

비동기 순차 머신의 강인한 상태 피드백 제어 및 VHDL 구현

| |
|----------|
| 논문 |
| 58-12-30 |

Robust State Feedback Control of Asynchronous Sequential Machines and Its Implementation on VHDL

양정민* ·곽성우†
(Jung-Min Yang · Seong Woo Kwak)

Abstract - This paper proposes robust state feedback control of asynchronous sequential machines with model uncertainty. The considered asynchronous machine is deterministic, but its state transition function is partially known before executing a control process. The main objective is to derive the existence condition for a corrective controller for which the behavior of the closed-loop system can match a prescribed model in spite of uncertain transitions. The proposed control scheme also has learning ability. The controller perceives true state transitions as it undergoes corrective actions and reflects the learned knowledge in the next step. An adaptation is made such that the controller can have the minimum number of state transitions to realize a model matching procedure. To demonstrate control construction and execution, a VHDL and FPGA implementation of the proposed control scheme is presented.

Key Words : Asynchronous sequential machines, Robust corrective control, Uncertain transitions, VHDL, FPGA

1. 서론

비동기 순차 머신(asynchronous sequential machine)의 피드백 제어, 또는 '교정 제어(corrective control)'는 전통적인 자동 제어(automatic control) 이론을 이용하여 비동기 머신의 안정 상태(stable state) 동작을 원하는 모델의 동작과 일치시키는 문제이다[1]-[3]. 페루프 시스템이 원하는 모델 입력-출력 동작을 정확하게 모사한다는 점에서 교정 제어는 종종 '모델 정합(model matching)'이라고도 불린다. 그림 1은 교정 제어의 기본 구조를 나타낸다. Σ 는 제어 대상 비동기 머신이며 C는 역시 비동기 머신으로 구현되는 교정 제어기이다. v는 외부 입력, u는 C가 만드는 제어 입력이며, x는 Σ 로부터 나오는 피드백 값이다. Σ_c 는 제어기 C에 의해 교정되는 페루프 시스템을 가리킨다.

머신 Σ 가 외부 입력 v 대신 C가 만드는 제어 입력 u로 동작하면서 출력 피드백 x를 C에 다시 전달해주는 그림 1의 구조는 자동 제어의 피드백 제어기 구조와 일치한다. 하지만 비동기 머신은 전역 클럭(global clock) 없이 동작하는 이산 사건 시스템(discrete event system)의 일종이므로 C가 제어 입력 u를 만드는 과정은 연속 시간 시스템에서의 동작과는 많이 다르다. 비동기 머신의 내부 논리 회로를 다시 설계하지 않고 머신의 동작을 교정한다는 점에서 교정 제어는 기존의 연속 시간 시스템에 대한 선형 제어나 이산 사건 시스템에 대한 관리 제어(supervisory control)와 뚜렷한 차별성을 가진다.

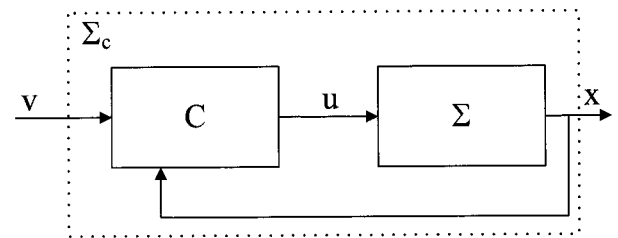


그림 1 교정 제어의 기본 구조.

Fig. 1 Basic configuration of corrective control.

본 논문에서는 모델 불확실성을 가지는 비동기 순차 머신에 대한 교정 제어를 다룬다. 고려하는 비동기 머신은 결정적(deterministic) 머신이나 상태 천이가 불확실한 상태-입력 조합을 내재하고 있다. 즉 불확실한 상태-입력 조합으로부터 머신이 천이하는 다음 상태를 제어기가 정확하게 알지 못하는 경우이다. 우주 입자나 급격한 전압 변화로 시스템의 논리출력 하나가 특정 값으로 고정되어 버리는 Single Event Latchup (SEL)[4] 고장이 이러한 불확실한 상태 천이를 유발하는 사건의 예이다. 본 논문에서는 이러한 모델 불확실성에도 불구하고 페루프 시스템의 안정 상태 동작이 원하는 모델의 동작과 일치하도록 하는 강인한(robust) 교정 제어기를 구현한다. 기존 연구들과 비교하여 본 논문의 기여도를 정리하면 다음과 같다.

i) 불확실성을 가진 비동기 머신에 대한 교정 제어 연구는 [1]과 [3],[5],[6]에서도 수행되었다. 하지만 [1]은 머신이 불확실한 상태 천이 대신 비결정적(non-deterministic) 동작을 유발하는 크리티컬 레이스(critical race)[7], 즉 비동기 머신의 설계 과정을 통해 발생하는 불확실성을 지닌다

* 정 회 원 : 대구가톨릭대 전자공학과 부교수

E-mail : jmyang@cu.ac.kr

† 교신저자, 정회원 : 계명대 전자공학과 부교수

접수일자 : 2009년 9월 30일

최종완료 : 2009년 11월 11일

고 가정하였다. 또 [3],[5],[6]의 연구는 비동기 머신이 외부 외란 입력에 의해서 원하지 않는 상태 천이가 일어났을 때 이를 되돌리는 문제에 국한되었다.

- ii) 이번 연구가 가지는 또 하나의 기여도는 제어기가 학습을 통해서 최적의 모델 정합 동작을 취하도록 하는 적응(adaptive) 능력을 가지는 기법을 제안한다는 사실이다. 머신이 불확실한 상태 천이를 한 번 거치게 되면 제어기는 그 상태 천이의 정확한 정보를 피드백을 통하여 알 수 있다. 따라서 머신의 도달가능성을 갱신하여 이전까지 알지 못했던 교정 제어 경로를 추가할 수 있다. 다시 말하면 학습을 통해서 제어기의 성능, 즉 모델 정합을 달성하기까지 소요되는 상태 천이 횟수를 줄일 수 있다.

이번 논문에서 제안하는 제어기의 응용 가능성을 입증하기 위해서 그림 1의 비동기 머신 교정 제어 시스템을 VHDL 코드로 구현하고 FPGA로 설계하는 실험을 실시한다. 실험을 통해서 제어기가 불확실성을 제거하면서 모델 정합을 이루는 교정 경로를 개선한다는 사실을 입증한다.

2. 불확실한 상태 천이를 가진 비동기 머신

2.1 모델링

이번 논문에서 다루는 비동기 머신은 출력이 머신의 현재 상태 값으로 나오는 입력/상태(input/state) 비동기 머신이다. 입력/상태 머신 Σ 를 유한 상태 머신(finite state machine)으로 표현하면 아래와 같다.

