論文2001-38SD-8-9

마이크로파이프라인 회로를 위한 지연 고장 테스트

(Path Delay Testing for Micropipeline Circuits)

康 容 碩 *, 許 璟 會 **, 姜 成 昊 *

(Yong-Seok Kang, Kyung-Hoi Huh, and Sungho Kang)

요 약

마이크로파이프라인 회로의 모든 연산 소자의 타이밍은 아주 중요하다. 스캔 플립플롭을 이용한 경로 지연고장 테스팅에 관한 기존 연구들은 두 개의 테스트 패턴 중 두 번째 패턴의 조절용이도가 높아야 한다는 점을 간과하였다. 본 논문에서는 작은 면적 오버헤드로 마이크로파이프라인 회로의 경로 지연고장을 테스트할 수 있는 새로운 스캔 래치 및 테스트 방법을 제안하였다. 새로운 스캔 래치를 사용하여 마이크로파이프라인의 경로지연고장을 테스트한 결과에서 기존연구에 비해 높은 경성 경로 지연고장 검출율을 얻었다. 또한 제안된 스캔 래치는 마이크로파이프라인의 고착고장 검출을 위한 BIST로 응용을 확대하기 쉽다.

Abstract

The timings of all computational elements in the micropipeline circuits are important. The previous researches on path delay testing using scan methods make little account of the characteristic of the path delay tests that the second test pattern must be more controllable. In this paper, a new scan latch is proposed which is suitable to path delay testing of the micropipelines and has small area overhead. Results show that path delay faults in the micropipeline circuits using the new scan are testable robustly and the fault coverage is higher than the previous researches. In addition, the new scan latch for path delay faults testing in the micropipeline circuits can be easily expanded to the applications such as BIST for stuck-at faults.

Ⅰ. 서 론

비동기 회로는 동기 회로에 비해 많은 장점을 갖는다. 각 회로 부분별로 동작을 수행해야 하는 부분만이 동작할 수 있는 구조를 갖기 때문에 저전력 설계가 가능하다. 또한 동기 회로의 경우 전역 클럭을 사용하여 데이터 처리 시간이 최악의 경우(worst-case)인 것을 가정하여 전역 클럭의 주파수를 결정하기 때문에 일반적으로 회로의 성능 이하로 동작하게 된다. 하지만 비동기 회로의 경우 자체적으로 데이터 처리 과정을 모니터링하여 동작을 수행하기 때문에 평균적인 경우(average-case)로 동작하여 고속 동작에 적합하다. 비동기 회로는 전역 신호가 없기 때문에 회로 설계를 모듈별로 할 수 있으며 전체적인 시스템 또한 모듈화되어 동작한다. 따라서 설계 시간 및 비용을 절약할 수 있다. 그리고 clock skew와 고속 클럭에 따른 전자기파(EMC)의 영향을 최소화 할 수 있는 구조를 갖추고 있다. 이러한 여러 장점에도 불구하고 비동기 회로를 설계하는데 가장 중요한 문제는 해저드와 레이스를 신중하게 고려해야 하는 것이다. 따라서 이 분야 연구의 주된 관심은 회로 합성과 검증 기술에 집중되어 왔다. 반

* 正會員, 延世大學校 電氣電子工學科
(Dept. of Electrical & Electronic Eng. Yonsei University)
** 正會員, 一眞電子部品研究所
(IL JIN Technology Center)
接受日字:2001年1月17日, 수정완료일:2001年7月16日

면에 제조된 비동기 회로가 물리적인 고장을 가지고 있는가를 효율적으로 확인하기 위한 테스트 기술에 대해서는 거의 무관심하였다. 하지만 비동기 회로의 크기가 커지고 상업적인 제품이 사용되기 시작하면서 테스팅은 중심적인 문제로 부각되기 시작하였다.[1~4]

비동기 회로에 대한 여러 관점들은 동기 회로보다 테스트하는 것을 더욱 어렵게 만들었다. 비동기 회로는 일반적으로 전역 신호를 갖고 있지 않기 때문에 회로 전체의 조절용이도(controllability)는 급격히 감소되어지고 따라서 일반적인 동기 회로의 테스트 방법인 "single stepped" 방식을 쉽게 취할 수 없다. 또한 비동기 회로는 동기 회로보다 상태유지 소자를 많이 갖고 있다고 간주되므로 테스트 벡터를 생성하는 것이 좀 더 어렵고 테스트 용이화 설계 기법들은 커다란 면적 오버헤드를 갖는다. 또한 비동기 회로의 특성상 비동기 회로는 고장이 있을 시에 해저드와 레이스를 갖게되고 이러한 지연 고장들은 기존의 동기 회로에서의 고착 고장 테스트 및 지연 고장 테스트 방법으로 검출하기 어려운 것으로 잘 알려져 있다.[5~11]

마이크로파이프라인 회로[12]는 비동기 아키텍처의 한 종류로 실제 회로에 많이 응용되고 있는 비동기 설계 방식이다. 마이크로파이프라인 회로는 일반적으로 다발 데이터(bundled data)를 가진 delay-insensitive 방식으로 분류된다. 하지만 실제적으로 delay-insensitive 제어회로에 의해 bounded delay 부분으로 구성된다. Delay-insensitive 모델은 소자와 와이어(wire)의 지연이 한계를 갖지 않는 방식을 말한다. 따라서 delay-insensitive 모델을 사용한 비동기 회로의 경우, 정확한 입력이 들어왔다는 것을 보장하기 위해 회로가 아무리 입력신호를 기다려도 정확성을 보장할 수 없다. Bounded-delay 모델에서는 주어진 충분한 시간 안에 입력의 응답으로 하부회로가 안정화되어 새로운 입력을 보낸다는 가정을 한다.

그림 1은 일반적인 마이크로파이프라인 회로의 구조를 보여준다. 다른 회로에서 데이터 값이 왼쪽의 이벤트 레지스터(Event Reg1)에 보내진다. 요구 신호(request signal)가 제어선 Rin에서 생성되면 데이터(Din)이 Reg1에 복사되며, 출력 Rout과 Ain에 신호 이벤트를 발생시킨다. 이 상태에서 이벤트 레지스터 Reg1은 응답 신호(acknowledge signal)가 입력 Aout으로 전달될 때까지 데이터를 안정되게 유지한다. Reg1에 의해 생성된 요구 신호는 다음 조합회로 블록(Logic1)의 출력이 안정화되도록 충분한 시간동안 지연된다. 두 번째 레지스터 Reg2의 입력 Rin에서 요구 신호를 받은 후에 데이터를 래치하고 출력 Ain에 이벤트를 발생시킨다. 그리고 다음 이벤트 레지스터를 위해 출력 Rout에 요구 신호를 생성시킨다. 데이터 처리과정은 나머지 마이크로파이프라인 상태에서 계속 반복된다.

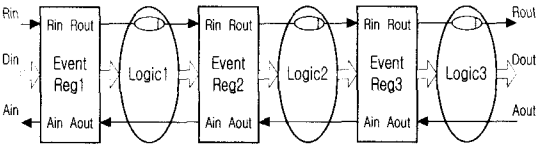


그림 1. 마이크로파이프라인 회로의 기본 구조
Fig. 1. Basic structure of micropipeline circuits.

마이크로파이프라인에 사용되는 레지스터들의 래치들에 데이터를 래치하거나 저장하는 것을 제어하는 데에는 다른 방법들이 이용된다. '통과(pass)'와 '포착(capture)'[Sutherland]이라는 두 가지 제어 신호를 기본적으로 사용한다는 점인데, 초기 상태에서 모든 레지스터 래치들은 각각의 래치 천이 제어 프로토콜에 따라서 입력값을 출력단으로 보낼 수도, 보내지 않을 수도 있다. 따라서, 새로운 데이터가 상태 레지스터에 의해 래치되지 않는 한 데이터 경로에 어떠한 천이도 일어나지 않는다는 점에서 '일반적으로 닫힌(normally closed)' 래치를 사용하는 것이 유리하다고 할 수 있다. 마이크로파이프라인 회로에서 비록 bounded-delay 비동기 회로에서 나타나는 헤저드(hazard)를 고려해야 하는 것이 제거되었으나, 여전히 제어 경로상에 외부적인 지연 소자를 추가함으로써 최악의 경우(worst-case) 성능에 맞추어야 한다. 이에 따라 조합회로 블록은 주어진 최악의 경우 시간 안에 동작하도록 되어야 한다. 따라서 마이크로파이프라인의 제어 회로의 핸드쉐이킹 신호뿐 아니라 모든 조합 소자들(computational elements)의 타이밍이 중요하다. 마이크로파이프라인의 정확한 동작을 보장하기 위해서는 이러한 타이밍에 이상이 있는 가를 테스트하여야 한다. 따라서 마이크로파이프라인 비동기회로는 반드시 조합회로 블록에 대한 경로지연 테스트가 필요하다.[13~15]

하지만 마이크로파이프라인을 테스트하기 위한 여러 방법[13~19]들이 발표되었지만 가장 중요한 고장 모델인 경로지연 고장을 효과적으로 검출할 수 있는 방법은 아직까지 개발되지 못했다. 본 논문에서는 마이크로파

이프라인 회로를 위한 지연고장 테스트를 위한 효과적인 스캔 셀 및 테스트 방법을 제안하였다. 기존의 마이크로파이프라인의 지연고장 테스트에 비해 보다 효과적이고 적은 하드웨어를 필요로하는 스캔 셀을 사용한다. 또한 benchmark 회로를 이용한 실험에서 제안한 스캔 셀을 사용하여 보다 좋은 결과를 얻었다.

