특집 02

융합IT 분야의 안전필수 임베디드 시스템의 신뢰성 검증을 위한 결함주입기술 동향

나종화 · 이동우 (한국항공대학교)

목 차 »

1. 서 론

최근 차세대 산업으로 주목받는 항공우주, 조선, 자동차, 철도, 및 의료 등의 분야에서 안전필수 임베디드 시스템 (safety critical embedded system or SCES)의 사용이 증가하면서 도요다 급발진 사고처럼 신뢰성에 문제가 발생하고 있다. 안전필수 임베디드 시스템의 하드웨어는 VDSM (Very Deep Sub-Micron)과 같은 최신 반도체 기술의 적용으로 고성능, 고직접, 및 저전력화가 급속하게 진행되고 있다. 이러한 하드웨어 기술의 발달을 기반으로 다양한 기능성과 편의성을 갖춘 복잡한 구조의 소프트웨어가 사용된다. 그러나 VDSM 공정이 적용된 하드웨어는 이전의 시스템에 비해 잡음여유도(noise margin)축소, 누설 전류(leakage current)의 영향증가 등으로 결함에 더 취약한 단점이 있다. 또한 소프트웨어는 많은 기능이 추가되면서 V&V는 더욱 어려워졌다. 안전필수 임베디드 시스템에서 신뢰성은 인명과 직결되므로 고장감내기능의 운용 및 이의 검증이 요구되지만 비용 등의 문제 때문에 검증되지 않은 제품들이 사용되어 수년전 발생한 도요다 급발진 사고처럼 사회문제가 발생하기도 한다.

안전필수 임베디드 시스템의 신뢰성을 평가하는 기법으로는 크게 1) 확률론적 안전성 평가 기법(PSA)[1-3], 2) 정형검증 기법(Formal Method)[4-12], 3) 결함주입 기법(Fault Injection)[13-27]이 있는데 산업적으로 많이 사용되는 기법이 결함주입 기법이다. 먼저 원자력 분야에서 노심 제어 시스템에서 사용되는 확률론적 안전성 평가 방법은 결함발생에 의한 고장사건의 위험도를 체계화 하여 확률적으로 분석하는 방법이다. 정형검증은 수학적인 기반을 둔 언어를 통해 시스템의 명세와 설계를 수행하고, 수학적 증명으로 시스템의 안정성을 증명하는 방법이다. 결함주입 기법은 시스템 내에 발생할 수 있는 결함을 인의 적으로 주입하고, 결함주입 이후의 시스템의 상태를 관찰하여 시스템의 안전성능을 평가한다.

이 세 가지 방법들 중에서 결함주입기법은 개념설계에서 정의된 고장과 관련된 결함에 대하여 한정적으로 결함을 주입하고 고장을 분석하는 기법으로서 산업에서 요구하는 저비용 및 소비자가 요구하는 최소한의 안전성을 확인할 수 있다. 이러한 이유로 각 안전필수 분야마다 특화된 국제안전인증규격을 제정하여 이를 운용한다. 국제안전인증규격은 안전필수 임베디드 시스템의 기능안전성 (functional safety)을 정의한 IEC 61508을 근간으로 하여 자동차는 ISO 26262, 항공은 DO-178C(SW), DO-254(HW), 철도는 EN 50128(SW) 등의 규격을 분야별로 특화시켜서 기능 안전성의 확보를 위한 안전인증제도를 수립하고 안전필수제품의 인증획득을 요구한다. 이러한 여러 안전인증들의 공통점은 시스템수준 및 부품수준에서 위험분석을 실시하여 중요한 위험을 초래하는 고장요인 및 결함을 식별하고 이들을 극복할 수 있는 결함감내 시스템을 설계하고 이를 검증하는 것이다. 이 안전설계 절차에서 사용하는 기법이 바로 결함주입기법이다.

결함주입 기법은 결함주입 대상에 따라 개별 하드웨어보드 시작품 또는 각종 센서와 모터 등의 액츄에이터 및 제어 하드웨어들이 연동된 시작품에 결함을 주입하는 물리적/하드웨어기반 결함주입 기법과 소프트웨어 compile 또는 runtime에 결함을 주입하는 소프트웨어 결함주입 (Software implemented fault injection or SWIFI), 그리고 SystemC 및 Verilog/VHDL RTL 시뮬레이션 모델에 결함을 주입하는 시뮬레이션 결함주입 (simulated fault injection) 기법으로 구분된다.

이 세 가지 결함주입 기법들 중에서 시뮬레이션 결함주입 기법이 중요한 의미를 갖는다. 먼저 소프트웨어의 결함은 Matlab/Simulink나 SCADE와 같은 모델기반 설계기법을 사용하는 경우, 그 도구에서 지원하는 자동소스코드 도구를 이용하여 개발하는 것이 일반적이다. 이때 SCADE를 이용하여 개발된 소프트웨어는 앞서 언급한 국제안전인증규격에서 요구하는 검증절차를 만족하는 기능을 탑재하고 있다. 한편 물리적/하드웨어기반 결함주입은 한번 결함을 주입하면 시료 시스템을 교체해야하기 때문에 고비용인 문제점이 있다. 이러한 이유로 안전인증에서는 하드웨어에서 시작하여 소프트웨어에서 발생하는 오류 (hardware induced software error)가 주요 관심사이며 이는 시뮬레이션 결함주입 기법으로 분석이 가능하다.

본 논문은 안전필수 임베디드 시스템의 신뢰성을 검증하기 위한 기존연구 방법들을 살펴본다. 특히 시뮬레이션 기반 결함주입방법에 의한 안전성능 검증 방법에 대해서 설명한다. 시뮬레이션 기반 결함주입 기법은 결함주입 방법에 따라 1) 모델수정에 의한 결함주입 기법, 2) 시뮬레이터 인터페이스 함수를 사용한 결함주입 기법, 3)시뮬레이터 커널기반 결함주입기법으로 구분한다. 각 결함주입 기법들의 방법과 장단점을 설명한다.

