 Λ 가 x 에서 z 까지 안정적으로 도달가능하다는 사실은 Λ 을 x 에서 z 로 천이시키는 (a 와 같은) 외부 입력이 존재함을 의미한다. 조건 1의 의미는 이 경우 머신 Σ 에서도 항상 x 에서 z 까지의 도달가능성이 보장되므로 교정 제어기를 설계할 수 있다는 뜻이다. 물론 제어 불가능 입력은 제어 입력으로 쓸 수 없기 때문에 제어 가능 입력만으로 도달가능성이 보장되어야 한다(식 (5) 참조).

ii) $a \in A_u$.

a 가 제어 불가능 입력이라면 그림 2에서 예시된 바대로 Σ 에 의한 상태 천이가 끝난 직후 교정 제어기가 작동되어야 한다. $s(x,a)=y$ 이므로 모델 정합을 이루려면 Σ 의 상태 y 에서 z 까지의 도달가능성이 보장되어야 한다. 이 조건을 일반적으로 표현하면 다음과 같다.

조건 2: Λ 에서 z 가 x 로부터 제어 불가능 입력 a 에 의해 안정적으로 도달가능 $\Rightarrow \Sigma$ 에서 z 가 $s(x,a)$ 로부터 제어 가능 입력만으로 안정적으로 도달가능, $\forall x, z \in X, a \in A_u$.

III. 페루프 시스템

그림 3은 제어 불가능 입력이 존재하는 입력/상태 비동기 머신을 위한 교정 제어 페루프 시스템이다. Σ 는 제어 대상 비동기 머신이며 C 는 역시 비동기 머신으로 구현되는 교정 제어기이다. $v \in A = (A_c \cup A_u)$ 는 외부 입력, $u \in A_c$ 는 제어기가 만드는 제어 입력이며, $x \in X$ 는 머신 Σ 의 상태 출력이다. 그림 3에서 표기된 것처럼 외부 입력 중 A_u 에 속하는 신호는 제어 불가능이므로 값이 변경되지 않고 그대로 Σ 로 전달된다. 그림 3에서 Σ_c 는 C 와 Σ 가 결합된 페루프 시스템이다. C 와 Σ 가 비동기 머신이므로 Σ_c 도 비동기 머신의 특성

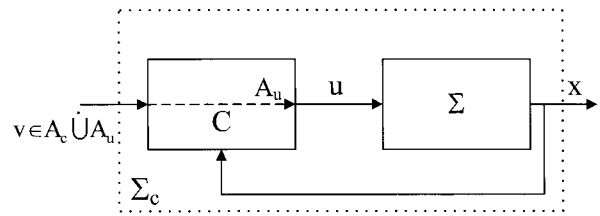


그림 3. 제어 불가능 입력이 존재하는 입력/상태 비동기 머신을 위한 교정 제어 시스템.

Fig. 3. Corrective control system for input/state asynchronous machines with the uncontrollable input set.

을 유지한다.

제어기 C 를 유한 상태 머신으로 표현하면 아래와 같다 [3-5,7].

$$C = (A \times X, A, \Xi, \xi_0, \Phi, \eta) \quad (6)$$

Ξ 는 C 의 상태 집합, $\xi_0 \in \Xi$ 는 초기 상태이며, Φ 와 η 는 제어기의 상태 천이 함수와 출력 함수를 각각 가리킨다. C 는 외부 입력 v 와 상태 피드백 x 를 동시에 입력으로 받고 출력으로 제어 입력 u 를 생성하므로 C 의 입력과 출력 집합은 각각 $A \times X, A$ 로 정의된다.