Ⅱ. 경로 지연 고장 테스트

지연 고장은 자주 회로의 기능이 아닌 성능에만 영향을 주기 때문에 비동기 회로는 대부분의 지연 고장에 대해 영향을 덜 받는 구조를 지닌다. 그러나 어떤 지연 고장에 대해서는 정확성에까지도 영향을 받기 때문에 그러한 고장을 테스트 할 수 있어야 한다. 불행하게도 비동기 회로는 다음과 같은 이유들 때문에 테스트하기 힘들다. 첫째, 모든 알려진 비동기 설계 기법들은 테스트용이도에 어느 정도 손해를 보더라도 일정 수준의 무해를 사용함으로써 정확한 동작을 보증한다. 둘째, 비동기 제어 회로는 일반적으로 동기 회로보다 많은 수의 피드백과 레지스터를 포함한다고 본다. 이것은 완전 스캔을 이용하여 테스트를 수행할 경우 굉장히 비쌀 수밖에 없음을 의미한다.

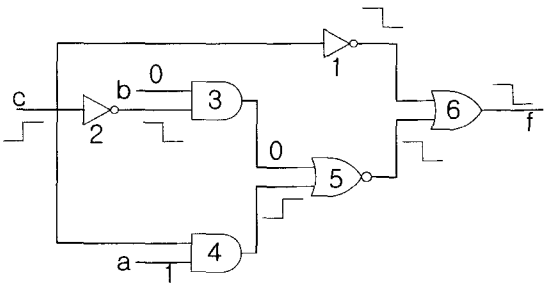


그림 2. HFRPDFT의 예
Fig. 2. Example of Hazard-free robust path delay fault testing.

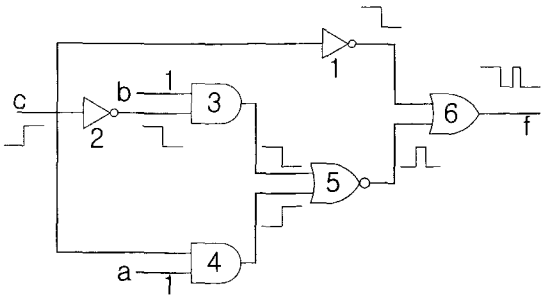


그림 3. RPDFT의 예
Fig. 3. Example of Robust path delay fault testing.

경로 지연고장을 검사하기 위해 생성된 검사입력은 여러 종류가 있다. HFR(hazard-free robust) 검사입력은 경로상의 신호가 동적 해저드(dynamic hazard)를 갖지 않고, 또한 회로의 나머지 부분의 지연과는 무관하게 경로 지연고장을 검출할 수 있음을 보장하는 쌍패턴 검사입력(two pattern test)을 말한다. 경성(robust) 검사입력은 특정한 경로의 지연 고장을 다른 회로의 지연과 무관하게 검출할 수 있다는 것만 보장하는 쌍패턴 검사입력이다. 위와 같은 검사 입력을 이용한 테스트 방법으로서 HFRPDFT(hazard- free robust path delay fault testing)와 RPDFT(robust path delay fault testing)가 있다. 이 두 방법은 그림 2와 그림 3에서 다음과 같이 설명된다. 그림 2는 HFRPDFT를 나타내는 예제로써, 경로가 {c,4,5,6}으로 정해지고, 쌍패턴 테스트 벡터인 <v0,v1>는 <100,101>(순서는 a,b,c)로 인가된 경우이다. 이 경우에는 경로를 따라 어떠한 해저드도 전파되지 않는다.

반면에 그림 3은 그림 3와 똑같은 회로의 RPDFT를 나타낸 것이다. 경로가 {c,1,6}으로 정해지고, 테스트 벡터인 <v0,v1>는 <110,111>(순서는 a,b,c)로 인가된 경우로써 그림의 파형에서 알 수 있듯이 게이트 6의 출력단에 1에서 0으로 바뀌는 동적 해저드가 존재한다. 따라서 주어진 경로에 대한 지연 고장을 검출할 수 없게 된다.

마이크로파이프라인 구조에 있어서 레지스터들은 스캔 테스트 기법을 사용하고 조합회로에 대한 경로 지연고장 검사를 수행하여야 한다. 조합 회로의 입력에 데이터가 도달했을 때, 출력 데이터는 어느 정도의 시간이 흐른 뒤에 래치되게 되는데, 이 조합 회로에 존재하는 지연 고장은 제어 신호와 데이터 신호가 분리되어 있는 마이크로파이프라인에 있어서 오동작을 일으키는 원인이 될 수 있다. 다시 말해, 조합 회로의 출력이 안정화되기도 전에 제어 신호가 먼저 레지스터를 래치하게 만든다면 원하지 않는 데이터가 래치에 저장되게 된다.

경로 지연 고장 테스트를 위해서는 쌍패턴 검사 입력을 가할 수 있는 효과적인 방법에 대한 연구가 선행되어야 한다. 지금까지 연구된 바에 의하면 크게 3가지 종류로 쌍패턴을 가하는 방법을 구분할 수 있다. 첫째, 한 번에 두 개의 패턴을 저장할 수 있는 확장된 스캔 기법(enhanced scan)을 사용한 것이다[20,21]. 이 방법은 회로에 포함된 래치의 수가 많으면 면적 오버헤드가

매우 큰 단점을 갖는다. 둘째, 쌍패턴 중 두 번째 패턴을 스캔 경로의 이동(shift) 동작에 의해 가하는 것이다[22]. 이 방법은 어떤 임의의 회로에 대하여 스캔 이동만으로 원하는 쌍패턴을 가할 수 있어야 하지만 그러한 회로는 거의 존재하지 않는다. 셋째, 회로의 파이프라인 특성을 사용한 방법이 있다[23]. 이 방법에서는 각각의 모듈이 지연 고장에 테스트 가능하다면 전체 파이프라인도 지연 고장에 테스트 가능하고 각 모듈들이 모든 가능한 출력 조합을 생성할 수 있다는 가정을 따른다. 이 경우에선 첫 번째 테스트 패턴은 스캔을 사용하여 원하는 값을 이동시킨 후 가하고 두 번째 패턴은 파이프라인에서 이전 단의 조합회로의 출력값을 사용하는 것이다.