논문의 구성은 다음과 같다. 2장에서는 임베디드 시스템의 신뢰성을 평가하기 위한 방법을 설명한다. 3장에서는 결함주입 대상에 따른 결함주입 기법을 종류별로 설명한다. 4장에서는 시뮬레이션 기반 결함주입 기법의 방법과 한 가지 사례를 설명한다.

2. 임베디드 시스템 신뢰성 평가기법

2.1 확률론적 안전성 평가 (Probabilistic Safety Assessment)

확률론적 안전성 평가 (PSA)는, 대규모의 복합적인 공학적 설비의 안전성을 정량적으로 평가하는 엔지니어링 기법이다[1]. PSA는 원자력 발전

소, 철도, 선박, 의료등 다양한 분야에서 광범위하게 사용되고 있다.

PSA는 5가지 절차에 의해 수행되어 진다. 1) 사고빈도평가 (Accident Freqency Analysis)는 기기의 고장, 인간의 실수 등 사고가 발생할 수 있는 모든 경우의 수를 분석하는 작업이다. 2) 사고발전 경위 분석 (Accident Progression Analysis)은 초기 사고로 인해 다른 계통에 미치는 영향을 분석하는 것이다. 3) 위험원 평가 (Source-Term Analysis)는 시스템에서 다루어지는 위해 물질의 특성을 분석한다. 4) 소외 영향 평가(Off-site Analysis)는 위해 물질로 인해 외부 인원에게 미치는 영향을 분석한다. 5) 위험도 평가(Risk Calculation)은 시스템 및 설비가 대중에게 미치는 위험도를 종합적으로 평가하는 것이다.

일반적으로 PSA를 통한 시스템의 안전성 검증은 평균적인 위험도만을 평가한다. 즉, 특정 순간에서의 위험도를 평가하는 것이 아닌, 연간 위험도와 같이 특정 기간의 평균 위험도를 평가한다. 따라서 시스템의 순간 위험도를 평가하기 위해서는 그 순간의 시스템 특성을 반영한 PSA 분석을 수행해야 한다. 또한, PSA 분석은 평가 시스템의 분석 자료에 대한 의존성이 높다. 즉, 분석에 사용되는 고장자료의 정확성에 따라, 분석결과에 큰 차이를 보이게 된다. 따라서 분석 자료에 대한 검증 절차가 철저히 이루어져야 한다. PSA을 적용한 사례로 원자력 분야의 KOLA-2, KOLA-3[2], 교량의 안전성 설계[3] 등이 있다.

2.2 정형검증기법 (Formal Method)

정형검증은 명세(specification)와 검증(verification)을 통해 이루어진다. 정형 명세(formal specification)는 시스템을 기술하고, 속성을 정의한다[4]. 시스템의 기술은 수학적 모델로 변환하여 논리적인 검증을 수행한다. 시스템의 속성은 기능적 행위(functional behavior), 시간적 행위(timing behavior), 성능, 내부구조(internal structure)등을 정의한다. 명세는 행위속성 정의에 효과적이다. 최근에는 시스템의 서로 다른 측면을 각각 다룰 수 있도록 서로 다른 명세언어를 통합하고 있다[4]. 대표적인 명세언어로는 순차적(sequential) 시스템에 초점을 맞춘 Z[5], VDM[6]과, 병렬적(concurrent) 시스템을 위한 CSP[7], CCS[8]등이 있다.

검증은 모델검사(model checking)과 정리 증명(theorem proving)을 통해 이루어진다. 모델검사는 시스템의 상태공간에서 모델의 무결성을 보장하는 것이다. 따라서 탐색 스페이스를 다루기 위한 데이터 구조를 만드는 것이 중요하다. 모델검사는 자동화가 가능하고, 검증시간이 빠르기 때문에, 비교적 단시간 내에 검증결과를 도출 할 수 있다. 또한 부분적인 모델검사가 가능하여, 시스템이 완벽하게 명세 되어 있지 않더라도, 부분적인 정보를 얻을 수 있다. 하지만 폭발적인 상태증가는 모델 검사의 단점으로 지적된다. 시스템의 상태 증가를 완화하기 위해, 부분적 순서정보 탐색(exploitation of partial order information), 지역성 축소(localization reduction), 의미 축소(semantic minimization)를 수행한다. 모델검사를 적용한 사례로는 IEEE std 896.1 cache coherence 프로토콜 검증[9]과 IEEE std 1596 Scalable Coherent Interface 검증[10] 등이 있다.

정리증명은 시스템과 속성을 수학적 로직의 공식으로 표현하는 기술이다. 즉 몇몇 공리와 추론규칙을 통해 정의되는 정형 시스템을 설계하고, 증명하여 시스템의 속성을 분석한다. 정리증명은 모델검사로는 다룰 수 없는 무한한 상태공간을 가지고 있는 시스템을 검증 할 수 있으며, 부분적인 자동화가 가능하다. 또한 증명과정에서 시스템에 대한 부수적인 정보를 얻을 수 있는 장점이

있다. 하지만 정리증명은 검증시간이 느리고, 잘못된 증명에 의한 시스템 검증에 결함이 발생할 수 있는 단점이 있다. 정리증명을 적용한 사례로는 펜티엄 프로세서의 STR division 알고리즘 검증[11]와 IBM의 PowerPC, System/390의 디자인 검증[12] 등이 있다.