제어 대상 비동기 머신 Σ 가 기준 모델 Λ 에 대해서 조건 1과 2를 모두 만족시킨다면 모델 정합을 이룩하는 제어기 C 가 존재한다. 모델 부정합이 일어나는 한 쌍의 상태에 대한 교정 제어기 동작을 예로 들어서 설명한다. 앞서 기술한 바대로 상태-입력 조합 (x,a) 에서 $\Sigma: s(x,a)=y$, $\Lambda: s'(x,a)=z(\neq y)$ 와 같은 모델 부정합이 존재한다고 가정하자. 모델 정합을 위해서는 안정 상태 x 에 있던 Σ 가 입력 a 를 받으면 $s(x,a)=y$ 에 도달하는 순간 다시 원하는 상태 z 로 천이되어야 한다. 조건 2에 의해서 z 는 y 로부터 제어 가능 입력만으로 도달가능하다. 다시 말하면 식 (5)에 의해서 $s(y,t)=z$ 가 되는 입력 스트링 $t=v_1v_2 \dots v_k \in A_c$ 가 존재한다. t 를 입력했을 때 Σ 가 거치는 중간 상태를 y_1, y_2, \dots, y_{k-1} 이라 하면 다음과 같은 관계가 성립한다.

$$s(y_{i-1}, v_i)=y_i \quad (y_0=y, y_k=z), \quad i=1, \dots, k \quad (7)$$

제어기 C 의 상태 집합 Ξ 는 기본적으로 초기 상태 ξ_0 를 포함한다. 또한 길이가 k 인 입력 스트링으로 교정 동작을 이루기 위해서는 k 개의 보조 상태 $\xi_1, \xi_2, \dots, \xi_k$ 를 추가로 정의해야 한다[4].

초기 상태 ξ_0 에 있던 C 는 상태 피드백으로 y 를 받은 순간 첫 번째 보조 상태 ξ_1 로 천이한다. ξ_1 에서 C 는 t 의 첫 번째 입력 문자 v_1 을 제어 입력으로 Σ 에 전달한다. 식 (7)에 의해서 v_1 을 받은 Σ 는 첫 번째 중간 상태 y_1 로 천이한다. y_1 을 다시 상태 피드백을 받은 C 는(그림 3 참조) 다음 상태 ξ_2 로 천이한 후 두 번째 제어 입력 알파벳 v_2 를 생성한다. 이 과정을 반복하여 Σ 가 목적 상태 z 에 도달하면 모델 정합이 완성된다. 이러한 일련의 교정 동작을 위한 상태 천이는 비동기적으로 아주 빠르게 실행되므로 외부 사용자에게는 안정 상태간의 천이만 관측된다. 그림 4는 이러한 교정 동작을 도식화한 모습이다.

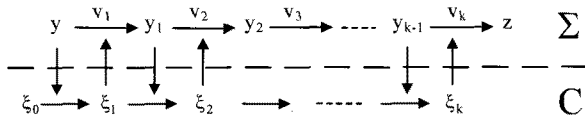


그림 4. 상태 y에서 z로의 교정 동작.
Fig. 4. Corrective action from state y to state z.

IV. VHDL 실험

1. 사례 연구 시스템: 비동기 오류 카운터

본 장에서는 제어 불능 입력이 존재하는 비동기 오류 카운터를 사례 연구 시스템으로 설정하고 제안된 제어기를 VHDL 코드로 구현하는 실험을 실시한다. 오류 카운터는 오류 탐지 및 복구(EDAC: Error Detection and Correction) 회로에서 실제 발생한 오류의 개수를 세는 데 주로 사용된다[12,13].