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

위 식에서 A는 입력 집합, X는 n개의 원소로 이루어진 상태 집합, $x_0 \in X$ 는 초기 상태이며, $f: X \times A \rightarrow X$ 는 상태 천이 함수이다. 비동기 머신은 전역 클럭이 없기 때문에 과도 상태(transient state)를 거치는 시간이 극히 짧다는 특징을 가진다. 따라서 외부 사용자에게는 안정 상태 동작만 관측된다. Σ 의 안정 상태 동작만을 따로 표현하기 위해서 'stable-state 머신 Σ_s '를 아래와 같이 정의한다[1],[8].

$$\Sigma_s = (A, X, x_0, s) \quad (2)$$

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

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

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 [1],[2]. 예를 들어 안정 상태 x에서 입력 a가 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. 단위 입력 대신 입력 스트링(string)을 s의 변수로 설정하면 다음과 같이 일반화할 수 있다.

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

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

Σ 는 불확실한 상태 천이를 가지는 r개의 상태-입력 조합을 가진다고 가정한다. 그러한 상태-입력 조합을 $(z_i, v_i) \in X \times A$, $i=1, \dots, r$ 라고 정의한다. 제어기 C는 (z_i, v_i) 에서 정확한 다음 안정 상태를 알지 못하고 n(i)개의 상태 $z_{i1}, \dots, z_{in(i)}$ 중 하나라는 사실만 미리 알고 있다. 즉

$$s(z_i, v_i) \in \{z_{i1}, z_{i2}, \dots, z_{in(i)}\}, \quad n(i) \geq 2 \quad (5)$$

이다. 본 논문에서는 각 입력 v_i 에 침자를 추가한 n(i)개의 문자 $v_{i1}, \dots, v_{in(i)}$ 를 새로 정의하여 불확실한 상태 천이를 표시하기로 한다. v_{ij} 는 z_{ij} 와 대응하며 stable recursion 함수 s는 아래와 같이 정의된다.

$$s(z_i, v_{ij}) := z_{ij}, \quad j=1, \dots, n(i) \quad (6)$$

물론 위 n(i)개의 상태 천이 중 한 개만이 실제로 정의된(하지만 아직 알지 못하는) 관계이다. v_{ij} 를 '불확실한 입력(uncertain input character)'이라고 부르기로 하고 모든 불확실한 입력의 집합을 $A_u := \{v_{ij} | 1 \leq i \leq r, 1 \leq j \leq n(i)\}$ 라고 정의한다.

2.2 도달가능성 행렬

비동기 머신 페루프 시스템의 동작을 기준 모델의 동작과 일치시키는 모델 정합 문제를 실현하기 위해서는 제어 대상 비동기 머신의 상태간 도달가능성이 기준 모델의 도달가능성보다 더 커야 한다. 어떤 상태 x' 가 다른 상태 x로부터 도달가능(stably reachable)하다는 의미는 $s(x, t) = x'$ 인 입력 스트링 $t \in A^+$ 가 존재한다는 뜻이다. 교정 제어기는 입력 스트링 중 하나를 선택하여 제어 입력으로 사용한다[1],[3]. 그런데 이번 논문에서 다루는 비동기 머신은 (6)과 같은 불확실한 상태 천이를 가지므로 이전 연구와는 다른 도달가능성 표현법이 필요하다. 먼저 아래와 같은 행렬 $R(\Sigma)$ 를 정의한다.

정의 1. 불확실한 입력 집합 A_u 를 가지는 입력/상태 비동기 순차 머신을 $\Sigma = (A \cup A_u, X, x_0, f)$ 라고 하고 상태 집합을 $X := \{x_1, \dots, x_n\}$ 라 하자. $n \times n$ 행렬 $R(\Sigma)$ 의 (p,q)번째 원소는 아래와 같이 정의된다.

$$R_{pq}(\Sigma) := \{t \in (A \cup A_u)^+ | s(x_p, t) = x_q, |t| \leq n-1\} \quad (7)$$

$R_{pq}(\Sigma)$ 는 머신 Σ 를 상태 x_p 에서 x_q 로 천이시키고 길이가 n-1보다 적거나 같은 모든 입력 스트링을 포함한다. x_p 와 x_q 사이에서 그러한 입력 스트링이 하나도 존재하지 않으면 $R_{pq}(\Sigma) = \emptyset$ 이다. n개의 상태를 가진 비동기 머신에서는 길이 n-1 이하의 입력 스트링만 체크하면 모든 도달가능성을 확인할 수 있다[2].

$t \in A^+$ 인 t가 $R_{pq}(\Sigma)$ 에 존재하는 경우는 기존 연구들과 마찬가지로 x_p 에서 x_q 까지의 도달가능성이 확실하게 보장된다. 문제는 $R_{pq}(\Sigma)$ 에 $t \in A^+$ 인 t가 하나도 없으며 $R_{pq}(\Sigma)$ 에 속한 모든 스트링이 불확실한 입력 집합 A_u 의 원소를 적어도 하나 이상 포함하고 있는 경우이다. 교정 제어를 실행하기 전에는 제어기가 임의의 불확실한 상태-입력 조합 (z_i, v_i) 의 정확한 다음 안정 상태를 전혀 알지 못한다. 하지만 v_i 가 유도하는 모든 불확실한 입력 $v_{i1}, \dots, v_{in(i)}$ 을 각각 포함하는 n(i)개의 입력 스트링이 $R_{pq}(\Sigma)$ 에 존재한다면 v_i 가 야기하는 불확실성에도 불구하고 x_p 로부터 x_q 까지의 도달가능성이 확실하게 보장된다고 말할 수 있다.

예제 1. $A = \{a, b, c\}$, $r=1$, $(z_1, u_1) = (x, b)$, $n(1)=2$ 라고 하자. 정의에 의해서 $A_u = \{b_1, b_2\}$ 이다. 이때 $R_{pq}(\Sigma) = \{ab_1, ab_2c\}$ 라고 가정하자. $R_{pq}(\Sigma)$ 가 가지는 두 입력 스트링은 모두 불확실한 입력 b_1 과 b_2 를 각각 포함한다. 하지만 b_1 이 참이라면 ab_1 이

확실한 입력 스트링이 되고 반대로 b_2 가 참이라면 ab_2c 가 확실한 입력 스트링이 된다. 따라서 (x,b) 가 가지는 불확실성에 상관없이 $R_{pq}(\Sigma)$ 는 x_p 에서 x_q 까지의 도달가능성을 보장한다.

위에서 설명한 도달가능성은 형식 언어(formal language)의 정규표현식(regular expression)으로 일반화할 수 있다. 정규표현식은 합집합 연산자(union operator) '+'와 연결(concatenation operator) 연산자 '.'로 원소 사이의 공통 규칙을 표현한 언어 집합이다[9]. 이번 논문에서는 + 연산자가 불확실한 입력을 처리할 때에는 다음과 같은 연산 규칙을 따른다고 추가로 설정한다.

$$av_{i1}\beta_1+av_{i2}\beta_2+\dots+av_{in(i)}\beta_{n(i)} := av_i(\beta_1+\beta_2+\dots+\beta_{n(i)}), \quad \alpha, \beta_1, \dots, \beta_{n(i)} \in (A \cup A_u)^+ \quad (8)$$

위 식 (8)에서 좌변의 의미는 현재의 정규표현식이 v_i 에 대한 모든 불확실한 입력을 각각 포함하는 스트링을 모두 가지며 각 스트링의 $v_{i1}, v_{i2}, \dots, v_{in(i)}$ 문자 앞에 연결되는 스트링(prefix)이 동일하다는 뜻이다. 따라서 $n(i)$ 개의 불확실한 입력을 묶어서 한 개의 확실한 입력 v_i 로 간주한다(식 (8)의 우변).

