

유한 상태 기계 용 고장 허용 시스템 비교

신예린, 박지운, 최소연, 김현규, 류도현, 유호영
충남대학교

전화: (042)821-6285, E-mail: hyyoo@cnu.ac.kr

Fault Tolerant Systems for Finite State Machines

Yerin Shin, Jiwoon Park, Soyeon Choi, Hyeonkyu Kim, Dohyun Ryu, and Hoyoung Yoo
Chungnam National University

요약

디지털 회로의 동작을 제어하는 상태 기계에 결함이 발생하면 전체 시스템에 오작동이 유발되어 큰 문제를 일으킬 수 있다. 이에 대한 해결책으로 기존의 유한 상태 기계에 추가적인 회로를 추가하여 오류 발생을 무효화하는 고장 허용 유한 상태 기계가 전통적으로 적용되며, 다양한 고장 허용 기술 중 적용의 용이성과 확장성 덕분에 부호화 기법이 가장 널리 적용된다. 본 논문에서는 다양한 부호화 기법에 대하여 하드웨어를 구현한 후 이에 대한 성능 지표를 비교 분석한다. 이를 통해 하드웨어 설계자가 적절한 상태 부호화 방법을 선택할 수 있도록 하며, 최종적으로 전체 시스템의 최적화를 이룰 수 있는 기준을 제공한다.

Abstract

When a fault or error occurs in a state machine that controls the operation of a device, malfunction may occur resulting in serious problems. As a solution to such a problem, fault tolerant techniques are widely employed to the conventional finite state machine by adding additional circuitries. In this paper, we provide pragmatic comparisons in terms of gate count, critical path delay, and area-times by comparing the most widely used encoding methods. As a result, the guideline to determine an appropriate encoding method of FSMs is provided for hardware designers leading to an overall improvement for the fault tolerant systems.

Keywords: 유한 상태 기계, 부호화, 고장 허용, 원-핫, 해밍 부호

I. 서론

유한 상태 기계 (Finite-State Machine, FSM)는 유한한 개수의 상태를 정의하고 이 상태 사이의 전이를 제어함으로써 기기의 순차 논리를 제어하는데 사용된다. 전체 시스템을 제어하는 역할을 담당하기 때문에 다른 회로에 비해 중요하며 오류에 더욱 민감하다 [1]. 예를 들어 우주 기상의 영향으로 기기가 방사선에 노출되게 되면 일시적으로 출력 오류를 일으키거나 상태의 단계가 바뀌어 원하지 않은 동작이 실행되는 SEU (Single Event Upsets)가 발생한다. 차량용 장비와 군용 장비에서 이러한 오작동은 심각한 문제를 야기할 수 있다. 이 문제를 해결하기 위해 오류가 발생하여도 전체 시스템이 올바르게 실행되도록 하는 고장 허용 (Fault Tolerant) 유한 상태 기계 (FSM)가

제안되었다 [2]. 고장 허용 (Fault Tolerant) 유한 상태 기계 (FSM)의 각 상태를 정의하는 부호화 (Encoding) 방법은 다양하게 존재하며, 부호화 방법에 따라 오류 정정 성능과 하드웨어 성능이 달라진다.

II. 본론

유한 상태 기계 (FSM)의 기본 구조를 그림 1에 나타내었다. 유한 상태 기계는 기본적으로 1) 다음 상태 생성부, 2) 상태 레지스터, 3) 출력 생성부로 구성된다. 외부 환경의 영향으로 상태 값이 변경됨에 따라 예상치 못한 상태로 변환되거나 원하지 않는 출력이 생성될 수 있다 [2]. 이를 방지하기 위해 고장 허용 (Fault Tolerant) 시스템에서는 추가적인 오류 검출/정정 회로를 구현한다 [3].

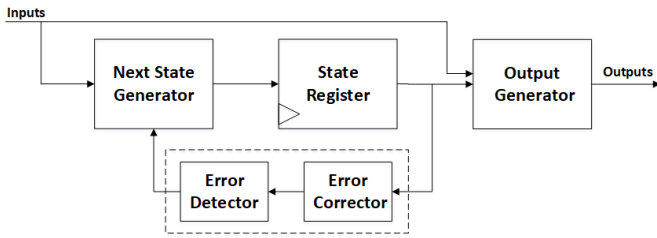


그림 1. 고정 허용 유한 상태 기계의 기본 구조.