기존의 마이크로파이프라인을 위한 경로 지연 고장 테스팅 방법은 위에서 말한 세 번째 방법과 유사한다. 즉 마이크로파이프라인의 경로 지연 고장을 테스트하기 위한 테스트 패턴을 <P1, P2>라 하면 이를 <P1p@P1s, P2p@P2s>과 같이 구분하여 가하는 것이다.(P1p와 P2p는 조합 회로의 입력에 가해지는 테스트 벡터, P1s와 P2s는 상태 벡터로서 초기에 상태 레지스터에 저장되게 된다). 그림 1과 같은 경우 Logic2의 경로 지연 고장을 테스트하기 위해서는 우선 첫 번째 파이프라인 단의 Reg1과 두 번재 단의 Reg2에 원하는 값을 스캔 경로를 통해 이도 시킨다. 이 후 Logic2의 주입력에 테스트 패턴 P1p과 스캔값 P2s를 가하고 첫째단의 요구 신호 Rout가 두 번째 단의 입력 Rin에 전달되어 조합 회로 Logic2가 최기화되어 안정된다. Logic2의 Aout을 통해 확인 신호를 받으면 Logic1의 출력을 Logic1의 지연 고장 테스트를 위한 두 번째 패턴으로 사용하여 Logic2에 P2p@P2s가 생성된다. 이와 함께 새로운 요구 이벤트가 Logic2의 Rin 입력으로 가해지면, 결과적으로 조합 회로 블록의 데이터 경로가 활성화된다. 만약 그 경로에 지연 고장이 존재한다면 조합 회로를 통해 나온 출력 값에는 비정상적인 천이가 존재하게 되어 Sout을 통해 확인할 수 있다. 즉, 테스트 패턴 P2p@P2s을 가하기 위해서는 Logic1의 입력값을 조절하여 Logic2의 두 번째 테스트 패턴으로 Logic1의 출력을 사용하는 것이다. 이 방법은 두 번째 테스트 패턴의 조절용이도가 떨어지는 단점을 갖게 되며 이러한 문제가 경로 지연 고장 테스트를 위해선 비효과적인 이유와 이를 해결하기 위한 방법을 IV장에 기술하였다.

Ⅲ. 기존 마이크로파이프라인 테스트 구조

실제 마이크로파이프라인에서 이슈가 되고 있는 고장들에는 다음과 같은 것들이 있다[15,17,18]. 우선, 마이크로파이프라인의 제어 부분의 고장이 있을 수 있고, 다음으로는 조합 회로 내의 고장을 들 수 있다. 마지막으로는 상태 레지스터를 이루고 있는 래치 부분의 고장이다.

제어 부분의 고장은 C-소자의 입력과 출력 부분, 요구 신호(Req)와 확인 신호(Ack)를 전달하는 선 부분의 고장으로 압축될 수 있다. 만약 제어 부분에 고착 고장이 있을 경우에는 핸드쉐이킹이 제대로 이루어지지 못해 마이크로파이프라인의 동작을 정지시킬 것이고[15], 따라서 쉽게 검출 가능하다. 조합 회로 내의 고장을 테스트하기 위해서는 우선 상태 레지스터 내의 모든 래치들을 비운 후, 마이크로파이프라인의 입력에 테스트 벡터들을 가하여 조합회로 블록의 응답과 정상 회로의 응답을 비교하면 된다. 마지막의 래치 부분의 고장은 고장이 생긴 래치를 '통과'나 '포착'모드로 영원히 빠뜨리게 된다. 따라서 각 고장에 해당하는 조합 회로의 고장과 동일한 방법으로 검출 가능하다. 이에 대한 설명은 뒤에 더 자세히 다룰 것이다.

1. 스캔 테스트 방법

스캔 테스트 기법의 마이크로파이프라인은 정상 동작과 스캔 테스트 동작의 두 가지 모드를 수행한다[14]. 정상 동작 모드에서는 스캔 테스트 기법이 추가되기 전의 동작과 같은 기능을 수행한다. 반면에 테스트 모드에서는 모든 래치들이 데이터를 하나씩 이동하는 레지스터(one shift register)와 같이 구성되어 마스터-슬레이브(master-slave) 플립플롭과 같은 일을 한다. 상태 레지스터들은 Aout이 클록 입력과 같이 사용되어 Aout의 제어에 따라 클럭되어 들어간다. 결과적으로, 테스트 패턴들은 스캔 입력을 통해 모든 마이크로파이프라인의 래치들에 저장될 수 있다. 그런 후에 정상 동작 모드로 변환하여 요구 신호(Req)만 가하면 테스트 패턴들에 따른 결과들이 상태 레지스터의 래치들에 저장되게 된다. 래치들에 저장된 값들을 관찰하고 싶을 때에는 다시 테스트 모드로 변환하여 테스트 패턴들을 가할 때와 같은 방법으로 스캔 출력 부분으로 하나씩 꺼내어 보면 된다. 이러한 테스트 기법을 이용하면 마

이크로파이프라인에 내장된 모든 고착 고장과 지연 고장들을 검출할 수 있다.

그림 4는 테스트 가능한 비동기 순차 회로에 사용된 스캔 래치의 구조로서, 두 개의 래치(L1, L2)와 하나의 MUX(Multiplexer)로 구성되어 있다.

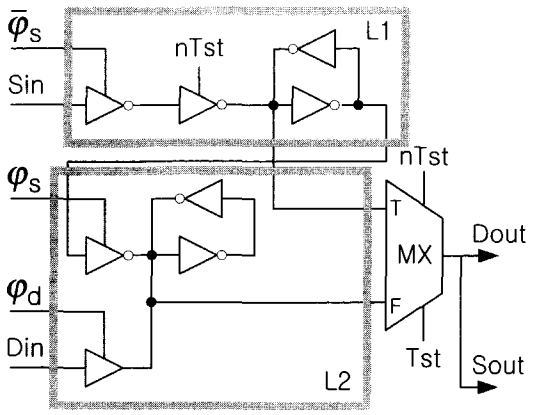


그림 4. 스캔 래치의 CMOS 구현
Fig. 4. CMOS implementation of scan latch.

정상동작 모드(Tst=0)에서는 L2의 tristate 인버터가 전송 클럭 신호(φ_s)가 0으로 고정되어 데이터의 흐름을 차단하여, L1의 출력 값은 L2로 들어오지 못하게 된다. φ_d 가 1로 되면 입력 데이터(Din)는 Dout으로 출력되어 나가고 φ_d가 0으로 바뀔 때 L2에 의해 래치된다.

스캔 모드(Tst=1)에서는 데이터 활성화 신호 φ_d는 0으로 고정되어 L2의 tristate 버퍼는 Din이 L2로 들어오는 것을 막게된다. 전송 클럭이 입력 φ_s 에 가해지면, φ_s가 1이 될 때 스캔 입력(Sin)에 안정화된 스캔 데이터는 L1에 래치된다. φ_s가 1인 동안, L2는 열려있어서 L1에 래치된 데이터들은 스캔 출력(Sout)으로 보내진다. φ_s가 0으로 바뀌면 L1에 저장된 데이터는 L2로 복사되고 스캔 경로 상의 다음 스캔 래치의 L1에 보내지게 된다.

테스트 모드(Tst=1, φ_s=0)에서 조합 회로 블록을 거쳐서 출력되는 값들이 L2에 저장된다. 이 때, L2에 저장되는 값들을 생성해낸 테스트 패턴들은 L1에 바뀌지 않고 저장되어 있게되어 조합 회로의 입력과 출력을 모두 관찰할 수 있다.

그림 5는 비동기 순차 회로의 테스트를 위한 2-bit 스캔 레지스터의 구조이다. 본래의 레지스터와 비교해 볼 때, 스캔 레지스터의 구조는 다음과 같은 4 개의 추가적인 신호들이 있다. 테스트 제어(Tst) 신호, 스캔 입력(Sin), 스캔 출력(Sout), 전송 클럭(φ_s) 들이 바로 그것이다.

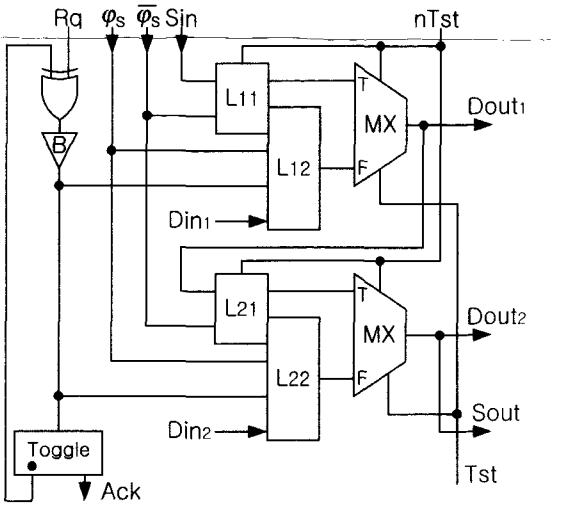


그림 5. 스캔 레지스터
Fig. 5. Scan register.