2.3 결함 주입 (Fault Injection) 기법

결함주입 기법은 정형기법처럼 주어진 명제를 증명하지는 못하지만, 사용법이 단순하고 지정된 시험목표를 쉽게 검증할 수 있는 장점이 있기 때문에 많이 사용되는 기법이다[13]. 이 기법은 시험하려는 타겟 시스템에 인위적으로 결함을 주입하고, 에러 및 고장 상태를 분석하여 시스템의 hazard 및 고장을 분석 평가하는 방법이다. 결함주입기법은 주입 대상에 따라 1) 하드웨어 기반 결함주입 방법, 2) 소프트웨어 기반 결함주입 방법, 3) 시뮬레이션 기반 결함주입 기법으로 분류한다. 다음 장에서는 각 결함주입 기법에 대하여 설명한다.

3. 결함주입을 이용한 신뢰성 평가

3.1 하드웨어기반 결함주입 기법

하드웨어 기반 결함주입기법은 하드웨어에 직접 결함주입을 수행하고 시스템의 안전성능을 평가한다. 하드웨어 기반 결함주입 기법은 결함에 의한 시스템의 변화를 분석하기 위해서 별도의 모니터 장치를 하드웨어와 결합해야 한다. 하드웨어 결함주입 방법은 실험 방법에 따라 4가지로 분류 할 수 있다[13].

- Pin level fault injection: IC의 pin에 직접 입력 단자를 연결하여 결함을 주입.
- Bus level fault injection: 프로세서의 입출력 버스에 소켓을 연결하여 결함을 주입.
- Heavy-ion radiation fault injection: 중입자를 하드웨어에 방사하여 결함을 주입.
- Power supply fault injection: 하드웨어에 인가 전력을 낮추거나, 높여 결함을 주입.

일반적으로 하드웨어 기반 결함주입기법은 실험 속도가 빠르며, 항공우주제품에 중입자(heavy-ion)를 주입하는 것처럼 실제 사용 환경에서 발생하는 것과 유사한 결함을 최종 제품에 주입하므로 정확한 결함/고장 분석이 가능하다. 그러나 이 방법은 결함주입 시 대상 하드웨어 자체에 파손 위험성을 가지고 있으며, 결함을 삽입 할 수 있는 위치와 표현 가능한 결함이 제한적이다. 또한 실험을 수행하고 모니터링 하기위한 별도의 장비가 필요로 한다. 하드웨어 기반 결함주입 방법은 대표적으로 pin-level fault injection을 수행하는 AFIT[14], RIFLE[15], chip-level fault injection 을 수행하는 FOCUS[16], Heavy-ion radiation fault injection을 수행하는 FIST[17] 등이 있다.

3.2 소프트웨어 기반 결함주입 기법

소프트웨어 기반 결함주입 기법은 소프트웨어 코드 또는 레지스터, 메모리 값에 결함을 주입하는 방법이다. 소프트웨어의 결함은 개발과정에서 발생하며, 운용 중에 발견된다. 그러므로 소프트웨어 운용 전에 임의적인 결함주입을 통해 고장감내형 소프트웨어를 평가하는 것은 매우 유용하다. 또한 결함주입을 통해 감추어진 버그에 대한 시스템의 영향을 미리 예측하고, 이를 대비 할 수 있다.

소프트웨어 기반 결함주입 기법은 자동 결함주입 기법과 수동 결함주입 기법으로 구분 할 수 있다. 자동화 결함주입 기법은 일반적으로 4가지

모듈로 구성되어 진다. 1) 소프트웨어 분석 모듈은 프로그램의 흐름정보와 메모리 점유율 등을 분석한다. 분석 결과를 통해 결함주입이 가능한 메모리주소 또는 흐름제어 값을 산출한다. 산출된 정보는 결함주입정보라고 하며, 결함정보 데이터베이스에 저장된다. 2) 실험제어 모듈은 결함주입정보를 조회하고, 결함주입 모듈을 호출하여, 결함주입 실험을 수행한다. 3) 결함주입 모듈은 결함주입정보에 기술된 결함위치, 시간, 값에 따라 프로그램에 결함을 주입한다. 4) 결함주입 분석 모듈은 결함주입 실험 결과를 분석한다.

소프트웨어 기반 결함주입 기법은 operating system, 응용 프로그램을 평가 하는데 주로 사용되어 진다. 결함주입을 수행하기 위해 특별한 하드웨어 장비를 필요로 하지 않는 장점이 있다. 그러나 코드 수정, 시스템 내부 상황 관찰의 어려움, 일부 결함모델 표현의 어려움과 같은 단점이 있다. 대표적으로 FIAT[18], XCEPTION[19], DOCTOR[20], EXFI[21]가 소프트웨어 기반 결함주입을 사용하고 있다.

3.3 시뮬레이션 기반 결함주입 기법

시뮬레이션 기반 결함주입 기법은 가상 시스템 또는 verilog, systemC와 같은 하드웨어 기술언어로 기술된 시뮬레이션 모델을 대상으로 결함을 주입하고, 결과를 분석하는 방법이다. 시뮬레이션 결함주입 기법은 하드웨어, 소프트웨어 기반 결함주입 기법에 비해 시스템의 내부 상태를 세부적으로 관찰 할 수 있으며, 결함주입 위치와 시간, 결함 값의 제어가 용이하다. 시뮬레이션 모델은 회로를 설계하기 위한 트랜지스터(transistor) 수준의 모델에서부터 임베디드 시스템을 설계하기 위한 트랜잭션(transaction) 수준의 모델까지 기술된다. 그러므로 기술 수준에 따라 다양한 관점에서 결함주입 시뮬레이션을 수행할 수 있다. 특히 시스템의 설계 단계에서 개발되는 시뮬레이션 모델을 대상으로 결함주입을 수행할 수 있기 때문에, 개발초기에 시스템의 성능 및 신뢰도에 대한 정보를 획득할 수 있다. 이를 통해 설계와 검증에서 소요되는 비용과 시간을 대폭 줄일 수 있다.

