디버거 결함주입을 이용한 임베디드 시스템 신뢰성 평가방안

A Study of fault injection method using debugger equipment

이재용, 이동우, 나종화
한국항공대학교

초 록

항공, 국방, 자동차, 철도, 조선, 원자력, 플랜트 분야에서 임베디드 시스템의 활용도가 높아지고 있다. 특히 항공 임베디드 시스템은 다른 분야에 비해 고신뢰성과 고성능이 동시에 요구된다. 안전필수 인증 절차를 준수하면 설계 및 개발 단계에서 발생할 수 있는 다양한 결함을 예방 할 수 있다. 그러나 soft error와 같은 예측 불가능한 transient 결함을 대처하기 어렵다. 이러한 문제를 해결하기 위하여 고장감내 방법론과 검증방법에 대한 연구가 수행되어야 한다. 본 논문은 고장감내 기법 검증 방법론에 하나인 결함주입 기법을 제안한다. 본 논문에서 제안하는 결함주입 기법은 1) 정량적 평가 가능, 2) 시스템을 대상으로 검증 가능 같은 장점이 있다.

1. 서 론

최근 항공, 국방, 자동차, 철도, 조선, 원자력, 플랜트등 다양한 산업분야에서 융합 IT가 활발하게 발전하고 있다.[1] 이러한 시스템에서 고장이나 오류가 발생할 경우, 막대한 인명 및 재산등의 피해가 발생 할 수 있다. 따라서 이러한 시스템의 경우 고장 심각도에 대응하는 고장 감내 기법을 적용하거나 고장 검증을 통해 신뢰성을 높이는 것이 중요하다. 신뢰성을 정량적으로 평가하는 대표적인 방법으로는 failure mode effect analysis(FMEA), fault tree analysis(FTA) 등의 방법이 있다. 그러나 이러한 방법들은 순간적으로 발생하는 결함(transient fault)의 시점을 예측하기 어렵고, 결함의 발생 위치의 예측이 힘들다. 또한 시스템 내부에서의 결함주입의 영향을 분석하기 어렵다는 단점이 있다. 순간적으로 발생하는 결함은 시스템에서 구성된 회로의 어떤 곳에도 발생 할 수 있으며, 반도체 공정의 단위가 작아짐에 따라 위험도는 증가하고 있다[2]. 순간적으로 발생할 수 있는 결함은 발생 시점과 위치에 따라 masking(latching- windows, electrical, logical)[3] 되거나, 특정 메모리 영역에 상주하는 latent fault[6] 될 수도 있고, 혹은 system failure 가 될 수도 있다. 이러한 결함의 영향을 분석적으로 추적하는 것은 매우 어렵다. 본 논문에서는 고장 감내 기법을 정량적으로 평가하는 결함주입기법을 소개하고 실용적으로 사용할 수 있는 실험방법을 설명한다.

2. 본 론

2.1 결함주입 방법의 정의

결함 주입 기법의 종류는 대상 시스템에 따라 1) 하드웨어 기반 결함 주입 기법 2) 소프트웨어 기반 결함 주입 기법 3) 시뮬레이션 기반 결함 주입 기법 4) 하이브리드 결함 주입 기법으로 구분 된다[3]. 결함 주입 기법은 결함에 의한 시스템의 오동작을 확인 할 수 있기 때문에, 시스템의 안정성을 평가하기 용이하다. 그러나 기존 연구에서 제안된 결함주입 기법은 다음과 같은 제반사항이 필요하다. 1)결함주입 시험을 하기 위해 각 기법별로 전용하드웨어 또는 소프트웨어가 필요하다. 2)결함주입 실험으로 전체 개발비용이 증가한다. 하드웨어 기반 결함주입 기법의 경우 결함주입 시험과정에서 시스템의 손상될 수 있는 위험성이 있다. 3)결함주입 실험을 실행하기 위해서 각 기법에 해당하는 전문화된 인력이 필요하다. 4) 시스템 내부의 레지스터나 메모리에 접근

하기 어렵다. 5) 반복적인 실험을 하기 힘들고, 능동적으로 시스템의 취약지점을 탐색, 분석하기 힘들다.