EDAC 회로로 많이 사용되는 코드 중의 하나인 (7,4) Hamming 코드의 예를 들어 오류 카운터의 역할을 설명하면 다음과 같다. (7,4) Hamming 코드에서는 4 비트(bit)의 데이터 비트와 3 비트의 여분 비트(parity bit)로 구성된 7 비트를 하나의 워드(word)로 정의한다. 이 코드는 워드 한 개에서 발생하는 1 비트의 오류를 탐지하고 자동으로 복구할 수 있으나 2 비트의 오류는 탐지만 할 수 있다. (7,4) Hamming 코드로 이루어진 EDAC 회로는 메모리에 데이터를 쓸 때 여분 비트를 발생시키는 인코더(encoder), 데이터와 여분 비트를 읽어와 오류를 탐지하고 복구하는 디코더(decoder)로 구성된다. 디코더에는 일반적으로 오류 발생을 알려주는 디지털 출력이 있는데 이 출력은 대부분 1 비트의 오류 발생과 2 비트의 오류 발생 여부를 각각 알려주는 신호 라인 두 개로 구성된다. 디코더의 출력을 이용하면 지금까지 일어난 오류의 횟수를 셀 수 있으며, 이와 같은 목적으로 제작된 카운터가 바로 오류 카운터이다[13,14].

(7,4) Hamming 코드 기반 EDAC 회로용으로 제작된 기존 오류 카운터의 문제점은 1 비트와 2 비트 오류를 구분하지 못하고 오류가 발생하면 무조건 카운터의 값이 1 증가하도록 설계되었다는 점이다[15]. 다시 말하면 2 비트 오류가 일어났을 경우 실제 오류 비트 발생 횟수보다 하나 적은 값이 카운터에 들어간다. EDAC를 사용하는 우주용 메모

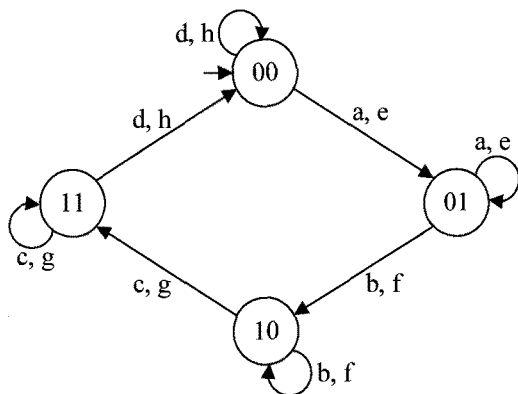


그림 5. 제어 불능 입력이 존재하는 비동기 오류 카운터 Σ.
Fig. 5. Asynchronous error counter Σ with uncontrollable inputs.

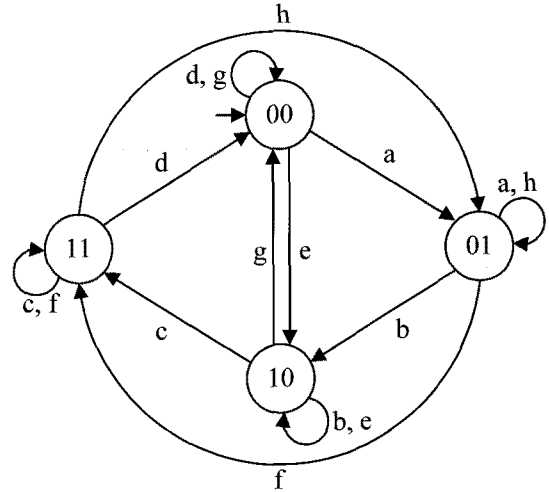


그림 6. 비동기 오류 카운터의 기준 모델 Δ.
Fig. 6. Prescribed model Δ of the asynchronous error counter.

리의 경우 주기적으로 메모리 스크립(memory scrubbing) 과정을 통하여 SEU (Single Event Upset) 등과 같은 오류가 누적되지 않도록 하고 있다[15]. 일반적으로 비동기 오류 카운터에 의해 측정된 SEU 발생율을 기반으로 메모리 스크립 주기를 결정한다. 그런데 오류 카운터에 잘못된 오류 발생 횟수가 저장되면 EDAC 회로의 복구 한계를 초과하는 오류가 축적되는 현상을 메모리가 감지하지 못하게 되므로 메모리 스크립 주기 설정에 영향을 주어 저장된 데이터의 신뢰성을 담보하는 데 문제가 발생할 수 있다.