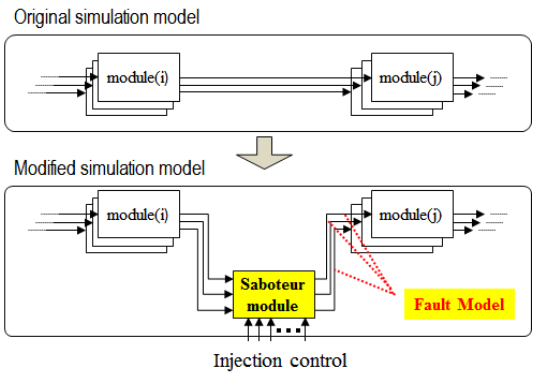
4. 시뮬레이션기반 결함주입기법의 구현방법

시뮬레이션기반 결함주입기법은 구현방법에 따라 모델수정에 의한 결함주입 방법, 시뮬레이터 인터페이스 함수에 의한 결함주입방법, 시뮬레이터 커널수정에 의한 결함주입 방법이 있다.

4.1 시뮬레이션 모델수정기반 결함주입 기법

시뮬레이션 모델수정기반 결함주입기법 (model-modified fault injection or MMFI)은 시험모델에 결함주입용 모듈을 추가하는 사보츄어 (saboteur) 기법과, 기존 모듈의 코드 일부를 수정하여 결함을 주입하는 뮤테이션 (mutation)기법으로 구분할 수 있다[22]. (그림 1)은 사보츄어 결함주입 기법을 이용하여 결함을 주입하는 원리를 설명한다. 사보츄어 기법을 시행하려면 먼저 원래의 시뮬레이션 모델을 수정하여 그림의 아래 부분에서 설명하듯이 시험모델의 입출력 포트 사이에 결함을 주입할 수 있는 결함주입 제어신호를 관리하는 사보츄어 모듈을 추가 설계한다. 사보츄어 모듈은 입력신호를 확인하고, 결함주입 제어신호가 인가되면 결함 값을 출력한다.

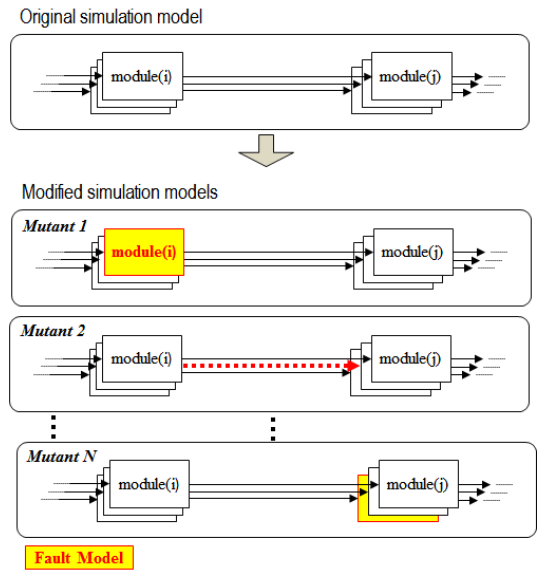
사보츄어 기법의 장점은 시험모델의 특정 신호 또는 포트에 직접적으로 접근하여 결함을 주입할



(그림 1) 사보츄어 결함주입 기법의 구조 및 동작방법.

수 있으며, 제어신호를 사용하여 다양한 형태의 결함시나리오를 시험모델에 주입 할 수 있다. 그러나 사보츄어 기법은 결함주입 모듈을 추가하는 시험모델의 특성이 변질 될 수 있다. 또한 최근의 system-on-a-chip 그리고 multicore/manycore 시스템에서는 시험모델의 복잡도가 증가하는 추세이므로 시험모델을 만드는 것도 어려울 뿐만 아니라, 시험모델을 만들더라도 시험기간등 비용이 급증하게 되므로 각종 ISO, IEC 등의 국제안전인증기준에서 요구하는 시스템 수준의 검증이 현실적으로 불가능하고 부품수준의 평가만이 가능한 문제점이 있다. 사보츄어 기법을 사용한 결함주입 시험환경으로 MEFISTO 환경을 개선한 MEFISTO-L, MEFISTO-C가 있다[23].

뮤테이션 (mutation) 기법은 시험모델의 모듈의 소스코드 일부를 수정한 뮤턴트(mutant) 모듈을 작성하고 시뮬레이션을 실시하여 결함을 주입한다. 뮤테이션 기법은 시험모델의 복잡도와는 상관없이 단일모듈 수준에서 결함모델을 적용할 수 있다. 또한 사보츄어 기법과는 달리, 결함주입 기법 적용에 따른 외부효과 (side effect)가 없다. 그러나 뮤테이션 기법은 결함주입 시나리오에 따라 다수의 뮤턴트들을 개발해야 하는데, 이러한



(그림 2) 뮤테이션 결함주입을 수행하기 위한 뮤턴트의 생성

뮤턴트들을 개발하고 관리하는 비용이 매우 높은 단점이 있다. 뮤테이션 기법을 사용한 결함주입 환경으로 ALIEN[24]이 있다.