예제 2. 예제 1에서 $R_{pq}(\Sigma)$ 에 속한 스트링으로 정규표현식을 유도하면 $a(b_1+b_2c)$ 이다. 이 표현식에 식 (8)을 다시 적용하면

$$a(b_1+b_2c) = a(b_1\varepsilon+b_2c) = ab(\varepsilon+c) \quad (9)$$

와 같이 된다. 위 식에서 ε 는 빈 스트링(empty string)을 가리킨다.

결론은 불확실성을 가지는 비동기 머신에 대한 도달가능성을 알기 위해서는 행렬 $R(\Sigma)$ 의 각 원소에 식 (8)을 포함하는 정규표현식 연산을 적용시켜야 한다는 것이다. 유도되는 정규표현식이 정상 입력 집합 A 의 원소로만 이루어진다면 불확실성에도 불구하고 도달가능성을 보장할 수 있다. 만약 식 (8)의 $\alpha, \beta_1, \dots, \beta_{n(i)}$ 스트링 중 하나가 A_u 에 속한 불확실한 입력을 가진다면 다시 (8)의 연산을 적용시켜 통합 도달경로가 존재하는지 확인해야 한다.

$R(\Sigma)$ 에 속한 스트링으로부터 유도될 수 있는 모든 정상 입력 정규표현식이 추가된 도달가능성 행렬을 $R^c(\Sigma)$ 라고 하자. 상태 x_p 에서 x_q 까지의 도달가능성 여부를 $R^c(\Sigma)$ 로 나타내면 다음과 같다.

$$x_q \text{가 } x_p \text{로부터 도달가능} \Leftrightarrow R^c_{pq}(\Sigma) \text{에 } t \in A^+ \text{인 } t \text{가 존재하거나 } A \text{의 원소로만 구성된 정규표현식이 존재} \quad (10)$$

3. 모델 정합 문제

3.1 제어기 존재 조건 및 동작 원리

제어 대상 비동기 머신 Σ 가 입력/상태 머신일 때에는 기존 모델도 입력/상태 머신이어야 한다. 또 모델 정합을 이룩하려면 임의의 입력에 대해서 머신과 모델의 출력 값이 동일해야 한다. 입력/상태 머신의 출력은 현재의 상태 값이므로 기존 모델은 제어 대상 머신 Σ 의 입력 집합 A 와 상태 집합 X , 그리고 초기 상태 x_0 를 똑같이 가져야 한다. 기존

모델을 Λ 라 하면 Λ 는 아래와 같은 stable-state 머신으로 표현된다.

$$\Lambda = (A, X, x_0, s') \quad (11)$$

(2)와 (11)을 비교하면 모델 Λ 는 머신 Σ 와 stable recursion 함수에서 차이가 난다. 또한 모델 Λ 는 식 (5)의 불확실한 상태 천이가 하나도 없는 정상적인 머신이다.

그림 1의 페루프 시스템 Σ_c 가 주어진 모델 Λ 와 모델 정합을 이루기 위해서는 앞서 기술했듯이 머신 Σ 가 가지는 임의의 두 상태에 대한 도달가능성이 모델 Λ 보다 항상 더 커야 한다. 기존 연구 [1]-[3]에서는 비동기 머신 Σ 의 상태 간 도달가능성을 'skeleton 행렬'로 간단하게 표현한 후 머신과 모델의 skeleton 행렬 사이의 부등식으로 모델 정합 조건을 나타내었다. 불확실한 상태 천이를 가진 비동기 머신 Σ 의 skeleton 행렬은 앞 장에서 구한 도달가능성 행렬 $R^c(\Sigma)$ 로부터 유도될 수 있다.

정의 2. 불확실한 입력 집합 A_u 를 가지는 입력/상태 비동기 순차 머신을 $\Sigma=(A \cup A_u, X, x_0, f)$ 라고 하고 상태 집합을 $X:=(x_1, \dots, x_n)$ 라 하자. $n \times n$ skeleton 행렬 $K(\Sigma)$ 의 (p,q) 번째 원소는 아래와 같이 정의된다.

$$K_{pq}(\Sigma) := \begin{cases} 1 & R^c_{pq}(\Sigma) \text{에 } t \in A^+ \text{인 } t \text{가 존재하거나 } A \text{의 원소로만 구성된 정규표현식이 존재} \\ 0 & \text{otherwise} \end{cases}$$

정의 2를 다시 말하면 불확실한 상태 천이가 있는 머신 Σ 에서 상태 x_q 가 x_p 로부터 도달가능하면 $K_{pq}(\Sigma)=1$ 이다. $A_u = \emptyset$ 이면 위 정의는 정상적인 비동기 머신에 그대로 적용할 수 있다. Λ 와의 모델 정합을 이룩하는 그림 1의 교정 제어기 C 가 존재할 조건을 skeleton 행렬로 표시하면 아래와 같다.

$$K(\Sigma) \geq K(\Lambda) \quad (12)$$

그림 1의 제어기 C 는 아래와 같은 입력/출력 비동기 머신으로 표현된다[1]-[3].

$$C = (A \times X, A, \Xi, \xi_0, \phi, \eta) \quad (13)$$

그림 1에서 C 가 외부 입력 v 와 상태 피드백 x 를 동시에 받으므로 C 의 입력 집합은 $A \times X$ 가 되며, C 의 출력 u 가 Σ 의 제어 입력으로 들어가므로 출력 집합은 A 로 정의된다. 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.

교정 제어기 C 는 모델 부정합이 발생하는 Σ 의 상태-입력 조합을 만나면 외부 입력 대신 원하는 상태 천이를 일으키는 제어 입력 스트링을 Σ 에 전달한다. 예를 들어 모델 Λ 와 머신 Σ 가 상태-입력 조합 (x_p, a) 에서 각각 $s'(x_p, a)=x_q$ 와 $s(x_p, a)=x_r$ 라는 상이한 상태 천이를 가진다고 가정하자. $s'(x_p, a)=x_q$ 는 x_q 가 x_p 로부터 도달가능하다는 의미이므로 정의 2에서 $K_{pq}(\Lambda)=1$ 이다. 그런데 제어기 존재 조건 (12)에 의해서 $K_{pq}(\Lambda)=1$ 이면 $K_{pq}(\Sigma)=1$ 이므로 머신 Σ 를 x_p 에서 x_q 로 보내는 입력 스트링이 $R^c_{pq}(\Sigma)$ 에 적어도 하나 존재한다. 따라서 Σ 가 안정 상태 x_p 에 머물러 있을 때 외부로부터 입력 $v=a$ 가 들어오면 제어기 C 는 a 대신 $R^c_{pq}(\Sigma)$ 에 속한 스트링을 Σ 에 차례차례 입력시킨다(그림 1 참조). Σ 는 전역 클럭이 없는 비동기 머신이므로 이러한 교정 동작은 순식간에 진행되어 외부 사용자에게는 외부 입력 a 에 대해서 머신이

x_p 에서 x_q 로 즉시 천이하는 동작이 관측될 것이다.