본 사례 연구에서는 1 비트와 2 비트 오류를 구분하지 못하는 4단 비동기 오류 카운터를 제어 대상 머신으로 설정한다. 그림 5는 비동기 오류 카운터 Σ의 상태 천이도이다. Σ의 상태 집합은 $X = \{00, 01, 10, 11\}$ 이며 초기 상태는 $x_0 = 00$ 으로 설정되었다. Σ의 제어 가능 입력 집합은 $A_c = \{a, b, c, d\}$ 이고 제어 불능 입력 집합은 $A_u = \{e, f, g, h\}$ 이다. 제어 가능 입력 집합의 원소는 1 비트 오류 발생, 제어 불능 입력 집합의 원소는 2 비트 오류 발생 사건을 각각 의미한다.

예를 들어 카운터 Σ가 안정 상태 00에 머물러 있다고 가정하자. 이때 외부로부터 입력 a가 들어오면 카운터는 1 비트 오류 발생을 감지하고 다음 상태 01로 천이한다. 즉 카운터 값이 1 증가한다. 그런데 안정 상태 01에서 다시 2 비트 오류 f가 발생했다고 하자. 발생한 비트의 개수가 2이므로 카운터 값은 2 증가하여 상태 11로 천이되어야 올바르게 된다. 하지만 앞에서 가정했듯이 머신 Σ은 1 비트와 2 비트 오류를 구분하지 못하므로 그림 5에 나타난 바대로 f를 받으면 카운터 값이 여전히 1씩 증가하여 상태 10으로 천이한다.

그림 6은 페루프 시스템이 따라가야 할 기준 모델 Δ의 상태 천이도이다. 그림 5와 달리 제어 불능 입력이 들어오면 카운터 값이 2씩 증가하도록 올바르게 정의되어 있다. 본 논문에서 제안하는 제어 기법을 이용하여 페루프 시스템이 안정 상태에서 Δ와 동일한 동작을 보이도록 카운터 Σ를 교정한다.

2. 제어기 설계

식 (6)의 교정 제어기 $C=(A \times X, A, \varepsilon, \xi_0, \phi, \eta)$ 를 설계하는 일은 C 가 가지는 모든 상태-입력 조합에 대해서 상태 천이 함수 ϕ 와 출력 함수 η 를 각각 정의하는 작업이다. 그림 5와 그림 6을 비교하면 제어 불능 입력에 대해서 카운터가 2씩 증가하도록 페루프 시스템을 꾸미면 모델 정합이 완성된다는 사실을 알 수 있다. 즉 기준 모델 Λ 와 머신 Σ 사이의 모델 부정합은 다음과 같다.

$$\begin{aligned} s(00,e) = 01 &:: s'(00,e) = 10 \\ s(01,f) = 10 &:: s'(10,f) = 11 \\ s(10,g) = 11 &:: s'(10,g) = 00 \\ s(11,e) = 00 &:: s'(11,h) = 01 \end{aligned} \tag{8}$$

위의 모델 부정합은 모두 제어 불능 입력 때문에 생긴다. 따라서 모델 정합을 이룩하는 교정 제어기가 존재할 조건은 앞 장에서 정의된 조건 2와 동일하다. 그림 5와 6을 보면 머신 Σ 와 모델 Λ 의 동작이 모두 조건 2를 만족함을 알 수 있다. 예를 들어 Λ 에서 상태 10은 상태 00으로부터 제어 불능 입력 e 에 의해서 안정적으로 도달가능하다. 그런데 머신 Σ 에서도 상태 10은 $s(00,e)$, 즉 상태 01로부터 제어 가능 입력만으로 안정적으로 도달가능하다($s(01,b) = 10$). 나머지 제어 불능 입력에 의한 상태 천이에서도 이와 같은 관계가 모두 성립하므로 조건 2에 의해서 모델 정합을 이룩하는 교정 제어기가 존재한다.