4.2 시뮬레이터 API를 이용한 결함주입 기법

시뮬레이터 API를 이용한 결함주입 기법은 시뮬레이터 커널과 인터페이스 할 수 있는 API함수(또는 명령어)를 사용하여 결함주입 기능을 구현한다. 상용 시뮬레이터는 시뮬레이션 커널을 공개하지 않는 대신, 사용자가 커스터마이징(customizing) 할 수 있는 API 함수를 지원한다. Verilog 시뮬레이션 환경에서는 Verilog Procedural Interface(VPI)[33]를 그리고 VHDL 시뮬레이션 환경에서는 VHDL Procedural Interface(VHPI)[34]를 지원한다. VPI/VHPI의 기능은 시뮬레이션 중인 시험모델의 구조탐색 및 상태를 조회하고, 값을 수정할 있으며, 시뮬레이션 진행 상황을 제어 할

수 있다. 또한 표준 C언어를 기반으로 네트워크 통신, 쓰레드를 사용한 병렬처리, 파일 입출력 등의 고급 기능을 개발할 수 있다. VPI/VHDL의 장점은 상용시뮬레이터에서 지원하는 시스템 태스크(user system task)를 이용하여 결함주입 시험에 필요한 기능을 개발하여 시험모델 수정 없이 결함을 주입할 수 있다. 또한 VPI/VHPI 함수는 표준화 되어있기 때문에, 결함주입을 목적으로 개발한 사용자 시스템 태스크를 여러 상용시뮬레이터에서 공통으로 사용할 수 있다. 그러나 VPI/VHPI을 활용한 결함주입 환경은 API 함수가 지원하는 범위 내에서 결함주입 기능을 개발할 수 있어서 기능이 제한적인 단점을 가지고 있다. VPI/VHPI을 활용한 결함주입시험은 [27-32]에 의해 연구가 수행되었다.

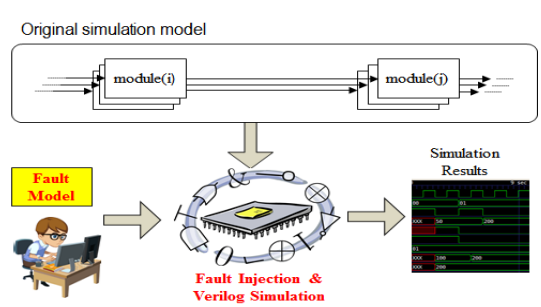
4.3 시뮬레이터 커널수정 결함주입기법

시뮬레이터 커널수정 결함주입기법 (kernel-modified fault injection or KMFI)은 시뮬레이션 커널 코드를 수정하여 결함주입 기능을 구현한다. 일반적으로 Verilog/VHDL RTL 시뮬레이션 커널은 discrete event based simulation 기법을 이용하여 각 event 별로 평가(evaluation)과 갱신(update) 프로세스를 반복적으로 실행하면서 RTL 모델을 시뮬레이션한다. 평가단계에서는 event를 수신한 프로시저의 연산을 처리하고, 갱신단계에서는 프로시저의 연산결과를 연결된 타 프로시저에 전달한다.

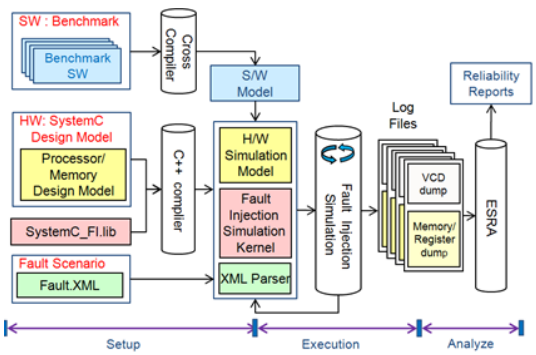
이러한 RTL 시뮬레이터가 원래의 회로설계검증기능을 유지하면서 동시에 결함주입기능을 추가한 것이 커널수정 결함주입기법 (KMFI) 이다. 이 방법은 시뮬레이터 커널에서의 메시지 후킹(message hooking) 기법을 사용하여, 결함주입 대상의 정상 값을 가로채서, 결함 값을 주입하고, 갱신단계에서 결함 값이 결함주입 대상에 적용되도록 한다.

이 기법의 운용사례를 (그림 3)에 설명하였다. 이 기법에서는 원래의 시뮬레이션 모델을 그대로 사용하는 대신에 사용자가 결함 특징 (fault attribute)를 입력하여 결함을 시뮬레이터에 미리 입력한다. 시뮬레이션이 실행되면 일반적인 RTL 시뮬레이션을 수행하다가 입력된 결함주입 event의 조건이 시뮬레이터의 상태변수와 일치하게 되면, 즉 결함이 발생하게 되면, 결함주입 프로세스를 실시한다. 이러한 반복된 절차를 종료하면 Verilog RTL 시뮬레이션의 결과인 VCD (value change dump) 화일을 결함을 주입하지 않은 시뮬레이션의 golden run VCD와 비교하여 고장을 분석한다.

시뮬레이션 커널기반 결함주입기법의 장점은 시뮬레이션 모델 변경 없이 결함주입이 가능하다. 또한 시뮬레이터 커널이 내부 루틴에 직접적으로 접근하기 때문에 결함주입에 의한 시뮬레이션 지연이 다른 시뮬레이션 결함주입 기법들과 비교하여 가장 짧기 때문에 시험성능이 제일 높다. 그러나 시뮬레이터 커널이 공개되어 있는 일부 시뮬레이터에서만 구현이 가능한 제한사항이 있다. 대표적인 시뮬레이터 수정 결함주입 기법으로는 아래 (그림 4)의 SystemC 기반의 SyFI[25]



(그림 3) 시뮬레이션 커널 수정 결함주입기법의 운용



(그림 4) SystemC를 이용한 커널수정 결함주입 시험환경

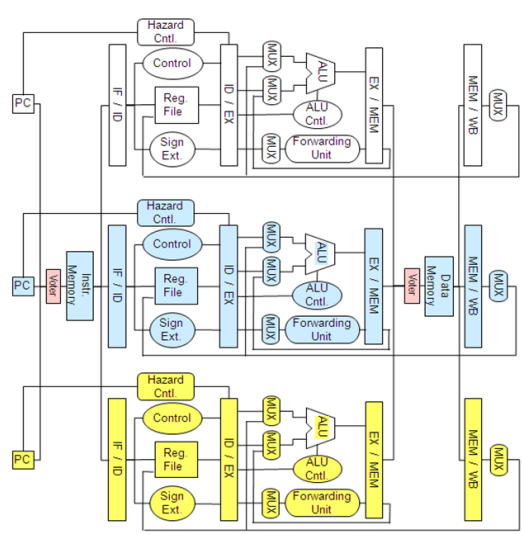
와 Verilog 기반의 VFI[26]가 있다.