(1) 정상 동작 모드(Tst=0, φ_s=0)

초기에 모든 레지스터의 L2 래치들은 닫힌 상태로, toggle 소자의 출력은 리셋 제어 신호에 의해 0으로 된 상태에서 시작한다. 요구 신호가 레지스터의 입력단 Rq에 들어오게 되면, 모든 L2 래치들은 버퍼 B에 의해 생성된 φ_d (φ_d=1)로 인해 열린 상태로 바뀌게 된다. 따라서 Din으로 들어온 데이터는 Dout 출력으로 전송되어 나가게 된다. 그런 후에 toggle 소자의 검게 칠해진 출력부를 통해 나오는 요구 event(Rq event)가 XOR 게이트를 통해 흘러 φ_d를 0으로 만들어 L2 래치들을 닫힌 상태로 바꾼다. 그 후, toggle 소자의 또 다른 출력 부분으로 확인 신호(Ack)가 생성되게 된다.

(2) 스캔 모드

Tst와 φ_d가 0으로 유지되는 동안 Sin을 통해 스캔 데이터들이 레지스터들로 저장되는 동시에, Sout으로 출력된 스캔 데이터는 또 다른 스캔 레지스터의 입력으로 들어가게 된다. 이러한 스캔 동작은 φ_s에 가해지는 클럭 신호에 의해 이루어진다.

(3) 테스트 모드

테스트하는 동안(Tst=1, φ_s=0), 테스트 벡터들은 첫

번째 래치 L1에 저장되고 이 래치들의 출력 값들은 MUX를 통해 상태 레지스터의 출력 Dout로 연결된다. Rq로부터 요구 신호를 받은 후에 테스트 결과들은 Din 입력을 통해 L2 래치들에 저장되게 된다. 데이터가 비동기 순차 회로를 통해 루프를 형성하므로 테스트 벡터들과 테스트 결과들은 서로 다른 래치에 저장되어 테스트 동안에 보존될 수 있다.

그림 6은 테스트 가능한 비동기 마이크로파이프라인 회로의 설계를 보여주고 있다. 이 구조는 비동기 회로 부분과 스캔 테스트 제어 논리(STCL : Scan Test Control Logic) 부분으로 크게 나눌 수 있다. 비동기 순차 회로의 상태 레지스터들은 스캔 래치들로 구성되어 있고, STCL 블록은 비동기 순차 회로와 테스트를 위한 회로간의 제어를 위해 사용되었다. 또한, STCL은 전체적인 스캔 이동을 위한 클럭 신호인 φ_s를 생성한다.

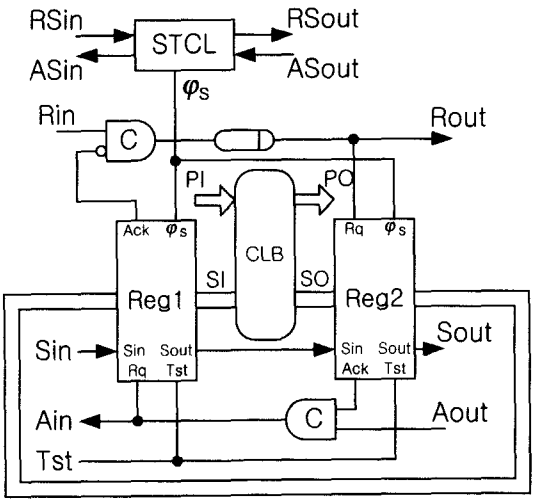


그림 6. 스캔을 사용한 비동기 마이크로파이프라인 회로
Fig. 6. Asynchronous micropipeline circuits using scan.

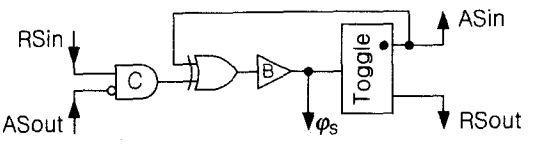


그림 7. 스캔 테스트 제어 로직
Fig. 7. Scan test control logic.

STCL 블록의 구조는 그림 7에 나타나 있다. STCL 블록은 2-상 천이 신호를 사용하며, 그림 5에서 φ_d 신호를 생성하는 방법과 같은 방식으로 φ_s및 다른 제어 신호를 생성한다. 스캔 테스트 제어 로직에 사용된 C-소자는 지연-무관성을 보장하기 위하여 사용되었다.

비동기 마이크로파이프라인 회로의 테스트를 위해 삽입된 STCL 블록은 정상 동작 모드에서는 전혀 사용되지 않는 추가적인 제어 블록이지만, 테스트하는 동안에 스캔 경로를 정확히 제어할 수 있도록 고장이 존재해서는 안된다.

STCL 블록은 지연 무관 회로이므로 STCL 내의 어떠한 고착 고장도 제어 신호를 출력부로 천이시키지 못하게 한다. 따라서 회로는 고착 고장에 대해서 모두 테스트 가능하다[2].

앞서 말했듯이 래치에는 포착-고착 고장(stuck at-capture) 과 통과-고착 고장(truck at-pass)와 같은 두 가지 종류의 고장이 존재한다. 포착-고착-고장은 tristate 버퍼나 인버터의 제어 선에 생기는 고착 고장으로 버퍼 혹은 인버터를 영원히 동작하지 못하게 만든다. 반면에 통과-고착 고장은 각각의 3상 소자들을 영원히 동작하게 만든다. 이러한 고장들은 하나의 스캔 레지스터에 속해있는 래치들을 통해 0에서 1로 바꾸어 가며 이동시키면 검출 가능하다. 즉, 고장이 존재하는 스캔 래치와 그 전의 스캔 래치의 상태가 테스트 모드에서 다르다면 L1 래치의 nTst의 입력에 1-고착 고장이 존재하는 것이다.

L2 래치의 φ_d에 존재하는 고착 고장은 스캔 모드와 테스트 모드에서 Din에 현재의 상태와 다른 논리 값을 넣어주고 Sout을 관찰하여 검출할 수 있다.

Ⅳ. 마이크로파이프라인의 경로지연고장 테스팅을 위한 새로운 스캔 셀

마이크로 파이프라인에서 단일 고착 고장을 테스트하기 위한 많은 연구가 진행되었다. 고착 고장 모델은 테스트 하에 있는 회로의 고장 동작을 나타내기 위해 가장 널리 사용되고 있다. 이러한 단일 고착 고장도 마이크로파이프라인의 구조에서 어떤 부분에서 발생하는가에 따라 세분화할 수 있다. 마이크로파이프라인에서의 고착 고장은 크게 세 가지로 구분할 수 있다.

- 래치 제어부의 고장
- 조합 논리 회로의 고장
- 래치에서의 고장

(1) 래치 제어부의 고착 고장

래치 제어 회로 상의 고착 출력 고장은 마이크로파이프라인을 halt 상태로 만든다. 마이크로파이프라인은 적어도 한 개의 step을 이동한 후 제어부에 고착 고장이 있는 입력에 존재하는 곳에서 halt 된다. 따라서 고착 고장들은 정상 동작 모드에서 쉽게 구별될 수 있다.

(2) 조합 회로에서의 고착 고장

조합 회로 내에서의 고착 고장을 설명하기 위해서 우선 하나의 가정을 필요로 한다. 즉, 마이크로파이프라인의 모든 래치들은 초기에 transparent 하다고 가정한다. 이러한 가정은 마이크로파이프라인의 래치 사이의 processing 회로를 한 개의 조합 회로로 처리할 수 있게 한다. 이러한 회로에서 단일 고착 고장을 검출하기 위해서는 이미 알려진 테스트 생성 기술들을 이용하여 테스트 벡터를 생성하여 사용하면 된다. 따라서 마이크로파이프라인의 테스트 과정은 크게 두 개의 과정으로 나눌 수 있다.

첫째, 마이크로파이프라인을 비운다. 즉, 모든 래치들은 transparent 하게 된다.

둘째, 마이크로파이프라인의 입력에 테스트 벡터를 가한 후 조합 회로의 응답을 고장이 없는 경우의 응답과 비교한다.

(3) 래치에서의 고착 고장

단일 고착 고장 : 래치의 입출력에 존재하는 임의의 고착 고장은 조합 회로에 대응하는 고장과 같게 된다. 따라서 별도로 고려하지 않아도 된다.

단일 고착-포착 고장(stuck-at-capture faults) : 래치의 단일 고착-포착 고장은 래치의 레지스터 비트를 항상 포착(capture) 위치에 있게 한다. 예를 들어, 래치의 입력인 enable 신호에 0-고착 고장은 고장난 래치를 항상 포착 모드에 있도록 한다. 이러한 고장의 영향으로, 고장이 있는 비트는 항상 1의 값을 갖던가 0의 값을 갖는 회로로 검출이 가능하다. 모든 마이크로파이프라인의 래치들이 transparent 할 때, 이 고장은 조합회로 내의 어떤 고착 고장과 상등(equivalent)하게 된다. 따라서, 고착-포착 고장은 조합 회로에 대한 고착 고장을 위한 일반적인 테스트 방법을 사용하여 검출할 수 있다.