3.2 제어 입력 스트링 및 학습을 통한 교정 경로 수정

조건 (10)을 만족시키는 $R_{pq}^c(\Sigma)$ 가 x_p 에서 x_q 까지의 도달 가능성을 보장하는 스트링을 둘 이상 포함할 때에는 어떤 스트링을 교정 동작에 쓸 것인가 하는 선택 문제가 대두된다. 제어기 C가 모델 부정합을 일으키는 어떤 상태-입력 조합의 동작을 교정하기 위해서는 사용하는 입력 스트링 길이 만큼의 보조 상태(auxiliary state)를 더 필요로 한다[2],[5]. 따라서 교정 제어기의 크기(상태 수)를 줄이기 위해서는 최대한 길이가 짧은 입력 스트링을 교정 동작에서 사용해야 한다.

$R_{pq}^c(\Sigma)$ 에는 불확실한 상태 천이를 통합한 정규표현식 (식 (8) 참조)이 있을 수 있다. 본 논문에서는 정규표현식의 길이를 표현식 안에 포함된 스트링들의 최대 길이로 정의한다. 이렇게 설정하는 이유는 불확실한 상태-입력 조합의 정확한 다음 안정 상태를 제어기가 알지 못하므로 여러 개의 불확실한 입력 스트링 중 최악의 경우 최대 길이를 가지는 스트링이 참이 될 수도 있기 때문이다.

예제 3. 예제 2의 도달가능성 행렬의 $R^c(\Sigma)$ 의 (p,q) 원소 $R_{pq}^c(\Sigma)$ 는 정규표현식 $t=ab(\epsilon+c)$ 를 가진다. 따라서 t의 길이는 $\max(|ab\epsilon|,|abc|)=3$ 이다.

불확실한 상태 천이는 머신이 해당 입력을 받아들여 실제 천이를 거치고 나면 확실한 상태 천이가 된다. 그림 1에서 볼 수 있듯이 제어기는 Σ 로부터 상태 피드백 x 를 계속 받으므로 Σ 가 상태 z_i 에 있을 때 제어 입력으로 v_i 를 받는다면 C는 식 (5)에 나온 가능한 n(i)개의 상태 $z_{i1}, z_{i2}, \dots, z_{in(i)}$ 중 정확한 다음 안정 상태 $s(z_i, v_i)$ 값을 관측하게 된다. 제어기가 얻은 이 정보를 이용하면 도달가능성 행렬 $R^c(\Sigma)$ 의 스트링을 갱신할 수 있다. 행렬 $R^c(\Sigma)$ 의 스트링에서 불확실성을 제거하면 경우에 따라서 교정 동작에서 쓰기로 설정한 최소 길이 스트링보다 더 짧은 입력 스트링이 생길 수도 있다.

본 논문에서는 학습을 통해서 얻은 정보를 이용하여 제어기 C가 교정 동작에 사용하는 입력 스트링을 조정하는 기법을 제안한다. 상태 피드백을 관측하여 행렬 $R^c(\Sigma)$ 을 갱신하고 새로운 입력을 선택하면 모델 정합을 달성하기까지 소요되는 상태 천이 횟수를 줄일 수 있으므로 제어기의 성능이 향상된다. 새로운 입력 스트링이 원래 입력 스트링보다 더 짧으므로 새로운 입력 스트링을 이용한 교정 동작은 원래 입력 스트링을 적용하기 위해서 제어기가 정의한 상태를 다시 이용하여 구현할 수 있다[10]. 머신 Σ 와 모델 Λ 가 교정 제어 가능 조건 (12)를 만족시킬 때 입력 스트링 조정 기법을 정리하면 다음과 같다.

알고리즘 1: 학습을 통한 교정 경로 수정

- a) 교정 제어를 시작하기 전에 구한 머신 Σ 의 도달가능성 행렬을 $R^c(\Sigma)(0)$ 이라고 한다.
- b) $s'(x_p, a)=x_q$ 이나 $s(x_p, a)=x_r \neq x_q$ 라 하자(모델 부정합). $R_{pq}^c(\Sigma)(0)$ 에 있는 정상 입력 집합 A 원소로만 이루어진 정규표현식 중 최소 길이를 가진 스트링을 t라고 한다. 또 $R_{pq}^c(\Sigma)$ 가 가지는 불확실한 스트링, 즉 불확실한 입력 집

합 A_0 의 문자를 적어도 하나 이상 포함하는 스트링 중 길이가 t보다 짧은 것들의 집합을 $S(t) \subset R_{pq}^c(\Sigma)$ 라고 정의한다.

- c) t와 S(t)에 속하는 모든 스트링을 제어 입력 스트링으로 따로따로 사용한다고 가정하고 그에 맞게 제어기 C의 상태 천이 함수 ϕ 와 출력 함수 η 를 정의한다. 앞서 기술했듯이 t와 S(t)의 모든 원소들을 제어 입력 스트링으로 사용하기 위해서 C가 필요한 보조 상태는 |t|개만 있으면 된다.
- d) 구현한 제어기 C와 머신 Σ 를 결합하여 그림 1의 페루프 시스템 Σ_c 를 완성하고 교정 제어를 실시한다. 교정 제어 실행 중 불확실한 상태 천이에 속하는 입력 v_1, \dots, v_r 중 하나가(식 (5) 참조) 머신 Σ 에 들어가면 행렬 $R^c(\Sigma)(0)$ 의 각 항목들을 갱신한다. 갱신된 행렬을 $R^c(\Sigma)(1)$ 이라 하자.
- e) $R_{pq}^c(\Sigma)(1)$ 와 $R_{pq}^c(\Sigma)(0)$ 을 비교하여 c)에서 구한 $R_{pq}^c(\Sigma)(0)$ 의 부분 집합 S(t)의 스트링 중 t'의 불확실성이 완전히 제거되었다면 다음 모델 정합 시에는 t 대신 t'를 제어 입력 스트링으로 사용한다($\cdot : |t'| < |t|$). c)에서 설정했듯이 제어기 C에는 t'에 대한 교정 동작도 이미 정의되어 있다.
- f) d) 단계로 돌아가서 제어 입력에 따라서 도달가능성 행렬을 다시 $R_{pq}^c(\Sigma)(2), R_{pq}^c(\Sigma)(3), \dots$ 으로 갱신하고 e)의 동작을 반복한다.