이러한 기존 결함주입 기법들의 문제점들을 극복하기 위해 본 논문에서는 디버거 결함주입기법을 제안한다. 디버거와 시스템에서 제공하는 디버그 포트를 이용하면 실행하는 프로그램과 연동하여 하드웨어 내부의 레지스터나 메모리에 접근할 수 있다. 또한 레지스터나 메모리를 읽고, 쓰는 기능을 제공하기 때문에 시스템 외부에서 결함을 주입할 수 있다. 그리고, 스크립트 기능을 활용하면 결함주입 절차의 전 과정을 관리 할 수 있으며, 결함주입 실험을 자동화하여 반복적인 실험을 수행하고 결과를 효율적으로 분석 할 수 있다. 또한, 디버거 결함주입기법은 결함주입을 위해 별도의 실험환경을 필요로 하지 않기 때문에 실무에서 효과적으로 적용이 가능하다.

2.1.1 하드웨어 기반 결함 주입 방법

하드웨어 기반 결함 주입 기법은 실제 하드웨어를 대상으로 결함을 주입하는 방법이다. 결함주입을 위해서는 하드웨어 결함 주입환경을 제작하여야 한다. 하드웨어 결함주입 환경은 실험 대상 임베디드 시스템의 동작 특성에 영향을 주지 않는 범위에서 개발되어야 한다. 즉 결함주입 환경에 의해 대상 임베디드 시스템의 동작에 지연이 발생하거나, 결함주입 환경 부착전과 다른 동작 패턴을 보일 경우 정확한 결함주입 시험을 실시할 수 없다. 실험을 위한 하드웨어 결함 주입기의 설계 방법에는 강제방법 (Forcing Technique) 방법과 삽입방법 (Insertion Technique)이 있다. 강제방법은 하드웨어 결함 주입기와 실험 대상 시스템의 손상 없이 결함을 주입하기위해 IC 단자나 연결단자에 직접 연결해서 결함을 주입하는 방법이다. 결함주입 방법은 실험 대상 시스템에 결함을 주입하고자 하는 부분의 연결을 해제하고 하드웨어 결함 주입기를 삽입하여 결함을 주입하는 방법이다.

2.1.2 소프트웨어 기반 결함 주입 방법

소프트웨어 기반 결함 주입 기법은 소프트웨어를 대상으로 결함을 주입하는 방법이다. 결함을 주입하는 방법은 소프트웨어를 수정하여 소프트웨어 코드 또는 레지스터, 메모리 값에 결함을 주입한다. 소프트웨어 기반 결함주입 기법의 결함주입 대상은 message queueing, semaphores, shared memory와 같은 interprocess communication(IPC) 코드 또는 socket 코드에 중점적으로 주입된다. 소프트웨어 기반 결함주입 기법에서는 1)레지스터 2) 메모리 3) 중단된 네트워크 패킷 4) 복제된 네트워크 패킷 5) 오류 조건 6) 신호기 등 거의 모든 영역이 결함주입의 대상이 된다. 대표적으로 SWFI TOOLS. FERRARI, XCEPTION, GOOFI, DOCTOR, FAILURE SIMULATION TOOL(FST), FTAPE 등은 소프트웨어 기반 결함주입 기법을 사용하고 있다. [7]

소프트웨어 결함 주입 기법은 시스템의 모든 운영과 세부사항을 대상으로 신뢰성 평가가 실행되기 때문에 시간이 많이 소요되고, 실험을 많이 실행해야 한다.

2.1.3 시뮬레이션 기반 결함 주입 방법

시뮬레이션 기반 결함 주입 기법은 실험대상 시스템의 세부사항을 포함한 시뮬레이션 모델을 제작하여 결함주입 하는 방법이다. 시뮬레이션 기반 결함 주입 기법은 실험 대상 시스템과 시뮬레이션 모델이 같은 동작을 하는 것을 전제조건으로 한다. 시뮬레이션 모형은 VHDL(VHSIC Hardware Description Language) 이나 Verilog 등의 하드웨어 기술 언어를 사용하여 개발한다.

2.1.4 하이브리드 결함 주입 방법

하이브리드 결함 주입 기법은 2개 이상의 결함주입 기법을 병행해서 결함 주입 실험을 하는 방법이다.[3] 2개 이상의 결함 주입 기법의 장점만을 사용하여 분석하는 방법으로 각각의 기법들이 가지는 약점에 대한 보완을 할 수 있다. 이러한 상호작용은 결함 주입 실험을 보다 더 완벽하게 할 수 있도록 한다. 예를 들면, 하이브리드 결함주입 기법은 소프트웨어 결함 주입 기법의 다양성과 하드웨어 모니터링을 모두 포함할 수 있다. 또 다른 예로 시뮬레이션 기반 결함주입 기법은 제어와 관찰을 가능하게 하지만 시뮬레이션 기반 결함주입 기법의 단점을 완화 시켜 줄 수 있는 다른 기법과 병행하는 식으로 실험을 진행한다면

보다 더 신뢰성 있는 결과를 얻어낼 수 있을 것이다.