단일 고착-이동 고장(stuck-at-pass faults) : 이 고장은 래치의 레지스터 비트가 항상 이동(pass) 모드에 있도록 한다. 래치의 enable 입력의 1-고착 고장은 고장이 있는 래치가 항상 transparent 하도록 한다. 이러한 종류의 고장을 검출하기 위해서는 두 개의 패턴을 이용한 테스트 방법이 필요하다.

경로 지연 고장(path delay faults) : 마이크로파이프라인에서 고려하는 지연 고장은 각 파이프라인 단계 사이에 존재하는 조합 회로에서 발생한다. 마이크로파이프라인 설계 방식은 제어 경로 상에 지연 소자를 추가하여 worst-case인 경우에 대해 계산된 시간 안에 조합 회로 내에서 신호가 전달된다는 가정을 한다. 즉, 지연 시간을 가정하였기 때문에 지연 고장에 대한 테스트를 반드시 수행하여야 회로의 정확한 동작을 보장할 수 있다.

잘 알려진 바와 같이 지연 고장의 테스트를 위해서는 두 개의 연속된 패턴을 가해 입력의 천이가 테스트하고 있는 경로를 주어진 시간 안에 통과할 수 있는가를 검출한다. 하지만 앞에서 언급한 지금까지의 비동기 테스트에 관한 연구는 이에 대해 만족할 만한 결과를 얻지 못 하고 있다.

마이크로파이프라인을 적용한 회로를 스캔을 이용하여 테스트하기 위해서는 경로 지연 고장 테스트를 반드시 고려해야 한다. 경로 지연고장 테스트는 그 특성상 2개의 연속된 패턴을 가해야 하므로 2개의 패턴을 가할 수 있는 방법에 대한 연구를 필요로 한다. 우선 스캔 구조를 변경하여 2개의 패턴을 모두 저장하여 사용할 수 있도록 하는 방법을 생각할 수 있다. 하지만 이는 많은 면적 오버헤드를 필요로 하므로 실제로 적용할 수 없어 거의 사용하지 않는다. 스캔을 이용한 파이프라인 설계를 위한 기존 연구[14,15,18]는 스캔을 사용하여 동기 순차회로를 테스트하는 방법을 적용하였다.

그림 1에 나타낸 파이프라인에서 Logic2의 경로 지연고장을 기존의 알고리듬을 적용하여 테스트하는 방법을 다음과 같다. 이 방법을 적용하기 위해서는 세 개의 상태 레지스터를 갖아야 테스트를 수행할 수 있다. 우선 테스트 패턴 p3와 p1은 각각 Reg1과 Reg2에 스캔 경로를 통해 저장된다. 그리고 테스트의 결과는 Reg3에 저장된다. 상태 레지스터에 테스트 패턴이 로드되면 조합 회로는 p1에 의해 초기화된다. 이후 두 번째 패턴은 p3가 Logic1을 통해 나온 패턴을 사용하여 Logic2에 가하게 된다. 이 패턴에 의해 Logic2의 데이터 경로가 활성화되고 고정된 시간 후에 Logic2의 응답이 Reg3에 래치되어 조합 회로의 고장 유무를 스캔으로 뽑아내 확인하다.

이와 같은 방법을 사용하게 되면 두 번째 패턴의 조절 용이도가 낮아지게 된다. 그 이유는 Reg1에 저장된 값을 그대로 사용하는 것이 아니라 Logic1의 출력값을

p2로 사용하게 되기 때문에 우리가 원하는 p2값을 얻기가 힘들다. 이는 특히 경로 지연 고장 테스트를 위해 사용되는 경성 지연 고장 테스트의 경우에 낮은 고장 검출률을 얻게 된다. 그 이유는 표 1에 나타낸 바와 같이 경성 지연 고장의 효과적인 테스트를 위해서는 두 번째 패턴의 값이 0 또는 1의 값을 갖는 경우가 많기 때문에 첫 번째 패턴에 비해 높은 조절 용이도를 가져야 한다. 동기 순차 회로에서 되먹임(feedback)이 있는 경우에는 플립플롭의 초기화가 테스트 검출률에 많은 영향을 미치기 때문에 첫 번째 패턴을 스캔 경로를 통해 가하는 것이 효과적이지만 마이크로파이프라인은 그 구조적 특성상 되먹임에 의해 발생하는 테스트 가능도가 낮아지는 문제가 거의 없다.

표 1. 경성경로지연 고장 테스트를 위한 경로외 입력 조건

Table 1. Offpath constraints of robust path delay fault testing

소자 종류	경로상의 전이	경로외 입력 제약조건
AND/NAND	Rising	X1
	Falling	11(no static hazard)
OR/NOR	Rising	00(no static hazard)
	Falling	X0

따라서, 본 연구에서는 두 번째 패턴을 스캔 경로를 통해 입력받고 첫 번째 패턴을 이전 단계의 조합 회로의 출력을 사용할 수 있도록 하는 새로운 스캔 래치를 제안한다. 이 방법을 사용하면 보다 효과적인 경로지연 고장 테스트가 가능하다.

새로 제안한 스캔 래치는 그림 9에 나타내었다. 그림 9에서 점선으로 표시된 부분의 안쪽에 있는 것이 1 비트의 스캔 래치로 두 개의 스캔 래치로 연결된 구조를 나타내었다.

그림 8에 나타난 스캔 래치의 동작 특성은 다음과 같다. 우선 스캔 이동 동작은 sck 신호에 의해 이루어진다. 이 때, tm 신호는 low 값을 갖는다. 이 경우 스캔 래치에 있는 두 개의 래치는 마스터-슬레이브 플립플롭과 같은 동작을 하게 되어 스캔 경로를 통한 스캔 이동을 수행할 수 있다. 그리고 Dout 신호에는 래치에 저장된 값이 나오게 된다. tm 신호가 high이고 sck 신호는 high를 유지하면 Dout에는 Din의 값이 나오게 되며 L2에 Din의 값이 저장된다. 이는 테스트 결과를 스캔이동을 통해 확인할 수 있도록 한다. 그리고 Din의 신호가 멀티플렉서를 통한 지연 시간을 최소화하기 위해 래치의 배치를 임계 경로에서 제외 시켰다.

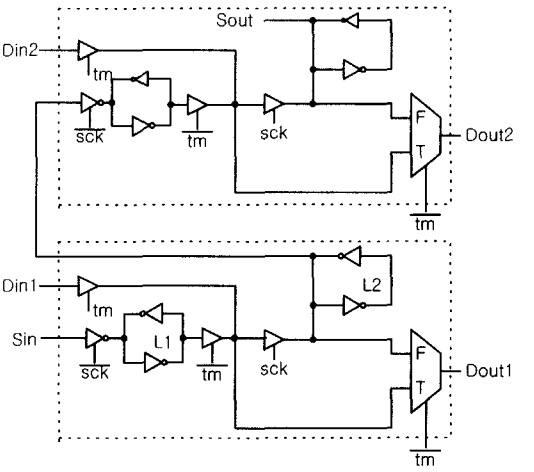


그림 8. 새로 제안한 스캔 래치

Fig. 8. New scan latch.

그림 9에 실제 마이크로파이프라인의 경로 지연고장 테스트를 위한 제어 신호의 동작 특성과 테스트 패턴의 생성 방법을 설명하였다. 우선 제어 신호의 동작 특성을 살펴보자. Logic2의 경로 지연 고장을 테스트하기 위해 우선 스캔 경로를 통해 스캔이동으로 그림과 같이 두 패턴 p3와 p2를 Reg1과 Reg2에 각각 입력한다. 이때 tm 신호는 low 상태를 유지하고 테스트 클럭 sck에 의해 스캔 이동 동작을 수행한다. p3를 사용하여 p1를 만들기 위해 Logic1을 최기화한다. 이를 위해 tm과 sck는 모두 low 상태를 유지해야 한다. 안정된 p1값을 얻기 위해 Logic1의 처리 시간보다 충분히 긴 시간을 유지시켜준다. Logic1의 출력값, 즉 p1으로 Logic2를 초기화하기 위해 tm 및 sck를 high로 만듭니다. 이 역시 Logic2의 처리 시간보다 충분히 긴 시간을 유지시켜 Logic2가 안정된 상태를 유지할 수 있도록 한다. 이때 Reg1의 L2에는 p1이 L1에는 p2가 저장되어 있게 된다. 이제 Logic2에 p2를 가하기 위해 tm 신호를 low로 한다. 이 상태는 마이크로파이프라인의 정상 동작 모드로 정상 동작 시간은 주어진 Logic2의 지연 시간 동안 유지한다. 이 시간 안에 정확한 데이터 처리가 수행되지 않으면 경로 지연고장으로 간주되며 따라서 이 시간은 앞서의 초기화 시간에 비해 짧고 Logic2에 대응하는 지연소자의 지연값과 같다. p2에 의한 천이 신

호가 출력에 전달되었는지 확인하기 위해 tm신호를 다시 high로 한다. 이 때 Reg3에는 테스트 결과가 저장되고 이 결과값을 스캔 경로를 통해 확인하기 위해 스캔이동 동작을 수행한다.