모델 정합 교정 제어기는 식 (8)에서 기술된 각각의 모델 부정합을 해결하는 제어기 모듈을 구성한 후 모든 모듈을 통합함으로써 완성된다. 본 절에서는 위 모델 부정합 중 제어 불능 입력 e 에 대한 모델 정합을 이루는 교정 제어 모듈의 동작 과정을 예시한다. 다른 모듈의 설계 과정은 다음에 설명하는 e 에 대한 교정 제어 모듈 설계와 유사하므로 생략한다.

먼저 초기 상태 ξ_0 에 있던 제어기 C 는 Σ 가 상태 00와 안정 조합을 이룰 때 transition 상태 $\xi_0(x) \in E$ 로 천이한다. 비동기 머신에서 입력 변화는 항상 머신이 안정 상태에 있을 때만 가능하므로[11] Σ 가 상태 00와 안정 조합을 이루는 순간 제어기는 초기 상태 ξ_0 와 구별되는 $\xi_0(x)$ 로 천이함으로써 제어 불능 입력 e 에 대처할 수 있는 자세를 가지게 된다. 그림 5에서 00과 안정 조합을 이루는 정상 입력 집합은 $\{d\}$ 이므로 C 의 상태 천이 함수 ϕ 는 아래와 같이 정의된다.

$$\begin{aligned} \phi(\xi_0(x,v)) &= \xi_0 \quad \forall (x,v) \neq (00,d) \\ \phi(\xi_0(00,d)) &= \xi_0(x) \end{aligned} \tag{9}$$

모델 부정합을 야기하는 입력 e 가 들어오기 전까지 제어기는 아무 교정 동작을 실행하지 않으므로 초기 상태 ξ_0 와 transition 상태 $\xi_0(x)$ 에서 C 는 출력 함수 값으로 외부 입력을 그대로 전달해준다.

$$\begin{aligned} \eta(\xi_0(x,v)) &= v \quad \forall (x,v) \in X \times A \\ \eta(\xi_0(x), (x,v)) &= v \quad \forall (x,v) \in X \times A \end{aligned} \tag{10}$$

제어 불능 입력 e 가 들어오면 제어기 C 는 e 에 의한 상태 천이가 끝나기를 기다린 후 Σ 가 다음 안정 상태로 진입했

음을 감지하는 순간부터 제어 입력을 생성한다. 그림 5에서 $s(00,e)=01$ 이며 머신 Σ 을 01에서 원하는 상태 10으로 다시 천이시키는 제어 입력 스트링은 'b'임을 관찰할 수 있다. 입력 스트링의 길이가 1이므로 모델 정합을 구현하기 위해서 C 는 한 개의 보조 상태 $\xi_1 \in E$ 를 다시 정의한다. 한편 상태 $\xi_0(x)$ 에서 $v \neq e$ 인 외부 입력 v 가 들어오면 제어기 C 는 초기 상태 ξ_0 로 복귀한다. $\xi_0(x)$ 에서 ξ_1 로 가는 제어기의 동작을 구현하면 아래와 같다.

$$\begin{aligned} \phi(\xi_0(x), (00,e)) &= \xi_0(x) \\ \phi(\xi_0(x), (00,v)) &= \xi_0 \quad \forall v \neq e \\ \phi(\xi_0(x), (01,e)) &= \xi_1 \\ \phi(\xi_1, (01,e)) &= \xi_1 \\ \eta(\xi_1, (x,v)) &= b \quad \forall (x,v) \in X \times A \end{aligned} \tag{11}$$

위 식에서 정의된 제어기 동작은 그림 4에서 도시한 일반적인 교정 동작과 동일한 메커니즘으로 작동된다. 제어기 C 로부터 제어 입력 b 를 받은 머신 Σ 는 최종적으로 상태 10으로 천이된다. 따라서 페루프 시스템은 입력 e 를 받아서 상태 01 대신 10으로 천이되는 동작을 보이므로 모델 Λ 과의 정합이 완성된다.