4.4 시뮬레이션 결함주입 시험사례

이 절에서는 32-bit RISC 프로세서와 이의 triple modular redundant (TMR) 프로세서 구조의 SystemC 시뮬레이션 모델 둘의 신뢰성을 SystemC Kernel-based Fault Injection (SyFI) 환경을 이용하여 각각을 비교 평가하는 사례를 설명한다. 다른 시뮬레이션 모델 (예를 들면 RTL 시뮬레이션 모델) 들에 대한 평가나 다른 결함 감내 기능 (예를 들면 dual modular reduntant (DMR) 또는 error detection coding 등)을 장착한 결함 감내 시스템의 결함주입 시험평가 및 결과들은 다른 문헌에 설명되어 있다.

(그림 4)에 설명된 SyFI의 실험절차는 설정, 실행, 평가의 3단계를 통해 결함주입 시뮬레이션을 수행한다. 먼저 설정단계는 (1) SystemC 하드웨어 시뮬레이션 모델, (2) 대상 하드웨어용 소프트웨어, (3) 결함설정파일을 생성한다. 평가용으로 32bit RISC 구조인 MIPS 3000 프로세서의 SystemC 시뮬레이션 모델을 개발하고 이의 triple modular redundant (TMR) 구조의 SystemC 시뮬레이션 모델을 개발하였다 (그림 5). SW는 MIBENCH embedded benchmark에서 GSM encoder, bitcount, and CRC 32 소스코드를 컴파일하여 MIPS/TMR MIPS의 실행화일을 생성하였다. 다음에는 결함주입 시뮬레이션을 수행하기 위하여 transient stuck-at-1/0 결함모델을 정의하였다. SyFI에서 사용가능한 결함속성은 <표 1>에 정의하였다. 시험 방법으로는 White box test 및 black box test가 시행되도록 결함 발생위치, 시간은 시뮬레이션 커널의 준비단계에서 임의적으로 설정하거나, 사용자가 선택할 수 있는 기능을 수행한다. 여기서는 시스템 수준에서의 신뢰성을 평가하기 위하여 black box 방식으로 시험하였다.

2번째인 실행단계에서는 <표 2>에 요약된 co-simulation 모델과 결함모델을 이용하여 SystemC 시뮬레이션을 실행하고, 지정된 방법에 따라 시험모델에 결함을 주입한다. 시뮬레이션을 완료하면 결과파형을 기록한 VCD(value change dump) 파일과, 메모리 덤프파일, 레지스터 파일을 분석파일로 추출한다. 3단계에서는 이러한 분석파일들을 이용하여 결함의 천이과정을 분석하고 고장을 판정한다. 개별적으로 실행된 결함주입 시뮬



(그림 5) MIPS 3000 프로세서 3개를 이용한 triple modular redundant 구조의 프로세서

〈표 1〉 결함속성

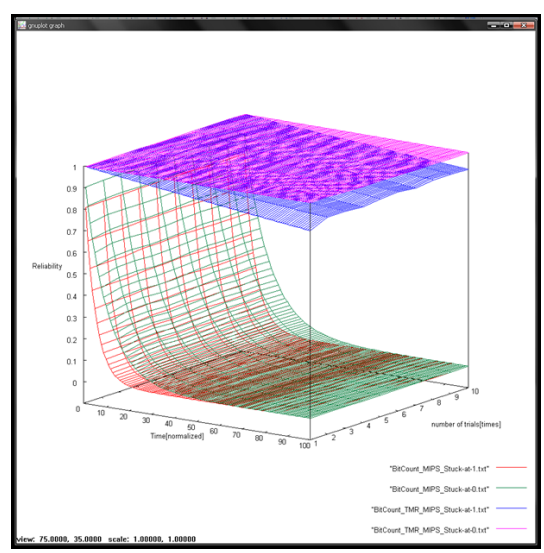
Fault Attributes	Value
Fault Type	Transient, Permanent, Intermittent
Fault Time	Random or Deterministic
Fault Location	Random or Deterministic
Fault Model	Stuck-at-0(1), Stuck-at-multi-bit, Bit flip, Open, short, Bridge
Fault Interval	periodic or aperiodic

〈표 2〉 Baseline MIPS3000 과 TMR MIPS의 결함주입 시험 타겟

Co-simulation Models	Contents	
Hardware models	MIPS processor	
	Triple modular redundancy (TMR) MIPS processor	
Software models	GSM encoder	Telecomm/ MiBench
	CRC32	Telecomm/ MiBench
	bitcount	automotive and industrial/ MiBench
Fault models	Transient stuck-at-0 fault model at random time & location	
	Transient stuck-at-1 fault model at random time & location	

레이션 결과 VCD 파일과 메모리 덤프파일을 결함주입을 수행하지 않고 산출된 golden run 시험결과와 비교한다.