알고리즘 1의 요지는 제어기 C가 현재 알려진 입력 스트링 중 가장 길이가 짧은 것을 선택하여 교정 제어에 적용함으로써 제어 동작의 최대 효율성을 꾀한다는 것이다. 알고리즘 1의 실행 및 실제 교정 제어기의 작동 과정은 다음 장 사례 연구를 통해서 설명한다.

4. VHDL 실험

4.1 사례 연구 시스템 및 제어기 설계

그림 2는 사례 연구 시스템으로 설정한 제어 대상 비동기 머신 $\Sigma=(AUA_u, X, x_0, f)$ 이다. Σ 는 그 자체로 stable-state 머신이다. 즉 $\Sigma=\Sigma_s$ 이다. $A=\{a,b,c,d\}$, $X=\{x_1, x_2, x_3, x_4\}$ 이며 초기 상태는 $x_0=x_1$ 이다. 그림에서 볼 수 있듯이 Σ 에는 불확실한 상태 천이 (x_2, a) 가 하나 존재하므로($r=1$) $A_u=\{a_1, a_2\}$ 이고 식 (6)의 정의에 따라서 $s(x_2, a_1)=x_1$, $s(x_2, a_2)=x_3$ 이다.

그림 2의 시스템 Σ 는 ROM을 포함하는 비동기 데이터 처리기로 해석될 수 있다[11]. 즉 Σ 의 x_2 는 ROM을 가리키며 d는 ROM 입력이며 a, c는 ROM의 출력으로서 데이터를 다른 저장 장소(x_1 또는 x_3)로 옮긴다. 이 데이터 처리기가 우주나 원자력 발전소 등 메모리 비트 변화를 일으킬 수 있는 방사능 환경에서 동작하여 a에 대한 ROM 매핑이 외부 사용자에게 불확실하게 된 경우를 생각한다. 즉 ROM 입력을 직접 인가하기 전까지는 a에 의해서 도달하는 다음 데이터 저장 장소를 정확하게 알지 못하며, x_1 와 x_3 중 하나라는 사실만 알고 있다.

그림 2로부터 Σ 의 skeleton 행렬 $K(\Sigma)$ 를 유도하면 다음과 같다(도달가능성 행렬 $R^c(\Sigma)$ 의 기술은 생략).

$$K(\Sigma) = \begin{pmatrix} 1111 \\ 1111 \\ 1111 \\ 1111 \end{pmatrix} \tag{14}$$

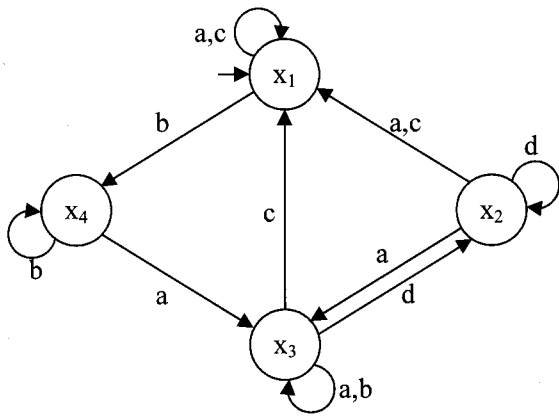


그림 2 제어 대상 비동기 머신 Σ.
Fig. 2 controlled asynchronous machine Σ.

즉 불확실한 상태 천이에도 불구하고 Σ의 임의의 상태는 다른 임의의 상태에서부터 도달가능하다. 예를 들어 Σ을 x2에서 x1까지 도달하게 하는 입력 스트링은 그림 2에서 R₂₁(Σ)={c, a₁, a₂c}로 유도된다. 식 (8)을 적용하여 정규표현식을 구하면 a₁+a₂c=a(ε+c)이므로 R₂₁^c(Σ)={c, a₁, a₂c, a(ε+c)}이고 조건 (10)에 의해서 x2는 x1로부터 도달가능하다. 다른 상태 조합의 도달가능성도 유사하게 찾을 수 있다.

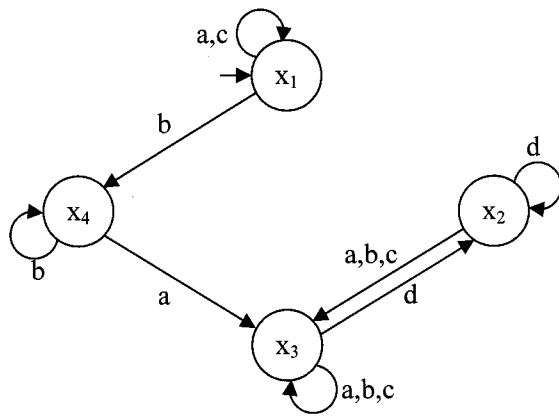


그림 3 기준 모델 Λ.
Fig. 3 Reference model Λ.

그림 3은 주어진 기준 모델 Λ=(A,X,x₀,s')이다. 즉 그림 2의 데이터 처리기 Σ가 현재 수행하는 작업을 위해서 만족시켜야 할 데이터 흐름의 명세이다. Λ의 도달가능성 행렬을 구하고 skeleton 행렬을 유도하면

$$K(\Lambda) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \quad (15)$$

이다. (14)와 (15)를 비교하면 K(Σ)≥K(Λ)이므로 모델 정합 조건 (12)가 만족된다. 그러므로 Σ_{cls}=Λ가 되게 하는 상태 피드백 제어기 C를 설계할 수 있다.

그림 2와 그림 3에서 Σ와 Λ는 다음과 같은 모델 부정합

을 보인다.

$$s(x_2,c)=x_1 \ :: \ s'(x_2,c)=x_3$$

$$s(x_2,b)=\emptyset \ :: \ s'(x_2,b)=x_3 \quad (16)$$

$$s(x_3,c)=x_1 \ :: \ s'(x_3,c)=\emptyset \quad (17)$$

제어기 설계의 예로서 모델 부정합 (16)을 없애주는 제어기 C의 동작을 기술한다. C의 제어 목적은 Σ가 안정 상태 x2에 있을 때 외부 입력 b를 받으면 Σ를 다음 안정 상태 x3로 천이시키는 것이다. ((16)에 나와 있듯이 Σ의 원래 동작에는 (x2,b)에서 stable recursion 함수 s가 정의되지 않았다.)

알고리즘 1을 적용하여 제어기 C의 설계 및 교정 경로 수정을 실행한다.

단계 a) 먼저 교정 제어를 시작하기 전에 구한 머신 Σ의 도달가능성 행렬을 R^c(Σ)(0)이라고 한다(R^c(Σ)(0)=R^c(Σ)). 그림 2에서 R₂₃(Σ)={a₁ba, a₂}이므로 제안된 정규표현식 연산을 적용하면

$$R_{23}^c(\Sigma)(0)=\{cba, a_1ba, a_2, a(ba+\epsilon)\} \quad (18)$$

와 같이 나온다.