2.2 디버거 결함주입 방법의 소개

디버거 결함주입기법은 디버거의 메모리나 입, 출력 장치등에 접근할 수 있는 기능을 이용하여 결함을 주입하는 기법이다. SoC(System on Chip)는 메모리나 입출력 장치등과 같은 모듈들을 하나로 집적된 IC이다. SoC는 패키지 형태나 집적화 된 사이즈로 인해 내부 신호나 자원에 대한 접근이 어렵다. 그러나 디버거를 사용하면 제한된 칩의 특정영역에 대한 접근이 가능하다. 디버거 결함주입기법은 기존 결함주입기법들의 단점들을 보완할 수 있는 기능을 제공한다. 본 논문에서 제안한 결함주입기법의 장점은 다음과 같다. 디버거 기반 결함주입 기법은 1) 결함주입 기능의 독립적 구성이 가능하다. 기존의 결함주입 기법의 경우 시스템자체의 수정이 필요 했지만 디버거 결함주입 기법은 결함주입 기능을 독립적으로 구성 할 수 있다. 2) 연속적인 실험을 위한 수정이 용이하다. 디버거 결함주입 기법은 스크립트 기능을 통해 연속적인 실험을 자동화하여 수행할 수 있으며, 독립적으로 결함주입 기능이 구성되어 있으므로 수정이 용이하다. 3) 영구적인 결함주입 실험이 가능하다. 디버거 결함주입 기법은 주입된 결함을 지속적으로 유지 할 수 있으며, 일시적으로 결함을 주입할 수도 있다. 4) 실험의 대상이 되는 시스템에서 큰 수정이나 훼손 없이 결함을 주입할 수 있다. 디버거 결함주입은 디버거의 기능을 이용하여 시스템 내부에 접근할 수 있기 때문에 실험대상에 대한 별도의 수정이 필요하지 않다. 5) 결함주입 대상이 명확하다. 디버거 결함주입은 결함주입 실험의 대상이 명확하기 때문에 결과에 따른 보완이나 수정이 용이하다.

2.3 디버거 결함주입 방법의 실험

디버거 결함주입 실험방법은 임베디드 시스템의 공간, 즉 레지스터나 메모리영역을 대상으로 하여 결함주입 실험방법이다. 실험환경은 Host PC, 디버거, 실험대상이 되는 시스템으로 구성된다.

결함주입 실험 절차는 다음과 같다. 1) 하드웨어 초기화 2)결함주입 대상의 file load 3) 결함을 주입하고자 하는 구간설정 4) 결함주입 5) 프로그램 수행 6) 결과 확인 7) 결함주입 전 정상적인 프로그램 수행 결과와 비교 및 분석

본 실험에 실험대상으로 사용된 시스템은 ARM7TDMI 프로세서를 사용하는 Evaluator-7T 보드를 사용한다.

결함주입 시험을 수행하면 결함이 시스템에 미치는 영향을 판단할 수 있다. 실험을 하기 전 준비 사항은 다음과 같다. a) Host PC를 디버거를 통해 실험대상 시스템과 연결한다. b) HostPC의 ARM workbench 환경에서 실행파일인 '.axf' 파일을 생성한다. '.axf'파일은 실험대상 시스템의 초기화 설정 파일인 'init.s' 파일과 실험의 대상이 되는 소스파일에 대한 빌드과정을 통해 생성된다. c) 정상적인 '.axf'을 실험대상 시스템에 디버거를 통해 다운로드 시킨다. c) 실험대상 시스템의 정상적인 동작을 확인한다. 이러한 사전 준비는 결함 주입이 미치는 영향에 대한 기준을 제공한다. 실험에 사용된 소스코드는 임베디드용 벤치마크로 많이 사용되는 MiBench 프로그램중 "rad2deg" 프로그램을 수정하여 사용하였다. "rad2deg" 프로그램은 "180.0 * rad / (PI)" 연산과 "PI * deg / 180.0" 연산을 수행하여 연산결과를 출력하는 프로그램이다. 다음 그림은 결함 주입을 하기 전 정상적으로 동작 하였을 때 나타는 결과이다.

```
0 degrees = 0.0 radians
5 degrees = 0.87 radians
10 degrees = 0.174 radians
15 degrees = 0.261 radians
20 degrees = 0.348 radians
25 degrees = 0.436 radians
30 degrees = 0.523 radians
35 degrees = 0.610 radians
40 degrees = 0.697 radians
45 degrees = 0.785 radians
50 degrees = 0.872 radians
```