결함주입 시험의 결과는 각 결함모델 별 고장률 (failure rate) z(t) 이다. 고장률과 신뢰도 함수 R(t)는 $R(t) = \exp\{-\int z(\tau)d\tau$ 의 관계가 있으므로 결함주입 시험에서 찾은 고장률을 대입하면 신뢰도 함수를 구할 수 있다. 이때 결함모델이 전체 시스템에 주입되면, 그때의 고장률은 시스템 수준의 고장률이 되며 이를 이용하여 시스템의 신뢰도를 구할 수 있다. (그림 6)에는 MIPS와 TMR MIPS co-simulation 모델의 transient stuck-at-1 과 0 결함모델의 신뢰도 그래프가 설명되어 있다. 그림에서서는 baseline 모델과 TMR 모델의 일반 상식적인 예상내용, 즉 MIPS 모델은 결함이 발생하면 곧 고장으로 시스템이 다운되는 상황이 전개되지만, TMR MIPS 모델은 90% 이상의 신뢰도를 유지한다는 사실을 검증할 수 있다. 고장률은 이와 같이 신뢰도 함수를 구하는 것뿐만 아니라 고장감내기능의 정상동작여부의 검증, 안전인증에서 요구하는 고장수목(fault tree analysis)의



(그림 6) Bit count SW를 MIPS와 TMR MIPS에서 실행하였을 경우 Stuck-at-1/0 결함에 대한 시스템의 신뢰도

base node 의 고장률을 구하는 방법 등으로 사용한다.

5. 결 론

최근 자동차, 조선, 항공우주, 철도, 의료, 로봇 등의 분야에서 융합 IT 산업을 차세대 국가산업으로 설정하여 산학연에서 융합IT 산업의 발전을 위하여 많은 노력을 기울이고 있다. 이러한 다양한 분야의 융합IT 산업에서 모두 적용되는 공통적인 특징이 바로 제품의 **기능 안전성(functional safety)의 검증을 요구하는 IEC/ISO/FAA등의 국제안전인증제도**이다. 이 국제안전인증제도는 국민의 안전을 보장하기 위한 최소한의 조건이란 명분이 있기 때문에 안전기술이 없는 산업체는 융합IT 분야의 시장진입이 불가능하게 될 것이다. 안전 때문에 기술 분야뿐만 아니라 사회의 깊숙한 분야까지 문제가 되고 있는 우리나라에서는 산업체가 안전기술을 확보하여 융합IT 제품에 적용함으로서 우리 국민의 안전을 도모하면서 동시에 선진 시장에 진입하는 것이 바람직할 것으로 사료된다.

여기서는 융합IT 분야에서 사용되는 안전필수 임베디드 시스템의 신뢰도를 평가하기 위해 연구된 다양한 기법을 살펴보았다. 여러 기법들 중에서 국제안전인증에서 요구하는 위험/고장/결함분석을 위하여 수행되어야 하는 결함주입기법을 설명 하였다. 결함주입 대상에 따른 하드웨어, 소프트웨어, 시뮬레이션 기반 결함주입 기법들을 살펴보았고, 각각의 장단점을 살펴보았다. 그중에서 최근에 많은 연구자들의 관심을 받고 있는 하드웨어에서 시작하여 소프트웨어에서 발생하는 결함과 고장을 분석하는 데 사용이 가능한 시뮬레이션 결함주입기법을 소개하였다. 여러 종류의 시뮬레이션 결함주입 기법들 중에서 가장 효율적으로 수행할 수 있는 커널기반 결함주입 기법에 대하여 소개하였다. 커널기반 결함주입 기법은 모델의 수정 없이도, 시험모델에 결함을 주입할 수 있다. 또한 상용 시뮬레이터 API를 이용한 결함주입 기법도 소개하였다. 이러한 결함주입기법은 기존의 임베디드 시스템 평가지표인 저전력, 실행시간 등과는 다르게 시스템의 안전성(safety) 및 보안성(security)등을 정량적으로 평가할 수 있어서 선진국에서는 dependability관련 많은 연구가 수행되고 있어서 많은 관심이 요구된다.

참 고 문 헌

[1] 박창규, 하재주, “확률론적 안전성 평가”, 브레인코리아, 2003.

[2] http://radar.ndsl.kr/radDetail.do?cn=GTB2009100308&topN=1 Kola 원전, 수명연장을 위해 PSA 수행

[3] 이창선, 선종완, 윤만근, 조효남, “체계신뢰성을 이용한 사장교의 확률적 안전도 분석에 관한 연구”, 2007 대한토목학회 정기학술대회, 2007, pp 2528-2531

[4] Edmund M. Clarke and Jeannette M. Wing "Formal Methods: State of the Art and Future", ACM Computer Surveys, vol. 28, iss.4, 1996, pp.626-643

[5] Spivey J M, "Introducing Z: a Specification Language and its Formal Semantics", Cambridge University Press, 1988

[6] Jones C B, "Systematic Software Development Using VDM", Prentice-Hall International, 1986

[7] Hoare C A R, "Communicating Sequential Processes.", Prentice-Hall International, 1985

[8] Milner A, "A Calulus of Communicating System", Volume 92 of Lecture Notes in Computer Science, 1980

[9] Clarke E M, Grumberg O, Hiraishi H, Jha S, Long D E, Mcmillan K L and Ness L A,

"Verification of the Futurebus+ cache coherence protocal", proc CHDL, 1993

[10] Dill D L, Drexler A J, Hu A J and Yang C H, "Protocol verification as a hardware design aid", IEEE International Conference on Computer Design:VLSI in computers and Processors, 1992, pp522-525

[11] Clarke E and Zhao X, "Verifying the SRT division algorithm using theorem proving techniques", 8th International Conference on Computer Aided Verification Number in Lecture Notes in Computer Science, 1996, pp111-122

[12] Kuehlmann A, Srinivasan A and Lapotin D P, "Verify a formal verification program for custom COMS circuits", IBM journal of Research and Development 39, 1995, pp149-165

[13] Yangyang Yu, Barry W. Johnson, "Fault Injection Techniques", Kluwer Academic Publisher, 2003, pp7-39

[14] Pedro Gil, Sara Blanc, Juan jose Serrano "Pin-Level hardware fault injection techniques", Fault inejction Techniques and tools for embedded systems reliability evaluation, 2003. pp63-79