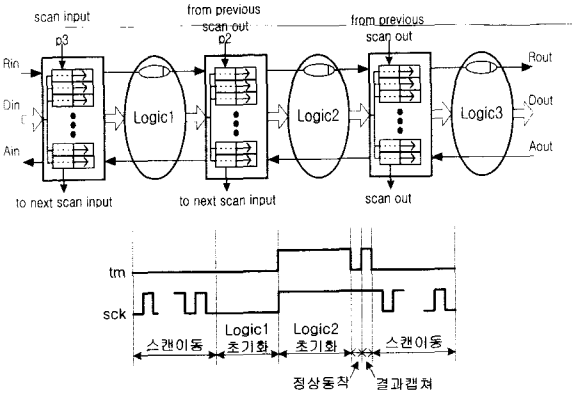


그림 9. 새로 제안된 스캔을 이용한 경로 지연 고장 테스트

Fig. 9. Path delay fault test using new scan.

V. ATPG를 이용한 마이크로파이프라인의 지연고장 테스트 패턴 생성

테스트하려는 조합 회로의 한 단계 전의 조합 회로를 통해 나오는 출력 값을 지연 고장 검출을 위한 첫 번째 테스트 패턴으로 이용할 경우, 이슈가 되는 문제는, 첫째, 우리가 원하는 지연 고장을 위한 첫 번째 테스트 패턴을 어느 정도까지 생성해 줄 수 있겠는가 하는 점과, 둘 째, 어떻게 해서 그 출력 값을 조절할 수 있겠는가 하는 것이다. 즉, 한 단계 전의 조합 회로의 입력 값에 과연 어떤 벡터를 넣어 주어야, 우리가 원하는 출력 값들이 나올 수 있는가에 대한 의문이다.

우선, 지연 고장 검출을 위한 테스트 패턴의 생성은 기존의 조합 회로에 대한 테스트 패턴 생성 방법을 이용하면 된다. 여기서 결정된 두 번째 벡터들은 스캔 입력을 통해 원하는 상태 레지스터의 래치들에 저장될 수 있다. 관심을 갖게 되는 첫 번째 패턴들은 주 입력을 제외한 조절 포인트들이 모두 한 단계 전의 주 입력들과 스캔 입력들이기 때문에, 실제로 조절 용이도가 많이 떨어진다고 볼 수 있다.

마이크로파이프라인에서의 지연 고장 검출을 위한 첫 번째 테스트 패턴의 조절 용이도를 알아보기 위해서 생각할 수 있는 방법은 ATPG의 고장 검출률을 이용하는 것이다. 이를 위해서 테스트하고자 하는 조합 회로의 바로 전 단계의 조합 회로의 출력 부분을 생성된 지연 고장을 위한 첫 번째 테스트 패턴에 따라 수정한 후, 출력 부분에 고장을 삽입한다. 즉, 원하는 '1'인 테스트 패턴을 요구하는 입력에 해당하는 전 단계의 출력 부분들과, '0'이 필요한 입력에 해당하는 전 단계의 출력 부분들에 인버터를 추가하여 하나의 AND 게이트의 입력에 모두 연결한다. 만약 원하는 테스트 패턴이 1이나 0과 상관없이 don't care 값 'X'인 경우에는 그 출력 부분은 수정하지 않고, 이제 AND 게이트의 출력 부분에 0-고착 고장을 삽입하면 된다.

ATPG는 삽입된 0-고착 고장을 검출할 수 있는 입력 패턴들을 생성하기 위해, AND 게이트의 출력부를 '1'로 만들 수 있는 입력 패턴들을 생성하게 된다. 이렇게 되면, 결과적으로 테스트 하고자 하는 조합 회로의 입력 부분에 지연 고장을 위한 첫 번째 테스트 패턴들을 생성 가능한지, 혹은 가능하지 못한지를 쉽게 알 수 있다. 실제로, 테스트하려는 조합 회로의 전체 경로에 대한 지연 고장 테스트 패턴을 생성하면 검출률이 100%까지 미치지 못하므로 전체 경로보다 적은 수의 경로에 대한 테스트 패턴들이 생성된다. 즉, 테스트 가능한 경로에 대하여 얻어진 첫 번째 패턴들을 가지고 위와 같은 실험을 하면, ATPG는 전 단계의 입력들을 조절하여 원하는 첫 번째 패턴들을 얻을 수 있는지를 고장 검출률로 나타내게 된다.

여기서 고장 검출률이 의미하는 것은 테스트하려는 조합 회로의 지연 고장 검출 가능한 경로에 대하여, 첫 번째 테스트 패턴의 생성이 가능하여 실제 검출 가능한 경로의 비이다. 또한, ATPG는 원하는 테스트 패턴을 생성하기 위한 전 단의 조합 회로의 입력 벡터도 생성해 주므로, 이 값들을 전 단의 상태 레지스터의 래치들에 스캔 입력을 통해 저장 시켜, 실제 테스트 생성에 이용할 수 있다.

ATPG를 이용한 마이크로파이프라인의 지연 고장 검출 패턴 생성을 위하여 다음과 같이, 마이크로파이프라인의 각각의 단계에서 같은 조합 회로를 사용했다고 가정을 하여 간단한 실험 예를 들었다. 이렇게 가정한 이유는 동기 회로에서 순차 회로를 각각의 시간에 따라 계속 전개해 나가면 마이크로파이프라인의 기본 구조와 비슷해지기 때문이다. 이와 같이 순차 회로를 전개한 그림이 그림 10에 보여지고 있다.

그림 11은 S27 벤치 회로인데, 이와 같은 순차 회로

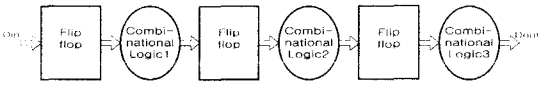


그림 10. 순차 회로의 전개
Fig. 10. Expansion of sequential circuits.

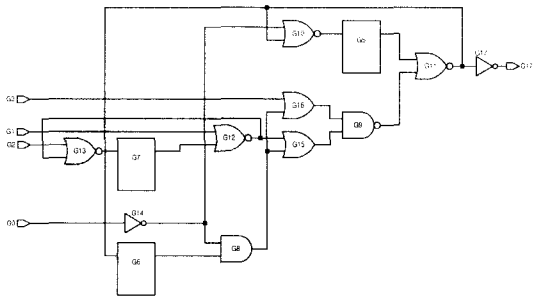


그림 11. S27 벤치 회로
Fig. 11. S27 benchmarck circuits.

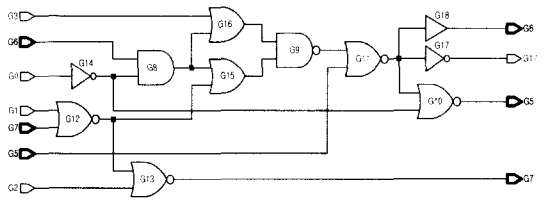


그림 12. CS27 벤치 회로(S27의 조합 회로 블록)
Fig. 12. CS27 benchmark circuits(Combination logic block of s27).

를 그림 10과 같이 전개하기 위해 조합 회로 블록과 플립플롭으로 분리해내면 플립플롭을 제외한 조합 회로의 구조는 그림 12와 같이 된다. 따라서 이러한 CS27 조합 회로가 순차적으로 이어진 마이크로파이프라인을 생각할 수 있다.