단계 b) R₂₃^c(Σ)(0) 원소 중 정상 입력 집합 A 원소로만 이루어진 스트링은 cba, a(ba+ε)이다. |cba|=3이며 정의에 의해서 정규표현식 a(ba+ε)의 길이는 max(|a₁ba|, |a₂|)=3이다. C는 cba를 이용하여 교정 동작을 구현하기로 한다. 알고리즘 1의 단계 b)에 나온 표현에 따라서 t=cba이다. 또 집합 S(t)는 (18)로부터 S(t)={a₂}가 된다.

단계 c) C의 상태 집합을 E라 하자. E는 기본적으로 초기 상태 ζ₀과 transition 상태 ζ₀(x)를 포함하며, 길이가 3인 입력 스트링 cba로 교정 동작을 이루기 위해서 세 개의 보조 상태 ζ₁, ζ₂, ζ₃를 추가로 정의해야 한다[1]-[3]. 초기 상태 ζ₀에 있던 C는 Σ가 x2와 안정 조합을 이룰 때, 즉 (x2,d)일 때(그림 2 참조) transition 상태 ζ₀(x)로 천이한 후 외부 입력이 b로 바뀌기를 기다린다. b 대신 다른 입력이 들어오면 C는 다시 초기 상태 ζ₀으로 되돌아간다.

외부 입력 b가 들어오는 순간 제어기 C는 교정 동작을 실행한다. b를 받은 C는 먼저 상태 ζ₁로 이동한 후 입력 스트링의 첫 번째 문자 c를 Σ에 넣어준다. c를 받은 Σ는 다음 상태 x1로 이동한다. 또 Σ가 주는 상태 피드백 x1을 받은 C는(그림 1 참조) 두 번째 보조 상태 ζ₂로 이동한 후 다시 입력 스트링의 두 번째 문자 b를 Σ에 전달한다. 이런 식으로 계속하면 Σ는 원하는 상태 x3에 도달한다. 이 모든 교정 동작이 클럭 없이 순식간에 진행되므로 외부 사용자에게는 페루프 시스템이 안정 상태 x2에 있다가 입력 b에 반응하여 다음 안정 상태 x3로 즉시 천이하는 모습만이 관측될 것이다.

단계 d) 구현한 제어기 C와 머신 Σ를 결합하여 교정 제어를 실시하고 도달가능성 행렬을 갱신한다. 예를 들어 Σ가 불확실한 상태 천이 (x2,a)를 통과했고 제어기 C가 상태 피드백을 받고 Σ가 실제 가지는 다음 안정 상태가 x3라는 사실을 알았다고 가정하자. R^c(Σ)(0)로부터 갱신된 행렬 R^c(Σ)(1)의 (2,3) 원소는 다음과 같다(식 (18)과 비교).

$$R_{23}^c(\Sigma)(1)=\{cba, a\} \quad (19)$$

불확실성이 제거되면 x2에서 x3까지 연결하는 입력 스트링 중 (x2,a)→x1을 포함하는 스트링들은 모두 소거된다.

단계 e) (19)의 R₂₃^c(Σ)(1)과 (18)의 R₂₃^c(Σ)(0)을 비교하면

최초에 사용했던 입력 스트링 cba보다 길이가 더 짧은 스트링 t:=a를 찾을 수 있다. 따라서 다음 모델 정합 시에는 cba 대신 a를 C에 대한 제어 입력 스트링으로 사용한다. cba를 사용하기 위해서 정의한 보조 상태 ξ_1, ξ_2, ξ_3 중 ξ_1 하나만 이용하여 교정 동작을 꾸밀 수 있으므로 제어기의 크기에는 변화가 없으며 교정 속도만 개선될 것이다.

4.2 VHDL 실험 결과

그림 2의 제어 대상 비동기 머신 Σ 와 제안된 강인 교정 제어기가 결합된 페루프 시스템(그림 1 참조)을 VHDL 코드를 이용하여 구현하였다. 이번 실험에서는 Altera사의 QUARTUS® II(ver.7.1)를 VHDL 컴파일러 및 합성기로 사용하였으며, 페루프 시스템이 설계된 target FPGA로 Altera의 EP1C6Q240C8를 채택하였다. 머신 Σ 의 총 상태가 네 개이므로 실험에서는 2 비트 변수 y_1y_0 을 이용하여 다음과 같이 각 상태를 표시하였다.

$$x_1=00, x_2=01, x_3=10, x_4=11$$

설계된 강인한 교정 제어기의 동작을 예시하기 위해서 이번 실험에서는 식 (16)에서 나타낸 (x_2,b) 에 대한 모델 정합 과정을 구현하였다. 교정 제어를 처음 시작할 때에는 불확실한 상태 천이에 대한 어떠한 확실한 정보도 알지 못하므로 앞 절 단계 a)와 b)에서 밝힌 바대로 $t=cba$ 를 제어 입력 스트링으로 사용한다. 그림 4는 이 경우에 대한 VHDL 실험 결과이다. 안정 상태 x_2 에 있던 머신 Σ 는($y_1=1, y_0=0$) 시간 19.0ns에서 외부 입력 a를 받는다. 이때 transition 상태 $\xi_0(x)$ 에 있던 교정 제어기가 작동하여 입력 스트링 순서대로

c, b, a의 제어 입력을 머신에 전달한다. 그림 4에서 a_ctl~d_ctl은 제어기가 만드는 제어 입력이라는 뜻이며 물리적으로는 외부 입력 a~d와 동일한 값이다. 그림 4에서 볼 수 있듯이 제어 입력을 받은 Σ 는 각각 x_1, x_4, x_3 로 이동하여 모델 정합을 완성한다. 교정 제어가 시작된 지점부터 끝난 지점까지의 경과 시간은 $48.4-22.5=25.9[ns]$ 이다.

그림 5는 머신 Σ 가 교정 제어 시작 후 불확실성을 제거한 후 제어기가 입력 스트링을 갱신하는 과정을 보여준다. Σ 가 가지는 불확실한 상태 천이는 (x_2,a) 라고 앞에서 밝혔다(그림 2). 그런데 그림 5의 시나리오에서는 상태 x_2 에 있던 Σ 가 시간 19.0ns에서 외부 입력 a를 받은 후 상태 x_3 로 천이하였다. 즉 불확실한 상태-입력 조합 (x_2,a) 의 다음 안정 상태가 $\{x_1, x_3\}$ 중 x_3 라는 사실이 밝혀졌다. 앞 절 단계 d)와 e)에서 기술했듯이 제어기는 도달가능성 행렬을 갱신한 후 (x_2,b) 의 모델 정합에 대한 입력 스트링으로 cba 대신 a를 선택할 수 있게 된다.

그림 5에서 Σ 는 39.0ns에서 발생한 외부 입력 d에 의해서 상태 x_2 로 되돌아간 후 69.0ns에서 외부 입력 b를 다시 받는다. 이때는 제어기가 cba 대신 a를 제어 입력 스트링으로 사용하므로 머신은 x_2 에서 원하는 상태 x_3 로 바로 천이한다. 그림 5에서 교정 제어를 완료하는 데 걸린 시간은 $77.9-72.1=5.8[ns]$ 로 관측된다. 그림 4의 결과와 비교하면 그림 5의 교정 제어 과정이 약 20ns정도 더 빠르다.