3. VHDL 실험 결과

식 (9)-(11)의 동작을 가지는 교정 제어 모듈과 그 외 다른 모델 부정합을 해결하는 제어 모듈이 모두 장착된 비동기 4단 오류 카운터를 VHDL 코드를 이용하여 구현하였다. 본 실험에서는 Altera의 QUARTUS® II(ver.7.1)를 VHDL 컴파일러 및 합성기로 사용하였으며, 페루프 시스템이 설계된 target FPGA는 Altera의 EP1C6Q240C8이다.

구현한 페루프 시스템이 정상 입력에 대한 상태 천이 동작을 제대로 보이는지 먼저 확인하기 위해서 외부 입력 a, b, c, d 를 연속으로 입력시키고 난 후 나오는 실험 결과를 그림 7에 나타내었다. 실험에서 모든 입력 신호는 펄스(pulse) 형태로 들어온다. 즉 어떤 입력에 대한 펄스가 생긴 후 펄스 값이 다시 0으로 떨어져도 다른 입력 펄스가 들어오기 전까지는 현재의 입력이 머신에 유효한 것으로 가정한다. 그림 7에서 볼 수 있듯이 오류 카운터는 해당 입력에 대해서 각각 다음 상태 01, 10, 11, 00으로 정상적으로 천이한다. 입력이 들어온 후 상태 천이되기까지 걸리는 시간 지연(time delay)은 5ns 미만으로 아주 작은 값이다 즉 구현된 실험 환경은 클릭 없이 동작하는 비동기 머신의 빠른 응답 특성을 잘 모사했다고 말할 수 있다.

그림 8은 제어 불능 입력이 들어왔을 때 교정 제어기가 작동하는 과정을 실험으로 보인 모습이다. 안정 상태 00에 있던 오류 카운터는 $t=29.444ns$ 에서 제어 불능 입력 e 를 받는다. e 가 제어 불능이므로 교정 제어기는 카운터가 e 에 의해서 다음 안정 상태 01에 도달하기를 일단 기다린다. 카운터의 상태가 01이 된 직후 제어기는 식 (11)에 따라서 제어 입력 b 를 생성한다. 그림 8에서 'b_ctl'은 교정 제어기가 만드는 '제어 입력 신호 b'라는 의미이다. b 를 받은 카운터는 $t=43.883ns$ 에서 원하는 상태 10으로 천이되어 모델 정합을 이룩한다. 그림 8에서 제어 불능 입력 e 가 발생하여 페루프 시스템이 다음 안정 상태 01로 천이한 후부터 원하는 상태

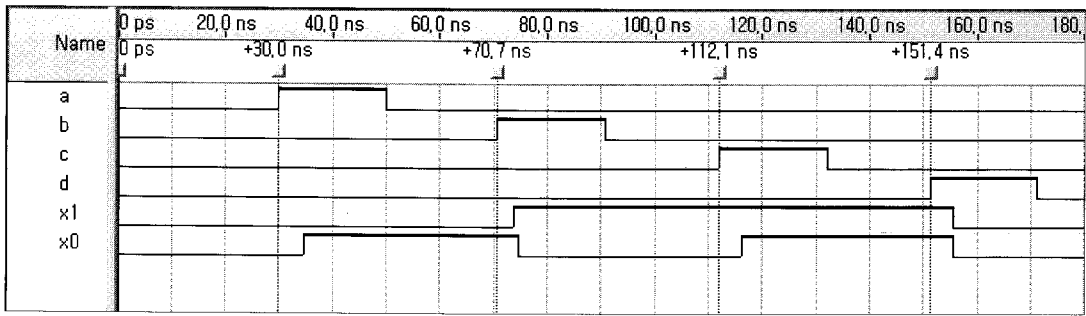


그림 7. VHDL 실험 결과: 제어 가능 입력에 의한 정상적인 상태 천이.

Fig. 7. VHDL experiment result: normal state transitions by controllable input characters.