이제, 테스트하려는 조합 회로에 지연 고장 검출을 위한 테스트 생성을 할 수 있고, 생성된 첫 번째 테스트 패턴을 가지고 ATPG를 이용해 한 단계 앞의 조합 회로의 입력 패턴을 얻을 수 있다. 만약, 그림 12에서 요구되는 첫 번째 패턴의 벡터가 10X(G18, G10, G13, G17은 주출력)이었다고 한다면, G18의 출력은 바로 AND 게이트의 입력으로 연결하고, G10에는 인버터를 추가하여 AND 게이트에 연결한다. G13에 요구되는 값은 X(don't care)이므로 회로 상의 변화는 가하지 않는다.

그림 13의 색칠된 부분이 바로 요구되는 첫 번째 패턴에 따라 ATPG 상에서 추가된 게이트들이다. 마지막으로 AND 게이트의 출력 부분에 0-고착 고장을 삽입하여 ATPG를 수행하면, 원하는 테스트 패턴의 벡터를 생성해낼 수 있는 입력 벡터들을 구할 수 있다.

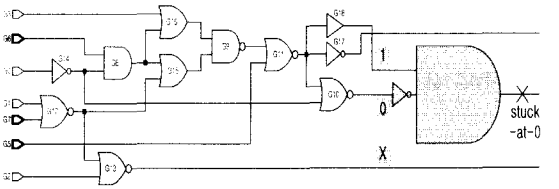


그림 13. 출력 부분을 원하는 패턴에 따라 수정한 후의 S27 조합 회로
Fig. 13. Combination circuits of S27 after output modification.

VI. 결 과

우선 마이크로파이프라인과 같은 구조를 ISCAS89 벤치마크 회로로 구성하기 위해 4장에서와 같이 동기 회로를 ATPG로 이용하여 실험을 하였다. 기존 방법[14,18]과의 비교된 결과를 표 2에 나타내었다. 경로수는 총 경로수가 5000개 이하인 경우 회로내의 모든 경로를 사용하였고 5000개 이상인 경우는 임의로 5000개의 경로를 생성하여 사용하였다. 표 2의 세 번째 칸은 기존 방법으로 검출할 수 있는 경로지연 고장을 말하는 것이고 네 번째 칸은 제안한 방법으로 검출할 수 있는 경로지연고장을 표시하였다. 즉, s420의 경우 총 경로수 738개 가운데 제안한 방법을 사용하면 594개의 경로지연 고장을 검출할 수 있는 반면 기존 방법을 사용하면 411개의 경로지연고장만을 검출할 수 있다. 따라서 결과에서 보듯 제안된 방법이 모든 회로에서 높은 고장 검출율을 나타내었다.

지금까지 설명한 마이크로파이프라인의 ATPG를 이용한 지연 고장 검출 패턴의 생성 방법을 보면, 굳이 비동기 마이크로파이프라인에만 적용할 수 있는 기법이 아니라는 것을 알 수 있다. 동기 순차 회로의 경우에는 플립플롭이 전 상태의 값들을 저장하고 있어 테스트하기 어려운 것으로 알려져 있다. 따라서 지금까지는 플립플롭을 스캔 플립플롭으로 변환하여 기능적 검증 방법[25]을 적용하는 것이 일반적이다. 기능적 검증 방법은 지연 고장 검출을 위한 첫 번째 패턴 벡터들을 스캔 플립플롭의 이동(shift)을 이용하여 원하는 위치에 가져다 놓고 클럭을 가해 조합 회로의 입력으로 사용

하고 조합 회로를 통해 나오는 출력 값들을 플립플롭에 저장하여 이 값들을 두 번째 테스트 패턴으로 사용하는 방법이다.

표 2. 마이크로파이프라인을 위한 경로지연 고장 테스트 결과 비교

Table 2. Comparison of robust path delay test for micropipelines.

회로	경로수	기존 방법[14,18]	제안 방법
s420	738	411	594
s444	1070	536	547
s510	738	716	722
s526	820	664	694
s713	5000	3888	3889
s820	984	906	950
s832	1012	912	953
s838	2018	1194	1652
s953	2312	1949	1206
s1196	5000	3566	3577
s1238	5000	4761	4761
s1423	5000	4456	4619
s1488	5000	4381	4755
s1494	5000	4351	4733
s9234	5000	3981	4026
s13207	5000	1070	1070
s15850	5000	3870	4011

그런데, 마이크로파이프라인에 적용한 방법을 보면 기능적 검증의 방법의 순서를 뒤바꾼 것과 같은 것으로 생각할 수 있다. 전 단계의 조합 회로(그림 1의 Logic1)가 테스트하려는 조합 회로(그림 1의 Logic2)와 같다고 생각하면, 앞서 언급한 것과 마찬가지로 마이크로파이프라인은 순차 회로를 전개한 것과 같기 때문이다. 다만, 다른 것이 있다면 마이크로파이프라인에서는 전 단계의 입력(그림 1의 Logic1 입력)은 모두 조절 가능한 것에 비해, 동기 순차 회로에 있어서는 오직 주입력들만 조절 가능하다는 것이다. 플립플롭의 출력이 조합 회로(그림 1의 Logic2)의 입력이 되기 때문에, 두 번째 테스트 패턴 벡터들을 본 논문에서 제안한 스캔 플립플롭에서 가지고 있기 위해서 스캔 플립플롭의 출력을 원하는 데로 조절할 수 없다.

따라서, ATPG 적용해서 동기 순차 회로의 원하는 첫 번째 테스트 패턴 벡터를 얻기 위한 주입력 값들을 생성하기 위해서는 플립플롭에서 들어오는 입력 부분에 'X' 값만을 갖도록 제약 조건을 주어야 한다. 따라서 플립플롭 수와 주입력 수와의 관계가 테스트 생성이 얼마나 가능할 것인가를 결정할 것이다. 플립플롭의 수가 주입력 수에 비해 월등히 많아진다면 그 만큼 출력 값을 원하는 데로 조절하기 위한 조절 용이도가 떨어질 것이기 때문이다.

표 3. ISCAS89 동기 회로에 대한 실험 결과

Table 3. Results for ISCAS89 synchronous circuits.

회로	입력수	출력수	플립플롭수	게이트수	ATPG 테스트 가능 / 테스트 가능 경로 (%)	테스트 가능 경로 / 전체 경로 (%)
S27	4	1	3	11	40/50 (80%)	50/56 (89.29%)
S298	3	6	14	119	49/343 (14.3%)	343/462 (74.2%)
S344	9	11	15	160	101/611 (16.5%)	611/710 (86.1%)
S349	9	11	15	161	100/611 (16.4%)	611/730 (83.7%)
S382	3	6	21	158	151/667 (21.1%)	667/800 (83.4%)
S510	19	7	6	211	0/729 (0%)	729/738 (98.8%)
S526	3	6	21	193	112/694 (16.1%)	694/820 (84.6%)
S820	18	19	5	289	170/980 (17.4%)	980/984 (99.6%)
S832	3	6	21	158	165/984 (16.8%)	984/1012 (97.2%)
S838	35	2	32	390	0/2018 (0%)	2018/2018 (100%)
S953	16	23	29	395	120/2302 (5.2%)	2302/2312 (99.6%)
S1196	14	14	18	529	3570/3577 (99.8%)	3577/6196 (57.7%)

표 3은 ATPG를 이용한 동기 순차 회로의 지연 고장 검출 패턴 생성에 관한 실험 결과이다. 실험은 우선 순차 회로의 모든 경로를 구하고, 순차회로를 그림 12와 같이 조합회로로 만들었다. 순차회로를 조합회로로 변형하면 회로의 모든 입력을 조절할 수 있으므로 마치 순차회로에서 플립플롭들이 두 개의 테스트 패턴을 저장할 수 있도록 한 확장된 스캔 구조를 사용하는 경우와 같다. 변형된 조합회로에서 우선 모든 테스트 가능한 경로지연고장을 생성한다. 이것은 [25]에서 발표된 방법을 사용하였다. 모든 테스트 가능한 경로지연 고장 중 본 논문에서 제안한 스캔 셀 및 ATPG를 이용한 경로지연 고장 테스트패턴 생성 방법을 이용하여 검출할 수 있는 고장의 수를 표에 나타내었다. 이 경로에 대하

여 순차 회로의 지연 고장 검출을 위한 테스트 생성을 하여, 그 결과를 순차 회로 칸에 나타내었다. 그림 12와 같이 순차회로를 변형한 조합 회로의 경우에는, 순차 회로에서 구한 경로를 이용하여 경로 지연고장 검출을 위한 테스트 패턴 생성한 결과를 표시하였다. 기능적 검증[25] 방법도 동기 순차회로에선 높은 고장검출율을 얻지 못하므로 본 논문에서 제안한 방법을 구현할 수 있는 스캔 셀을 작은 오버헤드로 구현하는 것이 바람직한 경로지연 고장 테스트방법이다.