그림 4와 그림 5의 실험 결과는 제안된 교정 제어기가 VHDL 코드로 구현될 수 있음을 입증한다. 또 제어기는 불확실한 상태 천이에 대해서 강인함을 보이고 불확실성이 제

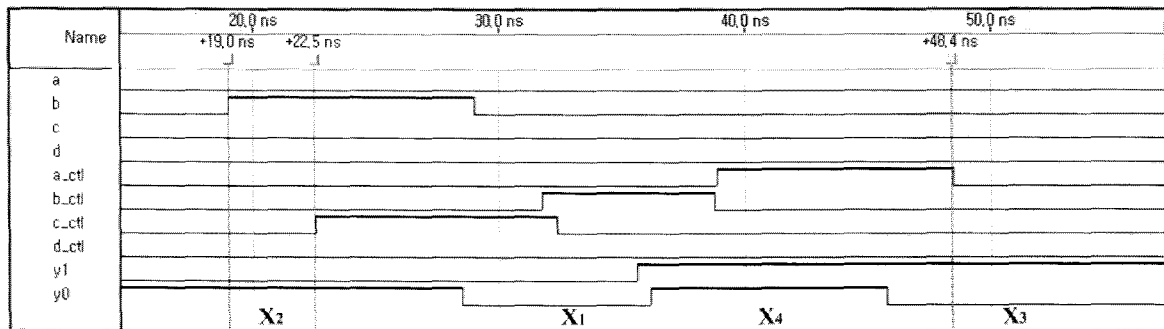


그림 4 VHDL 실험 결과: 입력 스트링 t=cba를 이용한 (x_2,b) 에 대한 모델 정합.

Fig. 4 Result of VHDL experiment: model matching for (x_2,b) using the input string t=cba.

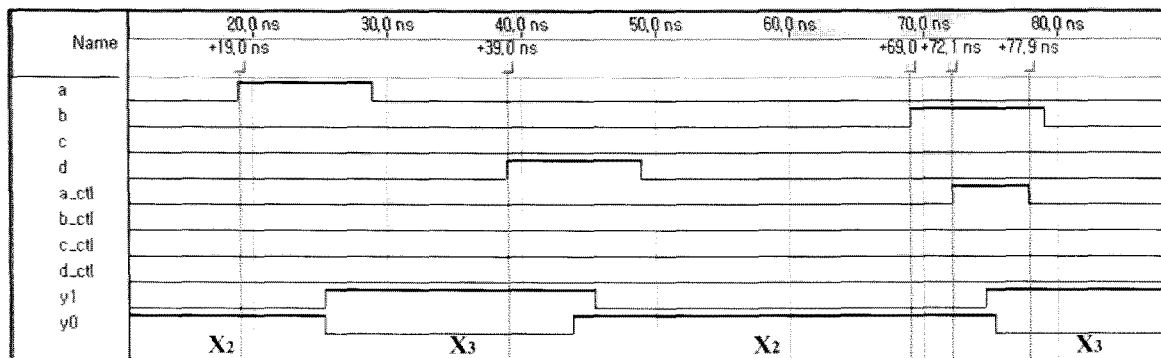


그림 5 VHDL 실험 결과: 불확실성 제거 과정 및 입력 스트링 t=a를 이용한 (x_2,b) 에 대한 모델 정합.

Fig. 5 Result of VHDL experiment: removing uncertainty and model matching for (x_2,b) using the input string t=a.

거될 때마다 제어 입력 스트링을 갱신하여 성능(속도)을 개선할 수 있다는 사실도 입증한다.

4.3 FPGA 실험 결과

4.2절에서 작성한 VHDL 코드를 바탕으로 FPGA를 설계하여 앞에서 기술한 교정 제어 시나리오를 구현하였다. 주요 외부 입력과 제어 입력, 그리고 상태 비트들의 동작 변화를 4-채널 디지털 오실로스코프를 통해서 추출한 후 시간에 대한 그래프로 나타내었다.

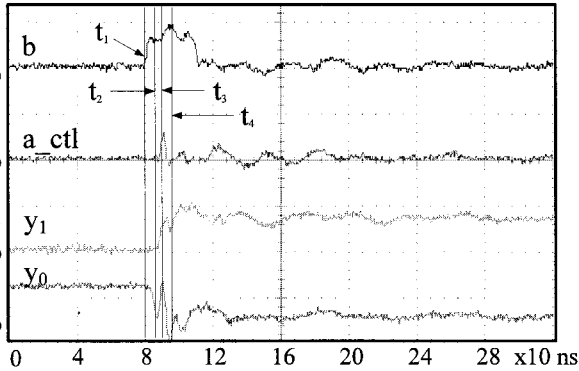


그림 6 FPGA 실험 결과: 입력 스트링 t=cba를 이용한 (x₂,b)에 대한 모델 정합.

Fig. 6 Result of FPGA experiment: model matching for (x₂,b) using the input string t=cba.

그림 6은 그림 4의 제어 시나리오를 설계된 FPGA 시스템에서 구동시킨 결과이다. y₁y₀=01, 즉 안정 상태 x₂에 있는 머신에 외부 입력 b가 들어오면 교정 제어기는 입력 스트링 cba를 차례로 생성하여 페루프 시스템의 다음 안정 상태가 x₃가 되게 교정 동작을 실시한다. 그림 6에서 t₁~t₃은 각각 제어 입력 c, b, a가 들어오는 시점이며 t₄는 교정 동작이 끝난 시점을 가리킨다. (그림 6에서는 세 개의 제어 입력 중 'a_ctl', 즉 마지막 제어 입력 a만 표시하였다.) 이 구간에서 상태 비트 y₁y₀이 x₁(00), x₄(11), x₃(10)으로 바뀌는 현상을 관측할 수 있다. 교정 동작의 총 소요 시간(t₄-t₁)은 약 16.3ns로 측정되었다.

그림 7은 그림 5의 제어 시나리오를 설계된 FPGA 시스템에서 구동시킨 결과이다. 그림 7(a)는 시스템이 불확실한 상태 천이 (x₂,a)를 통과하여 다음 안정 상태를 관측한 후 불확실성에 대한 정확한 정보를 얻는 과정이다. 시스템이 안정 상태 x₂(y₁y₀=01)에 있을 때 시간 t₁에서 입력 a가 들어와 시간 t₂에서 y₁y₀=10, 즉 상태 x₃로 천이하는 모습을 관측할 수 있다. 따라서 (x₂,a)의 다음 안정 상태는 x₃이다. 그림 7(b)는 그림 6과 동일한 모델 정합 문제를 t=cba 대신 t=a를 사용하여 구현한 모습이다. 시간 t₄에서 외부 입력 b가 들어오면 t₅에서 제어 입력 a_ctl이 생성되어 t₆에서 교정 동작이 완성된다. 교정 동작의 총 소요 시간 t₆-t₄는 약 10.5ns로 측정되었다. 사용된 제어 입력 스트링의 길이가 줄어들었으므로 그림 6의 결과보다 약 5.8ns의 시간이 단축되었다. 이 실험 결과는 제안된 제어 시스템이 불확실성이 제거될 때마다 제어 입력 스트링을 갱신하여 성능을 개선할 수 있다는 VHDL 실험 결과를 확증한다고 말할 수 있다.