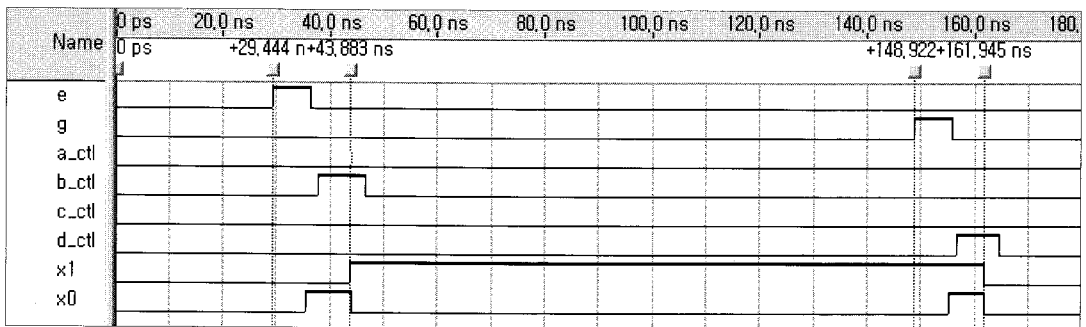


그림 8. VHDL 실험 결과: 제어 불능 입력이 발생했을 때의 모델 정합.

Fig. 8. VHDL experiment result: model matching against uncontrollable input characters.

10으로 다시 천이하기까지 걸리는 시간은 약 8ns로 관측된다. 이 정도의 시간 지연은 여전히 외부 사용자가 감지할 수 없는 아주 작은 값으로서 본 논문에서 제안된 교정 제어기의 활용가능성을 입증한다. 그림 8에서는 제어 불능 입력 g가 발생한 후 제어기가 가지는 교정 동작 실험 결과도 나타내었다(t=148.922ns 지점 이후 구간 참조). 페루프 시스템의 상태 천이 및 제어 입력 생성 과정은 앞의 경우와 유사하게 분석된다.

실험에서는 2 bit 오류 카운터에 대하여 다루었지만 2 bit 오류 카운터의 출력을 상위 bit 카운터를 위한 비동기 입력으로 사용하면 일반적인 n-bit (n≥2) 오류 카운터에도 직접 적용할 수 있다.

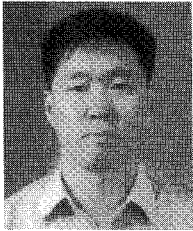
V. 결론

본 논문에서는 제어 불능 입력이 존재하는 입력/상태 비동기 머신을 위한 모델 정합 제어 시스템을 제안하였다. 제어기는 제어 불능 입력의 값을 바꿀 수 없으므로 모델 정합을 위해서는 기존 조건보다 더 제한적인 도달가능성 조건이 필요하다. 교정 제어기가 존재할 필요충분조건은 머신이 제어 불능 입력에 의해서 천이된 상태에서부터 목적 상태까지의 도달가능성을 항상 보장해야 한다는 것이다. 본 논문에서는 제안된 교정 제어 시스템이 비동기 머신의 특성을 잃지 않고 잘 동작됨을 보이기 위해서 비동기 오류 카운터를 사례 연구로 선정하고 페루프 시스템을 VHDL 코드로 구현하였다. VHDL 실험 결과를 통해서 본 논문에서 제안된 모델 정합이 비동기 디지털 회로로 직접 구현될 수 있음을 입증하였다.