[15] H. Madeira, M. Rela, F.MOreira, J.G. Silva "RIFLE:A general Purpose Pin-livel Fault Injector", 1st European Dependable Computing Conference, Berlin, Germany, 1994, pp199-216

[16] S. C. Gwan, "FOCUS:An Experimental Environment for Fault Sensitivity Analysis", IEEE Transactions on Computers, Vol. 41, No. 12 December 1992, pp1515-1526

[17] U. Gunneflo, J. karlsson, J. Torin, "Evaluation of Error Detection Schemes Using Fault Injection by Heavy-ion Radiation", IEEE 19th. International Symposium on Fault Tolerant Computing(FTCS-19), Chicago, MI, USA, June 1989, pp340-347

[18] Z. Segall, D. Vrsalovic, D. Siewiorek, D. Yaskin, J. Kownacki, J. Barton, R. Dancey, A. Robinson, T. Lin, "FIAT-Fault Injection Based Automated Testing environment", IEEE 18th Int, Symp. on Fault Tolerant Computing(FTCS-18), Tokyo, japan, June 1988, pp102-107

[19] J. Carreira, H. Madeira, J. Silva, "Xception:A Technique for the Experimental Evaluation of Dependability in Modern Computers", IEEE Transactions on Software Engineering, Vol. 24, N. 2, Feb. 1998, pp125-136

[20] S. Han, H. Rosenberg, K. Shin, "DOCTOR: an Integrated Software Fault Injection Environment", Internatinal Comupter Performance and Dependability Symposium, Erlangen, Germany, April 1995, pp204-213

[21] A. Benso, P. Prinetto, M. Rebaudengo, M. Sonza Reorda, "EXFI:A Lowcost Fault Injection System for Embedded Microprocessor-based Boards", ACM Transaction on Design Automation of Electronic Systems, Vol.3, No.4, October 1998. pp626-634

[22] Daniel Gil, Juan Carlos Baraza, Joaquin Gracia, Pedro Joaquin Gil, "VHDL Simulation based Fault Injection Techniques", Kluwer Academic Publisher,2003, pp159-176,

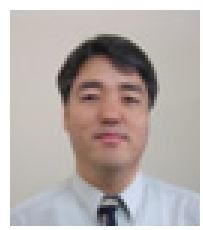
[23] Jean Arlat, Jerome Boue, Uves Crouzet, Eric Jenn, Joakin Aidemark, Peter Folkesson, Johan Karlsson, Joakim Ohlsson, Marcus Rimen, "MEFISTO:A Series of Prototype Tools for Fault Injection into VHDL Models", Kluwer Academic Publisher, 2003, pp177-193,

[24] C. Robach, M. Scholive, "Simulation-based Fault Injection and Testing using the Mutation Technique.", Kluwer Academic Publishers, 2003, pp195-215.

[25] Dongwoo Lee, Jongwhoa Na, "A Novel Simulation Fault Injection Method for Dependability Analysis" IEEE Design & Test

Computers, 2009, pp11-12
[26] Jongwhoa Na, Dongwoo Lee, "Simulated fault injection using simulator modification technique", ETRI Journal, Vol 33, No 1, 2011, pp50-59
[27] M. H. Haghbayan, A. Yazdanpanah, S. Karamati, R. Saeedi, Z. Navabi, “Generating Test Patterns for Sequential Circuits Using Random Patterns by PLI Functions”, Design & Test Symposium (EWDTS), 2010 East-West, 2010, pp456-461
[28] Das, S.R, Hossain, A. Li, J.F, Petriu, E.M, Biswas, S.N, Jone, W.B, Assaf, M.H, "Further studies on improved test efficiency in cores-based system-on-chips using ModelSim verification tool", Instrumentation and Measurement Technology Conference, 2009. I2MTC '09. IEEE, 2009, pp1132-1137
[29] C. Hescott, D. Ness and D. Lilja, "A Methodology for Stochastic Fault Simulation in VLSI Processor Architectures,"in MoBs, 2005.
[30] M. B. Santos, F.M. Gonçalves, I.C. Teixeira and J. P. Teixeira, "Defect-Oriented Verilog Fault Simulation of SoC Macros using aStratified Fault Sampling Technique", VLSI Test Symposium, 1999. Proceedings. 17th IEEE, 1999, pp326-332
[31] Zhaobo Zhang, Zhanglei Wang, Xinli Gu, Chakrabarty, K, "Physical defect modeling for fault insertion in system reliability test", Test Conference, 2009. ITC 2009. International, 2009, pp1-10
[32] Zainalabedin Navabi, “Digital System Test and Testable Design Using HDL Models and Architectures”, Springer, 2010, pp57-62
[33] "IEEE Standard Verilog Hardware Description Language: IEEE Std 1364-2001", IEEE Computer society, 2001, pp623-710
[34] IEEE Computer Society, "IEEE Standard VHDL Language Reference Manual: IEEE Std 1076c-2007"
[35] Open SystemC Initiative (OSCI), http://www.systemc.org/home

저 자 약 력



나 종 화

이메일 : jwna@kau.ac.kr

- 1985년 서강대학교 전자공학과
- 1988년 Wayne State University 컴퓨터공학과
- 1995년 University of Arizona 컴퓨터공학과
- 1998년~2005년 한세대학교 컴퓨터공학과
- 2005년~현재 항공대학교 전자공학과 교수
- 관심분야 : 고신뢰성 시스템 설계, 신뢰성 평가, 안전인증

이 동 우

이메일 : dongwoo81@kau.ac.kr

- 2006년 한세대학교 정보통신공학과 학사
- 2008년 한국항공대학교 항공전자공학과 석사
- 2008년~현재 한국항공대학교 항공전자공학과 박사 과정
- 관심분야 : 안전설계 및 검증기법, 결함주입 방법론, 안전인증