종합적으로, 마이크로파이프라인에서 본 논문에서 제안한 방법을 사용하면 보다 높은 고장검출율을 갖는 경로 지연고장 테스트가 가능하다. 또한 제안한 스캔 셀의 하드웨어 오버헤드가 기존 셀에 비해 작고 기존에 사용하던 ATPG를 사용하여 마이크로파이프라인 의 경로 지연고장을 테스트하는 것이 가능하므로 보다 효과적이고 효율적인 테스트가 가능하다. 동기 순차 회로의 경우에는 본 연구와 기존의 기능적 검증 방법이 스캔 플립플롭에 저장하는 패턴의 순서만 바뀐 것이므로, 스캔 플립플롭의 추가되는 오버헤드를 최소화하면서 동시에 두 가지 방법을 사용할 수 있도록 한다면 상당히 효과적일 것이다.

Ⅶ. 결 론

현재 설계되고 있는 많은 비동기 회로 중 가장 많이 사용되고 있는 마이크로파이프라인 구조의 지연 고장 테스팅을 위한 스캔 셀과 테스트 방법에 관하여 연구하였다. 아직까지 마이크로파이프라인을 위한 효과적인 경로 지연 고장 테스트를 위한 테스트 용이화 기법 연구는 미미한 상태인데, 여기서 제안한 새로운 스캔 구조와 ATPG를 이용한 테스트패턴 생성방법을 사용하여 보다 효과적이고 높은 고장 검출율을 얻을 수 있는 방법을 제안하였다. 실험을 통해 본 논문에서 제안한 방법이 작은 하드웨어 오버헤드로 높은 고장 검출율을 얻을 수 있었다. 또한 본 연구를 통해 마이크로파이프라인의 지연고장 및 고착고장 검출을 위한 내장된 자체 테스트 기법에 적용하기 용이한 장점을 갖는다.

참 고 문 헌

[1] A. Martin. "From communicating processes to delay-insensitive circuits," Technical report, Dept. of Computer Science, Caltech, 1989.

[2] P. A. Beerel and T. H.-Y. Meng. "Semi-modularity and self diagnostic asynchronous control circuits," In Advanced Research in VLSI, Proc. of the 1991 Univ. of CA/Santa Cruz Conference. The MIT Press, Cambridge, MA, 1991.

[3] P. Day and J. Viv. Woods, "Investigation into Micropipeline Latch Design Styles," to be published in IEEE Trans. on VLSI circuits, June 1995.

[4] J. A. Brzozowski and C.-J. Seger. "A unified framework for race analysis of asynchronous network," Journal of the ACM, 36(1), pp.20-45, Jan. 1989.

[5] P. A. Beerel and T. H.-Y. Meng. "Semi-modularity and testability of speed-independent circuits," Integration, the VLSI journal, 13(3), pp.301-322, Sep. 1992.

[6] Gerald R. Carson and Gaetano Borriello, "A testable CMOS asynchronous counter," IEEE Journal of Solid-State Circuits, 25(40), August 1990.

[7] A. Martin and Pieter J. Hazewindus, "Testing delay-insensitive circuits," Proc. of the 1991 UC Santa Cruz Conf. pp.118-132, The MIT Press, Cambridge, MA, 1991.

[8] P. J. Hazenwindus, "Testing Delay-Insensitive Circuits," Ph.D thesis, Caltech, 1992.

[9] Marly Roncken, Emile Aarts and Wim Verhaegh, "Optimal Scan for Pipelined Testing : An Asynchronous Foundation," Philips Research Laboratories, International Test Conference, IEEE 1996.

[10] Chin-Long Wey, Ming-Der Shieh, D. Fisher, "ASCLScan : ascan design for asynchronous sequential logic circuits," Proc. of IEEE International Conference on Computer Aided Design, pp. 159-162, 1993.

[11] D. Rana, S. P. Levitan, D. A. Carlson, and C. E. Hutchinson, "A testable asynchronous systolic array implementation of an IIR filter," Proc. of IEEE CICC, pp.90-93, May, 1986.

[12] I. E. Sutherland, "Micropipelines," Communications of the ACM, Vol. 32, no. 6, pp. 720-738, June 1989.

[13] O. A. Petlin, C. Farnsworth, S. B. Furber, "Built-in self test design of an asynchronous block sorter," Department of Computer Science, The University, Manchester. 1997.

[14] O. Petlin, S. Furber, "Scan testing of micropipelines," Proc. of IEEE VLSI Test Symposium, pp. 296-301, 1995.

[15] A. Khoche, E. Brunvand, "Testing micropipelines," Proc. of International Symposium on Advanced Research in Asynchronous Circuits and System, pp.239-246, 1994.

[16] Volker Schober, Thomas Kiel, "An asynchronous scan path concept for micropipelines using the bundled data convention," Proc. of IEEE International Test Conference, pp. 225-231, 1996.

[17] S. Pagey, G. Venkatesh, S. Sherlekar, "Issues in fault modelling and testing of micropipelines," Proc. of IEEE Asian Test Symposium, pp. 132-138, 1992.

[18] O. A. Petlin, S. B. Furber, "Scan testing of asynchronous sequential circuits," Proc. of Great Lakes Symposium on VLSI, pp. 224-229, 1995.

[19] O. A. Petlin, S. B. Furber, "Built-in self-testing of micropipelines," Department of Computer Science, University of Manchester, 1996.

[20] S. Devadas and K. Kuetzer, "Synthesis and optimization procedures for robustly delay-fault testable logic circuilts," Proc. of IEEE Design Automation Conference, pp. 221-227, 1990

[21] S. Devadas and K. Keutzer, "Design of integrated circuits fully testable for delay faults and multifaults," Proc. of IEEE International Test Conference, pp.284-293, 1990

[22] J. A. Waicukauski, E. Lindbloom, B. Rosen and V. Iyengar, "Transition fault simulation," IEEE Design and Test, pp. 32-38, Apr. 1987

[23] T. Hayashi, K. Hatayama, S. Ishiyama, and M. Takakura, "Two test generation methods for sequential circuits," Proc. of IEEE International Symposium Circuits and Systems, pp.1942-1945, 1989.

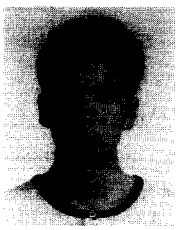
[24] S. Kang, B. Underwood and W. Law, "Path Delay Fault simultation for a Standard Scan Design Methodology," Proc. of IEEE International Conferece on Circuits Design, 1994

[25] B. Underwood, W.-O. Law, S. Kang, H. Konuk, "Fastpath: A Path-Delay Test Generator for Standard Scan Designs," Proc. of IEEE International Test Conference, pp. 154-163, 1994.

저 자 소 개

康 容 碩(正會員)

1995년 2월 연세대학교 전기공학과 (공학사). 1997년 8월 연세대학교 전기공학과 (공학석사). 현재 연세대학교 전기공학과 박사 과정 재학 중. 주관심 분야 VLSI & CAD, SoC, DfT



許 璟 會(正會員)

1998년 8월 연세대학교 전기공학과 (공학사). 2001년 2월 연세대학교 전기공학과 (공학석사). 현재 일진전자 부품연구소. 주관심 분야 VLSI & CAD, 테스팅, TFT-LCD 등임

姜 成 昊(正會員)

1986년 2월 서울대학교 제어계측공학과 (공학사). 1988년 5월 Electrical and Computer Eng., Univ. of Texas at Austin(공학 석사). 1992년 5월 Electrical and Computer Eng., Univ. of Texas at Austin(공학 박사). 1989. 11~1992. 8 미국 Schlumberger Inc. Research Scientist. 1992. 9~1992. 10 미국 The Univ. of Texas at Austin, Post Doctoral Fellow. 1992. 8~1994. 6 미국 Motorola Inc., Senior Staff Engineer. 1994. 9~1999. 8 연세대학교 기계전자공학부 조교수. 1999. 9~현재 연세대학교 기계전자공학부 조교수. 주관심 분야 테스팅, DFT, VLSI & CAD, Design Verification 등임