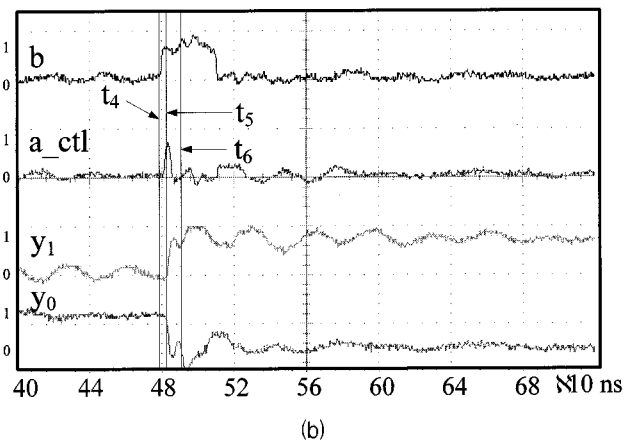
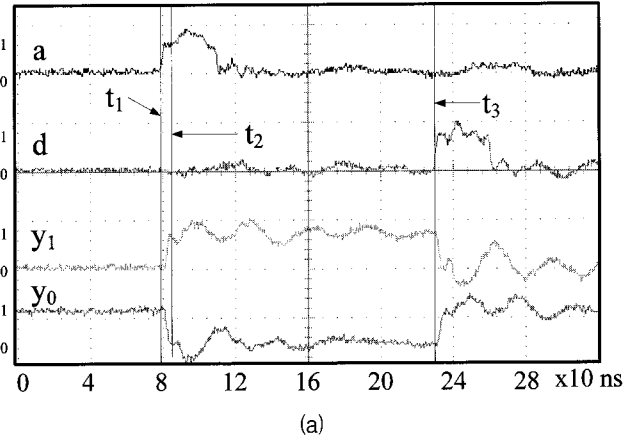


그림 7 FPGA 실험 결과: 입력 스트링 t=cba를 이용한 (x₂,b)에 대한 모델 정합.

Fig. 7 Result of FPGA experiment: model matching for (x₂,b) using the input string t=cba.

5. 결론

본 논문에서는 불확실한 상태 천이가 존재하는 입력/상태 비동기 머신을 위한 강인한 모델 정합 제어 시스템을 제안하였다. 교정 제어를 시작하기 전에 제어기는 불확실한 상태 천이에 대한 다음 안정 상태를 정확하게 알지 못하므로 모델 정합을 위해서는 기존 조건보다 더 제한적인 도달가능성 조건이 필요하다. 교정 제어기가 존재할 필요충분조건은 불확실한 상태 천이가 가질 수 있는 가능한 한 모든 상태 천이 경로에 대해서 목적 상태까지의 도달가능성을 머신이 보장해야 한다는 것이다. 또 교정 제어를 시작하고 난 후 불확실성이 제어될 때마다 도달가능성을 갱신하여 제어기의 성능을 높이는 방법을 제안하였다. 제안된 교정 제어 시스템이 비동기 머신의 특성을 잃지 않고 잘 동작됨을 보이기 위해서 본 논문에서는 비동기 머신과 제어기가 결합된 페루프 시스템을 VHDL 코드 및 FPGA 시스템으로 구현하였다. 실험 결과를 통해서 본 논문에서 제안된 모델 정합 방법이 비동기 디지털 회로로 직접 구현될 수 있음을 입증하였고 제어기가 불확실한 상태 천이에 대해서 강인성과 적응성을 모두 보유하고 있음을 증명하였다. 본 논문에서 제시된 방법은 내고장성 회로 설계, 고장극복 기법의 구현에 직접 응

용될 수 있다. 추후에는 보다 구체적인 사례 연구를 통하여 제시된 방법의 적용 및 현실적인 유용성을 입증할 예정이다.

참 고 문 헌

[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] X. Geng and J. Hammer, "Input/output control of asynchronous sequential machines," IEEE Transactions on Automatic Control, vol. 50, no. 12, pp. 1956-1970, 2005.

[3] 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.

[4] R. G. Useinov, "Analytical model of radiation induced or single event latchup in CMOS integrated circuits," IEEE Transactions on Nuclear Science, vol. 53, no. 4, pp. 1834-1838, 2006.

[5] 양정민, 곽성우, "비동기 상태 피드백 제어를 이용한 TMR 메모리 SEU 극복," 전기학회논문지, 제57권 제8호, pp. 1440-1446, 2008.

[6] 양정민, 곽성우, "외란 입력을 극복하기 위한 입력/출력 비동기 머신의 교정 제어," 전기학회논문지, 제58권 제3호, pp. 591-597, 2009.

[7] S. H. Unger, "Hazards, critical races, and metastability," IEEE Transactions on Computers, vol. 44, no. 6, pp. 754-768, 1995.

[8] Z. Kohavi, Switching and Finite Automata Theory (2nd ed.), New York: McGraw-Hill, 1978.

[9] M. Sipser, Introduction to the Theory of Computation (2nd ed.), Massachusetts: Thomson, 2006.

[10] 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.

[11] J. Sparsø and S. Furber (eds.), Principles of Asynchronous Circuit Design-A Systems Perspective, Kluwer Academic Publishers, 2001.

저 자 소 개

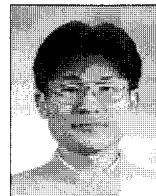


양 정 민 (楊正敏)

1971년 3월 31일생. 1993년 2월 한국과학기술원 전기 및 전자공학과 졸업(학사). 1995년 2월 한국과학기술원 전기 및 전자공학과 졸업(석사). 1999년 2월 한국과학기술원 전기 및 전자공학과 졸업(공학박). 1999년 3월~2001년 2월 한국전자통신연구원 컴퓨터·소프트웨어연구소 선임연구원. 2001년 3월~현재 대구가톨릭대학교 전자공학과 부교수. 주관심분야: 비동기 순차 머신 제어, 실시간 시스템 등.

Tel : 053-850-2736, Fax : 053-850-2704

E-mail : jmyang@cu.ac.kr



곽 성 우 (郭成祐)

1970년 3월 10일생. 1993년 2월 한국과학기술원 전기 및 전자공학과 졸업(학사). 1995년 2월 한국과학기술원 전기 및 전자공학과 졸업(석사). 2000년 2월 한국과학기술원 전기 및 전자공학과 졸업(공학박). 2000년~2002년 인공위성연구센터

선임연구원, 연구교수. 2003년~현재 계명대학교 전자공학과 부교수. 주관심분야: 실시간시스템, 내고장성 기법, 비동기 시스템, 위성 시스템 등.

Tel : 053-580-5926, Fax : 053-580-5165

E-mail : ksw@kmu.ac.kr