참고문헌

- [1] J. Sparsø and S. Furber (eds.), *Principles of Asynchronous Circuit design - A Systems Perspective*, Kluwer Academic Publishers, 2001.
- [2] M. B. Josephs, S. M. Nowick, and C. H. Van Berkel, "Modeling and design of asynchronous circuits," *Proceedings of the IEEE*, vol. 87, no. 2, pp. 234-242, 1999.
- [3] 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.
- [4] N. Venkatraman and J. Hammer, "On the control of asynchronous sequential machines with infinite cycles," *International Journal of Control*, vol. 79, no. 7, pp. 764-785, 2006.
- [5] J.-M. Yang and J. Hammer, "State feedback control of asynchronous sequential machines with adversarial inputs," *International Journal of Control*, vol. 81, no. 12, pp. 1910-1929, 2008.
- [6] X. Geng, "Model matching for asynchronous sequential machines," Ph.D. Dissertation, Department of Electrical and Computer Engineering, University of Florida, USA, 2003.
- [7] 양정민, "입력 외란이 존재하는 비동기 순차 머신의 모델 매칭," *전기학회논문지*, 제57A권 제1호, pp. 109-116, 2008.

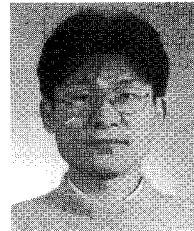
- [8] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, Boston, MA: Kluwer, 1999.
- [9] B. A. Brandin and W. M. Wonham, "Supervisory control of timed discrete-event systems," *IEEE Transactions on Automatic Control*, vol. 39, no. 2, pp. 329-342, 1994.
- [10] P. C. Y. Chen and W. M. Wonham, "Real-time supervisory control of a processor for non-preemptive execution of periodic tasks," *Real-Time Systems*, vol. 23, pp. 183-208, 2002.
- [11] Z. Kohavi, *Switching and Finite Automata Theory* 2nd Ed., McGraw-Hill, 1978.
- [12] J. A. Abraham, E. S. Davidson, and J. H. Patel, "Memory system design for tolerating single event upsets," *IEEE Transactions on Nuclear Science*, vol. 30, no. 6, pp. 4339-4344, 1983.
- [13] G. C. Cardarilli, M. Ottavi, S. Pontarelli, M. Re, and A. Salsano, "Fault tolerant solid state mass memory for space applications," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1353-1372, 2005.
- [14] A. M. Saleh, J. J. Serrano, and J. H. Patel, "Reliability of scrubbing recovery-techniques for memory systems," *IEEE Transactions on Reliability*, vol. 39, no. 1, pp. 114-122, 1990.
- [15] 광성우, 박홍영, "과학기술위성 1호 탑재 컴퓨터 (On-board Computer)에서의 SEUs (Single Event Upsets) 극복을 위한 메모리 운용 및 해석," *한국항공우주학회지*, 제32권 제1호, pp. 98-105, 2004.



양정민

1993년 한국과학기술원 전기및전자공학과 졸업(공학사). 1995년 한국과학기술원 전기및전자공학과 졸업(공학석사). 1999년 한국과학기술원 전기및전자공학과 졸업(공학박사). 1999년~2001년 한국전자통신연구원 컴퓨터·소프트웨어 기술연구소 선임연구원. 2001년~현재 대구가톨릭대학교 전자공학과 부교수. 관심분야는 비동기 순차 머신 제어, 실시간 시스템, 걸음새 연구.

트웨어 기술연구소 선임연구원. 2001년~현재 대구가톨릭대학교 전자공학과 부교수. 관심분야는 비동기 순차 머신 제어, 실시간 시스템, 걸음새 연구.



곽성우

1993년 한국과학기술원 전기및전자공학과 졸업(공학사). 1995년 한국과학기술원 전기및전자공학과 졸업(공학석사). 2000년 한국과학기술원 전기및전자공학과 졸업(공학박사). 2000년~2002년 인공위성연구센터 선임연구원, 연구교수. 2003년~현재 계명대 전자공학과 부교수. 관심분야는 위성 탑재 컴퓨터, 실시간 시스템, 비동기 시스템 설계, 내고장성 시스템 설계.

연구교수. 2003년~현재 계명대 전자공학과 부교수. 관심분야는 위성 탑재 컴퓨터, 실시간 시스템, 비동기 시스템 설계, 내고장성 시스템 설계.