본 논문에서는 이진 (Binary) 부호화, 원-핫 (One-Hot) 부호화, 해밍 (Hamming) 부호화 방법을 비교한다. 표 1은 유한 상태 개수가 8 개라고 가정한 상황에서 기계 부호화 방법에 따른 표현법을 나타내고 있다. 이진 (Binary) 부호화는 상태 값을 이진수로 나타내는 가장 기본적인 부호화 방법으로 오류 발생시 이미 정의되어 있는 다른 상태 값으로 전이 되어 오류의 검출/정정이 불가능하다. 원-핫 (One-Hot) 부호화는 상태 레지스터의 한 비트만을 1로 설정한다. 단일 오류 발생시 전체 상태 레지스터의 두 비트가 1이거나 모든 비트의 값이 0이 되어 원-핫 (One-Hot) 부호의 기본 논리를 무시하는 정의 되지 않은 상태 값으로 변경된다. 이 특성을 활용하여 상태 값의 패리티를 확인하는 오류 검출기를 통해 오류 발생 여부를 알 수 있다 [3]. 해밍 부호 (Hamming-n)은 주어진 상태 레지스터에 추가적인 패리티 레지스터를 활용하여 유효한 부호의 해밍 길이를 n 이상으로 유지한다. 해밍 부호 (Hamming-n)는 $n-1$ 개 오류 검출 또는 $\lfloor n/2 - 1 \rfloor$ 정정이 가능한 부호로써 하나 이상의 오류의 처리를 위하여 적용된다.

원-핫 (One-Hot) 부호화, 해밍 길이 2를 가지는 해밍 부호 (Hamming-2)의 경우, 단일 오류에 대한 오류 감지는 가능하지만 오류 정정이 불가능하므로 오류가 감지된 경우 유한 상태 기계 (FSM)가 초기 단계부터 다시 동작하도록 설정한다. 해밍 길이 3을 가지는 해밍 부호 (Hamming-3)의 경우, 원래 정의했던 상태 값과 인접한 값으로 상태 값이 변화하여 오류 검출뿐만 아니라 정정이 가능하다.

III. 실험결과 및 결론

다양한 부호화 기법을 하드웨어로 구현한 후 성능을 분석하여 표 2에 정리하였다. 고장 허용 유한 상태 기계의 동작 주파수는 200 MHz로 가정하였고, CMOS 180 nm 공정을 이용하여 합성을 진행하였다. 표 2에 의하면 가장 많은 플립플롭을 필요로 하는 원-핫 (one-hot) 부호화가 이진 (Binary) 부호화

표 1. 8개의 고유 상태에 대한 상태 기계 부호화

State	Binary	One-Hot	Hamming-2	Hamming-3
S0	000	00000001	0000	000000
S1	001	00000010	0011	000111
S2	010	00000100	0101	011001
S3	011	00001000	0110	011110
S4	100	00010000	1001	101010
S5	101	00100000	1010	101101
S6	110	01000000	1100	110011
S7	111	10000000	1111	110100

표 2. 고장 허용 유한 상태 기계 합성 결과

Encoding Methods	Equivalent gate count	Critical path delay	Area × Time
Binary	181.22	1.24	224.71
One-Hot	517.67	3.62	1873.97
Hamming 2	208.26	3.26	678.93
Hamming 3	324.76	3.35	1087.95

대비 면적이 2.9 배 크고 임계 경로 지연 시간 또한 2.9 배 길다. Hamming-2와 Hamming-3가 이진 (Binary) 부호화 대비 1.15 배, 1.79 배 면적을 2.63 배 2.7 배 지연 시간을 가진다. 실험 결과에 의하면 오류 정정이 필요한 경우 해밍 길이 3을 가지는 (Hamming-3) 부호를 적용하고, 오류의 영향력이 적고 저면적 구현이 요구되는 경우 해밍 길이 2를 가지는 (Hamming-2) 부호를 적용하는 것이 용이하다. 하드웨어 설계자는 제약 조건에 맞는 고장 허용 상태 기계를 설계하여 시스템의 성능을 최적화 할 수 있다.

ACKNOWLEDGEMENTS

이 논문은 2019년도 과학기술정보통신부 여대학원생 공학연구팀제 지원사업, IDEC, 국가보안기술연구소 (2019-0424-01)의 지원을 받아 수행된 연구임.

참고 문헌

- [1] G. Burke and S. Taft. "Fault tolerant state machines." in *Proceedings of the Military and Aerospace Programmable Logic Devices Workshop (MAPLD' 04)*. Jet Propulsion Laboratory, Pasadena, CA. 2004.
- [2] M. Berg, "A Simplified Approach to Fault Tolerant State Machine Design for Single Event Upsets," *Mentor Graphics Users' Group User2User Conference*, 2004.
- [3] C.E. Cummings, "State Machine Coding Styles for Synthesis," *SNUG (Synopsys Users Group) 1998 Proceedings*, section-TB1 (3rd paper), March 1998.