그림 1 디버거 결함주입 실험의 정상적인 결과

결함주입 기능이 동작하는지 확인하는 기본동작실험은 "rad2deg" 프로그램에서 피연산자 중 하나인 'PI' 를 대상으로 수행하였다. 'PI' 는 MiBench 프로그램중 PI 헤더파일에서 정의된 심볼릭 상수로서 프로그램에서 지속적으로 연산에 참여한다. 그러므로 연산결과가 ' 0.0 ' 또는 잘

못된 결과가 나올 것이라 예상했다. 그림2는 실험 대상이 수행하는 프로그램에서 연산의 기준이 되는 심볼릭 상수 'PI' 에 해당하는 메모리가 '0' 이 되는 결함을 주입한 후 나타나는 실험 결과를 보여주는 그림이다.

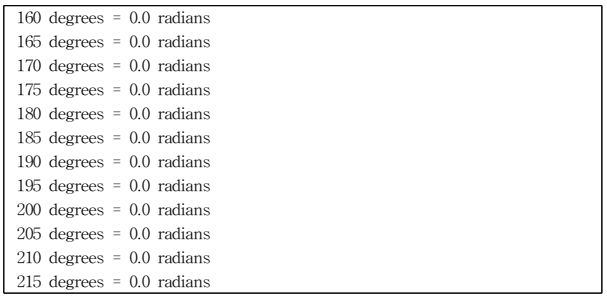

그림 2 디버거 결함주입 실험의 실험결과

디버거 결함주입 실험은 디버거에서 제공하는 메모리나 레지스터에 접근할 수 있는 기능을 이용하여 실험대상에 결함을 주입시킨 후 결함주입 전과 비교하여 실험대상의 신뢰성을 평가하는 실험이다. 신뢰성을 평가하기 위해서는 전체 메모리나 레지스터에 대하여 전체 비트 수*동작 Cycle 만큼의 실험이 필요하지만 현실적으로 시간과 비용이 많이 들게 된다. 따라서 무작위로 결함주입 대상을 선정하고 결함주입 실험을 수행한 후, 결함주입을 통해 얻은 결과를 통계적으로 분석하는 통계적 결함주입 기법이 병행 되어야 한다.

결함주입 실험을 무작위로 선정한 메모리 공간을 대상으로 반복 실험한 결과는 다음과 같다. 1) 정상적인 프로그램 수행 2) 시스템은 동작하지만 결과가 출력되지 않는 결과 3) 잘못된 연산 결과를 출력하는 결과 4) 시스템이 정지하여 제어가 되지 않아 리셋을 해야 하는 결과 5) 정상적으로 프로그램을 수행하다 어느 시점부터 출력결과가 나오지 않는 결과를 얻었다.

3. 결 론

본 논문에서는 항공분야에서 사용되는 임베디드 시스템의 신뢰도를 평가하기 위한 방법으로서 디버거 결함주입 방법에 대한 연구를 진행하였다. 기존에 시행되고 있던 결함주입기법들은 별도의 시스템이 필요 하거나, 실험대상 시스템의 파손, 실제 생산 될 제품과 테스트 환경이 다른 점 등 신뢰성 평가를 하기 어려운 점등의 단점들이 있다. 본 논문에서는 기존 결함주입 기법들의 단점들을 극복하고자 디버거 결함주입기법을 연구하였다. 현재 ARM 시리즈 임베디드 보드와 RVDS 디버거를 활용하여 탐색 수준의 결함주입 환경을 구축하고 결함주입 시험을 실시하였다. 향후 PowerPC, OpenRISC, MIPS, OpenSPARC 과 같은 다양한 시스템을 대상으로 결함주입 환경을 구축하고, 결함주입 방법 절차와 자동화에 대한 연구를 하고자 한다. 최종적으로 고신뢰성 관련 국제인증규격인 IEC 61508 및 ISO26262 등 고신뢰성 시스템/소프트웨어 설계에서 요구되는 결함주입 시험환경으로 개발 할 수 있다. 또한 임베디드 시스템의 취약성을 평가하기 위한 시험의 세부 실험방법으로 활용 가능하다.

참 고 문 헌

[1] 남순원, "산업용 Embedded 시스템의 기술 동향과 전망" , Embedded News, 2011

[2] Premkishore Shivakumar, Michael Kistler, etc, "Modeling the Effect of Technology Trends on the Soft Error Rate of Combinational Logic", 2002 International Conference on Dependable System and Networks, 2002

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